

A Unified Second-Order Accurate in Time MPM Formulation for Simulating Viscoelastic Liquids with Phase Change

HAOZHE SU*, Rutgers University

TAO XUE*, Rutgers University

CHENGGUIZI HAN, Rutgers University

CHENFANFU JIANG, University of Pennsylvania

MRIDUL AANJANEYA, Rutgers University

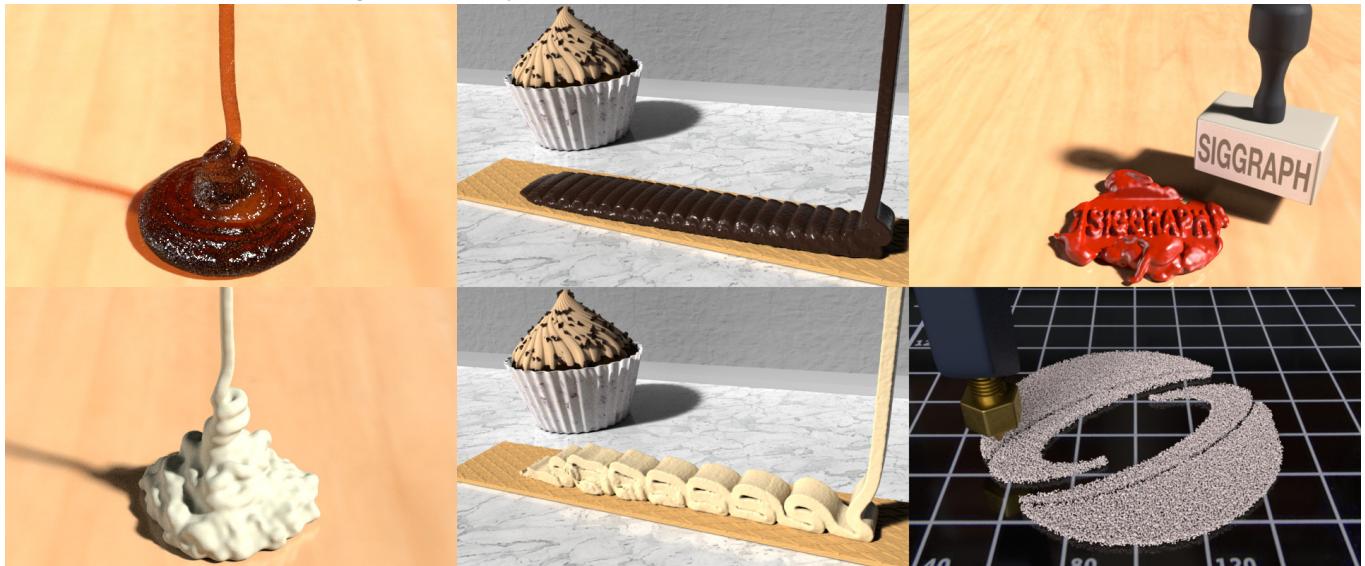


Fig. 1. Our unified constitutive model can simulate both Newtonian and non-Newtonian viscous liquids. (Left) Liquid rope coiling with Newtonian viscosity (top), and asymmetric buckling for non-Newtonian shaving cream (bottom). (Middle) Symmetric buckling with Newtonian chocolate (top), and buckling with shear-thickening for non-Newtonian cream (bottom). (Right) Object interaction with Newtonian ketchup bunnies (top), and 3D printing of shapes (bottom).

We assume that the viscous forces in any liquid are simultaneously *local* and *non-local*, and introduce the *extended POM-POM model* [McLeish and Larson 1998; Oishi et al. 2012; Verbeeten et al. 2001] to computer graphics to design a unified constitutive model for viscosity that generalizes prior models, such as Oldroyd-B, the Upper-conveyed Maxwell (UCM) model [Sadeghy et al. 2005], and classical Newtonian viscosity under one umbrella, recovering each of them with different parameter values. Implicit discretization of our model via backward Euler recovers the variational Stokes solver of [Larionov et al. 2017] for Newtonian viscosity. For greater accuracy, however, we introduce the second-order accurate Generalized Single Step Single Solve (GS4) scheme [Tamma et al. 2000; Zhou and Tamma 2004] to

*joint first authors

Authors' addresses: Haozhe Su, Rutgers University; Tao Xue, Rutgers University; Chengguizi Han, Rutgers University; Chenfanfu Jiang, University of Pennsylvania; Mridul Aanjaneya, Rutgers University.

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.

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2021/8-ART119 \$15.00
<https://doi.org/10.1145/3450626.3459820>

computer graphics, which recovers all prior second-order accurate time integration schemes to date. Using GS4 and our generalized constitutive model, we present a Material Point Method (MPM) for simulating various viscoelastic liquid behaviors, such as classical liquid rope coiling, buckling, folding, and shear thinning/thickening. In addition, we show how to couple our viscoelastic liquid simulator with the recently introduced non-Fourier heat diffusion solver [Xue et al. 2020] for simulating problems with phase change, such as melting chocolate and digital fabrication with 3D printing. While the discretization of heat diffusion is slightly different within GS4, we show that it can still be efficiently solved using an assembly-free Multigrid-preconditioned Conjugate Gradients solver. We present end-to-end 3D simulations to demonstrate the versatility of our framework.

CCS Concepts: • Computing methodologies → Physical simulation.

Additional Key Words and Phrases: viscoelastic, non-local, POM-POM, MPM, phase change, buckling, coiling, shear thinning/thickening

ACM Reference Format:

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 (August 2021), 18 pages. <https://doi.org/10.1145/3450626.3459820>



Fig. 2. Our unified constitutive model can generate the classical liquid rope coiling effect (top) for Newtonian viscosity [Larionov et al. 2017], and also asymmetric coiling and elastic effects for non-Newtonian liquids (middle and bottom) such as Oldroyd-B [Ram et al. 2015] and Pom-Pom [Bishko et al. 1997]. A columnar structure is first generated by high-frequency coiling, which collapses at a certain point and the process repeats - showcasing *secondary coiling*. The average Δt during simulations for each case is $\Delta t = 10^{-5}$ s.

1 INTRODUCTION

Viscous behaviors are exhibited by a wide range of everyday materials [Krishnan et al. 2010], such as oil, honey, mud, molasses, manufacturing materials such as melting polymer in 3D printers [Tymrak et al. 2014], many foods and additives [Ding et al. 2019], and bodily fluids (e.g., blood) [Müller et al. 2004]. Thus, there is a strong demand for formulating a generalized constitutive model that can capture various viscous behaviors, such as shear thinning/thickening, and developing tools that can reproduce these behaviors for animation and visual effects, such as spilling honey and melting/solidifying polymer. Proper constitutive modeling for viscosity requires accurate quantification of the relationship between material elongation and shear stresses. Classical Newtonian models [Batchelor and Batchelor 2000] undergo strain rates that are linearly proportional to the applied shear stress. In contrast, non-Newtonian models [Larson 1999] do not follow this restriction and have more variants.

In this paper, we revisit the relationship between shear stress and strain from the viewpoint of *non-local modeling* [Eringen 1992; Sobolev 2014] and develop a generalized constitutive model for viscosity with memory effects (i.e., temporal non-locality) that captures some of the most characteristic rheological fluid behaviors. Our approach draws inspiration from the non-Fourier diffusion model of [Xue et al. 2020], since we assume that the viscous forces in any fluid are simultaneously *local* and *non-local*. We introduce

the *extended POM-POM model* [McLeish and Larson 1998; Oishi et al. 2012; Verbeeten et al. 2001] to computer graphics to formulate the time scale of this (first order) non-local viscosity. The resulting constitutive model for viscosity generalizes previous models, such as Oldroyd-B [Oldroyd 1950], the Upper-convected Maxwell (UCM) model [Sadeghy et al. 2005], and classical Newtonian viscosity.

Numerical discretization of viscosity is challenging because of strongly non-linear terms present in both the viscous stress and its coupling with other physics, such as rigid bodies [Aanjaneya et al. 2019; Takahashi and Batty 2020; Takahashi and Lin 2019], free surface pressure [Larionov et al. 2017], or temperature [Goldade et al. 2019]. We first show that our method combined with a backward Euler discretization on staggered MAC grids recovers the variational Stokes solver in [Larionov et al. 2017] for Newtonian viscosity. For greater accuracy, however, we introduce the unified, second order accurate, time integration scheme *Generalized Single Step Single Solve (GS4)* [Tamma et al. 2000; Zhou and Tamma 2004] to computer graphics, which recovers all prior second order accurate time integration schemes to date, such as Crank-Nicholson, Newmark family, generalized- α , etc., and has controllable numerical dissipation to smooth high frequency numerical oscillations. We formulate an implicit discretization with GS4 that requires a *single* linear solve and is computationally equivalent to a first order backward Euler

