

# PolyStokes: A Polynomial Model Reduction Method for Viscous Fluid Simulation

JONATHAN PANUELOS, University of Toronto, Canada

RYAN GOLDADE, University of Waterloo, Canada

EITAN GRINSPUN, University of Toronto, Canada

DAVID LEVIN, University of Toronto, Canada

CHRISTOPHER BATTY, University of Waterloo, Canada



Fig. 1. Liquid rope coiling instability solved using tiled quadratic regions of size  $7^3$ , with 3-cell padding between. Cutaway view shown on the left, featuring regular grid cells in red and our reduced model interior regions in green.

Standard liquid simulators apply operator splitting to independently solve for pressure and viscous stresses, a decoupling that induces incorrect free surface boundary conditions. Such methods are unable to simulate fluid phenomena reliant on the balance of pressure and viscous stresses, such as the liquid rope coil instability exhibited by honey. By contrast, unsteady Stokes solvers retain coupling between pressure and viscosity, thus resolving these phenomena, albeit using a much larger and thus more computationally expensive linear system compared to the decoupled approach. To accelerate solving the unsteady Stokes problem, we propose a reduced fluid model wherein interior regions are represented with incompressible polynomial vector fields. Sets of standard grid cells are consolidated into super-cells,

Authors' addresses: Jonathan Panuelos, University of Toronto, Canada, jonathan.panuelos@mail.utoronto.ca; Ryan Goldade, University of Waterloo, Canada, rgoldade@uwaterloo.ca; Eitan Grinspun, University of Toronto, Canada, eitan@grinspun.com; David Levin, University of Toronto, Canada, diwlevin@cs.toronto.edu; Christopher Batty, University of Waterloo, 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](mailto:permissions@acm.org).

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0730-0301/2023/8-ART \$15.00

<https://doi.org/10.1145/3592146>

each of which are modelled using a quadratic field of 26 degrees of freedom. We demonstrate that the reduced field must necessarily be at least quadratic, with the affine model being unable to correctly capture viscous forces. We reproduce the liquid rope coiling instability, as well as other simulated examples, to show that our reduced model is able to reproduce the same fluid phenomena at a smaller computational cost. Furthermore, we performed a crowdsourced user survey to verify that our method produces imperceptible differences compared to the full unsteady Stokes method.

CCS Concepts: • **Computing methodologies** → **Computer Graphics; Modeling and simulation.**

Additional Key Words and Phrases: fluid simulation, Stokes, spatial adaptivity, dimension reduction

## ACM Reference Format:

Jonathan Panuelos, Ryan Goldade, Eitan Grinspun, David Levin, and Christopher Batty. 2023. PolyStokes: A Polynomial Model Reduction Method for Viscous Fluid Simulation. *ACM Trans. Graph.* 42, 4 (August 2023), 13 pages. <https://doi.org/10.1145/3592146>

## 1 INTRODUCTION

Viscous, incompressible liquids exhibit a variety of intriguing phenomena: a misplaced step on boggy ground leaves one's boot stuck in thick wet mud; cake batter poured from a mixing bowl folds rhythmically back and forth [Suleiman and Munson 1981]; drop

a stream of honey just so and one finds it coiling in a rope-like manner [Ribe et al. 2012]. Applications of viscous fluid simulation are also not limited to these minor delights—industrial applications use grease and lubricants, geologists study convection currents of the Earth’s mantle, and even daily life finds viscous fluids in foods and cosmetic products. Naturally then, viscous liquids have been a subject of interest in computer graphics going back more than twenty years [Carlson et al. 2002].

The physical behaviours of fluids have been distilled into a pair of conservation laws called the Navier-Stokes equations. In computer graphics, this system is typically simplified by way of operator splitting, which involves solving smaller subproblems in turn [Bridson 2015]. Advective and body forces are often solved separately, leaving just the coupled pressure and viscous shear terms. Considered together, these two terms are governed by the unsteady Stokes equations, which are the primary focus of our work. While frequently also solved separately [Batty and Bridson 2008], Larionov et al. [2017] showed that the coupling of these two terms is essential in simulating the correct coiling behaviour of liquid ropes; Takahashi and Batty [2020] likewise demonstrated its importance for coupling with rigid bodies. However, this increased fidelity comes at a cost: the resulting unified system of equations is much larger than for decoupled methods, leading even recent work on octree-based [Goldade et al. 2019] and multigrid [Shao et al. 2022] viscosity solvers to opt for decoupling.

We aim to reduce the cost of the unified Stokes solve by considering spatial adaptivity and model reduction. The former focuses computational cost at regions of interest while reducing resolution elsewhere [Losasso et al. 2004]; the latter captures most of the flow behaviour with as few degrees of freedom as possible [Treuille et al. 2006]. The method of Goldade et al. [2020] can be viewed as combining these: they propose a pressure projection method that couples one or more coarser interior domains, whose velocities are described by incompressible affine vector fields, to a surface region composed of uniform grid cells, treated with a standard discretization. This method achieves spatial adaptivity and dimension reduction without complicated stencils or data structures, thus making it relatively easy to implement.

Extending these ideas, we propose the first acceleration scheme for the unified Stokes solve via the use of incompressible *polynomial* vector fields. We show that an affine vector field is insufficient for the Stokes problem, and that a quadratic vector field is the lowest order model able to achieve proper, convergent treatment of viscous forces on the fluid interior. We confirm that our method retains the attractive features of the Stokes approaches, and in particular still reproduces the expected liquid coiling instability (Figure 1).

Finally, since any adaptive or reduced model entails some (preferably mild) deviation in behavior compared to the fully refined original model, we seek to evaluate this effect. While various numerical or geometric measures could be proposed, we observe that, in the context of animation, such measures are often a proxy for the “visual error” experienced by human viewers. Therefore, drawing inspiration from pioneering work by Um et al. [2017], we propose to interrogate this question *perceptually*: we perform a two-alternative forced-choice study with 500 participants to confirm that our method is

visually indistinguishable from the more costly fully uniform solution.

## 2 RELATED WORK

Standard Eulerian and hybrid methods for solving incompressible flow in computer graphics use staggered pressure and velocity samples, originally developed for the Marker-and-Cell (MAC) method of Harlow and Welch [1965]. This scheme was introduced into computer graphics by Foster and Metaxas [1996] who solved the pressure and velocity updates explicitly, assuming voxelized solid boundaries. In response to the timestep limitations imposed by explicit solvers, Stam [1999] developed an unconditionally stable method using semi-Lagrangian advection and decoupled implicit solvers for pressure and viscosity.

To leverage desirable properties of both Eulerian and Lagrangian methods, Particle-in-Cell (PIC) and related hybrid methods were developed. The first formulation of (PIC) was created by Harlow [1962], which suffered from excessive dissipation. Brackbill and Ruppel [1986] subsequently developed the Fluid-Implicit-Particle (FLIP) method as a remedy. Zhu and Bridson [2005] applied it to incompressible flow, and suggested a weighted average of the two methods to reduce noise. Jiang et al. [2015] constructed a locally affine variation of the (PIC) interpolation method in order to preserve angular momentum previously lost with each grid-to-particle interpolation, and was later extended to be locally polynomial by Fu et al. [2017]. We use the affine PIC (APIC) framework to perform the advection step of our implementation, though we note that the reduced Stokes solver we develop is agnostic to the choice of advection procedure. Bridson [2015] provides additional background on standard Eulerian/hybrid fluid simulators for computer animation.

*Large viscosity* presents additional challenges and opportunities. Carlson et al. [2002] focused on solving highly viscous liquids, adopting an implicit Laplacian-based smoothing of velocity in place of Stam’s Fast Fourier Transform (FFT)-based viscosity. Their treatment of free surface boundaries was found to produce incorrect translational motion, later corrected by Fält and Roble [2003]. Batty and Bridson [2008] adopted a variational approach to achieve improved zero-traction free-surface boundary conditions and thereby recover rotational motion needed for buckling fluids. Larionov et al. [2017] extended the variational approach to solve unsteady Stokes flow, coupling the pressure and viscosity equations together. This method is able to recover cylindrical coiling behaviour in place of the random buckling of prior decoupled methods, and forms the base uniform grid solver that we build our reduced method on top of. Outside of Eulerian volumetric approaches, Bergou et al. [2010] were able to successfully recreate meandering and coiling patterns of viscous threads on a conveyor using a discrete rod-based model, and Batty et al. [2012] adapted a thin shell model to simulate viscous sheets. Likewise, viscous liquids have seen extensive study using Lagrangian smoothed particle hydrodynamics approaches (e.g., [Takahashi et al. 2015; Weiler et al. 2018].) While more exotic non-Newtonian liquids have also been widely explored, we restrict our focus to purely Newtonian viscous liquids.

*Solid-Fluid Coupling* methods serve as a basis for our work. The two-way rigid body-fluid coupling of Batty et al. [2007] introduced

an operator that converted boundary fluid pressures into generalized forces that act on the solid. The solid's intended behaviour determines the constraints that define the degrees of freedom of its generalized velocity. Takahashi and collaborators extended this approach to incorporate viscous forces and simultaneous rigid-rigid contact handling [Takahashi and Batty 2020; Takahashi and Lin 2019]. Goldade et al. [2020] replaced the rigid body constraint used by Batty et al. [2007] for rigid-fluid coupling with an incompressible affine field intended to represent a coarse fluid region, drawing on the same analogy as the rigid (RPIC) and affine (APIC) variants of PIC developed by Jiang et al. [2015]. Extending this analogy, our polynomial extension of Goldade's model is akin to the polynomial extension of APIC to PolyPIC [Fu et al. 2017].

