A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling

TIANYI XIE, University of California, Los Angeles, USA
MINCHEN LI, University of California, Los Angeles, USA
YIN YANG, University of Utah, USA
CHENFANFU JIANG, University of California, Los Angeles, USA

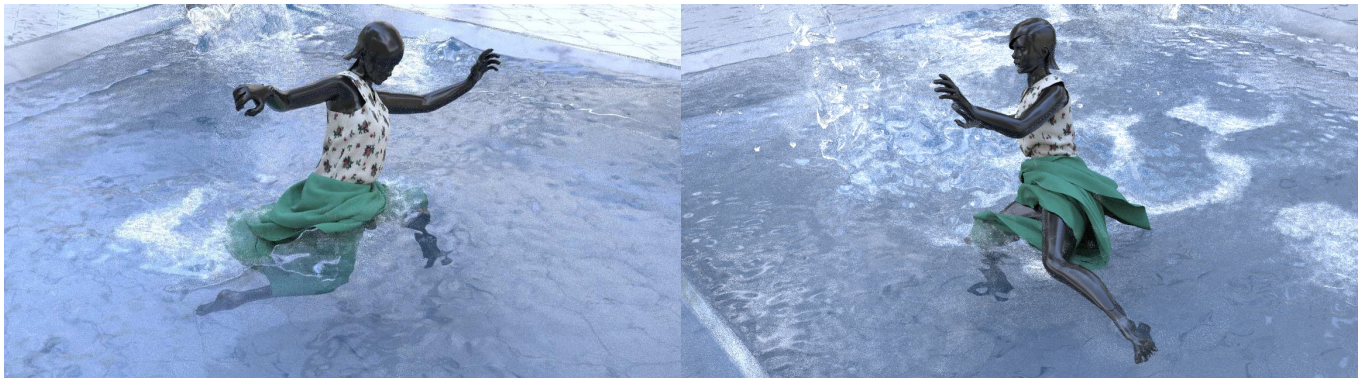


Fig. 1. **Kick water.** Our method accurately captures the complex interactions between the water, the multi-layer skirt, and the mannequin body without any interpenetration as the mannequin wearing the skirt kicks in a swimming pool and sends water flying.

We present a robust and efficient method for simulating Lagrangian solid-fluid coupling based on a new operator splitting strategy. We use variational formulations to approximate fluid properties and solid-fluid interactions, and introduce a unified two-way coupling formulation for SPH fluids and FEM solids using interior point barrier-based frictional contact. We split the resulting optimization problem into a fluid phase and a solid-coupling phase using a novel time-splitting approach with augmented *contact proxies*, and propose efficient custom linear solvers. Our technique accounts for fluids interaction with nonlinear hyperelastic objects of different geometries and codimensions, while maintaining an algorithmically guaranteed non-penetrating criterion. Comprehensive benchmarks and experiments demonstrate the efficacy of our method.

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

Additional Key Words and Phrases: Two-Way Coupling, Weakly Compressible Fluids, Elastic Solids, Time Splitting, Contact Proxy

ACM Reference Format:
Tianyi Xie, Minchen Li, Yin Yang, and Chenfanfu Jiang. 2023. A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling. *ACM Trans. Graph.* 42, 4, Article 000 (August 2023), 14 pages. https://doi.org/10.1145/3592115

Authors' addresses: Tianyi Xie, University of California, Los Angeles, USA, tianyixie77@g.ucla.edu; Minchen Li, University of California, Los Angeles, USA, minchernl@gmail.com; Yin Yang, University of Utah, USA, yin.yang@utah.edu; Chenfanfu Jiang, University of California, Los Angeles, USA, chenfanfu.jiang@gmail.com.

0730-0301/2023/8-ART000 $15.00
https://doi.org/10.1145/3592115

1 INTRODUCTION

The coupling of solids and fluids is common in nature but challenging to simulate. While solids are typically simulated using Lagrangian meshes, fluids are often discretized using Eulerian grids to accommodate topology changes. To accurately couple these distinct discretizations, sophisticated algorithms, such as the cut-cell method [Zarifi and Batty 2017], are often necessary. Unfortunately, they can be expensive and do not handle thin shells. Purely Eulerian [Teng et al. 2016; Valkov et al. 2015] or SPH [Akinci et al. 2012; Gissler et al. 2019] schemes have demonstrated successful two-way coupling by same-view discretization. They do not easily extend to nonlinear elastodynamics. Hybrid methods like MPM [Jiang et al. 2016] can simulate mixed materials, but can experience artificial stickiness unless resolved with more expensive schemes [Fang et al. 2020]. Furthermore, these methods do not ensure non-intersecting trajectories and often require additional correction procedures to handle accidentally penetrated fluids during advection.

We take the Lagrangian path and present a new method for coupling FEM solids and SPH fluids. By approximating solid, fluid, and interaction terms with potentials, we formulate two-way coupling as a monolithic optimization problem. Specifically, we draw inspiration from position-based fluids [Macklin and Müller 2013] and model weak incompressibility using a quadratic energy and a new *updated Lagrangian update rule* to track volume changes. We further symmetrize the discrete Laplacian-based viscosity and propose a *discrete quadratic potential* for better accuracy and robustness. We follow the Incremental Potential Contact [Li et al. 2020] model to enforce guaranteed separable boundary conditions and resolve frictional contacts at the interface.

The proposed formulation achieves strong coupling, but can be exceedingly inefficient when solved with Newton's method due to

the huge and dense Hessian of the fluid part, which is a result of the need for many particle neighbors for an accurate SPH discretization. This causes a significant computing bottleneck.

To tackle this issue, we propose a robust *proxy contact energy* formulation, splitting the time integration into a fluid phase and a solid-coupling phase. The fluid phase requires only one Newton iteration per time step, resulting in increased efficiency with nonlinear optimization occurring only during the solid-coupling phase. One of the key advantages of our quadratic proxy is its ability to effectively resolve instability caused by time splitting. This is achieved through its asymptotic approximation to the solid-fluid contact force. Additionally, time integration is maintained consistent through the cancellation of the proxy's contribution in the solid-coupling phase, resulting in only a small splitting error. Finally, we design a matrix-free conjugate gradient solver and a domain-decomposed solver to further enhance the computational efficiency.

In summary, our contributions include

- a unified penetration-free two-way coupling framework for weakly compressible SPH fluids and nonlinear elastic FEM solids in arbitrary codimensions;
- consistently modeled incompressibility and viscosity potentials for SPH fluids, incorporating advantages of Updated Lagrangian (UL) kinematics;
- a robust time splitting scheme with contact proxies that enables separate time integrations of solids and fluids;
- a matrix-free conjugate gradient (CG) solver and a Schur complement based domain-decomposed solver for efficiently solving the linear systems.

2 RELATED WORK

Many popular fluid solvers use Eulerian grids and a series of existing works focus on coupling Eulerian fluids with Lagrangian solids by resolving interactions between the grids and irregular mesh boundaries. The ghost fluid method [Fedkiw 2002; Fedkiw et al. 1999] was proposed to additionally discretize the Eulerian/Lagrangian interface. Early works considered weak coupling [Guendelman et al. 2005], which advances the solids and fluids alternatively. Strong coupling [Klingner et al. 2006] on the other hand solves a monolithic system and is often more robust. Chentanez et al. [2006] proposed a two-way coupling method between fluids and deformable solids by solving a combined system of fluid-cell pressure and solid-node velocity. The cut-cell method [Roble et al. 2005] is another widely used solution, often through the usage of virtual nodes. Batty et al. [2007] proposed a variational framework to strongly couple fluids and rigid bodies by casting the pressure solve into minimization. Subsequent extensions support deformable objects and thin shells [Robinson-Mosher et al. 2011, 2008], where elastic forces are explicitly applied and the coupling step is implicit. Assuming corotated linear elasticity, Zarifi and Batty [2017] incorporated the implicit solid dynamics into pressure projection, obtaining a symmetric positive-definite system. Later works also explored rigid-rigid [Takahashi and Batty 2020], rigid-fluid [Takahashi and Batty 2021], and solid-fluid [Takahashi and Batty 2022] coupling considering frictional contacts. Eulerian solids [Levin et al. 2011] were also explored, where coupling can be conveniently achieved in a purely Eulerian fashion [Teng et al. 2016; Valkov et al. 2015]. However they face challenges in numerical dissipation, volume conservation, and handling structures thinner than a grid cell. More recently, Brandt et al. [2019] built upon the immersed boundary method [Peskin 2002] and proposed a reduced solver to simulate real-time coupling, focusing on incompressible elastic materials and no-slip boundary conditions.

Fluids can also be directly modeled with Lagrangian meshes [Batty et al. 2012; Clausen et al. 2013; Klingner et al. 2006; Wang et al. 2020], enabling explicit coupling with solids. However, remeshing tends to become a bottleneck. Using particles is another popular strategy. SPH [Koschier et al. 2022] uses spatial sampling to approximate continuous functions and captures compelling fluid dynamics. Pioneering works [Becker and Teschner 2007; Monaghan 1994] used the Equation of State (EOS) for weakly compressible fluids, where the pressure is proportional to the density deviation. An explicit formulation may strictly restrict time step sizes. Incompressibility has also been enforced by solving a Pressure Poisson Equation (PPE) [Bender and Koschier 2015; Ihmsen et al. 2013; Solenthaler and Pajarola 2009]. This approach seeks to cancel out density or velocity divergence deviations caused by non-pressure forces through the use of pressure accelerations. SPH boundary handling techniques have been developed to prevent penetrations of fluid particles near solid boundaries [Becker and Teschner 2007; Becker et al. 2009b; Ihmsen et al. 2010]. One such method, proposed by Akinci et al. [2012], uses a single layer of boundary samples and has been applied to the coupling of fluids with both rigid bodies and elastic solids [Akinci et al. 2013]. Gissler et al. [2019] proposed a global formulation that unifies rigid body and fluid dynamics, in which the fluid pressure solver is linked to a second artificial pressure solver for rigid body particles. Koschier and Bender [2017] introduced an alternative method using density maps to represent dynamic rigid boundaries, eliminating the need for boundary particles. Bender et al. [2019] proposed using the volume contribution of boundary geometry to compute boundary forces, which reduces the cost of precomputation but cannot be applied to deformable bodies.

Solenthaler et al. [2007] used SPH to approximate the deformation gradient of linear elastic materials, but the resulting gradient is not rotation invariant. Becker et al. [2009a] addressed this issue by using shape matching to determine orientation and calculating forces in a rotated configuration. Peer et al. [2018] proposed an implicit scheme and applied kernel gradient correction [Bonet and Lok 1999] to obtain a first-order consistent SPH formulation for the deformation gradient. Incorporating solid particles into the preexisting fluid pressure solver can resolve contact handling, but SPH still faces numerical issues such as the zero-mode [Ganzenmüller 2015; Kugelstadt et al. 2021] when simulating elastic objects. Additionally, the pressure solver will treat solid objects as incompressible under compression, which may not be applicable in all cases.