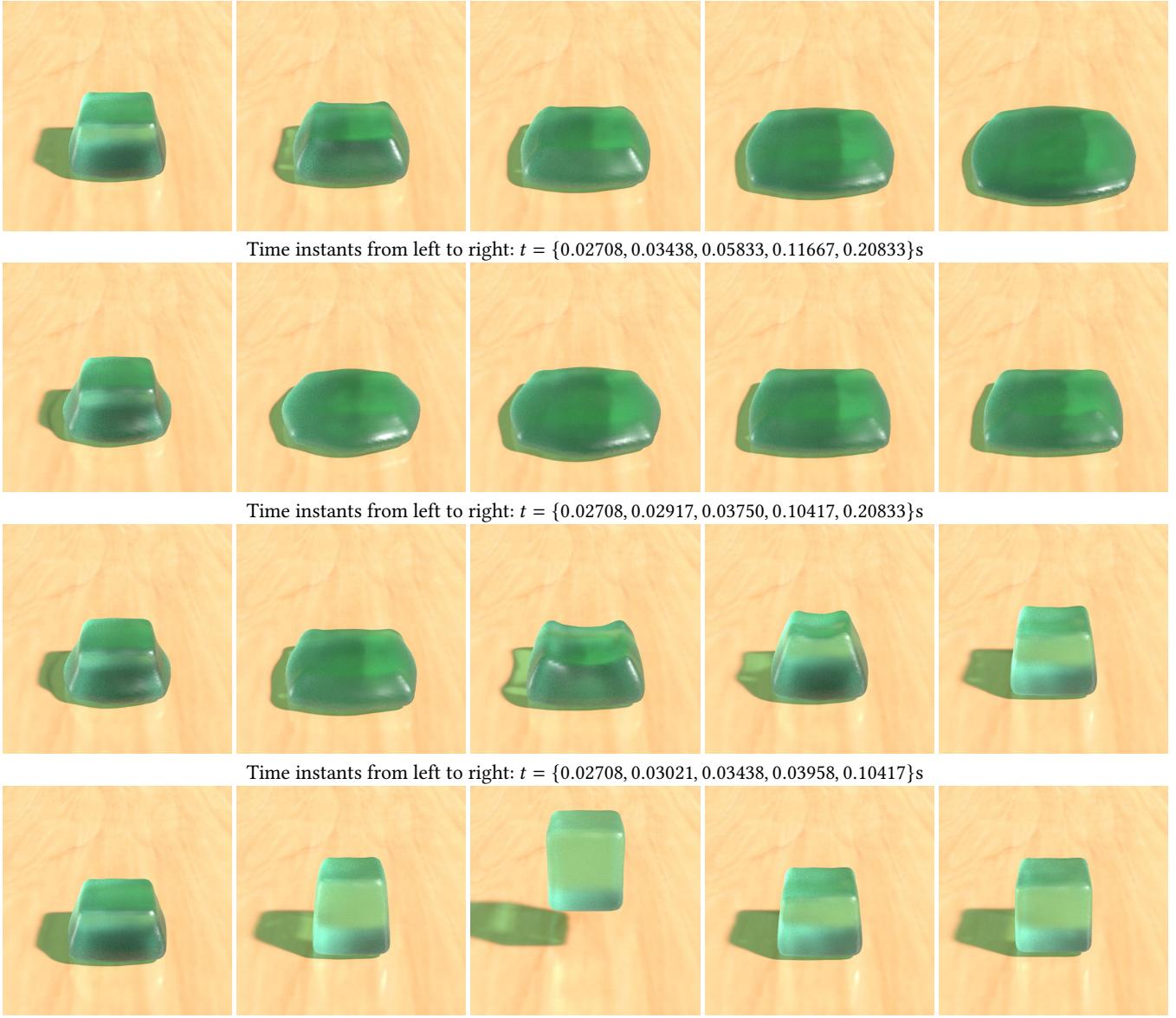


Fig. 3. Our unified constitutive model for viscosity based on non-local modeling can simulate Newtonian and non-Newtonian viscous liquids with different parameter values. A Newtonian cube ($F_v = 1$) spreads on the floor when dropped (row 1), while non-Newtonian cubes ($F_v = 0.01$) rebound due to shear-thickening viscous forces. As the Weissenberg number W_e decreases, the solid-like behavior of the viscous cube becomes more significant. Rows 2-4: $W_e = \{10, 1, 0.1\}$. The average Δt during simulation is 5×10^{-4} s for each case with an impact speed 50m/s. Specific parameters are given in Table 4(rows 1-4).

discretization, while providing the benefits of second order accuracy. The primary unknowns in our implicit non-Newtonian system include velocity, pressure, the local (Newtonian) portion in the total viscous stress, and the additional non-local (non-Newtonian) portion in the total viscous stress, which are fully coupled together. This makes it challenging to design an effective preconditioner for iterative Krylov solvers, such as preconditioned Conjugate Gradients. Thus, we also show how GS4 can be used to derive an explicit discretization that is straightforward to parallelize, and is stable for the high resolution 3D examples that we demonstrate in Section 7. Furthermore, we show how GS4 and our unified constitutive model

can be integrated with the Material Point Method (MPM) to provide a computational framework for simulating various viscous liquid behaviors, such as classical liquid rope coiling, buckling, folding, and shear thinning/thickening.

Finally, we show how our method can be coupled with heat transfer to simulate viscoelastic materials undergoing phase change, such as melting chocolate, or additive manufacturing of polymer during 3D printing. For this purpose, we employ the non-Fourier diffusion solver of [Xue et al. 2020], as its theoretical underpinnings are also based on non-local modeling, similar to our present work. Our numerical discretization is slightly different than [Xue et al.



Fig. 4. Our method captures rich interactions and surface details as eight bunnies made of Newtonian ketchup are dropped one by one on top of each other. The average Δt during simulation is 5×10^{-6} s with an impact speed of 30.7m/s .

2020], as it is derived within the framework of GS4, providing second order accuracy in time (in contrast to [Xue et al. 2020] which is only first order accurate in time). However, we show that our method still leads to a Poisson-style elliptic system for the temperature that can be efficiently solved using an assembly-free Multigrid-preconditioned Conjugate Gradients solver.

To summarize, our main contributions are the following:

- (1) introduction of *non-local modeling* for generalized rheology to computer graphics and development of a unified constitutive model for Newtonian and non-Newtonian viscous liquids,
- (2) a unified numerical discretization for rheology that recovers the variational Stokes solver of [Larionov et al. 2017] for Newtonian viscosity, and a variety of non-Newtonian models, such as the extended Pom-Pom model [McLeish and Larson 1998; Oishi et al. 2012, 2011; Verbeeten et al. 2001], Oldroyd-B, and the Upper Convective Maxwell (UCM) model,
- (3) introduction of the unified second order accurate *Generalized Single Step Single Solve* (GS4) [Tamma et al. 2000; Zhou and Tamma 2004] time integration scheme to computer graphics, and its use in implicit/explicit discretizations of our unified constitutive model with the Material Point Method (MPM),
- (4) integration of our method with the non-Fourier diffusion solver of [Xue et al. 2020] within the second-order GS4 framework for coupled simulation of phase change problems, and
- (5) end-to-end 3D simulations of classical liquid rope coiling, buckling, droplet impact, shear thinning/thickening, and polymer melting/solidification during 3D printing, demonstrating the various characteristics of general viscoelastic liquids.

2 RELATED WORK

Since the pioneering work of Terzopoulos and Fleischer [1988], the simulation of rate-dependent physics has always attracted attention from computer graphics researchers. Representative behaviors include viscosity, viscoelasticity, and viscoplasticity. Viscosity is most frequently incorporated for fluids [Bridson 2015], but sometimes it is also used as a damping model for solids to stabilize elastic oscillations [Brown et al. 2018; O'Brien and Hodges 1999]. Efficient and effective discretization schemes of Newtonian viscous fluids have been explored with particles through Smoothed Particle Hydrodynamics (SPH) [Band et al. 2018; Bender and Koschier 2017; Bender et al. 2020; Peer et al. 2015; Takahashi et al. 2015; Weiler et al. 2018], constrained dynamics [Barreiro et al. 2017] or the Material Point Method (MPM) [Ram et al. 2015], finite difference methods [Batty

and Bridson 2008; Goldade et al. 2019; Larionov et al. 2017], finite volume methods [Batty and Houston 2011], and Lagrangian meshes [Batty et al. 2012; Bergou et al. 2010]. Fluids become non-Newtonian when viscosity depends on strain rate, exhibiting either shear thinning or shear thickening. Such complex fluids have been simulated with grids [Goktekin et al. 2004], particles [Paiva et al. 2009; Yue et al. 2015] and meshes [Wojtan and Turk 2008; Zhu et al. 2015].

2.1 Phase Change Phenomena

Heat transport for simulating melting and solidification effects has been popular in computer graphics since the early work of [Carlson et al. 2002; Stora et al. 1999]. Since then, impressive animations have been generated for thermoelastic solid-liquid phase change with dynamically changing Lagrangian meshes [Clausen et al. 2013], particle methods [Iwasaki et al. 2010], and hybrid Lagrangian-Eulerian schemes such as MPM [Ding et al. 2019; Gao et al. 2018; Stomakhin et al. 2014]. All these prior works employ Fourier's law, where the heat flow depends linearly on the temperature gradient, and use the coupling between heat transfer and material transport, where the material properties change according to the temperature. In contrast, our work builds upon the generalized diffusion solver of [Xue et al. 2020] and is coupled to the temperature, providing richer simulations for phase change phenomena, such as polymer melting and solidification during additive manufacturing.

2.2 Comparison to Prior Art

Prior works on MPM simulations of viscous materials are based on *finite strain* solid mechanics, using constitutive models that relate the deformation strains to the viscous forces [Fang et al. 2019; Ram et al. 2015; Yue et al. 2015]. An Oldroyd-B visco-plastic model was proposed in [Ram et al. 2015] to simulate foams and sponges, while a similar non-Newtonian *Herschel-Bulkley* fluid model was used to formulate a visco-plastic model for dense foams in [Yue et al. 2015]. A finite strain integration scheme for general viscoelastic solids and non-equilibrated flow was proposed for MPM in [Fang et al. 2019]. In general, these formulations rely on the discretization of the upper-convected derivative of the right Cauchy Green strain [Bonet and Wood 1997] and require considerable effort to integrate them with existing MPM schemes that evolve the deformation gradient tensor [Stomakhin et al. 2013]. The implementation of MPM for multi-physics coupling problems, such as phase change [Stomakhin et al. 2014] and baking [Ding et al. 2019], has also been investigated in this finite stain framework. In contrast, our constitutive model directly provides the expression of the Cauchy stress for viscosity



Fig. 5. A non-Newtonian bunny bounces back instead of collapsing when dropped on the ground, displaying the *shear-thickening effect* of viscoelastic liquids. The average Δt during simulation is 5×10^{-7} s for each case with an impact speed of 29 m/s.

and alleviates the need for updating/evaluating the deformation gradient tensor as well as the right Cauchy Green strain. Our GS4-MPM discretization utilizes standard MPM for communicating material information between particles and grids and for approximating general *differential operators*, such as divergence and gradient, existing both in the governing equations of motion and the diffusion model. Thus, all coupled equations are discretized in a consistent manner.

