

ElastoMonolith: A Monolithic Optimization-based Liquid Solver for Contact-Aware Elastic-Solid Coupling

TETSUYA TAKAHASHI, Adobe, USA
CHRISTOPHER BATTY, University of Waterloo, Canada

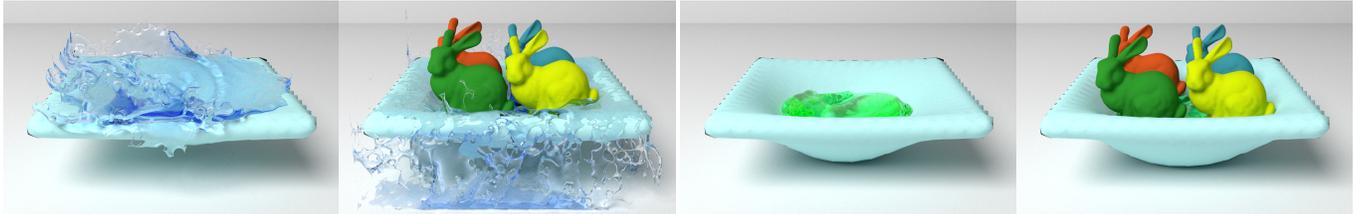


Fig. 1. Our optimization-based solver enables efficient and robust simulations of three-way coupling among inviscid/viscous fluids with elastic and rigid solids. A bunny-shaped inviscid (left) or viscous (right) liquid volume is dropped onto an elastic sheet, followed by multiple rigid bunnies thrown on top of them.

Simultaneous coupling of diverse physical systems poses significant computational challenges in terms of speed, quality, and stability. Rather than treating all components with a single discretization methodology (e.g., smoothed particles, material point method, Eulerian grid, etc.) that is ill-suited to some components, our solver, *ElastoMonolith*, addresses three-way interactions among standard particle-in-cell-based viscous and inviscid fluids, Lagrangian mesh-based deformable bodies, and rigid bodies. While prior methods often treat some terms explicitly or in a decoupled fashion for efficiency, often at the cost of robustness or stability, we demonstrate the effectiveness of a strong coupling approach that expresses all of the relevant physics within one consistent and unified optimization problem, including fluid pressure and viscosity, elasticity of the deformables, frictional solid-solid contact, and solid-fluid interface conditions. We further develop a numerical solver to tackle this difficult optimization problem, incorporating projected Newton, an active set method, and a transformation of the inner linear system matrix to ensure symmetric positive definiteness. Our experimental evaluations show that our framework can achieve high quality coupling results that avoid artifacts such as volume loss, instability, sticky contacts, and spurious interpenetrations.

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

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

ACM Reference Format:

Tetsuya Takahashi and Christopher Batty. 2022. ElastoMonolith: A Monolithic Optimization-based Liquid Solver for Contact-Aware Elastic-Solid Coupling. *ACM Trans. Graph.* 41, 6, Article 255 (December 2022), 20 pages. <https://doi.org/10.1145/3550454.3555474>

Authors' addresses: Tetsuya Takahashi, Adobe, USA, ttakahas@adobe.com; Christopher Batty, University of Waterloo, Canada, christopher.batty@uwaterloo.ca.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

0730-0301/2022/12-ART255 \$15.00

<https://doi.org/10.1145/3550454.3555474>

1 INTRODUCTION

Unified physical simulation of diverse materials within a single system is a longstanding challenge in computer animation, and across computational mechanics more broadly. From a user perspective, providing seamless interaction of different scene components can dramatically simplify the workflow, so that one need not cobble together disparate tools or their outputs to achieve a desired result. From a computational perspective, properly accounting for all interacting components in a tightly coupled fashion can yield solutions with greater accuracy and numerical stability, which the user experiences as higher quality and robustness.

However, realizing this lofty objective has been difficult, in part because different materials tend to be best suited to very different simulation methodologies. Among the many physical systems of interest, we focus on interactions between volumetric solids (both rigid and elastic) and fluids (both inviscid and viscous), as they rank among the most fundamental. The Lagrangian viewpoint clearly dominates for solids, while Eulerian or hybrid methods arguably dominate for fluids. One coupling strategy, popular because it dispenses with careful integration of each distinct subcomponent, is to treat all materials with essentially a single methodology (e.g., material point method [Jiang et al. 2016], smoothed particle hydrodynamics [Solenthaler et al. 2007], position-based dynamics [Macklin et al. 2014], pure Eulerian grids [Teng et al. 2016], etc.); however, this can sacrifice significant advantages of the preferred individual methods (e.g., treating a rigid body as a full finite element deformable can be unwieldy and inefficient).

The natural alternative is to use appropriate, modern models for each of the rigid, elastic, and fluid components. This too is a popular approach, but has often fallen short for a few reasons. First, there is a perception that a fully integrated implicit treatment is simply too difficult; second, for the sake of speed and convenience, past ostensibly strong fluid coupling work has still carved off some terms to be treated explicitly or separately from fluid incompressibility, such as elastic potential energy [Fang et al. 2020], fluid viscosity [Takahashi and Lin 2019], or mutual solid contacts for rigid bodies

[Batty et al. 2007] or deformables [Robinson-Mosher et al. 2008] (Takahashi and Batty [2020] provide a recent exception for rigid bodies); and third, no practical numerical treatment for such a strongly coupled, unified approach has yet been demonstrated.

In this work we show that, given current standard models for rigid bodies [Bender et al. 2014], elastic bodies [Kim and Eberle 2020; Smith et al. 2018], and liquids [Bridson 2015], such a monolithic approach, while not trivial, is nevertheless viable and effective. We propose a fully coupled optimization-based solver, *ElastoMonolith*, that robustly simulates hybrid particle/grid-based liquids, rigid bodies, stable Neo-Hookean elastic bodies, and all of their mutual coupling effects within one framework. Specifically, our formulation consists of a single constrained minimization problem, unifying the following physical components: incompressibility constraints and implicit viscosity integration for liquids; large deformations of hyperelastic solids; two-way fluid-solid coupling via pressure and viscous stress; and frictional contacts among all solids. To tackle this simultaneous formulation, we further propose a custom numerical optimizer, featuring a transformation of the inner linear system matrix to a sparse, symmetric positive definite (SPD) form designed to efficiently handle contributions from elastic potentials.

2 RELATED WORK

2.1 Viscous Liquids

Reproducing the familiar behaviors of viscous liquids has traditionally been a computational challenge. Early grid-based approaches focused on improving stability using implicit integration [Stam 1999] with free surfaces [Carlson et al. 2002], addressing spatial variations in viscosity [Rasmussen et al. 2004], and accounting for rotational effects to capture buckling and coiling phenomena [Batty and Bridson 2008; Larionov et al. 2017]. More recently there has been interest in accelerating viscous fluid simulations with multigrid solvers [Aanjaneya et al. 2019; Shao et al. 2022] or spatial adaptivity [Batty and Houston 2011; Goldade et al. 2019]. Takahashi and Batty [2020] also offered an extension for non-Newtonian viscosity.

We employ a unified pressure-viscosity formulation [Larionov et al. 2017] within a staggered grid, particle-in-cell-style fluid solver [Jiang et al. 2015; Zhu and Bridson 2005]. Other methodologies have also been proposed for viscous flows, e.g., Material Point Method (MPM) [Fang et al. 2019; Gao et al. 2017; Ram et al. 2015; Stomakhin et al. 2014; Su et al. 2021; Yue et al. 2015], Smoothed Particle Hydrodynamics (SPH) [Liu et al. 2021; Peer et al. 2015; Peer and Teschner 2017; Takahashi et al. 2015; Weiler et al. 2018], Finite Element Method (FEM) [Bargteil et al. 2007; Clausen et al. 2013; Wicke et al. 2010; Wojtan and Turk 2008], and Lagrangian simplicial elements [Batty et al. 2012; Bergou et al. 2010; Misztal and Bærentzen 2012; Misztal et al. 2014; Zhu et al. 2015].

2.2 Rigid-Body Contact Handling

Contact handling of rigid bodies has long been a key topic in computer animation, and we refer to the course notes of Andrews and Erleben [2021] for contact basics. To resolve rigid-body contacts (with a pyramidal friction cone approximation), velocity-level linear complementarity problem (LCP) formulations have been extensively used [Stewart 2000]. The LCP is typically solved with projected Gauss-Seidel (PGS) due to its simplicity, despite its slow convergence

rate (although acceleration techniques are available [Coevoet et al. 2020; Erleben 2007, 2017; Müller et al. 2017; Silcowitz et al. 2010b; Tonge et al. 2012]). As a stationary iterative approach, the convergence rate of PGS is fundamentally limited, and thus reformulations of the LCP into quadratic programs (QPs) have been presented to address contact handling as a minimization [Kaufman et al. 2008; Renouf and Alart 2005]. To more accurately handle friction cone constraints, quadratically constrained QP (QCQP) formulations have also been proposed and solved with various approaches, e.g., an interior point method [Todorov 2011], accelerated projected gradient descent [Mazhar et al. 2015], the alternating direction method of multipliers (ADMM) [Le Lidec et al. 2021], and an augmented Lagrangian method [Takahashi and Batty 2021]. Yet another family of approaches relies on nonlinear complementarity problem (NCP) formulations [Silcowitz et al. 2009, 2010a; Todorov 2010].

While recently proposed position-level contact handling approaches (e.g., [English et al. 2013; Ferguson et al. 2021; Müller et al. 2020]) tend to offer better robustness compared to velocity-level ones, position-level approaches are typically more costly. As such, in our framework, we adopt a velocity-level minimization formulation.

2.3 Elastic-Solid Contact Handling

The LCP-based contact handling formulation is also popular for elastic solids (although some authors instead employ penalty-based contact forces [Bridson et al. 2002; Fisher and Lin 2001; Tang et al. 2012; Xu et al. 2014], barrier methods [Lan et al. 2021; Li et al. 2020b, 2021], or a hybrid thereof [Wu et al. 2020]). Compared to the LCP approach for rigid bodies, the unique challenge for elastic solids is that it is non-trivial to form the *Delassus operator* [Duriez et al. 2006] because, unlike the rigid body mass matrix, the elastic stiffness matrix is not (block-) diagonal and thus more difficult to invert explicitly. Consequently, elasticity handling has often been decoupled from contact handling [Larionov et al. 2021], or the stiffness matrix has been replaced with block-diagonal matrices [Galoppo et al. 2006, 2007] or warped compliance [Saupin et al. 2008], so that the LCP is solved only approximately. Later, Otaduy et al. [2009] incorporated the block-diagonal approximation for the stiffness matrix within GS iterations, correctly solving the LCP. A similar approach was used for hair contacts with frictional cone constraints [Daviet et al. 2011]. Macklin et al. [2019a] also presented a related approximation technique based on geometric stiffness (which arises in constraint-based formulations) resulting in a quasi-Newton method.

Since approximating the stiffness matrix with a block-diagonal matrix can significantly slow down the convergence of iterative solvers in practice, recent approaches focus on solving the contact problem without forming the Delassus operator. Li et al. [2015] formulated a convex QP for non-penetration constraints (without supporting friction) and solved it with an extension of the *modified proportioning with reduced gradient projections* (MPRGP) method [Dostal and Schoberl 2005]. While the original MPRGP handles only box constraints, their extension supports linear constraints with a strict requirement that a mesh vertex can be involved in at most one contact constraint. However, their restricted contact configuration can cause penetration artifacts. Similarly, Verschoor and Jalba [2019] formulated a QP for frictional contacts and solved the corresponding indefinite system (due to the presence of Lagrange multipliers) with

their custom solver based on the conjugate residual (CR) method [Saad 2003]. Narain et al. [2016] and Daviet [2020] proposed ADMM-based operator splitting approaches which enable decoupling of elasticity and contact problems while correctly accounting for both elasticity and contacts. Overby et al. [2017] further extended the ADMM approach to enable early termination with (sticky) contact constraints via Uzawa iterations. Li et al. [2018] and Ly et al. [2020] presented formulations that avoid forming the Delassus operator by restricting contacts to be on the mesh vertices.

Our elastic-solid contact formulation is derived primarily from the velocity-level LCP, but we augment the formulation to the position-level to accurately evaluate elastic potentials and enable updating contact information for robustness. Our proposed method requires solving only sparse SPD systems, despite the presence of Lagrange multipliers, and fully avoids construction of the Delassus operator.

2.4 Two-Way Fluid-Solid Coupling

Initial two-way coupling approaches for Eulerian fluids and Lagrangian solids typically focused on weak coupling (e.g., [Gundelman et al. 2005]), but more recent work has emphasized strong coupling due to its relative robustness and stability [Batty et al. 2007; Chentanez et al. 2006; Klingner et al. 2006]. In particular, a popular variational formulation based on pressure [Batty et al. 2007] has been extended with viscosity [Hyde and Fedkiw 2019; Takahashi and Batty 2020; Takahashi and Lin 2019], friction [Narain et al. 2010; Takahashi and Batty 2021], and surface tension [Ruan et al. 2021]. Robinson-Mosher et al. [2009, 2011, 2008] also augmented the method of Batty et al. [2007] to include deformable objects, and this approach was later extended with a cut-cell formulation [Zarifi and Batty 2017] and to support reduced elastic solids [Lu et al. 2016].

Our new formulation shares the same fundamental goal with recent unified contact-aware fluid coupling formulations of Takahashi and Batty [2020, 2021], but their support for solid objects is limited to rigid bodies. Our work contrasts starkly in that we enable strong, implicit coupling of *elastic* solids with both inviscid/viscous liquids *and* rigid bodies, all within a unified optimization framework. To this end, unlike their formulations based on minimization of kinetic energy with respect to the dual variables (i.e., forces or impulses), we derive a more general formulation based on the minimization of inertial terms, deformation rate, and elastic potential with respect to primal variables under the incompressibility and frictional contact constraints (see Appendix C for further discussion). We further derive a reformulation that ensures strictly SPD matrices, thereby avoiding the difficulties of solving indefinite systems. Compared to the cut-cell coupling method of Zarifi and Batty [2017], we achieve better conditioning, viscosity handling, awareness of all solid contacts in the coupled fluid solve, and a consistent unified optimization formulation.

Our focus is on the coupling of Eulerian fluids and Lagrangian solids, which are natural choices for each. However, various other approaches and frameworks can achieve two-way coupling. MPM has widely been employed due to its (semi-)automatic handling of contacts [Ding and Schroeder 2020; Han et al. 2019; Hu et al. 2018; Jiang et al. 2017; Klár et al. 2016; Stomakhin et al. 2013] with extensions for free-slip boundaries [Fang et al. 2020]. SPH and particle-based methods, being fully Lagrangian, are also popular for two-way coupling

frameworks [Akinci et al. 2012; Band et al. 2018a,b; Becker et al. 2009; Gissler et al. 2019; Koschier and Bender 2017; Kuglstadt et al. 2021; Truong et al. 2021]. On the other extreme, a purely Eulerian fluid-solid coupling approach was proposed by Teng et al. [2016]. In a related vein, variants of the traditional immersed boundary method [Peskin 2002] smoothly spread the Lagrangian solid’s influence onto the Eulerian fluid grid [Brandt et al. 2019; Carlson et al. 2004; Guendelman et al. 2005]. Recent lattice Boltzmann solvers for turbulent smoke flows also support two-way rigid-fluid coupling [Li et al. 2020a; Lyu et al. 2021]. Lastly, Akbay et al. [2018] proposed an extended partitioned method, offering a middle ground between strong and weak coupling that uses a reduced order model to couple different components.

3 ELASTOMONOLITH

Our framework employs an optimization-based integrator to handle the fluid and solid systems and their coupling. For liquids, we adopt the affine particle-in-cell (APIC) framework [Jiang et al. 2015] and address the fully implicit update of grid-based dynamics on the staggered grid [Goktekin et al. 2004] as an optimization problem that relies on volume fractions [Larionov et al. 2017; Takahashi and Lin 2019]. For rigid bodies, we formulate velocity-level frictional contact handling as an optimization problem. For elastic solids, we treat their entire time integration process (including contact handling) as a position-level optimization. An overview of our method’s steps is listed in Algorithm 1.

Algorithm 1 ElastoMonolith

- 1: Map fluid velocity from particles to a grid
 - 2: Add explicit external forces to the grid and solids
 - 3: Solve the system with our custom optimizer (Algorithm 2)
 - 4: Map fluid velocity from the grid to particles
 - 5: Advect particles and update rigid body positions
-

To achieve stable integration and contact-aware coupling of inviscid/viscous liquids and rigid/elastic solids, and eliminate a range of critical artifacts (e.g., loss of fluid volume, solid interpenetrations, excessive loss of fluid and solid energies, and simulation instabilities), we formulate a single unified minimization problem with incompressibility and frictional contact constraints,

$$\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r = \arg \min_{\mathbf{d}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) \in \mathcal{D}, \mathbf{h}(\mathbf{x}_e, \mathbf{v}_r) \in \mathcal{H}} E(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r), \quad (1)$$

$$E(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) = E_f(\mathbf{u}) + E_e(\mathbf{x}_e) + E_r(\mathbf{v}_r), \quad (2)$$

where \mathbf{u} denotes the liquid velocity, \mathbf{x}_e the position of elastic solids, \mathbf{v}_r the velocity of rigid bodies, \mathcal{D} and \mathcal{H} the incompressibility and frictional contact constraint functions, respectively, and sets of vectors \mathcal{D} and \mathcal{H} , satisfying their corresponding constraints. The objective function terms $E_f(\mathbf{u})$, $E_e(\mathbf{x}_e)$, and $E_r(\mathbf{v}_r)$ correspond to inviscid/viscous liquids, elastic solids, and rigid bodies, respectively. We define these objectives and constraints in the following sections.

3.1 Rigid Body Formulation

Velocity-level contact handling of rigid bodies can be formulated as an energy minimization. Considering contacts given as hard constraints, the objective for rigid bodies can be written as the

kinetic energy-norm of the difference between velocities before and after contact handling:

$$E_r(\mathbf{v}_r) = \frac{1}{2} \|\mathbf{v}_r - \mathbf{v}_r^*\|_{\alpha \mathbf{M}_r}^2, \quad (3)$$

where \mathbf{v}_r^* and \mathbf{v}_r are the rigid body velocities before and after contact handling, respectively, \mathbf{M}_r is the rigid body mass matrix, and α ($= 1/(\Delta x)^3$ in 3D, where Δx is the grid cell size) is the solid mass scaling for dimensional consistency in the fluid-solid coupled system [Takahashi and Batty 2020]. The matrix-weighted vector norm $\|\cdot\|_{\mathbf{W}}$ is defined such that, for a vector \mathbf{y} , $\|\mathbf{y}\|_{\mathbf{W}} = \sqrt{\mathbf{y}^T \mathbf{W} \mathbf{y}} = \|\mathbf{W}^{\frac{1}{2}} \mathbf{y}\|_2$. Given fluid viscous stress \mathbf{s} , fluid pressure \mathbf{p} , and solid contact forces \mathbf{c} , the implicit velocity update for rigid bodies can be computed as

$$\mathbf{v}_r = \mathbf{v}_r^* + \Delta t (\alpha \mathbf{M}_r)^{-1} (\mathbf{F}_{r,s} \mathbf{s} + \mathbf{F}_{r,p} \mathbf{p} + \alpha \mathbf{J}_r^T \mathbf{c}), \quad (4)$$

where Δt denotes the timestep size, $\mathbf{F}_{r,s}$ and $\mathbf{F}_{r,p}$ the linear operators integrating fluid stress and pressure on the rigid body surface using liquid/air volume fractions [Takahashi and Batty 2020, 2021], and \mathbf{J}_r the Jacobian of the contact constraints for rigid bodies [Andrews and Erleben 2021]. While some previous approaches decouple normal and tangential components of the contact forces in the solver process [Kaufman et al. 2008], we treat them in a unified way since we observed no issues from doing so in our examples (and this approach was also taken in other recent work [Macklin et al. 2019a; Todorov 2011; Verschoor and Jalba 2019]).

