

A Generalized Constitutive Model for Versatile MPM Simulation and Inverse Learning with Differentiable Physics

HAOZHE SU*, Rutgers University, USA

XUAN LI*, University of California, Los Angeles, USA

TAO XUE, Nanjing University of Science and Technology, China

CHENFANFU JIANG, University of California, Los Angeles, USA

MRIDUL AANJANEYA, Rutgers University, USA

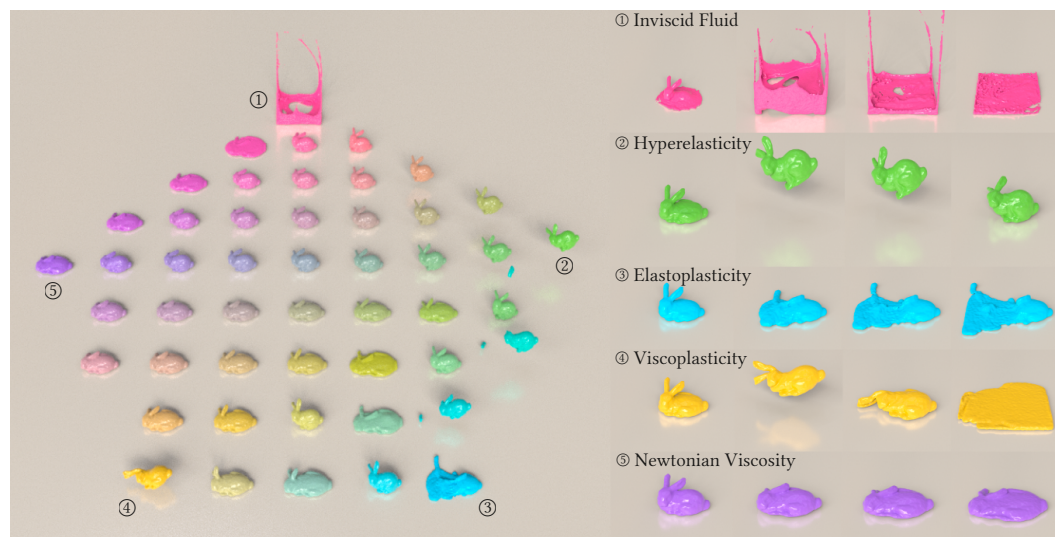


Fig. 1. (Left) Our generalized constitutive model allows the simulation of inviscid fluids, Newtonian viscosity, hyperelasticity, viscoplasticity, elastoplasticity (shown as the 5 corners of the pentagon) and other physical effects that arise due to a mixture of these behaviors (shown as sample points inside the pentagon). (Right) Snapshots of simulations at the pentagon corners.

We present a generalized constitutive model for versatile physics simulation of inviscid fluids, Newtonian viscosity, hyperelasticity, viscoplasticity, elastoplasticity, and other physical effects that arise due to a mixture of these behaviors. The key ideas behind our formulation are the design of a generalized Kirchhoff stress tensor that can describe hyperelasticity, Newtonian viscosity and inviscid fluids, and the use of pre-projection and post-correction rules for simulating material behaviors that involve plasticity, including elastoplasticity and viscoplasticity. We show how our generalized Kirchhoff stress tensor can be coupled together into a generalized

*joint first authors

Authors' addresses: Haozhe Su, Rutgers University, USA; Xuan Li, University of California, Los Angeles, USA; Tao Xue, Nanjing University of Science and Technology, China; Chenfanfu Jiang, University of California, Los Angeles, USA; Mridul Aanjaneya, Rutgers University, USA.

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

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

2577-6193/2023/8-ART1 \$15.00

<https://doi.org/10.1145/3606925>



Fig. 2. The characteristic pattern of liquid flowing down a fountain. (Top) Our generalized constitutive model using $\kappa > 0, \mu = 0, \eta = 0$ and no viscoplasticity or elastoplasticity. (Bottom) Reference results using the equation-of-state fluid model.

constitutive model that allows the simulation of diverse material behaviors by only changing parameter values. We present several side-by-side comparisons with physics simulations for specific constitutive models to show that our generalized model produces visually similar results. More notably, our formulation allows for inverse learning of unknown material properties directly from data using differentiable physics simulations. We present several 3D simulations to highlight the robustness of our method, even with multiple different materials. To the best of our knowledge, our approach is the *first* to recover the knowledge of unknown material properties without making explicit assumptions about the data.

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

Additional Key Words and Phrases: generalized constitutive model, viscosity, elasticity, plasticity, material point method, differentiable physics

ACM Reference Format:

Haozhe Su, Xuan Li, Tao Xue, Chenfanfu Jiang, and Mridul Aanjaneya. 2023. A Generalized Constitutive Model for Versatile MPM Simulation and Inverse Learning with Differentiable Physics. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 2, Article 1 (August 2023), 20 pages. <https://doi.org/10.1145/3606925>

1 INTRODUCTION

The quest to reproduce the behavior of physical materials digitally has been a longstanding pursuit of computer graphics [Bridson 2015; Hu et al. 2019; Jiang et al. 2016; Sifakis and Barbic 2012] and the VFX industry [Frost et al. 2017; Geiger et al. 2006; Hutchins et al. 2015; Rasmussen et al. 2004]. Apart from the natural appeal of creating immersive virtual worlds, researchers have also started applying physics simulation in the emerging disciplines of virtual surgery [Lee et al. 2019, 2018; Mitchell et al. 2015], digital fabrication [Ma et al. 2017], and soft robotics [Huang et al. 2021].

The standard approach to physics simulation uses a constitutive model that characterizes physical attributes of the materials being simulated, and discretization schemes for constructing discrete systems that are subsequently solved using numerical methods. While several researchers have explored generalized discretization schemes that can accommodate a large class of materials [Bouaziz et al. 2014; Hu et al. 2019; Jiang et al. 2016; Liu et al. 2017; Losasso et al. 2006; Macklin et al. 2014; Martin et al. 2010; Narain et al. 2016; Yan et al. 2016], there is very little prior work on generalized constitutive models [Fang et al. 2019; Su et al. 2021; Xue et al. 2020]. To some extent, this makes sense because specialized numerical solvers that are tailored for specific constitutive models tend to be much faster than generalized simulation methods [Aanjaneya 2018; Aanjaneya et al. 2019; Chu et al. 2017; Liu et al. 2016, 2017; McAdams et al. 2010; Zhang and Bridson 2014; Zhu et al. 2010].

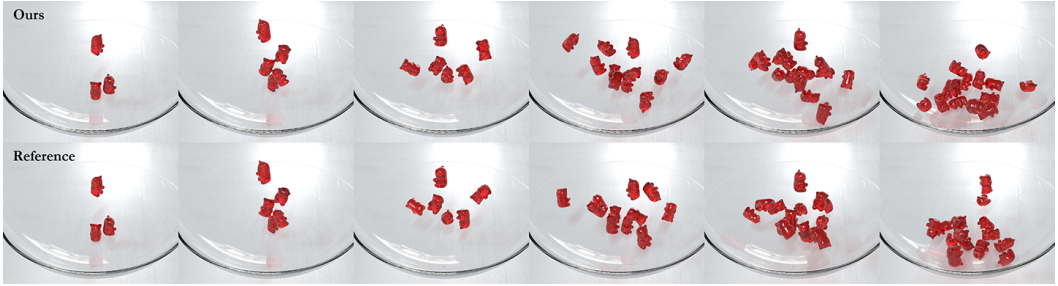


Fig. 3. Several hyperelastic gummy bears collide as they are dropped into a bowl. (Top) Results using our generalized model with $\kappa > 0, \mu > 0, \eta = 0$, with no viscoplasticity or elastoplasticity. (Bottom) Reference results simulated with traditional MPM.

Such an approach is also justified from the perspective of VFX [Frost et al. 2017; Geiger et al. 2006], where a particular physical phenomenon is simulated repeatedly with small variations.

However, there is a growing need for methods in machine learning and robotics that can automatically *infer* unknown physical parameters of materials from data, for learning, inference, planning and control. A particular focus area within this domain is the nascent (but rapidly growing) field of *differentiable physics* [Du et al. 2021; Geilinger et al. 2020; Holl et al. 2020; Li et al. 2023], where a physics engine is embedded as an additional layer inside a neural network, and the gradients of the loss function with respect to unknown physical parameters are backpropagated through the physics engine. Such an approach is known to be more efficient than other data-driven alternatives [Song and Boularias 2020a,b; Wang et al. 2020, 2021]. Unfortunately, for lack of a general solution, researchers have developed their own customized differentiable engines for studying particular phenomena of interest [Mozaffar and Cao 2021; Qiao et al. 2021; Srinivasan et al. 2021; Takahashi and Lin 2019b]. Those approaches are not feasible for exploratory tasks, such as manipulating unknown objects or locomotion on unknown terrain, where the material properties of the objects or the terrain are not known in advance. This motivates the need for a physics engine that is capable of simulating materials with widely different material properties by only changing the values of certain parameters during run-time, so as to be useful in gradient descent.

1.1 Contributions

Our work seeks to address this problem by designing a parameterized physics engine for inviscid fluids, fluids with Newtonian viscosity, hyperelastic solids, viscoplastic and elastoplastic materials. To achieve this, we design a generalized constitutive model that is capable of exhibiting all the aforementioned material behaviors and employ the versatile Material Point Method (MPM) [Hu et al. 2019; Jiang et al. 2016] for discretizing it in space, due to MPM’s ability to handle both large deformations and topology change in a unified fashion, while minimizing numerical dissipation. We present several side-by-side comparisons to highlight that our generalized model produces visually similar results as the specific constitutive models that have been used in prior work. Furthermore, we show how our framework can be implemented in the Taichi differentiable programming language [Hu et al. 2020] to automatically compute gradients for a given loss function by backpropagating through the simulation. This allows us to automatically learn unknown material properties of objects from simulation data. We demonstrate the robustness of our model in simulation, which can converge to the correct physics even for problems where multiple different materials interact with each other (see Figure 12).