3 OVERVIEW OF NON-LOCAL MODELING

In recent decades, there has been a significant upswing on the use of *non-local* theories for the modeling of material behaviors [Shaft et al. 2020; Srinivasa and Reddy 2017]. Such theories were of interest since the earliest days of finite deformation continuum mechanics [Eringen 1992; Eringen and Edelen 1972; Kröner 1967], but have not received much attention from the computer graphics community. In fact, some of the most attractive material behaviors, which are of great interest to computer graphics, such as shear thinning/thickening in viscous fluids, viscoelasticity in solid mechanics [Fang et al. 2019], and non-Fourier diffusion [Xue et al. 2020], demonstrate non-local effects in both space and time, and they can be recast within the unified framework of non-local theory, instead of individually modeling each one of them (see Section 4). In general, classical (local) models for a certain space-time physical field consist of the first gradients of primary variables and its time derivatives. For example, classical continuum mechanics relies on the first gradient of displacement (deformation gradient) and time derivatives of displacement (acceleration and velocity). In the context of non-local theory, higher order spatial gradients (*spatial non-locality*) and higher order time derivatives (*temporal non-locality*) are included to describe the material behaviors at different scales of space and time, including viscoelasticity, composite mixtures, and dislocations. See Section 4.1 for an illustration of how the non-local theory unifies the classical model with new models. In Table 1, we compare some of the well-known constitutive models in continuum mechanics and diffusion in terms of their non-locality.

Our motivation for choosing non-local modeling is that it provides us a wide range of design variables, specifically for viscous behaviors in fluids. Our constitutive model for viscous fluids emanating from this concept is shown to recover the classical Newtonian viscous model (local) and the 1st-order temporally non-local models,

Table 1. **Comparison between existing models.** NL: non-locality, ϕ : primary state variables that refer to displacement and temperature in continuum mechanics and diffusion; $\nabla\phi$: the first order spatial derivative of ϕ which drives the physics to evolve, such as the deformation gradient and heat flux. GE: Green's elasticity [Ogden 1997]; MVE: Maxwell viscoelasticity [Flügge 2013]; NV: Newtonian viscosity [Batchelor and Batchelor 2000]; OB: Oldroyd-B viscosity [Oldroyd 1950]; UCM: the Upper-convected Maxwell model [Sadeghy et al. 2005]; F: classical Fourier diffusion [Fick 1855; Fourier 1878]; C: Cattaneo diffusion [Cattaneo 1948]; J: Jeffery's diffusion [Joseph and Preziosi 1989]; and C-F: C-F diffusion [Xue et al. 2020].

	GE	MVE	NV	OB	UCM	F	C	J	C-F
0th NL ϕ	✓	✓	✓	✓	✓	✓	✓	✓	✓
0th NL $\nabla\phi$	✓	✓	✓	✓	✓	✓	✓	✓	✓
1st NL $\dot{\phi}$	✗	✗	✗	✓	✗	✗	✗	✓	✓
1st NL $\nabla\dot{\phi}$	✗	✗	✗	✓	✓	✗	✓	✓	✓

including Oldroyd-B and UCM. Our methodology of non-local modeling paves the way for achieving various kinds of viscous liquid behaviors, while being straightforward to implement.

4 GOVERNING EQUATIONS

In this section, we first introduce the concept of non-local modeling using diffusion as an example, and then derive our unified constitutive model for thermo-viscosity using this formulation.

4.1 Unified Non-local Constitutive Model

Non-local modeling is known to be more general than the traditional local theory [Eringen 1992; Tzou 2014], as described below.

4.1.1 Non-local Methodology. Let $\phi(\mathbf{x}, t)$ be a sufficiently smooth scalar field depending only on the local space-time coordinates \mathbf{x} and t . By introducing the characteristic scales of length (λ) and time (τ), a non-local definition for ϕ can be given as $\phi(\mathbf{x} + \lambda, t + \tau)$. If the position \mathbf{x} can vary continuously, then the expression for ϕ can be expanded using the Taylor series. For the sake of simplicity, we show the Taylor expansion assuming both \mathbf{x} and λ are scalar:

$$\phi(\mathbf{x} + \lambda, t + \tau) = \phi(\mathbf{x}, t) + \lambda \frac{\partial \phi}{\partial \mathbf{x}} + \tau \frac{\partial \phi}{\partial t} + \lambda^2 \frac{\partial^2 \phi}{\partial \mathbf{x}^2} + \lambda \tau \frac{\partial^2 \phi}{\partial \mathbf{x} \partial t} + \tau^2 \frac{\partial^2 \phi}{\partial t^2} + \dots \quad (1)$$

By doing this, non-locality can be classified into the following three kinds:

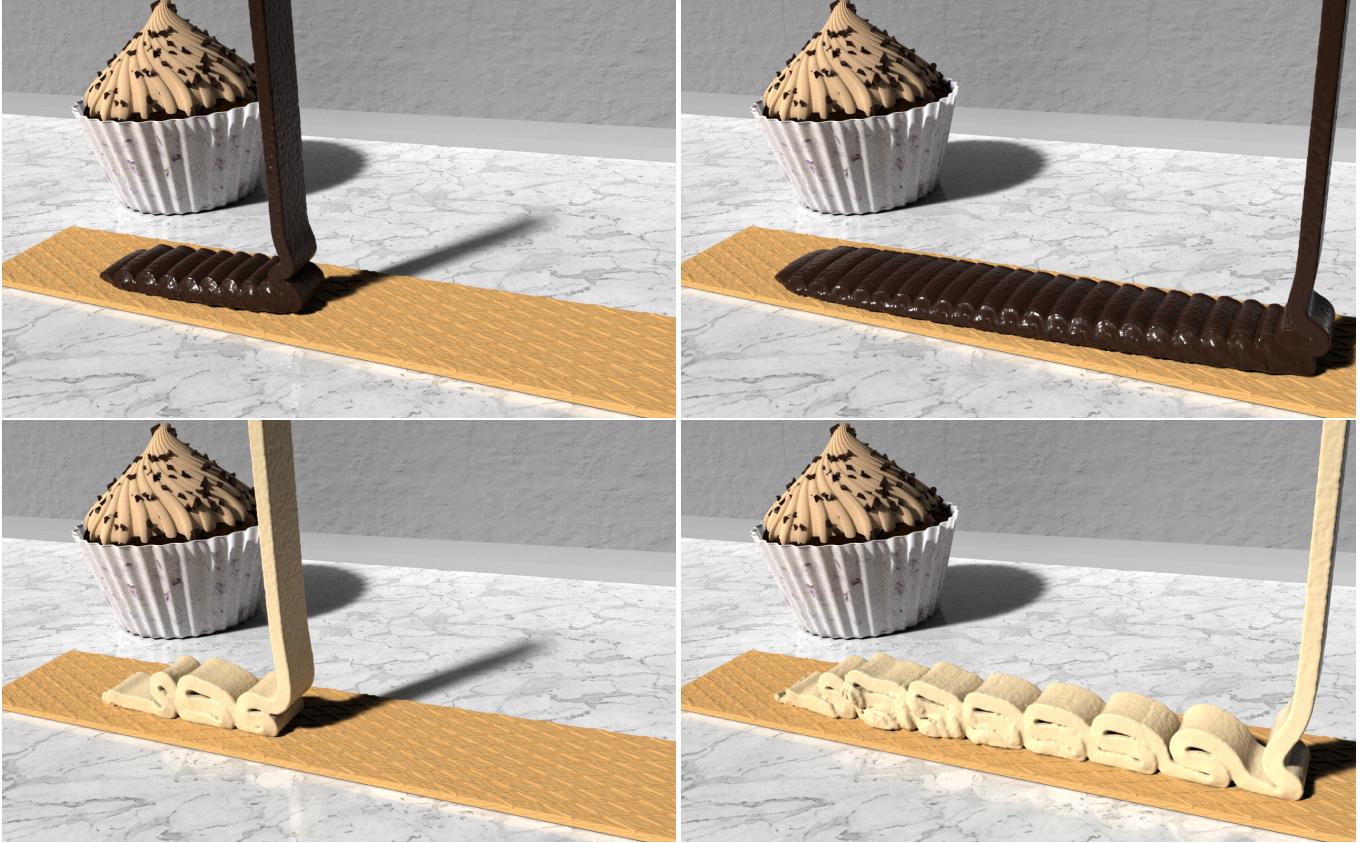


Fig. 6. Our method captures symmetric buckling patterns of Newtonian chocolate (top) and buckling with shear thickening for non-Newtonian cream (bottom). The average Δt during simulation is 10^{-4} s for Newtonian chocolate and 10^{-5} s for non-Newtonian cream.

- (1) **0-th order non-locality:** Only the 0-th order of λ and τ are considered, providing the traditional local model. That is:

$$\lim_{\lambda, \tau \rightarrow 0} \phi(x + \lambda, t + \tau) = \phi(x, t)$$

- (2) **1-st order non-locality:** The characteristic scale is the most significant one to demonstrate non-locality. Thus, we only focus on the following 1-st order non-local expression:

$$\phi(x + \lambda, t + \tau) \approx \phi(x, t) + \lambda \frac{\partial \phi}{\partial x} + \tau \frac{\partial \phi}{\partial t}$$

- (3) **higher order non-locality:** The characteristic scale is the n -th power of λ, τ , where n is the non-local order. As n increases, the characteristic scales become smaller, as $\lambda, \tau \ll 1$, i.e.,

$$\lambda > \lambda^2 > \dots > \lambda^n, \quad \tau > \tau^2 > \dots > \tau^n$$

Non-locality in space plays a significant role when the characteristic length scale λ is roughly at the same (or smaller) scale as the atomic mean free path length. Since our focus is on macroscopic simulations for computer graphics applications, we ignore non-locality in space, and only consider the following 1-st order non-locality in time:

$$\phi(x, t + \tau) \approx \phi(x, t) + \tau \frac{\partial \phi}{\partial t}(x, t) \quad (2)$$

For spatial non-locality, one can refer to the diffusive Johnson-Segalman model [Lu et al. 2000] for effects such as *shear banding*.

