

An Adaptive Variational Finite Difference Framework for Efficient Symmetric Octree Viscosity

RYAN GOLDADE and YIPENG WANG, University of Waterloo, Canada
MRIDUL AANJANEYA, Rutgers University, USA
CHRISTOPHER BATTY, University of Waterloo, Canada

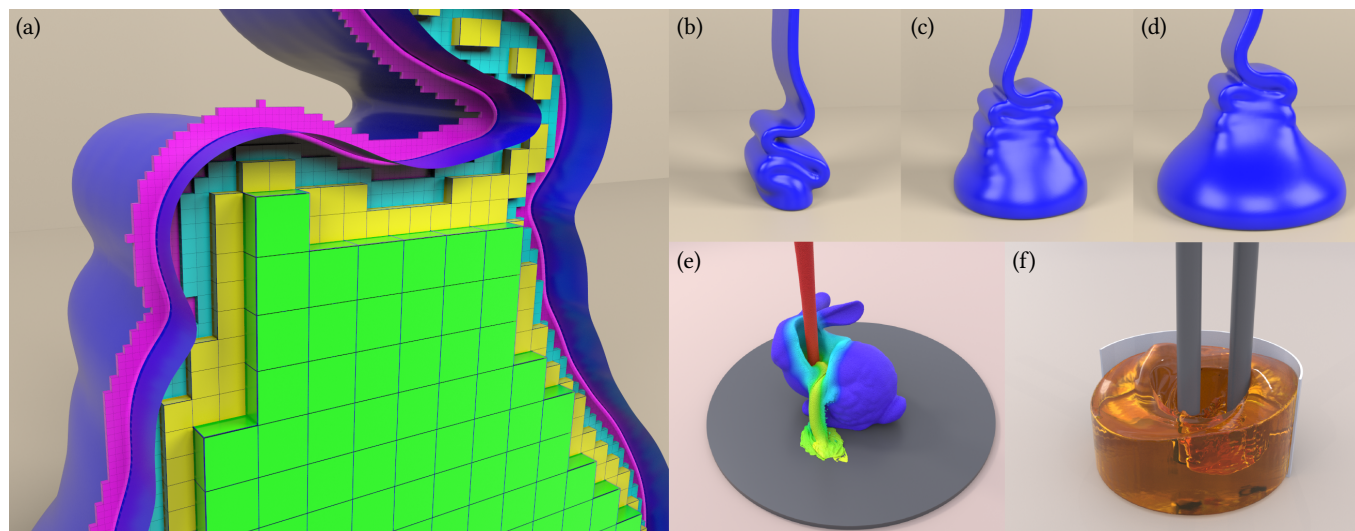


Fig. 1. Our adaptive viscosity discretization is constructed on a graded octree structure (a), and achieves speed-up factors for the linear solve ranging from 3.8 to 9.4 compared to the regular grid approach. Our method supports rotational effects observed with a buckling sheet of viscous liquid (b-d), spatially varying viscosity coefficients (e), and kinematic objects (f).

While pressure forces are often the bottleneck in (near-)inviscid fluid simulations, viscosity can impose orders of magnitude greater computational costs at lower Reynolds numbers. We propose an implicit octree finite difference discretization that significantly accelerates the solution of the free surface viscosity equations using adaptive staggered grids, while supporting viscous buckling and rotation effects, variable viscosity, and interaction with scripted moving solids. In experimental comparisons against regular grids, our method reduced the number of active velocity degrees of freedom by as much as a factor of 7.7 and reduced linear system solution times by factors between 3.8 and 9.4. We achieve this by developing a novel adaptive variational finite difference methodology for octrees and applying it to the optimization form of the viscosity problem. This yields a linear system that

Authors' addresses: Ryan Goldade, rgoldade@uwaterloo.ca; Yipeng Wang, University of Waterloo, 200 University Ave W., Waterloo, ON, N2L 3G1, Canada, yipeng.wang@uwaterloo.ca; Mridul Aanjaneya, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ, 08854, USA, mridul.aanjaneya@rutgers.edu; Christopher Batty, University of Waterloo, 200 University Ave W., Waterloo, ON, N2L 3G1, Canada, christopher.batty@uwaterloo.ca.

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. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2019/7-ART94 \$15.00
<https://doi.org/10.1145/3306346.3322939>

is symmetric positive definite by construction, unlike naive finite difference/volume methods, and much sparser than a hypothetical finite element alternative. Grid refinement studies show spatial convergence at first order in L_∞ and second order in L_1 , while the significantly smaller size of the octree linear systems allows for the solution of viscous forces at higher effective resolutions than with regular grids. We demonstrate the practical benefits of our adaptive scheme by replacing the regular grid viscosity step of a commercial liquid simulator (Houdini) to yield large speed-ups, and by incorporating it into an existing inviscid octree simulator to add support for viscous flows. Animations of viscous liquids pouring, bending, stirring, buckling, and melting illustrate that our octree method offers significant computational gains and excellent visual consistency with its regular grid counterpart.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: viscosity, liquid, octree, variational, symmetry

ACM Reference Format:

Ryan Goldade, Yipeng Wang, Mridul Aanjaneya, and Christopher Batty. 2019. An Adaptive Variational Finite Difference Framework for Efficient Symmetric Octree Viscosity. *ACM Trans. Graph.* 38, 4, Article 94 (July 2019), 14 pages. <https://doi.org/10.1145/3306346.3322939>

1 INTRODUCTION

Pressure and viscosity are the two fundamental internal forces that govern the motion of all Newtonian fluids; a stable and efficient

numerical treatment of viscosity is therefore required to animate many common liquids, ranging from olive oil and honey to molasses and tar. Over the past two decades effective time-implicit Eulerian methods for this problem have been developed [Batty and Bridson 2008; Carlson et al. 2002; Larionov et al. 2017; Rasmussen et al. 2004] and incorporated into standard animation software, such as Houdini [Side Effects Software 2018]. Unfortunately, compared to computing pressure forces on the same domain, viscous forces can often still be slower by an order of magnitude or more. There are two main reasons: large viscosity coefficients give rise to stiff linear systems that are generally slower to solve, and the boundary conditions necessary for plausible free surface behavior couple the different components of velocity together, yielding a system that is three times larger and contains twice as many non-zeros per row [Batty and Bridson 2008].

A potentially powerful acceleration strategy is the use of *spatial adaptivity*, which locally applies grid refinement to focus resolution on important flow features, such as the liquid surface or solid boundaries. While this strategy has been extensively explored for solving pressure forces [Aanjaneya et al. 2017; Ando et al. 2013; Batty et al. 2010; Brochu et al. 2010; Chentanez et al. 2007; Ferstl et al. 2014; Klingner et al. 2006; Losasso et al. 2006, 2004; Setaluri et al. 2014], very few papers have considered adaptive Eulerian viscosity [Batty and Houston 2011; Hong and Kim 2005]. Unfortunately, the viscosity model used by Hong and Kim does not support realistic rotational or bending motion [Batty and Bridson 2008], and the method of Batty and Houston uses tetrahedral meshes, which entail a significant performance penalty compared to modern octrees [Setaluri et al. 2014]. Moreover, prior work offered neither visual comparisons nor performance evaluations against regular grid simulations, leaving it unclear whether adaptivity is beneficial for viscous liquids, or if it actually harms efficiency and visual quality in practice. Therefore, we propose and evaluate an efficient and practical adaptive viscosity solver for staggered octree-based liquids that supports free surfaces and variable viscosity, while being geometrically compatible with the classic inviscid octree simulator of Losasso et al. [2006; 2004].

A central conceptual contribution of our paper is to develop and validate a new adaptive variational finite difference methodology for fluids, with the viscosity problem considered as a case study. The regular grid variational finite difference framework was first proposed by Batty et al. [2007] for incompressibility and solid-fluid coupling; this approach has since been applied to a variety of regular grid fluid problems [Ando et al. 2013, 2015a; Batty and Bridson 2008; Larionov et al. 2017; Narain et al. 2010]. Our novel generalization to the octree setting requires two core enhancements near T-junctions: first, the definition of modified sample points and control volumes for discretely approximating the necessary integrals; and second, the careful design of accurate adaptivity-aware finite difference stencils for derivative operators appearing in the problem’s variational form (in our case, stencils for velocity gradients). Compared to direct finite difference/volume discretizations, our approach guarantees symmetric positive definite (SPD) linear systems by construction; compared to a (hypothetical) finite element alternative, it yields much sparser linear systems and reduces back to simple finite differences in uniform regions. Furthermore, we confirm the accuracy of the proposed method with comparisons against regular grids and

with refinement studies showing velocity convergence at first order in L_∞ and second order in L_1 . Although popular octree pressure projection schemes in graphics offer second order accuracy in pressure, their velocity accuracy is only first order [Aanjaneya et al. 2017; Losasso et al. 2006]; our viscosity solver therefore offers comparable or better convergence rates.

We demonstrate the practical benefits of our octree viscosity method in two settings. First, we describe how one can replace the viscosity step of a standard regular grid simulator (Houdini) with our octree viscosity solver. This offers speed-up factors up to 9.4 for the linear solve and 8.8 for a full viscosity step, and enables simulation at much higher resolutions than is currently feasible. Second, we augment an existing inviscid octree liquid simulator [Aanjaneya et al. 2017] with our octree viscosity step to expand its functionality.

To summarize, our primary contributions are:

- the introduction of a novel adaptive variational finite difference methodology for octrees that guarantees symmetry;
- an efficient octree viscosity solver based on this methodology that handles free surfaces and variable coefficients;
- numerical experiments confirming convergence of the discretization under spatial refinement;
- the application of our method to (1) dramatically increase the speed or resolution of viscous flow simulations produced with a commercial regular grid simulator, and (2) add viscous effects to an inviscid octree liquid simulator.

2 RELATED WORK

Our work builds on standard staggered grid Eulerian techniques for fluid animation, as summarized by Bridson [2015]; we focus our review below on this category. Nevertheless, highly viscous liquids have also been simulated with various alternative techniques, including smoothed particle hydrodynamics (SPH) (e.g., [Andrade et al. 2015; Peer et al. 2015; Takahashi et al. 2015]) and specialized

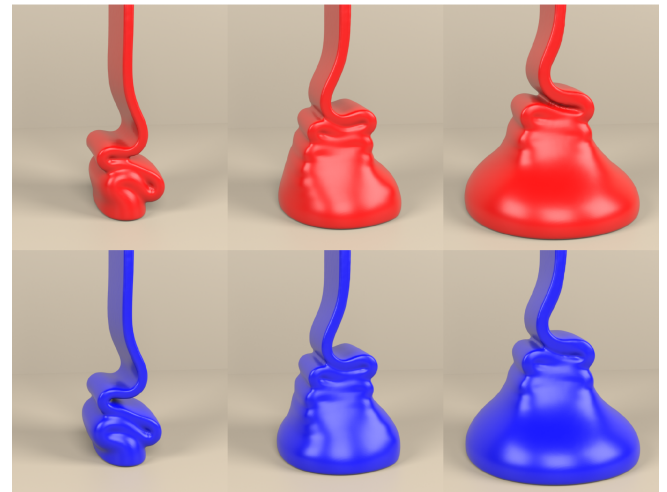


Fig. 2. **Viscous Buckling:** A buckling viscous sheet exhibits qualitatively consistent motion using a regular grid (red) and our octree-based (blue) viscosity solver. The motion matches closely over a long period.

lower-dimensional mesh-based thread and sheet models (e.g., [Batty et al. 2012; Bergou et al. 2010; Zhu et al. 2015]).