3.2 Elastic Solid Formulation

Implicit time integration for elastic solids can also be cast as an energy minimization problem, which combines terms for inertia and elastic potentials [Gast et al. 2015; Martin et al. 2011]. Given elastic solids discretized with meshes, elastic potentials are typically evaluated based on the positions of mesh vertices. Let us define the vertex positions \mathbf{x}_e^* before time integration and contact handling as

$$\mathbf{x}_e^* = \mathbf{x}_e^t + \Delta t \mathbf{v}_e^t + \Delta t^2 \mathbf{a}_e^t, \quad (5)$$

where \mathbf{x}_e^t , \mathbf{v}_e^t , and \mathbf{a}_e^t are vertex positions, velocities, and accelerations at time t , respectively. Then the objective based on vertex positions \mathbf{x}_e is

$$E_e(\mathbf{x}_e) = \frac{1}{2\Delta t^2} \|\mathbf{x}_e - \mathbf{x}_e^*\|_{\alpha \mathbf{M}_e}^2 + \alpha \Psi(\mathbf{x}_e), \quad (6)$$

where \mathbf{M}_e is a diagonal mass matrix for elastic solids, and $\Psi(\mathbf{x}_e)$ is the elastic potential function. Similar to the velocity update for rigid bodies in (4), we can write the implicit position update for elastic solids as

$$\mathbf{x}_e = \mathbf{x}_e^* + \Delta t^2 (\alpha \mathbf{M}_e)^{-1} (-\alpha \nabla \Psi(\mathbf{x}_e) + \mathbf{F}_{e,s} \mathbf{s} + \mathbf{F}_{e,p} \mathbf{p} + \alpha \mathbf{J}_e^T \mathbf{c}), \quad (7)$$

where $\mathbf{F}_{e,s}$ and $\mathbf{F}_{e,p}$ are linear operators integrating the viscous stress and fluid pressure on the elastic solid surface (further details in Sec. 3.2.2), respectively, and \mathbf{J}_e is the Jacobian of the contact constraints for elastic solids [Verschoor and Jalba 2019].

3.2.1 Implementation and Discussion. While our approach can, in principle, work with many choices of discretization, element type, and constitutive law for elastic solids, we use tetrahedral meshes with linear finite elements due to their popularity [Sifakis and Barbic 2012] and employ the stable Neo-Hookean model due to its

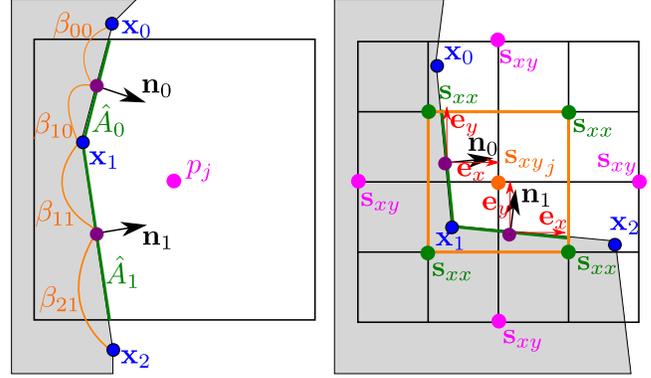


Fig. 2. 2D illustration for the cut-cell approach over cells (control volumes) for pressure \mathbf{p} (left) and viscous stress \mathbf{s}_{xy} (right). Left: The pink dot represents the pressure p_j in the cell. The gray area is an elastic solid domain, blue dots are solid vertices ($\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$), and green segments are polygons inside the cell (i.e., triangles clipped against the cell), with associated centroids shown as purple dots and normals ($\mathbf{n}_0, \mathbf{n}_1$) drawn as black arrows. The corresponding polygon areas are denoted \hat{A}_0, \hat{A}_1 in green. $\beta_{00}, \beta_{10}, \beta_{11}, \beta_{21}$ in orange represent the barycentric weights. Right: The orange dot represents \mathbf{s}_{xy_j} for stress cell j (orange square), and pink (resp., green) dots represent neighboring \mathbf{s}_{xy} (resp., \mathbf{s}_{xx}). The red arrows indicate the unit normals $\mathbf{e}_x, \mathbf{e}_y$.

robustness [Kim and Eberle 2020; Smith et al. 2018]. We use the optimization form of implicit Euler, which is simple and robust to large timesteps, with the tradeoff that the energy of the elastic solids can dissipate relatively quickly. For better energy preservation, it would be possible to employ higher order schemes (e.g., BDF2 [Bender et al. 2017]), blended schemes [Dinev et al. 2018a], or energy constraints [Dinev et al. 2018b; Kee et al. 2021]. Optimization-based damping models could also be incorporated (e.g., [Brown et al. 2018]) if desired.

3.2.2 Details of $\mathbf{F}_{e,p}$ and $\mathbf{F}_{e,s}$ for Cut-Cell Coupling. Fluid coupling forces for rigid bodies are readily accumulated to their centers of mass (via linear coupling operators $\mathbf{F}_{r,s}$ and $\mathbf{F}_{r,p}$), but for elastic solids, the relevant forces must instead be carefully distributed to solid surface vertices. As such, the construction approach for $\mathbf{F}_{r,s}$ and $\mathbf{F}_{r,p}$ using level-set-based volume evaluation is not applicable to elastic solids, because one cannot identify the contributions of each solid surface vertex onto fluid DOFs from the level-set values (since they lack positional information about the surface vertices). Thus, we instead assemble $\mathbf{F}_{e,p}$ and $\mathbf{F}_{e,s}$ using a cut-cell approach based more directly on the solid geometry.

$\mathbf{F}_{e,p}$ is a linear operator (matrix) that applies fluid pressure forces to elastic solids over their surfaces, taking into account fluid vs. solid (F/S) and liquid vs. air (L/A) volume fractions [Larionov et al. 2017; Takahashi and Lin 2019]. Assuming signed distance functions (SDFs) to represent L/A domains [Zhu and Bridson 2005], we can compute dimensionless volume fractions (e.g., using marching-cubes-style volume evaluations [Takahashi and Batty 2022]) as diagonal matrices corresponding to the locations of pressure, velocity, and viscous stress DOFs defined on the staggered grid [Goktekin et al. 2004], i.e., \mathbf{W}_L^p and \mathbf{W}_A^p for pressure, \mathbf{W}_L^u and \mathbf{W}_A^u for velocity, and \mathbf{W}_L^s and \mathbf{W}_A^s for viscous stress to account for L/A sub-grid geometry.

Given the surface triangles of the solids' tetrahedral FEM meshes in pressure cell j (see Figure 2 for 2D illustration), pressure \mathbf{p}_j is applied to the elastic solids using a cut-cell approach. We account for F/S sub-grid geometry via the polygons computed from the surface triangles by clipping the triangles against the cell (using the Sutherland–Hodgman algorithm). Then, taking both F/S and L/A sub-grid geometry into account, $\mathbf{F}_{e,p_{3 \times i,j}}$ (where i denotes the solid vertex index) in 3D can be computed by summing up the contributions over all the polygons in the cell as

$$\mathbf{F}_{e,p_{3 \times i,j}} = -\frac{\mathbf{W}_{Lj}^p}{\Delta x} \sum_k \beta_{ik} A_k \mathbf{n}_k, \quad (8)$$

where β_{ik} denotes the barycentric weights at the centroid of polygon k with respect to vertex i (and thus $\beta_{ak} + \beta_{bk} + \beta_{ck} = 1$ where a, b, c indicate the vertices of the triangle (polygon) k), and A_k and \mathbf{n}_k are the normalized area and unit outward normal (with respect to the solid) of polygon k , respectively. We define the normalized area A_k given the actual area of the polygon \hat{A}_k as $A_k = \hat{A}_k / (\Delta x)^2$ in 3D (A_k can exceed 1 for tilted polygons). In practice, since \mathbf{W}_L^p is independent of the solid meshes, we can first assemble $\mathbf{F}_{e,p}$ only with F/S geometry and then multiply by \mathbf{W}_L^p (i.e., $\mathbf{F}_{e,p} \leftarrow \mathbf{F}_{e,p} \mathbf{W}_L^p$) to account for L/A geometry. Our derivation of $\mathbf{F}_{e,p}$ based on the application of pressure forces onto the solids ultimately leads to an operator that is essentially equivalent to what Zarifi and Batty [2017] derived from the divergence theorem; however, our derivation extends naturally to the more difficult staggered viscous stress terms for the $\mathbf{F}_{e,s}$ operator (including the “stress reduction” to eliminate one redundant diagonal stress component [Larionov et al. 2017; Takahashi and Batty 2020]) as we explain below.

Given the viscous stress $\mathbf{s} = (s_{xx}^T, s_{xy}^T, s_{xz}^T, s_{yz}^T, s_{zz}^T, s_{yy}^T)^T$, we wish to apply these stresses to the surface of the elastic solids. Since s_{xx} (as well as s_{yy} and s_{zz}) are defined at the center of the cell j , similar to \mathbf{p} , we can compute $\mathbf{F}_{e,s_{xx}}$ by

$$\mathbf{F}_{e,s_{xx} 3 \times i+0,j} = \frac{\mathbf{W}_{Lj}^s}{\Delta x} \sum_k \beta_{ik} A_k \mathbf{n}_k^T \mathbf{e}_x, \quad (9)$$

using the x-directional unit vector $\mathbf{e}_x = (1, 0, 0)^T$. The outward normal \mathbf{n}_k is projected onto \mathbf{e}_x because s_{xx} applies x-directional forces only (i.e., but not y- or z-directional). We can define $\mathbf{F}_{e,s_{yy}}$ and $\mathbf{F}_{e,s_{zz}}$ analogously. Considering the y-directional force with s_{yy} (which can be computed by $\mathbf{F}_{e,s_{yy}} s_{yy}$), the stress reduction can simply be performed by substituting $s_{yy} = -(s_{xx} + s_{zz})$, leading to the y-directional force of $-\mathbf{F}_{e,s_{yy}} (s_{xx} + s_{zz})$.

Unlike the diagonal viscous stress components (s_{xx}, s_{yy}, s_{zz}), the off-diagonal component s_{xy} is defined on the cell edges in 3D (cell nodes in 2D, see Figure 2), exerting both x- and y-directional forces. As the x-directional forces are applied to the elastic solids based on the y-directional derivative of s_{xy} [Batty and Bridson 2008], we can compute $\mathbf{F}_{e,s_{xy} 3 \times i+0,j}$ with the outward normal \mathbf{n}_k projected onto the y-directional unit vector $\mathbf{e}_y = (0, 1, 0)^T$ by summing up the contributions from all the polygons in the s_{xy} cell j as

$$\mathbf{F}_{e,s_{xy} 3 \times i+0,j} = \frac{\mathbf{W}_{Lj}^s}{2\Delta x} \sum_k \beta_{ik} A_k \mathbf{n}_k^T \mathbf{e}_y. \quad (10)$$

The scale factor $\frac{1}{2}$ appears to ensure consistency in the force exchanges between fluids and solids [Takahashi and Batty 2020]. Similarly, the y-directional forces can be computed based on the x-directional derivative of s_{xy} , and thus we have

$$\mathbf{F}_{e,s_{xy} 3 \times i+1,j} = \frac{\mathbf{W}_{Lj}^s}{2\Delta x} \sum_k \beta_{ik} A_k \mathbf{n}_k^T \mathbf{e}_x. \quad (11)$$

These derivations naturally extend to defining $\mathbf{F}_{e,s_{xz}}$ and $\mathbf{F}_{e,s_{yz}}$.

3.3 Viscous Liquid Formulation

The fluid velocity update with implicit integration of viscosity can be formulated as a minimization problem [Batty and Bridson 2008],

$$\mathbf{u} = \arg \min_{\mathbf{u}} \int \frac{1}{2} \left(\rho \|\mathbf{u} - \mathbf{u}^*\|_2^2 + 2\Delta t \eta \left\| \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} \right\|_F^2 \right) dV, \quad (12)$$

where \mathbf{u}^* and \mathbf{u} denote the fluid velocity before and after time integration, respectively, ρ the fluid density, η the dynamic viscosity, and $\|\cdot\|_F$ the Frobenius norm. The first term accounts for inertia and the second introduces viscosity by penalizing the deformation rate (omitted for inviscid liquids). We discretize this objective using a variational finite difference approach on a staggered grid [Goktekin et al. 2004] with volume fractions [Larionov et al. 2017; Takahashi and Lin 2019]. Similar to the L/A case, we can compute F/S volume fractions, i.e., \mathbf{W}_F^p and \mathbf{W}_S^p for pressure, \mathbf{W}_F^u and \mathbf{W}_S^u for velocity, and \mathbf{W}_F^s and \mathbf{W}_S^s for viscous stress. With the volume fractions, we define a fluid mass matrix, $\mathbf{M}_f = \hat{\mathbf{M}}_f \mathbf{W}_F^u \mathbf{W}_L^u$, where $\hat{\mathbf{M}}_f$ is a diagonal fluid density matrix, and a block diagonal viscosity matrix, $\mathbf{N} = \mathbf{H}^{-1} \hat{\mathbf{N}} (\mathbf{W}_F^s \mathbf{W}_L^s)^{-1}$, where $\hat{\mathbf{N}}$ is the diagonal dynamic viscosity matrix, and \mathbf{H} is the stress reduction matrix of Takahashi and Batty [2020]. Then, we can independently discretize (12) for both F/S and L/A domains and combine these forms into the discrete objective

$$E_f(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) = \frac{1}{2} \left(\|\mathbf{u} - \mathbf{u}^*\|_{\mathbf{M}_f}^2 + 2\Delta t \left\| \hat{\mathbf{f}}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) \right\|_{\mathbf{N}}^2 \right), \quad (13)$$

$$\hat{\mathbf{f}}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) = \mathbf{D}^T \mathbf{u} - \mathbf{F}_{e,s}^T \frac{\mathbf{x}_e - \mathbf{x}_e^t}{\Delta t} - \mathbf{F}_{r,s}^T \mathbf{v}_r, \quad (14)$$

where $\hat{\mathbf{f}}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r)$ denotes the deformation rate function, and \mathbf{D} is the discrete tensor divergence operator accounting for F/S and L/A volume fractions [Takahashi and Batty 2020, 2021]. Contributions from solid motion must be accounted for in the viscous deformation as solid boundary conditions, yielding the last two terms of (14).

3.4 Constraints for Liquids

3.4.1 Incompressibility Constraint. The incompressibility (divergence-free) constraint for fluid mass preservation is given as

$$\nabla \cdot \mathbf{u} = 0. \quad (15)$$

We can discretize this constraint in a manner that includes the effects of solid motion, defining \mathbf{d} as

$$\mathbf{d}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) = \mathbf{G}^T \mathbf{u} - \mathbf{F}_{e,p}^T \frac{\mathbf{x}_e - \mathbf{x}_e^t}{\Delta t} - \mathbf{F}_{r,p}^T \mathbf{v}_r = 0, \quad (16)$$

where \mathbf{G} denotes a discrete gradient operator incorporating F/S and L/A volume fractions. We enforce this condition as a hard constraint, treating fluid pressure \mathbf{p} as a Lagrange multiplier, and multiplying by Δt to treat \mathbf{p} as a force rather than an impulse.

3.4.2 Viscosity as a Compliant Constraint. In (13), we expressed the effect of viscosity as a penalty on the deformation rate, where η dictates the balance between fully rigid and fully inviscid. To put this in a form more consistent with our other (hard) constraints, we instead interpret it as a new hard constraint, but one that has been “softened” with an amount of *compliance* (see e.g., [Macklin and Muller 2021; Macklin et al. 2016; Servin et al. 2006; Tournier et al. 2015]) governed by η . Since fluid viscous stress is defined in the continuous setting as $\mathbf{s} = 2\eta \left(\frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} \right)$, we can introduce the corresponding discrete constraint as

$$\mathbf{f}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}) = \hat{\mathbf{f}}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) - \frac{1}{2} \mathbf{N}^{-1} \mathbf{s} = 0, \quad (17)$$

and enforce it using the viscous stress \mathbf{s} *itself* as the Lagrange multiplier (and once again scaling by Δt). Having separated out this new constraint, we redefine the fluid objective simply as:

$$E_f(\mathbf{u}) = \frac{1}{2} \|\mathbf{u} - \mathbf{u}^*\|_{\mathbf{M}_f}^2. \quad (18)$$

This interpretation simplifies warm starting and removes some extra computations, yet is exactly equivalent. (Compare the fluid terms of our final formulation (20) to Equation 9 of Larionov et al. [2017].)

3.5 Frictional Contact Constraint

We define frictional contact constraints, accounting for non-penetration and dissipation between solids (both rigid and elastic), as

$$\mathbf{h}(\mathbf{x}_e, \mathbf{v}_r) = -\alpha \mathbf{J}_e \frac{\mathbf{x}_e - \mathbf{x}_e^t}{\Delta t} - \alpha \mathbf{J}_r \mathbf{v}_r = 0. \quad (19)$$

To enforce this hard constraint, we treat the contact force \mathbf{c} as a Lagrange multiplier, scaled by Δt . Letting $\mathbf{c} = (\mathbf{n}^T, \mathbf{z}^T)^T$, where \mathbf{n} and \mathbf{z} are normal and tangential components of \mathbf{c} , respectively, the solution must satisfy the following conditions due to the Signorini-Coulomb contact law: $0 \leq \mathbf{n}$ and $\|\mathbf{z}_j\|_2 \leq \mu_j \mathbf{n}_j$ (where μ denotes the friction coefficient, and j is the contact index). This formulation accounts for the maximal dissipation principle and supports the take-off, stick, and slip cases of frictional contact [Daviet et al. 2011; Li et al. 2018].

3.6 Monolithic System

Assembling our complete monolithic objective for fluids and solids including the incompressibility, deformation rate penalty, and frictional contact constraints, we obtain the following mixed minimization/maximization problem:

$$\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}, \mathbf{p}, \mathbf{c} = \arg \min_{\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r} \arg \max_{\mathbf{s}, \mathbf{p}, 0 \leq \mathbf{n}, \|\mathbf{z}_j\|_2 \leq \mu_j \mathbf{n}_j} E(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}, \mathbf{p}, \mathbf{c}), \quad (20)$$

$$E(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}, \mathbf{p}, \mathbf{c}) = E_f(\mathbf{u}) + E_e(\mathbf{x}_e) + E_r(\mathbf{v}_r) + \Delta t \mathbf{s}^T \mathbf{f}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}) + \Delta t \mathbf{p}^T \mathbf{d}(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) + \Delta t \mathbf{c}^T \mathbf{h}(\mathbf{x}_e, \mathbf{v}_r). \quad (21)$$

While this optimization problem is linear and a constrained maximization with respect to the dual variables (Lagrange multipliers) \mathbf{s} , \mathbf{p} , and \mathbf{c} , we can find their optimality by extremizing the objective via the KKT conditions, alongside the minimization with respect to the primal variables \mathbf{u} , \mathbf{x}_e , and \mathbf{v}_r [Nocedal and Wright 2006]. As such, our aim in what follows is to minimize the objective $E(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}, \mathbf{p}, \mathbf{c})$ satisfying the KKT conditions.