The Material-Point Method (MPM) [Jiang et al. 2016; Sulsky et al. 1995] combines Lagrangian and Eulerian representations to capture solid-fluid coupling [Fei et al. 2018; Stomakhin et al. 2014; Yan et al. 2018] and mixture [Gao et al. 2018; Tampubolon et al. 2017]. Fang et al. [2020] proposed a free-slip treatment, but did not consider separation. Recently, a FEM-MPM coupling method based on a variational barrier formulation [Li et al. 2020] has been proposed for coupling frictional and separable elastic materials [Li et al. 2022].

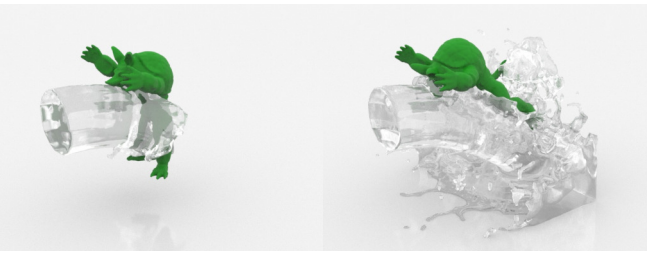


Fig. 2. **Shoot armadillo** with a high-speed water jet.

Our approach for solid-fluid coupling is inspired by this method and uses a similar purely Lagrangian framework.

3 FORMULATION

Here we derive a time integrator for a coupled system of solids and fluids by starting with the governing equations and then performing discretization. Subscripts s and f represent solid and fluid quantities.

3.1 Governing Equations

The governing equations for the coupled system are

$$\rho_s \frac{D\mathbf{v}_s}{Dt} = \nabla \cdot \sigma + \rho_s \mathbf{g} + \mathbf{f}_{s\to s} + \mathbf{f}_{f\to s}, \tag{1}$$

$$\rho_f \frac{D\mathbf{v}_f}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v}_f + \rho_f \mathbf{g} - \mathbf{f}_{f\to s}, \tag{2}$$

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

where ρ is density, $\mathbf{g}$ is gravity, $\mathbf{f}_{s\to s}$ is the self-contact force of solids, $\mathbf{f}_{f\to s}$ is the contact force exerted by fluids, σ is the Cauchy stress of solids, p is the fluid pressure, and μ is the dynamic viscosity [Bridson 2015].

At the interface between solids and fluids, we enforce the separable boundary condition

$$0 \le (\mathbf{v}_s - \mathbf{v}_f) \cdot \mathbf{n}_f \perp (\mathbf{f}_{f\to s} \cdot \mathbf{n}_f) \ge 0 \tag{4}$$

to prevent penetration while allowing separation [Batty et al. 2007], where $\mathbf{n}_f$ is the outward pointing normal of the fluid surface. This condition helps determine the normal component of $\mathbf{f}_{f\to s}$. For the tangential component (friction), let $\mathbf{u} = (\mathbf{I} - \mathbf{n}_f \otimes \mathbf{n}_f)(\mathbf{v}_s - \mathbf{v}_f)$ be the tangential relative velocity, we have

$$\begin{aligned} (\mathbf{I} - \mathbf{n}_f \otimes \mathbf{n}_f)\mathbf{f}_{f\to s} &= \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta} \cdot \mathbf{u} \\ \text{s.t.} \quad \|\boldsymbol{\beta}\| &\le \mu_t \mathbf{f}_{f\to s} \cdot \mathbf{n}_f \ \text{ and } \ \boldsymbol{\beta} \cdot \mathbf{n}_f = 0 \end{aligned} \tag{5}$$

following the Maximum Dissipation Principle [Moreau 2011], where μ_t is the friction coefficient. We enforce exact mass conservation by adopting Lagrangian methods to discretize both domains.

3.2 Solid Domain

We focus on nonlinear hyperelastic solids, where the elastic force is the negative gradient of an elastic potential. After discretizing the solid domain Ω_s as Lagrangian linear finite elements (triangles in 2D and tetrahedra in 3D), the total elastic potential is a piecewise constant summation of an elastic energy density function $\psi_s(\mathbf{F})$ (e.g. neo-Hookean) over the mesh domain: $\Psi_s(\mathbf{x}) = \sum_e V_e \psi_s(\mathbf{F}_e)$, where V_e is the rest volume of tetrahedron e, and $\mathbf{F} = \frac{\partial \mathbf{x}(\mathbf{X},t)}{\partial \mathbf{X}}$ is the deformation gradient with $\mathbf{X}$ and $\mathbf{x}$ the material and world space coordinates respectively [Sifakis and Barbic 2012]. For $\mathbf{f}_{s\to s}$, we follow Li et al. [2020]'s smooth barrier approach that guarantees non-penetration. We leave the discussion of $\mathbf{f}_{f\to s}$ to § 3.4.

3.3 Fluid Domain

Following SPH literature [Becker and Teschner 2007; Bender and Koschier 2015; Ihmsen et al. 2013; Macklin and Müller 2013], we discretize the fluid domain Ω_f with Lagrangian particles. To integrate fluids with optimization-based time integration, we approximate both the pressure and viscosity forces as conservative forces. We verify in the supplementary [Anonymous 2023] that these proposed potential energies are both convex and quadratic.

3.3.1 Incompressibility Potential. Pressure forces help preserve the volume of incompressible fluids. We thus model the incompressibility via a quadratic energy density function $\psi_{f,I}(J) = \frac{k_I}{2}(J-1)^2$ that penalizes the deviation of volume ratio $J = \rho_0/\rho$ from 1, where ρ_0 is the initial density. The use of a large stiffness value (k_I) in a convergent solve results in negligible visual compression, eliminating the need for higher degree polynomials in nearly incompressible fluids [Hyde et al. 2020]. The incompressibility potential is obtained by integrating $\psi_{f,I}(J)$ over the fluid domain Ω_f^0 in *material space*:

$$P_I(\mathbf{x}) = \sum_i \frac{k_I}{2} V_0 (J_i(\mathbf{x}) - 1)^2, \tag{6}$$

where we assumed all fluid particles have equal rest volume V_0, and J_i denotes the volume ratio of the i-th particle as a function of $\mathbf{x}$.

Updated Lagrangian. SPH literature often relates ρ to $\mathbf{x}$ through density summation in the world space. To obtain a linear relation between J and $\mathbf{x}$ so that the incompressibility potential stays quadratic in terms of $\mathbf{x}$, we track J in an *updated Lagrangian* fashion. Treating Ω^n as an intermediate reference space and differentiating the deformation map between Ω^n and Ω^{n+1} results in an update rule

$$J_i^{n+1} = J_i^n (1 + h \nabla \cdot \mathbf{v}_i^{n+1}), \tag{7}$$

where J_i^n and $\nabla \cdot \mathbf{v}_i^{n+1}$ can be approximated as

$$J_i^n = \frac{\rho_0}{\sum_j m_j W_{ij}}, \ \nabla \cdot \mathbf{v}_i^{n+1} = \sum_j \frac{m_j}{\rho_j^n} (\mathbf{v}_j^{n+1} - \mathbf{v}_i^{n+1}) \cdot \nabla_i W_{ij} \tag{8}$$

via SPH, and $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j)$ is a kernel function (e.g. Cubic Spline kernel [Monaghan 1992, 2005] or Spiky kernel [Müller et al. 2003]). Here J_i^n denote the reinitialized volume ratio of the i-th fluid particle at the beginning of time step n. Such reinitialization avoids accumulated density and particle distribution errors commonly seen in other updated Lagrangian solvers like MPM.

3.3.2 Viscosity Potential. Modeling viscosity via strain rate tensors [Bender and Koschier 2016; Peer et al. 2015; Peer and Teschner 2016; Takahashi et al. 2015] is possible, but may suffer from artifacts at the surface due to particle deficiencies. We follow Monaghan [2005] to use the more robust velocity Laplacian [Weiler et al. 2018] and derive its energy form.

Fig. 3. **Cream** is stirred, causing spoon deformation.

Combining SPH 1st-order derivatives and finite differences, the viscosity force can be computed as

$$\mathbf{f}_i(\mathbf{x}) = \nu m_i \nabla^2 \mathbf{v}_i^{n+1} = 2\nu(d+2)\sum_j \frac{m_i m_j}{\rho_j}\frac{\nabla_i W_{ij}(\mathbf{x}_{ij}^n)^T}{\|\mathbf{x}_{ij}^n\|^2 + 0.01\hbar^2}\mathbf{v}_{ij}^{n+1},$$

where $\mathbf{x}_{ij}^n = \mathbf{x}_i^n - \mathbf{x}_j^n$, $\mathbf{v}_{ij}^{n+1} = \mathbf{v}_i^{n+1} - \mathbf{v}_j^{n+1}$, $\hbar$ is the support radius of the kernel, ν and $d \in \{2, 3\}$ denote the kinematic viscosity and spatial dimension respectively. Directly applying this force violates momentum conservation as the mutual interaction forces are not equal. Thus, we perform a further approximation

$$\mathbf{f}_i(\mathbf{x}) \approx 4\nu(d+2)\sum_j \frac{m_i m_j}{\rho_i + \rho_j}\frac{\nabla_i W_{ij}(\mathbf{x}_{ij}^n)^T}{\|\mathbf{x}_{ij}^n\|^2 + 0.01\hbar^2}\mathbf{v}_{ij}^{n+1} \tag{9}$$

to solve this issue and also make the force integrable. Let $\mathbf{V}_{ij} = 4(d+2)\frac{m_i m_j}{\rho_i+\rho_j}\frac{(-\nabla_i W_{ij})(\mathbf{x}_{ij}^n)^T}{\|\mathbf{x}_{ij}^n\|^2+0.01\hbar^2}$, we can now gather and integrate all viscosity forces and obtain a quadratic viscosity potential

$$P_V(\mathbf{x}) = \frac{1}{4}\nu\hat{h}\sum_i\sum_j \|\mathbf{v}_{ij}^{n+1}\|^2_{\mathbf{V}_{ij}}, \tag{10}$$

where $\hat{h}$ is a constant scalar related to the time integration scheme. For example, $\hat{h} = h$ for implicit Euler as $\mathbf{v}^{n+1} = (\mathbf{x}^{n+1} - \mathbf{x}^n)/h$.