2.1 Viscous Liquid Simulation

Foster and Metaxas [1996] presented the earliest 3D implementation of the Navier–Stokes equations for liquids in computer graphics, including explicit time integration of viscosity. Because explicit integration of viscosity is subject to a particularly stringent stability restriction, Stam [1999] modeled viscosity using implicit diffusion (i.e., Laplacian smoothing) of velocity in the context of smoke simulation, and Carlson et al. [2002] applied this model to spatially varying viscosity and liquids with free surfaces. Fält and Roble [2003] used improved boundary conditions to support translation of free-flying liquid bodies, and Rasmussen et al. [2004] proposed an implicit/explicit integration method to properly treat spatially varying viscosity coefficients. Batty and Bridson [2008] developed an implicit variational finite difference viscosity model to allow for rotational motion of viscous free surfaces, thereby enabling realistic buckling and folding along with improved stability for variable viscosity. Larionov et al. [2017] further showed consistent viscous coiling can be achieved by treating viscosity and pressure terms simultaneously via the unsteady Stokes equations. Viscous forces have also been considered for multiphase flows, usually assuming constant viscosity per material [Hong and Kim 2005; Losasso et al. 2006]. Nearly all of the discretizations above yield SPD linear systems, which allow for more efficient numerical solvers; ours does the same.

While we focus on purely Newtonian flows, in which viscous forces are linear with respect to the deformation rate, complex non-Newtonian flows have also been considered on regular grids (e.g., [Goktekin et al. 2004; Losasso et al. 2006; Stomakhin et al. 2014; Yue et al. 2015]); in particular, the material point method (MPM) has gained in popularity. Ram et al. [2015] proposed an Oldroyd-B model for viscoelastic materials, and included a linear viscosity potential. Gao et al. [2017] recently developed an MPM scheme for octrees using carefully designed basis functions with C^1 continuity; their method is restricted to explicit time integration so far. Furthermore, even for regular grid MPM schemes the need for high order basis functions means implicit integration suffers from relatively dense matrices that are inefficient to solve [Yue et al. 2018], motivating users to revert to explicit methods.

More generally, hierarchically adaptive node-based finite element-style elasticity methods (e.g., [Capell et al. 2002; Grinspun et al. 2002]) could hypothetically be applied to our problem by replacing the elastic constitutive law with the fluid viscosity equations. However, our scheme adopts the staggered grids naturally preferred for incompressible fluids [Bridson 2015], it doesn't require higher order basis function constructions, and finite difference schemes can offer significantly sparser stencils even compared to low order finite elements. For example, Zhu et al. [2010] used this last observation to motivate their staggered finite difference scheme for regular grid elasticity. Observe that for a regular 2D staggered grid, only 9 velocity degrees of freedom (DOFs) are involved in a given matrix row (e.g., [Batty and Bridson 2008], Fig. 10); for linear nodal FVM/FEM, each row involves both velocity component DOFs of the

9 surrounding nodes ($9 \times 2 = 18$). This ratio is significantly worse in 3D: 15 staggered DOFs per row vs. 81 nodal DOFs per row.

2.2 Adaptive Viscous Liquid Simulation

2.2.1 Octrees. The first successful *inviscid* octree-based free surface flow solver in computer graphics was proposed by Losasso et al. [2004], although Shi and Yu had earlier proposed a non-symmetric discretization for octree smoke [Shi and Yu 2004]. Losasso et al. developed a symmetric positive definite finite volume Laplacian discretization for the pressure projection and a semi-Lagrangian advection step relying on node-based trilinear velocity interpolation, but omitted viscosity. This formulation relied on an inaccurate sloped gradient approximation, leading to motion errors for hydrostatic scenarios, so they later presented an enhanced version that constructs a single axis-aligned gradient shared by all child faces at a T-junction [Losasso et al. 2006]. To improve its efficiency, Setaluri et al. [2014] proposed a memory-efficient sparse paged grid (SPGrid) data structure that constructs the octree as a hierarchy of sparsely populated regular grids, rather than a standard pointer-based tree. They applied it to smoke simulation, and later Aanjaneya et al. [2017] applied it to liquid simulation using a finite volume power diagram discretization across T-junctions. We incorporate our viscosity discretization into this framework to demonstrate a fully adaptive solver for viscous liquids.

There are relatively few octree-based fluid solvers that specifically address viscosity, particularly in the case of free surfaces and variable coefficients. Hong and Kim [2005] reused the octree Laplacian of Losasso et al. to add implicit viscosity via velocity diffusion, though this simple model precludes support for rotational effects and variable viscosity [Batty and Bridson 2008]. Ferstl et al. [2014] employed a cut-cell finite element method with a multigrid solver for adaptive liquid animation, focusing on inviscid scenarios and pressure projection, but did not demonstrate nor elaborate on their treatment of viscosity. Nielsen and Bridson [2016] alluded to tree-based finite element viscosity in Maya's BiFröst, but omitted details.

Looking beyond computer graphics to computational physics, there exist many adaptive methods on nested regular grids that support viscosity, but most assume spatially constant viscosity without free surfaces (e.g., [Almgren et al. 1998; Guittet et al. 2015; Min and Gibou 2006]). We discuss a few pertinent exceptions. In geophysics, Gerya et al. [2013] proposed an adaptive implicit finite difference discretization for free surface variable viscosity flows, though it is limited to 2D quadtrees. It also treats the free surface using an approximate "sticky air" layer rather than a sharp boundary condition, and yields non-symmetric systems. Nikitin et al. [2011; 2008] presented octree Navier–Stokes solvers that consider the free surface conditions and treat viscosity explicitly, though this can lead to stability issues. Olshanskii et al. [2013] proposed a related implicit discretization, but it does not handle the free surface and its reliance on least-squares fitting to construct adaptive stencils introduces asymmetry. Guittet et al. [2015] achieved an SPD system for the Laplacian form on non-graded trees using a Voronoi diagram of the staggered velocity face samples; however, this comes at the cost of frequent unstructured mesh generation and highly non-local stencils. Moreover, neither Voronoi nor power diagram

discretizations of the Laplace operator can readily treat the cross-component derivative terms that arise for the more general viscosity PDE needed for free surfaces and variable coefficients. Lastly, setting aside viscosity, Horesh and Haber [2011] proposed a non-symmetric finite volume discretization of Maxwell's equations on octrees; their (adjoint of) curl stencils share some geometric similarities with our vector gradient stencils.

2.2.2 Tetrahedral Meshes. Tetrahedral meshes are another alternative for adaptive liquid simulation with either Eulerian or Lagrangian approaches. In the Eulerian setting, Klingner et al. [2006] used adaptive staggered tetrahedral meshes for smoke; Chentanez et al. [2007] extended this approach to liquids. To the best of our knowledge, the subsequent node-based finite volume viscosity scheme of Batty and Houston [2011] is the only prior Eulerian scheme to animate high viscosity liquids on adaptive tetrahedral meshes that supports rotational surface motion and spatially varying viscosity. However, earlier work supported Laplacian-style viscosity on closed domains or used (nearly) uniform-resolution tetrahedral meshes [Bonito et al. 2006; Elcott et al. 2007; Wendt et al. 2007].

The possibility of purely *Lagrangian* tetrahedral liquid animation was first hinted at by the finite element viscoplastic solid approaches of Bargteil et al. [2007] and Wojtan and Turk [2008]. Subsequently, incompressible Lagrangian liquid simulation with surface-conforming tetrahedral meshes was achieved by Mistzal et al. [2010; 2014] and Clausen et al. [2013], including natural handling of viscous free surfaces. However, while tetrahedral methods flexibly support adaptive viscous flows, the computational expense, complexity, and memory overhead of accessing, manipulating, and remeshing such structures makes them less attractive compared to modern optimized octrees [Setaluri et al. 2014].

3 GOVERNING EQUATIONS AND VARIATIONAL FORM

We focus on the equations of fluid viscosity as discussed by Batty and Bridson [2008]. Specifically, we seek to solve

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{1}{\rho} \nabla \cdot \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (1)$$

where \mathbf{u} is fluid velocity, ρ is fluid density, t is time, and μ is the viscosity coefficient, which may vary smoothly in space. At solid boundaries we apply a no-slip condition given by $\mathbf{u} = \mathbf{u}_{\text{solid}}$. We apply a zero traction condition given by $\mathbf{t} = \tau \mathbf{n} = \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \mathbf{n} = 0$ at free surfaces, where \mathbf{t} is the surface traction vector, \mathbf{n} is the free surface normal, and $\tau = \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ is the viscous shear (or deviatoric) stress tensor.

Batty and Bridson [2008] observe that the solution to a step of backward Euler on the PDE (1) can be expressed as the vector \mathbf{u} that minimizes the smooth energy

$$\iiint_{\Omega} \left(\frac{\rho}{2\Delta t} \|\mathbf{u} - \mathbf{u}^*\|_2^2 + \mu \left\| \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} \right\|_F^2 \right) dV \quad (2)$$

where Ω is the liquid domain, $\|\cdot\|_F$ indicates the Frobenius norm, and \mathbf{u}^* is the velocity before integrating viscosity. The natural boundary conditions enforce the free surface condition automatically, while the simpler no-slip solid boundary condition can be applied explicitly to the discrete equations.

3.1 Choice of viscosity model

Support for realistic rotational free surface motion and spatially varying viscosity necessitates the coupling of the velocity components through the cross-derivative terms in $(\nabla \mathbf{u})^T$; this makes the PDE more challenging than a simple Poisson problem or heat equation, for which a componentwise Laplacian suffices. As observed by Larionov et al. [2017], the viscosity model of Batty and Bridson [2008] strikes a middle ground between the simpler and more efficient Laplacian-style viscosity model [Carlson et al. 2002; Fält and Roble 2003; Hong and Kim 2005], which does not faithfully handle bending or variable viscosity, and Larionov's significantly more costly Stokes model, which tightly couples pressure and viscosity to recover viscous rope coiling and slightly stiffer surface details. Given these considerations and since this compromise model is widely adopted in commercial tools like Houdini, we consciously selected it as the basis for our work. Nevertheless, we anticipate that our framework will be applicable to the other two models, and indeed our results confirm this for the Laplacian model (Figure 3).