3.7 Minimization with Our Custom Optimizer

Naturally, general purpose off-the-shelf nonlinear optimizers could be applied here (e.g., sequential quadratic programming (SQP) or interior point methods with inner solvers, such as gradient descent and L-BFGS [Nocedal and Wright 2006]). However, tailoring one’s solver strategy to the domain / problem at hand can often yield significant computational advantages. We therefore present a custom-designed optimizer for our monolithic problem, which seeks to efficiently find a local minimum.

3.7.1 Projected Newton Method. To minimize the objective while satisfying the KKT conditions, we employ a second-order Newton-type method [Nocedal and Wright 2006]. Considering a variable $\mathbf{x} = (\mathbf{u}^T, \mathbf{x}_e^T, \mathbf{v}_r^T, \mathbf{s}^T, \mathbf{p}^T, \mathbf{c}^T)^T$ and its corresponding descent direction $\Delta \mathbf{x}$, we can compute $\Delta \mathbf{x}$ by minimizing the quadratic approximation of the objective $\frac{1}{2} (\Delta \mathbf{x})^T \mathbf{A} \Delta \mathbf{x} - \mathbf{b}^T \Delta \mathbf{x}$ (where $\mathbf{A} = \nabla^2 E(\mathbf{x})$ and $\mathbf{b} = -\nabla E(\mathbf{x})$), or equivalently solving $\mathbf{A} \Delta \mathbf{x} = \mathbf{b}$, while updating and satisfying the corresponding box constraints [Takahashi and Batty 2021] (we define the cone constraints on \mathbf{z} based on \mathbf{n} from the previous time step or prior iterations). Appendix A explicitly provides the first and second order derivatives of (20).

To ensure $\Delta \mathbf{x}$ is a descent direction, we perform Hessian projection on the elastic potential $\nabla^2 \Psi(\mathbf{x}_e)$, defining the projected elastic potential as \mathbf{K} so that $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e} \left(= \frac{\alpha \mathbf{M}_e}{\Delta t^2} + \alpha \mathbf{K} \right)$ is guaranteed to be SPD [Kim and Eberle 2020; Smith et al. 2018].

For correctness, whenever the position of elastic solids \mathbf{x}_e is updated within our Newton iterations, we also update the corresponding contact information, SDFs, and volume fractions.

3.7.2 A Sparse SPD Reformulation. While the SPD-projected Hessian \mathbf{K} ensures that $\Delta \mathbf{x}$ is a valid descent direction, the entire Hessian $\nabla^2 E$ is still indefinite due to the Lagrange multipliers. One could solve the indefinite system to obtain $\Delta \mathbf{x}$ (e.g., [Robinson-Mosher et al. 2009, 2008]), but, all else being equal, SPD systems are preferable for numerical efficiency, and effective solvers for them are widely available. We therefore reformulate our system to be SPD.

One way to transform the symmetric indefinite system to be SPD is to first eliminate the primal variables ($\Delta \mathbf{u}$, $\Delta \mathbf{x}_e$, $\Delta \mathbf{v}_r$) by Schur complement substitutions, leaving only the dual variables (\mathbf{s} , \mathbf{p} , \mathbf{c}) to be solved. This manipulation is easily applied for rigid bodies in contact [Baraff 1996] or immersed in inviscid fluid [Batty et al. 2007; Klingner et al. 2006] as the only nontrivial matrix inversion required is the block diagonal rigid body mass matrix, \mathbf{M}_r . Unfortunately, doing so for elastic objects further requires explicit construction of $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e}^{-1}$ as a step towards forming the Delassus operator [Duriez et al. 2006; Raghupathi and Faure 2006]; this inversion is prohibitively expensive both to compute and store because $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e}$ is not (block) diagonal. Another alternative is to address this inversion as a nested linear solve (e.g., using Cholesky decomposition and forward/backward solves) [Overby et al. 2017]. However, this approach requires such inversions for every inner iteration of the surrounding iterative solver, and thus can also be rather expensive. The block matrix operations presented by Zarifi and Batty [2017] could also be used to form an SPD system. However, their reformulation introduces extra additions and multiplications of \mathbf{K} , which considerably deteriorates the conditioning of the system, therefore delaying the

convergence of iterative solvers. Similarly, forming an SPD system using the normal equations based on the least squares method and solving it using conjugate gradient (i.e., conjugate gradient on the normal equations (CGNR) [Saad 2003], also known as conjugate gradient least squares (CGLS)) can be extremely slow due to the resulting ill-conditioned system with squared condition number.

Below, we outline our approach, motivated by the equivalence between penalty and compliant constraints [Macklin and Muller 2021; Macklin et al. 2016; Tournier et al. 2015; Zhu et al. 2010]; it requires a matrix factorization only once per Newton step (i.e., outer iteration) and no sparse triangular solves, and in most cases retains conditioning comparable to the original indefinite system.

Consider a KKT system of the general form

$$\begin{bmatrix} M + BDB^T & C \\ C^T & -S^{-1} \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad (22)$$

where M and D are SPD (block-) diagonal matrices, C represents constraints, S^{-1} is an optional compliance matrix (and the inverse of a (block-) diagonal constraint stiffness matrix S), x , λ , a and b are vectors, and, in the case of elastic solids, BDB^T is a block-decomposition of the stiffness matrix K . We define a new variable $z = DB^T x$ to form an augmented system

$$\begin{bmatrix} M & C & B \\ C^T & -S^{-1} & 0 \\ B^T & 0 & -D^{-1} \end{bmatrix} \begin{bmatrix} x \\ \lambda \\ z \end{bmatrix} = \begin{bmatrix} a \\ b \\ 0 \end{bmatrix}. \quad (23)$$

Next we apply a Schur complement reduction [Benzi et al. 2005, §5]: substitute the expression $x = M^{-1}(a - C\lambda - Bz)$ from the first row into the lower rows (and negating), to get a smaller system,

$$\begin{bmatrix} S^{-1} + C^T M^{-1} C & C^T M^{-1} B \\ B^T M^{-1} C & D^{-1} + B^T M^{-1} B \end{bmatrix} \begin{bmatrix} \lambda \\ z \end{bmatrix} = \begin{bmatrix} -b + C^T M^{-1} a \\ B^T M^{-1} a \end{bmatrix}, \quad (24)$$

which is indeed both sparse and SPD. After solving, one can straightforwardly recover x . This SPD reformulation effectively replaces the unknown variables (x, λ) with unknowns (λ, z) , so the size of the system will in general be different. Robinson-Mosher et al. [2011, §5.3] proposed a similar transformation in the context of fluid viscosity terms, though our derivation is simpler and more general.

3.7.3 Factorizing K . A challenge with this approach is that an effective factorization of the BDB^T term is not always known a priori. A natural BDB^T factorization is (often) readily available for viscous fluids [Goldade et al. 2019], solid damping [Robinson-Mosher et al. 2011], and even for solid elasticity [Kim and Eberle 2020]. In the elasticity case, the SPD-projected Hessian K can be factorized with $B = \text{vec} \left(\frac{\partial F}{\partial x_e} \right)^T$ and $D = \text{vec} \left(\frac{\partial^2 \Psi(x_e)}{\partial F^2} \right)$, where vec denotes vectorization, F is the deformation gradient with $9 (= 3 \times 3)$ DOFs, and D is a block-diagonal matrix with block size 9×9 [Kim and Eberle 2020]. Unfortunately, the size of D is the product of the DOF count of F and the number of FEM elements (which outnumber FEM vertices by about $4 - 6 \times$ for tetrahedra [Doran 2013; Labelle and Shewchuk 2007]). Thus, the DOF count of the newly introduced variable $z (= DB^T x)$ in 3D becomes at least approximately $36N (= 9 \times 4N)$, for N FEM vertices; this number far exceeds $3N$

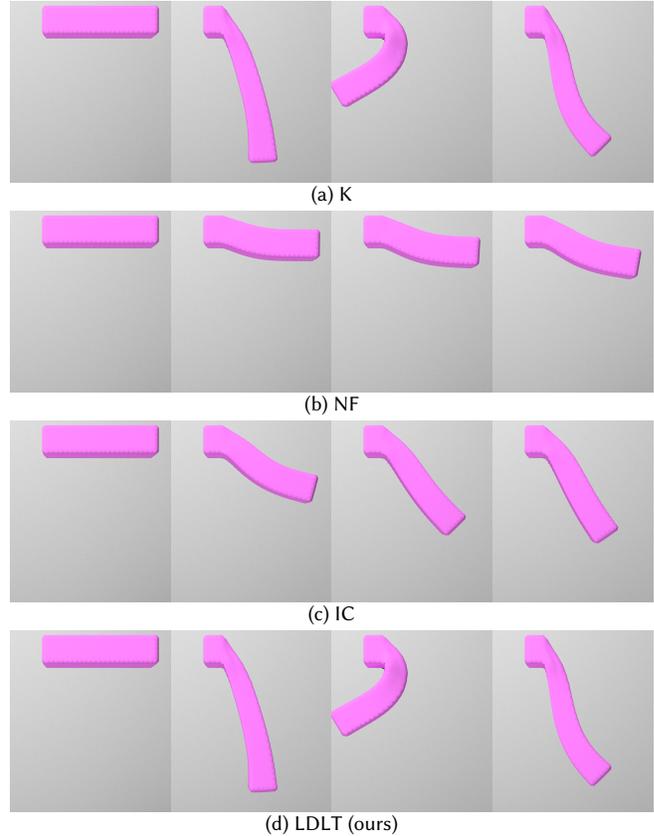


Fig. 3. An elastic beam hung from its fixed left end is simulated with different schemes (see §4.1.1). The schemes K and $LDLT$ (ours) generate comparable results preserving sufficient energy while NF and IC quickly dissipate energy leading to damped solid motions.

(which is the DOF count for the original variable x) leading to a much larger system.

One possible approach to reduce the number of DOFs of this natural factorization might be to ignore some eigenvalues in D and the corresponding blocks in B . While we can ignore eigenvalues clamped to 0 during the Hessian projection without compromising accuracy (we observed only around 15% of eigenvalues can be ignored in the scenario in Figure 3), removing positive eigenvalues makes BDB^T deviate from K , thereby slowing the convergence of Newton's method. Similarly, since the singular values and invariants (which have fewer DOFs compared to the deformation gradient) cannot exactly express K for stable Neo-Hookean materials in the form of BDB^T without further introducing matrix blocks [Kim and Eberle 2020; Smith et al. 2018, 2019] (unlike other materials, such as arigid-as-possible (ARAP) [Chao et al. 2010; Sorkine and Alexa 2007]), the approximation accuracy needs to be compromised, which negatively influences convergence. As a related scheme, Wang [2012] presented a Gauss-Newton-style quadratic approximation of the elastic potential to derive the BDB^T form, which similarly suffers from the large number of DOFs and slower convergence due to the less accurate Hessian. Another DOF reduction approach might be to employ hexahedral(-dominant) meshes as their element count

is proportional to the vertex count, leading to approximately $9N$ DOFs for z . However, compared to hexahedral meshes, tetrahedral meshes are much more extensively employed due to their simplicity and ease of mesh generation. Given these issues of the natural factorization, we instead rely on explicit LDLT factorization of \mathbf{K} .

To address our specific setting, we introduce a new unknown variable $\Delta \mathbf{g} = \mathbf{C}\mathbf{Q}^T \Delta \mathbf{x}_e$ using the LDLT decomposition of \mathbf{K} given by $\mathbf{K} = \mathbf{P}^T \hat{\mathbf{Q}} \mathbf{C} \hat{\mathbf{Q}}^T \mathbf{P} = \mathbf{Q} \mathbf{C} \mathbf{Q}^T$, where \mathbf{P} denotes a fill-reducing permutation matrix computed via approximate minimum degree reordering, $\hat{\mathbf{Q}}$ a lower triangular matrix, \mathbf{C} a diagonal matrix, and $\mathbf{Q} = \mathbf{P}^T \hat{\mathbf{Q}}$ (which is not lower triangular in general). This factorization lets us form the necessary augmented system, and then, because the diagonal blocks for $\Delta \mathbf{u}$, $\Delta \mathbf{x}_e$, and $\Delta \mathbf{v}_r$ become (block) diagonal mass matrices, we can eliminate these variables via Schur complement substitutions. For brevity, we relegate the details of our SPD reformulation of the complete monolithic system to Appendix B.

As the LDLT decomposition splits the square matrix \mathbf{K} into square matrices \mathbf{Q} and \mathbf{C} , the number of DOFs for $\Delta \mathbf{g}$ is exactly equal to that of $\Delta \mathbf{x}_e$, i.e., after eliminating $\Delta \mathbf{x}_e$ and introducing $\Delta \mathbf{g}$, the system size is the same. Since no box constraints are imposed on $\Delta \mathbf{x}_e$, the new variable $\Delta \mathbf{g}$ also has no box constraints. In addition, this SPD reformulation maintains the conditioning of the system at least for the solid part (since the eigenvalues of the new system are equivalent to the original eigenvalues scaled with D^{-1} , whose ratio between the minimum and maximum values is mostly within 100), and thus will not adversely affect the convergence rate of iterative solvers (see [Robinson-Mosher et al. 2011] for related discussions).

While Cholesky decomposition should in theory succeed when decomposing \mathbf{K} (as it is symmetric positive semidefinite), breakdown occurs frequently in practice due to small numerical errors (leading to sqrt applied to negative values) [Herholz and Alexa 2018]. Instead, our use of the LDLT decomposition allows us to identify any values in \mathbf{C} smaller than $\epsilon_c (= 10^{-10})$ and clamp them up to ϵ_c after the factorization.

The use of LDLT decomposition applied to \mathbf{K} might lead one to consider instead factorizing the *entire* original system (and then solving it with forward and backward substitution) since LDLT decomposition can handle symmetric indefinite systems. However, this approach is impractical due to the prohibitively large number of DOFs typically present in fluids, at least for scenarios we consider. Moreover, LDLT-based direct linear solvers are not applicable to our system because of the presence of non-negative and friction cone constraints imposed on the contact force \mathbf{c} .

With our SPD reformulation, the non-zero count can be larger than that of the original system due to fill-in and matrix multiplication. To reduce non-zeros for computational and memory efficiency, one might consider using an *incomplete* Cholesky decomposition, in exchange for compromising the accuracy of the projected Hessian \mathbf{K} . Unfortunately, we found that the resulting decomposition deviates too significantly from \mathbf{K} , leading to stagnation of Newton's method in 3D and thus significant energy loss (see Figure 3).

3.7.4 Box-Constrained Convex QP Solver. Due to numerical error, the \mathbf{A}_{cc} block in our final assembled system (32) (Appendix B) can be slightly indefinite in practice; we fix it to be SPD by scaling diagonal elements by $(1 + 10^{-3})$ [Takahashi and Batty 2020; Tan et al. 2012].

After this correction, the system matrix is strictly SPD, and our goal is to solve the box-constrained convex QP. We employ a variant of MPRGP presented by Takahashi and Batty [2021], which is an extension of the original MPRGP [Dostal and Schoberl 2005] and a fast active-set expansion technique [Kružík et al. 2020]. We apply incomplete Cholesky preconditioning to the sparse part of the system to accelerate convergence of MPRGP (similar to conjugate gradient (CG) [Shewchuk 1994]) and we address the low-rank dense submatrices $\mathbf{F}_{r,s}^T \mathbf{M}_r^{-1} \mathbf{F}_{r,s}$, $\mathbf{F}_{r,s}^T \mathbf{M}_r^{-1} \mathbf{F}_{r,p}$, and $\mathbf{F}_{r,p}^T \mathbf{M}_r^{-1} \mathbf{F}_{r,p}$ (caused by rigid body coupling) using the sequential multiplication (SM) approach [Takahashi and Batty 2020].

3.7.5 Warm Starting. Initializing the Lagrange multipliers (i.e., dual variables $\mathbf{s}, \mathbf{p}, \mathbf{c}$) to 0 in each Newton iteration is essentially equivalent to restarting the optimization from scratch. In addition, using values far from the solution slows down progress towards convergence or even makes Newton's method diverge. To ensure and accelerate convergence, we warm start using $\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}, \mathbf{p}, \mathbf{c}$ from the previous simulation step (or Newton iteration) as their initial values. We emphasize that since the contact information (pair of contacting solids, contact position, normal, contact force, etc.) is updated along with the elastic solid positions in each Newton iteration, we need to explicitly track contact information for warm starting. When previous results are unavailable (e.g., outside of liquid domains or for new contacts), we initialize them to 0.

3.7.6 Line Search. While the theoretically ideal step size for the Newton iteration is 1, it is typically necessary to use a smaller step size due to the Hessian projection and nonlinearity of the elastic potential. In addition, since our monolithic system (20) is a mixed minimization/maximization, it is not possible to simply perform line search with the objective $E(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r, \mathbf{s}, \mathbf{p}, \mathbf{c})$ to find a local minimum. Notably, prior mixed minimization/maximization formulations of unsteady Stokes flow [Larionov et al. 2017] and linearized implicit surface tension [Misztal et al. 2014] are free of this issue because a single linear solve suffices to achieve optimality without line search, owing to their quadratic objectives under linear equality constraints.

Therefore, we define a merit (objective) function for the line search $L(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r)$ based purely on kinetic energy minimization, with only primal variables, as

$$L(\mathbf{u}, \mathbf{x}_e, \mathbf{v}_r) = L_f(\mathbf{u}) + L_e(\mathbf{x}_e) + L_r(\mathbf{v}_r), \quad (25)$$

$$L_f(\mathbf{u}) = \frac{1}{2\alpha} \|\mathbf{u}\|_{\mathbf{M}_f}^2, \quad (26)$$

$$L_e(\mathbf{x}_e) = \frac{1}{2\Delta t^2} \|\mathbf{x}_e - \mathbf{x}_e^*\|_{\mathbf{M}_e}^2 + \Psi(\mathbf{x}_e), \quad (27)$$

$$L_r(\mathbf{v}_r) = \frac{1}{2} \|\mathbf{v}_r\|_{\mathbf{M}_r}^2, \quad (28)$$

where the merit function for fluids $L_f(\mathbf{u})$ is scaled by $1/\alpha$ to ensure energy consistency regardless of grid resolution. Notably, we can use $L_f(\mathbf{u})$ defined above for both inviscid and viscous liquids. The equivalence between our formulation (20) and the one based on kinetic energy minimization is spelled out in Appendix C (for strong two-way coupling of inviscid liquids and rigid bodies).

Although the dual variables are not used to find the step size in the line search, the descent direction $\Delta \mathbf{x}$ correctly computed based on the KKT condition ensures a valid update of the dual variables

given their linearity. In practice, Newton iterations at the early phase are almost always performed with a step size of 1, and thus the hard constraints are correctly satisfied.