4.2 Illustration of Non-Local Diffusion Modeling

To understand equation (2) and its implementation, we briefly show the non-local modeling for diffusion proposed in [Tzou 2014]. Consider the diffusion problem in its 0-th and 1-st order non-local forms:

$$\begin{aligned} \frac{DT}{Dt} + \nabla \cdot \mathbf{q}(\mathbf{x}, t + \tau_q) &= Q \\ \mathbf{q}(\mathbf{x}, t + \tau_q) &= -k \nabla T(\mathbf{x}, t + \tau_T) \end{aligned} \quad (3)$$

where k and Q represent the diffusive coefficient and external source/sink, τ_q and τ_T represent the characteristic time scale with respect to the heat flux \mathbf{q} and temperature T . One can treat $-k \nabla T$ as the total energy driven by the difference of temperature such that:

- (1) 0-th order flux and temperature ($\tau_q = \tau_T = 0$):

$$\mathbf{q}(\mathbf{x}, t) = -k \nabla T(\mathbf{x}, t)$$

This choice recovers classical Fourier diffusion [Fick 1855; Fourier 1878], which is a local model in both space and time.

- (2) 1-st order flux ($\tau_q \neq 0, \tau_T = 0$):

$$\mathbf{q}(\mathbf{x}, t) + \tau_q \frac{\partial \mathbf{q}(\mathbf{x}, t)}{\partial t} = -k \nabla T(\mathbf{x}, t) \quad (4)$$

Ignoring the non-locality in temperature (i.e., $\lambda_T = 0$), gives the Cattaneo diffusion model [Cattaneo 1948].

(3) 1-st order flux and temperature ($\tau_q \neq 0, \tau_T \neq 0$):

$$\mathbf{q}(\mathbf{x}, t) + \tau_q \frac{\partial \mathbf{q}(\mathbf{x}, t)}{\partial t} = -k \left[\nabla T(\mathbf{x}, t) + \tau_T \frac{\partial \nabla T(\mathbf{x}, t)}{\partial t} \right] \quad (5)$$

If we consider $\lambda \neq 0, \tau \neq 0$, with simple mathematical manipulations, one can derive Jeffery's diffusion model [Joseph and Preziosi 1989] and the C-F diffusion model [Xue et al. 2020].

4.3 Non-local Modeling for Viscosity

We now present how the non-local concept unifies the existing viscous models for recovering both Newtonian flow (local model) and non-Newtonian flow (non-local model), such as Oldroyd-B and UCM. To the best of our knowledge, our derivations below are novel and point out the ability of non-local modeling to unify different constitutive models for viscosity. The total viscosity force emanates from the shear deformation tensor $\mathbf{D} = (\nabla \mathbf{u}^T + \nabla \mathbf{u}) / 2$, where \mathbf{u} denotes the fluid velocity, and the temporally non-local viscous shear stress σ_v can be expressed as follows:

$$\sigma_v(\mathbf{x}, t + \tau_\sigma) = 2\mu \mathbf{D}(\mathbf{x}, t + \tau_D), \quad (6)$$

where τ_σ, τ_D represent the characteristic time scales of shear stress and shear stain, and μ is the coefficient of viscosity. The following three cases arise depending on the values of τ_σ and τ_D :

(1) $\tau_\sigma = 0$ and $\tau_D = 0$ (local shear stress):

$$\sigma_v(\mathbf{x}, t) = 2\mu \mathbf{D}(\mathbf{x}, t).$$

This choice recovers the standard Newtonian viscous stress.

(2) $\tau_\sigma \neq 0$ and $\tau_D = 0$ (UCM):

This choice directly leads to the upper-conveyed Maxwell (UCM) model [Sadeghy et al. 2005]. Note that for accommodating large deformations, the time derivative should be replaced by the *upper-conveyed* time derivative. Specifically:

$$\begin{aligned} \frac{\partial \sigma_v}{\partial t} \rightarrow \frac{\hat{D}\sigma_v}{Dt} &= \frac{\partial \sigma_v}{\partial t} + \mathbf{u} \cdot \nabla \sigma_v - (\nabla \mathbf{u})^T \cdot \sigma_v - \sigma_v \cdot (\nabla \mathbf{u}), \\ \tau_\sigma \frac{\hat{D}\sigma_v}{Dt} + \sigma_v &= 2\mu \mathbf{D}(\mathbf{x}, t). \end{aligned} \quad (7)$$

(3) $\tau_\sigma \neq 0$ and $\tau_D \neq 0$ (1-st order non-local shear stress):

$$\tau_\sigma \frac{\hat{D}\sigma_v(\mathbf{x}, t)}{Dt} + \sigma_v = 2\mu \left[\mathbf{D}(\mathbf{x}, t) + \tau_D \frac{\hat{D}\mathbf{D}(\mathbf{x}, t)}{Dt} \right]. \quad (8)$$

Equation (8) provides the definition for the Oldroyd-B viscous model (see equation (59) in [Oldroyd 1950]), and it can also be seen as a first-order expansion including elastic effects in colloidal suspensions [Fröhlich and Sack 1946]. By assuming both local and 1-st order temporally non-local terms to exist concurrently, $\sigma_v = \sigma_v^L + \sigma_v^{NL}$, $\mu = \mu_L + \mu_{NL}$, and $\tau_D = \mu_L \tau_\sigma$, $\tau_\sigma = \tau$, equation (8) can be rewritten as follows:

$$\begin{aligned} \sigma_v(\mathbf{x}, t) &= \sigma_v^L(\mathbf{x}, t) + \sigma_v^{NL}(\mathbf{x}, t), \\ \sigma_v^L(\mathbf{x}, t) &= 2\mu_L \mathbf{D}(\mathbf{x}, t), \\ \sigma_v^{NL}(\mathbf{x}, t) + \tau \frac{\hat{D}\sigma_v^{NL}(\mathbf{x}, t)}{Dt} &= 2\mu_{NL} \mathbf{D}(\mathbf{x}, t). \end{aligned} \quad (9)$$

Note that the time scale τ is independent of the shear stress σ_v^{NL} , and equation (9) is a linear constitutive model [Fang

et al. 2019] that can only be used to describe the shear thinning process, as shown in [Nagasaki et al. 2019].

4.4 Unified Constitutive Model for Viscosity

To further enhance the non-local viscous model in equation (8) and its degenerated equation (9), we introduce a *time scale tensor* $\tau_v(\sigma_v^{NL})$ following [Verbeeten et al. 2001], which is a function of the non-local viscous shear stress σ_v^{NL} and which allows us to investigate the nonlinear temporal non-locality, as shown below:

$$\begin{aligned} \sigma_v(\mathbf{x}, t) &= \sigma_v^L(\mathbf{x}, t) + \sigma_v^{NL}(\mathbf{x}, t), \quad \sigma_v^L(\mathbf{x}, t) = 2F_v\mu\mathbf{D}(\mathbf{x}, t), \\ \sigma_v^{NL}(\mathbf{x}, t) + \tau_v(\sigma_v^{NL}) \frac{\hat{D}\sigma_v^{NL}(\mathbf{x}, t)}{Dt} &= 2(1 - F_v)\mu\mathbf{D}(\mathbf{x}, t), \end{aligned} \quad (10)$$

where $F_v \in [0, 1]$ is a user-controlled parameter to adjust the amount of locality and non-locality in the total viscous shear stress σ_v . To investigate the non-linear shear behavior, we introduce the polymer material properties [McLeish and Larson 1998; Verbeeten et al. 2001], and define the time scale tensor $\tau_v(\sigma_v^{NL})$ as follows:

$$\begin{aligned} \tau_v &= \lambda_b \left\{ \frac{\alpha}{G_0} \sigma_v^{NL} + f(\sigma_v^{NL})^{-1} \mathbf{I} + G_0 \left[f(\sigma_v^{NL})^{-1} \right] (\sigma_v^{NL})^{-1} \right\}^{-1} \\ \frac{1}{\lambda_b} f(\sigma_v^{NL})^{-1} &= \frac{2}{\lambda_s} \left(1 - \frac{1}{\Lambda} \right) + \frac{1}{\lambda_b \Lambda^2} \left(1 - \frac{\alpha I_{\sigma\sigma}}{3G_0^2} \right), \\ \Lambda &= \sqrt{1 + \frac{I_\sigma}{3G_0}}, \quad \lambda_s = \lambda_{0s} e^{-v(\Lambda-1)}, \quad v = \frac{2}{Q}, \\ I_\sigma &= \text{trace}(\sigma_v^{NL}), \quad I_{\sigma\sigma} = \text{trace}(\sigma_v^{NL} \cdot \sigma_v^{NL}), \end{aligned} \quad (11)$$

where $\alpha \leq 0$ is the anisotropy, Λ and λ_b, λ_{0s} are the backbone tube stretch and the relaxation times of backbone tube orientation and stretch used in long-chain branched polymers [McLeish and Larson 1998], v is a user-defined parameter, $v = 2/Q$, where Q is the amount of arms at the end of a backbone, and G_0 is the plateau modulus.

4.4.1 Unification of viscous model. As shown in equation (11), the tensor $\tau_v(\sigma_v^{NL})$ leads to a strongly non-linear formulation, and tuning the various parameters leads to different viscous models:

- (1) $f(\sigma_v^{NL}) = 1$ and $\alpha = 0$: Oldroyd-B model.
- (2) $F_v = 0, f(\sigma_v^{NL}) = 1$ and $\alpha = 0$: UCM model.
- (3) $F_v = 1$: classical Newtonian model.

4.4.2 Shear Thinning and Thickening. Despite unifying existing viscous models, to further quantify how the different parameters affect non-Newtonian viscous behavior, we define a novel non-dimensional and non-local Weissenberg number (W_e), based on the relaxation time λ_b as $W_e = \frac{\lambda_b}{\lambda_s}$, where $\lambda_s = \lambda_{0s} e^{-v(\Lambda-1)}$. The traditional W_e can be defined in various ways [Pakdel and McKinley 1996]. Our definition of W_e allows to readily test the shear thinning and thickening behaviors. A lower value of W_e postpones the shear thinning behavior, while the shear thickening behavior is obtained when $W_e < 1$ ($\lambda_s > \lambda_b$) [Clemeur and Debbaut 2007] (see Figure 3).