In contrast to prior studies that have also investigated the formulation of constitutive models for different materials, our proposed methodology unifies various existing/newly proposed models

and allows for designing those material behaviors that potentially match with uncharted materials. For example, we show how the material properties span in the pentagon, as shown in Fig. 1, and numerous unknown material behaviors can be formed under the umbrella of our generalized constitutive model (see Figure 1). Moreover, the computational framework (see Section 5) emanating from such a generalized constitutive model can significantly alleviate the complexity of simulations that have to resort to the individual numerical methods used in prior studies [Raveendran et al. 2011; Stomakhin et al. 2013; Yue et al. 2015]. This can be more significant given that inverse learning with differentiable physics in general needs a material “library” to have a wider range of tuning space. We show that the integration of our proposed approach with the Taichi differentiable engine can readily transfer the initial guess to the target physics and yield excellent agreement with the ground truth (see Figure 10). Due to the versatility of our generalized constitutive model, the resulting inverse simulation with differentiable physics can readily handle complex scenarios with multiple objects and materials, which has been considered to be challenging in prior work [Wang et al. 2020].

2 RELATED WORK

We review relevant prior work in computer graphics which, to our knowledge, has considered generalized physics simulation from two different perspectives so far: a) generalized discretization approaches (which use different constitutive models, but the same spatial/temporal discretization), and b) generalized modeling approaches (which use the same constitutive model).

2.1 Generalized Discretization Approaches

Discretization on Meshes: Elastic and inelastic deformation (including viscoelasticity, plasticity and fracture) of non-rigid curves, surfaces and solids were introduced to computer graphics by the seminal works of Terzopoulos et al. [Terzopoulos and Fleischer 1988; Terzopoulos et al. 1987]. Since then, FEM discretization has been used for elasticity, plasticity and fracture [Bao et al. 2007; Martin et al. 2010; O’Brien and Hodgins 1999; Sifakis and Barbic 2012], viscoelasticity [Wojtan and Turk 2008], viscoplasticity [Bargteil et al. 2007], water [Thürey et al. 2010; Wojtan et al. 2010] and solid-fluid coupling [Clausen et al. 2013]. While many of these works were done by several different groups, leading to slightly different discretizations, a generalized approach for solid simulation on 1D, 2D and 3D meshes, termed *projective dynamics* (PD), was proposed in [Bouaziz et al. 2014; Liu et al. 2017; Narain et al. 2016] that can accommodate several different constitutive models. Projective dynamics succeeds *position-based dynamics* (PBD) [Macklin et al. 2014; Müller et al. 2007] which, albeit not as accurate as PD, is more efficient to be useful in real-time environments. Besides FEM, researchers have also explored discrete operators inspired by fluid mechanics for simulating co-dimensional fluid phenomena [Da et al. 2015, 2016; Zhu et al. 2015, 2014]. A mixed-dimensional discretization for elasticity on non-manifold surfaces was proposed in [Chang et al. 2019].

Discretization on Grids: Versatile simulation of sand, mud and snow was pioneered by the early work of [Sumner et al. 1999]. Later, a particle-level set approach was shown to be capable of simulating multiple interacting liquids with phase change [Losasso et al. 2006]. A monolithic system for pressure and viscosity was proposed in [Larionov et al. 2017], which later inspired monolithic formulations for viscosity with two-way coupling [Takahashi and Lin 2019a], generalized pressure-viscosity-contact simulation [Takahashi and Batty 2020], and frictional contact of granular materials with two-way rigid body coupling [Takahashi and Batty 2021]. Apart from fluids, uniform grids have also been used for simulating highly constrained thin strands [Sueda et al. 2011] with applications to biomechanics [Sachdeva et al. 2015], elastic solids with frictional contact [Levin et al. 2011], skin [Li et al. 2013], cloth [Weidner et al. 2018], and coupling fluids with elastic solids [Teng et al. 2016].

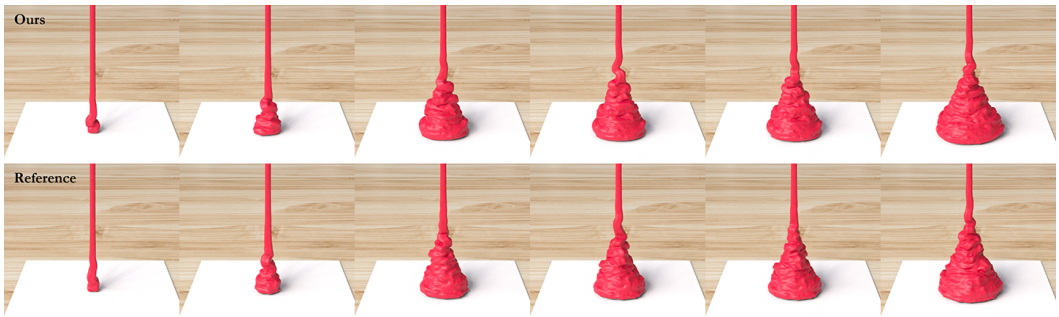


Fig. 4. Classical liquid rope coiling effect exhibited by a Newtonian viscous liquid when poured on a rigid surface. (Top) Results produced by using our generalized constitutive model. (Bottom) Reference results simulated using traditional MPM.

Discretization with Particles: The Smoothed Particles Hydrodynamics (SPH) method [Desbrun and Gascuel 1996] has been used for versatile simulation of fluids [Ihmsen et al. 2014] with viscosity [Peer et al. 2015], two-way coupled rigid bodies [Akinci et al. 2012], spray/foam and air bubbles [Ihmsen et al. 2012], elastic solids [Becker et al. 2009], granular materials [Alduán and Otaduy 2011; Ihmsen et al. 2013], surface tension and adhesion [Akinci et al. 2013], and snow [Gissler et al. 2020]. Particle discretizations based on *peridynamics* have been employed for simulating thin sheets with diffusion effects [Han et al. 2021] and solid-fluid coupling with fracture [Truong et al. 2021], where discontinuities introduced due to cracks make discrete differentials invalid. The versatile position-based dynamics framework [Müller et al. 2007] has been used to simulate solids, fluids, granular materials and two-way coupling between them [Macklin and Müller 2013; Macklin et al. 2014]. Particles have also been used to simulate large-scale multi-body dynamics problems with frictional contact [Mazhar et al. 2015].

Discretization with Hybrid Methods: The FLIP method was introduced to computer graphics through the seminal work of Zhu and Bridson [2005] for sand simulation. The Affine Particle-In-Cell (APIC) method [Jiang et al. 2015] was proposed later to reduce dissipation and prevent instabilities. In this domain, the *Material Point Method* (MPM) [Hu et al. 2019; Jiang et al. 2016] has emerged as a versatile method for simulating snow [Stomakhin et al. 2013], phase change [Gao et al. 2018b; Stomakhin et al. 2014], sponges and foam [Ram et al. 2015; Yue et al. 2015], wet and dry sand [Klár et al. 2016; Tampubolon et al. 2017], wet cloth [Fei et al. 2018], particle-laden fluids [Gao et al. 2018a], fracture [Wolper et al. 2019], baking and cooking [Ding et al. 2019], and diffusion-driven phenomena [Su et al. 2021; Xue et al. 2020]. MPM has the advantage of using particles for carrying material attributes, which avoids numerical dissipation issues characteristic of grid-based schemes. The use of a background grid permits the use of regular numerical stencils for force computations, allowing for good cache-locality and parallelism, and avoiding expensive neighbor lookups. The concept of hybridization (or more generally, *embedded* simulation) has also been proposed for solids [Sifakis et al. 2007a,b; Teran et al. 2005] for producing well-conditioned systems for biomechanics.

In summary, all the prior works referenced above use *different* constitutive models for simulating different material behaviors. This leads to different discrete systems per constitutive model, even if the same spatial discretization method was used, requiring the need for different numerical solvers. For example, MPM can use PCG with semi-implicitly linearized dynamic equations for snow simulation [Stomakhin et al. 2013], but requires an optimization-based solver for sponges and foam [Ram et al. 2015].

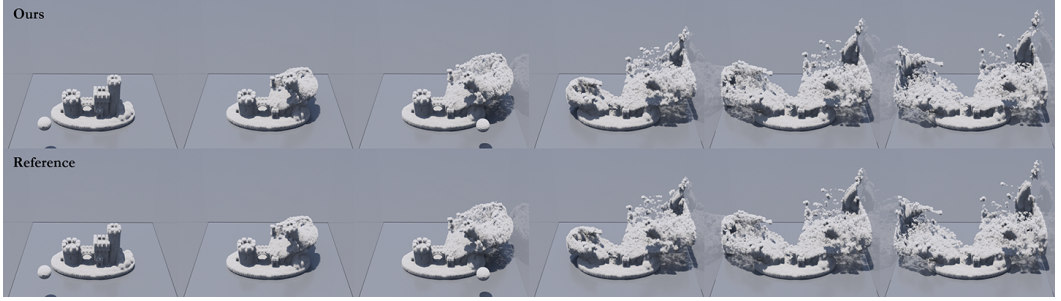


Fig. 5. Two snow balls collide with a snow castle. (Top) Our generalized model can achieve elastoplastic behavior when the critical coefficients f_c , f_s and the hardening factor f_h are appropriately selected while keeping the viscoplasticity turned off. (Bottom) Reference results simulated using [Stomakhin et al. 2013].

2.2 Generalized Constitutive Modeling

As far as we know, the prior works that are most similar to our proposed approach are [Fang et al. 2019; Nagasawa et al. 2019]. The method proposed in [Nagasawa et al. 2019] is limited to shear-thinning viscous liquids. While impressive behaviors can be simulated for both viscoelastic and elastoplastic materials by the method proposed in [Fang et al. 2019], it employs a damping method for viscosity which does not allow for arbitrary viscosity values.