To ensure the decrease of the objective L , we employ backtracking line search [Nocedal and Wright 2006]. For this purpose, the norm of the gradient, $\|\nabla L\|$ (or $\|\nabla E\|$), has sometimes been used as an alternative to L in past work (since L cannot be defined for complementarity-based formulations [Macklin et al. 2019a]); however, the value of L and $\|\nabla L\|$ are essentially unrelated [Nocedal and Wright 2006]. To enable a valid line search, it is therefore essential to formulate the optimization problem such that L is well-defined.

In practice, updating the positions of elastic solids can change which Eulerian fluid DOFs are valid (even though fluid particle positions are fixed during the optimization) and introduces some discontinuity that invalidates the value of L computed in the previous Newton iteration. Thus, after we evaluate valid DOFs, we compute L so that line search can be performed.

3.7.7 Algorithm of Our Custom Optimizer. Algorithm 2 summarizes the steps of our tailor-made solver for the constrained monolithic optimization problem (20).

We terminate the Newton iterations based on the infinity norm of the Newton step direction $\|\Delta \mathbf{x}^k\|_\infty$ [Takahashi and Batty 2021] or a maximum iteration count k_{\max} . In addition, we also end the Newton iterations if we cannot make progress with the line search (due to the discontinuity coming from the liquid DOF changes), i.e., when line search iteration count l equals the max line search count l_{\max} .

Algorithm 2 Our custom optimizer

```

1:  $k = 0$ 
2: do
3:   Compute fluid/solid and liquid/air domains
4:   Evaluate volume fractions
5:   Detect collisions between solids
6:   Evaluate  $L^k$ 
7:   Assemble the system
8:    $(\Delta \mathbf{s}^{k+1}, \Delta \mathbf{p}^{k+1}, \Delta \mathbf{c}^{k+1}, \Delta \mathbf{g}^{k+1}) = 0$ 
9:   Solve the SPD system with box constraints (32)
10:  Recover  $\Delta \mathbf{u}^{k+1}, \Delta \mathbf{x}_e^{k+1}, \mathbf{v}_r^{k+1}$ 
11:   $\Delta \mathbf{x}^{k+1} = (\Delta \mathbf{u}^{k+1}, \Delta \mathbf{x}_e^{k+1}, \Delta \mathbf{v}_r^{k+1}, \Delta \mathbf{s}^{k+1}, \Delta \mathbf{p}^{k+1}, \Delta \mathbf{c}^{k+1})$ 
12:  Compute  $\mathbf{x}^{k+1}$  and  $l$  via line search with  $\Delta \mathbf{x}^{k+1}$  and  $L^k$ 
13:   $k = k + 1$ 
14: while  $\epsilon < \|\Delta \mathbf{x}^k\|_\infty$  and  $k < k_{\max}$  and  $l < l_{\max}$ 
15: return  $\mathbf{u}^k, \mathbf{x}_e^k, \mathbf{v}_r^k, \mathbf{s}^k, \mathbf{p}^k, \mathbf{c}^k$ 

```

4 RESULTS AND DISCUSSIONS

Our method is implemented in C++17 and parallelized using OpenMP. All examples used adaptive timestepping with CFL numbers between 0.5 and 5.0 (empirically chosen based on visual quality), with 50 frames per second. For simplicity, we employ the pyramid approximation for friction constraints with friction coefficient $\mu = 0.5$. For inviscid liquid simulations, we used cut-cell-based area weighting

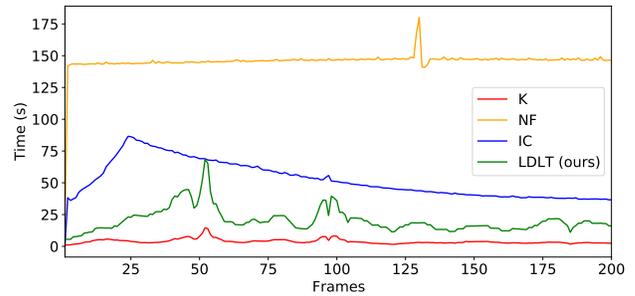


Fig. 4. Profiles of the total time to solve the elasticity optimization problem per frame for Figure 3. LDLT (ours) is more expensive than the baseline K given the absence of the contact constraints, but is faster than NF and IC.

Table 1. Simulation settings and results for Figure 3. The system size is denoted by n , the number of non-zeros by z , and the condition number by κ . The averaged Newton and MPRGP iteration counts per frame are denoted by N_{iter} and M_{iter} , respectively. The total time (s) per frame is denoted by T and the relative overhead with respect to the baseline K is denoted by r .

Scheme	n	z	κ	N_{iter}	M_{iter}	T	r
K	5.6k	186.8k	2.7×10^3	10.7	1.1k	3.8	
NF	79.6k	5,838.0k	1.5×10^{11}	10.0	1.0k	145.7	38.3
IC	5.6k	712.8k	3.1×10^2	100.0	2.3k	52.3	13.8
LDLT (ours)	5.6k	7,169.1k	1.4×10^3	10.5	0.5k	20.3	5.3

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. To mitigate position-level errors, we employ Baumgarte stabilization for solid contacts [Baumgarte 1972] (except for Figure 5), and use particle-level position correction once per frame [Macklin et al. 2014]. While we use precomputed SDFs for rigid bodies, we recompute SDFs for elastic solids based on the work of Fisher and Lin [2001] if the query point is inside of the solids, and otherwise, we compute the shortest distance to the nearest surface triangle. To accelerate SDF updates, collision detection, and element and triangle look-up, we use a bounding volume hierarchy. We detect solid contacts by checking whether solid surface vertices are inside of the other solids based on SDFs with a small collision margin. For Newton’s method, we use up to two iterations unless otherwise stated. To solve box-constrained convex QPs or SPD linear systems as inner problems, we employ MPRGP with a termination relative residual of 10^{-8} and maximum iteration count of 1,000 (setting $-\infty$ and ∞ as lower and upper bounds, respectively, if box constraints are unnecessary) unless otherwise stated. We used Symmetric Successive Over Relaxation (SSOR) preconditioning for Figure 3 and 5 as SSOR was more efficient than incomplete Cholesky (IC) preconditioning in these scenes, and in Figure 7 for comparison purposes. We executed all our simulations using “e2-standard-8” (8 cores with 32GB RAM) provided by Google Compute Engine.

4.1 Elastic Solid

4.1.1 SPD Reformulation. As a basic test to evaluate the efficacy of our SPD reformulation and justify the choice of the explicit LDLT decomposition, we experimented with an elastic beam (2.3k vertices,

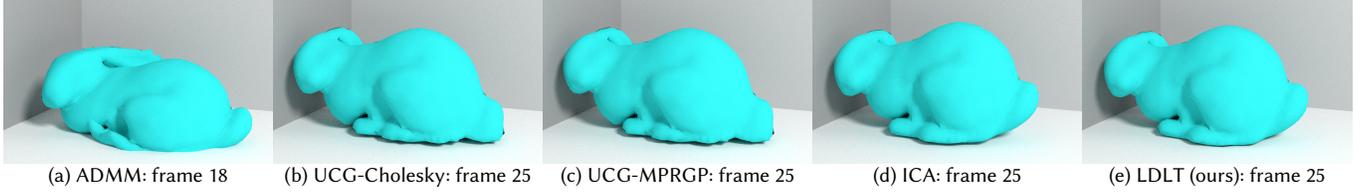


Fig. 5. An elastic bunny dropped onto a ground, simulated with different schemes (see §4.1.2). The ADMM scheme fails to satisfy the hard contact constraints, UCG-Cholesky and UCG-MPRGP do not support the traditional frictional contact constraints, while ICA and LDLT (ours) successfully generate natural contact response. LDLT (ours) is at least 6.2 \times faster than ICA.

8.8k elements, Young’s modulus $e = 10^4$ (Pa), Poisson’s ratio $\nu = 0.49$, and density 500 kg/m^3) with its left part fixed while excluding contacts, as shown in Figure 3. We evaluate the following four schemes:

- (1) K: baseline scheme using the projected Hessian \mathbf{K} [Kim and Eberle 2020; Smith et al. 2018], without applying our SPD reformulation;
- (2) NF: the natural factorization with $\mathbf{K} = \mathbf{BDB}^T$ is used to perform our SPD reformulation, computing \mathbf{D}^{-1} directly from the eigenvalues and eigenvectors [Kim and Eberle 2020; Smith et al. 2018], without calculating 9×9 block inversions;
- (3) IC: incomplete Cholesky factorization with no fill-in, approximating \mathbf{K} by $\mathbf{K} \approx \mathbf{BDB}^T$, is used to perform our SPD reformulation;
- (4) LDLT (ours): our LDLT-based SPD reformulation.

Because IC compromises the accuracy of \mathbf{K} , the Newton descent direction $\Delta \mathbf{x}$ does not correctly account for the residual (i.e., ∇E). As such, in this comparison, we use $\|\nabla E\|_\infty$ as a consistent termination criterion for Newton iterations, and use 10^3 as a threshold (which approximately corresponds to a relative residual of 10^{-4}). To finish the simulation within a reasonable time, we terminate the Newton iterations at a maximum iteration count of 10 for NF and 100 for IC while the schemes K and LDLT (ours) always converged within 100 iterations. Figure 4 compares profiles of computational costs, and Table 1 summarizes the simulation settings and averaged results over 200 frames.

For the tested scene, the baseline K is the fastest among these schemes due to its smaller system size, smaller number of non-zeros, and low condition number. However, if contacts were to become involved, the K scheme would need to address indefinite systems, unlike the other schemes under our SPD reformulation. While NF is sparser and has fewer non-zeros than ours, the system size is much larger, making the inner solver almost always fail to converge and thus giving an insufficiently accurate $\Delta \mathbf{x}$. Consequently, the minimization was not fully solved leading to damped motion with significant energy loss. Although the IC scheme can solve the inner problem quickly due to its smaller system size, smaller number of non-zeros, and low condition number, the resulting $\Delta \mathbf{x}$ is relatively inaccurate, as \mathbf{K} is only roughly approximated by the incomplete Cholesky decomposition. As a result, the optimization was also not sufficiently solved, leading again to damped motion. While our LDLT-based approach generally has more non-zeros, the system size is still small with a low condition number, and our inner solver was

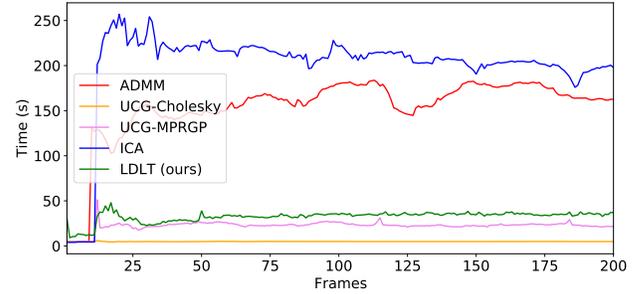


Fig. 6. Profiles of the total time for the elasticity-contact handling per frame for Figure 5. LDLT (ours) is slower than UCG-Cholesky and UCG-MPRGP which do not support the traditional contact constraints, but is significantly faster than ADMM and ICA.

Table 2. Simulation setting and results for Figure 5. c denotes the number of contacts, n system size, z the number of non-zeros, separating numbers as primal/dual for ICA. N_{iter} and M_{iter} denote the averaged outer (Newton or ADMM) and MPRGP iteration counts per frame, respectively. I_{iter} denotes iteration counts for UCG or ICA iterations. T denotes the total time (s) per frame.

Scheme	c	n	z	N_{iter}	I_{iter}	M_{iter}	T
ADMM	183.2	15.1k	555.1k	47.8		10.7k	153.9
UCG-Cholesky	422.1	16.4k	557.3k	1.0	99.4		4.9
UCG-MPRGP	422.1	16.4k	557.3k	1.0	109.3	20.6k	22.4
ICA	326.5	15.1k/1.0k	555.0k/1.0k	1.0	1.4k	215.3k/112.4k	200.2
LDLT (ours)	492.8	16.6k	31.1M	1.0		422.0	32.5

able to quickly compute $\Delta \mathbf{x}$. In addition, since $\Delta \mathbf{x}$ has the same accuracy as for scheme K, the Newton iteration also quickly converges, giving significantly better performance compared to the other SPD-reformulated schemes. The resulting behavior with our method is comparable to the baseline K, though our method is somewhat more expensive (5.3 \times), given the absence of contact constraints in this scenario.

4.1.2 Elastic Solid Contacts. To evaluate the efficiency of our SPD reformulation in contact handling with elastic solids, we experimented with an elastic bunny (5.0k vertices and 20.0k elements, Young’s modulus $e = 10^4$ (Pa), Poisson’s ratio $\nu = 0.49$, and density 500 kg/m^3), as shown in Figure 5. We evaluate the following five schemes:

- (1) ADMM: the baseline scheme K [Kim and Eberle 2020; Smith et al. 2018] combined with hard contact constraints handled via ADMM [Narain et al. 2016];
- (2) UCG-Cholesky: Uzawa conjugate gradient, which solves the KKT system ignoring box constraints, calculating $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e}^{-1}$ via precomputed Cholesky-based linear solve [Overby et al. 2017];
- (3) UCG-MPRGP: same as UCG above except $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e}^{-1}$ is addressed via MPRGP-based linear solve;
- (4) ICA: iterative constraint anticipation [Otaduy et al. 2009], which iteratively solves primal and dual problems using the diagonal approximation to form the Delassus operator;
- (5) LDLT (ours): our LDLT-based SPD reformulation.

To perform comparisons as consistently as possible, we use these schemes (except for ADMM) to solve the KKT system within the Newton iteration, and use only one Newton iteration, since our primary focus is to evaluate how these schemes efficiently enforce the contact constraints (even with early termination) given the critical importance of resolving contacts for stability. With the ADMM scheme, as it is necessary to perform sufficiently many ADMM iterations to enforce the hard constraints [Overby et al. 2017], we use up to 50 ADMM iterations (with manually adjusted ADMM parameters for faster convergence). In addition, following Narain et al. [2016], we treat the contacts as free-slip constraints to avoid further delay in convergence [Kaufman et al. 2008]. Since UCG is not designed to handle inequality constraints [Overby et al. 2017], we treat the contacts as sticky contacts for UCG. For ICA, we employ MPRGP instead of PGS (which was used by Otaduy et al. [2009]) to solve the inner primal/dual problems (as PGS was much slower), and we use up to 1,500 ICA iterations. In addition, we found that the (block-)diagonal approximation of $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e} \left(= \frac{\alpha \mathbf{M}_e}{\Delta t^2} + \alpha \mathbf{K} \right)$ using block GS decomposition (instead of elementwise GS) was unstable with both PGS and MPRGP, and therefore we use $\frac{\alpha \mathbf{M}_e}{\Delta t^2}$ to approximate the Delassus operator. Figure 6 compares profiles of the computational cost, and Table 2 summarizes the simulation settings and averaged results over 200 frames.

While the ADMM scheme preserves more energy of the bunny due to the large number of ADMM iterations compared to the other schemes, the non-penetration constraint is still violated because ADMM needs to fully converge to enforce the hard constraints [Overby et al. 2017]. The result is obvious ground penetrations. Given the first-order convergence rate of ADMM, many more iterations (along with system assembly) are necessary for convergence. As our method (with its second-order convergence rate) is already around 4.7× faster at this stage, we believe that our method is advantageous. Although both UCG-Cholesky and UCG-MPRGP converged faster than ours in this example, UCG is essentially not designed to handle box constraints and thus cannot support the traditional contact forces due to the Signorini-Coulomb contact law (note the bunny tail stuck to the ground). By contrast, our reformulation transforms the indefinite KKT system to SPD form, making it possible to handle the box constraints via MPRGP. While ICA can also handle the box constraints via MPRGP, decomposing the indefinite KKT system into the SPD primal and dual problems via the diagonal approximation significantly slows down the convergence of ICA, so it frequently fails to converge. Consequently, our method

Table 3. Simulation settings and results for Figure 7. The system size is denoted by n , the number of non-zeros by z , the condition number by κ , and the ratio of the condition number with respect to the original indefinite system by r . The averaged MPRGP iteration counts per frame are denoted by M_{iter} . The total time (s) per frame is denoted by T .

Scheme	n	z	κ	r	M_{iter}	T
Z1-Z2	179.8k	2.1M	6.8×10^{18}	3.5×10^3	7,255.6	270.7
LDLT SSOR	177.7k	8.8M	2.9×10^{11}	1.5×10^{-4}	3,511.4	166.2
LDLT IC (ours)	177.6k	8.7M	2.9×10^{11}	1.5×10^{-4}	599.0	58.0

is at least around 6.2× faster than ICA. Notably, as our method is solving for the dual variable \mathbf{c} but not primal variable \mathbf{x}_e , the bunny can sink gradually due to the accumulated numerical error; this can be prevented with a position-level feedback, e.g., using Baumgarte stabilization [Baumgarte 1972].

4.2 Two-Way Fluid-Solid Coupling

We next evaluate our SPD reformulation for two-way fluid-solid coupling (in the absence of solid-solid contacts) by placing an elastic beam (2.2k vertices and 9.0k elements, Young’s modulus $e = 10^4$ (Pa), Poisson’s ratio $\nu = 0.49$, and density 200 kg/m³) into a dambreak scenario with fluid density of 1,000 kg/m³, grid resolution of 80³, and 967.7k particles, as shown in Figure 7. We compare our method with the method of Zarifi and Batty [2017], which transforms the indefinite system into a different SPD form via the decomposition of $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e} \left(= \frac{\alpha \mathbf{M}_e}{\Delta t^2} + \alpha \mathbf{K} \right)$ into \mathbf{Z}_1 and \mathbf{Z}_2 as $\mathbf{H}_{\mathbf{x}_e \mathbf{x}_e} = \mathbf{Z}_1 + \mathbf{Z}_2$, where $\mathbf{Z}_1 = \sigma \frac{\alpha \mathbf{M}_e}{\Delta t^2}$ and $\mathbf{Z}_2 = (1 - \sigma) \frac{\alpha \mathbf{M}_e}{\Delta t^2} + \alpha \mathbf{K}$, and $\sigma (0 < \sigma \leq 1)$ is a blending parameter. We use $\sigma = 0.9$ suggested in their paper as we found that there is no clear performance benefit in using a smaller σ (although it leads to a slightly better conditioned system). We call their method Z1-Z2. In this comparison, we perform only one Newton iteration, use up to 5,000 MPRGP iterations, and evaluate IC preconditioning for the Z1-Z2 approach and both IC and SSOR preconditioning for our formulation. Figure 8 compares profiles of computational costs and Table 3 summarizes the simulation settings and averaged results over 200 frames.

As the SPD reformulation via Z1-Z2 involves addition and multiplication of \mathbf{Z}_2 , the conditioning of the resulting SPD system is deteriorated ($\kappa = 6.8 \times 10^{18}$), and the number of non-zeros increased ($z = 2.1\text{M}$) compared to the original indefinite system ($\kappa = 2.0 \times 10^{15}$ and $z = 1.2\text{M}$) [Zarifi and Batty 2017]. Consequently, IC preconditioned MPRGP fails to converge or requires many iterations. By contrast, our reformulation keeps the condition number lower ($\kappa = 2.9 \times 10^{11}$), and IC preconditioned MPRGP quickly converged despite the larger number of non-zeros ($z = 8.7\text{M}$), achieving a performance gain of 4.7×. Notably, the increased number of non-zeros using our SPD reformulation is moderate compared to the solid-only scenario in Sec. 4.1.1 (see Table 1) because our reformulation involves only the sub-matrices related to elastic solids (\mathbf{A}_{gp} , \mathbf{A}_{pg} , and \mathbf{A}_{gg}), while the fluid pressure block \mathbf{A}_{pp} (which is the largest component in the original system) remains untouched. By contrast, SSOR preconditioning with our formulation was not effective enough and almost always failed to converge in this two-way fluid-solid coupling scenario; IC preconditioned MPRGP was at least 2.9× faster.



