Monolith: A Monolithic Pressure-Viscosity-Contact Solver for Strong Two-Way Rigid-Rigid Rigid-Fluid Coupling

TETSUYA TAKAHASHI, Adobe, University of Maryland at College Park, University of North Carolina at Chapel Hill
CHRISTOPHER BATTY, University of Waterloo

Fig. 1. Our Monolith solver enables efficient and robust two-way simultaneous rigid-rigid and rigid-fluid coupling. (Left) Two hollow glass spheres containing inviscid liquid roll around within a basin as the liquid slides and splashes. (Middle) A boat carrying multiple loads is perturbed by ocean waves. (Right) When the glass spheres instead contain viscous liquid, the no-slip boundary condition, viscosity, and friction together bring the spheres more quickly to rest.

We propose *Monolith*, a monolithic pressure-viscosity-contact solver for more accurately, robustly, and efficiently simulating non-trivial two-way interactions of rigid bodies with inviscid, viscous, or non-Newtonian liquids. Our solver simultaneously handles incompressibility and (optionally) implicit viscosity integration for liquids, contact resolution for rigid bodies, and mutual interactions between liquids and rigid bodies by carefully formulating these as a single unified minimization problem. This monolithic approach reduces or eliminates an array of problematic artifacts, including liquid volume loss, solid interpenetrations, simulation instabilities, artificial "melting" of viscous liquid, and incorrect slip at liquid-solid interfaces. In the absence of solid-solid friction, our minimization problem is a Quadratic Program (QP) with a symmetric positive definite (SPD) matrix and can be treated with a single Linear Complementarity Problem (LCP) solve. When friction is present, we decouple the unified minimization problem into two subproblems so that it can be effectively handled via staggered projections with alternating LCP solves. We also propose a complementary approach for non-Newtonian fluids which can be seamlessly integrated and addressed during the staggered projections. We demonstrate the critical importance of a *contact-aware*, unified treatment of fluid-solid coupling and the effectiveness of our proposed Monolith solver in a wide range of practical scenarios.

CCS Concepts: • **Computing methodologies** → **Physically-based simulation**; **Monolithic coupling**.

Additional Key Words and Phrases: Fluid simulation, monolithic coupling, viscosity

Authors' addresses: Tetsuya Takahashi, Adobe, University of Maryland at College Park, University of North Carolina at Chapel Hill, ttakahas@adobe.com; Christopher Batty, University of Waterloo, christopher.batty@uwaterloo.ca.

0730-0301/2020/12-ART182 $15.00
https://doi.org/10.1145/3414685.3417798

ACM Reference Format:
Tetsuya Takahashi and Christopher Batty. 2020. Monolith: A Monolithic Pressure-Viscosity-Contact Solver for Strong Two-Way Rigid-Rigid Rigid-Fluid Coupling. *ACM Trans. Graph.* 39, 6, Article 182 (December 2020), 19 pages. https://doi.org/10.1145/3414685.3417798

1 INTRODUCTION

From water pouring into a cup, to honey being stirred with a spoon, to ships cruising in the ocean, diverse interactions between liquids and solid objects are pervasive in everyday life. At the same time, if we consider the cup being placed on a table, the spoon colliding with the sides of the honey jar, or the cargo load being carried by the ship, it is clear that the solid objects in these scenarios also interact with each other through surface contacts, which are similarly ubiquitous in our world. For various applications, including computer animation, virtual reality, mechanical engineering, and medical training, simulating these fundamental phenomena is essential.

In the fluid animation literature, various grid-based two-way coupling methods have been proposed to simulate the interactions between fluids and rigid objects, e.g., [Batty et al. 2007; Guendelman et al. 2005; Klingner et al. 2006; Robinson-Mosher et al. 2011, 2008]. While these approaches form a unified system to determine the correct exchange of forces between fluids and *individual* dynamic solids, they typically ignore the interactions *between* solid objects (i.e., these are treated in a separate calculation). As such, for example, when simulating a ship carrying a load of cargo, the ship is not aware of its cargo when resolving the two-way ocean-ship coupling. Thus, the ocean pushes up on the ship without considering the cargo's presence or weight, which can lead to spurious penetrations between the ship and its cargo, and hence severe stability issues. Similarly, when one addresses the contact between solid objects, the existence of fluids is seldom considered, i.e., the cargo is unaware of the ocean, and so weighs down on the ship without considering the water that supports both. Consequently, the computed contact impulses will push the ship downwards into the ocean with no

Fig. 2. Bunny-shaped inviscid liquid dropped into a bowl on the ground, with various options for pressure (P) and contact (C) coupling (see §6.1). (Left) P-C solve (decoupled). (Middle left) C-P solve (decoupled). (Middle right) PC-iterative solve (iterated weak coupling). (Right) Our proposed PC-unified solve (strong coupling). P-C solve violates fluid incompressibility, causing volume loss. C-P solve violates solid non-penetration, causing spurious oscillatory motion of the bowl. While both PC-iterative and PC-unified solves correctly preserve volume and handle bowl-ground contacts, PC-unified is approximately 10× faster.

resistance whatsoever, leading to potentially significant loss of fluid volume. Our core motivating observation is that to achieve correct dynamics for fluids, rigid bodies, and their interactions, even in such deceptively simple scenarios, it is vital to employ a monolithic approach that unifies the dynamics.

With these facts in mind, we propose a monolithic pressure-viscosity-contact solver that simultaneously handles incompressibility and (optionally) implicit viscosity integration for liquids, contact handling among rigid bodies, and mutual interactions between liquids and rigid bodies mediated by pressure and viscous forces. We take a variational approach, building each component as a minimization subproblem and unifying these subproblems to establish a single, monolithically coupled minimization problem. This monolithic scheme makes each subcomponent aware of the others, and thereby achieves more accurate, robust, and efficient simulations. In the absence of friction, our approach leads to a Quadratic Program (QP) with a sparse symmetric positive definite (SPD) matrix that can be efficiently handled by solving a single Linear Complementarity Problem (LCP) per step. With frictional contact added, our unified minimization problem can be decomposed into two subproblems, and efficiently solved using a staggered projections approach [Kaufman et al. 2008]; this requires iteratively solving a sequence of LCPs, yet preserves the attractive properties of the monolithic treatment.

We also propose a compatible approach to simulating a broad class of non-Newtonian fluids within our framework. We express implicit viscosity integration with nonlinear viscosity coefficients as a minimization problem, which helps to ensure the robustness of the simulation. To efficiently solve the minimization problem and seamlessly integrate it into our framework with staggered projections, we similarly decompose the problem into two subproblems that we solve in an alternating fashion.

In summary, our key contributions are:

- **Strong two-way rigid-fluid coupling within a unified pressure-viscosity (Stokes) solver** to realistically handle interactions between viscous fluids and individual rigid bodies via a single pressure-viscosity solve. We demonstrate that such a tightly-coupled Stokes approach to rigid-fluid interaction alleviates potentially severe artifacts for viscous liquids touching isolated dynamic rigid bodies.
- **A monolithic *contact-aware* treatment of fluid-solid interaction** that addresses both fluid internal forces (pressure, viscosity, or both) and the fluid-solid interactions noted above *at the same time as* contact handling among multiple rigid bodies. This monolithic solver significantly improves accuracy, robustness, and efficiency for scenes involving two-way rigid-rigid and rigid-fluid interactions, as compared to previous state-of-the-art techniques.
- **Staggered projections for non-Newtonian fluids**. This approach allows for treating non-Newtonian fluids within our framework along the lines of staggered projections for frictional contacts, while prioritizing the enforcement of incompressibility for liquids and non-penetration constraints for rigid bodies.

To the best of our knowledge, our proposed *Monolith* solver is the first method that handles constraints due to pressure, viscosity, and contacts, in a monolithic way for general two-way interactions of liquids and rigid bodies. Figure 1 demonstrates Monolith in action in a few complex scenarios.

2 RELATED WORK

Fluid and rigid body dynamics form the foundations of our method. The literature on each of these topics is extensive; we therefore review prior techniques for viscous fluid simulation and rigid body simulation in the supplementary material. In this section, we focus on prior two-way fluid-solid coupling methods which are relevant to our monolithic solver (§2.1), and discuss general monolithic and multiphysics coupling in physical simulation (§2.2).

2.1 Two-Way Fluid-Solid Coupling

To achieve two-way coupling between Eulerian fluids and Lagrangian solids, early approaches, like that of Guendelman et al. [2005], used weak two-way coupling schemes that alternately solve one-way fluid-to-solid and solid-to-fluid coupling. This can work well when either fluid or solid dominates the other in terms of density, but in scenarios with comparable densities, it is often necessary to use tiny time steps and/or many alternating one-way solves per step to achieve stable and plausible simulation. To improve robustness and stability, Klingner et al. [2006] presented a strong two-way coupling method that simultaneously considers both the rigid body and fluid dynamics within the pressure projection. This yields an SPD linear system, but their method relied on tetrahedral meshes to support irregular objects, leading to higher complexity and cost. Chentanez et al. [2006] presented a related strong two-way coupling method

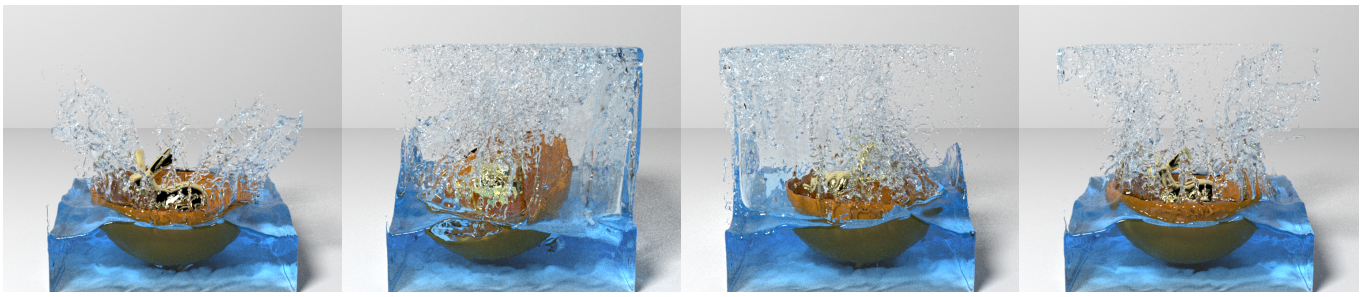

Fig. 3. A rigid bunny dropped onto a rigid bowl floating on inviscid liquid, with various options for pressure (P) and contact (C) coupling (see §6.1). (Left) P-C solve. (Middle left) C-P solve. (Middle right) PC-iterative solve. (Right) Our PC-unified solve. P-C solve violates incompressibility, generating smaller splashes and causing simulation failure. C-P solve experiences instabilities due to neglected contacts between the bunny and bowl, which causes sudden, intense impulses and topples the bowl. PC-iterative solve converges slowly and terminates before reaching a valid solution. The results exhibit instabilities due to violated incompressibility and non-penetration constraints. Our PC-unified solve generates plausible behaviors for the liquid, bowl, and bunny, with the bowl supporting the heavy bunny. The computational cost of PC-unified solve is comparable to P-C and C-P solves and at least 8× faster than PC-iterative.

for fluids and *deformable* objects. Their approach improves stability, but yields more costly non-symmetric linear systems.

Batty et al. [2007] proposed a strong two-way coupling method on Cartesian grids using a variational formulation leading to an SPD system. This method was extended for frictional forces in granular flow [Narain et al. 2010] and viscosity forces in Newtonian fluids [Takahashi and Lin 2019]. Robinson-Mosher et al. [2008] presented a two-way coupling approach for fluids interacting with both rigid and deformable solids by lumping the fluid and solid masses at their shared interface. They also argued for solving the unified system in symmetric indefinite form by treating solid velocities as unknown variables, because it increases sparsity compared to the method of Batty et al. [2007]. Later, this method was augmented to offer free-slip boundary conditions [Robinson-Mosher et al. 2009], to generate an SPD system including tightly coupled viscous forces (in Laplacian form) [Robinson-Mosher et al. 2011], and to incorporate a drag model for sub-grid solids [Hyde and Fedkiw 2019].

Due to the ubiquity of fluid-solid interactions, variations of both weak and strong two-way coupling have been extensively deployed in the graphics literature. A sampling of recent examples includes coupling with height field fluids [Jeschke et al. 2018], coupling with a stream function solver [Ando et al. 2015], coupling with compressible flow [Patkar et al. 2016], and coupling of articulated bodies to fluid [Lentine et al. 2011; Tan et al. 2011], among many others. However, these approaches do not consider simultaneous strong coupling alongside mutual rigid body contact.

While typical strong two-way coupling handles fluids and solids by solving a single unified problem, strong coupling can also be achieved by iteratively and alternatingly addressing the fluids and solids using weak coupling within one time step (which can be interpreted as solving the unified problem in block Gauss-Seidel fashion), although this may converge slowly. To accelerate convergence, Akbay et al. [2018] proposed an iterative (partitioned) two-way coupling method using a reduced-order model to interface between the materials.

Another popular approach is the immersed boundary method of Peskin [2002] which essentially treats fluids and solids as a single incompressible medium to simulate the two-way interaction. Philosophically similar approaches within the graphics community include the Rigid Fluid method of Carlson et al. [2004] and the purely Eulerian solid-fluid coupling scheme of Teng et al. [2016]. More recently, the immersed boundary method was augmented with a reduced elasticity solver to simulate fluid-solid interactions in real time [Brandt et al. 2019].

Under Material Point Method (MPM) schemes, collisions with different materials can be naturally handled by assigning different material properties to the particles. Because of this simplicity, two-way coupling has been achieved in various settings [Daviet and Bertails-Descoubes 2016; Ding and Schroeder 2020; Guo et al. 2018; Han et al. 2019; Hu et al. 2018; Klár et al. 2016; Yue et al. 2018]. Since the Lagrangian view is quite natural for solid objects, smoothed particle hydrodynamics (SPH) methods offer another consistent framework for formulating coupling problems. Significant effort has been applied to improving stability, robustness, and smoothness of boundary behavior, including two-way rigid-fluid interactions [Akinci et al. 2012; Band et al. 2018a,b; Becker et al. 2009; Koschier and Bender 2017]; Koschier et al. [2019] provide a broader review of the state of the art in SPH. Although SPH schemes are largely outside the scope of our paper, the method of Gissler et al. [2019] is pertinent, as discussed below.

The preceding discussion shows that many strong coupling methods have been proposed, but nearly all decouple collision-handling between solids from the fluid-solid coupling treatment. An important exception is the SPH method of Gissler et al. [2019] who considered strong two-way fluid-solid and solid-solid coupling to improve robustness. Since they also treat solid-solid interactions with SPH, this approach can be considered a penalty method. Penalty methods can be simple and efficient, but it can be difficult to tune parameters to obtain desirable simulation results. In a computational physics setting, Lu et al. [2017] considered contact-aware fluid-coupling, but their focus is a specialized problem of one-phase particulate Stokes suspensions in a voxelized domain under a boundary integral formulation. In contrast to these approaches, we focus on general viscous and inviscid grid-based liquid simulations with irregular surfaces and boundaries, based on a monolithic treatment of strong two-way fluid-solid coupling and LCP-based rigid body collision-handling.

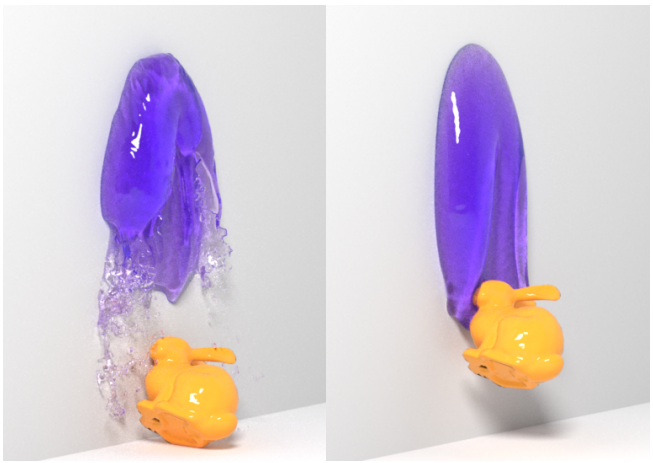

Fig. 4. A viscous liquid sphere on a wall dragged down by a colliding rigid-body bunny falling towards the ground, simulated with different pressure/viscosity coupling options (see §6.2). (Left) PC-VC-PC-unified solve (fluid-coupled contact, pressure decoupled from viscosity). (Right) Our proposed PVC-unified solve (strong coupling). The concluding pressure solve of the more typical PC-VC-PC-unified approach neglects viscosity and no-slip boundary conditions, leading to artificial melting and separation of the bunny from the liquid. Our PVC-unified approach accurately enforces boundaries and successfully captures and slows the descent of the bunny.