4.5 Governing Equations for Unified Viscous Flow

We use the unified constitutive model for viscous shear stress defined in equation (10) inside the Navier-Stokes equations as follows:

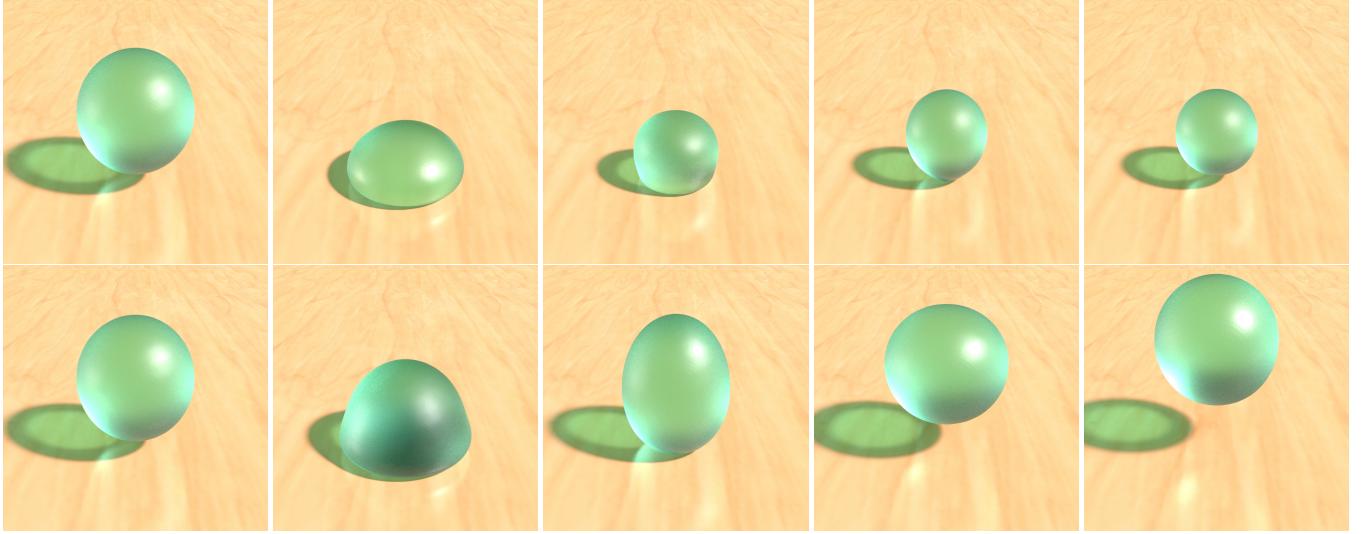


Fig. 7. (Top) The droplet impacting ground using $c_0 = 1$ suffers from severe volume loss, while the case with $c_0 = 10$ shows good volume preservation. The Δt for both cases are 2×10^{-5} s with an impact speed of 46 m/s.

$$\begin{aligned} \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} &= 0, \quad \rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot [-p\mathbf{I} + \boldsymbol{\sigma}_v(\mathbf{x}, \mathbf{u})] + \rho\mathbf{b}, \\ \boldsymbol{\sigma}_v(\mathbf{x}, t) &= \boldsymbol{\sigma}_v^L(\mathbf{x}, t) + \boldsymbol{\sigma}_v^{NL}(\mathbf{x}, t), \\ \boldsymbol{\sigma}_v^L(\mathbf{x}, t) &= 2F_v\mu\mathbf{D}(\mathbf{x}, t), \\ \boldsymbol{\sigma}_v^{NL}(\mathbf{x}, t) + \tau_v \frac{\hat{D}\boldsymbol{\sigma}_v^{NL}(\mathbf{x}, t)}{Dt} &= 2(1 - F_v)\mu\mathbf{D}(\mathbf{x}, t), \end{aligned} \quad (12)$$

where ρ is the density, \mathbf{u} is the velocity, $\boldsymbol{\sigma}_v$ is the viscous stress tensor, p is the pressure, and \mathbf{b} is the body force density.

4.6 Phase Change of Polymer Materials

As the role of additive manufacturing grows, demands for reliability are likely to increase. During 3D printing, a polymer filament is heated and deposited from a moving nozzle in a controlled fashion, forming layers of a specific shape. To simulate this process, we propose a novel formulation for thermo-visco-elastic coupling using our unified constitutive model for viscosity coupled with the generalized diffusion model proposed in [Xue et al. 2020]:

$$\begin{aligned} \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} &= \rho \dot{Q} \delta(\mathbf{x} - \mathbf{x}_s), \\ \rho \frac{D\mathbf{u}}{Dt} &= \nabla \cdot (-p\mathbf{I} + \boldsymbol{\sigma}) + \rho\mathbf{b}, \\ \rho c \frac{DT}{Dt} &= -\nabla \cdot \mathbf{q} + \nabla \cdot (-p\mathbf{u} + \boldsymbol{\sigma} \cdot \mathbf{u}) + \mathbf{u} \cdot \mathbf{b} + \rho c T_{inj} \dot{Q} \delta(\mathbf{x} - \mathbf{x}_s), \end{aligned} \quad (13)$$

where $\delta(\mathbf{x} - \mathbf{x}_s)$ represents a delta function located at \mathbf{x}_s , \dot{Q} is the volume source modeling the 3D printer nozzle located at \mathbf{x}_s , and T_{inj} is the temperature at \mathbf{x}_s . If the flow is assumed to be incompressible, the conservation of mass and energy equations in equation (13) can be simplified as follows:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= \dot{Q} \delta(\mathbf{x} - \mathbf{x}_s), \\ \rho c \frac{DT}{Dt} &= -\nabla \cdot \mathbf{q} + \mathbf{u} \cdot \mathbf{b} + \rho c T_{inj} \dot{Q} \delta(\mathbf{x} - \mathbf{x}_s), \end{aligned} \quad (14)$$

Since *latent heat* is an integral part of phase change [Voller and Swaminathan 1991], we introduce the solidus temperature T_m , where

the phase transition occurs, and index function I_s as shown below:

$$I_s \begin{cases} = 0 & T \geq T_m \quad \text{Fluid}, \\ = 1 & T < T_m \quad \text{Solid}, \end{cases}$$

and define the Cauchy stress tensor as a linear combination:

$$\boldsymbol{\sigma} = (1 - I_s)\boldsymbol{\sigma}_f + I_s\boldsymbol{\sigma}_s, \quad (15)$$

where $\boldsymbol{\sigma}_f$ and $\boldsymbol{\sigma}_s$ represent the deviatoric stresses in the fluid and solid phases, respectively. For $\boldsymbol{\sigma}_s$, we utilize the neo-Hookean elastic solid model $\boldsymbol{\sigma}_s = G J^{-5/3} \mathbf{B}' - p\mathbf{I}$, where G is the shear modulus, $\mathbf{B} = \mathbf{FF}^T$, $\mathbf{B}' = \mathbf{B} - 1/3 \text{tr} \mathbf{B}$, $J = \det(\mathbf{F})$, and \mathbf{F} is the deformation gradient tensor. For $\boldsymbol{\sigma}_f$, we use our unified constitutive model for viscosity coupled together with the C-F diffusion model recently proposed by [Xue et al. 2020].

$$\begin{aligned} \mathbf{q} &= \mathbf{q}^L + \mathbf{q}^{NL}, \quad \boldsymbol{\sigma}_v = \boldsymbol{\sigma}_v^L + \boldsymbol{\sigma}_v^{NL}, \\ \mathbf{q}^L &= -kF_T \nabla T, \quad \boldsymbol{\sigma}_v^L = 2F_v\mu(T)\mathbf{D}, \\ \tau \frac{\hat{D}\mathbf{q}^{NL}}{Dt} + \mathbf{q}^{NL} &= -k(1 - F_T) \nabla T, \quad \tau_v \frac{\hat{D}\boldsymbol{\sigma}_v^{NL}}{Dt} + \boldsymbol{\sigma}_v^{NL} = 2(1 - F_v)\mu(T)\mathbf{D}, \end{aligned}$$

where the specific upper-convected derivative is given in equation (7). Note, in particular, that we also use the upper-convected derivative for the non-local heat flux \mathbf{q}^{NL} to preserve the material-invariant formulation [Christov 2009; Mustafa 2015]. The heat influence on materials is revealed by the temperature-dependent viscosity coefficient value $\mu(T)$. Our proposed thermo-visco-elastic coupling model allows a wide range of applications involving phase-change phenomena. By tuning parameters F_v , F_T , and I_S , one can readily get:

- (1) Newtonian and non-Newtonian viscous fluid behavior,
- (2) Fourier and non-Fourier heat transfer
- (3) various combinations of temperature-dependent viscous fluid behaviors, and
- (4) various combinations of phase change models.

5 TIME DISCRETIZATION

In this section, we first derive a generalized viscous solver using a backward Euler discretization, and show that it recovers the variational Stokes solver of [Larionov et al. 2017]. Next, for better accuracy, we introduce the GS4 time integration scheme [Tamma et al. 2000; Zhou and Tamma 2004] and use it to derive a second order accurate time discretization for viscosity. Though the GS4 time integrator has been implemented to various engineering problems, such as thermal interface [Xue et al. 2018], heat transfer [Xue et al. 2016], and multi-body dynamics [Shimada and Tamma 2012], to the best of our knowledge, ours is the first work to use GS4 time integration to develop a second-order accurate numerical discretization of our unified constitutive model for viscous fluids.