Our chosen model also adopts the standard operator-splitting approach to treat advection, viscosity, and pressure projection in distinct decoupled stages [Bridson 2015; Stam 1999]. While this step-and-project approach entails some loss of accuracy (e.g., [Larionov et al. 2017; Mullen et al. 2009]), it is more efficient than a simultaneous solution and effective in many relevant scenarios.

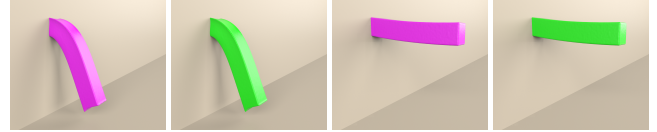


Fig. 3. The same frame for different discretizations of an initially horizontal viscous beam collapsing. Left pair: Our chosen viscosity model ([Batty and Bridson 2008]) naturally supports rotation and bending: (magenta) regular grid; (green) our octree framework. Right pair: The natural boundary conditions of simpler Laplacian models ($\mu(\nabla \mathbf{u})\mathbf{n} = 0$) inhibit bending, leading instead to excessively stiff shearing. Nevertheless, our octree framework is effective when applied to this model as well.

4 REGULAR GRID VARIATIONAL VISCOSITY

Our octree viscosity approach draws ideas from an earlier variational finite difference discretization of (2) on staggered *regular* grids [Batty and Bridson 2008], which we will briefly review. The quantities to be integrated are approximated at staggered sample points on the grid using centered finite differences for derivatives. Each sample point's contribution is scaled by the liquid volume within its surrounding cubic control volume. These contributions are summed to give the total discrete energy, which is then minimized with respect to velocity.

In the illustrations that follow, we have color-coded face-centered velocities u, v, w with red, green, and blue squares (Figures 4(a,b),5(a-c)), cell-centered stresses $\tau_{xx}, \tau_{yy}, \tau_{zz}$ with cyan, magenta, and yellow diamonds/octahedra (Figures 4(c,d),5(a-c)), and edge-centered stresses $\tau_{xy}, \tau_{yz}, \tau_{xz}$ with gray spheres (Figure 5(d-f)). Nodal stresses τ_{xy} in two spatial dimensions are shown as black disks (Figure 4(e)).

Finite difference stencils are indicated with colored arrows, color-coded by axis and pointing from the discrete value locations to the derivative locations. Gray arrows are used to indicate averaging.

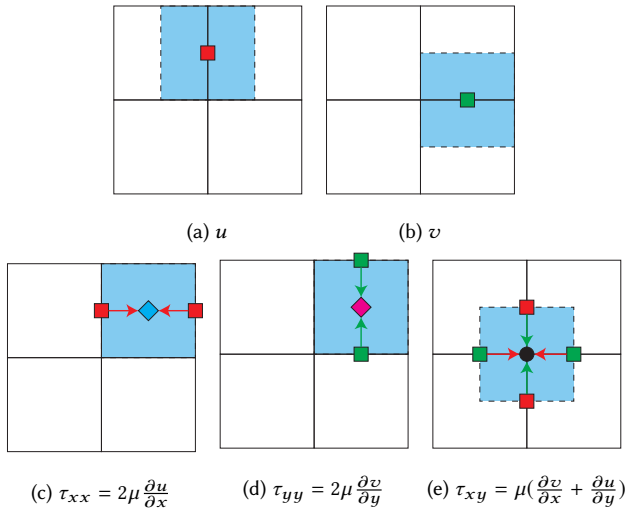


Fig. 4. Staggered regular grid variable locations for velocities (a,b) and stresses (c-e) in two spatial dimensions, with corresponding control volumes shaded in light blue. Colored arrows indicate the stencils for stresses.

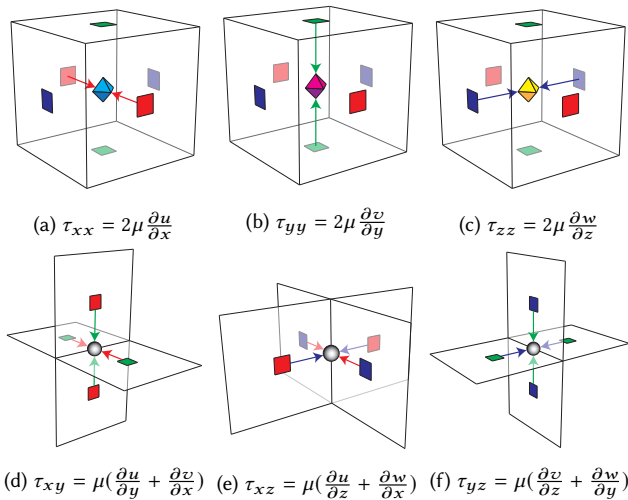


Fig. 5. Staggered regular grid variable locations for face velocities in red, green, blue, cell-centered stresses in cyan, magenta, yellow (a-c), and edge-centered stresses in gray (d-f) in 3D. Arrows indicate stress stencils.

Figure 4 illustrates this approach for viscosity in 2D. The components of the integral's first term are evaluated at the staggered velocity samples; the second term contributions lie at cell centers and nodes, since they relate to components of the stress tensor τ arising from centered differencing of velocities. This approach extends directly to 3D (Figures 5 and 6) with velocity samples on grid faces

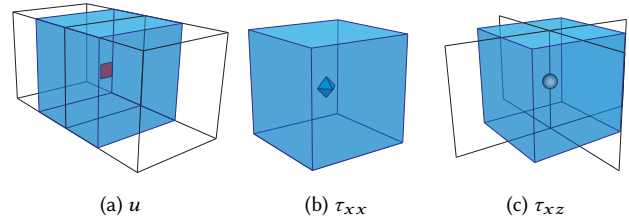


Fig. 6. Control volumes for 3D velocity and stress, shaded light blue.

and velocity gradient (or stress) stencils located at grid cell centers and edges. (Since stresses are essentially scaled velocity gradients, we will use the terms 'stress stencil' and 'velocity gradient stencil' interchangeably.) The resulting discrete energy in matrix form is

$$\frac{1}{2\Delta t} (u - u^*)^T W_u P (u - u^*) + u^T D^T K M W_\tau D u \quad (3)$$

where P is a diagonal matrix of per-velocity densities, W_u is a diagonal matrix of liquid volumes per velocity control volume, W_τ is a diagonal matrix of liquid volumes per stress control volume, Δt is the time step, D is the finite difference deformation rate operator such that $Du \approx \frac{\nabla u + (\nabla u)^T}{2}$, M is a diagonal matrix of viscosity coefficients, and K is a diagonal matrix of scale factors that doubles the contribution of cross-derivative terms to produce the correct Frobenius norm (e.g., in 2D $\|\tau\|_F^2 = \tau_{xx}^2 + 2\tau_{xy}^2 + \tau_{yy}^2$). Because the optimization problem is quadratic in u and convex, differentiating and equating to zero yields a symmetric positive definite system,

$$(W_u P + 2\Delta t D^T K M W_\tau D) u = W_u P u^*. \quad (4)$$

The resulting discrete equations match (up to a constant scale factor) a direct finite difference discretization of the PDE on the interior of the liquid, but handle the free surface boundary condition through the volume weight terms without complex special cases. With this groundwork established, we can introduce our new adaptive framework that enables discretizing on octrees.

5 OUR OCTREE VISCOSITY APPROACH

5.1 Adaptive Variational Finite Differences

The variational finite difference framework was proposed for pressure and solid-fluid coupling problems [Batty et al. 2007], and has since been applied to viscosity, stream functions, granular flow, and more [Ando et al. 2015a,b; Batty and Bridson 2008; Larionov et al. 2017; Narain et al. 2010]. The typical advantage of this approach is its simpler handling of difficult irregular boundary conditions on regular Cartesian grids, while reducing to standard staggered finite differences on the interior and preserving symmetric positive-definiteness. Rather than only using it to handle boundary conditions, however, we propose to further apply this variational finite difference perspective to support octree-based adaptivity. Doing so hinges on discretizing the integrals of the variational form in the presence of transitions between grid levels, which in practice involves two key changes near T-junctions: first, selecting appropriate variable locations and control volumes to integrate over, and second, designing finite difference operators to approximate derivative terms. While the variational form ensures symmetry and

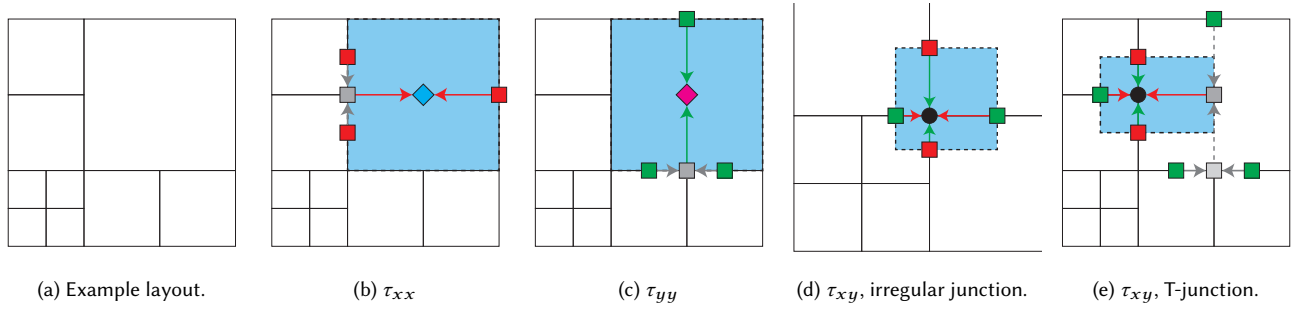


Fig. 7. Quadtree stress placement, stencils, and control volumes in two spatial dimensions. Gray squares indicate ghost samples constructed by averaging.

hints that a valid numerical scheme should exist, these discretization choices must nevertheless be approached carefully if one wishes to achieve consistent, accurate solutions and compact stencils that will be efficient in three dimensions.

To form the discrete viscous energy (3), the only derivative is the deformation rate operator D , which implies that we must modify the finite difference velocity gradient (or stress) stencils. Since symmetry follows from the variational form, the corresponding discrete tensor divergence operator arises implicitly as the (scaled) transpose of D . Similarly, the final linear system for octree viscosity remains positive definite with no additional effort. By contrast, typical direct staggered finite difference/volume approaches on octrees almost invariably lead to asymmetry even for simple Poisson problems, as discussed by previous authors [Batty 2017; Losasso et al. 2004]. Prior derivations of *symmetric* octree schemes for Poisson problems relied on a combination of subtle intuition and trial and error [Aanjaneya et al. 2017; Losasso et al. 2006, 2004]. Our more systematic adaptive variational finite difference framework instead preserves symmetry naturally and, with judicious discretization choices, yields accurate, efficient solutions on the challenging variable viscosity PDE.