Our approach is inspired by the recent work in [Su et al. 2021; Xue et al. 2020] based on *non-local modeling* [Eringen 1992; Sobolev 2014]. Unlike the prior works [Su et al. 2021; Xue et al. 2020] that focused on the specific phenomena of diffusion and viscosity, we explore a much broader spectrum of material behaviors, as illustrated in Figure 1. The key ideas behind our formulation are the design of a generalized Kirchhoff stress tensor that can describe hyperelasticity, Newtonian viscosity and inviscid fluids, and the use of pre-projection and post-correction rules for simulating material behaviors that involve plasticity, including elastoplastic flows and viscoplastic flows.

3 OVERVIEW OF THE GENERALIZED KIRCHHOFF STRESS TENSOR

We assume a primary additive decomposition of any arbitrary Kirchhoff stress tensor $\boldsymbol{\tau}$ into a volumetric stress tensor $\boldsymbol{\tau}^v$, a deviatoric stress tensor $\boldsymbol{\tau}^s$, and a rate-dependent stress tensor $\boldsymbol{\tau}^N$:

$$\boldsymbol{\tau} = \boldsymbol{\tau}^v + \boldsymbol{\tau}^s + \boldsymbol{\tau}^N \quad (1)$$

3.1 Volumetric Stress

The volume-dependent energy density can be defined as follows:

$$W^v(J) = \frac{\kappa}{2} \left[\frac{1}{2}(J^2 - 1) - \ln J \right], \quad (2)$$

where κ denotes the bulk modulus and J is the relative volume. The volumetric stress tensor is given by:

$$\boldsymbol{\tau}^v = \frac{\kappa}{2}(J^2 - 1)\mathbf{I} \quad (3)$$

3.2 Deviatoric Stress

The shear-dependent energy density is defined as follows:

$$W^s(\mathbf{F}) = \frac{\mu}{2} [\text{Tr}(\mathbf{F}^T \mathbf{F}) - 3], \quad (4)$$

where μ denotes the shear modulus, F is deformation gradient. The deviatoric stress tensor can be expressed as:

$$\tau^s = \mu \left[AA^T - \frac{1}{3} \text{Tr}(AA^T) I \right], \quad (5)$$

where $A = J^{-1/3} I$ is the isochoric deformation gradient.

3.3 Rate-Dependent Stress

In general, both fluids and soft solids share the same non-linear kinematics under large deformations. The primary distinction between a (hyper)elastic solid and a (Newtonian) fluid is that the former maintains a *permanent* memory of its initial (or reference) configuration, to which it relaxes to, whereas all configurations are equivalent in a simple fluid and only *rates of deformation* are important. To realize the shear stress induced by the rate of deformation, we add the Kirchhoff stress τ_N corresponding to the Newtonian-type viscous stress σ_N , as defined below:

$$\sigma^N = \frac{\eta}{J} \epsilon, \quad \tau^N = J \sigma^N = \eta \epsilon \quad (6)$$

where $\epsilon = \frac{1}{2} (\nabla v + \nabla v^T)$.

3.4 Unification

The total Kirchhoff stress tensor τ can be reformulated as:

$$\tau = \underbrace{\frac{\kappa}{2} (J^2 - 1) I}_{\tau^v} + \underbrace{\mu \left[AA^T - \frac{1}{3} \text{Tr}(AA^T) I \right]}_{\tau^s} + \underbrace{\eta \epsilon}_{\tau^N}. \quad (7)$$

Equation (7) can be combined with the following momentum equation for updating the physical state:

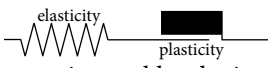
$$\rho \ddot{x} = \nabla \cdot \sigma - f^{\text{ext}} \quad (8)$$

where ρ is the material density, σ is the Cauchy stress tensor, and f^{ext} is the vector of external forces, such as gravity. By tuning μ and η in equation (7), equation (8) can be used to describe:

- (1) ($\eta = 0$): hyperelastic solids,
- (2) ($\mu = 0$): Newtonian viscous fluids,
- (3) ($\mu = 0, \eta = 0$): inviscid fluids.

3.5 Plastic Projection and Correction

To further cover elastoplasticity and non-Newtonian viscosity, we follow [Stomakhin et al. \[2013\]](#) to manipulate the elastic part of the deformation gradient before arriving at the total Kirchhoff stress and apply a plastic correction to the shear-dependent stress τ^s in equation (5).

3.5.1 Elastoplasticity. In order to handle the elastoplastic behaviors exhibited by materials like snow, we follow [Stomakhin et al. \[2013\]](#) and  introduce hyperparameters f_s, f_c and f_h , representing the stretching, compression and hardening factors respectively, to have more control over the material's response to deformation. We start with a 1D rheological example. The plastic element represents the response due to permanent deformation, without which the material will behave purely elastically. In 3D, the total deformation gradient is multiplicatively decomposed into elastic and plastic parts as $F = F^e F^p$. We follow [Stomakhin et al. \[2013\]](#) and apply a singular value decomposition to F^e , followed by clamping the singular values below a threshold. Note that in order to handle plastic hardening, the Lamé coefficients need to be updated accordingly.

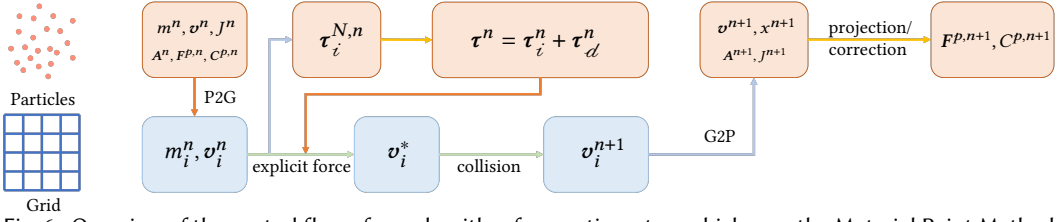


Fig. 6. Overview of the control flow of our algorithm for one time step, which uses the Material Point Method (MPM) to maintain a number of physical quantities on the particles and rasterizes them on a background Cartesian grid for computing forces, which are subsequently used for updating the entire state of the system.

3.5.2 Viscoplasticity. We follow Nagasawa et al. [2019] and draw the 1D rheological example, where an elastic spring is attached to a structure consisting of a Coulomb friction element with yield stress σ_Y and a dash-pot with viscosity η . Similar to [Yue et al. 2015], we introduce plasticity to the shear-dependent stress τ^s , and denote its scalar magnitude as $s = \|\tau^s\|_F$ and its normalized form $\bar{\tau}^s = \frac{1}{s}\tau^s$, where $\|\cdot\|_F$ calculates the Frobenius norm. The von-Mises yield condition [Simo and Hughes 2006] is:

$$\Phi(s) = s - \sqrt{\frac{2}{3}}\sigma_Y \leq 0 \quad (9)$$

makes precise the limits of the elastic regime in terms of a material-dependent yield stress σ_Y . When the inequality (9) is satisfied, the material's response remains elastic or viscoelastic. Otherwise, a limited plastic flow is introduced. We refer to the quantity $\max(0, \Phi(s))$ as the *yield excess*, and the inequality $\Phi(s) \leq 0$ as the *yield stress*. Following Yue et al. [2015], we adopt the volume-preserving time-dependent left Cauchy-Green tensor $\bar{b}^e = A^e(A^e)^T$ and express its time derivative as below:

$$\dot{\bar{b}}^e = \mathbf{L}\bar{b}^e + \bar{b}^e\mathbf{L}^T - \frac{2}{3}\text{Tr}(\tau^s)\gamma\bar{\tau}^s, \quad (10)$$

where $\mathbf{L} = \nabla\mathbf{v} = \dot{\mathbf{F}}\mathbf{F}^{-1}$. Equation (10) incorporates the combined effects of the flow field itself as well as the plastic flow. Further, we introduce a power law relating the plastic flow rate as follows:

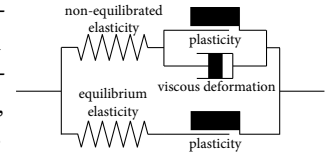
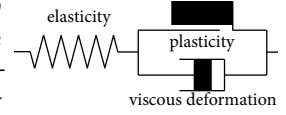
$$\gamma = \gamma(s) = \max\left(0, \frac{\Phi(s)}{\eta}\right)^{1/h} \quad (11)$$

where η is the viscosity coefficient. By varying the Herschel-Bulkley power h , our model can describe *shear-thinning* (when $h < 1$) and *shear-thickening* (when $h > 1$), as shown in Figures 8 and 9.

4 GENERALIZED CONSTITUTIVE MODEL

To handle both elastoplastic behavior and time-dependent viscoplasticity, we follow Fang et al. [2019] and update the 1D rheological example by connecting the previous models in parallel, representing the identical total deformation that the material is undergoing, that is, $\mathbf{F} = \mathbf{F}_{top} = \mathbf{F}_{bot}$. Note that the bottom part of this model is *time-independent*, whereas the introduction of viscoplasticity makes the top component *time-dependent*. The total stress tensor τ can be naturally computed by adding the two individual stresses: $\tau = \tau_i + \tau_d$, where the subscripts i and d denote the *time-independent* and *time-dependent* components, respectively. Further expanding the Kirchhoff stress gives us the following expression:

$$\tau = \underbrace{\tau_i^v + \tau_i^s + \tau_i^N}_{\tau_i} + \underbrace{\tau_d^v + \tau_d^s + \tau_d^N}_{\tau_d} \quad (12)$$



where $\tau_{\mathcal{d}}^N$ is in fact $\mathbf{0}$ since the Newtonian viscous stress is time-independent.