### Acceleration methods

*Spatial adaptivity* can be a useful strategy in trading accuracy for computational cost. Shi and Yu [2004] and Losasso et al. [2004] introduced the use of an octree data structure as a method for focusing computation on areas of interest by reducing grid resolution elsewhere, translating the uniform grid-based stable fluids method into stencils suited to irregular octree grids. Tetrahedral meshes have also been explored, first with precomputed static meshes by Feldman et al. [2005], then with dynamic meshes by Klingner et al. [2006], and later with embedded free surface and solid boundary conditions by Batty et al. [2010]. Batty and Houston [2011] and Goldade et al. [2019] considered spatially adaptive Eulerian solvers for the (decoupled) viscosity equations, on tetrahedral meshes and octree grids, respectively. Chentanez et al. [2007] used tetrahedral meshes with an algebraic multigrid solver for pressure. Similarly, methods based on Voronoi or power diagrams rather than tetrahedral meshes have also been developed [Brochu et al. 2010; de Goes et al. 2015; Sin et al. 2009], and later adapted to improve octree t-junctions [Aanjaneya et al. 2017]. Goldade et al. [2020] presented a reduced model pressure solver as a method for spatial adaptivity with uniform cells near the boundary and coarsened reduced model tiles on the interior, which we directly adopt for our method. Approaching the same computational efficiency goals, Edwards and Bridson [2014] used variable polynomial degrees ( $p$ -adaptivity) on a discontinuous Galerkin method to focus computation near the fluid surface. Compared to our use of polynomials to allow coarsening on interior cells, they apply higher-order polynomials on the fluid surface to resolve finer detail out of a globally coarse grid. Another similarity to our method is that both use polynomial elements or regions with irregular, non-convex shapes, as also achieved by Tao et al. [2022] via the virtual element method.

*Reduced fluid models*, such as ours, use reduced sets of variables to decrease computational cost. Treuille et al. [2006] used principal component analysis to find the reduced basis of a target dimension that minimizes reconstruction error. In place of a global basis, Wicke et al. [2009] constructed large cube-shaped simulation primitives called 'tiles' with local bases; coupling along faces is handled by shared boundary bases. Our reduced regions use a different interior model, but away from boundaries their shapes simplify to become rectangular tiles, coupled together by surrounding fine-grid cells.

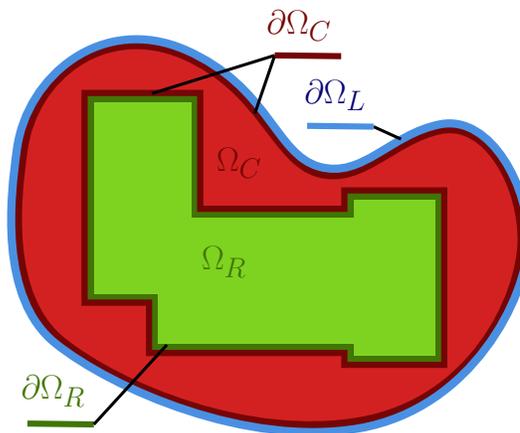


Fig. 2. **Schematic of various domains used in our reduced solver.** The Cartesian domain,  $\Omega_C$ , and reduced domain,  $\Omega_R$ , form a decomposition of the full liquid domain,  $\Omega_L = \Omega_C \cup \Omega_R$ , such that they share only a boundary,  $\Omega_C \cap \Omega_R = \partial\Omega_R \subseteq \partial\Omega_C$ .

De Witt et al. [2012] used Laplacian eigenfunctions as a divergence-free basis. This scheme was later extended by Cui et al. [2018], who reduced memory cost with discrete sine and cosine transforms as well as introducing eigenfunctions to support Neumann boundary conditions. Ando et al. [2015] solved for pressure on a reduced-dimension grid and constructed an upsampler to compute corrected velocities that respect the free surface boundary condition. Da et al. [2016] presented a boundary-only method for high surface tension fluids, avoiding computation of internal volumetric degrees of freedom. The reduced solver of Goldade et al. [2020] uses an affine basis to reduce degrees of freedom allocated to interior regions of the fluid. We extend this idea to higher-order polynomial bases for simulating the decoupled viscosity step and the coupled Stokes problem.

## 3 METHODS

### 3.1 A Reduced Model for Viscosity

We begin constructing our reduced model solver by first looking at only the decoupled viscosity step treated with implicit Euler,

$$\frac{\mathbf{u} - \mathbf{u}^*}{\Delta t} = \frac{1}{\rho} \nabla \cdot (\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top)), \quad (1)$$

where  $\mathbf{u}$  is the velocity field being solved for,  $\mathbf{u}^*$  is an input divergence-free velocity field,  $\rho$  is density,  $\mu$  is the viscosity coefficient, and  $\Delta t$  is the timestep size. The variational method of Batty and Bridson [2008] equivalently expresses the solution as the minimizer of

$$J[\mathbf{u}] = \iiint_{\Omega_L} \left( \frac{\rho}{2} \|\mathbf{u} - \mathbf{u}^*\|^2 + \Delta t \mu \left\| \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^\top}{2} \right\|_F^2 \right) dV. \quad (2)$$

We separate this integration into the uniform Cartesian regions,  $\Omega_C$ , and the reduced fluid regions,  $\Omega_R$ , such that  $\Omega_C \cap \Omega_R = \partial\Omega_R \subseteq \partial\Omega_C$  and  $\Omega_C \cup \Omega_R = \Omega_L$ . These domains and their boundaries are shown in Figure 2. Note that the boundary region of the Cartesian region,  $\partial\Omega_C$  includes both its shared boundary with the reduced region,  $\partial\Omega_R$ , and the true liquid boundary (surface)  $\partial\Omega_L$ . This relation holds

assuming we immerse the reduced fluid region(s) entirely within the liquid. While Figure 2 shows a single reduced region, in practice we follow Goldade et al. [2020] in dividing the interior fluid domain into many distinct reduced tiles, separated from each other by thin layers of uniform cells. We selected a default tile size of  $32^3$ , with a 3-cell padding, which we found heuristically worked well for our examples.

We split the velocity field into  $\mathbf{u}_C = \mathbf{u} I_{\Omega_C}$  and  $\mathbf{u}_R = \mathbf{u} I_{\Omega_R}$ , where  $I_A$  is the indicator function of a domain  $A$ . This yields the objective

$$\begin{aligned} J[\mathbf{u}_C, \mathbf{u}_R] = & \iiint_{\Omega_L} \frac{\rho}{2} \|\mathbf{u}_C - \mathbf{u}_C^*\|^2 + \frac{\rho}{2} \|\mathbf{u}_R - \mathbf{u}_R^*\|^2 \\ & + \rho(\mathbf{u}_C - \mathbf{u}_C^*) \cdot (\mathbf{u}_R - \mathbf{u}_R^*) \\ & + \Delta t \mu \left\| \frac{\nabla \mathbf{u}_C + (\nabla \mathbf{u}_C)^\top}{2} \right\|_F^2 + \Delta t \mu \left\| \frac{\nabla \mathbf{u}_R + (\nabla \mathbf{u}_R)^\top}{2} \right\|_F^2 \\ & + 2\Delta t \mu \left\langle \frac{\nabla \mathbf{u}_C + (\nabla \mathbf{u}_C)^\top}{2}, \frac{\nabla \mathbf{u}_R + (\nabla \mathbf{u}_R)^\top}{2} \right\rangle_F dV, \quad (3) \end{aligned}$$

with  $\langle \cdot, \cdot \rangle_F$  as the Frobenius inner product. Minimizing this objective function in  $\mathbf{u}_{C/R}$  solves the system

$$\frac{\mathbf{u}_C - \mathbf{u}_C^*}{\Delta t} = \frac{1}{\rho} \mu \nabla \cdot (\nabla \mathbf{u}_C + (\nabla \mathbf{u}_C)^\top) \quad \text{in } \Omega_C, \quad (4)$$

$$\frac{\mathbf{u}_R - \mathbf{u}_R^*}{\Delta t} = \frac{1}{\rho} \mu \nabla \cdot (\nabla \mathbf{u}_R + (\nabla \mathbf{u}_R)^\top) \quad \text{in } \Omega_R, \quad (5)$$

$$\mathbf{u}_C = \mathbf{u}_R \quad \text{on } \partial\Omega_R. \quad (6)$$

Notice that this is the viscosity system, Equation 1, solved separately for regions  $\Omega_C$  and  $\Omega_R$ , with two-way coupling being handled by the matching condition on their mutual boundary. We make the assumption of constant viscosity (note the  $\mu$  outside of the divergence, as compared to the original viscosity update in Equation 1), as a limitation of the reduced model. Because the reduced model does not use individual stress degrees of freedom on the interior, there is no definition of a spatially-varying viscosity field, and we thus necessarily assume a constant viscosity value in our work. (Generalizing to polynomially varying coefficients would be an interesting extension.)