For simplicity of implementation, similar to prior work [Aanjaneya et al. 2017; Ferstl et al. 2014], we assume that the octree is *graded*, i.e., cells sharing a face differ by no more than one level of resolution. (Note that this does not preclude cells that share only a node or edge from differing by more than one level, which allows for slightly more rapid coarsening.) We further assume that the free surface occurs only within the finest grid and that resolution changes occur only in the interior of the liquid; this avoids any potential interactions between T-junctions and boundary conditions. In regions of the octree without resolution changes, the variable locations, control volumes, and stencil structures all follow the template of the regular grid method described in Section 4, adjusting for the cell size in the corresponding level of the octree. We now adapt this regular grid template to discretize (4) near T-junctions, beginning with the simpler two dimensional case.

5.2 Two Dimensions

5.2.1 Variable Locations and Control Volumes. Our 2D quadtree layout is designed to follow naturally from the 2D regular grid setting presented in Figure 4. Similar to Losasso et al. [2006; 2004], we place velocity samples on each fine grid face at level transitions (see inset

figure). Similarly, we place stresses at cell centers and nodes, including the new T-junction (“dangling”) nodes arising at level transitions. To ensure the velocity and stress control volumes for each variable separately cover the entire integrable fluid volume, we stretch the control volumes into the adjacent coarse-grid cells at T-junctions (see inset figure and Figure 7). Unlike the regular grid case in which absolute cell volumes can be completely factored out such that volume fractions suffice, all quadtree control volumes must be appropriately scaled to reflect the relative grid cell size at the corresponding level of the quadtree.

The rectangular control volumes shown can partially overlap in one case at a T-junction. We tested a correction that modifies the control volume shapes to avoid this double-counting, but it had no discernible effect on either the observed convergence rate or the visual results, so we prefer the rectangular volumes for simplicity.

5.2.2 Finite Difference Stencils. To compute stresses at level transitions, we desire adaptive grid finite difference velocity gradient stencils that mimic the regular grid stencils of Figure 4; once again, we need not explicitly discretize the tensor divergence operator, as it arises implicitly through symmetry. We will consider cell-centered and node-centered stress stencils in turn.

Since only fine velocity samples are present on transition faces, coarse cell-centered stresses (τ_{xx} , τ_{yy}) incident on T-junctions will not have coarse velocity samples that align with their regular finite difference stencil. As shown in Figures 7(b) and 7(c), we address this by creating a coarse “ghost” velocity sample on the T-junction, which can be used in the regular stencil. This ghost sample does not exist as a real degree of freedom, but is simply an interpolated value constructed as the average of the two fine velocity samples. The remaining cross-derivative node-centered stresses (τ_{xy}) at transitions are of two types: *irregular* junctions, where different resolutions meet diagonally (Figure 7(d)), and *T-junctions* (Figure 7(e)).

Irregular junction stencil. The irregular junction stencil in Figure 7(d) is similar to the regular grid case (Figure 4(e)), except that the velocity samples are no longer evenly spaced. We simply adjust the denominator of the finite difference estimate to reflect the distance between the sample points. While no longer a true centered

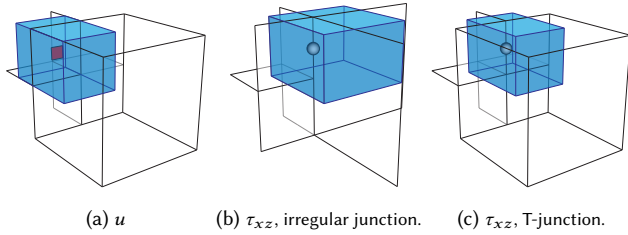


Fig. 8. Octree velocity samples, stress placement, and control volumes.

difference, this does not pose an issue: our linear system remains symmetric, and the 2D numerical results in Section 7.3 confirm the convergence of our discretization.

T-junction stencil. The T-junction stresses encounter a problem similar to the cell-centered stresses: the finite difference stencil required to compute the $\frac{\partial v}{\partial x}$ component in Figure 7(e) does not find a velocity sample when reaching out into the neighboring coarse cell. Once again, we create a ghost velocity sample by averaging the v -velocity faces of the coarse cell. However, a further wrinkle can arise if either one or both of the cells above and below the coarse cell are also subdivided. Figure 7(e) shows the case of one coarse face and two fine faces in the v -velocity direction. In such cases, we average the velocity values at the two fine faces to create an *intermediate* ghost velocity sample, and then average this ghost sample with the opposing coarse velocity sample to create a final ghost sample that completes the stencil.

Discussion. Because only a discrete velocity gradient D is required, our symmetry-preserving method possesses an attractive degree of conceptual simplicity. By contrast, Gerya et al. [2013] applied direct finite difference constructions separately to the vector gradient and tensor divergence operators on 2D quadtrees and arrived at a variety of possible viscosity discretizations, all of which exhibit asymmetry.

5.3 Three Dimensions — Basic Method

We now describe the essentials of our new discretization in three dimensions (i.e., on octrees), deferring an additional key accuracy enhancement to Section 5.4. Despite the apparent complexity of the task, we will show that the necessary stencil set reduces to a few classes of 3D configurations, which are unique up to appropriate rotations and reflections.

5.3.1 Variable Locations and Control Volumes. Similar to the quadtree case, velocity samples are placed at all fine faces surrounding T-junctions (Figure 8(a)). Stresses are placed at all cell centers and (fine) edge midpoints (Figures 8(b) and 8(c)); this creates four *edge-based* stresses per T-junction in 3D, as compared to the single nodal stress per T-junction in 2D. These figures also illustrate how the rectangular cuboid control volumes for fine cell samples are expanded into the adjacent coarse cells. As in 2D, these control volumes overlap slightly in some cases, but modifying the control volumes to correct for this did not affect the observed motion or convergence rate.

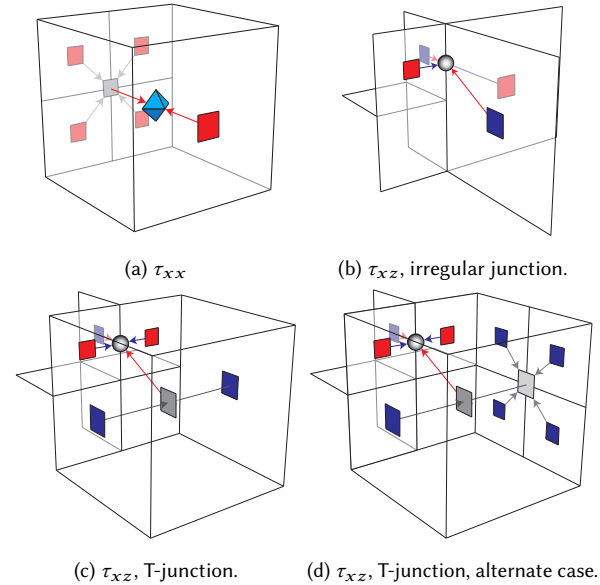


Fig. 9. Octree stress stencils for cell-centered and edge-centered locations. Gray squares indicate ghost velocity samples constructed by averaging.

5.3.2 Finite Difference Stencils. In constructing our 3D velocity gradient stencils, we aim to prioritize simplicity and compactness for the sake of ease of implementation and computational efficiency. Our stencil for the cell-centered stress τ_{xx} at a level transition is shown in Figure 9(a). This stencil requires creating ghost velocity samples at coarse velocity positions on T-junctions by averaging the four inset fine velocity samples. For cross-derivative edge-centered stresses, τ_{xz} , there are two basic scenarios, with similarities to their 2D counterparts: irregular junction stresses, on edges that are shared *diagonally* between coarse and fine cells (Figure 9(b)), and T-junction stresses, for edges that lie on a subdivided coarse face (Figure 9(c)).

Irregular junction stencil. Figure 9(b) shows the stress stencil for an irregular junction. In contrast to the 2D case, the stress sample does *not* lie in the same plane as the necessary velocity samples, leading to sloped velocity gradient estimates. We revisit this issue in Section 5.4, but for now we note that this is consistent with the sloped gradients for pressure adopted by Losasso et al. [2004] and that it does converge in practice. This is the only irregular junction case to consider; although face-grading allows cells sharing an edge diagonally to differ by two levels, the actual faces adjacent to the edge can only differ by one, which is covered by our stencil.

T-junction stencil. Figure 9(c) shows the τ_{xz} stress stencil on a T-junction face; it likewise has one sample out of alignment so we treat it with a similar sloped stencil. However, there is also no coarse velocity sample at the center of the coarse cell, as required for the $\frac{\partial w}{\partial x}$ finite difference stencil. Therefore, we create a ghost velocity sample by averaging the two opposing w -velocity faces of the coarse cell. In Figure 9(d), we illustrate a slight variation in which one of the two coarse faces is also subdivided. In this case, we create another intermediate ghost velocity sample at the coarse face,

which is subsequently averaged with the other coarse w -velocity sample (which may itself be a ghost sample as well) to generate the final cell-centered ghost sample. This completes our 3D stencil set.

Losasso et al. [2004] discussed a few choices for the distance value used for the denominator of their sloped gradient estimates, and observed that they are all equally effective; we used the in-plane distance between the sample points. Despite these sloped gradient estimates, our 3D numerical results exhibit approximately first order convergence under grid refinement (Section 7.3).

5.4 Three Dimensions — Enhanced Gradients

Using the method described so far we observed both numerical convergence and plausible viscous flows; however, we can improve the discretization further still. Close inspection of the fluid motion shows artificial damping of bending and rotation compared to regular grid simulations (see Figures 11(a) vs. 11(b)), and we traced the source of this issue back to the sloped gradient estimates. Prior authors studying adaptive discretizations for the Poisson problem also observed that orthogonality of *pressure* gradient stencils with respect to grid faces is critical to accuracy, especially in hydrostatic cases [Aanjaneya et al. 2017; Batty et al. 2010; Losasso et al. 2006]. These observations motivate our enhanced orthogonality-preserving construction for discrete *velocity* gradients.

Sloped gradients occur for stress variables on the midpoint of fine edges at resolution transitions, both in the irregular and T-junction cases (i.e., Figures 9(b)-9(d)). A 2D cross-section is shown in Figure 10(a). Letting Δ_L be the large cell width, Δ_s be the small cell width, and $\Delta = (\Delta_L + \Delta_s)/2$, then our sloped estimate for the top vertical edge is $\frac{\partial u}{\partial z} \approx (u_c - u_a)/\Delta$. A direct approach to form the desired axis-aligned gradients would be to apply additional (and in some cases nested) linear interpolations to yield fine ghost values u_{g1} and u_{g2} directly opposite each existing fine velocity sample (Figure 10(b)). This would suffice to provide an accurate estimate $(u_{g1} - u_a)/\Delta$ for the top edge, while symmetry would be preserved by construction. Unfortunately, forming u_{g1} requires complicated stencils involving additional neighboring velocities and several geometric cases depending on the local refinement pattern. We offer a simpler and more compact solution.