5 NUMERICAL METHOD

For numerical simulations, we explicitly track A and J at each particle. Additionally, to account for different types of plasticity, we also store and update the plastic deformation gradient F^p . Other variables can be obtained on the fly. We further replace the full relative volume J and the total isochoric deformation gradient A with only the elastic portions J^e and A^e when computing the elastic stress tensor. Integrating equations (8) and (7) provides the following governing equations:

$$\rho \frac{Dv}{Dt} = \nabla \cdot \sigma + \rho g, \quad \sigma = \frac{1}{J} \tau, \quad \tau = \tau_i + \tau_{\mathcal{d}}, \quad (13)$$

where

$$\begin{aligned} \tau_{\star} &= \tau_{\star}^v + \tau_{\star}^s + \tau_{\star}^N, & \tau_{\star}^v &= \frac{\kappa_{\star}}{2} [(J_{\star}^e)^2 - 1] I, \\ \tau_{\star}^s &= \mu_{\star} \left[\bar{b}_{\star}^e - \frac{1}{3} \text{Tr}(\bar{b}_{\star}^e) I \right], & \tau_{\star}^N &= \frac{\eta_{\star}}{2} (\nabla v + \nabla v^T), \end{aligned} \quad (14)$$

The subscript \star can be either i or \mathcal{d} , $\bar{b}_{\star}^e = A_{\star}^e (A_{\star}^e)^T$, $\eta_{\mathcal{d}} = 0$. We discretize equation (14) in space using the versatile Material Point Method (MPM) [Jiang et al. 2016], where the use of particles in conjunction with a background Cartesian grid provides a generalized interface for simulating both solid-like and fluid-like materials.

Notation. We use subscript i for quantities stored on grid nodes and drop the subscript for quantities stored on particles. We use subscripts e and p to denote elastic and plastic components. Superscript $*$ denotes the predicted value of a quantity that requires further modifications. Our full method is summarized below:

- (1) **Rasterization.** Mass and momentum are transferred from particles to the grid in a conservative form. See Section 5.1.
- (2) **Apply forces and update velocities on the grid.** We obtain the stress-based forces on the grid and apply the forward Euler method to update grid velocities. See Section 5.2
- (3) **Compute grid-based collision.** We modify grid velocities based on boundary conditions to avoid visual artifacts.
- (4) **Update deformation variables.** We update A , J , and apply pre-projection and post-correction to account for plasticity.
- (5) **Update particle velocities and positions.** After the velocities on the grid are finalized, we interpolate them at the particle locations using a linear combination of the FLIP velocity and the PIC velocity: $v^{n+1} = \alpha v_{p,\text{flip}}^{n+1} + (1 - \alpha) v_{p,\text{pic}}^{n+1}$, where $v_{p,\text{flip}}^{n+1} = v^n + \sum_i (v_i^{n+1} - v_i^n) w_{ip}$ and $v_{p,\text{pic}}^{n+1} = \sum_i v_i^{n+1} w_{ip}$, and α is the blending coefficient.

5.1 Rasterization

We rasterize quantities m and v at time t^n from particle locations to grid nodes as follows: $m_i^n = \sum m w_{ip}$, where w_{ip} are the interpolation weights from particles to grid nodes. Subsequently, v is rasterized according to the following normalization during the interpolation step: $v_i^n = \frac{1}{m_i^n} \sum m v^n w_{ip}$. In return, we can compute the rate of strain tensor at particles as shown below:

$$\nabla v^n = \sum_i v_i^n (\nabla w_{ip})^T, \quad \tau_i^n = \frac{\eta}{2} \left(\nabla v^n + (\nabla v^n)^T \right) \quad (15)$$

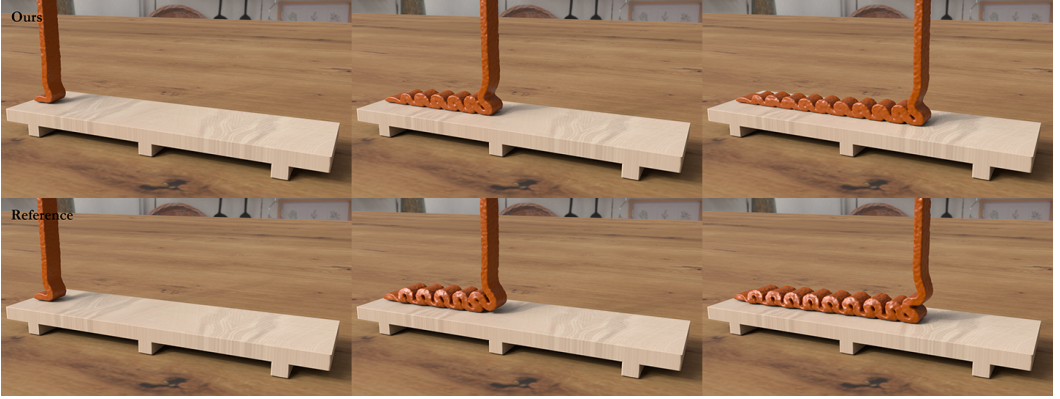


Fig. 7. Symmetric buckling patterns of Newtonian fluid using (top) our generalized model and (bottom) traditional MPM.

5.2 Velocity Update

Following conventional MPM, forces on the grid are computed as:

$$\mathbf{f}_i^n = - \sum V^n \boldsymbol{\sigma}^n \nabla w_i p = - \sum V^0 \boldsymbol{\tau}^n \nabla w_{ip} \quad (16)$$

where $V^n = J^n V^0$ and $\boldsymbol{\sigma}^n = \frac{1}{J^n} \boldsymbol{\tau}^n$ are used. The forward Euler update for grid velocities is straightforward: $\mathbf{v}_i^* = \mathbf{v}_i^n + \frac{\Delta t}{m_i} \mathbf{f}_i^n + \Delta t \mathbf{g}$, where \mathbf{g} represents the external accelerations (including gravity).

5.3 Grid-Based Collision

We closely follow the strategy proposed in [Stomakhin et al. 2013]. The grid velocity \mathbf{v}_i is transformed into the reference coordinates of the collision object and friction or other projection is applied, based on the specific boundary condition. The modified velocity \mathbf{v}_i^* is then transformed back to the world frame.

5.4 Update of Deformation Variables

We separately update \mathbf{A} and J using the following rules:

$$\mathbf{A}^{n+1} = e^{\text{dev}(\nabla \mathbf{v}^{n+1}) \Delta t} \mathbf{A}^n, \quad J^{n+1} = (1 + \Delta t \text{Tr}(\nabla \mathbf{v}^{n+1})) J^n, \quad (17)$$

where $\text{dev}(\nabla \mathbf{v}) = \nabla \mathbf{v} - \frac{1}{3} \text{Tr}(\nabla \mathbf{v}) \mathbf{I}$ and d is the ambient dimension. The exponential update ensures that $\det \mathbf{A} = 1$. Note that \mathbf{A} and J represent the *total* deformation state variables. Reconstruction of \mathbf{A}^e and J^e is necessary before computing the total Kirchhoff stress. Ideally, we can update the variables as listed below:

$$\begin{aligned} \mathbf{F}^{n+1} &= (J^{n+1})^{\frac{1}{3}} \mathbf{A}^{n+1}, \quad \mathbf{F}_{\star}^{e,*} = \mathbf{F}^{n+1} (\mathbf{F}_{\star}^{p,n})^{-1} \\ J_{\star}^{e,*} &= \det(\mathbf{F}_{\star}^{e,*}), \quad \mathbf{A}_{\star}^{e,*} = (J_{\star}^{e,*})^{-\frac{1}{3}} \mathbf{F}_{\star}^{e,*}, \end{aligned} \quad (18)$$

where \star represents i or d . In practice, we found that tracking \mathbf{C}_d^p instead of \mathbf{F}_d^p helps improve the performance since the eigendecomposition is no longer needed. We can calculate $\mathbf{A}_d^{e,*} (\mathbf{A}_d^{e,*})^T$, denoted as $\bar{\mathbf{b}}_d^{e,*}$, as: $\bar{\mathbf{b}}_d^{e,*} = \mathbf{F}^{n+1} (\mathbf{C}_d^{p,n})^{-1} (\mathbf{F}^{n+1})^T$, $\bar{\mathbf{b}}_d^{e,*} = \det(\bar{\mathbf{b}}_d^{e,*})^{-1/3} \bar{\mathbf{b}}_d^{e,*}$. To account for both the elastoplastic and the viscoplastic behaviors, we keep track of both $\mathbf{A}_i^e, \mathbf{F}_i^p, J_i^e$ and $\bar{\mathbf{b}}_d^e, \mathbf{C}_d^p$, each of which is associated with a set of parameters κ, μ .

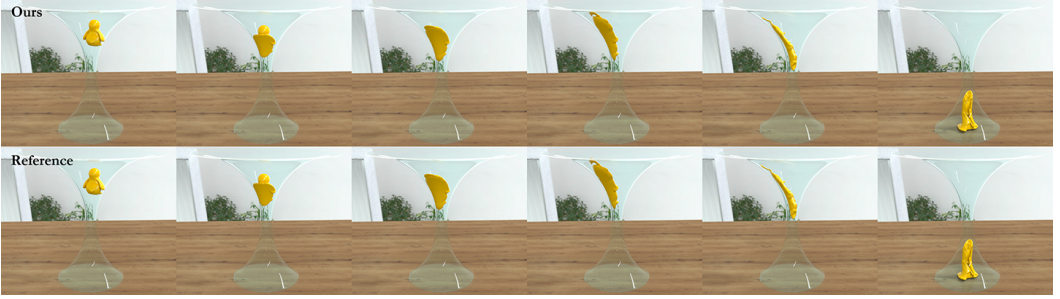


Fig. 8. A shear-thinning liquid flows immediately when it hits the hourglass, but largely holds its shape as it falls on the ground and the applied stresses decrease. Simulations produced by (top) our generalized model and (bottom) traditional MPM.