Fig. 7. An elastic beam (with its top fixed) deformed in a dam break scenario, simulated with different schemes (see §4.2). These approaches generate comparable visual results, but ours is at least $4.7\times$ and $2.9\times$ faster than Z1-Z2 and LDLT SSOR, respectively.

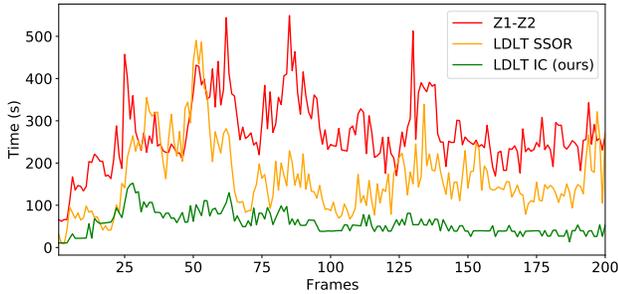


Fig. 8. Profiles of the total time for the entire pressure-elasticity handling phase per frame for Figure 7. Our method outperforms Z1-Z2 and LDLT SSOR at almost all the frames.

Table 4. Simulation settings and results for Figure 9. The system size is denoted by n , the number of non-zeros by z , the number of UCG or block GS iterations by G_{iter} , the number of MPRGP iterations by M_{iter} , and total time (s) per frame by T . (P/C) indicates the numbers associated with the pressure/contact handling.

Scheme	n (P/C)	z (P/C)	G_{iter}	M_{iter}	T (P/C)	T
PE-EC	35.9k/11.2k	17.0M/16.9M		211.6/244.7	80.4/50.0	130.4
EC-PE	35.8k/11.2k	17.0M/16.9M		204.3/397.5	81.6/91.3	173.0
PEC UCG	118.8k	824.3k	5,414.2			77.8
PEC LDLT GS	35.7k	19.3M	80.6	841.4/19,252.1		1062.9
PEC LDLT (ours)	35.9k	19.3M		1,900.1		253.2

4.3 Contact-Aware Coupling with Inviscid Liquids

Our next scenario examines our monolithic coupling approach when frictional contacts occur. We demonstrate interactions among inviscid liquids (density $1,000 \text{ kg/m}^3$), an elastic bowl (3.7k vertices and 12.8k elements, Young's modulus $e = 3 \times 10^4$ (Pa), Poisson's ratio $\nu = 0.49$, and density 200 kg/m^3), and the rigid ground, using a grid resolution of $128 \times 64 \times 128$ and 52.3k particles, as shown in Figure 9. We evaluate the following five possible pressure-elasticity-contact coupling schemes:

- (1) PE-EC: pressure-elasticity solve followed by elasticity-contact solve (once);
- (2) EC-PE: elasticity-contact solve followed by pressure-elasticity solve (once);
- (3) PEC UCG: unified pressure-elasticity-contact solve with UCG applied to the indefinite system (61);

- (4) PEC LDLT GS: unified pressure-elasticity-contact solve using our SPD reformulation and iterative block GS pressure and elasticity-contact solve;
- (5) PEC LDLT (ours): unified pressure-elasticity-contact solve with our SPD reformulation.

We consider elasticity in both subproblems within PE-EC and EC-PE schemes and apply our reformulation to make the subproblems SPD. Within the UCG iterations of PEC UCG (see (61) in Appendix D), while we directly invert the diagonal fluid mass matrix M_f , we use the Cholesky-based linear solve to address the inversion of $H_{x_e x_e}$ (as it is faster than MPRGP-based linear solve, see Table 2). PEC LDLT GS is equivalent to our proposed PEC LDLT (ours) except that the inner SPD system is decomposed into pressure and elasticity-contact subproblems and addressed using block GS (as splitting elasticity and contact handling significantly delays convergence [Ostaduy et al. 2009]) with up to 15 GS iterations to finish the simulation within a reasonable time (while full convergence typically requires more than 100 GS iterations). In this example, we use a termination relative residual of 10^{-6} for MPRGP. We summarize performance numbers in Table 4 and compare profiles of total time in Figure 10.

PE-EC completely failed to handle the bowl-ground contacts because the concluding elasticity-contact solve could not make any progress due to a failure of the line search; this failure resulted from a significant violation of contact constraints due to the update of the solid positions after the contact-oblivious pressure-elasticity solve. EC-PE also failed because the result of the first elasticity-contact solve was completely spoiled by the concluding pressure-elasticity solve. PEC UCG can efficiently prevent the penetration of the elastic bowl into the ground, but it only supports sticky contact constraints. Consequently, the lower parts of the elastic bowl continue sticking to the ground. While PEC LDLT GS supports the proper contact constraints, it almost always failed to converge due to the slow convergence of block GS and was very costly. PEC LDLT (ours) was able to efficiently and correctly handle the liquid flow, elastic deformations, two-way liquid-solid interactions, and bowl-ground contacts, achieving at least $4.2\times$ faster performance versus PEC LDLT GS.

4.4 Contact-Aware Coupling with Viscous Liquids

We next demonstrate our full monolithic formulation for simultaneous coupling of a *viscous* liquid (density $1,000 \text{ kg/m}^3$ and viscosity $100 \text{ kg/(s} \cdot \text{m)}$), an elastic beam (2.3k vertices and 8.8k elements, Young's modulus $e = 1 \times 10^4$ (Pa), Poisson's ratio $\nu = 0.49$, and density 200 kg/m^3), and rigid ground, using a grid resolution of 64^3 and 52.5k particles in Figure 11. We compare our method PVEC LDLT (ours) against PVEC UCG, which solves the viscous stress-eliminated indefinite system with SPD top-left block matrices to enable adopting UCG (see (62) in Appendix D) but not the full indefinite system (31) with indefinite top-left blocks due to $-\frac{\Delta t}{2} N^{-1}$. We invert the 2×2 top-left block matrices in (62) using MPRGP since the Cholesky factorization is not applicable to these blocks due to their excessive size. We summarize performance numbers in Table 5 and compare profiles of total time in Figure 12.

For PVEC UCG, we need a relatively more expensive MPRGP-based linear solve to invert the SPD top-left block matrices, in contrast to the case for inviscid liquids where we can use a more efficient

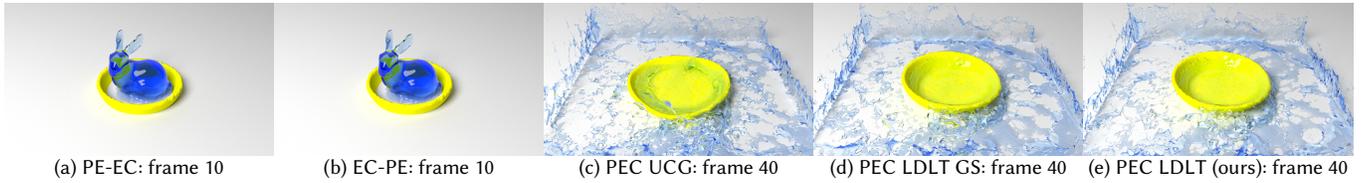


Fig. 9. A bunny-shaped inviscid liquid volume and an elastic bowl dropped onto the ground, simulated with different schemes (see §4.3). PE-EC fails to handle bowl-ground contacts as the last elasticity-contact solve cannot recover the significant penetration due to the first pressure-elasticity solve. EC-PE also fails as the first elasticity-contact handling is spoiled by the last pressure-elasticity solve. While PEC UCG can efficiently prevent the bowl-ground penetration, the traditional contact handling cannot be supported. Although PEC LDLT GS can correctly handle the scene, it is quite slow due to the blockwise decomposition. PEC LDLT (ours) can more efficiently address this coupling scenario, achieving at least a 4.2× performance gain over PEC LDLT GS.

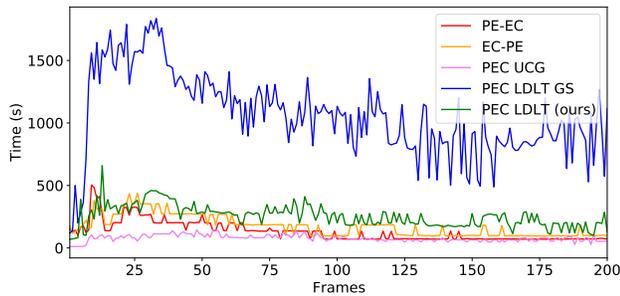


Fig. 10. Profiles of the total time for the entire pressure-elasticity-contact handling per frame for Figure 9. While PEC LDLT (ours) is slower than PEC UCG (which does not support the traditional contact handling), PEC LDLT (ours) is significantly more efficient than PEC LDLT GS.

Table 5. Simulation settings and results for Figure 11. The system size is denoted by n , the number of non-zeros by z , the number of UCG iterations by G_{iter} , the number of MPRGP iterations by M_{iter} , and total time (s) per frame by T .

Scheme	n	z	G_{iter}	M_{iter}	T
PVEC UCG	57.0k	1.2M	1,537.3	117.6k	632.0
PVEC LDLT (ours)	84.7k	19.4M		559.4	84.4

precomputed Cholesky-based linear solve. As this MPRGP-based linear solve is required in each UCG iteration, the total computational cost is quite expensive. In addition, PVEC UCG does not support the traditional contact constraints, and thus the elastic beam sticks to the ground and never separates even after viscous fluids flow from the top of the beam to its side. By contrast, PVEC LDLT (ours) can more efficiently support the coupling of the viscous fluids, elastic beam, and the ground without artifacts, achieving a 7.5× performance gain over PVEC UCG.

4.5 Complex Examples

We further demonstrate our method in a few more complex scenarios. Figure 13 compares interactions of liquids and a bowl with different material combinations (inviscid or viscous for the liquids and rigid or elastic for the bowl) dropped onto a rigid torus fixed in the air. We use liquid density $1,000 \text{ kg/m}^3$ (and viscosity $100 \text{ kg/(s} \cdot \text{m)}$ for viscous fluids), solid density 200 kg/m^3 (and 2.3k vertices, 8.8k

elements, Young’s modulus $e = 10^4$ (Pa), Poisson’s ratio $\nu = 0.49$ for the elastic bowl), and a grid resolution of $160 \times 80 \times 160$ with 101.7k particles. The total simulation time per frame was 8.6 s for (a) Inviscid and rigid, 352.0 s for (b) Inviscid and elastic, 12.4 s for (c) Viscous and rigid, and 574.9 s for (d) Viscous and elastic. For (a) and (c), the rigid bowl settles on top of the rigid torus supporting the load of liquid. By contrast, for (b) and (d), the elastic bowl deforms due to the liquid load, squeezes through the torus, and collides with the ground. Figure 14 shows the performance breakdown for (c) and (d) to illustrate the details of the computational cost.

Figure 1 demonstrates three-way coupling of bunny-shaped inviscid (left) or viscous (right) liquid volumes, multiple rigid bunnies, and an elastic sheet. This simulation used a liquid density of $1,000 \text{ kg/m}^3$ (and viscosity $100 \text{ kg/(s} \cdot \text{m)}$ for viscous fluids) and 3.9k vertices, 12.2k elements, Young’s modulus $e = 10^4$ (Pa), Poisson’s ratio $\nu = 0.49$ for the elastic sheet, a solid density of 200 kg/m^3 for the sheet and rigid bunnies, a grid resolution of 80^3 , and 52.5k particles. The total simulation time per frame was 225.3 s (left) and 211.7 s (right).

4.6 Discussions

4.6.1 Block Decomposition. In our experiments, it is typically advantageous to solve the system in a unified way rather than separately and iteratively solving the subcomponents of the system (e.g., using block GS), because our system components (pressure, viscous stress, elasticity, and contact forces) are algebraically tightly coupled. However, if a particular decomposition of the system makes it possible to efficiently solve the subcomponents [Bouaziz et al. 2014; Overby et al. 2017; Takahashi and Batty 2021], such decomposition approaches could be an effective alternative with sufficient outer iterations. Thus, it is worthwhile investigating how unified systems should be decomposed to maximize performance.

4.6.2 Time Step Size. While our optimization-based solver enables taking much larger time step sizes (e.g., CFL numbers of 5.0 or more), larger time steps do not necessarily lead to better overall performance and quality; more Newton iterations are often required, along with more inner solver iterations [Macklin et al. 2019b], additional SDF and volume fraction evaluations, and repeated contact updates [Otaduy et al. 2009]. In addition, energy dissipation for both fluids and solids can become significant. Instead, taking just one Newton iteration with a smaller time step can sometimes be more efficient and effective [Baraff and Witkin 1998; Kim and Eberle 2020]. Nonetheless, we believe that in practice, it is preferable for

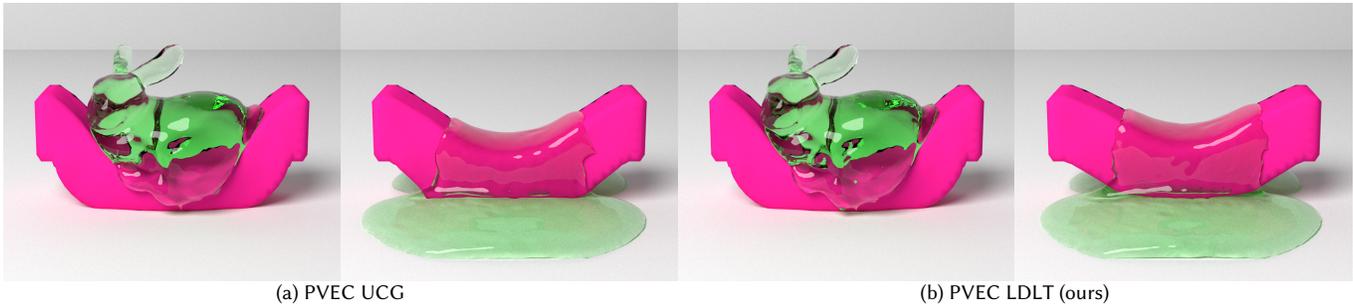


Fig. 11. A viscous bunny dropped onto an elastic beam with both its sides fixed, simulated with different schemes (see §4.4). PVEC UCG exhibits sticky contacts between the beam and the ground while PVEC LDLT (ours) generate plausible contact response, achieving 7.5× performance gain.

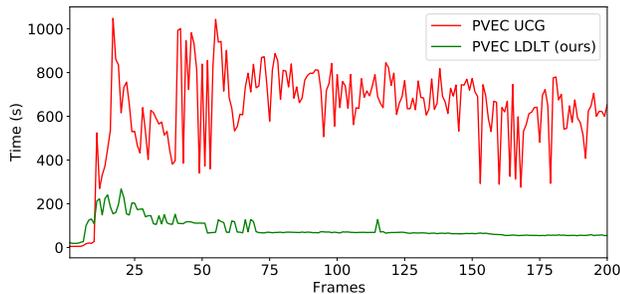


Fig. 12. Profiles of the total time for the entire pressure-viscosity-elasticity-contact handling per frame for Figure 11. PVEC LDLT (ours) is faster than PVEC UCG by a factor of 7.5.

users if time steps can be (almost) arbitrarily chosen, albeit with graceful performance degradation, compared to facing a sudden simulation breakdown with larger time steps (as is more often the case for weak or explicit coupling).

4.6.3 Inexact Newton Method. In general, performing more Newton iterations with relatively inaccurate descent directions can be more efficient compared to spending more linear solver time to find accurate descent directions [Nocedal and Wright 2006]. However, we observed that such inexact methods can frequently make Newton iterations diverge or suffer delayed convergence [Yan et al. 2018]. To quickly compute sufficiently accurate descent directions, it would be interesting to develop a variant of MPRGP based on, e.g., CR [Saad 2003], which guarantees a monotonic decrease of the residual [Fong and Saunders 2012; Macklin et al. 2019a] since the original MPRGP [Dostal and Schöberl 2005] (being based on CG [Shewchuk 1994]) does not provide this property. As an alternative for efficiently computing the descent direction (at the cost of accuracy), it would be interesting to develop a strategy to periodically skip re-evaluation of SDFs, volume fractions, and contact updates, with an analysis on acceptable inaccuracy [Yan et al. 2018]. Another interesting direction is to derive a sparse matrix decomposition with the optimal amount of fill-in, e.g., using nested dissection.

4.6.4 Elastic Solids. To correctly evaluate the elastic potentials and improve robustness (updating contact information), we allow position updates for elastic solids within the Newton iterations. As this position-level approach is typically more costly due to the re-evaluation of SDFs and volume fractions in each Newton iteration,

velocity-level approaches might be beneficial, although they are typically less accurate (unless the elastic potentials are quadratic [Ly et al. 2020]) and less robust due to their outdated contact information [Ostaduy et al. 2009].

4.6.5 Rigid Body Contact Handling. In contrast to elastic solids, we adopted velocity-level contact handling for rigid bodies, with their positional coordinates fixed during the Newton iterations, as we found it to be sufficiently robust. However, considering the improved robustness offered by position-level contact handling [English et al. 2013; Ferguson et al. 2021; Müller et al. 2020], it could be worth exploring these techniques despite their higher costs.

4.6.6 SPD Reformulation. Our SPD reformulation outperformed previous approaches for elastic-solid contact handling [Ostaduy et al. 2009] and two-way fluid-solid coupling [Zarifi and Batty 2017] in our examples. However, unlike these approaches, our reformulation requires LDLT decomposition for \mathbf{K} (which is $O(N^3)$ with N DOFs) and introduces additional non-zeros due to fill-in (especially for high-resolution and intricately shaped solids), negatively affecting efficiency and memory usage for subcomponents related to elastic solids (e.g., the scene with fewer fluid particles in Figure 9 is more costly than that in Figure 7 due to the elastic bowl which induces more non-zeros compared to the elastic beam). Thus, if such elastic solids are used, factorization-free scalable approaches might be preferable although factorization is still in practical use in recent simulators for complex deformable solids [Lan et al. 2021; Li et al. 2020b, 2021].

4.6.7 Preconditioning. In our experiments, IC preconditioning accelerates the convergence of MPRGP and was typically more efficient in the total solver time despite IC factorization being relatively expensive compared to SSOR (except for Figure 3 and 5) and modified incomplete Cholesky (MIC) [Bridson 2015]. However, it is still the most expensive part of our method, and it is worth investigating advanced preconditioners supporting box constraints, e.g., based on multiscale approaches [Aanjaneya 2018; Aanjaneya et al. 2019; Chen et al. 2021; Chentanez and Müller 2012; Demidov 2019; Lai et al. 2020; Shao et al. 2022].