We average the two fine face velocity components u_a and u_b together to create a coarse ghost velocity $u_g = (u_a + u_b)/2$ at the midpoint between them (Figure 10(c)); this enables a standard finite difference gradient estimate $\frac{\partial u}{\partial z} \approx (u_c - u_g)/\Delta$ between the existing coarse velocity u_c and the new ghost u_g . Because they lie in the same plane, this estimate is properly axis-aligned and measures (only) the correct component of the velocity gradient, unlike the sloped estimate. The stencil is also nearly as compact as the sloped estimate, requiring just a single additional velocity sample.

The velocity gradient components assigned to the two fine edges in this way share the same axis-aligned gradient estimate from the node. This approach is therefore conceptually similar to the method that Losasso et al. [2006] adopted to replace their earlier sloped pressure gradients, i.e., they construct a single axis-aligned gradient estimate at the center of the T-junction face, and assign it to all four surrounding fine faces. (However, it differs in that we construct

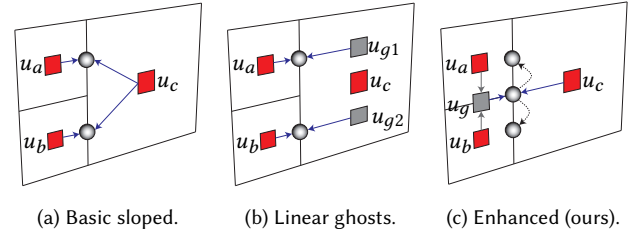


Fig. 10. Possible velocity gradient stencils at irregular junctions and T-junctions, shown as 2D slices of the 3D geometry. (a) Our basic method (§5.3) uses a low-quality sloped estimate. (b) One improved option would be to construct axis-aligned fine ghost points; however, the necessary additional linear interpolations (not shown) would involve several more cases and neighbor cell data. (c) In our proposed enhanced method (§5.4), fine face components are averaged together to create a coarse ghost; an axis-aligned estimate is then constructed at the node and used for *both* fine edges.

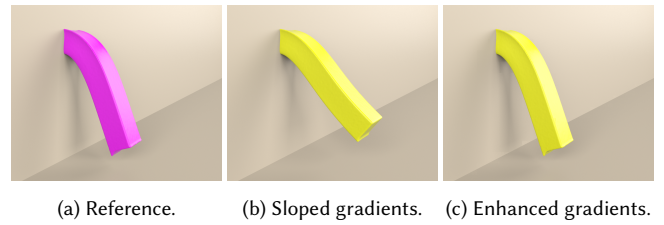


Fig. 11. The same frame of animation from three discretizations of a horizontal viscous beam released under gravity. (a) A regular grid reference simulation. (b) Our *basic* octree discretization with a two-level coarsened interior shows artificial stiffness due to sloped gradients. (c) Our *enhanced* octree discretization with corrected gradients matches the motion and detail of the reference (a), but its linear solve is 4.2 times faster.

orthogonal *edge-based velocity* gradients, whereas Losasso et al. construct *face-based pressure* gradients.) To implement this change, we modify the D operator accordingly and our method again guarantees symmetry by construction. This small but vital enhancement effectively eliminates spurious discretization-dependent stiffness, as shown in Figures 11(c) and 14, while the spatial convergence rate improves to second order in the L_1 norm (Figure 19).

6 IMPLEMENTATION AND APPLICATIONS

We integrated our proposed octree-based viscosity solver into two completely independent liquid simulators. To demonstrate that our method can significantly accelerate existing regular grid simulation pipelines, we implemented it as a plugin for Houdini [2018]; it acts as a drop-in replacement for Houdini’s viscosity step, leaving the rest of its regular grid simulator untouched. To demonstrate that our method can alternatively be used to add viscous effects to existing inviscid, purely octree-based liquid simulators, we incorporated it into the method of Aanjaneya et al. [2017].

In both implementations, we follow Setaluri et al. [2014] in constructing our octree as an aligned pyramid of regular grids instead of a conventional pointer-based tree. This allows querying neighboring elements (cells, faces, edges, nodes) at differing refinement levels with simple grid index offsets and scaling, eschewing pointer-based

traversals. This structure also provides localized memory accesses within a single level. We use sparse grid structures to restrict memory allocations to active regions of each level. For our Houdini implementation, we use its internal compressed-tile design as a sparse grid structure, and for our implementation of Aanjaneya’s method [Aanjaneya et al. 2017], we use SPGrid.

Both implementations also employ the standard optimization of constructing the final system (4) in a single pass, rather than forming and multiplying the individual sparse matrices that compose it. This is more involved than for regular grids, since the transposition D^T requires “reversing” the velocity gradient stencils whose transposes implicitly yield the discrete tensor divergence. Nevertheless, we found it to be much faster than the matrix multiplication approach.

6.1 Accelerating Regular Grid Simulators

Viscosity is a significant bottleneck in regular grid simulators: for some scenes using Houdini the viscosity step was up to two orders of magnitude slower than pressure projection. Fortunately, viscous flows possess smooth interior velocity fields, making this an ideal setting for spatial adaptivity without noticeable quality loss. This motivated us to accelerate Houdini’s regular grid fluid solver by replacing its viscosity step with our octree-based version. To do so, we sandwich our method between a pair of interpolation operators that transfer velocities between the regular and octree grids. The smoothness of the interior flow also enables a relatively simple refinement strategy: we set a band of finest resolution regular grid cells near interfaces and boundaries, and coarsen into the interior as rapidly as possible while respecting the face-grading rule.

6.1.1 Interpolation. We align the finest octree level with the source regular grid, allowing finest level velocity samples to be directly copied into the octree. We construct interior coarsened velocities in the octree by recursive weighted averaging of the next finer level samples, using the restriction operator proposed by Zhu et al. in the context of multigrid elasticity ([Zhu et al. 2010], §6). To transfer the octree velocity data back to the regular grid, we adopt the low-dissipation octree velocity interpolant of Setaluri et al. [2014]. We confirmed that interpolating to and from the octree induces a negligible amount of additional artificial dissipation for viscous flows by adding a round-trip grid-to-octree-to-grid transfer(-only) step to a regular grid solver, and comparing it against a standard regular grid solver on a beam bending scenario; any differences were visually indiscernible. The close match between our octree and regular grid results (e.g., Figure 14) further confirms this fact.

6.1.2 Additional Details. We use Houdini’s own multithreaded implementation of Jacobi-preconditioned conjugate gradients to solve the linear system in equation (4). Likewise, we used Houdini’s multithreaded library to iterate over its sparse grid structure in parallel. Lastly, again following Houdini, we exploited parallelism throughout our implementation, including during octree adaptation, both interpolation steps, and linear system construction.

6.1.3 Laplacian model. The speed of the simpler Laplacian model of viscosity [Carlson et al. 2002; Fält and Roble 2003] may be desirable in some cases, despite the clear limitations illustrated in Figure 3. Hong and Kim applied this model to octrees using Losasso’s discrete

octree Laplacian [Losasso et al. 2004], but this requires interpolating staggered face velocities to cell centres and back to faces. Our methodology enables a purely face-based Laplacian viscosity discretization by applying it to the simplified smooth energy

$$\iiint_{\Omega} \left(\frac{\rho}{\Delta t} \|\mathbf{u} - \mathbf{u}^*\|_2^2 + \mu \|\nabla \mathbf{u}\|_F^2 \right) dV, \quad (5)$$

which can be compared with (2). This form leads to one SPD linear system per velocity component.

6.2 Adding Viscosity to an Octree Liquid Simulator

We also incorporated our octree viscosity solver as a plugin into the fully adaptive *inviscid* simulator of Aanjaneya et al. [2017]. This required the minor modification that velocity values be interpolated from the slanted faces of the power diagram to the regular octree faces, and then interpolated back again after the viscous update.

7 RESULTS AND DISCUSSION

7.1 Regular Grid Plugin Examples

7.1.1 Timings. To evaluate the efficiency and effectiveness of our octree-based viscosity plugin for Houdini, we simulated a variety of viscous scenarios involving buckling, rotation, variable viscosity, and moving solids. The *Bunny Drop* and *Viscous Buckling* examples and the performance comparison in Figure 16 were simulated with a 32-core Ryzen 2990wx CPU; all remaining examples were simulated with a 16-core Xeon E5-2630 CPU. The conjugate gradient routine used double precision and a relative tolerance of 10^{-3} for all examples. We emphasize that only Houdini’s viscosity step was replaced with an octree method; every other step remains on a regular grid using Houdini’s standard implementation. Four-level octrees were used in all cases except where stated otherwise, since additional coarse levels yielded minimal benefit (see Section 7.1.5). To eliminate extraneous factors and ensure a consistent baseline, we implemented and used our own purely regular grid viscosity step [Batty and Bridson 2008] for all comparisons. (Performance profiling showed that our regular grid implementation was approximately 2% slower than Houdini’s on a representative scene; the simulation results were indistinguishable.)

Performance numbers for the octree and regular grid are presented in Table 1, with the octree linear solve being faster by factors ranging from 3.8 to 9.4. As expected in comparing the *overall* time between the regular grid and octree implementations, the additional overhead of the octree slightly reduces its net benefit: the total viscosity solve, including octree adaptation, interpolation, and matrix construction, yields speed-up factors of 3.5 to 8.8. Although viscosity is just one component of the complete liquid simulator, *overall* speed-up factors ranged from 2.4 to 6.3. In one case the simulation time dropped from more than a full day to just over four hours, which is a far more practical turn-around time. It should be noted that these speed-up factors are only available for scenes where the regular grid simulation succeeded; nevertheless, the octree enables simulations at even higher effective resolutions than are possible with the regular grid, as demonstrated by the *Bunny Drop* example.

7.1.2 Viscous Buckling. Figure 2 shows that our method can reproduce the familiar viscous buckling phenomenon. The octree and

Table 1. Timing breakdowns for our simulations on regular and octree grids. Only active voxels are included in the voxel counts.