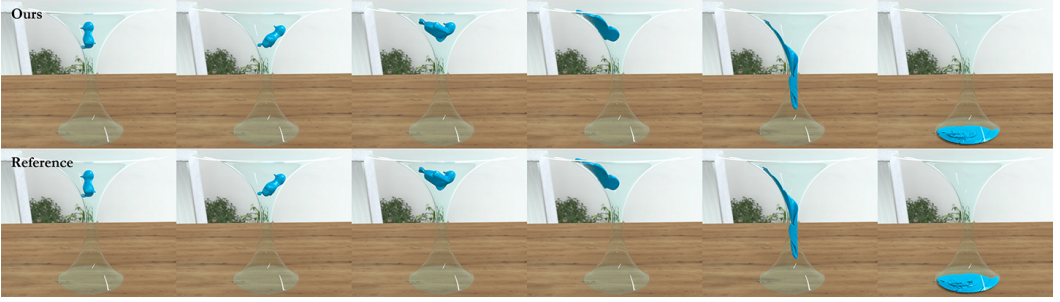


Fig. 9. A shear-thickening liquid exhibits resistance to flow initially and bounces elastically, but eventually flows down as the velocities decrease. Simulations produced by (top) our generalized constitutive model and (bottom) traditional MPM.

5.4.1 *Pre-Projection.* We model the elastoplasticity by setting a restriction on the singular values of the elastic deformation gradient:

$$\mathbf{F}_i^{e,*} = \mathbf{U}\Sigma^*\mathbf{V}^T, \quad \Sigma_{ii} = \max(f_c, \min(f_s, \Sigma_{ii}^*)) \quad (19)$$

Then, we can finalize the elastic/plastic deformation gradient:

$$\mathbf{F}_i^{e,n+1} = \mathbf{U}\Sigma\mathbf{V}^T, \quad \mathbf{F}_i^{p,n+1} = \mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{F}_i^{p,*} \quad (20)$$

as well as the Lamé coefficients:

$$\kappa_i^{n+1} = \kappa_{i,0}e^{f_h(1-\det\mathbf{F}_i^{p,n+1})}, \quad \mu_i^{n+1} = \mu_{i,0}e^{f_h(1-\det\mathbf{F}_i^{p,n+1})} \quad (21)$$

We use the formulas $J_i^{e,n+1} = \det\Sigma$, $\mathbf{A}_i^{e,n+1} = (J_i^{e,n+1})^{-\frac{1}{3}}\mathbf{F}_i^{e,n+1}$ to compute \mathbf{A}^e and J^e . Finally, we compute the Kirchhoff stress as:

$$\begin{aligned} \boldsymbol{\tau}_i^{n+1} = & \kappa_i^{n+1} \left((J_i^{e,n+1})^2 - 1 \right) \mathbf{I} + \mu_i^{n+1} \left(\bar{\mathbf{b}}_i^{e,n+1} - \frac{1}{3}\text{Tr}(\bar{\mathbf{b}}_i^{e,n+1})\mathbf{I} \right) \\ & + \frac{\eta}{2} \left(\nabla\mathbf{v}_i^{n+1} + (\nabla\mathbf{v}_i^{n+1})^T \right) \end{aligned} \quad (22)$$

where we denote $\bar{\mathbf{b}}_i^{e,n+1} = \mathbf{A}_i^{e,n+1}(\mathbf{A}_i^{e,n+1})^T$ for brevity.

5.4.2 *Post-Correction.* The main focus of modeling viscoplasticity is on the shear-dependent stress tensor $\boldsymbol{\tau}_d^s$. The predicted shear stress can be computed via:

$$\boldsymbol{\tau}_d^{s,*} = \mu_d \left(\bar{\mathbf{b}}_d^{e,*} - \frac{1}{3} \text{Tr}(\bar{\mathbf{b}}_d^{e,*}) \mathbf{I} \right). \quad (23)$$

Its Frobenius norm is obtained by $s_d^* = \|\boldsymbol{\tau}_d^{s,*}\|_F$. Next, we substitute s_d^* into the yield condition $\Phi(s) = s - \sqrt{\frac{2}{3}}\sigma_Y$. If it is satisfied, the response remains elastic and no post-correction is needed. Otherwise, the plastic correction is given by:

$$\bar{\mathbf{b}}_d^{e,n+1} - \bar{\mathbf{b}}_d^{e,*} = -\frac{2}{3} \Delta t \text{Tr}(\bar{\mathbf{b}}_d^{e,n+1}) \gamma(s_d^{n+1}) \bar{\mathbf{s}}_d^{n+1}. \quad (24)$$

where $s_d^{n+1} = \|\boldsymbol{\tau}_d^{n+1}\|_F$ and $\bar{\mathbf{s}}_d^{n+1} = \boldsymbol{\tau}_d^{n+1}/s_d^{n+1}$. As shown in [Yue et al. 2015], this function can be transformed into a single *scalar* formula as follows:

$$s_d^{n+1} - s_d^* = -2\bar{\mu}_d \Delta t \gamma(s_d^{n+1}), \quad (25)$$

where $\bar{\mu}_d = \frac{1}{3} \text{Tr}(\bar{\mathbf{b}}_d^{e,*}) \mu_d$. By solving equation (25), we can obtain $\bar{\mathbf{b}}_d^{e,n+1} = \frac{1}{\mu_d} \mathbf{s}_d^{n+1} + \frac{1}{3} \text{Tr}(\bar{\mathbf{b}}_d^{e,*})$. The corrected shear stress is:

$$\boldsymbol{\tau}_d^{e,n+1} = \mu_d \left(\bar{\mathbf{b}}_d^{e,n+1} - \frac{1}{3} \text{Tr}(\bar{\mathbf{b}}_d^{e,n+1}) \mathbf{I} \right) \quad (26)$$

Together with the volumetric stress and the Newtonian viscous stress, we can compute the total Kirchhoff stress as follows:

$$\boldsymbol{\tau}_d^{n+1} = \kappa_d^{n+1} [(J^{n+1})^2 - 1] \mathbf{I} + \mu_d \left(\bar{\mathbf{b}}_d^{e,n+1} - \frac{1}{3} \text{Tr}(\bar{\mathbf{b}}_d^{e,n+1}) \mathbf{I} \right) \quad (27)$$

Here we use J^{n+1} instead of $J_d^{e,n+1}$ because the plastic deformation is volume-preserving. Although we have already applied the correction directly to the shear stress, the isochoric deformation gradient \mathbf{A}_d^* and the right Cauchy-Green plastic deformation gradient tensor \mathbf{C}_d^p remain unchanged. So the next step is to incorporate this change in plasticity: $\mathbf{C}_d^{p,*} = \mathbf{F}^{n+1} \bar{\mathbf{b}}_d^{e,n+1} (\mathbf{F}^{n+1})^T$. We finalize the tensor by applying normalization: $\mathbf{C}_d^{p,n+1} = \det(\mathbf{C}_d^{p,*})^{-1/3} \mathbf{C}_d^{p,*}$.

6 INVERSE LEARNING WITH DIFFERENTIABLE PHYSICS

Our generalized model provides us a parameterized physics engine, and we employ inverse learning to reproduce given observations. We assume that particle sequences for objects of interest are provided. This can be accomplished using computer vision techniques like NR-NeRF [Chen et al. 2022; Tretschk et al. 2021]. By utilizing our generalized constitutive model, we can effectively replicate various material types using only one set of parameters.

The time stepping can be viewed as a mapping between two states: $(x^n, v^n, J^n, F^{p,n}, A^{p,n})$ and $(x^{n+1}, v^{n+1}, J^{n+1}, F^{p,n+1}, A^{p,n+1})$, subject to a set of constitutive model parameters Θ . For an explicit simulation scheme, this mapping is essentially a collection of basic mathematical operators, which can be differentiated using AutoDiff. Given a target particle sequence of an object $\{\bar{x}^n\}$, we use the following loss between the output sequence $\{x^n\}$ to guide the recovery of the unknown physical parameter set Θ :

$$L_\Theta(\{x^n\}, \{\bar{x}^n\}) = \sum_{f=1}^N \text{Sinkhorn}(x^{fN_s}, \bar{x}^f) \quad (28)$$

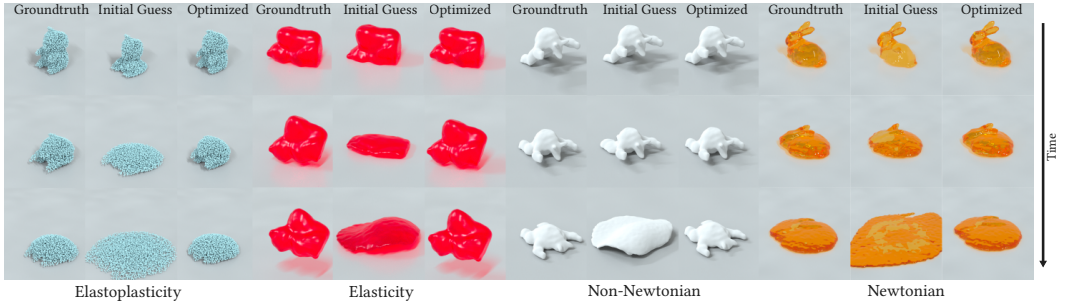


Fig. 10. With the same initial guess, our generalized model is able to mimic multiple types of materials via inverse learning.

where N is the total number of frames, and N_s is the number of substeps per frame. We assume the object is homogeneous, within which all particles share the same physical parameters. We use the Sinkhorn distance [Feydy et al. 2019] to compute the loss function between two point clouds.

Differentiability. Return mappings involve maximum operators, which make certain parameters like f_s , f_c , and σ_Y only piecewise differentiable. Without executing plasticity correction, gradients will not be applied to these parameters, resulting in gradient vanishing. To address this issue, we initialize f_s and f_c to values that are very close to 1, and σ_Y to a small value. This enables the activation of return mappings at the start, thereby alleviating the problem of gradient vanishing.