In particular, when more and larger rigid bodies are involved, the number of necessary MPRGP iterations becomes larger due to the low-rank dense submatrices (which are not preconditioned). In

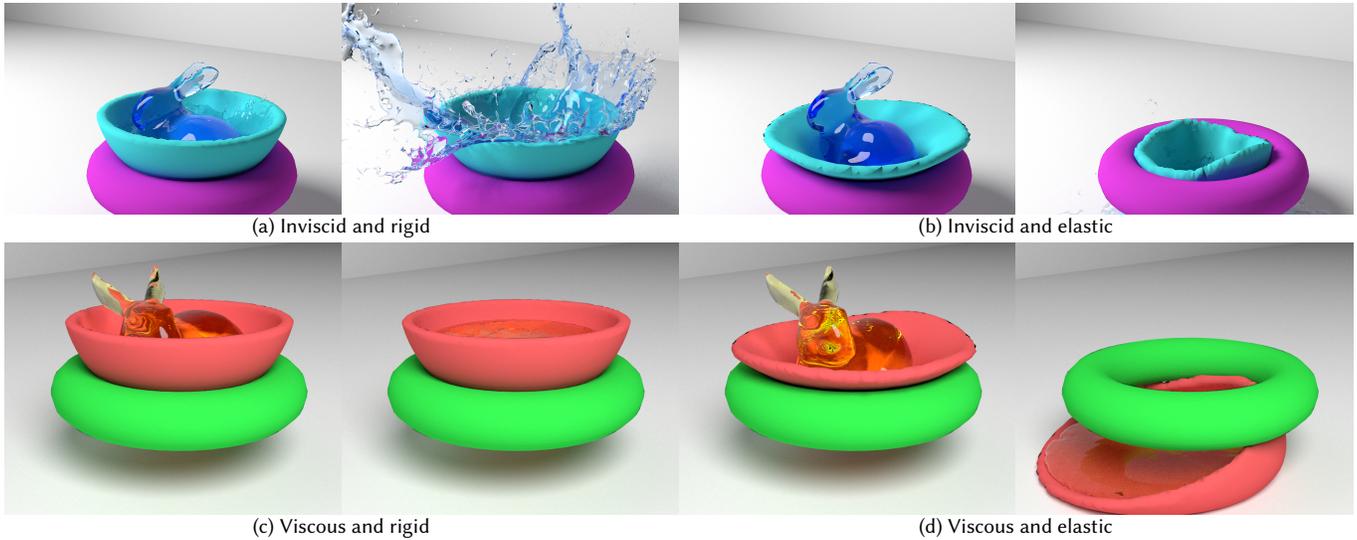


Fig. 13. A liquid bunny and a bowl dropped onto a rigid static torus, with different materials (inviscid or viscous for the liquids and rigid or elastic for the bowl).

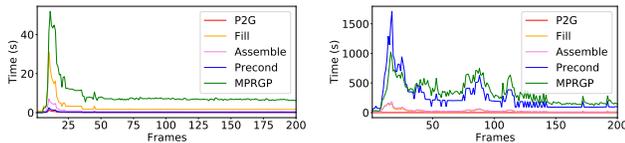


Fig. 14. The performance breakdown for Figure 13 (c) and (d). “P2G” represents the particle-to-grid transfer, “Fill” is forming the various matrices, “Assemble” is assembling the monolithic system, “Precond” is IC preconditioning, and “MPRGP” is the MPRGP solve.

our early experiments, although we tried applying SSOR preconditioning to the dense submatrices using SM [Takahashi and Batty 2020], the effect was not noticeable, probably because SSOR preconditioning was not powerful enough. As such, we are interested in exploring more effective preconditioning strategies which can be applied to the low-rank dense submatrices (e.g., [Aanjaneya 2018]).

4.6.8 Indefinite System. While we reformulated an indefinite system to be SPD due to the efficiency and wide availability of effective SPD solvers (e.g., CG [Shewchuk 1994] and MPRGP [Dostal and Schoberl 2005]), our reformulation can have some undesirable effects (e.g., denser non-zero patterns due to fill-in and matrix multiplications, and low-rank dense submatrices). As such, similar to some previous work [Robinson-Mosher et al. 2009, 2008], it will be interesting to investigate efficient indefinite solvers [Benzi et al. 2005], such as CR [Saad 2003] with extensions for constraint handling [Verschoor and Jalba 2019], and Uzawa iterations with preconditioning and inequality constraint handling [Gräser and Kornhuber 2007].

4.6.9 Level Set Representation. To efficiently treat liquids and solids in a unified framework, we mainly used a grid-based level set representation and evaluated SDFs, volume fractions, and contacts based on those level sets (along with the cut-cell approach for $F_{e,p}$ and $F_{e,s}$). However, the accuracy of these evaluations is fundamentally limited by the grid resolution, and these evaluations can be inconsistent (see e.g., [Takahashi and Lin 2019; Zarifi and Batty 2017]). Consistent

use of exact solid geometry for volume evaluations [Azevedo et al. 2016] and contact detection [Ferguson et al. 2021; Li et al. 2020b, 2021] may be preferable (e.g., for coupling with cloth and hair).

4.6.10 Contacts. Our contact constraint formulation is derived from the velocity-level LCP and augmented with Baumgarte stabilization [Baumgarte 1972] to compensate for accumulated position-level errors. However, Baumgarte stabilization can introduce extra energy, causing stability issues unless the stabilization parameters are carefully tuned. As an alternative, post-stabilization was developed for rigid bodies to address constraint drift [Cline and Pai 2003]; extending this technique to elastic solids and rigid-elastic solid coupling could address position-level constraint violations.

While we explicitly track persistent contacts for warm starting, there is no guarantee that the previous contact information is sufficiently helpful unless they are resting contacts. In general, there exist multiple pairs of contact forces and configurations that lead to the same global results, i.e., contact forces due to the Lagrange multipliers for the dual formulation can be ambiguous [Macklin et al. 2020; Zheng and James 2011]. Thus, it would be promising to explore primal formulations which are less likely to be affected by contact configurations [Li et al. 2020b, 2021; Macklin et al. 2020; Pan et al. 2019].

5 CONCLUSIONS

We proposed ElastoMonolith, a monolithic framework for challenging contact-aware coupling of inviscid/viscous fluids and rigid/elastic solids that can handle, in a unified way, fluid incompressibility, implicit viscosity integration, elastic deformations, frictional contacts, and mutual fluid-solid force exchanges due to pressure and viscous stress. To efficiently and robustly solve the constrained monolithic optimization problem, we presented our customized solver and evaluated its performance. Finally, we demonstrated our monolithic solver on difficult scenarios, showing that it eliminates artifacts and surpasses the capabilities of prior or alternative schemes.

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-2021-02524).

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.
- Sheldon Andrews and Kenny Erleben. 2021. Contact and Friction Simulation for Computer Graphics. In *ACM SIGGRAPH 2021 Courses* (Virtual Event, USA) (SIGGRAPH '21). Association for Computing Machinery, New York, NY, USA, Article 2, 124 pages.
- Vinicius C. Azevedo, Christopher Batty, and Manuel M. Oliveira. 2016. Preserving Geometry and Topology for Fluid Flows with Thin Obstacles and Narrow Gaps. *ACM Trans. Graph.* 35, 4, Article 97 (jul 2016), 12 pages.
- 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.
- David Baraff. 1996. Linear-time dynamics using lagrange multipliers. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 137–146.
- David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 43–54.
- 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.
- Christopher Batty and Ben Houston. 2011. A Simple Finite Volume Method for Adaptive Viscous Liquids. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 111–118.
- Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete Viscous Sheets. *ACM Transactions on Graphics* 31, 4, Article 113 (2012), 7 pages.
- 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 Tessenendorf, and Matthias Teschner. 2009. Direct Forcing for Lagrangian Rigid-Fluid Coupling. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (2009), 493–503.
- Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive Simulation of Rigid Body Dynamics in Computer Graphics. *Computer Graphics Forum* 33, 1 (2014), 246–270.
- Jan Bender, Matthias Müller, and Miles Macklin. 2017. A Survey on Position Based Dynamics. 2017. In *Proceedings of the European Association for Computer Graphics: Tutorials*. Eurographics Association, Article 6, 31 pages.
- Michele Benzi, Gene H Golub, and Jörg Liesen. 2005. Numerical solution of saddle point problems. *Acta numerica* 14 (2005), 1–137.
- Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Transactions on Graphics* 29, 4, Article 116 (2010), 10 pages.
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph.* 33, 4, Article 154 (2014), 11 pages.
- 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.
- Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph.* 21, 3 (2002), 594–603.
- George E. Brown, Matthew Overby, Zahra Foroortania, and Rahul Narain. 2018. Accurate Dissipative Forces in Optimization Integrators. *ACM Trans. Graph.* 37, 6, Article 282 (dec 2018), 14 pages.
- Mark Carlson, Peter Mucha, and Greg Turk. 2004. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid. *ACM Transactions on Graphics* 23 (2004).
- Mark Carlson, Peter J. Mucha, R. Brooks Van Horn, III, and Greg Turk. 2002. Melting and Flowing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 167–174.
- Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. 2010. A Simple Geometric Model for Elastic Deformations. *ACM Trans. Graph.* 29, 4, Article 38 (jul 2010), 6 pages.
- Jiong Chen, Florian Schäfer, Jin Huang, and Mathieu Desbrun. 2021. Multiscale Cholesky Preconditioning for Ill-Conditioned Problems. *ACM Trans. Graph.* 40, 4, Article 81 (jul 2021), 13 pages.
- 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.
- Nuttapong Chentanez and Matthias Müller. 2012. A Multigrid Fluid Pressure Solver Handling Separating Solid Boundary Conditions. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (2012), 1191–1201.
- 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* 32, 2, Article 17 (2013), 15 pages.
- 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.
- Eulalie Coevoet, Otman Benckekroun, and Paul G. Kry. 2020. Adaptive Merging for Rigid Body Simulation. *ACM Trans. Graph.* 39, 4, Article 35 (jul 2020), 13 pages.
- Gilles Daviet. 2020. Simple and Scalable Frictional Contacts for Thin Nodal Objects. *ACM Trans. Graph.* 39, 4, Article 61 (July 2020), 16 pages.
- Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. *ACM Trans. Graph.* 30, 6 (Dec. 2011), 1–12.
- D. Demidov. 2019. AMGCL: An Efficient, Flexible, and Extensible Algebraic Multigrid Implementation. *Lobachevskii Journal of Mathematics* 40, 5 (01 May 2019), 535–546.
- Dimitar Dinev, Tiantian Liu, and Ladislav Kavan. 2018a. Stabilizing Integrators for Real-Time Physics. *ACM Trans. Graph.* 37, 1, Article 9 (jan 2018), 19 pages.
- Dimitar Dinev, Tiantian Liu, Jing Li, Bernhard Thomaszewski, and Ladislav Kavan. 2018b. FEPR: Fast Energy Projection for Real-Time Simulation of Deformable Objects. *ACM Trans. Graph.* 37, 4, Article 79 (jul 2018), 12 pages.
- Ounan Ding and Craig Schroeder. 2020. Penalty Force for Coupling Materials with Coulomb Friction. *IEEE Transactions on Visualization and Computer Graphics* 26, 7 (2020), 2443–2455.
- Crawford Doran. 2013. *Isosurface stuffing improved: acute lattices and feature matching*. Ph.D. Dissertation. University of British Columbia.
- Zdenek Dostal and Joachim Schöberl. 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.
- Christian Duriez, Frederic Dubois, Abderrahmane Kheddar, and Claude Andriot. 2006. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE transactions on visualization and computer graphics* 12, 1 (2006), 36–47.
- Robert Elliot English, Michael Lentine, and Ronald Fedkiw. 2013. Interpenetration Free Simulation of Thin Shell Rigid Bodies. *IEEE Transactions on Visualization and Computer Graphics* 19, 6 (jun 2013), 991–1004.
- Kenny Erleben. 2007. Velocity-Based Shock Propagation for Multibody Dynamics Animation. *ACM Trans. Graph.* 26, 2 (June 2007), 12–es.
- Kenny Erleben. 2017. Rigid Body Contact Problems Using Proximal Operators. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Association for Computing Machinery, Article 13, 12 pages.
- Yu Fang, Minchen Li, Ming Gao, and Chenfanfu Jiang. 2019. Silly Rubber: An Implicit Material Point Method for Simulating Non-Equibrated Viscoelastic and Elastoplastic Solids. *ACM Trans. Graph.* 38, 4, Article 118 (jul 2019), 13 pages.
- 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 Trans. Graph.* 39, 4, Article 51 (2020).
- Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M. Kaufman, and Daniele Panozzo. 2021. Intersection-Free Rigid Body Dynamics. *ACM Trans. Graph.* 40, 4, Article 183 (jul 2021), 16 pages.
- S. Fisher and M.C. Lin. 2001. Fast penetration depth estimation for elastic bodies using deformed distance fields. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1. 330–336 vol.1.

- David Chin-Lung Fong and Michael A. Saunders. 2012. CG Versus MINRES: An Empirical Comparison. *Sultan Qaboos University Journal for Science* 17 (2012), 44–62.
- Nico Galoppo, Miguel A. Otaduy, Paul Mecklenburg, Markus Gross, and Ming C. Lin. 2006. Fast Simulation of Deformable Models in Contact Using Dynamic Deformation Textures. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 73–82.
- Nico Galoppo, Miguel A. Otaduy, Serhat Tekin, Markus Gross, and Ming C. Lin. 2007. Soft Articulated Characters with Fast Contact Handling. *Computer Graphics Forum* 26, 3 (2007), 243–253.
- Ming Gao, Andre Pradhana Tampubolon, Chenfanfu Jiang, and Eftychios Sifakis. 2017. An Adaptive Generalized Interpolation Material Point Method for Simulating Elastoplastic Materials. *ACM Trans. Graph.* 36, 6, Article 223 (2017), 223:1–223:12 pages.
- Theodore F. Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M. Teran. 2015. Optimization Integrator for Large Time Steps. *IEEE Transactions on Visualization and Computer Graphics* 21, 10 (2015), 1103–1115.
- 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.
- Tolga G. Goktekin, Adam W. Bargteil, and James F. O'Brien. 2004. A Method for Animating Viscoelastic Fluids. *ACM Trans. Graph.* 23, 3 (2004), 463–468.
- Ryan Goldade, Yipeng Wang, Mridul Aanjaneya, and Christopher Batty. 2019. An adaptive variational finite difference framework for efficient symmetric octree viscosity. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.
- Carsten Gräser and Ralf Kornhuber. 2007. On Preconditioned Uzawa-type Iterations for a Saddle Point Problem with Inequality Constraints. In *Domain Decomposition Methods in Science and Engineering XVI*, Olof B. Widlund and David E. Keyes (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 91–102.
- 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.
- 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).
- Philipp Herholz and Marc Alexa. 2018. Factor Once: Reusing Cholesky Factorizations on Sub-Meshes. *ACM Trans. Graph.* 37, 6, Article 230 (dec 2018), 9 pages.
- 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.
- Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics* 36 (07 2017), 1–14.
- 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.
- 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.
- 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), 164–111.
- Min Hyung Kee, Kiwon Um, Wooseok Jeong, and Junhyun Han. 2021. Constrained Projective Dynamics: Real-Time Simulation of Deformable Objects with Energy-Momentum Conservation. *ACM Trans. Graph.* 40, 4, Article 160 (jul 2021), 12 pages.
- Theodore Kim and David Eberle. 2020. Dynamic Deformables: Implementation and Production Practicalities. In *ACM SIGGRAPH 2020 Courses*. Article 23, 182 pages.
- 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, Nuttapon 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.
- J. Kružík, D. Horák, M. Cermák, L. Pospíšil, and M. Pecha. 2020. Active set expansion strategies in MPRGP algorithm. *Advances in Engineering Software* 149 (2020), 102895.
- Tassilo Kugelstadt, Jan Bender, José Antonio Fernández-Fernández, Stefan Rhys Jeske, Fabian Lössner, and Andreas Longva. 2021. Fast Corotated Elastic SPH Solids with Implicit Zero-Energy Mode Control. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 3, Article 33 (sep 2021), 21 pages.
- François Labelle and Jonathan Richard Shewchuk. 2007. Isosurface Stuffing: Fast Tetrahedral Meshes with Good Dihedral Angles. *ACM Trans. Graph.* 26, 3 (jul 2007), 57–es.
- 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.
- Lei Lan, Yin Yang, Danny Kaufman, Junfeng Yao, Minchen Li, and Chenfanfu Jiang. 2021. Medial IPC: Accelerated Incremental Potential Contact with Medial Elastics. *ACM Trans. Graph.* 40, 4, Article 158 (jul 2021), 16 pages.
- 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.
- Egor Larionov, Ye Fan, and Dinesh K. Pai. 2021. Frictional Contact on Smooth Elastic Solids. *ACM Trans. Graph.* 40, 2, Article 15 (April 2021), 17 pages.
- Quentin Le Lidec, Igor Kalevtykh, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. 2021. Differentiable simulation for physical system identification. *IEEE Robotics and Automation Letters* (2021). <https://hal.archives-ouvertes.fr/hal-03025616>
- Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George E. Brown, and Laurence Boissieux. 2018. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Trans. Graph.* 37, 4, Article 52 (July 2018), 15 pages.
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020b. Incremental Potential Contact: Intersection-and Inversion-Free, Large-Deformation Dynamics. *ACM Trans. Graph.* 39, 4, Article 49 (July 2020), 20 pages.
- Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph.* 40, 4, Article 170 (jul 2021), 24 pages.
- Siwang Li, Zherong Pan, Jin Huang, Hujun Bao, and Xiaogang Jin. 2015. Deformable Objects Collision Handling with Fast Convergence. *Comput. Graph. Forum* 34, 7 (oct 2015), 269–278.
- Wei Li, Yixin Chen, Mathieu Desbrun, Changxi Zheng, and Xiaopei Liu. 2020a. Fast and Scalable Turbulent Flow Simulation with Two-Way Coupling. *ACM Trans. Graph.* (2020).
- S. Liu, X. He, W. Wang, and E. Wu. 2021. Adapted SIMPLE Algorithm for Incompressible SPH Fluids with a Broad Range Viscosity. *IEEE Transactions on Visualization & Computer Graphics* 01 (jan 2021), 1–1.
- Wenlong Lu, Ning Jin, and Ronald Fedkiw. 2016. Two-Way Coupling of Fluids to Reduced Deformable Bodies. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 67–76.
- Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. 2020. Projective Dynamics with Dry Frictional Contact. *ACM Trans. Graph.* 39, 4, Article 57 (July 2020), 8 pages.
- Chaoyang Lyu, Wei Li, Mathieu Desbrun, and Xiaopei Liu. 2021. Fast and Versatile Fluid-Solid Coupling for Turbulent Flow Simulation. *ACM Trans. Graph.* 40, 6, Article 201 (dec 2021), 18 pages.
- M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and T. Y. Kim. 2020. *Primal/Dual Descent Methods for Dynamics*. Eurographics Association, Goslar, DEU.
- Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapon Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019a. Non-Smooth Newton Methods for Deformable Multi-Body Dynamics. *ACM Trans. Graph.* 38, 5, Article 140 (Oct. 2019), 20 pages.
- Miles Macklin and Matthias Müller. 2021. A Constraint-Based Formulation of Stable Neo-Hookean Materials. In *Motion, Interaction and Games*. Article 12, 7 pages.
- Miles Macklin, Matthias Müller, and Nuttapon Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*. 49–54.
- Miles Macklin, Matthias Müller, Nuttapon Chentanez, and Tae-Yong Kim. 2014. Unified Particle Physics for Real-time Applications. *ACM Transactions on Graphics* 33, 4, Article 153 (2014), 12 pages.
- Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapon Chentanez, Stefan Jeschke, and Matthias Müller. 2019b. Small Steps in Physics Simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Article 2, 7 pages.
- Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based Elastic Materials. *ACM Trans. Graph.* 30, 4, Article 72 (2011), 8 pages.
- Hammad Mazhar, Toby Heyn, Dan Negrut, and Alessandro Tasora. 2015. Using Nesterov's Method to Accelerate Multibody Dynamics with Friction and Contact. *ACM Trans. Graph.* 34, 3, Article 32 (May 2015), 14 pages.
- Marek Krzysztosf Misztal and Jakob Andreas Bærentzen. 2012. Topology-Adaptive Interface Tracking Using the Deformable Simplicial Complex. *ACM Trans. Graph.* 31, 3, Article 24 (jun 2012), 12 pages.
- Marek K. Misztal, Kenny Erleben, Adam Bargteil, Jens Fursund, Brian Bunch Christensen, J. Andreas Bærentzen, and Robert Bridson. 2014. Multiphase Flow of Immiscible Fluids on Unstructured Moving Meshes. *IEEE Transactions on Visualization and Computer Graphics* 20, 1 (jan 2014), 4–16.
- Matthias Müller, Nuttapon Chentanez, Miles Macklin, and Stefan Jeschke. 2017. Long Range Constraints for Rigid Body Simulations. In *Proceedings of the ACM SIGGRAPH*