5.1 Semi-implicit Backward Euler Scheme

We follow the splitting-scheme of [Stam 1999] to discretize equation (12) on a Cartesian grid [Harlow and Welch 1965], i.e., we first explicitly update the convection-related terms as shown below:

$$\begin{aligned} \frac{\mathbf{u}_* - \mathbf{u}_n}{\Delta t} + \mathbf{u}_n \cdot \nabla \mathbf{u}_n &= 0, \\ \frac{\boldsymbol{\sigma}_*^{NL} - \boldsymbol{\sigma}_n^{NL}}{\Delta t} + \mathbf{u}_n \cdot \nabla \boldsymbol{\sigma}_n^{NL} &= (\nabla \mathbf{u}_n)^T \cdot \boldsymbol{\sigma}_n^{NL} + \boldsymbol{\sigma}_n^{NL} \cdot (\nabla \mathbf{u}_n), \end{aligned} \quad (16)$$

followed by an implicit projection solve:

$$\begin{bmatrix} \rho \mathbf{I} & \Delta t \mathbf{D}^T & \Delta t \mathbf{D}^T & \Delta t \mathbf{G} \\ -2F_v \mathbf{D} & \mathbf{I} & 0 & 0 \\ -2(1 - F_v) \Delta t \mathbf{D} & 0 & (\tau_v + \Delta t) \mathbf{I} & 0 \\ -\mathbf{G}^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{n+1} \\ \boldsymbol{\sigma}_{n+1}^L \\ \boldsymbol{\sigma}_{n+1}^{NL} \\ p \end{bmatrix} = \begin{bmatrix} \Delta t \mathbf{b} + \rho \mathbf{u}_* \\ 0 \\ \tau_v \boldsymbol{\sigma}_*^{NL} \\ 0 \end{bmatrix} \quad (17)$$

where \mathbf{I} is the identity matrix, \mathbf{G} is the gradient operator for pressure, and \mathbf{D} is the discrete deformation rate operator for the velocity (i.e., $\mathbf{D}(\mathbf{u}) = 1/2 (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$). The negative transposes of \mathbf{G} and \mathbf{D} give us the discrete vector and tensor divergence operators, respectively. For fast numerical solutions using a Conjugate Gradients solver, a positive-definite formulation of equation (17) is provided in the appendix by applying the Schur-complement method. After solving for $\boldsymbol{\sigma}_{n+1}^L$, $\boldsymbol{\sigma}_{n+1}^{NL}$, and p_{n+1} , \mathbf{u}_{n+1} can be computed from equation (17). Once we set $F_v = 1$, equation (54) recovers the variational Stokes solve in [Larionov et al. 2017] for Newtonian viscosity.

5.2 Second order Semi-implicit GS4 Time Integration

Next, we show the generalized viscous solve using the second order accurate time integration scheme GS4 [Tamma et al. 2000; Zhou and Tamma 2004]. In contrast to first order accurate Euler schemes, the algorithmic time step $\tilde{t}_{n+W_1} = (n + W_1)\Delta t$ is introduced in equation (12), the divergence-free condition for the velocity \mathbf{u} , and all associated auxiliary equations. We follow the time-splitting on a Cartesian MAC grid for updating the following unknowns :

$$\Delta \dot{\mathbf{u}} = \Delta \dot{\mathbf{u}}_* + \Delta \dot{\mathbf{u}}_{**}, \quad \Delta \dot{\boldsymbol{\sigma}}^{NL} = \Delta \dot{\boldsymbol{\sigma}}_*^{NL} + \Delta \dot{\boldsymbol{\sigma}}_{**}^{NL}. \quad (18)$$

where $\Delta \dot{\mathbf{u}}_*$ and $\Delta \dot{\boldsymbol{\sigma}}_*^{NL}$ are evaluated explicitly, while $\Delta \dot{\mathbf{u}}_{**}$ and $\Delta \dot{\boldsymbol{\sigma}}_{**}^{NL}$ are evaluated implicitly. Specifically, we explicitly update the convection and upper-convection terms as follows:

$$\begin{aligned} (\dot{\mathbf{u}}_n + W_1 \Lambda_6 \Delta \dot{\mathbf{u}}_*) + \mathbf{u}_n \cdot \nabla \mathbf{u}_n &= 0, \\ (\dot{\boldsymbol{\sigma}}_n^{NL} + W_1 \Lambda_6 \Delta \dot{\boldsymbol{\sigma}}_*^{NL}) + \mathbf{u}_n \cdot \nabla \boldsymbol{\sigma}_n^{NL} &= (\nabla \mathbf{u}_n)^T \cdot \boldsymbol{\sigma}_n^{NL} + \boldsymbol{\sigma}_n^{NL} \cdot (\nabla \mathbf{u}_n), \end{aligned}$$

to compute $\Delta \dot{\mathbf{u}}_*$ and $\Delta \dot{\boldsymbol{\sigma}}_*^{NL}$. Subsequently, we use these values to compute the following quantities as shown below:

$$\begin{aligned} \tilde{\mathbf{u}}_* &= \mathbf{u}_n + W_1 \Lambda_6 \Delta \dot{\mathbf{u}}_*, \quad \tilde{\mathbf{u}}_* = \mathbf{u}_n + W_1 \Lambda_4 \dot{\mathbf{u}}_n \Delta t + W_2 \Lambda_5 \Delta \dot{\mathbf{u}}_* \Delta t, \\ \tilde{\boldsymbol{\sigma}}_*^{NL} &= \dot{\boldsymbol{\sigma}}_n^{NL} + W_1 \Lambda_6 \Delta \dot{\boldsymbol{\sigma}}_*^{NL}, \quad \tilde{\boldsymbol{\sigma}}_*^L = 2F_v \mu \mathbf{G} \tilde{\mathbf{u}}_*, \\ \tilde{\boldsymbol{\sigma}}_*^{NL} &= \boldsymbol{\sigma}_n^{NL} + W_1 \Lambda_4 \dot{\boldsymbol{\sigma}}_n^{NL} \Delta t + W_2 \Lambda_5 \Delta \dot{\boldsymbol{\sigma}}_*^{NL} \Delta t, \end{aligned} \quad (19)$$

where $W_1 \Lambda_6$, $W_1 \Lambda_4$, $W_2 \Lambda_5$, λ_4 , λ_5 are parameters defined in equation (46). Next, we formulate the algorithmic projection step as follows:

$$\begin{aligned} \rho \left(\tilde{\mathbf{u}}_* + W_1 \Lambda_6 \Delta \dot{\mathbf{u}}_{**} \right) + \nabla p - \nabla \cdot \left(\tilde{\boldsymbol{\sigma}}_*^L + \eta W_1 \Delta \boldsymbol{\sigma}_{**}^L \right) \\ - \nabla \cdot \left(\tilde{\boldsymbol{\sigma}}_*^{NL} + \eta W_2 \Lambda_5 \Delta \dot{\boldsymbol{\sigma}}_{**}^{NL} \Delta t \right) = 0 \\ \left(\tilde{\boldsymbol{\sigma}}_*^L + \eta W_1 \Delta \boldsymbol{\sigma}_{**}^L \right) = 2F_v \mu \mathbf{D} \left(\tilde{\mathbf{u}}_* + \eta W_2 \Lambda_5 \Delta \dot{\mathbf{u}}_{**} \Delta t \right) \\ \left(\tilde{\boldsymbol{\sigma}}_*^{NL} + W_2 \Lambda_5 \eta \Delta \dot{\boldsymbol{\sigma}}_{**}^{NL} \Delta t \right) + \boldsymbol{\tau}_v \left(\tilde{\boldsymbol{\sigma}}_*^L + W_1 \Lambda_6 \Delta \dot{\boldsymbol{\sigma}}_{**}^L \right) \\ = 2(1 - F_v) \mu \mathbf{D} \left(\tilde{\mathbf{u}}_* + \eta W_2 \Lambda_5 \Delta \dot{\mathbf{u}}_{**} \Delta t \right) \\ \nabla \cdot \left(\tilde{\mathbf{u}}_* + \eta W_2 \Lambda_5 \Delta \dot{\mathbf{u}}_{**} \Delta t \right) = 0 \end{aligned} \quad (20)$$

and our viscous solve is given by the following linear system:

$$\begin{bmatrix} \rho W_1 \Lambda_6 \mathbf{I} & \eta W_1 \mathbf{D}^T & \eta W_2 \Lambda_5 \Delta t \mathbf{D}^T & \mathbf{G} \\ -2F_v \mu \eta W_2 \Lambda_5 \Delta t \mathbf{D} & \mathbf{W}_1 \mathbf{I} & 0 & 0 \\ -2(1 - F_v) \mu \eta W_2 \Lambda_5 \Delta t \mathbf{D} & 0 & (\eta W_2 \Lambda_5 \Delta t + \boldsymbol{\tau}_v W_1 \Lambda_6) \mathbf{I} & 0 \\ \eta W_2 \Lambda_5 \Delta t \mathbf{G}^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{\mathbf{u}}_{**} \\ \Delta \boldsymbol{\sigma}_{**}^L \\ \Delta \dot{\boldsymbol{\sigma}}_{**}^{NL} \\ p \end{bmatrix} = \begin{bmatrix} -\rho \tilde{\mathbf{u}}_* - \mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^L - \mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^{NL} \\ 2\mu F_v \mathbf{D} \tilde{\mathbf{u}}_* - \tilde{\boldsymbol{\sigma}}_*^L \\ 2\mu(1 - F_v) \mathbf{D} \tilde{\mathbf{u}}_* - \left(\tilde{\boldsymbol{\sigma}}_*^{NL} + \boldsymbol{\tau} \tilde{\boldsymbol{\sigma}}_*^{NL} \right) \\ -\mathbf{G}^T \tilde{\mathbf{u}}_* \end{bmatrix} \quad (21)$$

where η is a user-defined parameter. Setting $\eta = 0$ makes this step explicit for $\Delta \dot{\mathbf{u}}_{**}$, $\Delta \boldsymbol{\sigma}_{**}^L$, and $\Delta \dot{\boldsymbol{\sigma}}_{**}^{NL}$. We transfer equation (21) to its positive-definite form (see Appendix C), and after solving for the unknowns $\Delta \boldsymbol{\sigma}_{**}^L$, $\Delta \dot{\boldsymbol{\sigma}}_{**}^{NL}$, p , the increment in velocity $\Delta \dot{\mathbf{u}}_{**}$ can be computed by equation (21), and the total increment by equation (18) to update the state to time t^{n+1} as follows:

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta \dot{\mathbf{u}} \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + \lambda_4 \dot{\mathbf{u}}_n \Delta t + \lambda_5 \Delta \dot{\mathbf{u}} \Delta t \\ \dot{\boldsymbol{\sigma}}_{n+1}^{NL} &= \dot{\boldsymbol{\sigma}}_n^{NL} + \Delta \dot{\boldsymbol{\sigma}}^{NL} \\ \boldsymbol{\sigma}_{n+1}^{NL} &= \boldsymbol{\sigma}_n^{NL} + \lambda_4 \dot{\boldsymbol{\sigma}}_n^{NL} \Delta t + \lambda_5 \Delta \dot{\boldsymbol{\sigma}}^{NL} \Delta t \end{aligned} \quad (22)$$

5.3 Second-order Partitioned and Explicit GS4 Scheme

Setting $\eta = 0$ in equation (21), the monolithic GS4 solve becomes:

$$\begin{aligned} \rho W_1 \Lambda_6 \mathbf{I} \Delta \dot{\mathbf{u}}_{**} + \mathbf{G} p &= -\rho \tilde{\mathbf{u}}_* - \mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^L - \mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^{NL} \\ \boldsymbol{\tau}_v W_1 \Lambda_6 \Delta \dot{\boldsymbol{\sigma}}_{**}^{NL} &= 2\mu(1 - F_v) \mathbf{D} \tilde{\mathbf{u}}_* - \left(\tilde{\boldsymbol{\sigma}}_*^{NL} + \boldsymbol{\tau} \tilde{\boldsymbol{\sigma}}_*^{NL} \right) \end{aligned} \quad (23)$$

In order to formulate a Poisson equation for p , we take the divergence of the update for \mathbf{u}_{n+1} in equation (22) as follows:

$$\mathbf{G}^T \mathbf{u}_{n+1} = \underbrace{\mathbf{G}^T (\mathbf{u}_n + \lambda_4 \dot{\mathbf{u}}_n \Delta t + \lambda_5 \Delta \dot{\mathbf{u}} \Delta t) + \lambda_5 \Delta t \mathbf{G}^T \Delta \dot{\mathbf{u}}_{**}}_{\text{Known Value}} = 0 \quad (24)$$

Setting $\mathbf{G}^T \mathbf{u}_{n+1} = 0$, taking the divergence of the first equation in (23) and substituting $\mathbf{G}^T \Delta \mathbf{u}_{**}$ from equation (24) yields:

$$\rho W_1 \Lambda_6 \mathbf{G}^T \Delta \mathbf{u}_{**} + \mathbf{G}^T \mathbf{G} p = -\mathbf{G}^T \left(\rho \tilde{\mathbf{u}}_* + \mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^L + \mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^{NL} \right) \quad (25)$$

and the Poisson equation for p in the GS4 framework is given by

$$\begin{aligned} \mathbf{G}^T \mathbf{G} p &= \frac{\rho W_1 \Lambda_6}{\lambda_5 \Delta t} \mathbf{G}^T \mathbf{u}_n + \frac{\rho (\lambda_4 W_1 \Lambda_6 - \lambda_5)}{\lambda_5} \mathbf{G}^T \mathbf{u}_n \\ &\quad - \mathbf{G}^T \mathbf{D}^T \left(\tilde{\boldsymbol{\sigma}}_*^L + \tilde{\boldsymbol{\sigma}}_*^{NL} \right). \end{aligned} \quad (26)$$

The structure of the above GS4-based Poisson equation is the same as that of the pressure Poisson equation in incompressible flow (albeit with a different right hand side), and can be readily incorporated into existing solvers. For further simplicity, one can also use the weakly compressible condition [Batchelor and Batchelor 2000] to evaluate the pressure at t_{n+W_1} , which leads to a fully explicit scheme:

$$\tilde{p}_p = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\tilde{\rho}_p}{\rho_0} \right)^Y - 1 \right] \quad (27)$$

where c_0 is the sound speed, ρ_0 is the reference density, $\gamma = 7$, and ρ_p is the density at particle locations. The pressure evaluation is given in Appendix D. We use equation (57) to integrate with MPM for fluid simulations (see Section 6), and Section 7 compares the influence of c_0 on the volume loss of the droplet (see Figure 7).

5.4 Boundary Conditions

We use the no-slip boundary condition ($\mathbf{u} = 0$) to handle interactions between the solid and the fluid, which can be readily implemented with MPM [Fang et al. 2019]. The free surface boundary condition for non-local viscous flow is given by:

$$\mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{n}^T = 0 \quad (28)$$

where $\boldsymbol{\sigma} = -p \mathbf{I} + \boldsymbol{\sigma}_v^L + \boldsymbol{\sigma}_v^{NL}$ denotes the total stress and \mathbf{n} represents a unit vector that is normal and external to the surface. For three-dimensional Cartesian coordinates, equation (28) leads to the following boundary condition for the free surface pressure:

$$\begin{aligned} p &= (\sigma_v)_{xx} n_x^2 + (\sigma_v)_{yy} n_y^2 + 2(\sigma_v)_{xy} n_x n_y \\ &\quad + (\sigma_v)_{xx} n_x^2 + (\sigma_v)_{zz} n_z^2 + 2(\sigma_v)_{xz} n_x n_z \\ &\quad + (\sigma_v)_{zz} n_z^2 + (\sigma_v)_{yy} n_y^2 + 2(\sigma_v)_{zy} n_z n_y \end{aligned} \quad (29)$$

where $\boldsymbol{\sigma}_v = \boldsymbol{\sigma}_v^L + \boldsymbol{\sigma}_v^{NL}$. To implement the above in our GS4-based Poisson system (equation (26)), we treat the free surface as a non-zero Dirichlet boundary condition using the value in equation (29). To integrate with MPM, the pressure on particles located at the free surface can be directly evaluated using equation (29).

6 INTEGRATION WITH MPM

We now describe the integration of our GS4 discretization from Section 5 with the Material Point Method (MPM) for simulating the polymer melting/solidification process in equation (13). In contrast to the traditional MPM with first-order Euler scheme, our method uses the GS4 explicit scheme for the dynamics while the GS4 semi-implicit solve (see Appendix B) is used for non-Fourier thermal processes. To the best of our knowledge, our integration of GS4-MPM is novel and has not been explored in prior work. We reserve subscripts p, q, r for quantities stored on particles, and i, j, k for

Table 2. Physical quantities stored on particles and grid nodes.

Particle	Description	Grid
x_p	position	x_i
v_p	velocity	v_i
a_p	acceleration	a_i
F_p	deformation gradient	--
$\boldsymbol{\sigma}_p^L$	local viscous stress	--
$\boldsymbol{\sigma}_p^{NL}$	non-local viscous stress	$\boldsymbol{\sigma}_i^{NL}$
$\boldsymbol{\sigma}_p^S$	solid stress	--
T_p	temperature	T_i
\mathbf{q}_p^L	local diffusion flux	\mathbf{q}_i^L
\mathbf{q}_p^{NL}	non-local diffusion flux	\mathbf{q}_i^{NL}
--	force	f_i
V_p	volume	--
m_p	mass	m_i
J_p	volume change	--
p_p	pressure	--

quantities stored on grid nodes. Table 2 summarizes the various quantities stored on particles and grid nodes.

6.1 Algorithm Description

We now describe our method in detail. Figure 8 gives a high-level overview. At the beginning of each time step, we evaluate algorithmic qualities $\tilde{x}_p, \tilde{v}_p, \tilde{a}_p, \tilde{T}_p, \tilde{\dot{T}}_p$, and $\tilde{\mathbf{q}}_p^{NL}$ as follows:

$$\begin{aligned} \tilde{a}_p &= \boldsymbol{a}_p^n, \quad \tilde{v}_p = \boldsymbol{v}_p^n + W_1 \Lambda_4 \boldsymbol{a}_p^n \Delta t, \\ \tilde{x}_p &= \mathbf{x}_p^n + W_1 \Lambda_1 \boldsymbol{v}_p^n \Delta t + W_2 \Lambda_2 \boldsymbol{a}_p^n \Delta t^2, \\ \tilde{\dot{T}}_p &= \dot{T}_p^n, \quad \tilde{T}_p = T_p^n + W_1 \Lambda_4 \dot{T}_p^n \Delta t, \end{aligned} \quad (30)$$

where $W_1 \Lambda_1$ and $W_2 \Lambda_2$ are given in equation (46) in Appendix A. Based on the particle location \tilde{x}_p , we evaluate the quadratic B-spline weight \tilde{w}_{ip} and its gradient $\nabla \tilde{w}_{ip}$ of particle p at grid node i at time t_{n+W_1} . We rasterize qualities $m_p, \tilde{a}_p, \tilde{v}_p$, and \tilde{T}_p at time t_{n+W_1} and $\boldsymbol{a}_p^n, \boldsymbol{v}_p^n, (\boldsymbol{\dot{\sigma}}_p^{NL})^n, (\boldsymbol{\sigma}_p^{NL})^n, T_p^n, \dot{T}_p^n, (\mathbf{q}_p^{NL})^n$, and $\dot{\mathbf{q}}_p^{NL}$ at time t_n from particles to grids. m_p is first rasterized as $\tilde{m}_i^n = \sum_p m_p \tilde{w}_{ip}$. Then, the remaining quantities are rasterized according to the following normalization during the interpolation step as shown below:

$$A_i = \frac{1}{\tilde{m}_i} \sum_p A_p m_p \tilde{w}_{ip} \quad \text{for } A_i \text{ at } t_n \text{ and } t_{n+W_1} \quad (31)$$

where A_i is a generalized variable. We also have $\nabla \tilde{T}_p = \sum_i \tilde{T}_i \nabla \tilde{w}_{ip}$, $\nabla \tilde{v}_p = \sum_i \tilde{v}_i \nabla \tilde{w}_{ip}$, and $\nabla \tilde{a}_p = \sum_i \tilde{a}_i \nabla \tilde{w}_{ip}$. The deformation gradient tensor $\tilde{\mathbf{F}}_p$ can then be computed at particle locations as follows:

$$\tilde{\mathbf{F}}_p = \left(\mathbf{I} + W_1 \Lambda_1 \Delta t \nabla \tilde{v}_p + W_2 \Lambda_2 \Delta t^2 \nabla \tilde{a}_p \right) F_p^n \quad (32)$$