Scene	Regular grid voxel count	Octree grid voxel count	Grid type	Linear solve	Solve speed-up	Viscosity total	Viscosity speed-up	Simulation total	Simulation speed-up
Viscous Buckling	137K (initial) - 2.7M (final)	76K (initial) - 587K (final)	Regular	3h47m		3h50m		4h32m	
			Octree	59m48s	3.8×	1h05m	3.5×	1h47m	2.5×
Viscous Beam	757K	227K	Regular	2h30m		2h31m		2h40m	
			Octree	35m34s	4.2×	37m20s	4×	46m35s	3.4×
Melting Bunny	2.2M (initial) - 3.2M (final)	422K (initial) - 1.3M (final)	Regular	50h05m		50h19m		52h43m	
			Octree	13h11m	3.8×	13h32m	3.7×	15h25m	3.4×
Letter Mixer	139K (initial) - 4.8M (final)	65K (initial) - 952K (final)	Regular	71h20m		72h59m		96h23m	
			Octree	12h56m	5.5×	15h27m	4.7×	39h02m	2.5×
Goey Armadillo	4.8M	976K	Regular	26h20m		26h28m		27h55m	
			Octree	2h49m	9.4×	3h00m	8.8×	4h25m	6.3×
Bunny Drop	9.0M	1.1M	Regular	—		—		—	
			Octree	73h01m	N/A	76h32m	N/A	96h00m	N/A

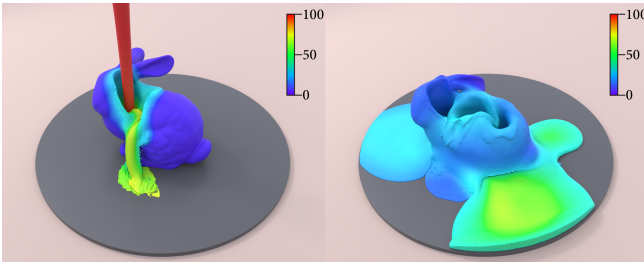


Fig. 12. **Melting Bunny**: Hot liquid is poured onto a viscous bunny, melting holes into it. The viscosity is a function of the liquid temperature, visualized with a pseudocolor map. Linear solve speed-up factor: 3.8.

regular grid results match closely throughout the simulation despite the octree’s lower computational cost, with only slight drifts out of phase. As demonstrated in the accompanying video, increasing the width of the fine grid cell layer at the liquid surface offers a small perceptible improvement towards matching the regular grid example, but it comes with a significant computational cost. The viscosity steps for two-, three-, and four-voxel wide fine layer examples took 1h05m, 2h05m, and 2h25m with a DOF count in the final frame of 1.9M, 2.2M, and 2.7M, respectively.

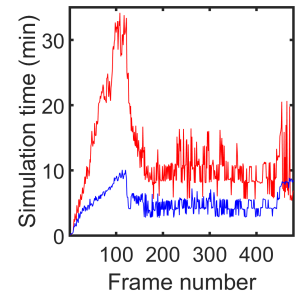
This example also illustrates that the degree of timing improvement is inherently problem- and geometry-dependent, since coarsening becomes possible only with volumes of sufficient depth. Because the volume of the pile gradually increases, the benefit increases in correspondence. In the initial frame, the two-voxel fine layer octree reduced the active DOFs in the viscosity solve from 412K to 277K, while the last frame went from 8.2M velocity samples to 1.9M.

7.1.3 Melting Bunny (Variable Viscosity). Next, we melt a highly viscous bunny by pouring hot liquid onto it, showcasing our variable viscosity support (Figure 12). We define the viscosity coefficient as a function of per-particle temperatures. We mimic heat diffusion at each timestep by transferring temperature from the particles to the

grid, applying a simple blurring pass over the grid-based temperature field, and finally transferring the updated temperature back to the particles (more intricate thermodynamics could be incorporated if desired, e.g., [Carlson et al. 2002; Stomakhin et al. 2014]).

7.1.4 Letter Mixer. Viscous letters are piled up and mixed together with scripted moving solids in Figure 13. A standard no-slip condition applied to the discrete viscosity equations forces the fluid velocity to match the solid, causing the viscous liquid to be dragged alongside moving solids.

This example also highlights conditions under which we see large benefits (see inset figure). Since strong viscous forces are working hard to oppose gravity when all of the letters are fully stacked up, these earlier frames are approximately three times as costly for the regular grid (red) and twice as costly for the octree (blue) compared to when the letters have settled into the container. At that point, the velocity field is no longer changing dramatically between time steps so *warm-starting* the solver is very effective and both regular grid and octree methods reach convergence more quickly. Regardless, the octree-based simulation’s total time is faster by a factor of 2.5.



7.1.5 Viscous Beam. High frequency velocity modes are quickly damped out in viscous liquids, resulting in a smooth field with small variation across many grid cells. Because of this, we observe in Figure 14 that our enhanced octree discretization experiences no visually significant difference in rotation rates compared to regular grids on a viscous beam test. The basic (sloped) discretization is significantly more damped due to the poor gradient estimates.

Examining the timings for these beam tests reveals that the majority of the performance improvement is often achieved after only

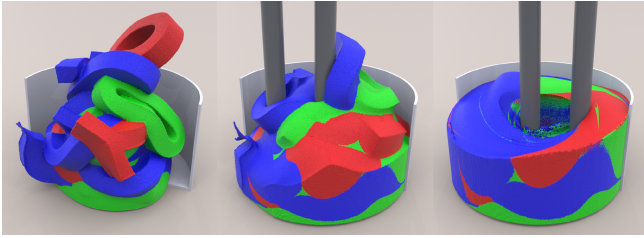
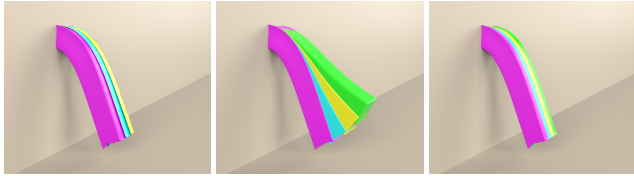


Fig. 13. **Letter Mixer:** Viscous letters are piled into a container and stirred by scripted moving solids. Linear solve speed-up factor: 5.5.



(a) Regular grids. (b) Sloped gradients. (c) Enhanced gradients.

Fig. 14. **Viscous Beam:** (Left) Overlaid strictly regular grid viscous beams released under gravity exhibit no significant differences in bending rates at different resolutions (●=base simulation, ●=half-resolution, ●=quarter-resolution). (Middle) With our basic sloped gradient discretization, interior-coarsened octrees yield noticeable over-stiffening (●, ●, ●, ● indicate 0, 1, 2, and 3 levels of interior coarsening). (Right) With our enhanced axis-aligned gradients (same color-coding), the octree simulations are dramatically improved, closely matching the reference regular grid results in magenta. Linear solve speed-up factor for the coarsest octree: 4.2.

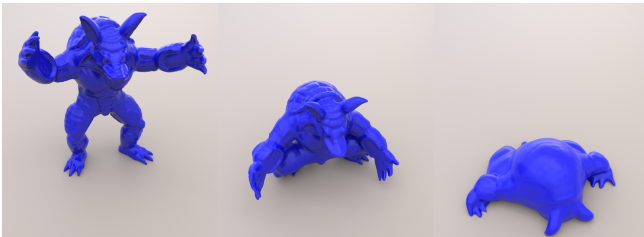


Fig. 15. **Goopy Armadillo:** A highly viscous armadillo released from a standing position slowly collapses. Linear solve speed-up factor: 9.4.

a single level of interior coarsening. The viscosity step in the magenta (regular grid) simulation averaged 75.8 seconds per frame, cyan (1-level coarsened) averaged 27.6 seconds, yellow (2-levels) averaged 19.8 seconds, and green (3-levels) averaged 18.7 seconds. This reflects a natural diminishing of returns: every subsequent step leaves far fewer active DOFs available to be coarsened.

7.1.6 Goopy Armadillo. In Figure 15 a strongly viscous armadillo is released from a standing position and slowly collapses. The large yet compact liquid volume, strong viscous forces, and high degree of deformation under gravity make this our most accelerated example: we see a speed-up factor of 9.4 for the linear solve and 8.8 for the entire viscosity solve. Furthermore, because the viscosity solve

is a significant bottleneck in this scenario, the entire simulation performance improved by a factor of 6.3.

Figure 16(a) plots the linear solve time as a function of fine-grid (effective) resolution for a single frame in the armadillo example for the regular grid (red) and octree (blue) methods, illustrating that our method is more beneficial for higher resolution problems. For the same resolution the smaller systems generated by our method also require fewer iterations, as shown in Figure 16(c). Convergence of the conjugate gradient routine stalled completely at around 14M DOFs for the regular grid, even when periodically recomputing the residual to reduce round-off effects. By contrast, the octree system succeeded up to 140M regular DOFs, i.e., even beyond the range of the graphs. If the regular grid viscosity had converged at higher resolutions, these performance trends suggests that octree viscosity would yield even larger speed-up factors. Figure 16(b) shows that the overhead of both discretizations entails only a small additional computational cost compared to that of the linear solver and is not much worse for the octree.

7.1.7 Bunny Drop. Our proposed method can simulate viscous liquids at higher (effective) grid resolutions than the regular grid. In Figure 17, a very fine grid resolution is necessary to capture the collision between the falling viscous bunny and the two stationary thin wires. The corresponding linear system for the regular grid is so challenging that conjugate gradient fails to converge. Our method significantly reduces the size of the linear system from 27.9M DOFs for the regular grid to 4.1M for the octree grid and allows conjugate gradient to succeed. Because our adaptive method maintains a fine grid resolution at the liquid surface, it is also able to capture the bunny’s collisions along the wires.

7.2 Pure Octree Simulator Examples

Figure 18 demonstrates that our method also works seamlessly in a purely adaptive setting, as a viscosity plugin for the inviscid octree framework of Aanjaneya et al. [2017]. Figure 18 (left) shows a source pouring (Newtonian) ketchup. Figure 18 (right) shows four armadillos initially stacked together falling under gravity and collapsing into a pile. We use our basic (sloped gradient) octree viscosity method here, illustrating that even this simpler approach can yield qualitatively plausible motion in many cases. Both examples use an effective finest octree resolution of 512^3 with 4 levels, and a level-set based interface tracking resolution of 2048^3 . Table 2 gives a timing breakdown.

Table 2. Average timing breakdown (in seconds) for pure octree examples.