Choosing a particular model for the velocity field in the reduced region,  $\mathbf{u}_R$ , determines its generalized velocity degrees of freedom,  $\mathbf{v}_R$ , via  $\mathbf{u}_R = \mathbf{J}^\top \mathbf{v}_R$ . Adopting the usual staggered grid discretization on the Cartesian region [Batty and Bridson 2008], we can then construct a discretization of the corresponding first-order optimality conditions of the combined variational problem in terms of uniform Cartesian velocities,  $\mathbf{u}_C$ , and generalized velocities,  $\mathbf{v}_R$ , for the reduced model:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^\top & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_C \\ \mathbf{v}_R \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}_C \mathbf{W}_F^u \mathbf{W}_L^u \mathbf{u}^* \\ \frac{1}{\Delta t} \mathbf{M}_R \mathbf{v}_R^* \end{bmatrix}, \quad (7)$$

$$\mathbf{A}_{11} = \frac{1}{\Delta t} \mathbf{M}_C \mathbf{W}_F^u \mathbf{W}_L^u + 2\mathbf{W}_F^u \mathbf{D}^\top (\mathbf{W}_F^\tau)^{-1} \mu \mathbf{W}_L^\tau \mathbf{D} \mathbf{W}_F^u, \quad (8)$$

$$\mathbf{A}_{12} = 2\mathbf{W}_F^u \mathbf{D}^\top \mu \mathbf{D} \mathbf{J}^\top, \quad (9)$$

$$\mathbf{A}_{22} = \frac{1}{\Delta t} \mathbf{M}_R + 2\mathbf{J} \mathbf{D}^\top \mu \mathbf{D} \mathbf{J}^\top. \quad (10)$$

Here,  $\mathbf{D}$  is the discrete symmetric gradient operator, and  $\mathbf{M}_C, \mathbf{M}_R$  are mass matrices. As defined in Larionov et al. [2017], weights  $W_L^a$  represent the volume fraction of the unit cell around a point sample  $a$  that is within the liquid domain  $\Omega_L$ . Analogously,  $W_F^a$  are the fractional cell weights for the fluid domain  $\Omega_F$ , which represents the entire simulation domain outside of any solids.

Thus  $\mathbf{A}_{11}$  and  $\mathbf{A}_{22}$  express the independent updates of the fluid in uniform and reduced regions, respectively, and  $\mathbf{A}_{12}$  expresses their coupling. We have assumed that all cell weights associated with  $\mathbf{v}_R$  are unity, which is true if those samples are immersed within the active fluid domain,  $\Omega_R \subset \Omega_{L \cap F}$ , such that its boundary is sufficiently far from both boundary conditions,  $\text{dist}(\partial\Omega_R, \partial\Omega_L \cup \partial\Omega_F) > \Delta x$  where  $\text{dist}(A, B) = \inf\{\|x - y\| \mid x \in A, y \in B\}$  is a distance metric. We now simply need to determine an appropriate mapping  $\mathbf{J}$  from the uniform degrees of freedom to generalized velocities.

### 3.2 Affine Velocity Fields

As in the work of Goldade et al. [2020], we initially define an affine description of a reduced fluid velocity field as

$$\mathbf{u}_R(\mathbf{x}) = \mathbf{u}_{const} + \mathcal{G}(\mathbf{x} - \mathbf{x}_{COM}) \quad (11)$$

where  $\mathcal{G} = \nabla \mathbf{u}_R$  is the gradient 2-tensor and  $\mathbf{x}_{COM}$  is the reduced region's center of mass. Since the 3D velocity gradient matrix has the structure

$$\mathcal{G}_{3D} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix}, \quad (12)$$

it follows that its trace gives the divergence,  $\text{Tr}(\mathcal{G}_{3D}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = \nabla \cdot \mathbf{u}$ , and similarly for the 2D case. This relation implies that enforcing the usual incompressibility constraint can be done by simply enforcing zero trace,  $\nabla \cdot \mathbf{u}_R = \text{Tr}(\mathcal{G}) = 0$ . The required degrees of freedom for representing  $\mathcal{G}$  are thus reduced by 1, as shown in the following forms:

$$\mathcal{G}_{2D} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & -a_{11} \end{bmatrix}, \quad \mathcal{G}_{3D} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & -(a_{11} + a_{22}) \end{bmatrix}. \quad (13)$$

The velocity within some interior region  $\Omega_R$  can thus be represented by a generalized velocity vector  $\mathbf{v}_R$  of 5 elements in 2D (2 constant and 3 linear) and 11 elements in 3D (3 constant and 8 linear). We define a matrix  $\mathbf{C}$  that gives the Euclidean velocity at any point  $\mathbf{x}$ . The 3D case follows straightforwardly from the following 2D case,

$$\begin{aligned} \mathbf{u}_R &= \mathbf{C}(\mathbf{x}) \mathbf{v}_R \quad (14) \\ &= \begin{bmatrix} 1 & 0 & \tilde{x} & \tilde{y} & 0 \\ 0 & 1 & -\tilde{y} & 0 & \tilde{x} \end{bmatrix} \mathbf{v}_R \quad (15) \end{aligned}$$

with  $\tilde{x} = x - x_{COM}$  and similarly for  $\tilde{y}$ . We will also use this equation when needed to find a least-squares fit of our generalized velocities to a given set of uniform velocities.

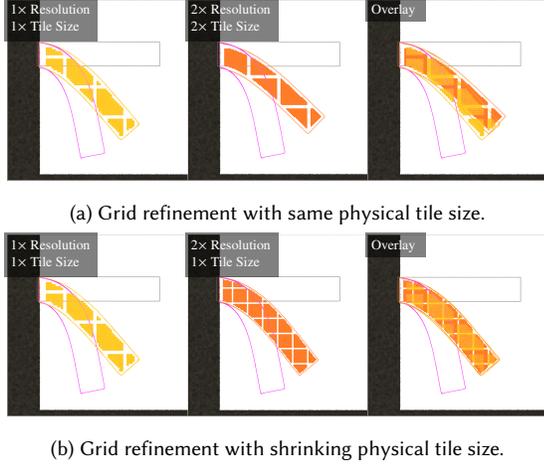


Fig. 3. **The affine model (yellow, orange) fails to match the reference simulation (pink outline) on a viscous beam test.** (a) A comparison between a base resolution affine simulation (yellow, left) and a simulation with both double spatial resolution and tile dimensions (orange, middle), such that the reduced tiles retain the same physical size under refinement. The right column overlays the two. (b) A similar comparison between a base resolution result (yellow, left) and a double resolution result *without* modifying the tile dimensions (orange, middle). All images show 2D cutaway views of 3D simulations with thin lines representing fluid surfaces for each method, reduced model tiles shown filled in with their respective colours, and solid boundaries shown in black. Uniform grid cells are not shown for clarity, but take up all gaps between interior regions and the fluid surface. Initial conditions are shown in grey, consisting of a homogeneous fluid with  $\rho = 1$  and  $\mu = 100$ . Results are for  $t = 1.6$ , using a *decoupled* pressure-viscosity fluid solver. All affine results sharply disagree with the uniform reference, even under spatial refinement.

Using this transformation, we can define a generalized mass matrix using the fluid's kinetic energy as follows:

$$\iiint_{\Omega_R} \frac{\rho_R}{2} \|\mathbf{u}_R\|^2 dV = \iiint_{\Omega_R} \frac{\rho_R}{2} \|\mathbf{C}\mathbf{v}_R\|^2 dV \quad (16)$$

$$= \frac{1}{2} \mathbf{v}_R^T \left( \iiint_{\Omega_R} \rho_R \mathbf{C}^T \mathbf{C} dV \right) \mathbf{v}_R \quad (17)$$

$$= \frac{1}{2} \mathbf{v}_R^T \mathbf{M}_R \mathbf{v}_R \quad (18)$$

The matrix  $\mathbf{M}_R = \iiint_{\Omega_R} \rho_R \mathbf{C}^T \mathbf{C} dV$  is exactly the required generalized mass matrix.

### 3.3 Problems with the Affine Description

By the above definition of an affine field, the velocity gradient  $\mathcal{G} = \nabla \mathbf{u}_R$ , which corresponds to  $\mathcal{G} = \mathbf{D}\mathbf{J}^T \mathbf{v}_R$  in the discrete setting. Since  $\mathcal{G}$  is constant in an affine field, it thus follows that the  $\mathbf{J}\mathbf{D}^T \mathbf{D}\mathbf{J}^T$  term in Equation 10 becomes *zero*, since it represents a second derivative:  $\mathbf{A}_{22} = \frac{1}{\Delta t} \mathbf{M}_R + 2\mathbf{J}\mathbf{D}^T \mu \mathbf{D}\mathbf{J}^T = \frac{1}{\Delta t} \mathbf{M}_R$ . This simplification implies that we no longer solve for viscous forces at all within the reduced region. The second equation is now strictly a velocity matching constraint on the boundary between the reduced model and the rest of the fluid, with no interior dynamics being performed.

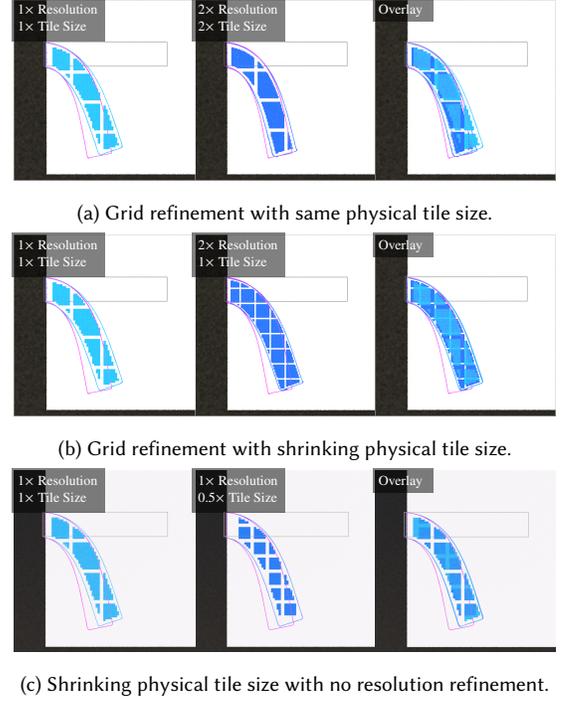


Fig. 4. **The quadratic model (sky blue, dark blue) more closely matches the reference simulation (pink outline) on a viscous beam test.** (a) A comparison between a base resolution affine simulation (sky blue, left) and a simulation with both double spatial resolution and tile dimensions (dark blue, middle), such that the reduced tiles retain the same physical size under refinement. The right column overlays the two. (b) A similar comparison between a base resolution result (sky blue, left) and a double resolution result *without* modifying the tile dimensions (dark blue, middle). (c) A comparison between a base simulation result (sky blue, left), and an identical resolution result with half the tile size (thus halving the physical size). The visualizations and problem parameters mimic Figure 3, and the results are again computed with a *decoupled* pressure-viscosity fluid solver. The quadratic model's results show much closer agreement with the reference result compared to the affine model in Figure 3.