3.4 Coupling

3.4.1 Barrier Potential for Non-penetration. To couple the solid domain Ω_s with the fluid domain Ω_f, we use the separable boundary condition (Eq. 4), which enforces non-interpenetration constraints between these two domains. To model these constraints, we first define a distance function

$$d(\partial\Omega_s^t, \mathbf{x}_f) = \min_{\mathbf{x}_s}\|\mathbf{x}_s - \mathbf{x}_f\|, \quad \mathbf{x}_s \in \partial\Omega_s^t,\ \mathbf{x}_f \in \Omega_f^t, \tag{11}$$

which measures the distance between $\mathbf{x}_f$, a point in the fluid domain, and the surface of the solid domain. Then the primal component of the constraints can be expressed as

$$d(\partial\Omega_s^t, \mathbf{x}_f) \geq 0, \quad \forall t \geq 0,\ \forall \mathbf{x}_f \in \Omega_f^t. \tag{12}$$

We then adopt the barrier formulation from Li et al. [2020] to model all the constraints in Eq. 4 between solids and fluids, and obtain a barrier potential

$$\int_{\partial\Omega_f^t} b(d(\partial\Omega_s^t, \mathbf{x}_f), \hat{d})d\mathbf{x}_f, \tag{13}$$

where the barrier energy density $b(d, \hat{d})$ is piecewise smooth and only activated when $d < \hat{d}$, improving efficiency and approximately satisfying the complementary slackness condition. As d approaches 0, the value of $b(d, \hat{d})$ monotonically increases to infinity, providing arbitrarily large repulsion to avoid interpenetration.

Since our solids and fluids domains are respectively discretized as meshes and particles, the barrier potential (Eq. 13) in 3D can be numerically integrated as

$$\begin{aligned} B_{\text{sf}}(\mathbf{x}_s, \mathbf{x}_f) &= \sum_{q\in\mathcal{Q}_f} s_q b(\min_{e\in\mathcal{B}_s} d^{PT}(\mathbf{x}_q, e), \hat{d}) \\ &= \sum_{q\in\mathcal{Q}_f} s_q \max_{e\in\mathcal{B}_s} b(d^{PT}(\mathbf{x}_q, e), \hat{d}), \end{aligned} \tag{14}$$

where $\mathcal{Q}_f$ is the set of all SPH fluid particles, $\mathcal{B}_s$ is the set of all boundary triangles of the solids, $d^{PT}(\mathbf{x}_q, e)$ measures the distance between particle $\mathbf{x}_q$ and triangle e, and s_q is set to $2\left(\frac{J_q^n V_0}{\pi}\right)^{\frac{1}{2}}$ in 2D and $\pi\left(\frac{3J_q^n V_0}{4\pi}\right)^{\frac{2}{3}}$ in 3D, which is the integration weight (boundary area) of each fluid particle,. Here the min-max transformation is based on the non-ascending property of the barrier function. However, the max operator here makes the barrier potential difficult to optimize efficiently with gradient-based methods. Fortunately, due to the local support of the barrier function $b(d, \hat{d})$ as $\hat{d}$ is small, we can simply approximate the barrier potential as

$$B_{\text{sf}}(\mathbf{x}_s, \mathbf{x}_f) = \sum_{q\in\mathcal{Q}_f}\sum_{e\in\mathcal{B}_s} s_q b(d^{PT}(\mathbf{x}_q, e), \hat{d}), \tag{15}$$

which may result in overestimated contact forces near the edges and nodes on the mesh boundary, but we have not observed any artifacts in our experiments.

3.4.2 Friction Potential. Following Li et al. [2020], we model the local friction forces $\mathbf{f}_k$ for every active solid-fluid contact pair k. Formally, the friction force is defined as

$$\mathbf{f}_k(\mathbf{x}_s, \mathbf{x}_f) = -\mu_t\lambda_k T_k(\mathbf{x}_s, \mathbf{x}_f) f_1(\|\mathbf{u}_k\|)\frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}, \tag{16}$$

where λ_k is the contact force magnitude, $T_k(\mathbf{x}_s, \mathbf{x}_f) \in \mathbb{R}^{3n\times 2}$ is the consistently oriented sliding basis, and $\mathbf{u}_k$ is the relative sliding displacement, which can be computed as $\mathbf{u}_k = T_k(\mathbf{x}_s, \mathbf{x}_f)^T([\mathbf{x}_s^T, \mathbf{x}_f^T]^T - [(\mathbf{x}_s^n)^T, (\mathbf{x}_f^n)^T]^T)$. Here f_1 is a smoothly approximated function designed for the smooth transition between sticking and sliding modes. To make this friction formulation fit into optimization time integration, Li et al. [2020] further approximated the sliding basis $T(\mathbf{x}_s, \mathbf{x}_f)$ and contact force $\lambda_k(\mathbf{x}_s, \mathbf{x}_f)$ explicitly as $T(\mathbf{x}_s^n, \mathbf{x}_f^n)$ and $\lambda_k(\mathbf{x}_s^n, \mathbf{x}_f^n)$. Then the semi-implicit friction force is integrable with the friction

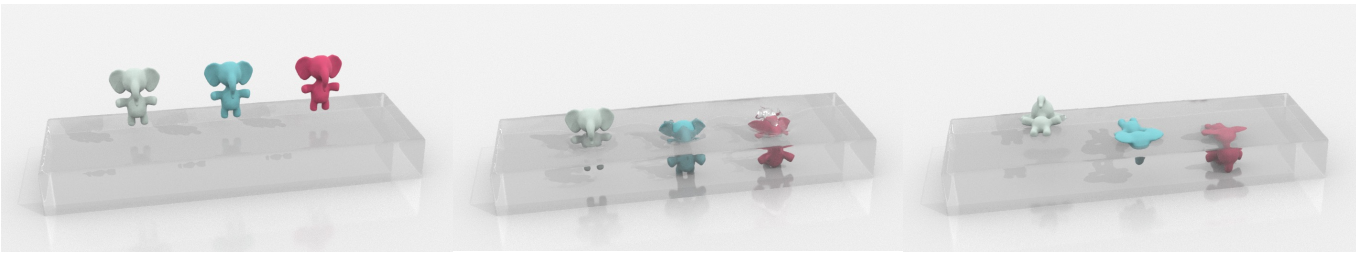


Fig. 4. **Buoyancy**. Three elastic elephants with different densities (from left to right: 200, 700, and 1200 kg/m^3) fall into the water, demonstrating buoyancy.

potential computed as

$$D_{\text{sf}}(\mathbf{x}_s, \mathbf{x}_f) = \sum_{k \in \mathcal{A}^n} \mu_t \lambda_k^n f_0(\|\mathbf{u}_k\|), \tag{17}$$

where f_0 is defined by the relation $f_0' = f_1$ and $\mathcal{A}^n$ is the set containing all activate particle-triangle contact pairs at the previous time step n.

3.5 Optimization Time Integrator

With the above potential energies modeling all the solid and fluid forces, now we can build a unified two-way solid-fluid coupling framework. By stacking all nodal positions and velocities of SPH particles and FEM nodes as $\mathbf{x} = [\mathbf{x}_f^T, \mathbf{x}_s^T]^T$ and $\mathbf{v} = [\mathbf{v}_f^T, \mathbf{v}_s^T]^T$, we define $\Psi(\mathbf{x}) = \Psi_s(\mathbf{x}_s)$, $P(\mathbf{x}) = P_I(\mathbf{x}_f) + P_V(\mathbf{x}_f)$ and $C_{\text{sf}}(\mathbf{x}) = B_{\text{sf}}(\mathbf{x}_s, \mathbf{x}_f) + D_{\text{sf}}(\mathbf{x}_s, \mathbf{x}_f)$. Combined with the solid-solid contact potential $C_{\text{ss}}(\mathbf{x})$ from IPC, our solid-fluid coupling problem can be solved in a monolithic manner applying implicit Euler time integration

$$\begin{cases} \mathbf{v}^{n+1} = \mathbf{v}^n + h\mathbf{M}^{-1}(\mathbf{f}_{\text{ext}} - \nabla P(\mathbf{x}^{n+1}) - \nabla\Psi(\mathbf{x}^{n+1}) - \nabla C(\mathbf{x}^{n+1})) \\ \mathbf{x}^{n+1} = \mathbf{x}^n + h\mathbf{v}^{n+1} \end{cases}, \tag{18}$$

which is equivalent to

$$\mathbf{x}^{n+1} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}^n\|_{\mathbf{M}}^2 + h^2(P(\mathbf{x}) + \Psi(\mathbf{x}) + C(\mathbf{x})) \tag{19}$$

with the mass matrix $\mathbf{M}$, time step size h, the predictive position $\hat{\mathbf{x}}^n = \mathbf{x}^n + h\mathbf{v}^n + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}$ and the total contact potential $C(\mathbf{x}) = C_{\text{sf}}(\mathbf{x}) + C_{\text{ss}}(\mathbf{x})$.

4 EFFICIENT SOLVER

A straightforward way to robustly solve the time-stepping optimization problem (Eq. 19) is to apply the projected Newton's method with line search [Li et al. 2020]. At every iteration, the search direction $\mathbf{p}$ can be computed by solving the linear system

$$\begin{bmatrix} \mathbf{H}_f & \mathbf{G} \\ \mathbf{G}^T & \mathbf{H}_s \end{bmatrix} \mathbf{p} = \begin{bmatrix} \mathbf{g}_f \\ \mathbf{g}_s \end{bmatrix}. \tag{20}$$

Here $\mathbf{H}_f$ and $\mathbf{H}_s$ are the (projected) Hessian matrices w.r.t. the position of fluids and solids respectively, and $\mathbf{G} = \frac{\partial^2 E}{\partial \mathbf{x}_f \partial \mathbf{x}_s}$ denotes the coupling submatrix. Nevertheless, solving this linear system can be a severe bottleneck in practice. One reason is that SPH techniques need sufficient neighbors to accurately approximate physical quantities, which results in a much larger and denser fluid Hessian matrix $\mathbf{H}_f$ compared to the solid one. In addition, the optimization may require many iterations to converge due to the sharpness of barrier energy, especially in contact-rich cases.

Since our fluid energies are all quadratic, we separate them from the highly nonlinear solids and contact energies via a robust time splitting scheme (§ 4.1) so that the fluid part can be solved within a single Newton iteration per time step. We then propose efficient methods to solve the domain-decomposed linear systems (§ 4.2).