Implementation. We developed our explicit MPM simulator with our generalized constitutive model using the Taichi differentiable programming language [Hu et al. 2020] to utilize its AutoDiff system. Similar to the approach taken in [Li et al. 2023], we integrated the MPM simulator as a differentiable layer within the PyTorch framework [Paszke et al. 2019], which allows for organized management of tunable design variables by PyTorch. This integration also enables convenient application of PyTorch optimizers like ADAM. Figure 11 illustrates the computational graph of our inverse learning framework. Note that AutoDiff requires access to all substeps at the time of backpropagation. To optimize storage, we only save states every hundred substeps, and the intermediate substeps between two stored steps are recomputed on the GPU as necessary.

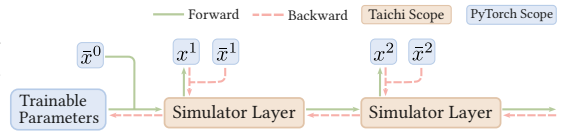


Fig. 11. The computational graph of our inverse learning framework implemented in the Taichi language. \bar{x}^i is the ground truth data. x^i is the output of the simulator layer.

7 EXPERIMENTAL RESULTS

7.1 Forward Simulations and Comparisons

We simulated all examples on an Intel(R) Xeon(R) CPU E5-1620 v4 @ 3.50GHz. Specific timings for all examples are summarized in Table 1 and the various parameters used are listed in Table 2.

Inviscid Liquid Spray. We simulated a weakly-compressible liquid example to first validate that our generalized model can capture liquid behaviors and recover simulations from the reference constitutive model. Figure 2 shows how the inviscid fluid spouts from a nozzle and flows down under gravity to fill the fountain. Note that the reference model only requires updating the fluid density, which is equivalent to updating J . In contrast, our generalized model also requires keeping track of the matrix A and performing additional operations, such as the reconstruction of the deformation gradient, followed by a projection step. This could be optimized by checking if the

Table 1. (Left) **Simulation Statistics** All simulations were run on Intel(R) Xeon(R) CPU E5-1620 v4 @ 3.50GHz. # **P**: The total number of MPM particles in the simulation. **G**: The background grid resolution. Simulation time **RT** is measured in average seconds per substep. **dt**: time step size of a substep, measured in 1×10^{-5} s.

Table 2. (Right) **Material Parameters** Parameters for all the test materials. **M** represents the maximum value of a float. The unit of κ is $\times 10^3$ and the unit of μ is $\times 10^2$.

Simulation	# P	G	RT	dt	Simulation	κ_t	μ_t	κ_d	μ_d	η	σ_Y	h	f_h	f_s	f_c
Fountain (Fig. 2 (top))	1.06M	256^3	1.13	11.0	Fountain (Fig. 2)	0.1	0	0	0	0	M	1	1	M	-M
Fountain (Fig. 2 (bottom))	1M	256^3	1.05	10.5	Coiling (Fig. 4)	10	0	0	0	50	M	1	1	M	-M
Coiling (Fig. 4 (top))	748K	$128^2 \times 256$	0.79	3	Buckling (Fig. 7)	10	0	0	0	50	M	1	1	M	-M
Coiling (Fig. 4 (bottom))	748K	$128^2 \times 256$	0.83	3	Gummy Bears (Fig. 3)	1.7	3.6	0	0	0	M	1	1	M	-M
Buckling (Fig. 7 (top))	318K	$128^2 \times 256$	0.29	3	Snow Castle (Fig. 5)	1.2	5.8	0	0	0	M	1	10	1.002	0.9925
Buckling (Fig. 7 (bottom))	316K	$128^2 \times 256$	0.30	3	Hourglass (Fig. 9)	0	0	8.3	18.3	0.1	0.1	5	1	M	-M
Gummy Bears (Fig. 3 (top))	1.5M	256^3	2.05	5	Hourglass (Fig. 8)	0	0	8.3	18.3	0.1	50	0.9	1	M	-M
Gummy Bears (Fig. 3 (bottom))	1.5M	256^3	1.81	5											
Snow Castle (Fig. 5 (top))	1M	256^3	2.12	5											
Snow Castle (Fig. 5 (bottom))	1M	256^3	1.91	5											
Hourglass (Fig. 9 (top))	1M	256^3	2.03	5											
Hourglass (Fig. 9 (bottom))	1M	256^3	2.00	5											
Hourglass (Fig. 8 (top))	1M	256^3	2.12	5											
Hourglass (Fig. 8 (bottom))	1M	256^3	2.01	5											

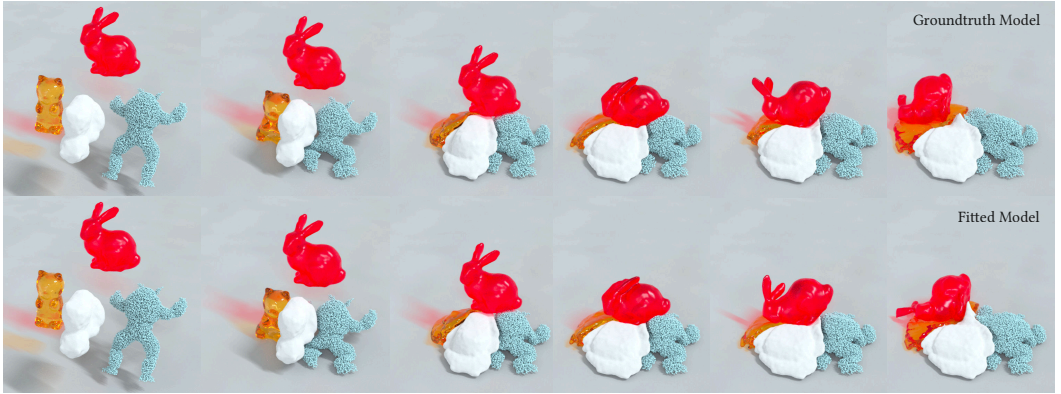


Fig. 12. We use the fitted model obtained from Figure 10 and test its generalizability by using it to simulate different shapes in the same scene. Simulation results with the fitted models closely reproduce the behaviors of the ground truth models.

shear modulus is 0 and deciding whether it is necessary to track **A**. The timings listed in Table 1 reflect this optimization.

Jet Coiling and Buckling. We simulated jet buckling using our generalized constitutive model. Figure 4 (top) shows that our method successfully reproduces the liquid rope coiling effect for classical Newtonian viscous liquids, such as honey. Additionally, we simulated a layer of viscous chocolate folding onto a board (Figure 7). Our method captures the characteristic folding behaviors and produces smooth merging and symmetric buckling patterns. The astute reader may notice some visual differences between the reference results and our simulations. In the reference model, the force exerted by pressure exhibits an approximate linear relationship with the seventh power of density. However, in our generalized model, the pressure varies quadratically with respect to density. We believe this is the cause for the visual differences evident in Figure 7.

Elastoplastic Snow Castle. We threw two snowballs successively at a snow castle, showcasing that our generalized model can simulate brittle materials. We set $f_h > 1$ and f_c, f_s close to 1 to activate both the elastic and plastic regimes. The system abruptly switches from the elastic regime

to the plastic regime once the plastic criterion is violated. Consequently, the snow castle breaks apart after the impart of snowballs, illustrating the typical elastoplastic behavior of snow.

Gummy Bears. We sequentially dropped hyperelastic gummy bears with random orientations onto a plate (Figure 3), which bounce back and create interesting collisions before settling down.

Hourglass. Our method can produce *shear-thinning* (see Figure 8) and *shear-thickening* behaviors (see Figure 9), both of which are typical non-Newtonian viscous behaviors [Su et al. 2021]. For the shear-thickening material, the rate of shear strain increases abruptly on impact, which triggers different mechanisms of viscosity. Strong elastic effects come into play. Subsequently, the object bounces back immediately. Following this elastic behavior, as the second impact gets weak and the momentum gradually decreases, the shear strain decreases and the object begins to flow outwards. On the other hand, the shear-thinning material flows immediately when it hits the hourglass, but largely holds its shape as it falls on the ground and the applied stresses decrease.

7.2 Inverse Learning

Material Fitting. Our generalized model is advantageous in inverse learning since it enables us to replicate various material types using a single set of parameters. In this experiment, we use identical initial parameters and allow the differentiable simulation to optimize the generalized material parameters to fit distinct materials. The initial guesses and optimized values are presented in the supplemental video. It is important to note that the objective of inverse learning is to replicate the behaviors of a given material, which means that the learned parameters may differ from the ground truth model and may not possess physical significance. In addition, it is worth mentioning that all the examples of material fitting discussed in this context involve synthetic data. The results displayed in Figure 10 illustrate that the losses in particle distribution are reduced to a sufficient degree to generate observations that are visually similar.

Generalization. The generalized model that we learned can be extended to various shapes and scenes. Here we included all four learned generalized models in a single scene, where they interact with each other. We permute the initial shapes and use different initial velocities. As illustrated in Figure 12, our learned models can closely replicate the behaviors of the ground truth models.

8 CONCLUSION

We proposed a generalized constitutive model for simulating a large class of materials including inviscid fluids, Newtonian viscosity, hyperelasticity, viscoelasticity and viscoplasticity. Our model is a strict generalization of existing constitutive models that are used in computer graphics. Along with this generalized constitutive model, we developed a generalized computational framework that integrates well with the Material Point Method (MPM) [Jiang et al. 2016] and allows for visually realistic simulations with large deformations.

We demonstrated that our method captures a wide range of material behaviors, which can be elastic, inelastic or plastic, or a mixture of these properties, such as liquid rope coiling, buckling, shear thinning/thickening, snow and hyperelasticity. We also demonstrated the ability of our formulation to allow inverse learning of unknown material properties directly from data using differentiable physics simulations, without making explicit assumptions about the data.