2.2 Monolithic and Multiphysics Coupling

Conceptually, we follow in a long tradition in physics-based animation that investigates the visual importance of careful and tight coupling of different forces and physical systems for stability, correctness, and realism. For example, in Eulerian fluids, Mullen et al. [2009] showed that coupling advection and pressure yields improved energy preservation, while Larionov et al. [2017] showed that tight pressure-viscosity coupling enables viscous rope coiling. In rigid body dynamics, Kaufman et al. [2008] emphasized the balance of normal and friction forces in delicate solid contact. In solid dynamics, Shinar et al. [2008] developed a unified treatment of two-way coupled rigid and deformable bodies, and Otaduy et al. [2009] proposed an efficient formulation for tight coupling of cloth and deformable dynamics with frictional contact. In a fabrication context, Chen et al. [2017] showed that correct finite element collision response hinges critically on awareness of an object's internal material model. As with these and many other prior studies, our contribution includes both identifying when and why strong coupling is necessary and proposing a method to support it.

3 BACKGROUND

In this section, we summarize the fundamental formulations and notation that we adopt for the dynamics of viscous fluids, rigid bodies, and their interactions. We build upon a particle-in-cell, pressure-viscosity-coupled liquid simulator [Larionov et al. 2017] (§3.1), a velocity-level implicit contact-handling mechanism for rigid bodies [Kaufman et al. 2008; Stewart 2000] (§3.2), and prior decoupled variational formulations for strong two-way coupling between fluid forces and isolated (i.e., not in mutual contact) rigid bodies (§3.3).

3.1 Viscous Fluid Dynamics Formulation

The incompressible Navier-Stokes equations for viscous fluids are

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla\mathbf{p} + \frac{1}{\rho}\nabla\cdot\boldsymbol{\tau} + \frac{1}{\rho}\mathbf{f}_v, \tag{1}$$

$$\boldsymbol{\tau} = 2\boldsymbol{\eta}\left(\frac{\nabla\mathbf{u} + (\nabla\mathbf{u})^T}{2}\right), \tag{2}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{3}$$

where t denotes time, $\frac{D}{Dt}$ material derivative, $\mathbf{u}$ (u, v, w) velocity, ρ density (which is constant in our simulation), $\mathbf{p}$ pressure, $\boldsymbol{\tau}(\tau_{xx}, \tau_{xy}, \tau_{xz}, \tau_{yy}, \tau_{yz}, \tau_{zz})$ symmetric viscous stress tensor, $\mathbf{f}_v$ external force per volume, and $\boldsymbol{\eta}$ dynamic viscosity. The dynamic viscosity can vary in space and time, though we consider only Newtonian fluids for now (i.e., the effective viscosity coefficient is independent of viscous stress), and extend this to support certain non-Newtonian fluids in §5. To advance the simulation step, we perform advection with the affine particle-in-cell (APIC) approach [Jiang et al. 2015], apply external forces, and then simultaneously handle the pressure and viscosity terms with a unified unsteady Stokes solver [Larionov et al. 2017]. Finally, we update the particles and perform position corrections on them to counteract minor volume drift due to numerical error [Takahashi and Lin 2019].

3.1.1 Unified Pressure-Viscosity Formulation. We address the pressure and viscosity terms in a unified and implicit way using

$$\frac{\mathbf{u}^{t+\Delta t} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla\mathbf{p}^{t+\Delta t} + \frac{1}{\rho}\nabla\cdot\boldsymbol{\tau}^{t+\Delta t}, \tag{4}$$

$$\boldsymbol{\tau}^{t+\Delta t} = 2\boldsymbol{\eta}\left(\frac{\nabla\mathbf{u}^{t+\Delta t} + \left(\nabla\mathbf{u}^{t+\Delta t}\right)^T}{2}\right), \tag{5}$$

$$\nabla\cdot\mathbf{u}^{t+\Delta t} = 0, \tag{6}$$

where $\mathbf{u}^*$ denotes the intermediate velocity after advection and external force steps, and Δt denotes the time step size. We can cast the implicitly integrated unsteady Stokes problem above into a minimization over the simulation domain Ω, similar to Larionov et al. [2017]. For natural solid boundary conditions, this is

$$\mathbf{p}, \boldsymbol{\tau} = \underset{\mathbf{p},\boldsymbol{\tau}}{\arg\min}\, \frac{1}{2}\int_\Omega \left(\rho\left\|\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla\mathbf{p} + \frac{\Delta t}{\rho}\nabla\cdot\boldsymbol{\tau}\right\|^2 + \frac{\Delta t}{2\boldsymbol{\eta}}\|\boldsymbol{\tau}\|_F^2\right) d\Omega, \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Intuitively, the first term minimizes the fluid's new kinetic energy, which on its own would rigidify the velocity; the second term penalizes large stresses, using $\boldsymbol{\eta}$ to balance between rigid and perfectly inviscid motion. The free surface case is similar, and they can be merged upon discretization.

3.1.2 Spatial Discretization. We adopt a variational finite difference approach [Batty et al. 2007; Batty and Bridson 2008; Larionov et al. 2017] , which approximates the integral by summing up the contributions of cell-sized control volumes weighted by their partial volumes. Given a signed distance function to represent fluid/solid domains, we can compute the fluid and solid volume fractions as diagonal matrices $\mathbf{W}_F^u$ and $\mathbf{W}_S^u$ (whose range is $[0, 1]$), respectively.

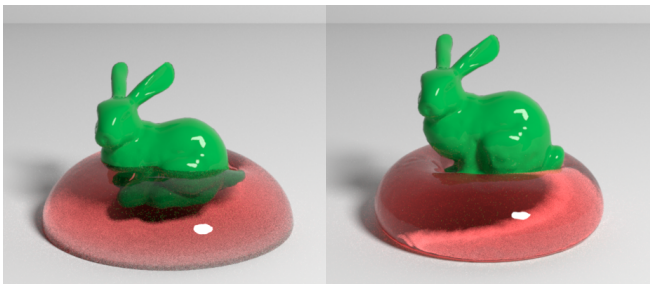

Fig. 5. A rigid bunny dropped onto a viscous sphere, simulated with different coupling options (see §6.2). (Left) PC-VC-PC-unified solve. (Right) Our PVC-unified solve. Due to the decoupled pressure and viscosity treatment, the viscous sphere simulated with PC-VC-PC-unified solve experiences artificial melting and inaccurate boundary behavior, causing tangential slip and faster sinking of the rigid bunny. With PVC-unified solve, the viscous sphere better supports the bunny and more realistically deforms under the load.

Here, the superscript and subscript indicate weight matrices for velocity variables u, and fluid domain F or solid domain S, respectively. Similarly, we compute weight matrices for viscous stress as $\mathbf{W}_F^s$ and $\mathbf{W}_S^s$. We can likewise decompose the simulation domain into the liquid and air domains, and compute corresponding volume fractions for pressure $\mathbf{W}_L^p$ and $\mathbf{W}_A^p$, velocity $\mathbf{W}_L^u$ and $\mathbf{W}_A^u$, and viscous stress $\mathbf{W}_L^s$ and $\mathbf{W}_A^s$ (for details, see the supplementary material, as well as the work of Larionov et al. [2017] and Takahashi and Lin [2019]).

With these weight matrices, we can spatially discretize (7) for solid boundaries and free surfaces separately and combine the resulting discrete formulations (per [Larionov et al. 2017]). We obtain the following unified minimization problem:

$$\begin{aligned}\boldsymbol{\tau}, \mathbf{p} &= \arg\min_{\boldsymbol{\tau},\mathbf{p}} E_f(\boldsymbol{\tau}, \mathbf{p}),\\ E_f(\boldsymbol{\tau}, \mathbf{p}) &= \frac{1}{2}\left(\left\|\mathbf{u}^* - \Delta t \mathbf{P}^{-1}(\mathbf{W}_L^u)^{-1}\left(\mathbf{G}\mathbf{W}_L^p\mathbf{p} + \mathbf{D}^T\mathbf{W}_L^s\boldsymbol{\tau}\right)\right\|_{\mathbf{M}_f}^2\right.\\ &\quad\left. + \frac{\Delta t}{2}\|\boldsymbol{\tau}\|_{\mathbf{N}^{-1}\mathbf{W}^s}^2\right),\end{aligned} \tag{8}$$

where $\mathbf{P}$ denotes a diagonal fluid density matrix, $\mathbf{G}$ a discrete gradient operator (for pressure), $\mathbf{D}$ a discrete deformation rate operator (for velocity, i.e., $\mathbf{D}\mathbf{u} \approx \frac{\nabla\mathbf{u}+(\nabla\mathbf{u})^T}{2}$), and $\mathbf{N}$ a dynamic viscosity coefficient matrix (which is SPD). We also use $\mathbf{M}_f = \mathbf{P}\mathbf{W}^u$, $\mathbf{W}^u = \mathbf{W}_F^u\mathbf{W}_L^u$, $\mathbf{W}^s = \mathbf{W}_F^s\mathbf{W}_L^s$, and matrix-weighted vector norm $\|\mathbf{y}\|_{\mathbf{W}} = \sqrt{\mathbf{y}^T\mathbf{W}\mathbf{y}} = \left\|\mathbf{W}^{\frac{1}{2}}\mathbf{y}\right\|_2$. This minimization problem for pressure and stress is convex and quadratic, so the optimality conditions yield a sparse SPD linear system that can be efficiently solved with e.g., preconditioned conjugate gradients (PCG). Importantly, this contrasts with a traditional Stokes problem, written in terms of pressure and velocity unknowns, which yields an indefinite system.

3.2 Rigid Body Dynamics Formulation

We consider a system of n rigid bodies with positional coordinates (position and quaternion) $\mathbf{x}$, velocity coordinates (linear and angular velocities) $\mathbf{v}$, and mass matrix $\mathbf{M}_r$. Collisions between rigid bodies must be addressed by satisfying several constraints, e.g., non-penetration. While some previous work has formulated such constraints at the position-level using DAEs [Ascher and Petzold 1998], the non-smooth nature of collision constraints requires specialized techniques that make efficiency challenging [Anitescu and Hart 2004]. It is also preferable to hold positions and orientations of bodies fixed during collision resolution, since re-evaluation of the fluid volume fraction weights would otherwise be required at each iteration of our coupled solver. Therefore, we address rigid body collisions at the velocity-level.

To model collisions between rigid bodies, we consider impulses along the normal and tangential directions in an implicit manner following Stewart [2000]. However, unlike the traditional approach, we model impulses as forces so that they can be consistently integrated with pressure and viscosity for fluids, as detailed in §4.

3.2.1 Normal Contact Force. Given a collision constraint vector $\boldsymbol{\phi}(\mathbf{x}) \geq 0$, its Jacobian $\mathbf{J}_\lambda$, normal contact forces $\boldsymbol{\lambda}$, and a diagonal matrix of coefficients of restitution $\mathbf{R}$ ($0 \leq \mathbf{R}_i \leq 1$ with solid index i), we model elastic impact as an LCP according to the Signorini condition; this dictates nonnegative contact forces and separating relative velocities between contacting rigid bodies,

$$0 \leq \boldsymbol{\lambda}^{t+\Delta t} \perp \mathbf{J}_\lambda\mathbf{v}^{t+\Delta t} \geq -\mathbf{J}_\lambda\mathbf{R}\mathbf{v}^*, \tag{9}$$

where $\mathbf{v}^*$ denotes the intermediate solid velocity after external forces are applied. The contact forces can be applied to update the velocity of the rigid bodies to be non-penetrating using

$$\mathbf{v}^{t+\Delta t} = \mathbf{v}^* + \Delta t\mathbf{M}_r^{-1}\mathbf{J}_\lambda^T\boldsymbol{\lambda}^{t+\Delta t}. \tag{10}$$

To compute the unknown contact forces $\boldsymbol{\lambda}$, we can combine these formulations, and arrive at the system

$$\mathbf{J}_\lambda\mathbf{M}_r^{-1}\mathbf{J}_\lambda^T\boldsymbol{\lambda}^{t+\Delta t} = -\mathbf{J}_\lambda\mathbf{C}\mathbf{v}^* \text{ s.t. } 0 \leq \boldsymbol{\lambda}^{t+\Delta t}, \tag{11}$$

where $\mathbf{C} = (\mathbf{I} + \mathbf{R})$. Using kinetic energy minimization, this can be reformulated as a QP by rearranging using $\mathbf{C}$ while taking the box constraints into account:

$$\boldsymbol{\lambda} = \arg\min_{0\leq\boldsymbol{\lambda}} \frac{1}{2}\left\|\mathbf{C}\mathbf{v}^* + \Delta t\mathbf{M}_r^{-1}\mathbf{J}_\lambda^T\boldsymbol{\lambda}\right\|_{\mathbf{M}_r}^2. \tag{12}$$

While minimizing the kinetic energy with normal contact forces leads to perfectly inelastic collisions, scaling the intermediate solid velocity $\mathbf{v}^*$ with $\mathbf{C}$ decreases the relative weight of the contact force in the minimization, accounting for inelastic and elastic collisions (with spatially varying coefficients of restitution e.g., [Wang et al. 2017]). The system matrix in (11) is SPD (but not an M-matrix nor diagonally dominant in general) with box constraints, so general convex QP solvers can be applied.

3.2.2 Tangential friction Force. Given the friction force $\mathbf{z}$ and the Jacobian of the friction constraint $\mathbf{J}_z$, we can rewrite the velocity update for rigid bodies as

$$\mathbf{v}^{t+\Delta t} = \mathbf{v}^* + \Delta t\mathbf{M}_r^{-1}(\mathbf{J}_\lambda^T\boldsymbol{\lambda}^{t+\Delta t} + \mathbf{J}_z^T\mathbf{z}^{t+\Delta t}). \tag{13}$$

Using the commonly used symmetric pyramid approximation of the Coulomb friction cone, the friction force can be computed according to the maximal dissipation theorem as

$$\mathbf{z} = \arg\min_{-\text{diag}(\boldsymbol{\mu})\boldsymbol{\lambda}\leq\mathbf{z}\leq\text{diag}(\boldsymbol{\mu})\boldsymbol{\lambda}} \mathbf{z}^T\mathbf{J}_z\left(\mathbf{C}\mathbf{v}^* + \Delta t\mathbf{M}_r^{-1}(\mathbf{J}_\lambda^T\boldsymbol{\lambda} + \mathbf{J}_z^T\mathbf{z})\right), \tag{14}$$

where $\boldsymbol{\mu}$ denotes friction coefficients. Similar to the minimization problem (12) for $\boldsymbol{\lambda}$, this minimization is also quadratic with respect to the friction force $\mathbf{z}$ being under the box constraints. The system can likewise be made SPD by separating the box constraints and can be solved with a general convex QP solver.

3.2.3 Unified Handling. Unless there is no friction at all (i.e., $\boldsymbol{\mu} = 0$), these contact and friction forces must be coupled since they naturally depend on each other [Kaufman et al. 2008], i.e., if contact forces are changed, it is necessary to update friction forces, and vice versa. As such, a unified treatment is typically necessary for robust coupling and can be formulated as

$$\boldsymbol{\lambda}, \mathbf{z} = \underset{\substack{0 \le \boldsymbol{\lambda}, \\ -\mathrm{diag}(\boldsymbol{\mu})\boldsymbol{\lambda} \le \mathbf{z} \le \mathrm{diag}(\boldsymbol{\mu})\boldsymbol{\lambda}}}{\arg\min} \frac{1}{2} \left\| \mathbf{C}\mathbf{v}^* + \Delta t \mathbf{M}_r^{-1} (\mathbf{J}_\lambda^T \boldsymbol{\lambda} + \mathbf{J}_z^T \mathbf{z}) \right\|_{\mathbf{M}_r}^2. \tag{15}$$

In the rigid body literature, this unified formulation has been addressed as a single non-symmetric LCP. A classical approach is Lemke's method (based on a direct solver) [Lloyd 2005; Tan et al. 2012] although it is undesirable for larger problems by the nature of direct solvers. To avoid forming the non-symmetric LCP, it is necessary to separate out the friction constraint; one typically decomposes the unified problem to iteratively and alternatingly address contact and friction. For example, Kaufman et al. [2008] alternated solving separate symmetric LCPs for normal and friction forces, known as staggered projections, while Tonge et al. [2012] solved contact and friction problems in each iteration of stationary iterative solvers, such as projected Gauss-Seidel (PGS) or projected Jacobi.