4.1 Time Splitting

4.1.1 Baseline Time Splitting. Intuitively, we can split the original time integration into a **fluid phase**

$$\begin{cases} \widetilde{\mathbf{v}}_f = \mathbf{v}_f^n + h\mathbf{M}_f^{-1}(-\nabla_f P([(\widetilde{\mathbf{x}}_f)^T, (\mathbf{x}_s^n)^T]^T) + \mathbf{f}_f) \\ \widetilde{\mathbf{x}}_f = \mathbf{x}_f^n + h\widetilde{\mathbf{v}}_f \end{cases} \tag{21}$$

and a **solid-coupling phase**

$$\begin{cases} \mathbf{v}^{n+1} = \begin{bmatrix} \widetilde{\mathbf{v}}_f \\ \mathbf{v}_s^n \end{bmatrix} + h\mathbf{M}^{-1}(-\nabla\Psi(\mathbf{x}^{n+1}) - \nabla C(\mathbf{x}^{n+1}) + \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_s \end{bmatrix}) \\ \mathbf{x}^{n+1} = \mathbf{x}^n + h\mathbf{v}^{n+1} \end{cases}, \tag{22}$$

where $\mathbf{f}_f$ and $\mathbf{f}_s$ are the external forces on the fluids and the solids respectively. In the fluid phase, we solve for an intermediate state for the fluid particles in a single Newton's iteration, ignoring contact. Then the highly nonlinear barrier force is resolved in the solid-coupling phase along with elasticity, where the fluid Hessian $\mathbf{H}_f$ reduces to a block-diagonal matrix $\frac{\partial^2 C(\mathbf{x})}{\partial \mathbf{x}_f^2}$. In this setting, nonlinear optimization only happens for fluid boundaries and solid DOFs in the solid-coupling phase. The details of this *Baseline Time Splitting Scheme* can be found in the supplementary material [Anonymous 2023] .

Although this baseline splitting strategy indeed brings a significant performance gain, severe instabilities can happen at the solid-fluid interface if the time step size is not sufficiently small, especially when simulating viscous fluids (Fig. 7). For example, fluid particles may stick to the solid boundaries. This is an artifact also seen in existing SPH fluid solvers, and it is typically addressed by sampling particles at solid boundaries to exert boundary pressures [Akinci et al. 2012; Becker et al. 2009b; Ihmsen et al. 2010]. In light of this, we consistently augment the fluid phase with proxy forces for solid-fluid contact to improve stability while avoiding any particle sampling overhead.

4.1.2 Time Splitting with Contact Proxy. We introduce a solid-fluid contact proxy energy $\hat{C}_{\text{sf}}(\mathbf{x})$ into the fluid phase to efficiently exert

Fig. 5. **Bob** simulated with (A) Joint Optimization, (B) Time Splitting with Contact Proxy, and (C) Baseline Time Splitting. For this example, baseline time splitting can also produce visually plausible results, and our proxy-assisted scheme is 3× faster than joint optimization.

approximated interaction forces between the boundaries of solids and fluids. In the following discussions, we will also write contact energy $C(\mathbf{x})$ as the sum of the solid-fluid part ($C_{\text{sf}}(\mathbf{x})$) and the solid-solid part ($C_{\text{ss}}(\mathbf{x})$) for clarity. To ensure consistency with the original PDE, we cancel the contribution of this contact proxy in the solid-coupling phase. The resulting time integration becomes

$$\begin{cases} \widetilde{\mathbf{v}} = \mathbf{v}^n + h\mathbf{M}^{-1}(-\nabla P(\widetilde{\mathbf{x}}) + \mathbf{f}_{\text{ext}} - \nabla \hat{C}_{\text{sf}}(\widetilde{\mathbf{x}})) \\ \widetilde{\mathbf{x}} = \mathbf{x}^n + h\widetilde{\mathbf{v}} \end{cases}$$
$$\begin{cases} \mathbf{v}^{n+1} = \widetilde{\mathbf{v}} + h\mathbf{M}^{-1}(-\nabla \Psi(\mathbf{x}^{n+1}) - \nabla C(\mathbf{x}^{n+1}) + \nabla \hat{C}_{\text{sf}}(\mathbf{x}^{n+1})) \\ \mathbf{x}^{n+1} = \mathbf{x}^n + h\mathbf{v}^{n+1} \end{cases} \tag{23}$$

where the fluid phase now also implicitly updates the solid boundary nodes near the fluids to an intermediate state. For $\hat{C}_{\text{sf}}(\mathbf{x})$, a straightforward choice is $\hat{C}_{\text{sf}}(\mathbf{x}) = \gamma C_{\text{sf}}(\mathbf{x})$, where $\gamma \in (0, 1)$ is a user-defined constant coefficient. For simplicity, we choose $\hat{C}_{\text{sf}}(\mathbf{x}) = \frac{1}{2}C_{\text{sf}}(\mathbf{x})$ in all our simulations. But to ensure our fluid phase still only contains linear forces, we apply the 2nd-order Taylor expansion of $\frac{1}{2}C_{\text{sf}}(\mathbf{x})$ at $\mathbf{x}^n$ for the approximation in the fluid phase, i.e.

$$\hat{C}_{\text{sf}}(\mathbf{x}) = \frac{1}{2}\left(C_{\text{sf}}(\mathbf{x}^n) + \nabla C_{\text{sf}}(\mathbf{x}^n)(\mathbf{x} - \mathbf{x}^n) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}^n\|^2_{\nabla^2 C_{\text{sf}}(\mathbf{x}^n)}\right), \tag{24}$$

while in the solid-coupling phase, we simply use $\hat{C}_{\text{sf}}(\mathbf{x}) = \frac{1}{2}C_{\text{sf}}(\mathbf{x})$. In the supplementary [Anonymous 2023] , we prove that our time splitting scheme with contact proxy only has an $O(h^4)$ mismatch compared to implicit Euler solution. Reformulating both phases (Eq. 23) as optimization problems, we obtain

$$\begin{aligned} \widetilde{\mathbf{x}} &= \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}^n\|^2_{\mathbf{M}} + h^2(P(\mathbf{x}) + \hat{C}_{\text{sf}}(\mathbf{x})), \\ \mathbf{x}^{n+1} &= \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \widetilde{\mathbf{x}}\|^2_{\mathbf{M}} + h^2(\Psi(\mathbf{x}) + \frac{1}{2}C_{\text{sf}}(\mathbf{x}) + C_{\text{ss}}(\mathbf{x})), \end{aligned} \tag{25}$$

where $\hat{\mathbf{x}}^n = \mathbf{x}^n + h\mathbf{v}^n + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}$ with $\mathbf{f}_{\text{ext}} = [\mathbf{f}_f^T, \mathbf{f}_s^T]^T$.

In addition to avoiding fluid particle sticking issues without extra expensive costs, another benefit of our method is that it helps reduce the number of Newton's iterations for solving the problem. Typically, the barrier method takes many Newton iterations when resolving high-speed impacts. With our scheme, when high-speed fluid particles are colliding with a deformable object, their speed will be significantly reduced after the fluid phase due to the contact proxy. The reduced speed will then be taken into the solid-coupling phase, which makes the nonlinear optimization easier to solve (by having less contact constraint set changes). The details of our proxy-based time splitting scheme can be found in Alg. 1.

Algorithm 1 Time Splitting with Contact Proxy

$\mathbf{x} \leftarrow \mathbf{x}^n$, $\hat{\mathbf{x}}^n \leftarrow \mathbf{x}^n + h\mathbf{v}^n + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}$
SPH Neighbor Search & Density Update
$\hat{C}_{\text{sf}}(\mathbf{x}) \leftarrow$ 2nd Taylor Expansion of $\frac{1}{2}C_{\text{sf}}(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^n$
// Fluid Phase
$\mathbf{H} \leftarrow h^2\left(\nabla^2 P(\mathbf{x}) + \nabla^2 \hat{C}_{\text{sf}}(\mathbf{x})\right) + \mathbf{M}$
$\mathbf{p} \leftarrow -\mathbf{H}^{-1}\left(h^2(\nabla P(\mathbf{x}) + \nabla \hat{C}_{\text{sf}}(\mathbf{x})) + \mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}^n)\right)$
$\widetilde{\mathbf{x}} \leftarrow \mathbf{x} + \mathbf{p}$
// Solid-Coupling Phase
do
  $\mathbf{H} \leftarrow h^2\left(\nabla^2\Psi(\mathbf{x}) + \frac{1}{2}\nabla^2 C_{\text{sf}}(\mathbf{x}) + \nabla^2 C_{\text{ss}}(\mathbf{x})\right) + \mathbf{M}$
  $\mathbf{g} \leftarrow h^2\left(\nabla\Psi(\mathbf{x}) + \frac{1}{2}\nabla C_{\text{sf}}(\mathbf{x}) + \nabla C_{\text{ss}}(\mathbf{x})\right) + \mathbf{M}(\mathbf{x} - \widetilde{\mathbf{x}})$
  $\mathbf{p} \leftarrow -\mathbf{H}^{-1}\mathbf{g}$
  $\alpha \leftarrow$ Backtracking Line Search with CCD
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{p}$
while $\frac{1}{h}\|\mathbf{p}\| > \epsilon$
$\mathbf{x}^{n+1} \leftarrow \mathbf{x}$, $\mathbf{v}^{n+1} \leftarrow (\mathbf{x} - \mathbf{x}^n)/h$
return $\mathbf{x}^{n+1}, \mathbf{v}^{n+1}$

Similarly, one can also separate elasticity from contact energy using the contact proxy. In this fashion, we would have a three-phase (fluid, solid, and contact) time splitting scheme

$$\begin{aligned} \widetilde{\mathbf{x}} &= \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}^n\|^2_{\mathbf{M}} + h^2(P(\mathbf{x}) + \hat{C}_{\text{sf}}(\mathbf{x})), \\ \widetilde{\mathbf{x}}' &= \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \widetilde{\mathbf{x}}\|^2_{\mathbf{M}} + h^2(\Psi(\mathbf{x}) + \hat{C}_{\text{sf}}(\mathbf{x}) + \hat{C}_{\text{ss}}(\mathbf{x})), \\ \mathbf{x}^{n+1} &= \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \widetilde{\mathbf{x}}'\|^2_{\mathbf{M}} + h^2\left(\frac{1}{3}C_{\text{sf}}(\mathbf{x}) + \frac{1}{2}C_{\text{ss}}(\mathbf{x})\right), \end{aligned} \tag{26}$$

where $\hat{C}_{\text{sf}}(\mathbf{x})$ and $\hat{C}_{\text{ss}}(\mathbf{x})$ are the 2nd-order Taylor expansion of $\frac{1}{3}C_{\text{sf}}(\mathbf{x})$ and $\frac{1}{2}C_{\text{ss}}(\mathbf{x})$ respectively. However, this aggressive splitting scheme only applies to inversion-robust constitutive models, e.g. the fixed corotated model [Stomakhin et al. 2012]. While inversion can be prevented with guarantee at the solid phase where the elasticity energy is considered, it may not hold at the contact phase. Despite this limitation, the three-phase splitting scheme can still

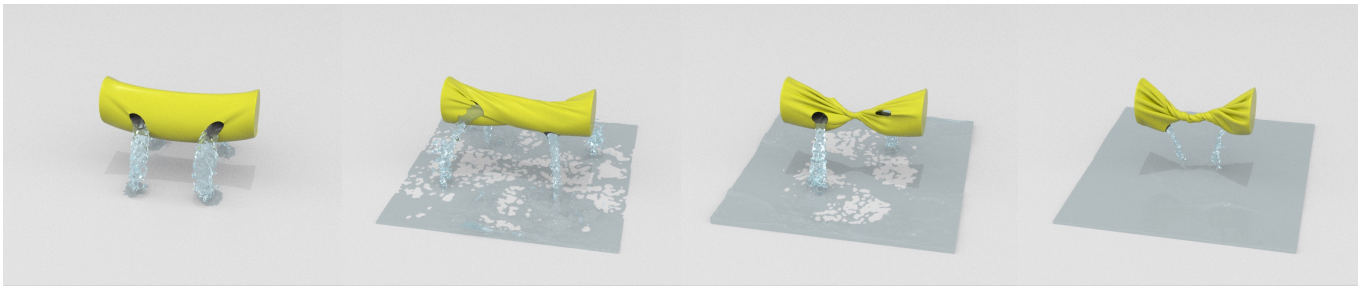


Fig. 6. **Twist cylinder**. A cylindrical cloth with four holes is twisted, squeezing out water from the inside.

work properly for inversion-robust constitutive models in practice to further accelerate the simulation.