	Source	Armadillos
Level set advection	7.2	17
Reinitialization	6.2	18
Velocity advection	1.9	1.98
Viscous update	18.4	50.3
Projection	9.6	22.7
Velocity extrapolation	1.2	2.2
Grid adaptation	3	4.2
Total time step	48	117

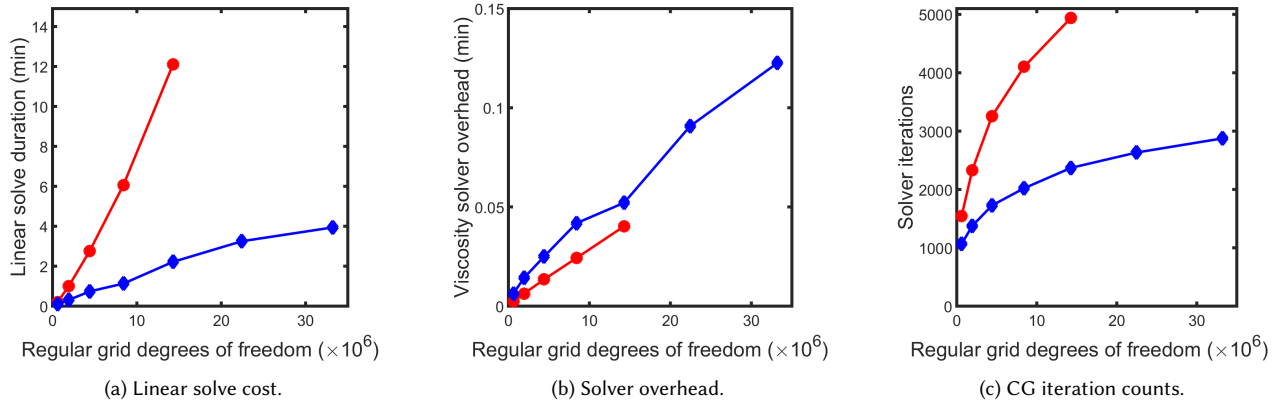


Fig. 16. Performance of the first frame for *Goopy Armadillo*. The X-axis indicates the number of active DOFs for the regular grid; for the corresponding octree, this is the *effective* resolution. (a) The linear system solve for the regular grid (red) requires significantly more compute time than for the equivalent octree (blue). The regular grid also failed at higher resolutions for which the octree succeeds. (b) The overhead for the octree solve (matrix setup, interpolation, etc.) is slightly larger than for the regular grid, but still inexpensive relative to the large improvement of the solve time. (c) The number of conjugate gradient iterations also increases more slowly for the octree than the corresponding regular grid.

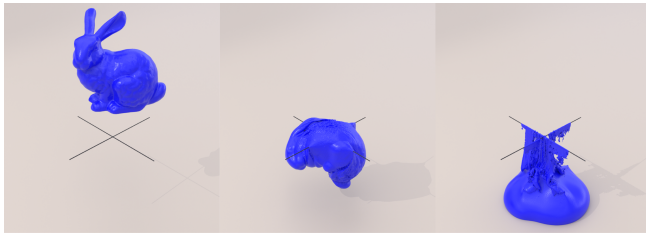


Fig. 17. **Bunny Drop**: A viscous bunny is dropped on two thin wires. The regular grid viscosity failed to solve at this grid resolution.

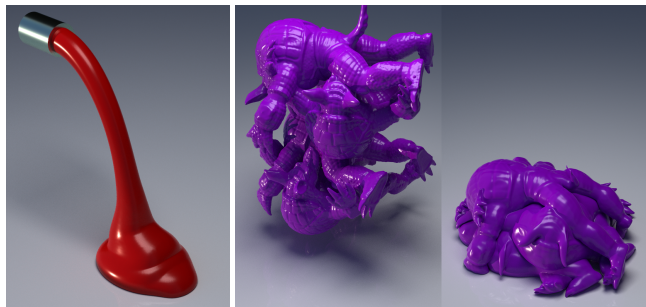


Fig. 18. **Pure Octree Simulator**: (Left) A source pours (Newtonian) ketchup. (Right) A stack of armadillos falls into a pile.

7.3 Refinement Studies

To quantitatively evaluate the spatial accuracy of our octree method, we constructed analytical tests representing a single time step of the viscosity PDE (1) on a closed box domain in 2D and 3D for a particular vector field and spatially varying viscosity function. Beginning from an irregular initial refinement pattern, we recursively subdivided every cell, evaluating the velocity error at each refinement

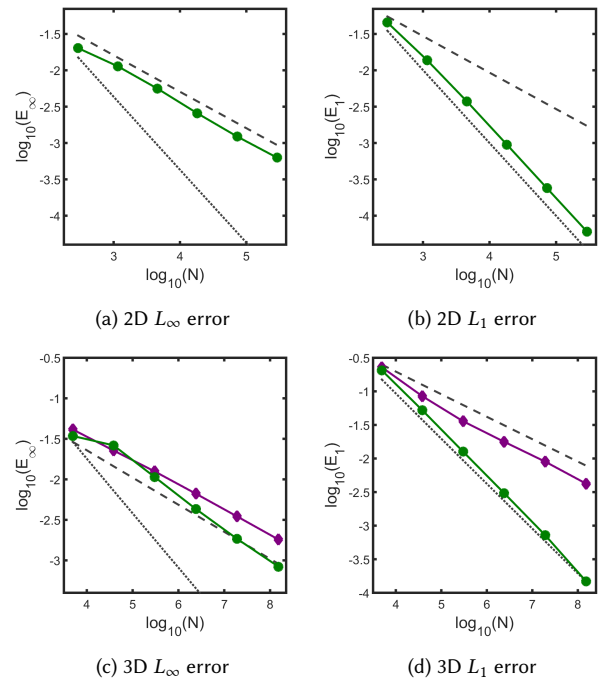


Fig. 19. Log-log plots of L_1 and L_∞ velocity error (labelled E_1 and E_∞ , respectively) vs. cell count N under refinement in two (top) and three (bottom) dimensions. Dashed and dotted guide lines indicate ideal first and second order slopes. In 3D, comparing the sloped (purple diamonds) and enhanced (green circles) gradient discretization error behavior confirms that the enhanced method experiences significantly improved convergence.

step. The results are plotted in Figure 19; the specific test cases and data for the plots are given in our supplemental material. Approx-

mately first order convergence is consistently observed in the L_∞ norm, with our enhanced approach exhibiting lower error than the sloped approach in 3D. However, under the L_1 norm the enhanced approach achieves full second order accuracy, whereas the sloped approach remains first order. Note that first order accurate velocity solutions are consistent with previous staggered octree schemes for pressure [Aanjaneya et al. 2017; Losasso et al. 2006]. While these methods offer second order accuracy in the *pressure*, the pressure gradient and velocity field remain first order.

7.4 Degrees of Freedom and Matrix Sparsity

Our method achieves its efficiency by reducing the number of active DOFs. For example, the octree viscosity DOF count for the *Gooney Armadillo* was 7.7 times smaller than for the regular grid. Although velocity samples at transition regions have more non-zeros per matrix row than in uniform regions, T-junctions are a relatively small subset of the domain and our stencils are fairly compact so the average number of non-zeros per DOF is still comparable to a regular grid. For example, in the first frame of Figure 15 each matrix row contained, on average, 14.67 non-zeros for the regular grid and 16.03 non-zeros for the octree. Hence, the sparsity of our octree linear systems is not appreciably worse than for regular grids.

If one were to derive a hexahedral finite element viscosity discretization on an octree, it could also provide a DOF reduction. However, a typical node-based linear FEM would lead to much denser matrices than ours. In the first frame of Figure 15 our octree consists of 3.39M active face DOFs, and assuming roughly 15 non-zeros per DOF, this yields about 50.85M non-zeros for our proposed method. The same octree has 1.06M active nodes, and each FEM node would contain three velocity DOFs, resulting in 3.18M node-based DOFs. Because regular grid FEM would require 81 non-zeros per DOF on uniform regions [Zhu et al. 2010] (and T-junctions would worsen this) the linear system would contain approximately 257.8M non-zeros, i.e., more than a factor of five more matrix non-zeros than our method for about the same number of unknowns.

8 CONCLUSIONS AND FUTURE WORK

We have demonstrated the viability of our novel adaptive variational finite difference framework by applying it to develop a symmetric positive definite octree discretization for viscous liquids. Our performance numbers confirm its efficiency relative to regular grids, and our visual and numerical experiments clarify the key influence of orthogonal gradient estimates in achieving high-quality results. We hope our work, like that of Setaluri et al. [2014], will help to further dispel the perception that the benefits of octree methods are outweighed by their overhead or complexity. To encourage adoption, we will be releasing the Houdini plugin version of our code.

We retained Houdini’s parallelized Jacobi-preconditioned conjugate gradient for our tests because, despite the large number of iterations it required, alternative off-the-shelf solvers that we tried (e.g., algebraic multigrid) did not exhibit competitive performance. We expect that the development of viscosity-specific numerical linear solvers and preconditioners, such as geometric multigrid or domain decomposition, will be vital to achieving further performance gains as has been the case for Poisson problems. Our guarantee of positive

definiteness will likely be useful in this regard, but since our system is more complex than the Poisson equation and not an M-matrix, it remains a non-trivial task. Adapting ideas from staggered multigrid elasticity schemes (e.g., [Zhu et al. 2010]) may be a viable avenue.

For simplicity, we adopted a uniformly refined surface and a (face-)graded interior. Lifting these restrictions could offer greater flexibility at the cost of increased stencil complexity. Relatedly, compared to a more classical finite difference/volume approach, our method has two potential error sources: first, our various stencils are often not centered differences, and second, near level transitions the discrete tensor divergence operator constructed by symmetry does not have an obvious finite difference/volume interpretation. More complex but *non-symmetric* stencils would likely enable second order convergence in L_∞ (e.g., Horesh and Haber [2011] describe a non-symmetric second order octree scheme for Maxwell’s equations).

If one wished to trade off speed for better coiling and surface details, we anticipate that upgrading to an indefinite Stokes (i.e., pressure-coupled) model will be straightforward, at least conceptually: add $\iiint_{\Omega} -p \nabla \cdot u$ to (2) and discretize in the same fashion; the positive definite form of Larionov et al. [2017] should likewise be possible with additional effort. Looking beyond viscosity, regular grid variational finite differences have also been applied to solid-fluid interaction [Batty et al. 2007], granular flow [Narain et al. 2010], and stream functions [Ando et al. 2015b] among other problems; Zhu et al. [2010] similarly used staggered finite differences for elasticity, which can likewise be expressed in a variational form. Our work naturally opens the door to *systematic, symmetry-preserving* extensions of all such staggered finite difference methods to octrees.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (RGPIN-04360-2014, CRDPJ-499952-2016) and the Rutgers University start-up grant. We would like to thank Cristin Barghiel from SideFX Software for their generous software donation as well as their dynamics team for their helpful discussions on maximizing parallelism. We would also like to thank the RenderMan team for their education licenses and the anonymous reviewers for their valuable feedback.