The issue is thus not in the representative power of an affine field with respect to velocity, but rather of the viscous stresses. Because an affine field lacks a second derivative, it lacks any information for being able to evolve itself, and consequently it induces no stresses back onto the surrounding uniform grid. An alternative interpretation is to view the action of viscosity as a Laplacian smoothing operation, converting high frequency velocity modes to low frequency. Since the affine field lacks any high frequency modes, the viscous update results in no net change.

We demonstrate a practical result of this issue with a simple collapsing viscous beam example, shown in Figure 3. Here, the affine method is considerably stiffer than the reference solution using a high-resolution uniform grid, even under spatial refinement. The two rows of Figure 3 also show that the large errors also persist whether the physical tile size is set to shrink with the grid scale or remains unchanged.

### 3.4 Polynomial Velocity Fields

The lack of information with which to evolve viscosity in the affine reduced regions motivates our construction of a higher-order *polynomial* representation of the velocity. In place of the affine description, Equation 11, we construct the following quadratic model:

$$\mathbf{u}_R(\mathbf{x}) = \mathbf{u}_{const} + \mathcal{G}(\mathbf{x} - \mathbf{x}_{COM}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_{COM})^\top \mathcal{H}(\mathbf{x} - \mathbf{x}_{COM}) \quad (19)$$

where  $\mathcal{H}$  is the velocity Hessian 3-tensor with  $\mathcal{H}_{i,j,k} = \frac{\partial^2 u_i}{\partial x_j \partial x_k}$ . Each page  $i$  of  $\mathcal{H}$  is symmetric, thus requiring only 18 degrees of freedom rather than the full 27. We also enforce incompressibility by applying the usual  $\nabla \cdot \mathbf{u}_R = 0$  constraint, producing a total of 26 degrees of freedom. That is, 3 for the constant term, 8 for the linear term, and 15 for the quadratic term.

Using this new transformation matrix,  $\mathbf{C}$ , we can use the same definitions of the generalized mass matrix,  $\mathbf{M}_R$ , and transfer matrix,  $\mathbf{J}$ , as in the affine case. Applying this to the viscosity system, we thus retain the  $\mathbf{J}\mathbf{D}^\top\mathbf{D}\mathbf{J}^\top$  term that canceled away for affine fields, and consequently are able to resolve the missing dynamics within the interior regions. This extension of the affine model into a quadratic model is similar in spirit to the generalization of APIC into PolyPIC, which seeks to retain higher-order polynomial modes during particle-grid transfer operations [Fu et al. 2017; Jiang et al. 2015].

Applying this quadratic field to the same viscous beam problem, we get the results shown in Figure 4. Once again, the top subfigure shows the result for increasing the resolution and interior region size such that they physically represent the same space, while the middle subfigure shows the result of increasing just the spatial resolution without increasing the interior region size. The bottom subfigure keeps the same resolution but halves the tile size, effectively making a similar comparison as Figure 4(a) in terms of spatial scale of the tiles, but with a constant overall resolution. In all cases, the quadratic model performs significantly better than the affine model, achieving much closer results to the reference. Perhaps counterintuitively, increasing grid resolution while using the same tile count in Figure 4(a) shows slightly improved results, compared to increasing the *number* of tiles in Figure 4(b). We attribute this effect to the greater expressiveness of the padding layer of uniform cells between tiles, which in (a) must thicken to preserve the tile physical size, but shrinks along with grid cell size in (b). We conclude that convergence error is strongly influenced by the padding size between interior regions and that keeping a padding of only two cells limits accuracy.

### 3.5 Reduced Model for Stokes

We can now combine our new reduced viscosity system with the reduced pressure system of Goldade et al. [2020] to construct our full reduced Stokes system. We begin with the variational Stokes objective function of Larionov et al. [2017]:

$$J[\mathbf{u}, p, \tau] = \iiint_{\Omega_L} \frac{\rho}{2\Delta t} \|\mathbf{u} - \mathbf{u}^*\|^2 - p\nabla \cdot \mathbf{u} + \tau : \varepsilon(\mathbf{u}) - \frac{1}{4\mu} \|\tau\|_F^2 dV \quad (20)$$

where  $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^\top)$  is the deformation rate tensor. We then apply the same separation of  $\Omega_C$  and  $\Omega_R$  domains as used in Section 3.1. That is, we define a uniform Cartesian region,  $\Omega_C$ ,

and an internal reduced fluid region,  $\Omega_R$ , that combined span the entire liquid,  $\Omega_C \cup \Omega_R = \Omega_L$ , with a shared boundary,  $\Omega_C \cap \Omega_R = \partial\Omega_R \subseteq \partial\Omega_C$ . This shared boundary must be sufficiently far from both boundary conditions, such that  $\text{dist}(\partial\Omega_R, \partial\Omega_R \cup \partial\Omega_F) > \Delta x$ . Due to this immersion, it follows that  $\partial\Omega_L = \partial\Omega_C \setminus \partial\Omega_R$

Applying these two domains,

$$\begin{aligned} J[\mathbf{u}_C, \mathbf{u}_R, p, \tau_L] = & \iiint_{\Omega_C} \frac{\rho}{2\Delta t} \|\mathbf{u}_C - \mathbf{u}_C^*\|^2 + \rho(\mathbf{u}_C - \mathbf{u}_C^*) \cdot (\mathbf{u}_R - \mathbf{u}_R^*) \\ & - p\nabla \cdot \mathbf{u}_C + \tau : \varepsilon(\mathbf{u}_C) - \frac{1}{4\mu} \|\tau\|_F^2 dV, \\ & + \iiint_{\Omega_R} \frac{\rho}{2\Delta t} \|\mathbf{u}_R - \mathbf{u}_R^*\|^2 + \rho(\mathbf{u}_C - \mathbf{u}_C^*) \cdot (\mathbf{u}_R - \mathbf{u}_R^*) \\ & - p\nabla \cdot \mathbf{u}_R + \tau : \varepsilon(\mathbf{u}_R) + \mu\|\varepsilon(\mathbf{u}_R)\|^2 dV. \end{aligned} \quad (21)$$

Although  $\tau$  is defined throughout the liquid, we define actual degrees of freedom only on  $\Omega_C$ . For the complementary set, we use the relation  $\tau|_{\Omega_R \setminus \partial\Omega_R} = 2\mu\varepsilon(\mathbf{u}_R)$ . Likewise, we define  $p$  solely on  $\Omega_C$ , considering that incompressibility within the reduced fluid region is enforced by the definition of the reduced model.

Minimizing the above objective function solves the system

$$\frac{\rho}{\Delta t}(\mathbf{u}_C - \mathbf{u}_C^*) + \nabla p + \nabla \cdot \tau = 0 \quad \text{in } \Omega_C, \quad (22)$$

$$\frac{\rho}{\Delta t}(\mathbf{u}_R - \mathbf{u}_R^*) + \nabla p + \nabla \cdot \tau + \mu\nabla \cdot \varepsilon(\mathbf{u}_R) = 0 \quad \text{in } \Omega_R, \quad (23)$$

$$(-p\mathbf{I} + \tau)\hat{\mathbf{n}} = 0 \quad \text{on } \partial\Omega_L, \quad (24)$$

$$\nabla \cdot \mathbf{u}_C + \nabla \cdot \mathbf{u}_R = 0 \quad \text{on } \Omega_L, \quad (25)$$

$$\varepsilon(\mathbf{u}_C) - \frac{1}{2\mu}\tau + \varepsilon(\mathbf{u}_R) = 0 \quad \text{on } \Omega_C, \quad (26)$$

noting that  $\mathbf{u}_C$ ,  $p$ , and  $\tau$  are defined only on  $\Omega_C$ , and  $\mathbf{u}_R$  is defined only on  $\Omega_R$ . Because our new representation is immersed inside the fluid far from the physical boundary, we do not need any special treatment of the boundary conditions, and simply reuse the free surface and solid boundary conditions from the fully uniform method of Larionov et al. [2017].