3.2.4 Stabilization. While in theory collision constraints enforce non-penetration between rigid bodies, in practice penetrations can occur due to numerical error and discrete collision handling. To address this issue, stabilization techniques are commonly employed, e.g., Baumgarte stabilization [Baumgarte 1972] and post-stabilization [Anitescu and Hart 2004; Cline and Pai 2003]. Inspired by DAE formulations, they incorporate the degree of constraint violation, expressed by $\boldsymbol{\phi}(\mathbf{x})$, to increase the contact force and thereby correct overlaps. We modify the contact force computation, (11), to become

$$\mathbf{J}_\lambda \mathbf{M}_r^{-1} \mathbf{J}_\lambda^T \boldsymbol{\lambda}^{t+\Delta t} = -\left(\mathbf{J}_\lambda \mathbf{C}\mathbf{v}^* + \beta \frac{\min(\boldsymbol{\phi}(\mathbf{x}), 0)}{\Delta t} \right) \text{ s.t. } 0 \le \boldsymbol{\lambda}^{t+\Delta t}, \tag{16}$$

where β denotes a tunable parameter to control the influence of the constraint violation. To prevent frequent switching on and off of contacts due to the imprecision of floating point arithmetic, we also use a small penetration margin ϵ (= $0.1\Delta x$ in our examples, where Δx denotes grid cell width) setting $\boldsymbol{\phi}(\mathbf{x}) \leftarrow \boldsymbol{\phi}(\mathbf{x}) + \epsilon$.

3.3 Fluid-Solid Two-Way Coupling

Strong two-way coupling of fluids was previously achieved by separately coupling fluid pressure [Batty et al. 2007] and viscosity [Takahashi and Lin 2019] to rigid bodies in distinct stages. We rely on the same mechanisms for the exchange of forces, but will later show that it is often critical to integrate these simultaneously.

3.3.1 Pressure. Considering pressure forces from fluids to rigid bodies, the velocity update for the rigid bodies, taking volume fractions into account, can be computed by

$$\mathbf{v}^{t+\Delta t} = \mathbf{v}^* + \Delta t \mathbf{M}_r^{-1} \mathbf{J}_p \mathbf{W}_L^p \mathbf{p}^{t+\Delta t}, \tag{17}$$

where $\mathbf{J}_p$ denotes a linear function which integrates the pressures over the surface of the rigid body to determine the net pressure forces applied to it [Bridson 2015]. (We discuss the construction of $\mathbf{J}_p$, which depends on the solid weights $\mathbf{W}_S^u$, in §4.4.1.) Consistent physical dimensions for fluids and rigid bodies can be implicitly achieved by internally scaling the rigid body mass [Batty et al. 2007; Bridson 2015], and one can then formulate a minimization problem for the pressure needed to update both the fluid velocity and the rigid body velocity to enforce fluid incompressibility:

$$\mathbf{p} = \underset{\mathbf{p}}{\arg\min} \frac{1}{2} \left(\left\| \mathbf{u}^* - \Delta t \mathbf{P}^{-1} (\mathbf{W}_L^u)^{-1} \mathbf{G} \mathbf{W}_L^p \mathbf{p} \right\|_{\mathbf{M}_f}^2 + \left\| \mathbf{v}^* + \Delta t \mathbf{M}_r{}^{-1} \mathbf{J}_p \mathbf{W}_L^p \mathbf{p} \right\|_{\mathbf{M}_r}^2 \right). \tag{18}$$

This approach naturally enforces *free-slip* boundary conditions at the fluid-solid interface. This energy is quadratic with respect to pressure, leading to a single SPD linear solve that is mostly sparse, except for dense submatrices due to the added solid components [Batty et al. 2007; Bridson 2015; Robinson-Mosher et al. 2008].

3.3.2 Viscosity. Similar to pressure forces, given viscous forces from fluids to rigid bodies, the rigid body velocity update is

$$\mathbf{v}^{t+\Delta t} = \mathbf{v}^* + \Delta t \mathbf{M}_r^{-1} \mathbf{J}_s \mathbf{W}_L^s \boldsymbol{\tau}^{t+\Delta t}, \tag{19}$$

where $\mathbf{J}_s$ denotes a linear function, dependent on $\mathbf{W}_S^u$, which integrates viscous stress over the surface of the rigid body to determine the net viscous forces on it (details in §4.4.1). Combining this velocity update with implicit integration for the fluid's viscosity yields

$$\boldsymbol{\tau} = \underset{\boldsymbol{\tau}}{\arg\min} \frac{1}{2} \left(\left\| \mathbf{u}^* - \Delta t \mathbf{P}^{-1} (\mathbf{W}_L^u)^{-1} \mathbf{D}^T \mathbf{W}_L^s \boldsymbol{\tau} \right\|_{\mathbf{M}_f}^2 + \frac{\Delta t}{2} \left\| \boldsymbol{\tau} \right\|_{\mathbf{N}^{-1}\mathbf{W}^s}^2 + \left\| \mathbf{v}^t + \Delta t \mathbf{M}_r{}^{-1} \mathbf{J}_s \mathbf{W}_L^s \boldsymbol{\tau} \right\|_{\mathbf{M}_r}^2 \right). \tag{20}$$

This minimization enforces *no-slip* boundary conditions at the fluid-solid interface. It is also quadratic with respect to viscous stresses and, like pressure, requires solving one SPD linear system. Notably, we differ from Takahashi and Lin [2019] in preferring stress unknowns over velocity unknowns, for compatibility with (8) and to ensure our eventual unified variational form is a minimization.

4 MONOLITH SOLVER

The two-way coupling formulations above effectively reproduce many interactions between fluids and rigid bodies while handling the collisions between rigid bodies in a separate process [Batty et al. 2007; Takahashi and Lin 2019]. This decoupled approach works well if few or no collisions between rigid bodies are detected, but artifacts quickly arise in the presence of potential contacts. Because the preceding formulations are entirely oblivious to contacts *between* rigid bodies themselves, the applied fluid forces will necessarily be incorrect. Conversely, the minimization problem(s) for inter-object contact handling are unaware of the existence of fluids, and thus the collision solver applies contact and friction forces to rigid bodies without feeling any resistance from fluids. The lack of mutual

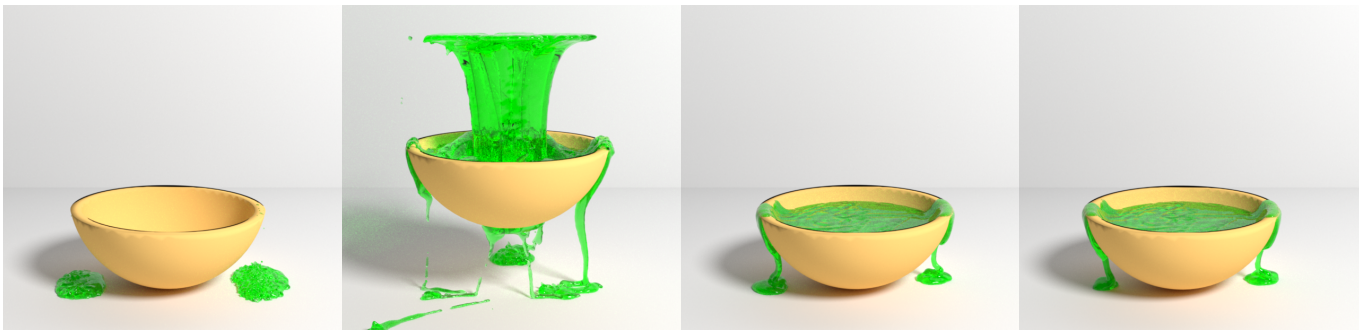

Fig. 6. A viscous liquid dragon dropped onto a bowl on the ground, with different coupling options (see §6.3). (Left) PV-C solve. (Middle left) C-PV solve. (Middle right) PVC-iterative solve. (Right) Our PVC-unified solve. While PV-C solve rapidly loses liquid volume due to violation of the incompressibility constraint, C-PV solve suffers from stability issues that cause the bowl to jump suddenly, due to neglecting contacts between the bowl and ground. While both PVC-iterative solve and PVC-unified solve generate plausible results, PVC-unified solve is approximately 7× faster than PVC-iterative solve.

awareness between fluid forces and contact forces causes serious visual artifacts, such as significant fluid volume loss and neglected contacts between rigid bodies.

Furthermore, prior strong two-way fluid-solid coupling schemes also addressed pressure and viscosity independently from one another, as noted in §3.3. This decoupling causes issues of its own, such as artificial fluid melting and incorrect boundary conditions, since the optimal balance of pressure and viscous stress is not achieved. To address all of these issues, it is essential to solve the dynamics in a fully monolithic way.

4.1 Monolithic Formulation

We aim to merge the individual minimizations for pressure, viscosity and frictional contact into a single variational problem. Considering all relevant forces and explicitly enforcing dimensional consistency between fluids and rigid bodies by scaling the rigid body mass, normal contact force, and friction force with $\alpha = 1/(\Delta x^3)$, the rigid body velocity update becomes

$$\mathbf{v}^{t+\Delta t} = \mathbf{v}^* + \Delta t(\alpha \mathbf{M}_r)^{-1}\left(\mathbf{J}_\lambda{}^T \alpha \boldsymbol{\lambda}^{t+\Delta t} + \mathbf{J}_z{}^T \alpha \mathbf{z}^{t+\Delta t} + \mathbf{J}_p \mathbf{W}_L^p \mathbf{p}^{t+\Delta t} + \mathbf{J}_s \mathbf{W}_L^s \boldsymbol{\tau}^{t+\Delta t}\right). \tag{21}$$

This combined rigid body update can be coupled with the Stokes problem (8) for fluid pressure and viscosity using its variational formulation [Batty et al. 2007; Larionov et al. 2017; Takahashi and Lin 2019]. Our careful choice of individual formulations (and consistent use of force-based unknowns) enables us to easily combine them to obtain a single monolithic constrained minimization problem

$$\boldsymbol{\tau}, \mathbf{p}, \boldsymbol{\lambda}, \mathbf{z} = \underset{\substack{\boldsymbol{\tau}, \mathbf{p}, 0 \le \boldsymbol{\lambda}, \\ -\mathrm{diag}(\boldsymbol{\mu})\boldsymbol{\lambda} \le \mathbf{z} \le \mathrm{diag}(\boldsymbol{\mu})\boldsymbol{\lambda}}}{\arg\min} E_f(\boldsymbol{\tau}, \mathbf{p}) + E_r(\boldsymbol{\tau}, \mathbf{p}, \boldsymbol{\lambda}, \mathbf{z}), \tag{22}$$

where the rigid body term $E_r(\boldsymbol{\tau}, \mathbf{p}, \boldsymbol{\lambda}, \mathbf{z})$ is defined by

$$E_r(\boldsymbol{\tau}, \mathbf{p}, \boldsymbol{\lambda}, \mathbf{z}) = \frac{1}{2}\left\| \mathbf{C}\mathbf{v}^* + \Delta t(\alpha \mathbf{M}_r)^{-1}\left(\mathbf{J}_\lambda{}^T \alpha \boldsymbol{\lambda} + \mathbf{J}_z{}^T \alpha \mathbf{z} + \mathbf{J}_p \mathbf{W}_L^p \mathbf{p} + \mathbf{J}_s \mathbf{W}_L^s \boldsymbol{\tau}\right)\right\|_{\alpha \mathbf{M}_r}^2, \tag{23}$$

and the fluid term $E_f(\boldsymbol{\tau}, \mathbf{p})$ is still defined as in (8). This minimization reduces to the Stokes problem (8) without rigid bodies present, and to the contact problem (15) without fluids present. Our formulation (22) is *monolithic* in the sense that it is a single unified problem; various optimization approaches could be applied to solve (22) without invalidating this property. In the following, we present our approach for efficiently addressing this monolithic problem.

4.2 Minimization

The unified objective function is quadratic with respect to $\boldsymbol{\tau}, \mathbf{p}, \boldsymbol{\lambda}$, and $\mathbf{z}$, and thus the optimality conditions yield the following SPD system with frictional contact constraints:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & \mathbf{A}_{14} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} & \mathbf{A}_{23} & \mathbf{A}_{24} \\ \mathbf{A}_{13}^T & \mathbf{A}_{23}^T & \mathbf{A}_{33} & \mathbf{A}_{34} \\ \mathbf{A}_{14}^T & \mathbf{A}_{24}^T & \mathbf{A}_{34}^T & \mathbf{A}_{44} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{p} \\ \boldsymbol{\lambda} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \end{bmatrix}, \quad \text{s.t. } 0 \le \boldsymbol{\lambda}, -\mathrm{diag}(\boldsymbol{\mu})\boldsymbol{\lambda} \le \mathbf{z} \le \mathrm{diag}(\boldsymbol{\mu})\boldsymbol{\lambda}, \tag{24}$$

where

$$\begin{aligned}
\mathbf{A}_{11} &= \frac{1}{2}\mathbf{N}^{-1}\mathbf{W}_F^s\mathbf{W}_L^s + \Delta t \mathbf{W}_L^s \mathbf{D}\mathbf{W}_F^u \mathbf{P}^{-1}(\mathbf{W}_L^u)^{-1}\mathbf{D}^T\mathbf{W}_L^s \\
&\quad + \frac{\Delta t}{\alpha}\mathbf{W}_L^s\mathbf{J}_s^T\mathbf{M}_r^{-1}\mathbf{J}_s\mathbf{W}_L^s, \\
\mathbf{A}_{12} &= \Delta t \mathbf{W}_L^s \mathbf{D}\mathbf{W}_F^u \mathbf{P}^{-1}(\mathbf{W}_L^u)^{-1}\mathbf{G}\mathbf{W}_L^p + \frac{\Delta t}{\alpha}\mathbf{W}_L^s\mathbf{J}_s^T\mathbf{M}_r^{-1}\mathbf{J}_p\mathbf{W}_L^p, \\
\mathbf{A}_{13} &= \Delta t \mathbf{W}_L^s \mathbf{J}_s^T \mathbf{M}_r^{-1}\mathbf{J}_\lambda^T, \quad \mathbf{A}_{14} = \Delta t \mathbf{W}_L^s \mathbf{J}_s^T \mathbf{M}_r^{-1}\mathbf{J}_z^T, \\
\mathbf{A}_{22} &= \Delta t \mathbf{W}_L^p \mathbf{G}^T \mathbf{W}_F^u \mathbf{P}^{-1}(\mathbf{W}_L^u)^{-1}\mathbf{G}\mathbf{W}_L^p + \frac{\Delta t}{\alpha}\mathbf{W}_L^p\mathbf{J}_p^T\mathbf{M}_r^{-1}\mathbf{J}_p\mathbf{W}_L^p, \\
\mathbf{A}_{23} &= \Delta t \mathbf{W}_L^p \mathbf{J}_p^T \mathbf{M}_r^{-1}\mathbf{J}_\lambda^T, \quad \mathbf{A}_{24} = \Delta t \mathbf{W}_L^p \mathbf{J}_p^T \mathbf{M}_r^{-1}\mathbf{J}_z^T, \\
\mathbf{A}_{33} &= \Delta t \alpha \mathbf{J}_\lambda \mathbf{M}_r^{-1}\mathbf{J}_\lambda^T, \quad \mathbf{A}_{34} = \Delta t \alpha \mathbf{J}_\lambda \mathbf{M}_r^{-1}\mathbf{J}_z^T, \\
\mathbf{A}_{44} &= \Delta t \alpha \mathbf{J}_z \mathbf{M}_r^{-1}\mathbf{J}_z^T, \\
\mathbf{b}_1 &= \mathbf{W}_L^s \mathbf{D}\mathbf{W}_F^u \mathbf{u}^* - \mathbf{W}_L^s\mathbf{J}_s^T\mathbf{v}^*, \quad \mathbf{b}_2 = \mathbf{W}_L^p\mathbf{G}^T\mathbf{W}_F^u\mathbf{u}^* - \mathbf{W}_L^p\mathbf{J}_p^T\mathbf{v}^*, \\
\mathbf{b}_3 &= -\alpha\left(\mathbf{J}_\lambda\mathbf{C}\mathbf{v}^* + \beta\frac{\min(\boldsymbol{\phi}(\mathbf{x}), 0)}{\Delta t}\right), \quad \mathbf{b}_4 = -\alpha\mathbf{J}_z\mathbf{v}^*.
\end{aligned}$$

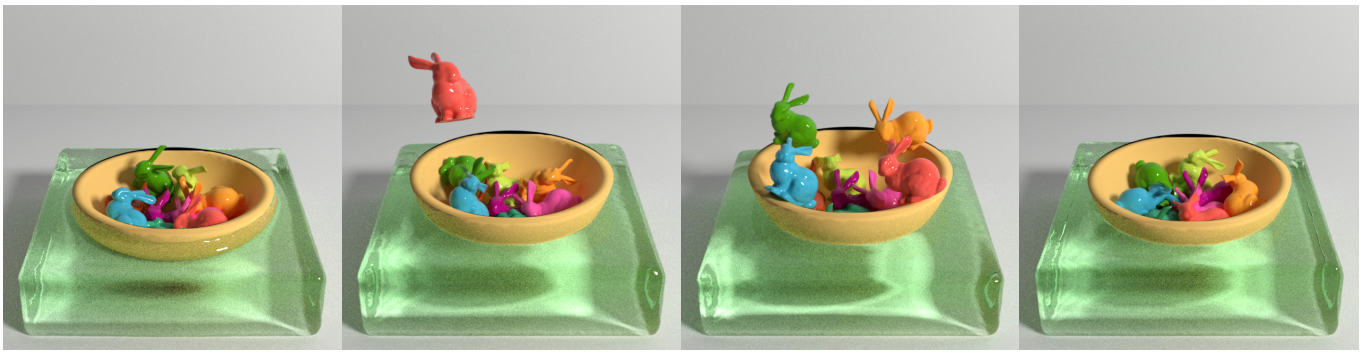

Fig. 7. Multiple rigid bunnies dropped into a bowl floating on viscous liquid, with different coupling options (see §6.3). (Left) PV-C solve at frame 147. (Middle left) C-PV solve at frame 54. (Middle right) PVC-iterative solve at frame 17. (Right) PVC-unified solve at frame 147. Due to the neglected incompressibility constraint and implicit viscosity integration, PV-C solve causes slight volume loss for the viscous liquid and unnatural fluid crawling behavior at the surface of the bowl. C-PV solve causes sudden and intense impulses, due to the neglected contact handling, which drives the orange bunny to jump out of the bowl. PVC-iterative solve with early termination fails to appropriately handle incompressiblity, implicit viscosity integration, and/or contact handling, leading to an explosion of the simulation immediately after this frame. PVC-unified solve generates plausible results with a reasonable cost comparable to PV-C and C-PV.