- / *Eurographics Symposium on Computer Animation*. Article 14, 10 pages.
- Matthias Müller, Miles Macklin, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. 2020. Detailed Rigid Body Simulation with Extended Position Based Dynamics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Article 10, 12 pages.
- 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.
- Rahul Narain, Matthew Overby, and George E. Brown. 2016. ADMM \supseteq Projective Dynamics: Fast Simulation of General Constitutive Models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 21–28.
- 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.
- Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (second ed.). Springer, New York, NY, USA.
- 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.
- Matthew Overby, George E. Brown, Jie Li, and Rahul Narain. 2017. ADMM \supseteq Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE Transactions on Visualization and Computer Graphics* 23, 10 (oct 2017), 2222–2234.
- Zherong Pan, Bo Ren, and Dinesh Manocha. 2019. GPU-Based Contact-Aware Trajectory Optimization Using a Smooth Force Model. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Article 4, 12 pages.
- Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. 2015. An Implicit Viscosity Formulation for SPH Fluids. *ACM Trans. Graph.* 34, 4, Article 114 (2015), 10 pages.
- A. Peer and M. Teschner. 2017. Prescribed Velocity Gradients for Highly Viscous SPH Fluids with Vorticity Diffusion. *IEEE Transactions on Visualization and Computer Graphics* 23, 12 (2017), 2656–2662.
- Charles Peskin. 2002. Peskin, C.S.: The immersed boundary method. *Acta Numerica* 11, 479–517. *Acta Numerica* 11 (01 2002), 479 – 517.
- Laks Raghupathi and François Faure. 2006. QP-Collide: A new approach to collision treatment. In *Journées du groupe de travail Animation et Simulation, GTAS 06*.
- Daniel Ram, Theodore Gast, Chenfanfu Jiang, Craig Schroeder, Alexey Stomakhin, Joseph Teran, and Pirouz Kavehpour. 2015. A Material Point Method for Viscoelastic Fluids, Foams and Sponges. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. 157–163.
- N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw. 2004. Directable Photorealistic Liquids. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 193–202.
- Mathieu Renouf and Pierre Alart. 2005. Conjugate gradient type algorithms for frictional multi-contact problems: applications to granular materials. *Computer Methods in Applied Mechanics and Engineering* 194, 18 (2005), 2019 – 2041.
- 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.
- Liangwang Ruan, Jinyuan Liu, Bo Zhu, Shinjiro Sueda, Bin Wang, and Baoquan Chen. 2021. Solid-Fluid Interaction with Surface-Tension-Dominant Contact. *ACM Trans. Graph.* 40, 4, Article 120 (jul 2021), 12 pages.
- Yousef Saad. 2003. Iterative Methods for Sparse Linear Systems. *Notes* 3, 2nd Edition (2003), xviii+528. arXiv:0806.3802 <http://www.stanford.edu/class/cme324/saad.pdf>
- Guillaume Saupin, Christian Duriez, Stéphane Cotin, and Laurent Grisoni. 2008. Efficient Contact Modeling using Compliance Warping. In *computer graphics international*.
- Martin Servin, Claude Lacoursière, and Niklas Melin. 2006. Interactive Simulation of Elastic Deformable Materials. In *SIGRAD Conference*. 22–32.
- Han Shao, Libo Huang, and Dominik L. Michels. 2022. A Fast Unsmoothed Aggregation Algebraic Multigrid Framework for the Large-Scale Simulation of Incompressible Flow. *ACM Transaction on Graphics* 41, 4, Article 49 (07 2022).
- Jonathan Richard Shewchuk. 1994. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. (August 1994).
- 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*. Article 20, 50 pages.
- Morten Silcowitz, Sarah Niebe, and Kenny Erleben. 2009. Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation. *VRIPHYS 2009 - 6th Workshop on Virtual Reality Interactions and Physical Simulations*, 105–114.
- Morten Silcowitz, Sarah Niebe, and Kenny Erleben. 2010a. A nonsmooth nonlinear conjugate gradient method for interactive contact force problems. *The Visual Computer* 26 (2010), 893–901.
- Morten Silcowitz, Sarah Niebe, and Kenny Erleben. 2010b. Projected Gauss-Seidel Subspace Minimization Method for Interactive Rigid Body Dynamics - Improving Animation Quality using a Projected Gauss-Seidel Subspace Minimization Method. *GRAPP 2010 - Proceedings of the International Conference on Computer Graphics Theory and Applications* 229, 38–45.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (mar 2018), 15 pages.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Trans. Graph.* 38, 1, Article 3 (feb 2019), 15 pages.
- 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.
- Olga Sorkine and Marc Alexa. 2007. As-Rigid-as-Possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. 109–116.
- Jos Stam. 1999. Stable Fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. 121–128.
- David E. Stewart. 2000. Rigid-Body Dynamics with Friction and Impact. *SIAM Rev.* 42, 1 (2000), 3–39.
- Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A Material Point Method for Snow Simulation. *ACM Trans. Graph.* 32, 4, Article 102 (2013), 102:1–102:10 pages.
- 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* 33, 4, Article 138 (2014), 11 pages.
- Haozhe Su, Tao Xue, Chengguizi Han, Chenfanfu Jiang, and Mridul Aanjaneya. 2021. A Unified Second-Order Accurate in Time MPM Formulation for Simulating Viscoelastic Liquids with Phase Change. *ACM Trans. Graph.* 40, 4, Article 119 (jul 2021), 18 pages.
- Tetsuya Takahashi and Christopher Batty. 2020. Monolith: a monolithic pressure-viscosity-contact solver for strong two-way rigid-rigid fluid-fluid coupling. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–16.
- Tetsuya Takahashi and Christopher Batty. 2021. Frictional Monolith: A Monolithic Optimization-based Approach for Granular Flow with Contact-Aware Rigid-Body Coupling. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–16.
- Tetsuya Takahashi and Christopher Batty. 2022. Fast Marching-Cubes-Style Volume Evaluation for Level Set Surfaces. *Journal of Computer Graphics Techniques (JCGT)* (2022).
- Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C. Lin. 2015. Implicit Formulation for SPH-based Viscous Fluids. *Computer Graphics Forum* 34, 2 (2015), 493–502.
- 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, 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.
- Min Tang, Dinesh Manocha, Miguel A. Otaduy, and Ruofeng Tong. 2012. Continuous Penalty Forces. *ACM Trans. Graph.* 31, 4, Article 107 (July 2012), 9 pages.
- Yun Teng, David I. W. Levin, and Theodore Kim. 2016. Eulerian Solid-Fluid Coupling. *ACM Trans. Graph.* 35, 6, Article 200 (2016), 8 pages.
- Emanuel Todorov. 2010. Implicit nonlinear complementarity: A new approach to contact dynamics. *Proceedings - IEEE International Conference on Robotics and Automation*, 2322 – 2329.
- Emanuel Todorov. 2011. A convex, smooth and invertible contact model for trajectory optimization. 1071–1076.
- Richard Tonge, Feodor Benevolenski, and Andrey Voroshilov. 2012. Mass Splitting for Jitter-Free Parallel Rigid Body Simulation. *ACM Transactions on Graphics* 31 (2012).
- Maxime Tournier, Matthieu Nesme, Benjamin Gilles, and François Faure. 2015. Stable constrained dynamics. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.
- Nghia Truong, Cem Yuksel, Chakrit Watcharopas, Joshua Aaron Levine, and Robert M. Kirby. 2021. Particle Merging-and-Splitting. *IEEE Transactions on Visualization and Computer Graphics* (2021).
- Mickeal Verschoor and Andrei C. Jalba. 2019. Efficient and Accurate Collision Response for Elastically Deformable Models. *ACM Trans. Graph.* 38, 2, Article 17 (March 2019), 20 pages.
- Zheng Wang. 2012. *Isometrically deforming cloth with simultaneous collision handling*. Ph.D. Dissertation. University of British Columbia.
- Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum* 37, 2 (2018), 145–155.
- Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, and James F. O’Brien. 2010. Dynamic Local Remeshing for Elastoplastic Simulation. *ACM Transactions on Graphics* 29, 4, Article 49 (2010), 11 pages.

- Chris Wojtan and Greg Turk. 2008. Fast Viscoelastic Behavior with Thin Features. *ACM Transactions on Graphics* 27, 3, Article 47 (2008), 8 pages.
- Longhua Wu, Botao Wu, Yin Yang, and Huamin Wang. 2020. A Safe and Fast Repulsion Method for GPU-Based Cloth Self Collisions. *ACM Trans. Graph.* 40, 1, Article 5 (dec 2020), 18 pages.
- Hongyi Xu, Yili Zhao, and Jernej Barbic. 2014. Implicit Multibody Penalty-Based Distributed Contact. *Visualization and Computer Graphics, IEEE Transactions on* 20 (09 2014), 1266–1279.
- Guowei Yan, Wei Li, Ruigang Yang, and Huamin Wang. 2018. Inexact Descent Methods for Elastic Parameter Optimization. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. Article 253, 14 pages.
- 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.
- 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*. Article 7, 11 pages.
- Changxi Zheng and Doug L. James. 2011. Toward High-Quality Modal Contact Sound. *ACM Trans. Graph.* 30, 4, Article 38 (jul 2011), 12 pages.
- 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 and Robert Bridson. 2005. Animating Sand As a Fluid. *ACM Trans. Graph.* 24, 3 (2005), 965–972.
- Yongning Zhu, Efthychios 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), 18 pages.

A FIRST AND SECOND ORDER DERIVATIVES

The first order derivatives of the monolithic system (20) are

$$\begin{bmatrix} \frac{\partial E}{\partial \mathbf{u}} \\ \frac{\partial E}{\partial \mathbf{x}_e} \\ \frac{\partial E}{\partial \mathbf{v}_r} \\ \frac{\partial E}{\partial \mathbf{p}} \\ \frac{\partial E}{\partial \mathbf{c}} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_f(\mathbf{u} - \mathbf{u}^*) + \Delta t \mathbf{D} \mathbf{s} + \Delta t \mathbf{G} \mathbf{p} \\ \frac{\alpha \mathbf{M}_e}{\Delta t^2} (\mathbf{x}_e - \mathbf{x}_e^*) + \alpha \nabla \Psi(\mathbf{x}_e) - \mathbf{F}_{e,s} \mathbf{s} - \mathbf{F}_{e,p} \mathbf{p} - \alpha \mathbf{J}_e^T \mathbf{c} \\ \alpha \mathbf{M}_r (\mathbf{v}_r - \mathbf{v}_r^*) - \Delta t \mathbf{F}_{r,s} \mathbf{s} - \Delta t \mathbf{F}_{r,p} \mathbf{p} - \Delta t \alpha \mathbf{J}_r^T \mathbf{c} \\ \Delta t \mathbf{D}^T \mathbf{u} - \mathbf{F}_{e,s}^T (\mathbf{x}_e - \mathbf{x}_e^*) - \Delta t \mathbf{F}_{r,s}^T \mathbf{v}_r - \frac{\Delta t}{2} \mathbf{N}^{-1} \mathbf{s} \\ \Delta t \mathbf{G}^T \mathbf{u} - \mathbf{F}_{e,p}^T (\mathbf{x}_e - \mathbf{x}_e^*) - \Delta t \mathbf{F}_{r,p}^T \mathbf{v}_r \\ - \alpha \mathbf{J}_e (\mathbf{x}_e - \mathbf{x}_e^*) - \Delta t \alpha \mathbf{J}_r \mathbf{v}_r \end{bmatrix}. \quad (29)$$

The second order derivatives of the monolithic system (20) are

$$\begin{bmatrix} \frac{\partial^2 E}{\partial \mathbf{u}^2} & \frac{\partial^2 E}{\partial \mathbf{u} \partial \mathbf{x}_e} & \frac{\partial^2 E}{\partial \mathbf{u} \partial \mathbf{v}_r} & \frac{\partial^2 E}{\partial \mathbf{u} \partial \mathbf{s}} & \frac{\partial^2 E}{\partial \mathbf{u} \partial \mathbf{p}} & \frac{\partial^2 E}{\partial \mathbf{u} \partial \mathbf{c}} \\ \frac{\partial^2 E}{\partial \mathbf{x}_e \partial \mathbf{u}} & \frac{\partial^2 E}{\partial \mathbf{x}_e^2} & \frac{\partial^2 E}{\partial \mathbf{x}_e \partial \mathbf{v}_r} & \frac{\partial^2 E}{\partial \mathbf{x}_e \partial \mathbf{s}} & \frac{\partial^2 E}{\partial \mathbf{x}_e \partial \mathbf{p}} & \frac{\partial^2 E}{\partial \mathbf{x}_e \partial \mathbf{c}} \\ \frac{\partial^2 E}{\partial \mathbf{v}_r \partial \mathbf{u}} & \frac{\partial^2 E}{\partial \mathbf{v}_r \partial \mathbf{x}_e} & \frac{\partial^2 E}{\partial \mathbf{v}_r^2} & \frac{\partial^2 E}{\partial \mathbf{v}_r \partial \mathbf{s}} & \frac{\partial^2 E}{\partial \mathbf{v}_r \partial \mathbf{p}} & \frac{\partial^2 E}{\partial \mathbf{v}_r \partial \mathbf{c}} \\ \frac{\partial^2 E}{\partial \mathbf{s} \partial \mathbf{u}} & \frac{\partial^2 E}{\partial \mathbf{s} \partial \mathbf{x}_e} & \frac{\partial^2 E}{\partial \mathbf{s} \partial \mathbf{v}_r} & \frac{\partial^2 E}{\partial \mathbf{s}^2} & \frac{\partial^2 E}{\partial \mathbf{s} \partial \mathbf{p}} & \frac{\partial^2 E}{\partial \mathbf{s} \partial \mathbf{c}} \\ \frac{\partial^2 E}{\partial \mathbf{p} \partial \mathbf{u}} & \frac{\partial^2 E}{\partial \mathbf{p} \partial \mathbf{x}_e} & \frac{\partial^2 E}{\partial \mathbf{p} \partial \mathbf{v}_r} & \frac{\partial^2 E}{\partial \mathbf{p} \partial \mathbf{s}} & \frac{\partial^2 E}{\partial \mathbf{p}^2} & \frac{\partial^2 E}{\partial \mathbf{p} \partial \mathbf{c}} \\ \frac{\partial^2 E}{\partial \mathbf{c} \partial \mathbf{u}} & \frac{\partial^2 E}{\partial \mathbf{c} \partial \mathbf{x}_e} & \frac{\partial^2 E}{\partial \mathbf{c} \partial \mathbf{v}_r} & \frac{\partial^2 E}{\partial \mathbf{c} \partial \mathbf{s}} & \frac{\partial^2 E}{\partial \mathbf{c} \partial \mathbf{p}} & \frac{\partial^2 E}{\partial \mathbf{c}^2} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_f & \mathbf{O} & \mathbf{O} & \Delta t \mathbf{D} & \Delta t \mathbf{G} & \mathbf{O} \\ \mathbf{O} & \hat{\mathbf{H}}_{\mathbf{x}_e \mathbf{x}_e} & \mathbf{O} & -\mathbf{F}_{e,s} & -\mathbf{F}_{e,p} & -\alpha \mathbf{J}_e^T \\ \mathbf{O} & \mathbf{O} & \alpha \mathbf{M}_r & -\Delta t \mathbf{F}_{r,s} & -\Delta t \mathbf{F}_{r,p} & -\Delta t \alpha \mathbf{J}_r^T \\ \Delta t \mathbf{D}^T & -\mathbf{F}_{e,s}^T & -\Delta t \mathbf{F}_{r,p}^T & -\frac{\Delta t}{2} \mathbf{N}^{-1} & \mathbf{O} & \mathbf{O} \\ \Delta t \mathbf{G}^T & -\mathbf{F}_{e,p}^T & -\Delta t \mathbf{F}_{r,p}^T & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & -\alpha \mathbf{J}_e & -\Delta t \alpha \mathbf{J}_r & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{bmatrix}, \quad (30)$$

$$\hat{\mathbf{H}}_{\mathbf{x}_e \mathbf{x}_e} = \frac{\alpha \mathbf{M}_e}{\Delta t^2} + \alpha \nabla^2 \Psi(\mathbf{x}_e).$$

B MONOLITHIC SPD REFORMULATION

The system matrix due to the second order derivatives (30) is symmetric but indefinite due to the presence of Lagrange multipliers (even after the Hessian projection). To derive an SPD formulation, we first augment the system with a newly introduced unknown

variable $\Delta \mathbf{g}$, using $\mathbf{K} = \mathbf{Q} \mathbf{C} \mathbf{Q}^T$ via LDLT decomposition:

$$\begin{bmatrix} \mathbf{M}_f & \mathbf{O} & \mathbf{O} & \Delta t \mathbf{D} & \Delta t \mathbf{G} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \frac{\alpha \mathbf{M}_e}{\Delta t^2} & \mathbf{O} & -\mathbf{F}_{e,s} & -\mathbf{F}_{e,p} & -\alpha \mathbf{J}_e^T & \alpha \mathbf{Q} \\ \mathbf{O} & \mathbf{O} & \alpha \mathbf{M}_r & -\Delta t \mathbf{F}_{r,s} & -\Delta t \mathbf{F}_{r,p} & -\Delta t \alpha \mathbf{J}_r^T & \mathbf{O} \\ \Delta t \mathbf{D}^T & -\mathbf{F}_{e,s}^T & -\Delta t \mathbf{F}_{r,p}^T & -\frac{\Delta t}{2} \mathbf{N}^{-1} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \Delta t \mathbf{G}^T & -\mathbf{F}_{e,p}^T & -\Delta t \mathbf{F}_{r,p}^T & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & -\alpha \mathbf{J}_e & -\Delta t \alpha \mathbf{J}_r & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \alpha \mathbf{Q}^T & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & -\alpha \mathbf{C}^{-1} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}^T \\ \Delta \mathbf{x}_e^T \\ \Delta \mathbf{v}_r^T \\ \Delta \mathbf{s}^T \\ \Delta \mathbf{p}^T \\ \Delta \mathbf{c}^T \\ \Delta \mathbf{g}^T \end{bmatrix} = \begin{bmatrix} -\left(\frac{\partial E}{\partial \mathbf{u}}\right)^T & -\left(\frac{\partial E}{\partial \mathbf{x}_e}\right)^T & -\left(\frac{\partial E}{\partial \mathbf{v}_r}\right)^T & -\left(\frac{\partial E}{\partial \mathbf{s}}\right)^T & -\left(\frac{\partial E}{\partial \mathbf{p}}\right)^T & -\left(\frac{\partial E}{\partial \mathbf{c}}\right)^T & \mathbf{O} \end{bmatrix}^T. \quad (31)$$