Discretizing the optimality conditions for our objective, and its solid boundary condition analog, results in the matrix-vector equation:

$$\begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}_C & 0 & \mathbf{G} & \mathbf{D}^\top \\ 0 & \frac{1}{\Delta t}\mathbf{M}_R + 2\mathbf{J}\mathbf{D}^\top\mu\mathbf{D}\mathbf{J}^\top & \mathbf{J}\mathbf{G} & \mathbf{J}\mathbf{D}^\top \\ \mathbf{G}^\top & \mathbf{G}^\top\mathbf{J}^\top & 0 & 0 \\ \mathbf{D} & \mathbf{D}\mathbf{J}^\top & 0 & -\frac{1}{2}\mu^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_C \\ \mathbf{v}_R \\ \mathbf{p} \\ \tau \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}_C\mathbf{u}_C^* \\ \frac{1}{\Delta t}\mathbf{M}_R\mathbf{v}_R^* \\ 0 \\ 0 \end{bmatrix}. \quad (27)$$

We follow Goldade et al. [2020] in using a full uniform grid over the whole fluid domain for applying advection and external forces. Therefore, before solving the Stokes system, we set up the interior tile structure of reduced regions and for each tile estimate  $\mathbf{v}_R^*$  by a least squares fit to the Cartesian velocities that it is replacing, according to Equation 14. Transferring back to the Cartesian samples afterwards simply requires evaluating the quadratic field at each velocity sample.

### 3.6 Alternative Matrix-Vector Forms

This system is amenable to transformation into more convenient forms via Schur complements. Stress degrees of freedom may be eliminated to produce the pressure-velocity form:

$$\begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}_C + \mathbf{V} & \mathbf{VJ}^\top & \mathbf{G} \\ \mathbf{JV} & \frac{1}{\Delta t} \mathbf{M}_R + \mathbf{JVJ}^\top & \mathbf{JG} \\ \mathbf{G}^\top & \mathbf{G}^\top \mathbf{J}^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_C \\ \mathbf{v}_R \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}_C \mathbf{u}_C^* \\ \frac{1}{\Delta t} \mathbf{M}_R \mathbf{v}_R^* \\ 0 \end{bmatrix} \quad (28)$$

where  $\mathbf{V} = 2\mathbf{D}^\top \boldsymbol{\mu} \mathbf{D}$  is the discrete volume-weighted viscosity operator.

We opted instead to eliminate the Cartesian velocities and subsequently the generalized velocities to produce the pressure-stress form

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \mathbf{G}^\top \mathbf{u}_C^* + \Delta t \mathbf{G}^\top \mathbf{J}^\top \mathbf{B}^{-1} \mathbf{M}_R \mathbf{v}_R^* \\ \mathbf{D} \mathbf{u}^* + \Delta t \mathbf{D} \mathbf{J}^\top \mathbf{B}^{-1} \mathbf{M}_R \mathbf{v}_R^* \end{bmatrix}, \quad (29)$$

where

$$\mathbf{A}_{11} = \Delta t \mathbf{G}^\top \mathbf{M}_C^{-1} \mathbf{G} + \mathbf{G}^\top \mathbf{J}^\top \mathbf{B}^{-1} \mathbf{J} \mathbf{G} \quad (30)$$

$$\mathbf{A}_{12} = \Delta t \mathbf{G}^\top \mathbf{M}_C^{-1} \mathbf{D}^\top + \mathbf{G}^\top \mathbf{J}^\top \mathbf{B}^{-1} \mathbf{J} \mathbf{D}^\top \quad (31)$$

$$\mathbf{A}_{21} = \Delta t \mathbf{D} \mathbf{M}_C^{-1} \mathbf{G} + \mathbf{D} \mathbf{J}^\top \mathbf{B}^{-1} \mathbf{J} \mathbf{G} \quad (32)$$

$$\mathbf{A}_{22} = \Delta t \mathbf{D} \mathbf{M}_C^{-1} \mathbf{D}^\top + \mathbf{D} \mathbf{J}^\top \mathbf{B}^{-1} \mathbf{J} \mathbf{D}^\top + \frac{1}{2} \boldsymbol{\mu}^{-1} \quad (33)$$

$$\mathbf{B} = \frac{1}{\Delta t} \mathbf{M}_R + 2 \mathbf{J} \mathbf{D}^\top \boldsymbol{\mu} \mathbf{D} \mathbf{J}^\top \quad (34)$$

The pressure-stress formulation is appealing as it is guaranteed to be symmetric positive-definite (SPD), and consequently allows for more efficient linear algebra techniques, notably preconditioned conjugate gradient. This system differs from the SPD system given by Larionov et al. [2017] only by the second term of each  $\mathbf{A}$  block. The  $\mathbf{B}$  matrix here has a block diagonal structure and mutually couples all pressure and viscous stress samples around each interior region's boundary. Assuming that viscous stencils of one interior region do not reach into a different interior region, i.e., the distance between any two regions is at least  $2\Delta x$ , then we can independently invert each  $26 \times 26$  block, one for each interior region.

The inverses of these blocks are dense, which could become problematic for performance when distributed out with the surrounding matrix multiplies. Instead of assembling the entire matrix  $\mathbf{A}$  before solving, we can instead store each matrix factor and perform matrix-vector multiplies from right to left when the action of  $\mathbf{A}$  is required [Bridson 2015]. Because of the small size of  $\mathbf{B}$ , this approach significantly reduces the computational cost of multiplying by  $\mathbf{A}$ . We found that for a sample matrix-vector multiply with 115395 degrees of freedom, taken from real simulation data, the naïve multiply takes 16193ms while the factored takes 2637ms, corresponding to a factor of 6 improvement.

Regarding implementation, our reduced model can be easily added to an existing variational Stokes solver [Larionov et al. 2017]. The second terms of each  $\mathbf{A}_{ij}$  block all have  $\mathbf{J}^\top \mathbf{B}^{-1} \mathbf{J}$  in place of the  $\mathbf{M}_C^{-1}$  in the first term. This means we can simply apply the same iteration over the fluid domain as variational Stokes, replacing any faces that fall on the reduced regions with the scalar given by  $\mathbf{J}^\top \mathbf{B}^{-1} \mathbf{J}$ .

Which of the lower two videos more closely resembles the top video?

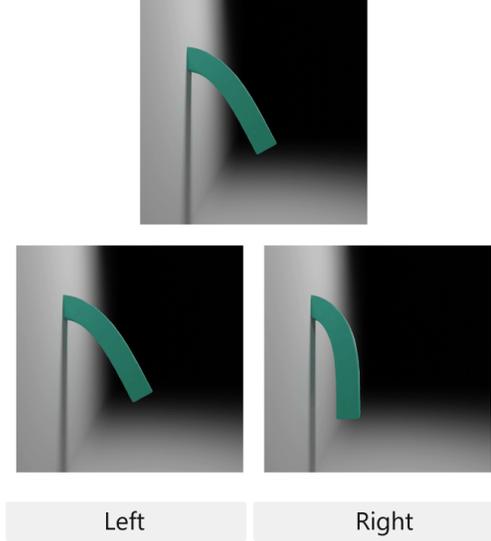


Fig. 5. A trial question from our two-alternative forced choice study.

## 4 PERCEPTUAL EVALUATION

Prior work introducing new adaptivity or model-reduction techniques has often left the task of qualitatively assessing the resulting *visual error* to the reader, or suggested (perhaps implicitly) that such error is negligible. Others have relied on numerical or geometric error measures for evaluation, though these may be only indirectly related to human perception.

We seek to more directly demonstrate that the performance improvements offered by our proposed technique are not accompanied by significantly deteriorated visual quality. We follow the path paved by Um et al. [2017], who performed perceptual evaluations on fluid animation, comparing several existing techniques to reference real-world videos. Taking a cue from their work, we have conducted a two-alternative forced choice study that compares our reduced method to the results from a standard fully refined uniform grid Stokes-based solver.

To more precisely assess the quality difference between our method and the standard solver, we additionally perform simulations with the standard solver using perturbed viscosity coefficients, and include these results in our comparisons. Should any visual differences compared to the baseline method be significant, this additional data gives us a measure of how far away our method is from the baseline, by comparing the perceptual distance of our method to the baseline with the perceptual distance of the baseline to some known viscosity perturbation.

### 4.1 Survey Methodology

We conduct our perceptual study by presenting participants with a video of a cantilevered viscous beam simulated by a baseline fully

Table 1. **Simulation methods used for our 2-alternative forced choice study.**

Label	Method
$U_n$	Fully uniform method with n% viscosity perturbation
$U$	Baseline fully uniform method
$PS$	Our reduced method

refined uniform method, along with two options. Each option contains a video of the same scene simulated with either our reduced method, the baseline method, or one of the viscosity perturbations of baseline method, as given in Table 1. For each question the two options are selected randomly with replacement. We ask the viewer to select which of the two options more closely resembles the reference video, having presented the three videos (the reference and two options) simultaneously. Our question format is shown in Figure 5.

Each viewer is asked ten of these questions. Given enough participants, this approach provides us with a count of how many viewers selected each option for every potential pair of methods, including comparisons of a method against itself. Given the large number of participants required, we perform this survey on the crowd-sourcing platform Prolific, filtering for English-speaking participants on desktop clients.

To improve the quality of our survey data, we include three practice questions before our trials to introduce participants to our question format. This acts as a comprehension check to ensure viewers understand the task. After the three sample questions, we enforce a thirty second washout period before allowing participants to move on to the trial questions.

In order to mitigate against potential data noise from random clicking responses, we include an attention check at the middle and the end, where the reference video is compared against a half-speed simulation.