9 LIMITATIONS AND FUTURE WORK

Our model has generated a large number of complex examples, but there remains much work to be done. Although we provided an intuitive interpretation for achieving various material behaviors, different parameters to adjust the material properties were tuned by hand and it would be interesting

to automatically calibrate them to measured models. Additionally, it would be interesting to explore the design of implicit solvers to allow for large time steps. Besides, since the spatial discretization was integrated with MPM, adaptive Lagrangian formulations [Su et al. 2022] may boost the computational efficiency and circumvent visual artifacts, such as numerical fracture, that exist in simulations with large deformations. Finally, while our focus was on the material responses of water, snow, elastic and inelastic solids, it would be interesting to investigate other materials, such as wet and dry sand, frictional contact mechanics, and multi-physics coupling problems under the same generalized umbrella.

ACKNOWLEDGMENTS

We thank Pingying Chen for narrating our video. H. S. and M. A. were supported in part by the National Science Foundation under awards CCF-2110861, IIS-2132972, IIS-2238955 and CCF-2312220, while X. L. and C. J. were supported in part by awards ECCS-2023780, IIS-2153851 and CCF-2153863. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

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. *Proceedings of the ACM in Computer Graphics and Interactive Techniques 2*, 2, Article 14 (2019), 21 pages.
- Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2013. Versatile Surface Tension and Adhesion for SPH Fluids. *ACM Trans. Graph.* 32, 6, Article 182 (2013), 8 pages.
- Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile Rigid-Fluid Coupling for Incompressible SPH. *ACM Trans. Graph.* 31, 4, Article 62 (2012), 8 pages.
- Iván Alduán and Miguel A. Otaduy. 2011. SPH Granular Flow with Friction and Cohesion. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 25–32.
- Zhaosheng Bao, Jeong-Mo Hong, Joseph Teran, and Ronald Fedkiw. 2007. Fracturing Rigid Materials. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 370–378.
- Adam W. Bargteil, Chris Wojtan, Jessica K. Hodgins, and Greg Turk. 2007. A Finite Element Method for Animating Large Viscoplastic Flow. In *ACM SIGGRAPH 2007 Papers*. 16–es.
- Markus Becker, Markus Ihmsen, and Matthias Teschner. 2009. Corotated SPH for Deformable Solids. *Natural Phenomena*, 27–34.
- 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.
- Robert Bridson. 2015. *Fluid simulation for computer graphics*. CRC Press.
- Jumyung Chang, Fang Da, Eitan Grinspun, and Christopher Batty. 2019. A Unified Simplicial Model for Mixed-Dimensional and Non-Manifold Deformable Elastic Objects. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 11 (2019), 18 pages.
- Hsiao-yu Chen, Edith Tretschk, Tuur Stuyck, Petr Kadlec, Ladislav Kavan, Etienne Vouga, and Christoph Lassner. 2022. Virtual elastic objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15827–15837.
- Jieyu Chu, Nafees Bin Zafar, and Xubo Yang. 2017. A Schur Complement Preconditioner for Scalable Parallel Fluid Simulation. *ACM Trans. Graph.* 36, 5, Article 163 (2017).
- Pascal Clausen, Martin Wicke, Jonathan R. Shewchuk, and James F. O’Brien. 2013. Simulating Liquids and Solid-Liquid Interactions with Lagrangian Meshes. *ACM Trans. Graph.* 32, 2, Article 17 (2013), 15 pages.
- Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double Bubbles sans Toil and Trouble: Discrete Circulation-Preserving Vortex Sheets for Soap Films and Foams. *ACM Trans. Graph.* 34, 4, Article 149 (2015), 9 pages.
- Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-Only Liquids. *ACM Trans. Graph.* 35, 4, Article 78 (2016), 12 pages.
- Mathieu Desbrun and Marie-Paule Gascuel. 1996. Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96*. Springer-Verlag, 61–76.
- Mengyuan Ding, Xuchen Han, Stephanie Wang, Theodore F Gast, and Joseph M Teran. 2019. A thermomechanical material point method for baking and cooking. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 192.
- Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2021. DiffPD: Differentiable Projective Dynamics. *ACM Trans. Graph.* 41, 2, Article 13 (2021), 21 pages.

- A Cemal Eringen. 1992. Vistas of nonlocal continuum physics. *International journal of engineering science* 30, 10 (1992), 1551–1565.
- Yu Fang, Minchen Li, Ming Gao, and Chenfanfu Jiang. 2019. Silly rubber: an implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–13.
- Yun (Raymond) Fei, Christopher Batty, Eitan Grinspun, and Changxi Zheng. 2018. A Multi-Scale Model for Simulating Liquid-Fabric Interactions. *ACM Trans. Graph.* 37, 4, Article 51 (2018), 16 pages.
- Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, and Gabriel Peyré. 2019. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 2681–2690.
- Ben Frost, Alexey Stomakhin, and Hiroaki Narita. 2017. Moana: Performing Water. In *ACM SIGGRAPH 2017 Talks (SIGGRAPH '17)*. Article 30.
- Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. 2018a. Animating fluid sediment mixture in particle-laden flows. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.
- Ming Gao, Xinlei Wang, Kui Wu, Andre Pradhana, Eftychios Sifakis, Cem Yuksel, and Chenfanfu Jiang. 2018b. GPU optimization of material point methods. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12.
- Willi Geiger, Mohen Leo, Nick Rasmussen, Frank Losasso, and Ron Fedkiw. 2006. So Real It'll Make You Wet. In *ACM SIGGRAPH 2006 Sketches (SIGGRAPH '06)*.
- Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bäcker, Bernhard Thomaszewski, and Stelian Coros. 2020. ADD: Analytically Differentiable Dynamics for Multi-Body Systems with Frictional Contact. *ACM Trans. Graph.* 39, 6, Article 190 (2020), 15 pages.
- Christoph Gissler, Andreas Henne, Stefan Band, Andreas Peer, and Matthias Teschner. 2020. An Implicit Compressible SPH Solver for Snow Simulation. *ACM Trans. Graph.* 39, 4, Article 36 (2020), 16 pages.
- Chengguizi Han, Tao Xue, and Mridul Aanjaneya. 2021. A Lagrangian Particle-based Formulation for Coupled Simulation of Fracture and Diffusion in Thin Membranes. *Computer Graphics Forum* 40, 7 (2021), 97–108.
- Philipp Holl, Vladlen Koltun, and Nils Thuerey. 2020. Learning to Control PDEs with Differentiable Physics. *ICLR (2020)*.
- Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. 2020. DiffTaichi: Differentiable Programming for Physical Simulation. *ICLR (2020)*.
- Yuanming Hu, Xinxin Zhang, Ming Gao, and Chenfanfu Jiang. 2019. On Hybrid Lagrangian-Eulerian Simulation Methods: Practical Notes and High-Performance Aspects. In *ACM SIGGRAPH 2019 Courses*. Article 16, 246 pages.
- Zhao Huang, Yuanming Hu, Tao Du, Siyuan Zhou, Hao Su, Joshua B. Tenenbaum, and Chuang Gan. 2021. PlasticineLab: A Soft-Body Manipulation Benchmark with Differentiable Physics. *ArXiv abs/2104.03311 (2021)*.
- David Hutchins, Olun Riley, Jesse Erickson, Alexey Stomakhin, Ralf Habel, and Michael Kaschak. 2015. Big Hero 6: Into the Portal. In *ACM SIGGRAPH 2015 Talks (SIGGRAPH '15)*. Article 52.
- Markus Ihmsen, Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2012. Unified spray, foam and air bubbles for particle-based fluids. *The Visual Computer* 28 (2012), 669–677.
- Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. 2014. Implicit Incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics* 20 (2014), 426–435.
- Markus Ihmsen, Arthur Wahl, and Matthias Teschner. 2013. A Lagrangian Framework for Simulating Granular Material with High Detail. *Comput. Graph.* 37, 7 (2013), 800–808.
- 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 (jul 2015), 10 pages. <https://doi.org/10.1145/2766996>
- 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.
- Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-prager elastoplasticity for sand animation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.
- Egor Larionov, Christopher Batty, and Robert Bridson. 2017. Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11.
- Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable Muscle-actuated Human Simulation and Control. *ACM Trans. Graph.* 38, 4, Article 73 (2019), 13 pages.
- Seunghwan Lee, Ri Yu, Jungnam Park, Mridul Aanjaneya, Eftychios Sifakis, and Jehee Lee. 2018. Dexterous Manipulation and Control with Volumetric Muscles. *ACM Trans. Graph.* 37, 4, Article 57 (2018).
- David I. W. Levin, Joshua Litven, Garrett L. Jones, Shinjiro Sueda, and Dinesh K. Pai. 2011. Eulerian Solid Simulation with Contact. *ACM Trans. Graph.* 30, 4, Article 36 (2011), 10 pages.
- Duo Li, Shinjiro Sueda, Debanga R. Neog, and Dinesh K. Pai. 2013. Thin Skin Elastodynamics. *ACM Trans. Graph.* 32, 4, Article 49 (2013), 10 pages.
- Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. 2023. PAC-NeRF: Physics Augmented Continuum Neural Radiance Fields for Geometry-Agnostic System Identification.