Since $\mathbf{C}$ mediates elastic collision response, it appears only in $\mathbf{b}_3$, so as to avoid repulsive pressure, stress, or friction forces. Similar to (16), we have adopted the stabilization technique by modifying $\mathbf{b}_3$.

Our formulation avoids explicitly forming $(\alpha\mathbf{M}_r)^{-1}$ since this can produce matrix entries that are too small to handle with numerical solvers, e.g., conjugate gradient (CG). In our experiments, solvers using double-precision floating-point almost always failed to converge with explicit formation of $(\alpha\mathbf{M}_r)^{-1}$; our formulation avoids these issues, enabling the solution of numerically difficult systems, even with large density/mass ratios for fluids and rigid bodies.

As before, rigid body contact and friction forces depend on each other. We therefore use a staggered projections approach [Kaufman et al. 2008] to address the friction force $\mathbf{z}$ separately from the other variables ($\mathbf{p}$, $\boldsymbol{\tau}$, and $\boldsymbol{\lambda}$). This splitting is preferable for two reasons: first, it makes the system easier to solve since $\mathbf{z}$, which depends on $\boldsymbol{\lambda}$, is decoupled from the computation of $\boldsymbol{\lambda}$, and second, contact forces are often more tightly coupled to fluid pressure and viscous stresses than are the friction forces.

We arrive at the following approach: we first solve an LCP for $\mathbf{z}$ with iteration index k,

$$\begin{aligned} \mathbf{A}_{44}\mathbf{z}^{k+1} &= \mathbf{b}_4 - \mathbf{A}_{14}^T\boldsymbol{\tau}^k - \mathbf{A}_{24}^T\mathbf{p}^k - \mathbf{A}_{34}^T\boldsymbol{\lambda}^k \\ \text{s.t. } -\operatorname{diag}(\boldsymbol{\mu})\boldsymbol{\lambda}^k &\le \mathbf{z}^{k+1} \le \operatorname{diag}(\boldsymbol{\mu})\boldsymbol{\lambda}^k, \end{aligned} \tag{25}$$

and then a Mixed LCP (MLCP) for $\boldsymbol{\tau}$, $\mathbf{p}$, and $\boldsymbol{\lambda}$:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{13}^T & \mathbf{A}_{23}^T & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}^{k+1} \\ \mathbf{p}^{k+1} \\ \boldsymbol{\lambda}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 - \mathbf{A}_{14}\mathbf{z}^{k+1} \\ \mathbf{b}_2 - \mathbf{A}_{24}\mathbf{z}^{k+1} \\ \mathbf{b}_3 - \mathbf{A}_{34}\mathbf{z}^{k+1} \end{bmatrix}, \quad \text{s.t. } 0 \le \boldsymbol{\lambda}^{k+1}. \tag{26}$$

We iteratively and alternately solve these systems, choosing specifically to solve $\mathbf{z}$ first and then $\boldsymbol{\tau}$, $\mathbf{p}$, and $\boldsymbol{\lambda}$. Since the latter are generally more important for both fluid and rigid body dynamics, we prefer to terminate the iteration with optimal $\boldsymbol{\tau}$, $\mathbf{p}$, and $\boldsymbol{\lambda}$. When the effect of friction is weak, fewer staggered iterations are required and when $\boldsymbol{\mu} = 0$, solving a single MLCP suffices.

4.3 LCP Solver

In each of the staggered iterations, we need to solve an LCP for the friction force and an MLCP for pressure, stress, and normal forces; the latter can be treated as an LCP by simply setting negative and positive infinity as box constraints for the pressure and stress variables. Various approaches have been used in the past to solve LCPs, e.g., PATH solver [Ferris and Munson 2000], QL solver [Schittkowski 2005], and PGS [Catto 2020; Coumans 2020; Smith 2008]. These approaches are known to be slow, although in the context of rigid body contact problems PGS has been extensively used because of its simplicity and flexibility [Catto 2020; Coumans 2020; Smith 2008; Tonge et al. 2012]. One key difference from previous rigid body contact work is that our LCP includes pressure and stress variables which yield a very large number of degrees of freedom (DOFs), making it preferable to employ sparse iterative solvers.

Fortunately, these LCPs are SPD, and can be reformulated as convex QPs [Boyd and Vandenberghe 2004], as suggested by the minimization problems giving rise to them. In particular, our QPs are sparse and have only box constraints, which makes it possible to use efficient specialized solvers, e.g., Modified Proportioning with Reduced Gradient Projections (MPRGP) [Dostal and Schoberl 2005; Dostl 2009]. We therefore employ MPRGP to solve both of the LCPs.

MPRGP is a variant of CG with gradient projections for box constraints and, like CG, the convergence of MPRGP can be accelerated with a preconditioner. While previous work in fluid animation used Modified Incomplete Cholesky (MIC) [Gerszewski and Bargteil 2013; Narain et al. 2010], we employ a Successive-Over-Relaxation (SOR) preconditioner, since we found it to be more effective for our system with viscosity, both in terms of computational cost per iteration (even with a single thread) and the total number of iterations. Our observations are consistent with those of Yun and Han [2002], who reported deterioration of MIC preconditioning, and, more generally, it is known that MIC is less effective for non-M-matrices, whose inverse cannot be well-approximated by the MIC decomposition.

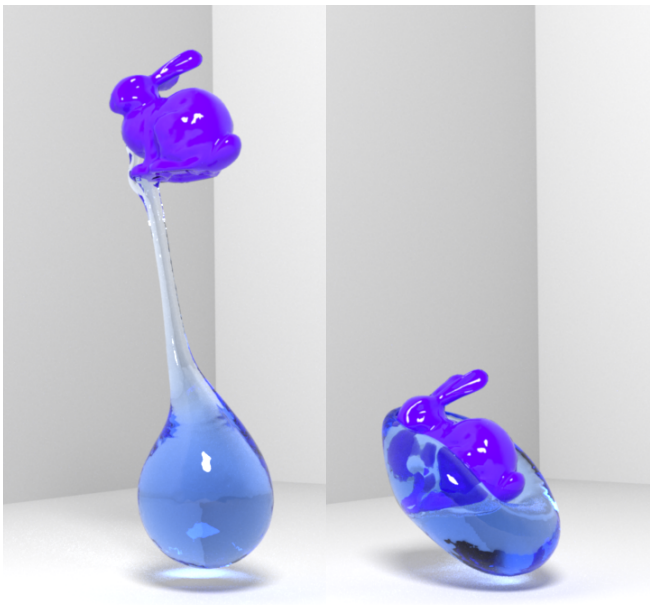

Fig. 8. A viscous sphere and a rigid bunny collide in the air. (Left) With an inconsistent $\mathbf{J}_s$ computation. (Right) With our consistent definition of $\mathbf{J}_s$. The inconsistent $\mathbf{J}_s$ leads to spurious non-zero values in the right hand side of the system even with uniform velocities over fluids and rigid bodies, which incorrectly freezes the bunny in mid-air. Our consistent formulation ensures the values in the right hand side vanish by construction for uniform velocities of fluids and rigid bodies and generates artifact-free motions.

To perform SOR preconditioning in parallel, we use a coloring scheme. An obvious approach to coloring DOFs would be to decompose the domain with multiple blocks, as done for deformable objects by Zhu et al. [2010]. However, it is difficult to ensure a consistently even decomposition of the space because the liquids and rigid bodies are in constant motion from frame to frame. Hence, we use pointwise red-black coloring for pressure and each component of the viscous stress, and color the contacts between rigid bodies based on the Welsh-Powell algorithm [Welsh and Powell 1967].

4.4 Implementation Details

4.4.1 Consistent force exchange. To ensure a precise exchange of forces between fluids and rigid bodies, it is essential to enforce consistency of the force integration. That is, since $\mathbf{J}_p$ and $\mathbf{J}_s$ (which depend on $\mathbf{W}_S^u$) apply pressure and stress on rigid bodies, they must be defined in a manner consistent with the $\mathbf{G}$ and $\mathbf{D}$ operators, respectively, that apply pressure and stress to the fluid itself. We guarantee this as follows. Given a pressure field, the discrete update it applies to a regular fluid body is $-\mathbf{W}_F^u\mathbf{G}\mathbf{W}_L^p\mathbf{p}$. If we instead consider a rigid body of the same shape to be updated by the same pressure field, the net effect should be equivalent to aggregating this applied pressure force within the body's volume into a rigid body force (and torque). We express this with an aggregation matrix $\mathbf{Q}$ to get $-\mathbf{Q}\mathbf{W}_F^u\mathbf{G}\mathbf{W}_L^p\mathbf{p}$. We observe that, for consistency, this effective body force on the rigid body must be equal to our desired expression for the pressure integrated over the *surface* of the rigid body $\mathbf{J}_p\mathbf{W}_L^p\mathbf{p}$. We can therefore define $\mathbf{J}_p$ as $\mathbf{J}_p = -\mathbf{Q}\mathbf{W}_S^u\mathbf{G}$ (where we replaced the fluid volumes $\mathbf{W}_F^u$ with rigid body volumes $\mathbf{W}_S^u$). This is also consistent with the continuous surface-to-volume integral transformation used by Batty et al. [2007]. Similarly, viscous stress forces applied to fluids ($-\mathbf{W}_F^u\mathbf{D}^T\mathbf{W}_L^s\boldsymbol{\tau}$) can be aggregated as $-\mathbf{Q}\mathbf{W}_F^u\mathbf{D}^T\mathbf{W}_L^s\boldsymbol{\tau}$. Since this needs to be equal to $\mathbf{J}_s\mathbf{W}_L^s\boldsymbol{\tau}$, we obtain $\mathbf{J}_s = -\mathbf{Q}\mathbf{W}_S^u\mathbf{D}^T$.

While $\mathbf{J}_p$ and $\mathbf{J}_s$ can be assembled with the definition above, we can significantly reduce the number of entries in $\mathbf{J}_p$ and $\mathbf{J}_s$ since forces inside of rigid bodies cancel each other out, and only nonzeros over the surface of rigid bodies are meaningful [Batty et al. 2007]. In practice, we directly assemble $\mathbf{J}_p$ and $\mathbf{J}_s$, similar to Bridson [2015] and Takahashi and Lin [2019] without explicitly forming $\mathbf{Q}$. We note that $\mathbf{J}_p$ and $\mathbf{J}_s$ defined in this way can completely cancel $\mathbf{b}_1$ and $\mathbf{b}_2$ for uniform velocity fields over fluids and rigid bodies, e.g., in freefall scenes. Figure 8 compares our definition with that of Takahashi and Lin [2019].

4.4.2 Stress reduction. Similar to Larionov et al. [2017], we reduce the number of viscous stress variables using the fact that the divergence-free constraint renders the stress tensor traceless. In 2D, we have $\tau_{xx} + \tau_{yy} = 0$, so we eliminate τ_{yy} by substituting $\tau_{yy} = -\tau_{xx}$ for $\mathbf{D}$ and $\mathbf{J}_s$, and treat $\tau_{xx} = \eta\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right)$ as the unknown variable in the system. Similarly, we have $\tau_{xx} + \tau_{yy} + \tau_{zz} = 0$ in 3D, and eliminate τ_{zz} by substituting $\tau_{zz} = -(\tau_{xx} + \tau_{yy})$. Then, we treat τ_{xx} and τ_{yy} as the unknown variables with the constraints of $\tau_{xx} + \frac{1}{2}\tau_{yy} = \eta\left(\frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}\right)$ and $\frac{1}{2}\tau_{xx} + \tau_{yy} = \eta\left(\frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right)$, which can be naturally enforced by introducing an auxiliary matrix $\mathbf{H}$ as $\frac{1}{2}\mathbf{H}\boldsymbol{\tau} = \mathbf{N}\mathbf{D}\mathbf{u}$. In practice, we merge $\mathbf{H}$ into $\mathbf{N}$ for simplicity (i.e., $\mathbf{N} \leftarrow \mathbf{H}^{-1}\mathbf{N}$), modifying the structure of $\mathbf{N}$ from diagonal (without the stress reduction) to block-diagonal.

4.4.3 Sequential multiplication. For strong two-way coupling of rigid bodies with fluid pressure and viscous stress, it is known that dense submatrices can be formed [Batty et al. 2007; Bridson 2015; Robinson-Mosher et al. 2008; Takahashi and Lin 2019] (e.g., $\mathbf{W}_L^s\mathbf{J}_s^T\mathbf{M}_r^{-1}\mathbf{J}_s\mathbf{W}_L^s$ in $\mathbf{A}_{11}$). The size of the dense submatrices also scales poorly, making it worse at higher resolutions [Robinson-Mosher et al. 2008]. While forming these dense submatrices explicitly does make it possible to apply preconditioning to the entire system, leading to fewer total iterations, the computational cost per solver iteration increases significantly. However, iterative solvers like CG and MPRGP require only the result of matrix-vector multiplication, and not the matrix itself; we exploit this fact, noting that the contributions of the dense blocks can be more efficiently computed via multiple matrix-vector multiplications [Bridson 2015]. For example, to compute $\mathbf{A}\mathbf{x} = \mathbf{y}$, where $\mathbf{A} = \mathbf{W}_L^s\mathbf{J}_s^T\mathbf{M}_r^{-1}\mathbf{J}_s\mathbf{W}_L^s$, we can compute $\mathbf{y} = (\mathbf{W}_L^s\mathbf{J}_s^T(\mathbf{M}_r^{-1}(\mathbf{J}_s\mathbf{W}_L^s\mathbf{x})))$ without forming the dense matrices. We use this technique to handle $\frac{\Delta t}{\alpha}\mathbf{W}_L^s\mathbf{J}_s^T\mathbf{M}_r^{-1}\mathbf{J}_s\mathbf{W}_L^s$ in $\mathbf{A}_{11}$, $\frac{\Delta t}{\alpha}\mathbf{W}_L^s\mathbf{J}_s^T\mathbf{M}_r^{-1}\mathbf{J}_p\mathbf{W}_L^p$ in $\mathbf{A}_{12}$, and $\frac{\Delta t}{\alpha}\mathbf{W}_L^p\mathbf{J}_p^T\mathbf{M}_r^{-1}\mathbf{J}_p\mathbf{W}_L^p$ in $\mathbf{A}_{22}$. We found that this approach, with SOR preconditioning applied only on the remaining sparse part of the matrix, is generally faster (ranging from about 0.95× to 10× compared to using explicit dense submatrices), although the total number of iterations required can be larger.