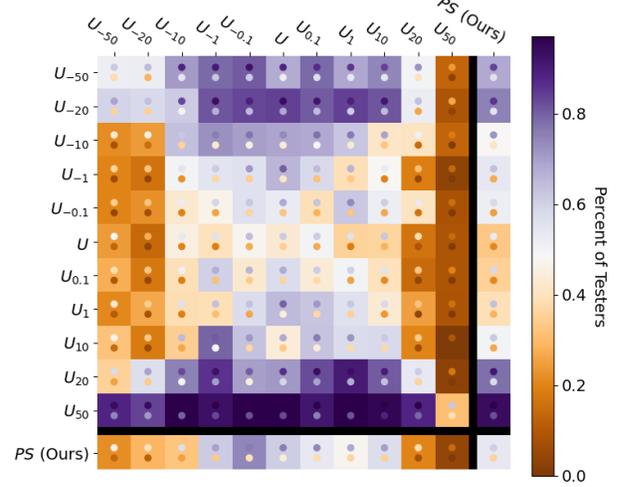
Only survey results from users correctly passing both the comprehension checks at the start and the awareness tests at the middle and end were used for analysis. Out of 500 participants, 440 were considered admissible.

## 4.2 Survey Results

Our survey provides us with a count of which option users selected when comparing each pair to the baseline method. We employ bootstrap resampling to find confidence intervals (CI) on these measurements. Note that in order to account for potential intra-user bias, we perform this resampling on a per-user basis. That is, we treat a single user’s entire response set as a single sample during bootstrapping, rather than sampling individual question responses.

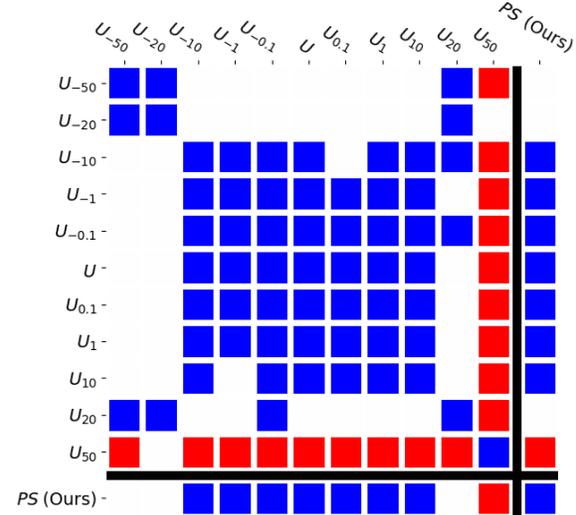
We also recorded the order of the pairs during data collection—i.e., we distinguish a comparison of A against B from that of B against A. Looking at comparisons of identical pairs (for example,  $U_{-20}$  vs.  $U_{-20}$ ), we found a significant left-right framing bias, where the left option was favoured at a 56.2% probability, with a 95%CI of 50.3%-63.7%. This kind of framing bias has been previously observed in the psychophysics literature when comparing similar or identical information [Robertson and Lunn 2020], or when users are under

Testers that chose the column method over the row method as being closer to the zero perturbation simulation.



(a)

Significant perceptual distances in pairwise comparisons



(b)

Fig. 6. **Results for our 2AFC study on a cantilevered viscous beam scene.** Column labels were presented on the left and row labels were presented on the right. (a): Heatmap with the percentage of users that selected the column method over the row method shown as the box colour, 95% confidence intervals shown as small circles for each box. (b): A plot indicating significantly different perceptual distances in red (95% intervals that are either entirely above 75% or entirely below 25%), and equal perceptual distances in blue (95% confidence interval crosses 50%). The diagonals being coloured blue validates the coinflip symmetrization for identical pair comparisons. Method labels are as described in Table 1

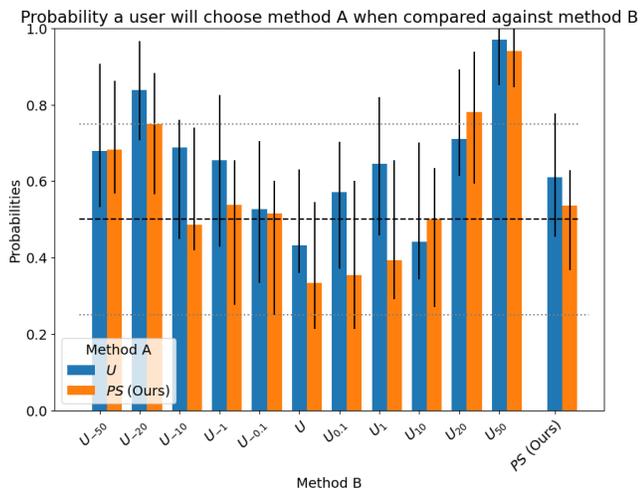


Fig. 7. A comparison of user responses for the reference method (0.0) and our reduced method (R) when paired against each of the other methods. Solid black lines indicate 95% confidence intervals, dashed black line indicates the 50% threshold, and dotted grey lines indicate the 75% and 25% thresholds. For instance, the orange bar at 10.0 represents the probability a user would select our reduced method when presented in a choice against a 10% viscosity perturbation. The blue bar at 10.0 represents the probability a user would select the reference method when compared to the same 10% viscosity perturbation.

time pressure [Roskes et al. 2011]. To counterbalance this framing bias, we symmetrize the mirrored pairs via coin flipping. Whenever an A vs. B sample is used as a measurement in the bootstrapping analysis, a coin is flipped to decide whether it is added to the A vs. B bin, or the B vs. A bin. Note that this does not affect what answer the user gave, just which bin of the two symmetric bins it is counted towards. Using this coinflip symmetrization ensured that CI’s for identical pair comparisons included 50%, verifying that it works to counterbalance our measured framing bias.

We summarize this symmetrized data in Figure 6. Figure 6a presents a heat map of the percentage of users that selected the column method when compared against the row method along with this values’ 95% confidence intervals. Figure 6b indicates pairs that are perceptually distant (red) and pairs that are perceptually close (blue) according to this data. We consider pairs to be perceptually distant if the confidence intervals fall either completely above the 75% threshold or completely below the 25% threshold. This means that when comparing these two against the baseline, users preferentially select one over the other more than 75% of the time, a threshold we borrow from the psychophysics literature [Eilertsen et al. 2016; Ulrich and Miller 2004]. We consider pairs to be perceptually close if their confidence intervals cross the 50% threshold.

To more directly compare our method (PS) to the baseline method (U), we extract their corresponding column slices from the heatmap and plot them together in bar form in Figure 7. This chart reveals that there is no significant perceptual difference between the baseline

Table 2. Scene parameters for presented examples. All examples used a padding size of 3 cell widths.

Scene	$\mu$	$\rho$	dx	Max DoFs	Tilesize
Armadillo	2000	1000	1.5e-3	3.2e6	32
Honey Coil	35	1000	1e-3	1.3e6	16
Jam Jar	400	1000	2e-3	1.7e6	32
Toothpaste	50	1000	5e-3	1e6	16
Viscous Beam	100	1	1/256	6.9e5	16
Octopus	1000	500	3e-3	5.4e5	32

method and our method, given that their 95%CI’s overlap when compared against any of the viscosity perturbations. Looking at the comparison against  $U_{50}$ , we see that both the baseline and our method are perceptually distinguishable against such a large viscosity perturbation. For comparisons against smaller viscosity perturbations—anything between  $U_{-10}$  and  $U_{10}$ —neither the baseline method nor our method is perceptually distinguishable.

Taken together, this implies that the error incurred by using our reduced method is less significant than a 20% change in viscosity. Furthermore, because perturbations smaller than 20% are indistinguishable from the baseline method, we can conclude that our method is perceptually competitive with the baseline.

## 5 REDUCED MODEL RESULTS

Our method was implemented as a plugin that replaces the pressure and viscosity steps in Houdini’s fluid simulator [SideFX 2022]. All other aspects of the fluid solver were left unmodified. Scene parameters are given in Table 2; the scenes were run on a desktop with a 4-core R5 1400 CPU with 32GB RAM. All reduced simulations were run with 3-cell padding size.

### 5.1 Piling Armadillos

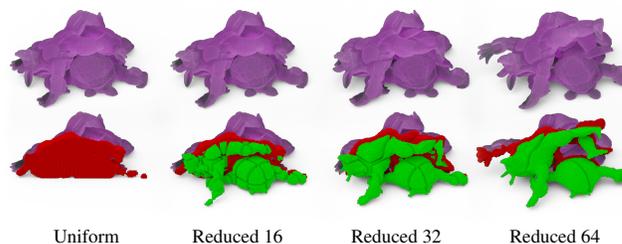


Fig. 8. Piling viscous armadillos at frame 180 using the fully uniform method, and our method with three different tile sizes: 16, 32, and 64. Rendered with the simulated fluid surface (above), and a cutaway view showing the reduced tiles (below). Note the erroneous stiffness of the 64 tile size, due to tiles connecting limbs to the bulk of the body.

We simulate dropping highly viscous armadillos using three different tile sizes to demonstrate the influence of tile size on simulation output and runtime. Renders are shown in Figure 8 alongside a cutaway view showing the reduced regions. Comparisons for runtime costs and number of degrees of freedom are shown on Figure 9.

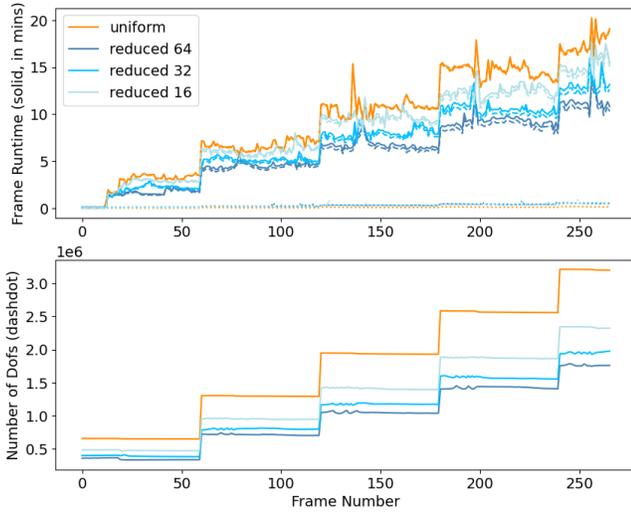


Fig. 9. **Runtime (above) and degree of freedom (below)** for the armadillo drop example using the fully uniform method, and our method with three different tile sizes: 16, 32, and 64. Runtime is split into solve-only cost (dashed), setup cost (dotted), and total cost (solid).