4.2 Solving Linear Systems

In our time splitting scheme, solving large sparse linear systems dominates both the computational and memory costs of each phase. We thus devise matrix-free and Schur-complement based strategies to solve them efficiently.

4.2.1 Fluid Phase. Since 2-ring neighbors of SPH particles need to be considered in our formulation, both constructing and directly factorizing the Hessian matrix can cost a significant amount of time and memory. Therefore, we devise a matrix-free conjugate gradient (CG) solver to efficiently solve for the intermediate state of fluids.

As all energy potentials are quadratic in this phase, the energy gradient $\mathbf{g}(\mathbf{x})$ is merely a linear function of $\mathbf{x}$ with constant coefficient matrix $\mathbf{H}(\mathbf{x})$. Thus, the product between $\mathbf{H}(\mathbf{x})$ and an arbitrary vector $\mathbf{p}$ can be expressed as

$$\mathbf{H}(\mathbf{x})\mathbf{p} = \mathbf{g}(\mathbf{p}) - \mathbf{g}(\mathbf{0}). \tag{27}$$

This allows us to compute gradients to evaluate the matrix-vector product, and we only need to acquire the 3×3 diagonal blocks of the Hessian for block-Jacobi preconditioning in our CG solver.

4.2.2 Solid-Coupling Phase. As the fluid energy potential is not included in this phase, the components of the Hessian matrix become

$$\mathbf{H}_f = \frac{\partial^2 C(\mathbf{x})}{\partial \mathbf{x}_f^2}, \quad \mathbf{G} = \frac{\partial^2 C(\mathbf{x})}{\partial \mathbf{x}_s \partial \mathbf{x}_f}, \quad \mathbf{H}_s = \frac{\partial^2 C(\mathbf{x})}{\partial \mathbf{x}_s^2} + \frac{\partial^2 \Psi(\mathbf{x})}{\partial \mathbf{x}_s^2}. \tag{28}$$

Although this linear system is no longer that intractable, it is not optimal to directly factorize the whole system given the considerable number of nonzeros in $\mathbf{H}_f$ and $\mathbf{G}$ when fluid resolution is high.

We thus design a domain decomposed linear solver that treats $\mathbf{H}_f$ and $\mathbf{H}_s$ separately. Based on Schur complement [Zhang 2006], the inverse of our Hessian matrix can be expressed as

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{H}_f^{-1} + \mathbf{H}_f^{-1}\mathbf{G}(\mathbf{H}/\mathbf{H}_f)^{-1}\mathbf{G}^T\mathbf{H}_f & -\mathbf{H}_f^{-1}\mathbf{G}(\mathbf{H}/\mathbf{H}_f)^{-1} \\ -(\mathbf{H}/\mathbf{H}_f)^{-1}\mathbf{G}^T\mathbf{H}_f & (\mathbf{H}/\mathbf{H}_f)^{-1} \end{bmatrix}, \tag{29}$$

where $\mathbf{H}/\mathbf{H}_f = \mathbf{H}_s - \mathbf{G}^T\mathbf{H}_f^{-1}\mathbf{G}$ is the Schur complement of block $\mathbf{H}_f$. Since the nonzeros of $\mathbf{H}_f$ only exist in the diagonal blocks, it is trivial to obtain its inverse matrix $\mathbf{H}_f^{-1}$. We can then apply the CHOLMOD [Chen et al. 2008] LLT solver to factorize $\mathbf{H}/\mathbf{H}_f$, which is only in the size of solid DOFs, and then the search direction can be computed via matrix-vector products and back-solves. When there is no solid-fluid interaction, $\mathbf{H}/\mathbf{H}_f$'s sparsity pattern remains identical with $\mathbf{H}_s$. Only when two solid nodes i and j are interacting with the same fluid particle, the 3×3 block $(\mathbf{H}/\mathbf{H}_f)_{i,j}$ (in 3D) will become non-zero. Typically, this only happens for neighboring mesh primitives and thus the sparsity pattern of $\mathbf{H}/\mathbf{H}_f$ is mostly nice.

Note that when the three-phase time splitting scheme (Eq. 26) is used, our domain decomposed solver can also be applied to the solid and contact phases since their systems share a similar structure with the solid-coupling phase here.

5 EXPERIMENTS AND EVALUATION

Our code is implemented in C++ with Eigen for basic linear algebra operations and Intel TBB for multi-threading. The time step size of all our simulations is adaptively chosen by the SPH CFL condition and a user-defined upper bound. We set the support radius of our SPH kernel function to $2d_f$, where d_f is the particle diameter. In our implementation, we use the cubic Spline kernel for density estimation and the Spiky kernel for gradient calculation. For Fig. 6, 4, 2 and 5, we employ our three-phase time splitting scheme, showing its efficacy when the constitutive models are compatible with mesh inversion. For the rest of the simulations, we stick with our two-phase time splitting scheme. Most experiments are performed on a 24-core 3.50GHz Intel i9-10920X machine, except that the comparative study with ElastoMonolith [Takahashi and Batty 2022] is performed on the "e2-standard-8" (8 cores with 32GB RAM) Google Compute Engine for fairness. We demonstrate that our method achieves efficient and robust solid-fluid coupling. The parameters and timing breakdown of our simulations are provided in Table 1 and Fig. 13 respectively.

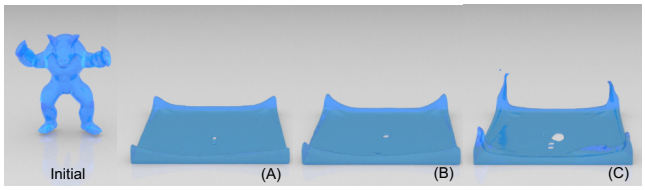
5.1 Ablation Study

5.1.1 Time Splitting Evaluation. Three simulations (Fig. 5 and Fig. 7) are performed to demonstrate the efficiency of time splitting and the efficacy of our proposed contact proxy on maintaining stability.

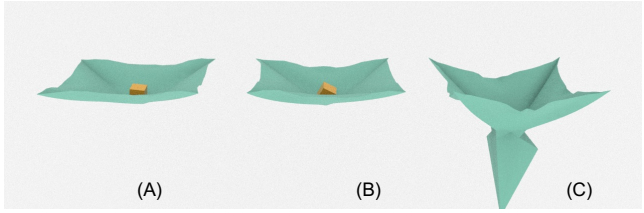
To begin with, we need to take care of choosing a proper time step h. First of all, it has to be restricted by the CFL condition. Otherwise, severe volume loss may be observed due to SPH approximation error. Additionally, in contrast to the joint optimization (Eq. 19), the time splitting scheme usually requires smaller time steps to stay stable, which imposes a second time step constraint. However, we observed that in practice, even using the largest CFL time step, our proxy-assisted time splitting can still work properly and produce

Table 1. **Simulation statistics** including duration of each frame (Δt_{frame}, $[s]$), time step size upperbound (Δt, $[s]$), number of fluid particles (N_{fluid}), number of solid vertices (N_{solid}), incompressibility coefficient (k_I, $[Pa]$), dynamic viscosity (ν_f, $[Pa \cdot s]$), fluid particle diameter (d, $[mm]$), fluid density (ρ_f, $[kg/m^3]$), Young's modulus (E, $[Pa]$), Possion's ratio (ν_s), solid density (ρ_s, $[kg/m^3]$) and the average simulation time for each frame (T, $[min]$). Timing statistics are measured on a 24-core 3.50GHz Intel i9-10920X machine except for Fig. 11, which is tested on the "e2-standard-8" (8 cores with 32GB RAM) Google Compute Engine. Note that examples marked with * contain codimensional materials, whose parameter settings are not covered here.

Scene	Δt_{frame}	Δt	N_{fluid}	N_{solid}	k_I	ν_f	d_f	ρ_f	E	ν_s	ρ_s	T
Fig. 1 Kick Water*	1/24	6×10^{-3}	1M	43K	2.5×10^5	0.1	25	1000	-	-	500	37.9
Fig. 2 Shoot Armadillo	1/24	4×10^{-3}	103K	16K	1×10^5	0	10	1000	1×10^5	0.3	200	1.3
Fig. 3 Cream	1/24	4×10^{-3}	159K	9K	3×10^4	25	3	1000	5×10^8	0.49	1000	1.8
Fig. 4 Buoyancy	1/24	5×10^{-3}	787K	66K	2×10^5	1	10	1000	1×10^5	0.4	200/700/1200	5.9
Fig. 5 Bob	1/24	4×10^{-3}	97K	2.3K	2×10^5	0	15	1000	1×10^5	0.3	500	0.3
Fig. 6 Twist Cylinder*	1/24	5×10^{-3}	486K	12K	4×10^4	0	5	1000	-	-	500	7.9
Fig. 7a Viscous Armadillo	1/48	4×10^{-3}	238K	0	1×10^5	100	10	1200	-	-	-	0.4
Fig. 9 Dam Break	1/24	5×10^{-3}	280K	0	2×10^5	0.005	25	1000	-	-	-	0.4
Fig. 11a Liquid Bunnys	1/50	4×10^{-3}	52K	3.7K	1×10^5	0	10	1000	4×10^3	0.49	200	0.4
Fig. 11b Liquid Bunnys	1/50	4×10^{-3}	101K	4.5K	6×10^4	0	6.4	1000	1×10^3	0.49	200	1.0
Fig. 14 Angry Cow*	1/24	5×10^{-3}	789K	13K	1×10^5	0.2	10	1000	1×10^5	0.45	100/700	4.9



(a) A **viscous armadillo** dropped onto the ground.



(b) **Cube on cloth**. An elastic cube is dropped onto a square cloth with four corners fixed.