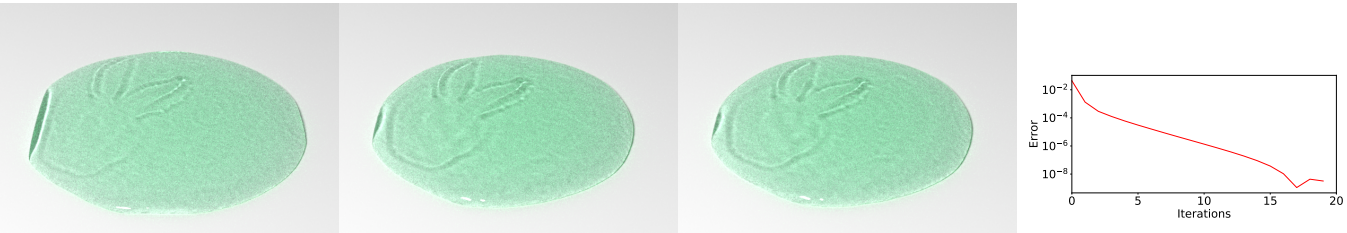

Fig. 9. A liquid bunny with shear thinning viscosity dropped onto the ground using our nonlinear viscosity handling, with 1, 2, and 3 solver iterations from left to right. The flow behaves more Newtonian with fewer iterations, while with more iterations it exhibits shear thinning effects (note the better-preserved bunny head/ear pattern and slightly slower spreading towards the end of the simulation). The right figure plots the log-scale convergence at frame 10 with respect to number of staggered projection iterations.

4.4.4 One-way solid-to-fluid coupling. Our formulation can naturally support one-way solid-to-fluid coupling for static or prescribed-motion rigid objects. Considering one-way coupled rigid bodies (i.e., with infinite mass), we can define $\mathbf{M}_r^{-1} = 0$, indicating no contributions from the rigid objects in several submatrices of the system matrix (e.g., $\mathbf{W}_L^s \mathbf{J}_s^T \mathbf{M}_r^{-1} \mathbf{J}_s \mathbf{W}_L^s$ in $\mathbf{A}_{11}$ will vanish), and their contributions appear only in the right hand side of the system. Thus, in practice, we decompose $\mathbf{J}_s$ into $\mathbf{J}_{s,1}$ and $\mathbf{J}_{s,2}$ for one-way and two-way coupled rigid bodies, respectively. $\mathbf{J}_p$, $\mathbf{J}_\lambda$, and $\mathbf{J}_z$ are similarly decomposed to support both one-way and two-way coupling.

4.4.5 Diagonal rescaling. Due to the structure of $\Delta t \alpha \mathbf{J}_\lambda \mathbf{M}_r^{-1} \mathbf{J}_\lambda^T$ in $\mathbf{A}_{33}$ and $\Delta t \alpha \mathbf{J}_z \mathbf{M}_r^{-1} \mathbf{J}_z^T$ in $\mathbf{A}_{44}$, it is mathematically guaranteed that these matrices are SPD. In practice, however, these can be non-SPD because of numerical errors [Smith 2008; Tan et al. 2012], and MPRGP could diverge in some situations. To prevent divergence, we slightly increase the diagonal elements by rescaling them with $(1 + \gamma)$, where we typically use $\gamma = 1.0 \times 10^{-4}$.

4.4.6 Level-set shrinkage. The contact constraints theoretically ensure no penetration between rigid bodies. In practice, however, it is preferable to allow some penetrations to avoid frequent contact switches as mentioned in §3.2.4. In addition, because of the velocity-level collision handling, some penetrations cannot be perfectly resolved even though stabilization techniques are employed. As such, there can be small penetrations between rigid bodies, which would make the system ill-conditioned or unsolvable since volume fractions may become inconsistent. To address this issue, we slightly shift the signed distance of solid objects inward by ζ ($= 0.01\Delta x$ in our examples) for volume fraction evaluation.

4.4.7 Mass consistency. While the effective mass of objects (determined by the approximate volume fractions) can be used to achieve perfect balance in ideal hydrostatic fluid tests [Batty et al. 2007; Bridson 2015], in our monolithic formulation, we prefer to use the actual mass computed from the density and volume of rigid bodies' true geometry for consistency among the pressure, viscosity, and frictional contact forces.

4.5 Algorithm

Algorithm 1 summarizes our complete monolithic solver.

Algorithm 1 Monolith solver

1: Map velocity from particles to grid
2: Add external force to grid
3: Add external force to rigid bodies
4: Compute fluid/solid and liquid/air domains
5: Evaluate volume fractions $\mathbf{W}_F^u, \mathbf{W}_F^s, \mathbf{W}_L^p, \mathbf{W}_L^u, \mathbf{W}_L^s$
6: Detect collisions between rigid bodies
7: Determine valid pressure, velocity, and viscous stress DOFs
8: Assemble $\mathbf{G}, \mathbf{D}, \mathbf{P}, \mathbf{N}, \mathbf{C}, \mathbf{J}_s, \mathbf{J}_p, \mathbf{J}_\lambda, \mathbf{J}_z$
9: Assemble the system
10: Solve the system with staggered projections
11: Apply pressure and viscosity forces to fluids
12: Apply pressure, viscosity, contact, and friction forces to solids
13: Update rigid body positions and signed distance
14: Advect particles
15: Correct particle positions for volume preservation

5 NON-NEWTONIAN FLUIDS

While we have assumed Newtonian fluids so far, we now consider augmenting our formulations to handle more general, non-Newtonian fluids. Among non-Newtonian fluids, we focus on those with non-Newtonian *viscosity*, which can be formulated with an effective or apparent viscosity that is dependent on the shear rate $\dot{\boldsymbol{\gamma}} (= \mathbf{W}_L^s \mathbf{D} \mathbf{W}_F^u \mathbf{u})$, but not the history of fluid deformations. That is, the viscous stress can be defined as $\boldsymbol{\tau} = 2(\mathbf{W}_F^s)^{-1}(\mathbf{W}_L^s)^{-1}\mathbf{N}(\dot{\boldsymbol{\gamma}})\dot{\boldsymbol{\gamma}}$, using an effective viscosity function, $\mathbf{N}(\dot{\boldsymbol{\gamma}})$. Fluids in this category include shear thickening (dilatant), shear thinning (pseudoplastic), and Bingham plastic fluids. Variants of non-Newtonian viscosity have been previously considered for Lagrangian finite elements and MPM (see e.g., [Bargteil et al. 2007; Yue et al. 2015; Zhu et al. 2015]). However, a simple and robust *implicit* non-Newtonian viscosity of the form we consider has not been deployed for grid-based viscous liquids; that is, we treat the effective viscosity coefficient itself in an implicit fashion so it responds accurately to rapid changes in deformation behavior. In this work, we employ the Herschel-Bulkley model, as described by Sahu et al. [2007], because it offers a unified representation for a variety of non-Newtonian fluids (Herschel-Bulkley was also used in an elastoplastic context by Yue et al. [2015] for MPM-based foam). The effective viscosity at the location of each

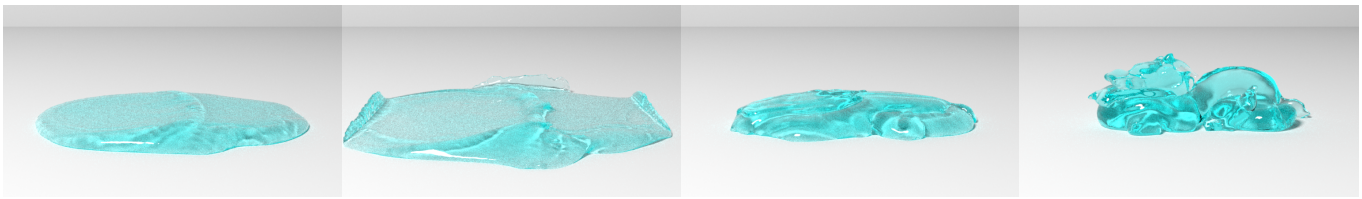

Fig. 10. A liquid dragon dropped onto the ground with different viscosity models. (Left) Newtonian viscosity. (Middle left) Shear thinning viscosity. (Middle right) Shear thickening viscosity. (Right) Bingham-type viscosity. Due to the large shear rates induced by the collision with the ground, the effective viscosity of the shear thinning model becomes smaller, causing the liquid to flow more rapidly than its Newtonian counterpart. By contrast, shear thickening increases the effective viscosity leading to a more viscous effect. Finally, the dragon with Bingham-type viscosity shows a more solid-like behavior at low shear rates.

stress DOF can be computed using

$$\boldsymbol{\eta} = \begin{cases} \boldsymbol{\eta}_0 = k\dot{\boldsymbol{\gamma}}_0^{r-1} + \boldsymbol{\tau}_0\dot{\boldsymbol{\gamma}}_0^{-1} & (\left\|\dot{\boldsymbol{\gamma}}\right\|_F \le \dot{\boldsymbol{\gamma}}_0) \\ k\left\|\dot{\boldsymbol{\gamma}}\right\|_F^{r-1} + \boldsymbol{\tau}_0\left\|\dot{\boldsymbol{\gamma}}\right\|_F^{-1} & (\text{otherwise}) \end{cases}, \tag{27}$$

where $\boldsymbol{\eta}_0$ denotes the limiting viscosity, r the flow index, $\boldsymbol{\tau}_0$ the yield stress, k the consistency index, and $\dot{\boldsymbol{\gamma}}_0$ the threshold for yielding. With $0 < \boldsymbol{\tau}_0$, fluids behave as Bingham-type materials, which flow if $\dot{\boldsymbol{\gamma}}$ is sufficiently large. With $r < 1$ and $r > 1$, fluids are shear thinning and shear thickening, respectively, while fluids are Newtonian with $r = 1$ and $\boldsymbol{\tau}_0 = 0$. In the actual evaluation of $\left\|\dot{\boldsymbol{\gamma}}\right\|_F$, we interpolate $\dot{\boldsymbol{\gamma}}$ from the neighbors because of the staggered arrangement of $\dot{\boldsymbol{\gamma}}$.

One approach to addressing the effective viscosity is to substitute the effective viscosity and shear rate for viscous stress ($\boldsymbol{\tau} = 2(\mathbf{W}_F^s)^{-1}(\mathbf{W}_L^s)^{-1}\mathbf{N}(\dot{\boldsymbol{\gamma}})\dot{\boldsymbol{\gamma}}$), obtaining the following formulation:

$$\begin{aligned} \mathbf{p}, \dot{\boldsymbol{\gamma}} = &\underset{\mathbf{p},\dot{\boldsymbol{\gamma}}}{\arg\min} \\ &\frac{1}{2}\left(\left\|\mathbf{u}^* - \Delta t\mathbf{P}^{-1}(\mathbf{W}_L^u)^{-1}\left(\mathbf{G}\mathbf{W}_L^p\mathbf{p} + 2\mathbf{D}^T(\mathbf{W}_F^s)^{-1}\mathbf{N}(\dot{\boldsymbol{\gamma}})\dot{\boldsymbol{\gamma}}\right)\right\|_{\mathbf{M}_f}^2 \right. \\ &\left. + \frac{\Delta t}{2}\left\|2\dot{\boldsymbol{\gamma}}\right\|_{\mathbf{N}(\dot{\boldsymbol{\gamma}})(\mathbf{W}^s)^{-1}}^2\right). \end{aligned} \tag{28}$$

Unfortunately, this minimization problem is not quadratic with respect to shear rate $\dot{\boldsymbol{\gamma}}$, and nonlinear solvers such as (quasi-)Newton's method or nonlinear CG would be necessary, typically requiring a significant computational effort.

Given the fact that, in our formulation, the unique source of nonlinearity (aside from advection) is in the viscosity coefficient computation, we decouple this computation from the rest. Mathematically, we reformulate the unified minimization problem as

$$\begin{aligned} \mathbf{p}, \boldsymbol{\tau}, \dot{\boldsymbol{\gamma}} = &\underset{\mathbf{p},\boldsymbol{\tau}=2(\mathbf{W}_F^s)^{-1}(\mathbf{W}_L^s)^{-1}\mathbf{N}(\dot{\boldsymbol{\gamma}})\dot{\boldsymbol{\gamma}}}{\arg\min} \\ &\frac{1}{2}\left(\left\|\mathbf{u}^* - \Delta t\mathbf{P}^{-1}(\mathbf{W}_L^u)^{-1}\left(\mathbf{G}\mathbf{W}_L^p\mathbf{p} + \mathbf{D}^T\mathbf{W}_L^s\boldsymbol{\tau}\right)\right\|_{\mathbf{M}_f}^2 + \frac{\Delta t}{2}\left\|\boldsymbol{\tau}\right\|_{\mathbf{N}^{-1}(\dot{\boldsymbol{\gamma}})\mathbf{W}^s}^2\right). \end{aligned} \tag{29}$$

To solve this minimization, we can alternate between computing the effective viscosity and solving the quadratic minimization in the same fashion as for (8). The computational process is summarized in Algorithm 2. At the end of the each iteration, pressure and viscous stress are optimal for the current viscosity $\mathbf{N}^k$, satisfying the divergence-free constraints and implicit viscosity update.

Our formulation for non-Newtonian fluids can be naturally integrated into the monolithic solver by simultaneously solving contact forces alongside pressure and viscous stress. Mathematically, this coupling of non-Newtonian fluids and frictional contacts can be formulated by replacing $E_f(\boldsymbol{\tau}, \mathbf{p})$ in (22) with (29). Moreover, our iterative procedure for the effective viscosity update and system solve can be considered as a kind of staggered projection, and thus the viscosity update can be handled alongside the friction solve (followed by the pressure-viscosity-contact solve) within the staggered projections approach.

Algorithm 2 Non-Newtonian fluid solver

1: Initialize shear rate by $\dot{\boldsymbol{\gamma}}^1 = \mathbf{W}_L^s\mathbf{D}\mathbf{W}_F^u\mathbf{u}^*$
2: **for** k = 1, 2, ... **do**
3: Compute effective viscosity $\mathbf{N}^k(\dot{\boldsymbol{\gamma}}^k)$ based on (27)
4: Assemble the system matrix $\mathbf{A}^k$ with $\mathbf{N}^k(\dot{\boldsymbol{\gamma}}^k)$
5: Solve $\mathbf{A}^k\mathbf{x}^{k+1} = \mathbf{b}$, where $\mathbf{x} = (\boldsymbol{\tau}^T, \mathbf{p}^T)^T$ and $\mathbf{b} = (\mathbf{b}_1^T, \mathbf{b}_2^T)^T$
6: Update shear rate by $\dot{\boldsymbol{\gamma}}^{k+1} = \frac{1}{2}\mathbf{N}^{-1}(\dot{\boldsymbol{\gamma}}^k)\mathbf{W}_F^s\mathbf{W}_L^s\boldsymbol{\tau}^{k+1}$

In addition to this compatibility benefit, as compared to the fully nonlinear formulation, (28), our staggered projections approach has several added advantages. First, the structure of the system remains the same as for Newtonian fluids. As such, we can use the same LCP solver and preconditioner, significantly aiding in our solver optimization and minimizing additional implementation and maintenance effort. Furthermore, given the importance of pressure, viscous stress, and contact forces, it is preferable to shift residuals to the less critical, nonlinear viscosity component. Since the staggered projections approach can terminate the iteration with optimal pressure, viscous stress, and contact forces for the currently available variables, all of the residuals can be shifted onto the nonlinear viscosity and friction parts. This residual control allows us to use intermediate results as solutions with early termination if efficiency is more important than accuracy, while maintaining solver robustness with larger time steps. With the fully unified nonlinear formulation, (28), it can be difficult to control the effect of the residual, and early termination would lead to divergent velocities, unstable viscosity integration, and penetrations between rigid bodies.

Figure 9 illustrates a liquid bunny with shear thinning viscosity using different numbers of iterations, along with a convergence plot. A single iteration would correspond to an explicit treatment of these nonlinear effects e.g., [Zhu et al. 2015]. With a sufficient number of iterations, the liquid exhibits more accurate shear thinning behaviors: it easily flows on impact with the ground due to high shear

Fig. 11. A rigid bunny dropped into a block of liquid with different viscosity models. (Left) Newtonian. (Middle left) Shear thinning. (Middle right) Shear thickening. (Right) Bingham-type. Due to the high shear rates caused by the falling rigid bunny, the effective viscosity of the shear thinning model decreases compared to the Newtonian model allowing the bunny to sink more quickly. Conversely, the effective viscosity of the shear thickening model increases with the large shear rates, so the bunny falls more slowly. When the shear rate is insufficiently high, the liquid with Bingham-type viscosity exhibits solid-like behavior, and thus rapidly slowing the falling bunny.

rates while the flow later slows down due to small shear rates. We found that the staggered projections can stably and quickly decrease the residual, and the resulting behaviors do not change dramatically with many iterations. Thus, in practice, we use two iterations for nonlinear viscosity.