By eliminating $\Delta \mathbf{u}$, $\Delta \mathbf{x}_e$, and $\Delta \mathbf{v}_r$ from this system via substitution, we obtain an SPD and much smaller system:

$$\begin{bmatrix} \mathbf{A}_{ss} & \mathbf{A}_{sp} & \mathbf{A}_{sc} & \mathbf{A}_{sg} \\ \mathbf{A}_{ps} & \mathbf{A}_{pp} & \mathbf{A}_{pc} & \mathbf{A}_{pg} \\ \mathbf{A}_{cs} & \mathbf{A}_{cp} & \mathbf{A}_{cc} & \mathbf{A}_{cg} \\ \mathbf{A}_{gs} & \mathbf{A}_{gp} & \mathbf{A}_{gc} & \mathbf{A}_{gg} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{s} \\ \Delta \mathbf{p} \\ \Delta \mathbf{c} \\ \Delta \mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_s \\ \mathbf{b}_p \\ \mathbf{b}_c \\ \mathbf{b}_g \end{bmatrix}, \quad (32)$$

$$\mathbf{A}_{ps} = \mathbf{A}_{sp}^T, \mathbf{A}_{cs} = \mathbf{A}_{sc}^T, \mathbf{A}_{cp} = \mathbf{A}_{pc}^T, \mathbf{A}_{gs} = \mathbf{A}_{sg}^T, \mathbf{A}_{gp} = \mathbf{A}_{pg}^T, \mathbf{A}_{gc} = \mathbf{A}_{cg}^T,$$

$$\mathbf{A}_{ss} = \frac{1}{2\Delta t} \mathbf{N}^{-1} + \mathbf{D}^T \mathbf{M}_f^{-1} \mathbf{D} + \frac{1}{\alpha} \mathbf{F}_{e,s}^T \mathbf{M}_e^{-1} \mathbf{F}_{e,s} + \frac{1}{\alpha} \mathbf{F}_{r,s}^T \mathbf{M}_r^{-1} \mathbf{F}_{r,s}, \quad (33)$$

$$\mathbf{A}_{sp} = \mathbf{D}^T \mathbf{M}_f^{-1} \mathbf{G} + \frac{1}{\alpha} \mathbf{F}_{e,s}^T \mathbf{M}_e^{-1} \mathbf{F}_{e,p} + \frac{1}{\alpha} \mathbf{F}_{r,s}^T \mathbf{M}_r^{-1} \mathbf{F}_{r,p}, \quad (34)$$

$$\mathbf{A}_{sc} = \mathbf{F}_{e,s}^T \mathbf{M}_e^{-1} \mathbf{J}_e^T + \mathbf{F}_{r,s}^T \mathbf{M}_r^{-1} \mathbf{J}_r^T, \quad (35)$$

$$\mathbf{A}_{sg} = -\mathbf{F}_{e,s}^T \mathbf{M}_e^{-1} \mathbf{Q}, \quad (36)$$

$$\mathbf{A}_{pp} = \mathbf{G}^T \mathbf{M}_f^{-1} \mathbf{G} + \frac{1}{\alpha} \mathbf{F}_{e,p}^T \mathbf{M}_e^{-1} \mathbf{F}_{e,p} + \frac{1}{\alpha} \mathbf{F}_{r,p}^T \mathbf{M}_r^{-1} \mathbf{F}_{r,p}, \quad (37)$$

$$\mathbf{A}_{pc} = \mathbf{F}_{e,p}^T \mathbf{M}_e^{-1} \mathbf{J}_e^T + \mathbf{F}_{r,p}^T \mathbf{M}_r^{-1} \mathbf{J}_r^T, \quad (38)$$

$$\mathbf{A}_{pg} = -\mathbf{F}_{e,p}^T \mathbf{M}_e^{-1} \mathbf{Q}, \quad (39)$$

$$\mathbf{A}_{cc} = \alpha \mathbf{J}_e \mathbf{M}_e^{-1} \mathbf{J}_e^T + \alpha \mathbf{J}_r \mathbf{M}_r^{-1} \mathbf{J}_r^T, \quad (40)$$

$$\mathbf{A}_{cg} = -\alpha \mathbf{J}_e \mathbf{M}_e^{-1} \mathbf{Q}, \quad (41)$$

$$\mathbf{A}_{gg} = \frac{\alpha}{\Delta t^2} \mathbf{C}^{-1} + \alpha \mathbf{Q}^T \mathbf{M}_e^{-1} \mathbf{Q}, \quad (42)$$

$$\begin{aligned} \mathbf{b}_s &= \frac{1}{\Delta t^2} \frac{\partial E}{\partial \mathbf{s}} - \frac{1}{\Delta t} \mathbf{D}^T \mathbf{M}_f^{-1} \frac{\partial E}{\partial \mathbf{u}} + \frac{1}{\alpha} \mathbf{F}_{e,s}^T \mathbf{M}_e^{-1} \frac{\partial E}{\partial \mathbf{x}_e} + \frac{1}{\Delta t \alpha} \mathbf{F}_{r,s}^T \mathbf{M}_r^{-1} \frac{\partial E}{\partial \mathbf{v}_r} \\ &= \frac{1}{\Delta t} \mathbf{D}^T \mathbf{u}^* - \frac{1}{\Delta t^2} \mathbf{F}_{e,s}^T (\mathbf{x}_e^* - \mathbf{x}_e^t) - \frac{1}{\Delta t} \mathbf{F}_{r,s}^T \mathbf{v}_r^* + \mathbf{F}_{e,s}^T \mathbf{M}_e^{-1} \nabla \Psi(\mathbf{x}_e) \\ &\quad - \mathbf{A}_{ss} \mathbf{s} - \mathbf{A}_{sp} \mathbf{p} - \mathbf{A}_{sc} \mathbf{c}, \end{aligned} \quad (43)$$

$$\begin{aligned} \mathbf{b}_p &= \frac{1}{\Delta t^2} \frac{\partial E}{\partial \mathbf{p}} - \frac{1}{\Delta t} \mathbf{G}^T \mathbf{M}_f^{-1} \frac{\partial E}{\partial \mathbf{u}} + \frac{1}{\alpha} \mathbf{F}_{e,p}^T \mathbf{M}_e^{-1} \frac{\partial E}{\partial \mathbf{x}_e} + \frac{1}{\Delta t \alpha} \mathbf{F}_{r,p}^T \mathbf{M}_r^{-1} \frac{\partial E}{\partial \mathbf{v}_r} \\ &= \frac{1}{\Delta t} \mathbf{G}^T \mathbf{u}^* - \frac{1}{\Delta t^2} \mathbf{F}_{e,p}^T (\mathbf{x}_e^* - \mathbf{x}_e^t) - \frac{1}{\Delta t} \mathbf{F}_{r,p}^T \mathbf{v}_r^* + \mathbf{F}_{e,p}^T \mathbf{M}_e^{-1} \nabla \Psi(\mathbf{x}_e) \\ &\quad - \mathbf{A}_{ps} \mathbf{s} - \mathbf{A}_{pp} \mathbf{p} - \mathbf{A}_{pc} \mathbf{c}, \end{aligned} \quad (44)$$

$$\mathbf{b}_c = \frac{1}{\Delta t^2} \frac{\partial E}{\partial \mathbf{c}} + \mathbf{J}_e \mathbf{M}_e^{-1} \frac{\partial E}{\partial \mathbf{x}_e} + \frac{1}{\Delta t} \mathbf{J}_r \mathbf{M}_r^{-1} \frac{\partial E}{\partial \mathbf{v}_r} \quad (45)$$

$$= -\frac{\alpha}{\Delta t^2} \mathbf{J}_e (\mathbf{x}_e^* - \mathbf{x}_e^t) - \frac{\alpha}{\Delta t} \mathbf{J}_r \mathbf{v}_r^* + \alpha \mathbf{J}_e \mathbf{M}_e^{-1} \nabla \Psi(\mathbf{x}_e) \quad (46)$$

$$- \mathbf{A}_{cs} \mathbf{s} - \mathbf{A}_{cp} \mathbf{p} - \mathbf{A}_{cc} \mathbf{c}, \quad (47)$$

$$\mathbf{b}_g = -\mathbf{Q}^T \mathbf{M}_e^{-1} \frac{\partial E}{\partial \mathbf{x}_e} \quad (48)$$

$$= \frac{\alpha}{\Delta t^2} \mathbf{Q}^T \mathbf{x}_e^* - \frac{\alpha}{\Delta t^2} \mathbf{Q}^T \mathbf{x}_e - \alpha \mathbf{Q}^T \mathbf{M}_e^{-1} \nabla \Psi(\mathbf{x}_e) \quad (49)$$

$$- \mathbf{A}_{gs} \mathbf{s} - \mathbf{A}_{gp} \mathbf{p} - \mathbf{A}_{gc} \mathbf{c}. \quad (50)$$

After we solve this SPD system, we can obtain $\Delta \mathbf{u}$, $\Delta \mathbf{x}_e$, $\Delta \mathbf{v}_r$ by

$$\Delta \mathbf{u} = \mathbf{M}_f^{-1} \left(-\frac{\partial E}{\partial \mathbf{u}} - \Delta t \mathbf{D} \Delta \mathbf{s} - \Delta t \mathbf{G} \Delta \mathbf{p} \right) \quad (51)$$

$$= -(\mathbf{u} - \mathbf{u}^*) - \Delta t \mathbf{M}_f^{-1} (\mathbf{D}(\mathbf{s} + \Delta \mathbf{s}) + \mathbf{G}(\mathbf{p} + \Delta \mathbf{p})), \quad (52)$$

$$\Delta \mathbf{x}_e = \Delta t^2 \mathbf{M}_e^{-1} \left(-\frac{1}{\alpha} \frac{\partial E}{\partial \mathbf{x}_e} + \frac{1}{\alpha} \mathbf{F}_{e,s} \Delta \mathbf{s} + \frac{1}{\alpha} \mathbf{F}_{e,p} \Delta \mathbf{p} + \mathbf{J}_e^T \Delta \mathbf{c} - \mathbf{Q} \Delta \mathbf{g} \right)$$

$$= -(\mathbf{x}_e - \mathbf{x}_e^*) + \Delta t^2 \mathbf{M}_e^{-1} \left(-\nabla \Psi(\mathbf{x}_e) + \frac{1}{\alpha} \mathbf{F}_{e,s} (\mathbf{s} + \Delta \mathbf{s}) + \frac{1}{\alpha} \mathbf{F}_{e,p} (\mathbf{p} + \Delta \mathbf{p}) + \mathbf{J}_e^T (\mathbf{c} + \Delta \mathbf{c}) - \mathbf{Q} \Delta \mathbf{g} \right),$$

$$\Delta \mathbf{v}_r = \mathbf{M}_r^{-1} \left(-\frac{1}{\alpha} \frac{\partial E}{\partial \mathbf{v}_r} + \frac{\Delta t}{\alpha} \mathbf{F}_{r,s} \Delta \mathbf{s} + \frac{\Delta t}{\alpha} \mathbf{F}_{r,p} \Delta \mathbf{p} + \Delta t \mathbf{J}_r^T \Delta \mathbf{c} \right) \quad (53)$$

$$= -(\mathbf{v}_r - \mathbf{v}_r^*) + \Delta t \mathbf{M}_r^{-1} \left(\frac{1}{\alpha} \mathbf{F}_{r,s} (\mathbf{s} + \Delta \mathbf{s}) + \frac{1}{\alpha} \mathbf{F}_{r,p} (\mathbf{p} + \Delta \mathbf{p}) + \mathbf{J}_r^T (\mathbf{c} + \Delta \mathbf{c}) \right). \quad (54)$$

C DISCUSSIONS ON KINETIC ENERGY MINIMIZATION FORMULATION AND OURS

For the strong two-way coupling of inviscid fluids and rigid bodies, previous works (e.g., [Batty et al. 2007; Takahashi and Batty 2020, 2021]) derived their formulations based on the kinetic energy minimization with pressure forces (or impulses), i.e., dual variables or Lagrange multipliers:

$$\mathbf{p} = \arg \min_{\mathbf{p}} \frac{1}{2} \left(\|\mathbf{u}\|_{\mathbf{M}_f}^2 + \|\mathbf{v}_r\|_{\mathbf{M}_r}^2 \right), \quad (55)$$

$$\mathbf{u} = \mathbf{u}^* - \Delta t \mathbf{M}_f^{-1} \mathbf{G} \mathbf{p}, \mathbf{v}_r = \mathbf{v}_r^* + \Delta t \mathbf{M}_r^{-1} \mathbf{F}_{r,p} \mathbf{p}. \quad (56)$$

This minimization problem is quadratic and can be handled by solving the linear system for unknown pressure forces \mathbf{p} :

$$\left(\mathbf{G}^T \mathbf{M}_f^{-1} \mathbf{G} + \mathbf{F}_{r,p}^T \mathbf{M}_r^{-1} \mathbf{F}_{r,p} \right) \mathbf{p} = \frac{1}{\Delta t} \left(\mathbf{G}^T \mathbf{u}^* - \mathbf{F}_{r,p}^T \mathbf{v}_r^* \right). \quad (57)$$

While we focus on inviscid liquids here for simplicity, viscous fluids can also be addressed in the same way with viscous stress \mathbf{s} [Takahashi and Batty 2020].

However, unfortunately, this approach based on the kinetic energy minimization with forces (dual variables) cannot incorporate elastic solids [Bridson 2015] because there is no explicit relation from \mathbf{p} to \mathbf{x}_e (i.e., it is defined only implicitly due to the elastic potential), unlike liquids and rigid bodies (56), preventing us from expressing the kinetic energy minimization for elastic solids through

the dual variables. Thus, we take a different approach that minimizes the inertia and elastic potential on primal variables under the incompressibility and frictional contact constraints, which in the end leads to the minimization on primal variables and maximization on dual variables (Lagrange multipliers).

As a sub-component of the monolithic system (20), our formulation for the strong coupling of inviscid liquids and rigid bodies can be given in the mixed minimization/maximization form as

$$\mathbf{u}, \mathbf{v}_r, \mathbf{p} = \arg \min_{\mathbf{u}, \mathbf{v}_r} \arg \max_{\mathbf{p}} E(\mathbf{u}, \mathbf{v}_r, \mathbf{p}), \quad (58)$$

$$E(\mathbf{u}, \mathbf{v}_r, \mathbf{p}) = \frac{1}{2} \left(\|\mathbf{u} - \mathbf{u}^*\|_{\mathbf{M}_f}^2 + \|\mathbf{v}_r - \mathbf{v}_r^*\|_{\mathbf{M}_r}^2 \right) + \Delta t \mathbf{p} \left(\mathbf{G}^T \mathbf{u} - \mathbf{F}_{r,p}^T \mathbf{v}_r \right). \quad (59)$$

Since this optimization problem is a quadratic minimization with respect to the primal variables (\mathbf{u} and \mathbf{v}_r) and a linear maximization with respect to the dual variables (\mathbf{p}), we can find the optimality by extremizing the objective and solving the resulting symmetric indefinite system due to the KKT condition once (without using Newton's method):

$$\begin{bmatrix} \mathbf{M}_f & \mathbf{O} & \Delta t \mathbf{G} \\ \mathbf{O} & \mathbf{M}_r & -\Delta t \mathbf{F}_{r,p} \\ \Delta t \mathbf{G}^T & -\Delta t \mathbf{F}_{r,p}^T & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v}_r \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_f \mathbf{u}^* \\ \mathbf{M}_r \mathbf{v}_r^* \\ \mathbf{0} \end{bmatrix}. \quad (60)$$

The first and second rows can be eliminated by Schur complement substitutions into the third row, and we arrive at the SPD linear system (57) derived from the kinetic energy minimization approach, proving the equivalence between the kinetic energy minimization formulation (55) and ours (58) when primal variables can be explicitly expressed with dual variables. As such, since the kinetic energy minimization formulation is a subset of our formulation, our approach is more general, enabling us to handle the case that the relation between primal and dual variables is defined only implicitly, e.g., for elastic solids.

D PEC UCG AND PVEC UCG FORMULATIONS

The indefinite system used with PEC UCG in Sec. 4.3 is given as

$$\begin{bmatrix} \mathbf{M}_f & \mathbf{O} & \Delta t \mathbf{G} & \mathbf{O} \\ \mathbf{O} & \mathbf{H}_{\mathbf{x}_e \mathbf{x}_e} & -\mathbf{F}_{e,p} & -\alpha \mathbf{J}_e^T \\ \Delta t \mathbf{G}^T & -\mathbf{F}_{e,p}^T & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & -\alpha \mathbf{J}_e & \mathbf{O} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{x}_e \\ \Delta \mathbf{p} \\ \Delta \mathbf{c} \end{bmatrix} = \begin{bmatrix} -\frac{\partial E}{\partial \mathbf{u}} \\ -\frac{\partial E}{\partial \mathbf{x}_e} \\ -\frac{\partial E}{\partial \mathbf{p}} \\ -\frac{\partial E}{\partial \mathbf{c}} \end{bmatrix}. \quad (61)$$

The indefinite system used with PVEC UCG in Sec. 4.4 is given as

$$\begin{bmatrix} \mathbf{M}_f + 2\Delta t \mathbf{D} \mathbf{N} \mathbf{D}^T & -2\Delta t \mathbf{D} \mathbf{N} \mathbf{F}_{e,s}^T & \Delta t \mathbf{G} & \mathbf{O} \\ -2\mathbf{F}_{e,s} \mathbf{N} \mathbf{D}^T & \mathbf{H}_{\mathbf{x}_e \mathbf{x}_e} + \frac{2}{\Delta t} \mathbf{F}_{e,s} \mathbf{N} \mathbf{F}_{e,s}^T & -\mathbf{F}_{e,p} & -\alpha \mathbf{J}_e^T \\ \Delta t \mathbf{G}^T & -\mathbf{F}_{e,p}^T & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & -\alpha \mathbf{J}_e & \mathbf{O} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{x}_e \\ \Delta \mathbf{p} \\ \Delta \mathbf{c} \end{bmatrix} = \begin{bmatrix} -\frac{\partial E}{\partial \mathbf{u}} - 2\Delta t \mathbf{D} \mathbf{N} \frac{\partial E}{\partial \mathbf{s}} \\ -\frac{\partial E}{\partial \mathbf{x}_e} + \frac{2}{\Delta t} \mathbf{F}_{e,s} \mathbf{N} \frac{\partial E}{\partial \mathbf{s}} \\ -\frac{\partial E}{\partial \mathbf{p}} \\ -\frac{\partial E}{\partial \mathbf{c}} \end{bmatrix}. \quad (62)$$