Fig. 7. Simulation results of (A) Joint Optimization, (B) Time Splitting with Contact Proxy, and (C) Baseline Time Splitting. While directly applying time splitting results in instability at the boundaries, our results with contact proxy are consistent with joint optimization.

stable simulation results. Hence, for comparison, we use the largest CFL time step for both schemes to maximize their performance as smaller h typically takes more Newton's iterations in total to simulate a frame. For joint optimization, since direct factorization is intractable, we solve Eq. 20 using the block-Jacobi preconditioned conjugate gradient solver with the fluid part matrix free.

As shown in Table 2, our time splitting scheme is significantly (up to 6×) faster than joint optimization, especially for cases (e.g. Fig. 2) involving contacts between fluids and deformable solids. This improvement stems from no longer having to solve for incompressibility of fluids repeatedly within a time step. Moreover, one

Table 2. **Statistics of different time stepping schemes**: Joint Optimization (Joint), Baseline Time Splitting (TS) and Time Splitting with Contact Proxy (TSCP). Our proposed TSCP is much faster than both the Joint and TS.

Scene	Sec/Frame			# Newton Iter./Frame		
	Joint	TS	TSCP	Joint	TS	TSCP
Fig. 5	66.1	38.0	22.5	63.5	117.3	37.1
Fig. 7a	41.3	32.3	25.5	16.5	29.0	10.5
Fig. 2	486.4	158.5	79.8	176.7	187.2	85.8
Fig. 6	2408.2	722.7	472.2	389.1	419.3	219.5

can also find out that Newton's iterations are much less with our proxy-assisted time splitting scheme. As discussed in § 4.1.2, this is because the challenging high-speed impacts are already partially resolved in the fluid phase. Another benefit of time splitting is the support of different error tolerances for the two phases. Errors in the fluid phase are sourced from the solution deviation of the CG solver, while in the solid phase they are directly controlled by the tolerance of Newton's method. Typically, setting a slightly higher tolerance for fluids yields better performance while still producing visually plausible results.

Aside from efficiency, our proposed contact proxy also improves the stability of time splitting scheme. Though simulation results of the baseline time splitting scheme look fine in the case of inviscid fluids, situations get worse when it is applied to viscous fluids. In Fig. 7a, a viscous armadillo is dropped to the ground. In this example, the baseline time splitting scheme produces severe sticky artifacts at the boundary, and the fluid surface could not finally calm down. By consistently applying our contact proxy to exert boundary pressure in the fluid phase, the artifacts can be well resolved as demonstrated in Fig. 7a. Similarly, our idea of contact proxy is also applicable to further separate elasticity from IPC contact while maintaining stability, leading to our three-phase scheme (Fig. 7b).

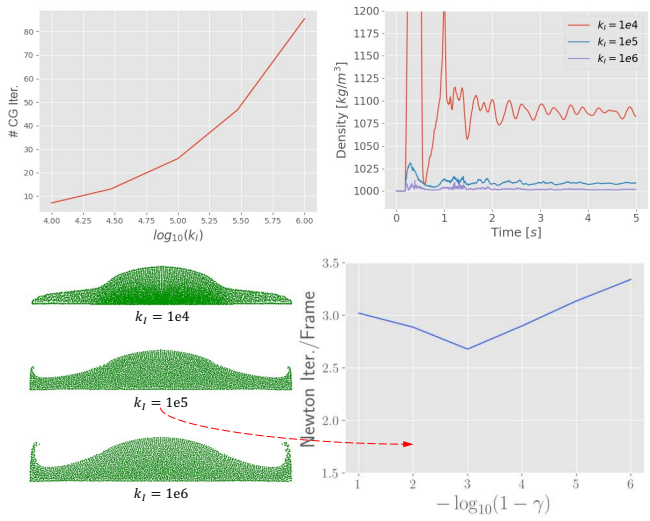


Fig. 8. **Statistics of simulations with different stiffness parameter k_I and contact proxy coefficient γ. Upper left**: the relation between CG iterations and k_I. **Upper right**: density tracking with different k_I. **Bottom left**: simulations with different k_I. A larger k_I preserves volume better but results in more CG iterations. **Bottom right**: number of Newton iterations per frame with different γ when $k_I = 10^5 Pa$.

Table 3. **Time and memory cost of different solvers in example 2**. The costs are measured per time step in units [s] and [MB] respectively, and 'hess' refers to the cost of constructing the hessian matrix. The baseline uses the Conjugate Gradient (CG) method for the fluid phase and CHOLMOD LLT for the solid and contact phases. Our method instead employs a matrix-free CG solver for the fluid phase and a domain-decomposed solver for the solid and contact phases, thereby improving efficiency and saving memory.

Solver	Fluid Phase		Solid Phase	Contact Phase	Mem.
	hess	solve	solve	solve	
CG + LLT	14.9	0.49	1.45	0.43	12375
Ours	0.15	0.59	1.11	0.25	1469

It is also important to choose an appropriate coefficient γ for the contact proxy. Striking a balance between 0 and 1 is essential for better performance. A coefficient too close to 0 would degenerate the scheme to baseline time splitting, again causing stability issues and artifacts. Conversely, a coefficient approaching 1 may hinder optimization convergence due to the excessively small scaling of the barrier contact energy in the solid-coupling phase. The ideal value of γ can be scene-dependent. In Fig. 8, we simulate a scene with varying contact proxy coefficients and document the average number of Newton iterations per frame. While a coefficient of $\frac{1}{2}$, which is used in all our simulations, may not be optimal, our experiments indicate that a broad range of γ all perform adequately well.

5.1.2 Linear Solver Evaluation. For the fluid phase, we designed a matrix-free conjugate gradient (CG) solver that calculates the matrix-vector product via gradient computation to avoid the expensive computational and memory costs of direct factorization (§ 4.2.1).

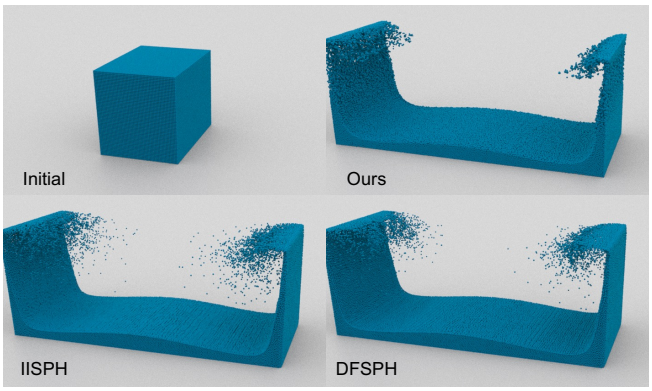


Fig. 9. **Dam break** with 280K SPH particles. Compared to incompressible SPH solvers IISPH [Ihmsen et al. 2013] and DFSPH [Bender and Koschier 2015], our weakly compressible formulation produces stable fluid dynamics without visually evident volume loss.

However, the performance improvement from this approach will be less significant if the number of CG iterations required for convergence is too large, making the cost of computing gradients higher than constructing the Hessian once. In our fluid phase, the number of CG iterations is proportional to the stiffness k_I of the incompressibility energy. A larger k_I can better preserve the volume of the fluids but also results in a worse-conditioned system, demanding more iterations to converge (Fig. 8). In practice, by setting k_I to a proper value, we can efficiently solve the systems within 50 CG iterations without obvious fluid volume loss.

We test the performance of our matrix-free CG solver together with the domain-decomposed solver we designed for the solid-coupling phase on the Shoot Armadillo example (Fig. 2), and present the results in Table 3. Our matrix-free CG solver significantly boosts efficiency (20× faster) and reduces memory costs by avoiding the construction of the Hessian matrix. On the other hand, our domain decomposed solver is 40% faster than directly factorizing the solid and contact systems.

5.2 Comparisons

In this section, we compare our method with several popular SPH fluid solvers and a state-of-the-art solid-fluid coupling method ElastoMonolith [Takahashi and Batty 2022]. We leveraged the open-source library SPlisHSPlasH[1] to implement the SPH fluid simulators. To compare our method with ElastoMonolith, we set up two scenes from their paper with identical parameters and run all the simulations using "e2-standard-8" (8 cores with 32GB RAM) Google Compute Engine for fairness.

5.2.1 Fluid Dynamics. While most existing SPH fluid solvers focus on incompressible fluids, our formulation treats fluids as weakly compressible, allowing us to couple fluids with deformable solids in a unified framework. We run a dam break simulation to compare our method with two SPH fluid solvers IISPH [Ihmsen et al. 2013] and DFSPH [Bender and Koschier 2015]. These methods typically

[1]https://github.com/InteractiveComputerGraphics/SPlisHSPlasH

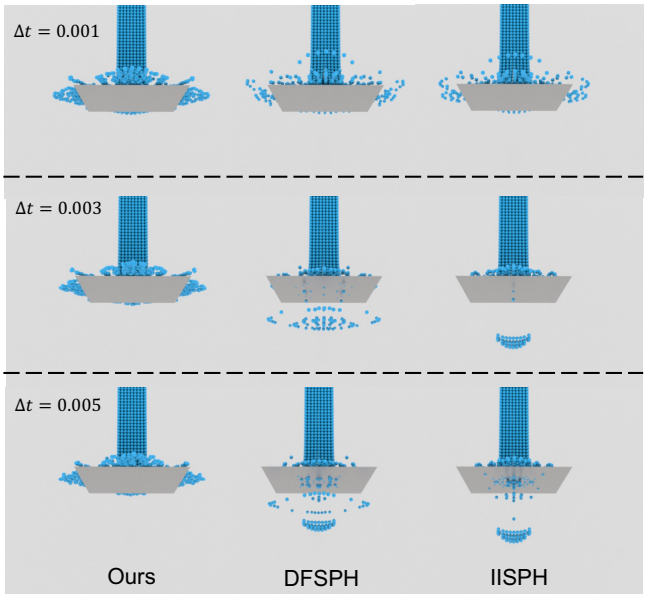
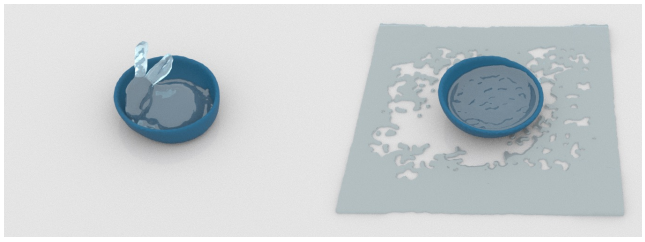
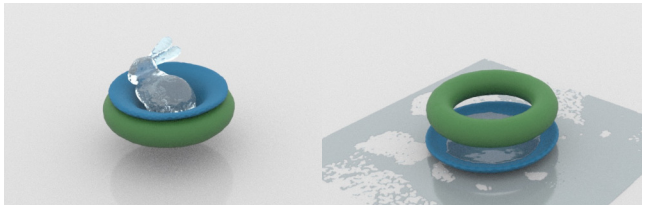


Fig. 10. **Penetration test** where high-speed fluid particles fall onto a fixed thin metal plate. Unlike IISPH/DFSPH with volume map [Bender et al. 2019], our method does not suffer from any particle penetration artifacts regardless of the time step sizes.