6 RESULTS AND DISCUSSION

Our method is implemented in C++ and parallelized using OpenMP. We used a convergence criterion of 10^{-6} on the infinity norm of the relative residual and assign maximum iteration counts per scenario. We used a CFL number of 3.0 with adaptive timestepping, and used 50 frames per second. We executed simulations for Figures 2, 3, 4, 5, and 6 on an Intel Xeon E5-2680 v2 (using only 8 threads) with 16GB RAM, for Figures 7, 10, and 11 on an Intel Xeon X5560 (using 16 threads) with 16GB RAM, and for Figures 15, 17, and 18 on an Intel Core i7-9700 (using 8 threads) with 16GB RAM. All reported statistics are averaged over the effective simulation length, excluding results from simulation failures. Except where specifically mentioned, we do not include frictional contacts nor non-Newtonian fluids, and thus perform a single LCP solve for each time step. When there are no box constraints, we use SOR-CG, because SOR-CG is about 2-3× faster than SOR-MPRGP in practice.

6.1 Contact-Aware Coupling with Inviscid Liquids

To demonstrate the benefits of our monolithic coupling even in the absence of viscosity, we first experiment with purely inviscid liquids and rigid bodies. In these experiments, we use a specialized solver that uses cut-cell-based area weighting for $\mathbf{W}_F^u$ and $\mathbf{W}_S^u$ [Ng et al. 2009] and the ghost-fluid method for $\mathbf{W}_L^p$ and $\mathbf{W}_L^u$ [Gibou et al. 2002] (instead of volume weights) for better accuracy. We evaluate four possible pressure-contact coupling schemes:

(1) P-C solve: pressure solve first followed by contact solve;
(2) C-P solve: contact solve first followed by pressure solve;
(3) PC-iterative solve: iterative *partitioned* pressure and contact solves, (i.e., staggered projections on the unified formulation);
(4) PC-unified solve: our proposed method that solves pressure and contact *monolithically*.

P-C solve, C-P solve, and too few (staggered projection) iterations of PC-iterative solve are instances of weak coupling, while PC-iterative solve with sufficiently many iterations and PC-unified solve are instances of strong coupling. Simulation settings and performance numbers for these experiments are summarized in Table 1. Profiles of total time for the entire pressure and contact handling phase (i.e., excluding other simulator components) are compared in Figure 12.

6.1.1 Inviscid liquid bunny drop. In Figure 2, we drop a bunny-shaped volume of inviscid liquid (density 1,000 kg/m^3) into a bowl (density 200 kg/m^3) resting on the ground. The liquid and bowl are two-way coupled; one-way coupling is applied to the liquid-ground and bowl-ground pairs. Although the bowl-ground coupling is one-way, it is essential to make the system aware of their collisions to prevent the liquid from pushing the bowl into the ground.

For P-C solve, while bowl-ground contacts are properly addressed by the concluding contact solve, incompressibility is not enforced because the contact solve disturbs the divergence-free velocity fields initially computed by the pressure solve. Consequently, we observe significant volume loss as the liquid in the bowl disappears (note the reduced p_{dof} count in Table 1 due to the smaller liquid volume). Conversely, for C-P solve, the volume of liquid is preserved due to the concluding pressure solve, but the pressure solve also destroys the non-penetration enforcement of the initial contact solve. This leads to solid penetrations which cause objectionable oscillatory behaviors in the bowl's motion.

For PC-iterative solve and PC-unified solve, the resulting behaviors are comparable due to consistent solutions upon convergence (see supplementary material). However, despite relatively few contacts, the number of staggered projection iterations needed by PC-iterative solve can be large, resulting in a higher total cost regardless of the efficiency of SOR-CG over SOR-MPRGP. Here, our proposed PC-unified solve is approximately 10× faster than PC-iterative solve, and comparable to the P-C and C-P solves. While iterative convergence of the partitioned solves towards the strong coupling result

Table 1. Simulation settings and performance results. P, C, and PC denote the total time per frame for pressure, contact, and pressure-contact solve, respectively. T denotes the total time per frame for the entire pressure and contact handling phase. Here, P + C are not necessarily equal to T because of the solver overhead, e.g., updates of the right hand side in the staggered projections for PC-iterative. st_{iter} denotes the number of iterations for the staggered projections per simulation step. p_{iter}, c_{iter}, and pc_{iter} denote the number of iterations for each of pressure, contact, and pressure-contact solves, respectively. p_{dof} and c_{dof} denote the number of pressure and contact DOFs (i.e., the number of unknown pressure variables and the number of contacts) in each solve, respectively. Empty cells indicate N/A. Gray rows indicate early-terminated simulations because of failures.

Scene	Grid resolution	Particles	Scheme	P (s)	C (s)	PC (s)	T (s)	st_{iter}	p_{iter}	c_{iter}	pc_{iter}	p_{dof}	c_{dof}
Figure 2 (left)	256 × 128 × 256	418.5k	P-C	4.86	0.56		5.43		38.1	3.3		73.0k	2.4
Figure 2 (middle left)	256 × 128 × 256	418.5k	C-P	10.29	0.47		10.77		68.8	1.7		143.6k	4.0
Figure 2 (middle right)	256 × 128 × 256	418.5k	PC-iterative	108.89	4.12		115.87	10.5	71.6	3.2		142.5k	2.5
Figure 2 (right)	256 × 128 × 256	418.5k	PC-unified			11.59	11.59				51.3	142.5k	2.6
Figure 3 (left)	128 × 128 × 128	2,697.9k	P-C	22.53	0.66		24.41		161.6	2.6		357.0k	2.1
Figure 3 (middle left)	128 × 128 × 128	2,697.9k	C-P	36.86	0.76		37.82		187.2	4.7		448.5k	4.2
Figure 3 (middle right)	128 × 128 × 128	2,697.9k	PC-iterative	249.68	0.07		257.43	21.5	181.2	2.6		417.8k	3.3
Figure 3 (right)	128 × 128 × 128	2,697.9k	PC-unified			32.14	32.14				108.1	389.3k	2.3

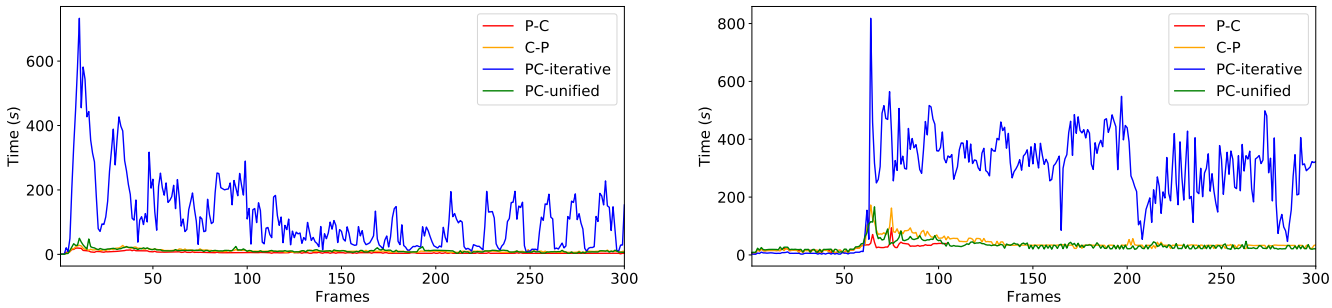

Fig. 12. Profiles of the total times for the entire pressure and contact handling phase per frame for Figure 2 (left) and Figure 3 (right). PC-unified is as efficient as P-C and C-P and is approximately 10× faster than PC-iterative.

could potentially be accelerated, (e.g., by extending the technique of Akbay et al. [2018] to our LCP problem), we believe that our tightly unified approach will be faster, considering that the cost of PC-unified solve is already comparable to P-C and C-P solves.

6.1.2 Rigid bunny drop. In Figure 3, we drop a rigid bunny (density 3,000 kg/m^3) into a rigid bowl (density 500 kg/m^3) floating on liquid (density 1,000 kg/m^3). We use a maximum of 200 for both solver iterations and staggered projection iterations.

Since P-C solve allows contact handling to override incompressibility, splashes caused by the bowl are smaller than they should be. Furthermore, the pressure solve frequently failed to converge due to stuck particles inside rigid bodies caused by collision handling, leading to simulation failure. For C-P solve, the neglected contact handling failed to properly address bunny-bowl collisions, resulting in simulation blow-up and causing the bowl and bunny to topple, despite few detected contacts (see Table 1). Although PC-iterative solve theoretically converges to the same result as PC-unified solve, it was halted before convergence due to the large number of required iterations; it thus fails to satisfy the necessary constraints, and generates implausible results. Our PC-unified solve successfully generated plausible behaviors for the liquid, bunny, and bowl with a reasonable cost, comparable to that of P-C and C-P solves, and is approximately 8× faster than (non-converged) PC-iterative solve.

6.2 Two-Way Rigid-Fluid Coupling with Stokes

While the benefits of the unified (Stokes) treatment for pressure and viscosity have previously been demonstrated for fluid-only problems (i.e., elimination of artificial melting and improved coiling [Larionov et al. 2017]), we explore below the advantages offered by our combining it with strong two-way fluid-solid coupling.

We compare two possible schemes: 1) PC-VC-PC-unified solve, which separates viscosity and pressure, and 2) PVC-unified solve, which is our proposed monolithically coupled pressure-viscosity-contact solver. PC-VC-PC-unified solve is based on a traditional decoupled viscous flow solver [Batty and Bridson 2008] that performs, in order, a pressure solve, a viscosity solve, and a second pressure solve. We augment it with the decoupled two-way fluid-solid coupling formulations (§3.3) and two-way rigid-rigid coupling formulations (§4). PC-VC-PC-unified solve does *not* generate the same results as PVC-unified solve because it performs the predetermined solver sequence only once, and neglects interactions between pressure and viscosity forces. We use viscous stresses as unknowns for the viscosity solve (unlike Batty and Bridson [2008] and Takahashi and Lin [2019] who use velocity unknowns) since rigid-rigid coupling combined with a viscosity solve in velocity unknowns leads to less desirable symmetric *indefinite* systems.

Table 2. Simulation settings and performance results. PC, VC, and PVC denote the total time per frame for pressure-contact, viscosity-contact, and pressure-viscosity-contact solves, respectively. T denotes the total time for the entire pressure, viscosity, and contact handling phase. $\mathrm{pc}_{\mathrm{iter}}$, $\mathrm{vc}_{\mathrm{iter}}$, and $\mathrm{pvc}_{\mathrm{iter}}$ denote the number of iterations for each of pressure-contact, viscosity-contact, and pressure-viscosity-contact solve, respectively. $\mathrm{p}_{\mathrm{dof}}$, $\mathrm{s}_{\mathrm{dof}}$, and $\mathrm{c}_{\mathrm{dof}}$ denote the number of pressure, stress, and contact DOFs in each solve, respectively. Empty cells indicate N/A.

Scene	Grid resolution	Particles	Scheme	PC (s)	VC (s)	PVC (s)	T (s)	$\mathrm{pc}_{\mathrm{iter}}$	$\mathrm{vc}_{\mathrm{iter}}$	$\mathrm{pvc}_{\mathrm{iter}}$	$\mathrm{p}_{\mathrm{dof}}$	$\mathrm{s}_{\mathrm{dof}}$	$\mathrm{c}_{\mathrm{dof}}$
Figure 4 (left)	$128 \times 128 \times 128$	562.1k	PC-VC-PC-unified	15.36	179.47		194.84	45.5	395.1		95.6k	474.3k	1.8
Figure 4 (right)	$128 \times 128 \times 128$	562.1k	PVC-unified			344.57	344.57			942.0	90.7k	451.8k	3.1
Figure 5 (left)	$128 \times 128 \times 128$	709.1k	PC-VC-PC-unified	1.88	302.96		304.84	78.1	1,983.0		109.5k	530.9k	0.0
Figure 5 (right)	$128 \times 128 \times 128$	709.1k	PVC-unified			474.95	474.95			1,990.4	107.0k	522.7k	0.0

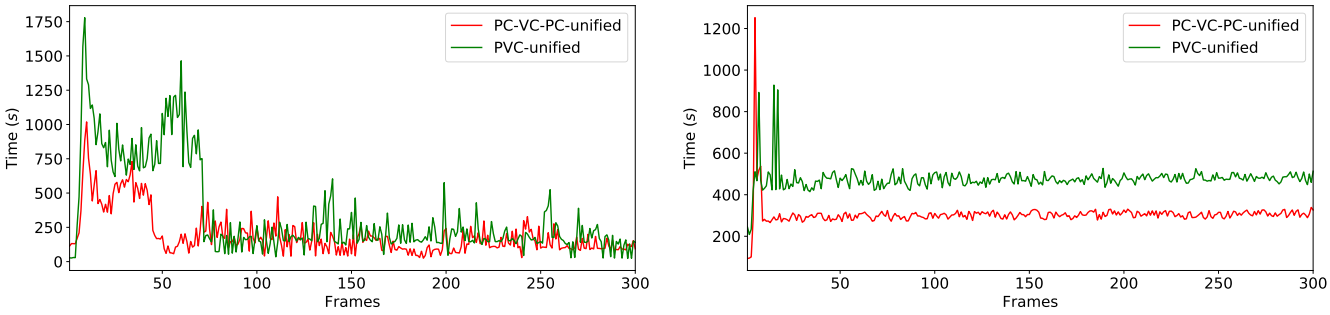

Fig. 13. Profiles of the total time of the entire pressure, viscosity, and contact handling phase per frame for Figure 4 (left) and Figure 5 (right). In our examples, PVC-unified is approximately 1.66 × slower than PC-VC-PC-unified, which suffers from artificial melting of viscous liquids.

Simulation settings and performance numbers for our experiments are summarized in Table 2, with profiles of total time for the pressure, viscosity, and contact handling phase given in Figure 13. We set the maximum number of solver iterations to 2,000.

6.2.1 Viscous sphere dragged by bunny. In Figure 4, we throw a viscous sphere (density 1,000 $\mathrm{kg/m^3}$; dynamic viscosity 1,000 $\mathrm{kg/(s \cdot m)}$) and a rigid bunny (density 2,000 $\mathrm{kg/m^3}$) toward a wall.

Although the viscosity solve applies the expected no-slip boundary condition, the second pressure projection of the PC-VC-PC-unified solve disturbs the results of the viscosity solve and destroys the no-slip boundary condition. Consequently, the viscous sphere unnaturally peels off from the wall while failing to support the bunny, both of which we identify as new and disturbing artificial melting artifacts. In addition, because of the decoupled pressure and viscosity treatment, the velocity field of the viscous fluid is less smooth than expected, causing particles to scatter.

With our PVC-unified solve, the no-slip boundary condition is always strictly enforced, so the viscous sphere remains stuck to the wall, captures the thrown rigid bunny on contact, and slows the bunny's descent towards the ground. Furthermore, our approach produces appropriately smooth velocity fields, and visually reasonable dynamics throughout. Although the computational cost for the unified treatment is higher (approximately 1.77×) than the artifact-prone decoupled approach, we believe the more natural solid interactions justify the extra expense.

6.2.2 Bunny on viscous hemisphere. In Figure 5, we drop a rigid bunny (density 3,000 $\mathrm{kg/m^3}$) onto a viscous hemisphere (density 1,000 $\mathrm{kg/m^3}$; viscosity 10,000 $\mathrm{kg/(s \cdot m)}$). Due to the decoupled pressure and viscosity handling of PC-VC-PC-unified solve, artificial melting occurs, resulting in faster sinking of the bunny and (no-slip-violating) crawling of fluid up the bunny's sides, despite a high viscosity coefficient. By contrast, with PVC-unified solve, the viscous fluid more tightly supports the bunny, slowing down its sinking without spurious crawling artifacts. This improved accuracy again comes at a modest cost: PVC-unified solve is about 1.56× slower than PC-VC-PC-unified solve.

6.3 Contact-Aware Coupling with Viscous Liquids

Having separately shown the importance of tight pressure-contact and pressure-viscosity coupling, we next demonstrate our full monolithic formulation for simultaneous coupling of viscous liquids and rigid bodies with contact. We compare four possible schemes:

(1) PV-C solve: Stokes solve, followed by contact solve;
(2) C-PV solve: contact solve, followed by Stokes solve;
(3) PVC-iterative solve: iterative *partitioned* Stokes and contact solves (i.e., staggered projections on the unified problem);
(4) PVC-unified solve: our proposed method that solves pressure, viscosity, and contact monolithically.