- In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=tVkrbkz42vc>
- Haixiang Liu, Nathan Mitchell, Mridul Aanjaneya, and Eftychios Sifakis. 2016. A scalable Schur-complement fluids solver for heterogeneous compute platforms. *ACM Trans. Graph.* 35, 6, Article 201 (2016), 12 pages.
- Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2017. Quasi-Newton Methods for Real-Time Simulation of Hyperelastic Materials. *ACM Trans. Graph.* 36, 3, Article 23 (2017), 16 pages.
- Frank Losasso, Tamar Shinar, Andrew Selle, and Ronald Fedkiw. 2006. Multiple Interacting Liquids. *ACM Trans. Graph.* 25, 3 (2006), 812–819.
- Li-Ke Ma, Yizhong Zhang, Yang Liu, Kun Zhou, and Xin Tong. 2017. Computational Design and Fabrication of Soft Pneumatic Objects with Desired Deformations. *ACM Trans. Graph.* 36, 6, Article 239 (2017), 12 pages.
- Miles Macklin and Matthias Müller. 2013. Position Based Fluids. *ACM Trans. Graph.* 32, 4, Article 104 (2013), 12 pages.
- Miles Macklin, Matthias Müller, Nuttapon Chentanez, and Tae-Yong Kim. 2014. Unified Particle Physics for Real-Time Applications. *ACM Trans. Graph.* 33, 4, Article 153 (2014), 12 pages.
- Sebastian Martin, Peter Kaufmann, Mario Botsch, Eitan Grinspun, and Markus Gross. 2010. Unified Simulation of Elastic Rods, Shells, and Solids. *ACM Trans. Graph.* 29, 4, Article 39 (2010), 10 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 (2015), 14 pages.
- A. McAdams, E. Sifakis, and J. Teran. 2010. A Parallel Multigrid Poisson Solver for Fluids Simulation on Large Grids. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA ’10)*. 65–74.
- Nathan Mitchell, Mridul Aanjaneya, Rajsekhar Setaluri, and Eftychios Sifakis. 2015. Non-manifold Level Sets: A Multivalued Implicit Surface Representation with Applications to Self-collision Processing. *ACM Trans. Graph.* 34, 6, Article 247 (2015), 9 pages.
- Mojtaba Mozaffar and Jian Cao. 2021. Additive manufacturing process design with differentiable simulations. arXiv:2107.10919 [cs.CE]
- Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position Based Dynamics. *J. Vis. Comun. Image Represent.* 18, 2 (2007), 109–118.
- Kentaro Nagasawa, Takayuki Suzuki, Ryohei Seto, Masato Okada, and Yonghao Yue. 2019. Mixing Sauces: A Viscosity Blending Model for Shear Thinning Fluids. *ACM Trans. Graph.* 38, 4, Article 95 (2019), 17 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 (SCA ’16)*. 21–28.
- James F. O’Brien and Jessica K. Hodgins. 1999. Graphical Modeling and Animation of Brittle Fracture. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’99)*. 137–146.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- 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).
- Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming C. Lin. 2021. Efficient Differentiable Simulation of Articulated Bodies. *ICML* (2021).
- 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 (SCA ’04)*. 193–202.
- Karthik Raveendran, Chris Wojtan, and Greg Turk. 2011. Hybrid Smoothed Particle Hydrodynamics. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Vancouver, British Columbia, Canada) (SCA ’11)*. Association for Computing Machinery, New York, NY, USA, 33–42. <https://doi.org/10.1145/2019406.2019411>
- Prashant Sachdeva, Shinjiro Sueda, Susanne Bradley, Mikhail Fain, and Dinesh K. Pai. 2015. Biomechanical Simulation and Control of Hands and Tendinous Systems. *ACM Trans. Graph.* 34, 4, Article 42 (2015), 10 pages.
- 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.
- Eftychios Sifakis, Kevin G. Der, and Ronald Fedkiw. 2007a. Arbitrary Cutting of Deformable Tetrahedralized Objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 73–80.

- Eftychios Sifakis, Tamar Shinar, Geoffrey Irving, and Ronald Fedkiw. 2007b. Hybrid simulation of deformable solids. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 81–90.
- Juan C Simo and Thomas JR Hughes. 2006. *Computational inelasticity*. Vol. 7. Springer Science & Business Media.
- SL Sobolev. 2014. Nonlocal diffusion models: Application to rapid solidification of binary mixtures. *International Journal of Heat and Mass Transfer* 71 (2014), 295–302.
- Changkyu Song and Abdeslam Boularias. 2020a. Identifying mechanical models of unknown objects with differentiable physics simulations. In *Learning for Dynamics and Control*. PMLR, 749–760.
- Changkyu Song and Abdeslam Boularias. 2020b. Learning to Slide Unknown Objects with Differentiable Physics Simulations. In *Proceedings of Robotics: Science and Systems (RSS)*, Corvallis, Oregon, 2020.
- Sangeetha Grama Srinivasan, Qisi Wang, Junior Rojas, Gergely Klár, Ladislav Kavan, and Eftychios Sifakis. 2021. Learning Active Quasistatic Physics-Based Models from Data. *ACM Trans. Graph.* 40, 4, Article 129 (2021), 14 pages.
- Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.
- Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.
- 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 (2021).
- Haozhe Su, Tao Xue, Chengguizi Han, and Mridul Aanjaneya. 2022. A-ULMPM: An Adaptively Updated Lagrangian Material Point Method for Efficient Physics Simulation without Numerical Fracture. *Computer Graphics Forum* 41, 2 (2022), 325–341.
- Shinjiro Sueda, Garrett L. Jones, David I. W. Levin, and Dinesh K. Pai. 2011. Large-Scale Dynamic Simulation of Highly Constrained Strands. *ACM Trans. Graph.* 30, 4, Article 39 (2011), 10 pages.
- Robert W. Sumner, James F. O'Brien, and Jessica K. Hodgins. 1999. Animating Sand, Mud, and Snow. *Computer Graphics Forum* 18 (1999).
- Tetsuya Takahashi and Christopher Batty. 2020. Monolith: A Monolithic Pressure-Viscosity-Contact Solver for Strong Two-Way Rigid-Rigid Rigid-Fluid Coupling. *ACM Trans. Graph.* 39, 6, Article 182 (2020), 16 pages.
- Tetsuya Takahashi and Christopher Batty. 2021. Frictional Monolith: A Monolithic Optimization-Based Approach for Granular Flow with Contact-Aware Rigid-Body Coupling. *ACM Trans. Graph.* 40, 6, Article 206 (2021), 20 pages.
- Tetsuya Takahashi and Ming C. Lin. 2019a. A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling. *Computer Graphics Forum* 38 (2019).
- Tetsuya Takahashi and Ming C. Lin. 2019b. Video-Guided Real-to-Virtual Parameter Transfer for Viscous Fluids. *ACM Trans. Graph.* 38, 6, Article 237 (2019), 12 pages.
- Andre Pradhana Tampubolon, Theodore Gast, Gergely Klár, Chuyuan Fu, Joseph Teran, Chenfanfu Jiang, and Ken Museth. 2017. Multi-species simulation of porous sand and water mixtures. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11.
- Yun Teng, David I. W. Levin, and Theodore Kim. 2016. Eulerian Solid-Fluid Coupling. *ACM Trans. Graph.* 35, 6, Article 200 (2016), 8 pages.
- Joseph Teran, Eftychios Sifakis, Silvia S. Blemker, Victor Ng-Thow-Hing, Cynthia Lau, and Ronald Fedkiw. 2005. Creating and simulating skeletal muscle from the visible human data set. *IEEE Trans Vis Comput Graph* 11 (2005).
- Demetri Terzopoulos and Kurt Fleischer. 1988. Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. *SIGGRAPH Comput. Graph.* 22, 4 (1988), 269–278.
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. *SIGGRAPH Comput. Graph.* 21, 4 (1987), 205–214.
- Nils Thürey, Chris Wojtan, Markus Gross, and Greg Turk. 2010. A Multiscale Approach to Mesh-Based Surface Tension Flows. In *ACM SIGGRAPH 2010 Papers*. Article 48.
- Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2021. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12959–12970.
- Nghia Truong, Cem Yuksel, Chakrit Watcharopas, Joshua A. Levine, and Robert M. Kirby. 2021. Particle Merging-and-Splitting. *IEEE Transactions on Visualization and Computer Graphics* (2021), 12 pages.
- Kun Wang, Mridul Aanjaneya, and Kostas Bekris. 2020. A First Principles Approach for Data-Efficient System Identification of Spring-Rod Systems via Differentiable Physics Engines. *Proceedings of Machine Learning Research* 120 (2020), 1–15.
- Kun Wang, Mridul Aanjaneya, and Kostas Bekris. 2021. Sim2Sim Evaluation of a Novel Data-Efficient Differentiable Physics Engine for Tensegrity Robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1694–1701.
- Nicholas J. Weidner, Kyle Piddington, David I. W. Levin, and Shinjiro Sueda. 2018. Eulerian-on-Lagrangian Cloth Simulation. *ACM Trans. Graph.* 37, 4, Article 50 (2018), 11 pages.

- Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2010. Physics-Inspired Topology Changes for Thin Fluid Features. *ACM Trans. Graph.* 29, 4, Article 50 (2010), 8 pages.
- Chris Wojtan and Greg Turk. 2008. Fast Viscoelastic Behavior with Thin Features. *ACM Trans. Graph.* 27, 3 (2008), 1–8.
- Joshuah Wolper, Yu Fang, Minchen Li, Jiecong Lu, Ming Gao, and Chenfanfu Jiang. 2019. CD-MPM: continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.
- Tao Xue, Haozhe Su, Chengguizi Han, Chenfanfu Jiang, and Mridul Aanjaneya. 2020. A novel discretization and numerical solver for non-fourier diffusion. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.
- Xiao Yan, Yun-Tao Jiang, Chen-Feng Li, Ralph R. Martin, and Shi-Min Hu. 2016. Multiphase SPH Simulation for Interactive Fluids and Solids. *ACM Trans. Graph.* 35, 4, Article 79 (2016), 11 pages.
- Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2015. Continuum foam: A material point method for shear-dependent flows. *ACM Transactions on Graphics (TOG)* 34, 5 (2015), 1–20.
- Xinxin Zhang and Robert Bridson. 2014. A PPPM Fast Summation Method for Fluids and Beyond. *ACM Trans. Graph.* 33, 6, Article 206 (2014), 11 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.
- Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional Surface Tension Flow on Simplicial Complexes. *ACM Trans. Graph.* 33, 4, Article 111 (2014), 11 pages.
- Yongning Zhu and Robert Bridson. 2005. Animating Sand as a Fluid. *ACM Trans. Graph.* 24, 3 (2005), 965–972.
- Yongning Zhu, Eftychios Sifakis, Joseph Teran, and Achi Brandt. 2010. An Efficient Multigrid Method for the Simulation of High-resolution Elastic Solids. *ACM Trans. Graph.* 29, 2, Article 16 (2010), 18 pages.