(a) A liquid bunny dropped into a bowl.



(b) A liquid bunny and an elastic bowl dropped onto a static torus.

Fig. 11. **Liquid Bunnys**. Compared to ElastoMonolith [Takahashi and Batty 2022], our method achives an over 5× speedup for both of these two examples with exactly the same scene setups.

use particle resampling [Akinci et al. 2013, 2012] or implicit representation [Bender et al. 2019; Koschier and Bender 2017] to exert boundary counter-forces. Our method instead employs IPC [Li et al. 2020] for more robust solid-fluid coupling, with penetration-free guarantee. We uniformly enforce the same CFL condition for all methods along with an upperbound at $5ms$, and use the volume map [Bender et al. 2019] for their boundary handling. As shown in Fig. 9, though our formulation does not strictly enforce incompressibility, it produces natural fluid dynamics without visually observable volume loss. On the other hand, our method (0.45 min/frame) is slower than IISPH (0.31 min/frame) and DFSPH (0.15 min/frame) due to the more sophisticated boundary handling strategy. However, our proposed approach can couple SPH fluids and elastic solids with arbitrary constitutive models, while most existing SPH methods [Kugelstadt et al. 2021; Peer et al. 2018] treat elastic solids as incompressible, which is not generally applicable.

To resolve coupling and boundary conditions, existing SPH methods usually extend the density field into the solid region through either implicit [Bender et al. 2019; Koschier and Bender 2017] or explicit approaches [Akinci et al. 2012], and rely on pressure solvers to exert boundary forces. Consequently, a small time step size is occasionally necessary to prevent particles from infiltrating the solid region, particularly for fast-moving particles. In contrast, our method consistently ensures penetration-free particle trajectories, as demonstrated in Fig. 10. Furthermore, the implicit boundary representation is not applicable to deformable objects due to efficiency considerations, and the explicit one (boundary resampling) can result in bumpy surfaces and inaccurate pressure forces.

5.2.2 Solid-Fluid Coupling. We then compare our method with ElastoMonolith [Takahashi and Batty 2022], which couples Eulerian fluids with Lagrangian solids in a monolithic manner. Following their experiment setting, we run two solid-fluid coupling simulations with identical parameters using our method (Fig. 11). The timing of our method for these two scenes are 24.1 sec/frame and 62.8 sec/frame respectively, both of which are over 5× faster than ElastoMonolith according to their reported timings (253.2 sec/frame and 352.0 sec/frame). Coupling Eulerian fluids with Lagrangian solids requires dealing with geometric differences and it is often needed to perform SPD reformulation to make the linear system tractable. As stated in ElastoMonolith, this SPD reformulation can introduce many additional non-zeros to the system, especially when contacts are rich and solids are intricately shaped. Conversely, our method treats solids and fluids from a unified Lagrangian viewpoint, where solid-solid and solid-fluid contacts are resolved in a unified manner.

5.3 Complex Scenarios

We then evaluate the efficiency and robustness of our method in more complicated scenarios.

Buoyancy. We drop three elastic elephants with varying densities into the water (1000 kg/m^3) in Fig. 4. The light grey elephant (200 kg/m^3) floats on the surface; the blue elephant (700 kg/m^3) is around half immersed in the water; and the red elephant (1200 kg/m^3) sinks to the bottom. This demonstrates that our method correctly captures the buoyancy behavior.

Varying Friction. We drop three viscous bunnies onto the slope with different coefficients of friction (orange bunny: 0.5, green bunny: 0.03, blue bunny: 0) in Fig. 12. All three bunnies share the same dynamic viscosity coefficients 100 kg/m^3 and the angle of slope is 30°.

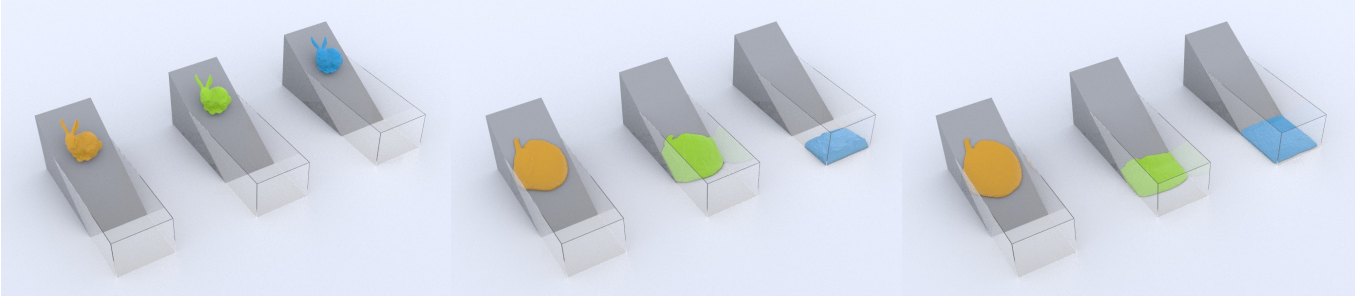


Fig. 12. **Varying friction**. Three viscous bunnies are dropped onto the slope with different coefficients of friction μ (from left to right: 0.5, 0.03, 0). Our method supports adjustable solid-fluid boundary friction.

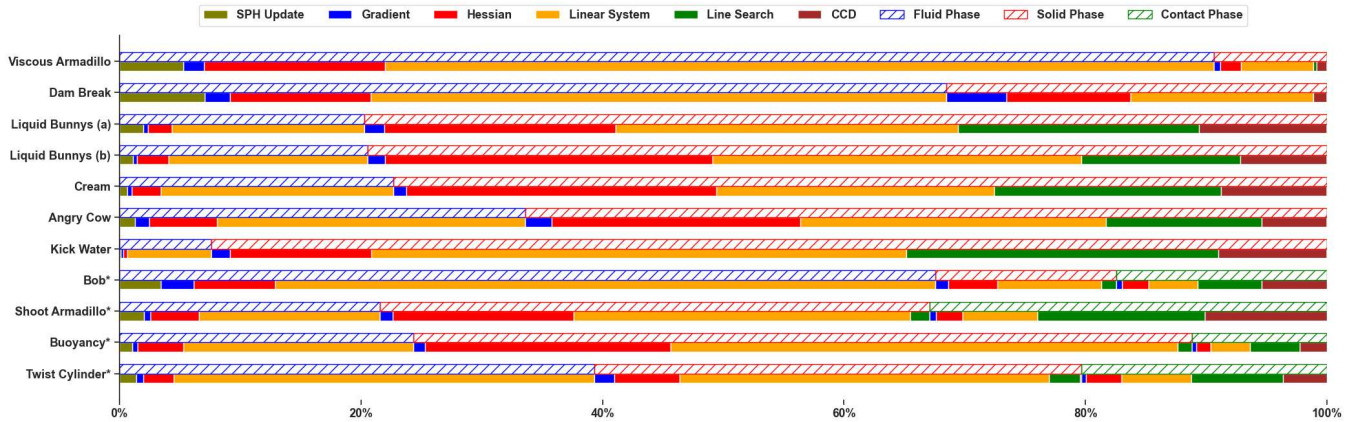


Fig. 13. **Timing breakdown**. We show the timing profile of different simulation phases and plot the proportions of the major routines. Examples marked with * are simulated using our three-phase time splitting scheme. Other examples are generated with the two-phase scheme. In particular, SPH update (including neighborhood search and density update) only occurs in the fluid phase, line search happens in the solid and contact phases for non-linear optimization, and continuous collision detection (CCD) is counted when IPC contact energy is considered.

Twist Cylinder. Coupling fluids with thin shells is challenging since penetration can easily happen without careful treatments. As stated in [Zarifi and Batty 2017], Eulerian fluids may flow through solids if their thickness is less than a grid cell size. Conversely, our approach adopts a unified Lagrangian view and a penetration-free state is guaranteed by IPC. In Fig. 6, we simulate twisting a cylinder full of water. The cylinder is modeled as a thin shell with a $2mm$ thickness, and there are two holes in the front and back sides of this cylinder respectively. The left side and right side are rotated at $72^\circ/s$ and are slowly moved towards each other at $2cm/s$. As we twist the cylinder, the water gets squeezed out through the holes. This simulation demonstrates our method produces stable simulation results with a penetration-free guarantee.

Cream. This example exhibits the coupling behaviors of viscous fluids and elastic solids (Fig. 3). We use an elastic spoon to stir the cream in a porcelain bowl. The spoon handle rotates around the y-axis at $360^\circ/s$ ($0.2m/s$) while the bowl is fixed to the table. As shown in our simulation results, the spoon gets deformed due to the resistance forces it receives from the viscous cream while stirring.

Angry Cow. We then show our framework can simulate natural physical behaviors of geometries in arbitrary codimensions (0, 1, 2, and 3) as well as their interactions. In this scene (Fig. 14), the codimension-0,1,2 objects respectively refer to fluid particles, rubber bands and the leather pad. A deformable cow is launched by the slingshot, hitting the wall consisting of rigid cubes, and then falling into the water pool, producing interesting physical behaviors. The density of the rigid cubes and the cow are $100kg/m^3$ and $700kg/m^3$ respectively.

Kick Water. In this example (Fig. 1), we show a scene where a mannequin dressed in a multilayer skirt kicks in a large water pool, extending the original example in Li et al. [2021] to involve complex interactions between fluid particles and garments. As the mannequin moves in the water, our method produces natural deformation of the skirt caused by the contact with water; as it kicks out of the water at a high speed, the resulting water splash is also correctly captured. Our method well resolves the contacts among fluids particles, thin garments and rapidly moving complex boundaries with a penetration-free guarantee.

Fig. 14. **Angry cow**. We show our method can simulate the coupling of materials in arbitrary codimensions, including fluid particles, rods (the rubber bands), thin shells (the leather pad), deformable solids (the cow), and rigid bodies (the cubes). We launch an angry cow with a slingshot, and the cow hits through the wall and then falls into the water. Interactions between various materials are all accurately captured.

6 CONCLUSION

We presented a unified two-way strong coupling framework for weakly-compressible SPH fluids and nonlinear elastic FEM solids. To achieve this, we modeled solid-fluid interactions as contact forces between SPH particles and FEM boundary elements, applying IPC for guaranteed non-penetration and stability. As we track the volume change of SPH particles in an updated Lagrangian fashion, the incompressibility energy stays quadratic and nice particle distributions are maintained. Utilizing a symmetric approximation of discrete viscosity forces, we proposed a viscosity potential that fits into optimization time integration. We then proposed a time splitting scheme with a contact proxy to efficiently solve the time integration optimization while maintaining robustness. The performance is further boosted by our matrix-free conjugate gradient method and a domain-decomposed solver based on Schur complement.