As expected, larger tile sizes allow for greater degree of freedom (DoF) reductions, with a corresponding runtime cost reduction. Although smaller tile sizes do require more reduced regions, each contributing additional DoFs, much of the difference is from extra uniform cells required in the padding between tiles. We also note that we notice no effect of tile size on the number of iterations required for the conjugate gradient solve. The number of DoFs saved thus directly corresponds to a cost reduction, by reducing the cost of each matrix-vector multiply.

There is, however, a resolution limit to the size of the tiles, as shown on Figure 8. Each reduced region is limited in its resolvable field; tiles large enough to connect a limb to the bulk of the body, as observed for a tile size of 64, do not have enough freedom to allow the limbs to move independently, thus exhibiting a locking artifact.

## 5.2 Dough Octopus

We simulate throwing a doughy octopus at a wall. This scene is similar to the armadillo example but the octopus undergoes a larger range of deformations due to being stretched under gravity. Our method is again able to closely match the result of the fully uniform solver. 120 frames were simulated in 198.0 minutes by a fully uniform method, and 139.6 minutes by our reduced method. Average degrees of freedom were reduced from 528K to 302K.

## 5.3 Honey Coil

Since we retain the uniform grid on the surface of the liquid, replacing only interior domains with reduced polynomial fields, our method is able to retain the desirable free surface accuracy of the standard Stokes solver. We demonstrate this claim by replicating the liquid rope coiling instability, as shown in Figure 1.

We additionally demonstrate that at least a quadratic field is required by comparing against an affine simulation, as shown in

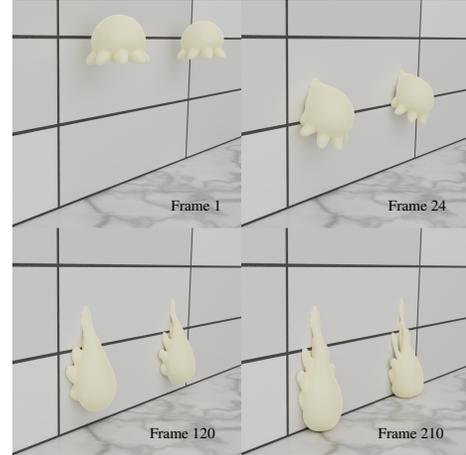


Fig. 10. **Doughy octopus** thrown at a wall at frames 1, 24 (immediately after impact), 120, and 210. Left octopus is simulated with the fully uniform solver, and right octopus with our reduced method.

Figure 11. The affine method creates a much more erratic coil, especially visible at early stages (frames 120 and 240). This failure to correctly simulate the coiling instability is consistent our observations in Section 3.3. Recall that affine fields were shown to lack proper representation of viscous stresses. Thus although the near-surface cells satisfy the proper free surface condition required to resolve the instability (given that they are simulated with the uniform Stokes discretization), there are reduced affine blocks inside the fluid that cannot evolve in a sufficiently accurate manner to capture the correct effect.

## 5.4 Conveyor Belt Toothpaste

Figure 12 shows a similar liquid coiling test placed on a moving conveyor belt. Like that of Larionov et al. [2017], our method is able to capture some of the various "sewing patterns" that arise as the belt speed changes [Bergou et al. 2010]. This demonstrates that our method respects the moving solid boundary conditions of the standard Stokes solver without any change in its treatment. Because reduced regions are placed sufficiently far in the interior of the fluid, any variational (volume fraction) weights involved in resolving the moving solid boundary condition only interact with the uniform grid domain.

## 5.5 Jam Jar

We drop a pre-fractured glass jar filled with jam. The glass shards were simulated using Houdini's built-in rigid body solver, and were allowed to interact with the fluid via Houdini's soft coupling mechanism. We see that our method naturally handles these mutual interactions.

## 5.6 Viscous Beam Timing

We also performed a timing test on the cantilevered viscous beam scene ( $\rho = 1$ ,  $\mu = 100$ ). A fully uniform treatment of the scene contains 695K DoFs; our method nearly halves that to 354K DoFs.

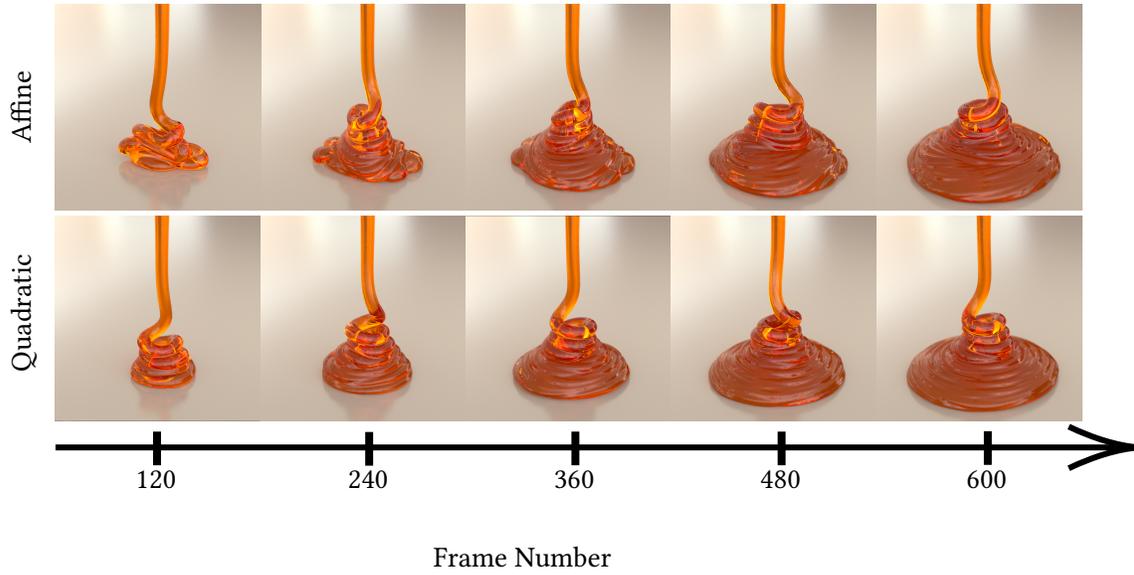


Fig. 11. **Frame sequence of a liquid rope-coiling instability** simulated using an affine reduced model (top) and a quadratic reduced model (bottom). The affine model deviates from the expected coiling achieved by the quadratic model, especially in the first two image pairs.

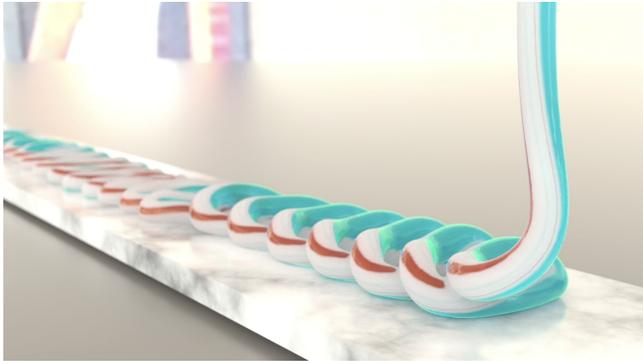


Fig. 12. **Snapshot of a liquid rope coil on a conveyor belt.**



Fig. 13. **Jam jar drop** on frame 250, simulated using our reduced method.

This halving in degrees of freedom was associated with a similar decrease in runtime of the Stokes solve, reducing an average frame-time of 1084s for the fully uniform method to 607s for our method.

This runtime reduction is largely from savings in the linear solve step, where the reduced number of DoFs decreases the size of each matrix-vector multiply in the conjugate gradient (CG) iteration.

Our method does, however, require extra cost in setting up the reduced system, with the average setup time (all steps in the Stokes solve excluding the CG solve) increasing from 58s for the standard method to 204s in our method. Our method is therefore primarily useful for simulations where the setup cost is negligible relative to the solve cost, which is typical for high resolution simulations; or for geometries where the DoFs can be greatly reduced, such as those with large bulk fluids (i.e., lower surface-to-volume ratio).

## 6 DISCUSSION AND CONCLUSIONS

We have presented the first acceleration method for unified Stokes simulation, leveraging incompressible polynomial velocity fields on coarse interior tiles. Our two-alternative forced choice study has demonstrated that errors associated with our reduction method do not deteriorate visual quality in a perceptually significant manner. We have additionally shown that our quadratic model overcomes limitations of a simpler affine model, and that the reduction in degrees of freedom resulting from our method yields reduced computational costs compared to the uniform discretization.

We envision several avenues for further performance increases that were not yet explored in this work. We used a single global tile size to dictate the maximum dimensions of our reduced regions [Goldade et al. 2020]. Given that these reduced regions constitute a kind of spatial adaptivity approach, it is possible to introduce *tile-level* adaptivity. That is, allow the maximum size of the interior reduced tiles to vary spatially. One could imagine either a simple

approach of using larger tiles deeper in the fluid, or a more "physics-aware" approach of using the least squares residual of the polynomial velocity fit to direct which tiles should be coarsened or refined.