We summarize simulation settings and performance numbers in Table 3 and show profiles of total time for handling pressure, viscosity, and contact in Figure 14.

6.3.1 Viscous dragon into a bowl. In Figure 6, we drop a dragon-shaped viscous liquid volume (density 1,000 $\mathrm{kg/m^3}$; viscosity 100 $\mathrm{kg/(s \cdot m)}$) into a bowl (density 100 $\mathrm{kg/m^3}$) on the ground. For PV-C solve, although bowl-ground contacts are properly resolved by

Table 3. Simulation settings and performance results. PV, C, and PVC denote total time per frame for pressure-viscosity, contact, and pressure-viscosity-contact solves, respectively. T denotes the total time per frame for the entire pressure, viscosity, and contact handling phase. Here, the sum PV + C is not necessarily equal to T because of solver overhead. st_{iter} denotes the number of iterations for the staggered projections per simulation step. pv_{iter}, c_{iter}, and pvc_{iter} denote the number of iterations for each of pressure-viscosity, contact, and pressure-viscosity-contact solves, respectively. p_{dof}, s_{dof}, and c_{dof} denote the number of pressure, stress, and contact DOFs in each solve, respectively. Empty cells indicate N/A. Gray rows indicate early terminated simulations because of failures.

Scene	Grid resolution	Particles	Scheme	PV (s)	C (s)	PVC (s)	T (s)	st_{iter}	pv_{iter}	c_{iter}	pvc_{iter}	p_{dof}	s_{dof}	c_{dof}
Figure 6 (left)	$128 \times 128 \times 128$	462.7k	PV-C	95.02	0.64		95.66		696.7	5.8		21.0k	97.6k	4.0
Figure 6 (middle left)	$128 \times 128 \times 128$	462.7k	C-PV	382.81	0.01		382.83		746.4	0.8		118.2k	568.6k	1.9
Figure 6 (middle right)	$128 \times 128 \times 128$	462.7k	PVC-iterative	687.15	0.03		719.49	7.0	951.6	3.6		84.1k	413.1k	4.0
Figure 6 (right)	$128 \times 128 \times 128$	462.7k	PVC-unified			103.30	103.30				183.7	84.0k	413.4k	4.0
Figure 7 (left)	$64 \times 64 \times 64$	348.2k	PV-C	338.02	0.07		338.09		4,773.7	354.5		50.7k	237.4k	72.8
Figure 7 (middle left)	$64 \times 64 \times 64$	348.2k	C-PV	735.52	0.05		735.57		7,310.1	72.4		55.8k	262.8k	68.8
Figure 7 (middle right)	$64 \times 64 \times 64$	348.2k	PVC-iterative	3,816.06	0.14		3,838.82	9.0	888.0	7.5		53.5k	252.0k	22.3
Figure 7 (right)	$64 \times 64 \times 64$	348.2k	PVC-unified			1,177.49	1,177.49				4,336.7	55.5k	261.6k	72.4

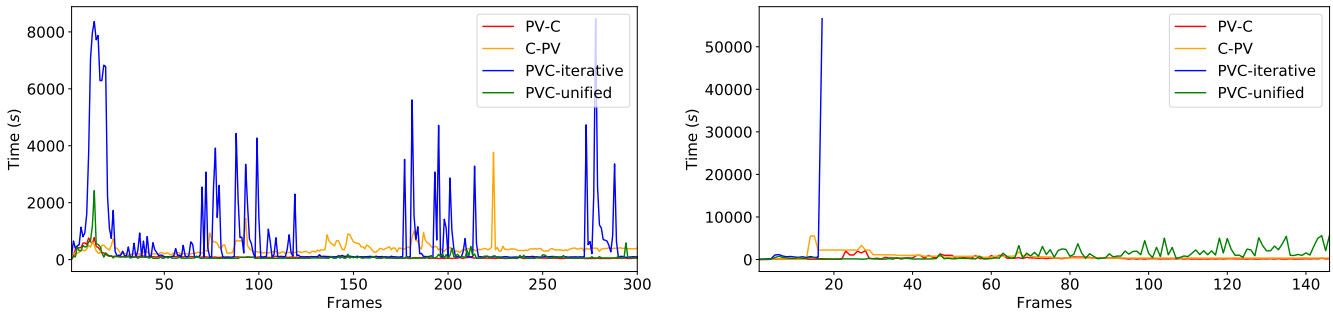

Fig. 14. Profiles of the total time for the entire pressure, viscosity, and contact handling phase per frame for Figure 6 (left) and Figure 7 (right). PVC-unified is as fast as PV-C and C-PV and is around 7× faster than PVC-iterative.

the prioritized contact handling, fluid incompressibility and viscosity integration are spoiled, causing significant volume loss. Conversely, for C-PV solve, the prioritized Stokes solve yields plausible liquid behavior, but the decoupled contact handling leaves severe bowl-ground penetrations that drive the bowl to jump unnaturally. PVC-iterative solve and PVC-unified solve generated plausible and visually comparable results since they converge to the same solution. However, PVC-iterative solve requires many alternating iterations to converge, and thus PVC-unified solve is $\sim 7\times$ faster.

6.3.2 Multiple rigid bunnies dropped into a bowl floating on viscous fluid. In Figure 7, we drop nine bunnies (density 2,000 kg/m^3) into a rigid bowl (density 500 kg/m^3) floating on viscous liquid (density 1,000 kg/m^3; viscosity 100 $kg/(s \cdot m)$). We use a maximum of 20,000 solver iterations and 200 staggered projection iterations.

While PV-C solve properly handles contacts between the bunnies and the bowl, the neglected pressure-viscosity handling makes the viscous fluids behave unnaturally, e.g., exhibiting loss of fluid volume and crawling of fluid on the bowl. For C-PV solve, although the viscous fluid behaves naturally, contact-handling is neglected, which causes severe penetrations, driving the orange rigid bunny to jump to the outside of the bowl. PVC-iterative solve failed to converge for this example, causing the simulation to explode and leave nothing in the domain, since neither constraints for incompressibility, implicit viscosity integration, nor non-penetration are satisfied.

Our PVC-unified solve successfully generated plausible behaviors for the viscous liquid, bunny, and bowl, at a reasonable cost comparable to PV-C and C-PV solves. Although it is not entirely fair to compare PVC-unified and PVC-iterative solves due to the failure of the iterative approach, the latter took 56,587.2s for one frame (before exploding) whereas PVC-unified solve took 1,177.5s on average. This suggests that our PVC-unified solve offers a significant performance improvement compared to PVC-iterative solve. We also note that the larger number of required iterations in this scene is not specifically due to incorporating contacts, since the PV-C and C-PV solves also require similar iteration counts.

6.4 Non-Newtonian Fluids

Next, we demonstrate the effectiveness of our nonlinear viscosity handling. We used different sets of parameters for the Herschel-Bulkley model of (27), i.e., $(r, \dot{\boldsymbol{\gamma}}_0, \boldsymbol{\tau}_0, k)$, to model and compare Newtonian, shear thinning, shear thickening, and Bingham-type viscosity, and used two staggered projection iterations.

6.4.1 Dragon drop. In Figure 10, we drop a liquid dragon with parameters $(1.0, 1.0, 0.0, 1.1 \times 10^1)$ for Newtonian, $(0.5, 1.0, 1.0, 1.0 \times 10^1)$ for shear thinning, $(2.0, 1.0, 1.0, 1.0 \times 10^1)$ for shear thickening, and $(1.0, 1.0, 1.0 \times 10^3, 1.0 \times 10^1)$ for Bingham-type viscosity. The simulation used a grid resolution of $128 \times 128 \times 128$ with 194.6k particles, and the total time for the pressure-viscosity handling per frame was

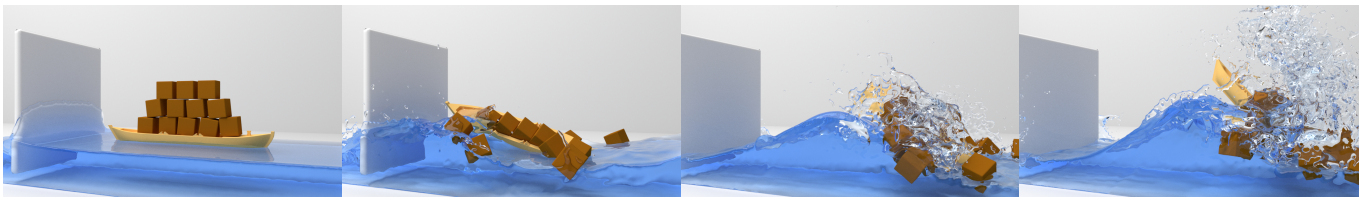

Fig. 15. A boat carrying multiple boxes being perturbed by waves created by an oscillating board.

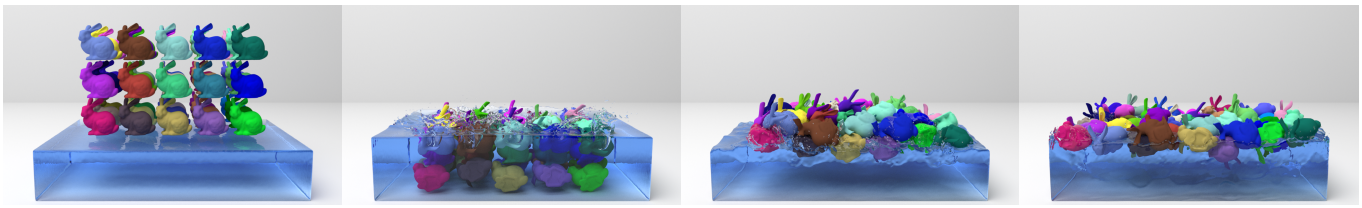

Fig. 16. 45 bunnies dropped onto an inviscid liquid pool.

13.09 s, 16.27 s, 18.65 s, and 74.84 s for Newtonian, shear thinning, shear thickening, and Bingham-type viscosity, respectively.

For the shear thinning model, the effective viscosity decreases under the high shear rates from the collision with the ground, and the liquid flows more vividly than in the Newtonian case. Then, as the shear rate decreases again, the effective viscosity increases, leading to a stiffer effect than the Newtonian model. For the shear thickening model, by contrast, when the dragon hits the ground, the effective viscosity instead rises, yielding a more damped effect. When the shear rates later decrease, the effective viscosity decreases and the liquid begins to flow more like the Newtonian model. With Bingham-type viscosity, due to the large yield stress $\boldsymbol{\tau}_0$, the liquid shows a more solid-like effect (except where large shearing is applied), so the original shape of the dragon is better preserved.

6.4.2 Bunny drop into a viscous block. To confirm that our non-Newtonian model is also compatible with our rigid-body coupling approach, we drop a rigid bunny into a tank of viscous liquid with parameters $(1.0, 1.0, 0.0, 1.0 \times 10^1)$ for Newtonian, $(0.5, 1.0, 1.0, 1.0 \times 10^1)$ for shear thinning, $(2.0, 1.0, 1.0, 1.0 \times 10^1)$ for shear thickening, and $(1.0, 1.0, 1.0 \times 10^3, 1.0 \times 10^2)$ for Bingham-type viscosity, as shown in Figure 11. Natural bunny-ground contact at the bottom of the tank is achieved through the combination of our nonlinear viscosity handling with monolithic pressure-viscosity-contact handling. The simulation used a grid resolution of 64×64×64 with 261.9k particles, and the total time for the pressure, viscosity, and contact handling per frame was 186.23 s, 209.87 s, 339.32 s, and 486.62 s for Newtonian, shear thinning, shear thickening, and Bingham-type viscosity, respectively.

For the shear thinning model, the strong shearing induced by the bunny falling into the liquid causes the effective viscosity to decrease and the bunny to sink more quickly than for the Newtonian model. For the shear thickening model, the effective viscosity instead increases, so the liquid behaves more viscous, and the bunny sinks more slowly. With Bingham-type viscosity, regions where the yield stress is not exceeded exhibit solid-like behavior, so the descent of the bunny is more strongly impeded.

6.5 Complex Examples

The following more complex examples further demonstrate the capabilities of Monolith. In these examples, we include friction ($\mu = 0.1$) and use two staggered projection iterations.

Figure 15 shows that our solver can efficiently and robustly handle complex interactions among inviscid ocean liquid, a kinematically scripted wave-inducing board, a wooden boat, and multiple wooden boxes. This simulation used a grid resolution of $240 \times 96 \times 96$ with 2.8M particles. The total simulation time per frame is 242.04 s, and the average contact count is 23.2 per simulation step.

In Figure 16, we simulate 45 rigid bunnies dropped onto a liquid pool, demonstrating that our solver can handle complex interactions involving many contacts. This simulation used a grid resolution of $225 \times 150 \times 150$ with 8.2M particles. The total simulation time per frame is 788.91 s, and the average contact count is 105.9 per simulation step.

Figures 17 and 18 demonstrate two hollow glass spheres containing inviscid and viscous liquid, respectively, rolling around in a basin. In this example, we used a CFL number of 0.5 for adaptive timestepping. The simulations of Figures 17 and 18 used, respectively, grid resolutions of $256 \times 256 \times 256$ and $128 \times 128 \times 128$ with 385.8k and 48.2k particles taking 81.44s and 260.93s per frame, for the most challenging period from frame 15 to 50. Naturally, the viscous simulation is significantly more costly due to the larger system to be solved. These scenarios highlight that our monolithic solver enables physically-consistent energy exchanges, i.e., the energy of the hollow glass spheres can change significantly by transferring energy to and from the liquids contained in them. This effect, along with the free-slip liquid boundary condition and the absence of viscosity, accounts for the longer term rolling in the basin with inviscid liquid. By contrast, in the viscous case, this coupling, combined

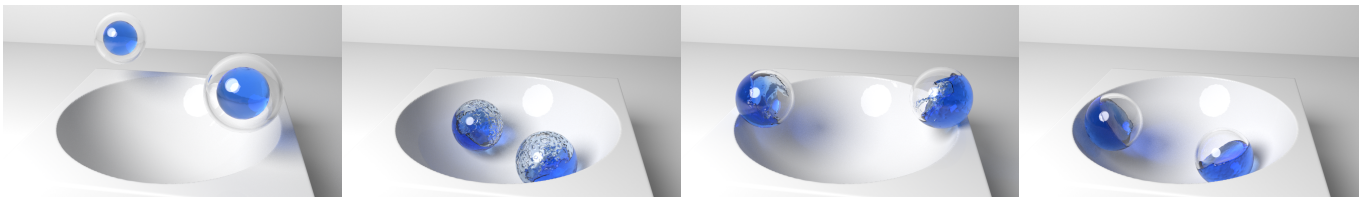

Fig. 17. Two hollow glass spheres containing *inviscid* liquid balls are thrown into a basin, shown at frames 1, 28, 55, 170 from left to right.

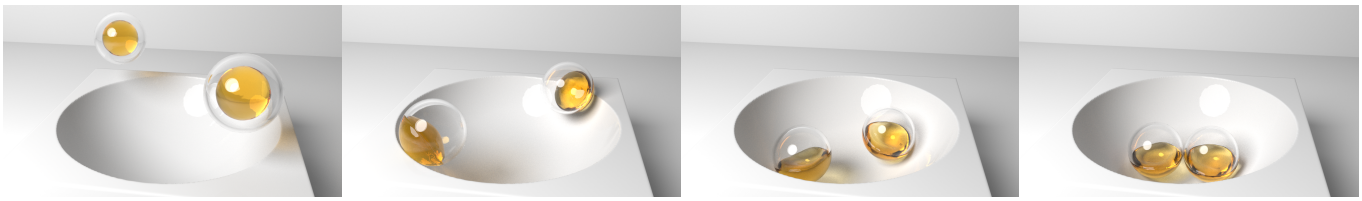

Fig. 18. Two spherical glass hulls containing *viscous* liquid balls are thrown into a basin, shown at frames 1, 16, 48, 75 from left to right.

with liquid viscosity, no-slip liquid boundary conditions, and frictional contacts between the spheres and basin, causes rapid energy dissipation.

6.6 Discussion

6.6.1 Solver choice. Our fully monolithic treatment significantly improves accuracy, robustness, and efficiency for general scenarios. In restricted scenarios, however, decoupling might be preferred. For example, if the benefits of unifying pressure and viscosity are not needed (e.g., accurate boundary/coupling behavior, lack of artificial melting, etc.) PC-VC-PC-unified solve offers slightly better performance. If, in addition, there is no contact between rigid bodies, two-way coupling with a velocity-based viscosity solver should be preferred [Takahashi and Lin 2019].