Compared to existing works [Takahashi and Batty 2022; Zarifi and Batty 2017] coupling Eulerian fluids with Lagrangian elastic solids, our method treats both fluids and solids in a Langrangian manner, avoiding the need to handle different spatial discretizations. Under such a unified view, our method achieves more convenient and robust two-way coupling, even between fluids and codimensional solids. Likewise, different from existing SPH methods [Kugelstadt et al. 2021; Peer et al. 2018] that treat all materials as SPH particles, our formulation enjoys both the efficiency of SPH fluids and the accuracy of FEM solids.

There are many meaningful future research directions. First, when fluid DOFs dominate, building and querying the spatial hash for each fluid particle can become a considerable cost. In fact, since there is no solid-fluid contact for interior particles, we can construct the spatial data structure only in the intersection between the extended bounding boxes of the fluids and each solid for better efficiency, just like the i-BVH scheme in Lan et al. [2022]. In addition, the adhesion between solids and fluids is also an interesting behavior to model. Similar to the barrier energy, adhesion forces can be exerted on close solid-fluid primitive pairs but in the opposite direction. Modeling adhesion via resolving the surface tension of fluids is also an interesting future work.

ACKNOWLEDGMENTS

We would like to express our sincere thanks to the anonymous reviewers for their valuable feedback. This work is partially supported by NSF 2153851, 2153863, 2023780, 2301040, 2008915, 2244651, and 2008564.

REFERENCES

Nadir Akinci, Jens Cornelis, Gizem Akinci, and Matthias Teschner. 2013. Coupling elastic solids with smoothed particle hydrodynamics fluids. *Computer Animation and Virtual Worlds* 24, 3-4 (2013), 195–203.

Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–8.

Anonymous. 2023. A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling: Supplemental Document. (2023).

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 100–es.

Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete viscous sheets. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–7.

Markus Becker, Markus Ihmsen, and Matthias Teschner. 2009a. Corotated SPH for Deformable Solids.. In *NPH*. 27–34.

Markus Becker and Matthias Teschner. 2007. Weakly compressible SPH for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 209–217.

Markus Becker, Hendrik Tessendorf, and Matthias Teschner. 2009b. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (2009), 493–503.

Jan Bender and Dan Koschier. 2015. Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics symposium on computer animation*. 147–155.

Jan Bender and Dan Koschier. 2016. Divergence-free SPH for incompressible and viscous fluids. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2016), 1193–1206.

Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. 2019. Volume maps: An implicit boundary representation for SPH. In *Motion, Interaction and Games*. 1–10.

Javier Bonet and T-SL Lok. 1999. Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in applied mechanics and engineering* 180, 1-2 (1999), 97–115.

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 (TOG)* 38, 6 (2019), 1–16.

Robert Bridson. 2015. *Fluid simulation for computer graphics*. AK Peters/CRC Press.

Yanqing Chen, Timothy A Davis, William W Hager, and Sivasankaran Rajamanickam. 2008. Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)* 35, 3 (2008), 1–14.

Nuttapong Chentanez, Tolga G Goktekin, Bryan E Feldman, and James F O'Brien. 2006. Simultaneous coupling of fluids and deformable bodies. In *ACM SIGGRAPH 2006 Sketches*. 65–es.

Pascal Clausen, Martin Wicke, Jonathan R Shewchuk, and James F O'brien. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Transactions on Graphics (TOG)* 32, 2 (2013), 1–15.

Yu Fang, Ziyin Qu, Minchen Li, Xinxin Zhang, Yixin Zhu, Mridul Aanjaneya, and Chenfanfu Jiang. 2020. IQ-MPM: an interface quadrature material point method for non-sticky strongly two-way coupled nonlinear solids and fluids. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 51–1.

Ronald P Fedkiw. 2002. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *J. Comput. Phys.* 175, 1 (2002), 200–224.

Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. 1999. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics* 152, 2 (1999), 457–492.

Yun Fei, Christopher Batty, Eitan Grinspun, and Changxi Zheng. 2018. A multi-scale model for simulating liquid-fabric interactions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–16.

Georg C Ganzenmüller. 2015. An hourglass control algorithm for Lagrangian smooth particle hydrodynamics. *Computer Methods in Applied Mechanics and Engineering* 286 (2015), 87–106.

Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. 2018. Animating fluid sediment mixture in particle-laden flows. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

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 (TOG)* 38, 1 (2019), 1–13.

Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. 2005. Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 973–981.

David AB Hyde, Steven W Gagniere, Alan Marquez-Razon, and Joseph Teran. 2020. An implicit updated lagrangian formulation for liquids with large surface energy. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–13.

Markus Ihmsen, Nadir Akinci, Marc Gissler, and Matthias Teschner. 2010. Boundary Handling and Adaptive Time-stepping for PCISPH. *VRIPHYS'10*, 79–88.

Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. 2013. Implicit incompressible SPH. *IEEE transactions on visualization and computer graphics* 20, 3 (2013), 426–435.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Courses*. 1–52.

Bryan M Klingner, Bryan E Feldman, Nuttapong Chentanez, and James F O'brien. 2006. Fluid animation with dynamic meshes. In *ACM SIGGRAPH 2006 Papers*. 820–825.

Dan Koschier and Jan Bender. 2017. Density maps for improved SPH boundary handling. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–10.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2022. A Survey on SPH Methods in Computer Graphics. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 737–760.

Tassilo Kugelstadt, Jan Bender, José Antonio Fernández-Fernández, Stefan Rhys Jeske, Fabian Löschner, and Andreas Longva. 2021. Fast corotated elastic SPH solids with implicit zero-energy mode control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 4, 3 (2021), 1–21.

Lei Lan, Danny M Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. 2022. Affine body dynamics: Fast, stable & intersection-free simulation of stiff materials. *arXiv preprint arXiv:2201.10022* (2022).

David IW Levin, Joshua Litven, Garrett L Jones, Shinjiro Sueda, and Dinesh K Pai. 2011. Eulerian solid simulation with contact. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–10.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020), 49.

Minchen Li, Danny M Kaufman, and Chenfanfu Jiang. 2021. Codimensional incremental potential contact. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–24.

Xuan Li, Yu Fang, Minchen Li, and Chenfanfu Jiang. 2022. BFEMP: Interpenetration-free MPM–FEM coupling with barrier contact. *Computer Methods in Applied Mechanics and Engineering* 390 (2022), 114350.

Miles Macklin and Matthias Müller. 2013. Position based fluids. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

Joe J Monaghan. 1992. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics* 30 (1992), 543–574.

Joe J Monaghan. 1994. Simulating free surface flows with SPH. *Journal of computational physics* 110, 2 (1994), 399–406.

Joe J Monaghan. 2005. Smoothed particle hydrodynamics. *Reports on progress in physics* 68, 8 (2005), 1703.

Jean Jacques Moreau. 2011. On unilateral constraints, friction and plasticity. In *New variational techniques in mathematical physics*. Springer, 171–322.

Matthias Müller, David Charypar, and Markus H Gross. 2003. Particle-based fluid simulation for interactive applications.. In *Symposium on Computer animation*, Vol. 2.

Andreas Peer, Christoph Gissler, Stefan Band, and Matthias Teschner. 2018. An implicit SPH formulation for incompressible linearly elastic solids. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 135–148.

Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. 2015. An implicit viscosity formulation for SPH fluids. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

Andreas Peer and Matthias Teschner. 2016. Prescribed velocity gradients for highly viscous SPH fluids with vorticity diffusion. *IEEE transactions on visualization and computer graphics* 23, 12 (2016), 2656–2662.

Charles S Peskin. 2002. The immersed boundary method. *Acta numerica* 11 (2002), 479–517.

Avi Robinson-Mosher, Craig Schroeder, and Ronald Fedkiw. 2011. A symmetric positive definite formulation for monolithic fluid structure interaction. *J. Comput. Phys.* 230, 4 (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 Transactions on Graphics (TOG)* 27, 3 (2008), 1–9.

Doug Roble, Nafees bin Zafar, and Henrik Falt. 2005. Cartesian grid fluid simulation with irregular boundary voxels. In *ACM SIGGRAPH 2005 Sketches*. 138–es.

Eftychios Sifakis and Jernej Barbic. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *Acm siggraph 2012 courses*. 1–50.

Barbara Solenthaler and Renato Pajarola. 2009. Predictive-corrective incompressible SPH. In *ACM SIGGRAPH 2009 papers*. 1–6.

Barbara Solenthaler, Jürg Schläfli, and Renato Pajarola. 2007. A unified particle model for fluid–solid interactions. *Computer Animation and Virtual Worlds* 18, 1 (2007), 69–82.

Alexey Stomakhin, Russell Howes, Craig A Schroeder, and Joseph M Teran. 2012. Energetically Consistent Invertible Elasticity.. In *Symposium on Computer Animation*, Vol. 1.

Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

Deborah Sulsky, Shi-Jian Zhou, and Howard L Schreyer. 1995. Application of a particle-in-cell method to solid mechanics. *Computer physics communications* 87, 1-2 (1995), 236–252.

Tetsuya Takahashi and Christopher Batty. 2020. Monolith: a monolithic pressure-viscosity-contact solver for strong two-way rigid-rigid rigid-fluid coupling. (2020).

Tetsuya Takahashi and Christopher Batty. 2021. FrictionalMonolith: a monolithic optimization-based approach for granular flow with contact-aware rigid-body coupling. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–20.

Tetsuya Takahashi and Christopher Batty. 2022. ElastoMonolith: A Monolithic Optimization-Based Liquid Solver for Contact-Aware Elastic-Solid Coupling. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–19.

Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C Lin. 2015. Implicit formulation for SPH-based viscous fluids. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 493–502.

Andre Pradhana Tampubolon, Theodore Gast, Gergely Klár, Chuyuan Fu, Joseph Teran, Chenfanfu Jiang, and Ken Museth. 2017. Multi-species simulation of porous sand and water mixtures. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11.

Yun Teng, David IW Levin, and Theodore Kim. 2016. Eulerian solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–8.

Boris Valkov, Chris H Rycroft, and Ken Kamrin. 2015. Eulerian method for multiphase interactions of soft solid bodies in fluids. *Journal of Applied Mechanics* 82, 4 (2015), 041011.

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional surface tension flow using moving-least-squares particles. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 42–1.

Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A physically consistent implicit viscosity solver for SPH fluids. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 145–155.

Xiao Yan, C-F Li, X-S Chen, and S-M Hu. 2018. MPM simulation of interacting fluids and solids. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 183–193.

Omar Zarifi and Christopher Batty. 2017. A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–11.

Fuzhen Zhang. 2006. *The Schur complement and its applications*. Vol. 4. Springer Science & Business Media.