In our test cases, we showed that Hessian information is required to accurately treat viscous forces in the interior reduced domains, requiring us to use a quadratic velocity field. In cases where viscous forces may not be significant—in very still flows for example—an affine field may be sufficient, and preferable given its fewer degrees of freedom. Ideas of  $p$ -adaptivity may thus be applied [Edwards and Bridson 2014], where tiles are free to add or subtract degrees of freedom when deemed necessary.

Finally, we see a fruitful future for the use of perceptual studies as a tool for evaluating (or even developing) new simulation techniques. Much emphasis has traditionally been previously placed on numerical measures of error, and while such metrics are useful for assessing mathematical accuracy, in many graphics applications the viewer is our final adjudicator. As such, more value should be placed on measuring errors in this perceptual space. For example, a concrete experiment would be to test how high the residual tolerance for conjugate gradient can be before becoming visually distinguishable. While heuristics have suggested a standard of  $10^{-4}$ , it may be possible to get away with a much larger tolerance. Along the same lines, Harrison et al. [2004] showed that, in some cases, animated character's limb lengths can change by as much as 20% before viewers will notice; exploiting perceptual factors might allow similarly extreme liquid volume changes to be rendered invisible.

## ACKNOWLEDGMENTS

This research was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), including a Canada Graduate Scholarship-Doctoral Program, and Discovery Grants [RGPIN-2021-03733, RGPIN-2021-02524, RGPIN-2017-05524]; as well as by the Connaught Fund [503114], the Ontario Early Researchers Award [ER19-15-034], and the Canada Research Chairs Program. We thank SideFX Software for providing Houdini software licenses, as well as Adobe Research and Autodesk for their gifts. Finally, we thank members of the Dynamic Graphics Project lab for the insightful conversations.

## 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.* 36, 4 (2017), 1–12.
- Ryoichi Ando, Nils Thürey, and Chris Wojtan. 2015. A dimension-reduced pressure solver for liquid simulations. In *Comput Graph Forum*, Vol. 34. Wiley Online Library, 473–480.
- Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.* 26, 3 (2007), 100–es.
- 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 *ACM Trans. Graph.* 30, 4 (2011), 1–7.
- Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete viscous sheets. *ACM Trans. Graph.* 31, 4 (2012), 1–7.
- Christopher Batty, Stefan Xenos, and Ben Houston. 2010. Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. In *Comput. Graph. Forum*, Vol. 29. Wiley Online Library, 695–704.
- Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Trans. Graph.* 29, 4 (2010), 1–10.

- Jeremiah U Brackbill and Hans M Ruppel. 1986. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J. Comput. Phys.* 65, 2 (1986), 314–343.
- Robert Bridson. 2015. *Fluid simulation for computer graphics*. CRC press.
- Tyson Brochu, Christopher Batty, and Robert Bridson. 2010. Matching fluid simulation elements to surface geometry and topology. In *ACM SIGGRAPH 2010 papers*. 1–9.
- Mark Carlson, Peter J Mucha, R Brooks Van Horn III, and Greg Turk. 2002. Melting and flowing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 167–174.
- Nuttapong Chentanez, Bryan E Feldman, François Labelle, James F O'Brien, and Jonathan R Shewchuk. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 219–228.
- Qiaodong Cui, Pradeep Sen, and Theodore Kim. 2018. Scalable laplacian eigenfluids. *ACM Trans. Graph.* 37, 4 (2018), 1–12.
- Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-only liquids. *ACM Trans. Graph.* 35, 4 (2016), 1–12.
- Fernando de Goes, Corentin Wallez, Jin Huang, Dmitry Pavlov, and Mathieu Desbrun. 2015. Power particles: an incompressible fluid solver based on power diagrams. *ACM Trans. Graph.* 34, 4 (2015), 50–1.
- Tyler De Witt, Christian Lessig, and Eugene Fiume. 2012. Fluid simulation using Laplacian eigenfunctions. *ACM Trans. Graph.* 31, 1 (2012), 1–11.
- Essex Edwards and Robert Bridson. 2014. Detailed water with coarse grids: combining surface meshes and adaptive discontinuous Galerkin. *ACM Trans. Graph.* 33, 4 (2014), 1–9.
- Gabriel Eilertsen, Jonas Unger, and Rafal K Mantiuk. 2016. Evaluation of tone mapping operators for HDR video. In *High dynamic range video*. Elsevier, 185–207.
- Henrik Fält and Douglas Roble. 2003. Fluids with extreme viscosity. In *ACM SIGGRAPH 2003 Sketches & Applications*. 1–1.
- Bryan E Feldman, James F O'Brien, and Bryan M Klingner. 2005. Animating gases with hybrid meshes. *ACM Trans. Graph.* 24, 3 (2005), 904–909.
- Nick Foster and Dimitri Metaxas. 1996. Realistic animation of liquids. *Graph. models image process.* 58, 5 (1996), 471–483.
- Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. 2017. A polynomial particle-in-cell method. *ACM Trans. Graph.* 36, 6 (2017), 1–12.
- Ryan Goldade, Mridul Aanjaneya, and Christopher Batty. 2020. Constraint bubbles and affine regions: reduced fluid models for efficient immersed bubbles and flexible spatial coarsening. *ACM Trans. Graph.* 39, 4 (2020), 43–1.
- 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 (2019), 1–14.
- Francis H Harlow. 1962. *The particle-in-cell method for numerical solution of problems in fluid dynamics*. Technical Report. Los Alamos Scientific Lab., N. Mex.
- Francis H Harlow and J Eddie Welch. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids* 8, 12 (1965), 2182–2189.
- Jason Harrison, Ronald A Rensink, and Michiel Van De Panne. 2004. Obscuring length changes during animated motion. *ACM Trans. Graph.* 23, 3 (2004), 569–573.
- Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Trans. Graph.* 34, 4 (2015), 1–10.
- Bryan M Klingner, Bryan E Feldman, Nuttapong Chentanez, and James F O'Brien. 2006. Fluid animation with dynamic meshes. In *ACM SIGGRAPH 2006 Papers*. 820–825.
- Egor Larionov, Christopher Batty, and Robert Bridson. 2017. Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids. *ACM Trans. Graph.* 36, 4 (2017), 1–11.
- Frank Losasso, Frédéric Gibou, and Ron Fedkiw. 2004. Simulating water and smoke with an octree data structure. In *ACM SIGGRAPH 2004 Papers*. 457–462.
- Neil M Ribe, Mehdi Habibi, and Daniel Bonn. 2012. Liquid rope coiling. *Annual review of fluid mechanics* 44 (2012), 249–266.
- Deirdre A Robertson and Peter D Lunn. 2020. The effect of spatial location of calorie information on choice, consumption and eye movements. *Appetite* 144 (2020), 104446.
- Marieke Roskes, Daniel Sligte, Shaul Shalvi, and Carsten KW De Dreu. 2011. The right side? Under time pressure, approach motivation leads to right-oriented bias. *Psychol. Sci.* 22, 11 (2011), 1403–1407.
- Han Shao, Libo Huang, and Dominik L Michels. 2022. A Fast Unsmoothed Aggregation Algebraic Multigrid Framework for the Large-Scale Simulation of Incompressible Flow. (2022).
- Lin Shi and Yizhou Yu. 2004. Visual smoke simulation with adaptive octree refinement. In *Computer Graphics and Imaging*. 13–19.
- SideFX. 2022. *Houdini*.
- Funshing Sin, Adam W Bargteil, and Jessica K Hodgins. 2009. A point-based method for animating incompressible flow. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics symposium on computer animation*. 247–255.
- Jos Stam. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 121–128.

- Shawqi M Suleiman and Bruce R Munson. 1981. Viscous buckling of thin fluid layers. *Phys. Fluids* 24, 1 (1981), 1–5.
- Tetsuya Takahashi and Christopher Batty. 2020. Monolith: a monolithic pressure-viscosity-contact solver for strong two-way rigid-rigid rigid-fluid coupling. (2020).
- Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C Lin. 2015. Implicit formulation for SPH-based viscous fluids. In *Comput. Graph. Forum*, Vol. 34. Wiley Online Library, 493–502.
- Tetsuya Takahashi and Ming C Lin. 2019. A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling. In *Comput Graph Forum*, Vol. 38. Wiley Online Library, 49–58.
- Michael Tao, Christopher Batty, Mirela Ben-Chen, Eugene Fiume, and David IW Levin. 2022. VEMPIC: particle-in-polyhedron fluid simulation for intricate solid boundaries. *ACM Trans. Graph.* 41, 4 (2022), 1–22.
- Adrien Treuille, Andrew Lewis, and Zoran Popović. 2006. Model reduction for real-time fluids. *ACM Trans. Graph.* 25, 3 (2006), 826–834.
- Rolf Ulrich and Jeff Miller. 2004. Threshold estimation in two-alternative forced-choice (2AFC) tasks: The Spearman-Kärber method. *Percept Psychophys.* 66, 3 (2004), 517–533.
- Kiwon Um, Xiangyu Hu, and Nils Thuerey. 2017. Perceptual evaluation of liquid simulation methods. *ACM Trans. Graph.* 36, 4 (2017), 1–12.
- Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A physically consistent implicit viscosity solver for SPH fluids. In *Comput. Graph. Forum*, Vol. 37. Wiley Online Library, 145–155.
- Martin Wicke, Matt Stanton, and Adrien Treuille. 2009. Modular bases for fluid dynamics. *ACM Trans. Graph.* 28, 3 (2009), 1–8.
- Yongning Zhu and Robert Bridson. 2005. Animating sand as a fluid. *ACM Trans. Graph.* 24, 3 (2005), 965–972.