REFERENCES

- Mridul Aanjaneya, Ming Gao, Haixiang Liu, Christopher Batty, and Eftychios Sifakis. 2017. Power diagrams and sparse paged grids for high resolution adaptive liquids. *ACM Trans. Graph. (SIGGRAPH)* 36, 4 (2017), 140.
- Ann S. Almgren, John B. Bell, and Phillip Colella. 1998. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comp. Phys.* 142, 1 (1998), 1–46.
- Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2013. Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph. (SIGGRAPH)* 32, 4 (2013), 103.
- Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2015a. A dimension-reduced pressure solver for liquid simulations. *Computer Graphics Forum (Eurographics)* 34, 2 (2015), 473–480.
- Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2015b. A stream function solver for liquid simulations. *ACM Trans. Graph. (SIGGRAPH)* 34, 4 (2015), 53.
- Luiz Fernando de Souza Andrade, Marcos Sandim, Fabiano Petronetto, Paulo Pagliosa, and Afonso Paiva. 2015. Particle-based fluids for viscous jet buckling. *Computers and Graphics* 52 (2015), 106–115.
- Adam W. Bargteil, Chris Wojtan, Jessica K. Hodgins, and Greg Turk. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. (SIGGRAPH)* 26, 3 (2007), 16.

- Christopher Batty. 2017. A cell-centred finite volume method for the Poisson problem on non-graded quadrees with second order accurate gradients. *J. Comp. Phys.* 331 (2017), 49–72.
- Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. (SIGGRAPH)* 26, 3 (2007), 100.
- Christopher Batty and Robert Bridson. 2008. Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Symposium on Computer Animation*. 219–228.
- Christopher Batty and Ben Houston. 2011. A simple finite volume method for adaptive viscous liquids. In *Symposium on Computer Animation*. 111–118.
- Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete viscous sheets. *ACM Trans. Graph. (SIGGRAPH)* 31, 4 (2012), 113.
- Christopher Batty, Stefan Xenos, and Ben Houston. 2010. Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. *Computer Graphics Forum (Eurographics)* 29, 2 (2010), 695–704.
- Miklos Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Trans. Graph. (SIGGRAPH)* 29, 4 (2010), 116.
- Andrea Bonito, Marco Picasso, and Manuel Laso. 2006. Numerical simulation of 3D viscoelastic flows with free surfaces. *J. Comp. Phys.* 215, 2 (2006), 691–716.
- Robert Bridson. 2015. *Fluid simulation for computer graphics, 2nd edition*. AK Peters / CRC Press.
- Tyson Brochu, Christopher Batty, and Robert Bridson. 2010. Matching fluid simulation elements to surface geometry and topology. *ACM Trans. Graph. (SIGGRAPH)* 29, 4 (2010), 47.
- Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović. 2002. A multiresolution framework for dynamic deformations. In *Symposium on Computer Animation*. 41–47.
- Mark Carlson, Peter J. Mucha, R. Van Horn, and Greg Turk. 2002. Melting and flowing. In *Symposium on Computer Animation*. 167–174.
- Nuttapong Chentanez, Bryan E. Feldman, François Labelle, James F. O’Brien, and Jonathan Richard Shewchuk. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *Symposium on Computer Animation*. 219–228.
- Pascal Clausen, Martin Wicke, Jonathan Richard Shewchuk, and James F. O’Brien. 2013. Simulating liquids and solid-liquid interactions with Lagrangian meshes. *ACM Trans. Graph.* 32, 2 (2013), 17.
- Sharif Elcott, Yiyong Tong, Eva Kanso, Peter Schröder, and Mathieu Desbrun. 2007. Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph.* 26, 1 (2007), 4.
- Henrik Fält and Doug Roble. 2003. Fluids with extreme viscosity. In *SIGGRAPH Sketches*. 1.
- Florian Ferstl, Rudiger Westermann, and Christian Dick. 2014. Large-scale liquid simulation on adaptive hexahedral grids. *IEEE TVCG* 20, 10 (2014), 1405–1417.
- Nick Foster and Dimitris Metaxas. 1996. Realistic animation of liquids. *Graphical Models and Image Processing* 58, 5 (1996), 471–483.
- Ming Gao, Andre Pradhana Tampubolon, Chenfanfu Jiang, and Eftychios Sifakis. 2017. An adaptive generalized interpolation material point method for simulating elastoplastic materials. *ACM Trans. Graph. (SIGGRAPH Asia)* 36, 6 (2017), 223.
- TV. Gerya, D. A. May, and T. Duretz. 2013. An adaptive staggered grid finite difference method for modeling geodynamic Stokes flows with strongly variable viscosity. *Geochemistry, Geophysics, Geosystems* 14, 4 (2013), 1200–1225.
- Tolga G. Goktekin, Adam W. Bargteil, and James F. O’Brien. 2004. A method for animating viscoelastic fluids. *ACM Trans. Graph. (SIGGRAPH)* 23, 3 (2004), 463–468.
- Eitan Grinspun, Peter Krysl, and Peter Schröder. 2002. CHARMS: A simple framework for adaptive simulation. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (2002), 281–290.
- Arthur Guittet, Maxime Theillard, and Frédéric Gibou. 2015. A stable projection method for the incompressible Navier-Stokes equations on arbitrary geometries and adaptive Quad/Octrees. *J. Comp. Phys.* 292 (2015), 215–238.
- Jeong-Mo Hong and Chang-Hun Kim. 2005. Discontinuous fluids. *ACM Trans. Graph. (SIGGRAPH)* 24, 3 (2005), 915–920.
- Lior Hoshes and Eldad Haber. 2011. A second order discretization of Maxwell’s equations in the quasi-static regime on octree grids. *SIAM J. Sci. Comput.* 33, 5 (2011), 2805–2822.
- Bryan M. Klingner, Bryan E. Feldman, Nuttapong Chentanez, and James F. O’Brien. 2006. Fluid animation with dynamic meshes. *ACM Trans. Graph. (SIGGRAPH)* 25, 3 (2006), 820–825.
- Egor Larionov, Christopher Batty, and Robert Bridson. 2017. Variational Stokes: a unified pressure-viscosity solver for accurate viscous liquids. *ACM Trans. Graph. (SIGGRAPH)* 36, 4 (2017), 101.
- Frank Losasso, Ronald Fedkiw, and Stanley Osher. 2006. Spatially adaptive techniques for level set methods and incompressible flow. *Computers & Fluids* 35, 10 (2006), 995–1010.
- Frank Losasso, Frédéric Gibou, and Ron Fedkiw. 2004. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH)* 23, 3 (2004), 457–462.
- Chohong Min and Frédéric Gibou. 2006. A second order accurate projection method for the incompressible Navier-Stokes equations on non-graded adaptive grids. *J. Comp. Phys.* 219, 2 (2006), 912–929.
- Marek Misztal, Robert Bridson, Kenny Erleben, Andreas Baerentzen, and Francois Anton. 2010. Optimization-based fluid simulation on unstructured meshes. In *VRIPHYS*. 11–20.
- Marek Misztal, Kenny Erleben, Adam W. Bargteil, B. Bunch Christensen, Andreas Baerentzen, and Robert Bridson. 2014. Multiphase flow of immiscible fluids on unstructured moving meshes. *IEEE TVCG* 20, 1 (2014), 4–16.
- Patrick Mullen, Keenan Crane, Dmitry Pavlov, Yiyong Tong, and Mathieu Desbrun. 2009. Energy-preserving integrators for fluid animation. *ACM Trans. Graph. (SIGGRAPH)* 28, 3 (2009), 38.
- Rahul Narain, Abhinav Golas, and Ming C. Lin. 2010. Free-flowing granular materials with two-way solid coupling. *ACM Trans. Graph. (SIGGRAPH Asia)* 29, 6 (2010), 173.
- Michael B. Nielsen and Robert Bridson. 2016. Spatially adaptive FLIP fluid simulations in Bifrost. In *ACM SIGGRAPH talks*. 41.
- K. D. Nikitin, M. A. Olshanskii, K. M. Terekhov, and Y. V. Vassilevski. 2011. A numerical method for the simulation of free surface flows of viscoplastic fluid in 3D. *Journal of Computational Mathematics* (2011), 605–622.
- K. D. Nikitin and Yuri V. Vassilevski. 2008. Free surface flow modelling on dynamically refined hexahedral meshes. *Russian Journal of Numerical Analysis and Mathematical Modelling* 23, 5 (2008), 469–485.
- Maxim A. Olshanskii, Kirill M. Terekhov, and Yuri V. Vassilevski. 2013. An octree-based solver for the incompressible Navier-Stokes equations with enhanced stability and low dissipation. *Computers & Fluids* 84 (2013), 231–246.
- Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. 2015. An implicit viscosity formulation for SPH fluids. *ACM Trans. Graph. (SIGGRAPH)* 34, 4 (2015), 114.
- Daniel Ram, Theodore Gast, Chenfanfu Jiang, Craig Schroeder, Alexey Stomakhin, Joseph Teran, and Pirouz Kavehpour. 2015. A material point method for viscoelastic fluids, foams and sponges. In *Symposium on Computer Animation*. 157–163.
- Nick Rasmussen, Doug Enright, Duc Nguyen, Sebastian Marino, N. Sumner, Willi Geiger, Samir Hoon, and Ron Fedkiw. 2004. Directable photorealistic liquids. In *Symposium on Computer Animation*. 193–202.
- Rajsekhar Setaluri, Mridul Aanjaneya, Sean Bauer, and Eftychios Sifakis. 2014. SPGrid: A sparse paged grid structure applied to adaptive smoke simulation. *ACM Trans. Graph. (SIGGRAPH Asia)* 33, 6 (2014), 205.
- Lin Shi and Yizhou Yu. 2004. Visual smoke simulation with adaptive octree refinement. *Computer Graphics and Imaging* (2004), 13–19.
- Side Effects Software. 2018. Houdini.
- Jos Stam. 1999. Stable fluids. In *SIGGRAPH*. 121–128.
- Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Trans. Graph. (SIGGRAPH)* 33, 4 (2014), 138.
- Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C. Lin. 2015. Implicit formulation for SPH-based viscous fluids. *Computer Graphics Forum (Eurographics)* 34, 2 (2015), 493–502.
- Jeremy D. Wendt, William Baxter, Ipek Oguz, and Ming C. Lin. 2007. Finite volume flow simulations on arbitrary domains. *Graphical Models* 69, 1 (2007), 19–32.
- Chris Wojtan and Greg Turk. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph. (SIGGRAPH)* 27, 3 (2008), 47.
- Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2015. Continuum foam: A material point method for shear-dependent flows. *ACM Trans. Graph.* 34, 5 (2015), 160.
- Yonghao Yue, Breannan Smith, Peter Yichen Chen, Maytee Chantharayukhonthorn, Ken Kamrin, and Eitan Grinspun. 2018. Hybrid Grains: Adaptive Coupling of Discrete and Continuum Simulations of Granular Media. *ACM Trans. Graph. (SIGGRAPH Asia)* 37, 6 (2018), 283.
- Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015. Codimensional non-Newtonian fluids. *ACM Trans. Graph. (SIGGRAPH)* 34, 4 (2015), 115.
- Yongning Zhu, Eftychios Sifakis, Joseph Teran, and Achi Brandt. 2010. An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Trans. Graph.* 29, 2 (2010), 16.