6.6.2 Position-level errors. While we used velocity-level contact formulations for rigid bodies to avoid repeated volume weight evaluations per simulation step, updated rigid bodies may have small penetrations due to linearization and numerical error, despite position-level stabilization [Anitescu and Hart 2004; Baumgarte 1972; Cline and Pai 2003]. To address this, it might be necessary to allow an extra margin for collision handling, or to further apply post-stabilization. The position-level error is also related to the volume loss of fluids. While we use local position-corrections for particles to preserve a uniform distribution [Macklin and Müller 2013], a global approach would be more effective [Kugelstadt et al. 2019].

6.6.3 Non-Newtonian fluids. Our extension to non-Newtonian fluids is quite effective in practice. However, depending on the parameters used, the number of required iterations could increase and drive up the total cost, given the expense of each unified solve. In addition, since nonlinearity can lead to multiple solutions, an implausible viscosity distribution could be obtained, although the behaviors remain stable due to our staggered projections approach.

6.6.4 Convergence. In some of our experiments with viscous fluids, the number of required iterations fluctuates, occasionally leading to the solver failing to fully converge within the specified maximum iteration count. This is due to ill-conditioning, and the relatively large number of DOFs in the system. Although we employed an SOR preconditioner due to its practical effectiveness and parallel nature, it would be beneficial to develop a more advanced preconditioner for our problem, e.g., multigrid [Aanjaneya 2018; Aanjaneya et al. 2019; Lai et al. 2020]. Likewise, it may be worth exploring whether efficient sparse direct QP solvers, such as the recently proposed NASOQ scheme [Cheshmi et al. 2020], could be helpful in circumventing conditioning issues with iterative solvers.

7 CONCLUSIONS AND FUTURE WORK

We have proposed Monolith, a monolithic pressure-viscosity-contact solver that simultaneously handles divergence-free fluid constraints, implicit viscosity integration, frictional contacts between rigid bodies, and mutual interactions of liquids and rigid bodies governed by pressure and viscosity. Our method is variational in nature, naturally enforcing the appropriate boundary conditions and leading to a single unified optimization problem. We addressed the unified problem via a single LCP solve in the absence of friction, and via staggered projections in the presence of friction. We also proposed a new iterative approach to handle non-Newtonian fluids, which can be seamlessly integrated with our treatment of frictional contacts. We demonstrated the efficacy of our monolithic solver in a wide range of scenarios, eliminating artifacts of prior schemes and achieving at least one order of magnitude faster performance than comparable quality alternatives for both inviscid and viscous liquids.

As technical artists and other end-users continue to demand ever greater flexibility and physical realism from their tools, the challenges of integrated multiphysics simulation, such as those we have studied, will become increasingly salient. We believe that our monolithic, contact-aware coupling strategy can be extended to many different types of fluid and solid materials, e.g., deformable solids, cloth, liquid with surface tension, and viscoelastic materials. Likewise, scalability in the face of diverse interacting systems remains a hurdle; in addition to improved numerical methods, we would like

to explore spatial adaptivity and GPU acceleration techniques to further improve the performance of our framework.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable suggestions and comments. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (Grant RGPIN-04360-2014).

REFERENCES

Mridul Aanjaneya. 2018. An Efficient Solver for Two-way Coupling Rigid Bodies with Incompressible Flow. *Computer Graphics Forum* 37, 8 (2018), 59–68.

Mridul Aanjaneya, Chengguizi Han, Ryan Goldade, and Christopher Batty. 2019. An Efficient Geometric Multigrid Solver for Viscous Liquids. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 14 (July 2019), 21 pages.

Muzaffer Akbay, Nicholas Nobles, Victor Zordan, and Tamar Shinar. 2018. An extended partitioned method for conservative solid-fluid coupling. *ACM Transactions on Graphics* 37 (2018), 1–12.

Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile Rigid-fluid Coupling for Incompressible SPH. *ACM Transactions on Graphics* 31, 4, Article 62 (2012), 62:1–62:8 pages.

Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2015. A stream function solver for liquid simulations. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

Mihai Anitescu and Gary D. Hart. 2004. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. *Internat. J. Numer. Methods Engrg.* 60, 14 (2004), 2335–2371.

Uri M. Ascher and Linda R. Petzold. 1998. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations* (1st ed.). Society for Industrial and Applied Mathematics, USA.

Stefan Band, Christoph Gissler, Markus Ihmsen, Jens Cornelis, Andreas Peer, and Matthias Teschner. 2018a. Pressure Boundaries for Implicit Incompressible SPH. *ACM Trans. Graph.* 37, 2, Article 14 (2018).

Stefan Band, Christoph Gissler, Andreas Peer, and Matthias Teschner. 2018b. MLS pressure boundaries for divergence-free and viscous SPH fluids. *Computers & Graphics* 76 (2018), 37 – 46.

Adam W. Bargteil, Chris Wojtan, Jessica K. Hodgins, and Greg Turk. 2007. A Finite Element Method for Animating Large Viscoplastic Flow. *ACM Transactions on Graphics* 26, 3, Article 16 (2007).

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A Fast Variational Framework for Accurate Solid-fluid Coupling. *ACM Trans. Graph.* 26, 3, Article 100 (2007).

Christopher Batty and Robert Bridson. 2008. Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 219–228.

J. Baumgarte. 1972. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering* 1, 1 (1972), 1 – 16.

Markus Becker, Hendrik Tessendorf, and Matthias Teschner. 2009. Direct Forcing for Lagrangian Rigid-Fluid Coupling. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (2009), 493–503.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, USA.

Christopher Brandt, Leonardo Scandolo, Elmar Eisemann, and Klaus Hildebrandt. 2019. The reduced immersed method for real-time fluid-elastic solid interaction and contact simulation. *ACM Transactions on Graphics* 38 (2019).

Robert Bridson. 2015. *Fluid Simulation for Computer Graphics*. A K Peters/CRC Press.

Mark Carlson, Peter Mucha, and Greg Turk. 2004. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid. *ACM Transactions on Graphics* 23 (2004).

Erin Catto. 2020. Box2D. https://box2d.org/

Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman. 2017. Dynamics-Aware Numerical Coarsening for Fabrication Design. *ACM Trans. Graph.* 36, 4, Article 84 (2017).

Nuttapong Chentanez, Tolga G. Goktekin, Bryan E. Feldman, and James F. O'Brien. 2006. Simultaneous Coupling of Fluids and Deformable Bodies. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 83–89.

Kazem Cheshmi, Danny M. Kaufman, Shoaib Kamil, and Maryam Mehri Dehnavi. 2020. NASOQ: Numerically Accurate Sparsity-Oriented QP Solver. *ACM Trans. Graph.* 39, 4, Article 96 (2020).

M. B. Cline and D. K. Pai. 2003. Post-stabilization for rigid body simulation with contact and constraints. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, Vol. 3. 3744–3751 vol.3.

Erwin Coumans. 2020. Bullet physics library. http://bulletphysics.org/

Gilles Daviet and Florence Bertails-Descoubes. 2016. A Semi-implicit Material Point Method for the Continuum Simulation of Granular Materials. *ACM Trans. Graph.* 35, 4, Article 102 (2016), 102:1–102:13 pages.

O. Ding and C. Schroeder. 2020. Penalty Force for Coupling Materials with Coulomb Friction. *IEEE Transactions on Visualization and Computer Graphics* 26, 7 (2020), 2443–2455.

Zdenek Dostal and Joachim Schoberl. 2005. Minimizing Quadratic Functions Subject to Bound Constraints with the Rate of Convergence and Finite Termination. *Computational Optimization and Applications* 30, 1 (2005), 23–43.

Zdenek Dostl. 2009. *Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities* (1st ed.). Springer Publishing Company, Incorporated.

Michael Ferris and Todd Munson. 2000. Complementarity Problems in GAMS and the Path Solver. *Journal of Economic Dynamics and Control* 24 (2000), 165–188.

Dan Gerszewski and Adam W. Bargteil. 2013. Physics-based Animation of Large-scale Splashing Liquids. *ACM Transactions on Graphics* 32, 6, Article 185 (2013), 6 pages.

Frederic Gibou, Ronald P. Fedkiw, Li-Tien Cheng, and Myungjoo Kang. 2002. A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains. *J. Comput. Phys.* 176, 1 (2002), 205 – 227.

Christoph Gissler, Andreas Peer, Stefan Band, Jan Bender, and Matthias Teschner. 2019. Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling. *ACM Transactions on Graphics* 38 (2019), 1–13.

Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. 2005. Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph.* 24 (2005), 973–981.

Qi Guo, Xuchen Han, Chuyuan Fu, Theodore Gast, Rasmus Tamstorf, and Joseph Teran. 2018. A Material Point Method for Thin Shells with Frictional Contact. *ACM Trans. Graph.* 37, 4, Article 147 (2018).

Xuchen Han, Theodore F. Gast, Qi Guo, Stephanie Wang, Chenfanfu Jiang, and Joseph Teran. 2019. A Hybrid Material Point Method for Frictional Contact with Diverse Materials. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 17 (2019).

Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A Moving Least Squares Material Point Method with Displacement Discontinuity and Two-Way Rigid Body Coupling. *ACM Trans. Graph.* 37, 4, Article 150 (2018).

David A.B. Hyde and Ronald Fedkiw. 2019. A unified approach to monolithic solid-fluid coupling of sub-grid and more resolved solids. *J. Comput. Phys.* 390 (2019), 490 – 526.

Stefan Jeschke, Tomáš Skřivan, Matthias Müller-Fischer, Nuttapong Chentanez, Miles Macklin, and Chris Wojtan. 2018. Water Surface Wavelets. *ACM Trans. Graph.* 37, 4, Article 94 (2018).

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The Affine Particle-in-cell Method. *ACM Trans. Graph.* 34, 4, Article 51 (2015), 51:1–51:10 pages.

M. Danny Kaufman, Shinjiro Sueda, L. Doug James, and K. Dinesh Pai. 2008. Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.* (2008), 164–11.

Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-prager Elastoplasticity for Sand Animation. *ACM Trans. Graph.* 35, 4, Article 103 (2016), 103:1–103:12 pages.

Bryan Klingner, Bryan Feldman, Nuttapong Chentanez, and James O'Brien. 2006. Fluid animation with dynamic meshes. *ACM Trans. Graph.* 25 (2006), 820–825.

Dan Koschier and Jan Bender. 2017. Density Maps for Improved SPH Boundary Handling. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. Article 1, 10 pages.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2019. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. In *Eurographics 2019 - Tutorials*.

Tassilo Kugelstadt, Andreas Longva, Nils Thurey, and Jan Bender. 2019. Implicit Density Projection for Volume Conserving Liquids. *IEEE Transactions on Visualization and Computer Graphics* (2019), 1–1.

Junyu Lai, Yangang Chen, Yu Gu, Christopher Batty, and Justin W.L. Wan. 2020. Fast and Scalable Solvers for the Fluid Pressure Equations with Separating Solid Boundary Conditions. *Computer Graphics Forum* 39, 2 (2020), 23–33.

Egor Larionov, Christopher Batty, and Robert Bridson. 2017. Variational Stokes: A Unified Pressure-viscosity Solver for Accurate Viscous Liquids. *ACM Trans. Graph.* 36, 4, Article 101 (July 2017), 101:1–101:11 pages.

Michael Lentine, Jon Gretarsson, Craig Schroeder, Avi Robinson-Mosher, and Ronald Fedkiw. 2011. Creature Control in a Fluid Environment. *IEEE Transactions on Visualization and Computer Graphics* 17 (2011), 682 – 693.

J. E. Lloyd. 2005. Fast Implementation of Lemke's Algorithm for Rigid Body Contact Simulation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 4538–4543.

Libin Lu, Abtin Rahimian, and Denis Zorin. 2017. Contact-aware simulations of particulate Stokesian suspensions. *J. Comput. Phys.* 347 (2017), 160–182.

Miles Macklin and Matthias Müller. 2013. Position Based Fluids. *ACM Transactions on Graphics* 32, 4, Article 104 (2013), 5 pages.

Patrick Mullen, Keenan Crane, Dmitry Pavlov, Yiying Tong, and Mathieu Desbrun. 2009. Energy-preserving integrators for fluid animation. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 1–8.

Rahul Narain, Abhinav Golas, and Ming C. Lin. 2010. Free-flowing Granular Materials with Two-way Solid Coupling. *ACM Transactions on Graphics* 29, 6, Article 173 (2010), 10 pages.

Yen Ting Ng, Chohong Min, and Frédéric Gibou. 2009. An efficient fluid–solid coupling algorithm for single-phase flows. *J. Comput. Phys.* 228, 23 (2009), 8807 – 8829.

Miguel A Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus Gross. 2009. Implicit contact handling for deformable objects. In *Computer Graphics Forum*, Vol. 28. 559–568.

Saket Patkar, Mridul Aanjaneya, Wenlong Lu, Michael Lentine, and Ronald Fedkiw. 2016. Towards positivity preservation for monolithic two-way solid-fluid coupling. *J. Comput. Phys.* 312 (2016).

Charles Peskin . 2002. Peskin, C.S.: The immersed boundary method. Acta Numerica 11, 479-517. *Acta Numerica* 11 (01 2002), 479 – 517.

Avi Robinson-Mosher, R. Elliot English, and Ronald Fedkiw. 2009. Accurate Tangential Velocities for Solid Fluid Coupling. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '09)*. 227–236.

Avi Robinson-Mosher, Craig Schroeder, and Ronald Fedkiw. 2011. A Symmetric Positive Definite Formulation for Monolithic Fluid Structure Interaction. *J. Comput. Phys.* 230, 4 (Feb. 2011), 1547–1566.

Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. 2008. Two-way Coupling of Fluids to Rigid and Deformable Solids and Shells. *ACM Trans. Graph.* 27, 3, Article 46 (2008), 9 pages.

K. C. Sahu, P. Valluri, P. D. M. Spelt, and O. K. Matar. 2007. Linear instability of pressure-driven channel flow of a Newtonian and a Herschel-Bulkley fluid. *Physics of Fluids* 19, 12 (2007), 122101.

K. Schittkowski. 2005. *QL: A fortran code for convex quadratic programming - user's guide, version 2.11.* Technical Report.

Tamar Shinar, Craig Schroeder, and Ronald Fedkiw. 2008. Two-way coupling of rigid and deformable bodies. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 95–103.

Russell Smith. 2008. Open Dynamics Engine. http://www.ode.org/

David E. Stewart. 2000. Rigid-Body Dynamics with Friction and Impact. *SIAM Rev.* 42, 1 (2000), 3–39.

Tetsuya Takahashi and Ming C. Lin. 2019. A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling. *Computer Graphics Forum* 38, 2 (2019), 49–58.

Jie Tan, Yuting Gu, Greg Turk, and C. Liu. 2011. Articulated Swimming Creatures. *ACM Trans. Graph.* 30 (2011), 58.

Jie Tan, Kristin Siu, and C. Karen Liu. 2012. *Contact Handling for Articulated Rigid Bodies Using LCP.* Technical Report GIT-GVU-15-01-2. Georgia Institute of Technology, School of Interactive Computing.

Yun Teng, David I. W. Levin, and Theodore Kim. 2016. Eulerian Solid-Fluid Coupling. *ACM Trans. Graph.* 35, 6, Article 200 (2016), 8 pages.

Richard Tonge, Feodor Benevolenski, and Andrey Voroshilov. 2012. Mass Splitting for Jitter-Free Parallel Rigid Body Simulation. *ACM Transactions on Graphics* 31 (2012).

Jui-Hsien Wang, Rajsekhar Setaluri, Doug L. James, and Dinesh K. Pai. 2017. Bounce Maps: An Improved Restitution Model for Real-Time Rigid-Body Impact. *ACM Trans. Graph.* 36, 4, Article 150 (2017).

D. J. A. Welsh and M. B. Powell. 1967. An upper bound for the chromatic number of a graph and its application to timetabling problems. *Comput. J.* 10, 1 (01 1967), 85–86.

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, Article 160 (2015), 160:1–160:20 pages.

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.* 37, 6, Article 283 (2018), 19 pages.

Jae Yun and Yu-Du Han. 2002. Modified incomplete Cholesky factorization preconditioners for a symmetric positive definite matrix. *Bulletin of the Korean Mathematical Society* 39 (01 2002).

Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015. Codimensional non-Newtonian Fluids. *ACM Trans. Graph.* 34, 4, Article 115 (2015), 9 pages.

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, Article 16 (2010).