We use equation (27) to evaluate the pressure \tilde{p}_p . Next, we evaluate the physical quantities $\tilde{\mathbf{q}}_p^L, \tilde{\boldsymbol{\sigma}}_p^L, \tilde{\boldsymbol{\sigma}}_p^{NL}, \tilde{\boldsymbol{\sigma}}_p^{NL}$, and $\tilde{\mathbf{q}}_p^{NL}$ at particle

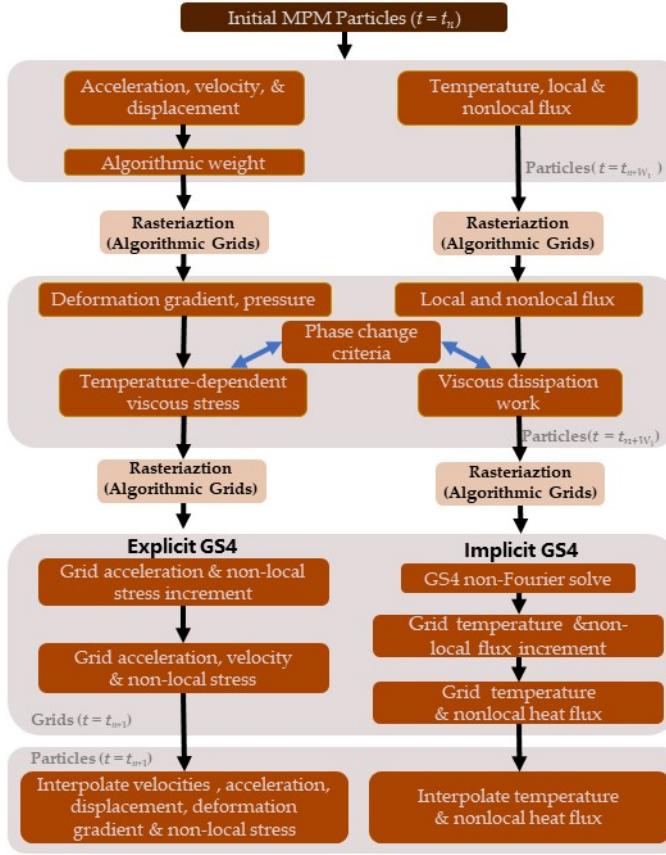


Fig. 8. Overview of our proposed method in conjunction with the Material Point Method (MPM) for phase change problems. Our method benefits from the interplay of grids and particles. We rasterize all algorithmic quantities from particles to grids and use GS4 schemes to solve mechanical and thermal systems, respectively. We consider the coupling where the stress is temperature-dependent in the dynamic system (left column) while the thermal evolution (right column) provides the temperature distribution and determines when and where the phase change happens.

locations as follows:

$$\begin{aligned} \tilde{\mathbf{q}}_p^L &= -F_T k \nabla \tilde{T}_p, \quad \tilde{\sigma}_p^L = F_v \mu(\tilde{T}_p) \left[\nabla \tilde{\mathbf{v}}_p + (\nabla \tilde{\mathbf{v}}_p)^T \right] \\ \tilde{\dot{\sigma}}_p^{NL} &= (\dot{\sigma}_n^{NL})_p + W_1 \Lambda_6 (\Delta \dot{\sigma}_*^{NL})_p, \\ \tilde{\dot{\mathbf{q}}}_p^{NL} &= (\dot{\mathbf{q}}_n^{NL})_p + W_1 \Lambda_6 (\Delta \dot{\mathbf{q}}_*^{NL})_p, \quad (33) \\ \tilde{\sigma}_p^{NL} &= (\sigma_n^{NL})_p + W_1 \Lambda_4 \Delta t (\dot{\sigma}_n^{NL})_p + W_2 \Lambda_5 \Delta t (\Delta \dot{\sigma}_*^{NL})_p, \\ \tilde{\dot{\mathbf{q}}}_p^{NL} &= (\dot{\mathbf{q}}_n^{NL})_p + W_1 \Lambda_4 \Delta t (\dot{\mathbf{q}}_n^{NL})_p + W_2 \Lambda_5 \Delta t (\Delta \dot{\mathbf{q}}_*^{NL})_p, \end{aligned}$$

where

$$\begin{aligned} (\Delta \dot{\sigma}_*^{NL})_p &= \frac{1}{W_1 \Lambda_6} \left[(\nabla \tilde{\mathbf{v}}_p)^T \cdot \tilde{\sigma}_p^{NL} + \tilde{\sigma}_p^{NL} \cdot (\nabla \tilde{\mathbf{v}}_p) - (\dot{\sigma}_n^{NL})_p \right] \\ (\Delta \dot{\mathbf{q}}_*^{NL})_p &= \frac{1}{W_1 \Lambda_6} \left[(\nabla \tilde{\mathbf{v}}_p)^T \cdot \tilde{\mathbf{q}}_p^{NL} + \tilde{\mathbf{q}}_p^{NL} \cdot (\nabla \tilde{\mathbf{v}}_p) - (\dot{\mathbf{q}}_n^{NL})_p \right] \quad (34) \end{aligned}$$

and the viscosity coefficient $\mu(\tilde{T}_p)$ is evaluated as per the temperature \tilde{T}_p based on the following Cross-WLF model [Bilovol 2003]:

$$\mu = \frac{\mu_0}{1 + 5.66 \times 10^3 (\mu_0 \dot{\gamma})^{-0.75}}, \quad \mu_0 = 3.32 \times 10^9 \exp \left[-\frac{20.194(\tilde{T}_p - 100)}{\tilde{T}_p - 48.4} \right],$$

where $\dot{\gamma}$ is the shear rate, calculated as $\dot{\gamma} = \sqrt{2(\mathbf{D}\mathbf{u}) : (\mathbf{D}\mathbf{u})}$. After this step, we rasterize the resulting quantities $\tilde{\mathbf{q}}_p^L$, $\tilde{\sigma}_p^L$, $\tilde{\dot{\sigma}}_p^{NL}$, $\tilde{\mathbf{q}}_p^{NL}$, and $\tilde{\dot{\mathbf{q}}}_p^{NL}$ at time t_{n+W_1} from particles to grids. Next, we evaluate $\Delta \mathbf{a}_i$ and $\Delta \dot{\sigma}_i^{NL}$ using the explicit GS4 scheme, and $\Delta \dot{T}_i$ and $\Delta \dot{\mathbf{q}}_i^{NL}$ using the semi-implicit GS4 scheme (see Appendix B).

6.1.1 Explicit scheme for dynamics simulation. The stresses at time t_{n+W_1} are computed as follows:

$$\begin{aligned} \tilde{\sigma}_p^f &= -\tilde{\mathbf{p}}_p \mathbf{I} + \tilde{\sigma}_p^L + \tilde{\sigma}_p^{NL}, \\ \sigma_p &= (1 - I_s) \tilde{\sigma}_p^f + I_s \tilde{\sigma}_p^s \end{aligned} \quad (35)$$

where $\tilde{\sigma}_s$ is computed using \tilde{F}_p according to the stress evaluation in standard MPM [Stomakhin et al. 2013]. Then, the internal stress-based force per grid node is computed as $\tilde{\mathbf{f}}_i = -\sum_p V_p \tilde{\sigma}_p \nabla \tilde{\mathbf{w}}_{ip}$, and used to update the nodal acceleration as shown below:

$$(\Delta \mathbf{a})_i = \frac{1}{W_1 \Lambda_6} \left(\frac{\tilde{\mathbf{f}}_i + \mathbf{g}}{\tilde{m}_i} - \tilde{\mathbf{a}}_i^n \right), \quad (36)$$

where \mathbf{g} is the acceleration due to gravity. We compute $\Delta \dot{\sigma}_i^{NL}$ as:

$$\Delta \dot{\sigma}_i^{NL} = \frac{1}{W_1 \Lambda_6 \tilde{\tau}_v} \left[2\mu(\tilde{T})(1 - F_v) \nabla \tilde{\mathbf{u}}_i - \tilde{\sigma}_i^{NL} - \tilde{\tau}_v \tilde{\dot{\sigma}}_i^{NL} \right] \quad (37)$$

where the nonlinear term of $\tilde{\tau}_v$ is given by equation (11) at time t_{n+W_1} . We update all physical quantities on grids nodes as follows:

$$\begin{aligned} \mathbf{a}_i^{n+1} &= \mathbf{a}_i^n + \Delta \mathbf{a}_i \\ \mathbf{v}_i^{n+1} &= \mathbf{v}_i^n + \lambda_4 \Delta t \mathbf{a}_i^n + \lambda_5 \Delta t \Delta \mathbf{a}_i \\ (\dot{\sigma}_i^{NL})^{n+1} &= (\dot{\sigma}_i^{NL})^n + \Delta \dot{\sigma}_i^{NL} \\ (\sigma_i^{NL})^{n+1} &= (\sigma_i^{NL})^n + \lambda_4 \Delta t \dot{\sigma}_i^n + \lambda_5 \Delta t \Delta \dot{\sigma}_i^{NL} \end{aligned} \quad (38)$$

Collisions with objects are treated following [Stomakhin et al. 2013].

6.1.2 Implicit scheme for non-Fourier heat transfer. We compute $\Delta \dot{T}_i$ and $\Delta \dot{\mathbf{q}}_i^{NL}$ using a modification of the non-Fourier solve from [Xue et al. 2020] (see equations (51) and (53) in Appendix B), as follows:

$$\begin{aligned} \left[W_1 \Lambda_6 \mathbf{I} - \frac{\Delta t W_2 \Lambda_5 (F_T \tau W_1 \Lambda_6 + \Delta t W_2 \Lambda_5)}{(W_2 \Lambda_5 \Delta t + W_1 \Lambda_6 \tau)} k \nabla^2 \right] \Delta \dot{T}_i &= \\ (F_T k \nabla^2 - \mathbf{I}) \tilde{T}_i + \frac{\tau (\nabla \cdot \dot{\mathbf{q}}^{NL})_i + (\nabla \cdot \tilde{\mathbf{q}}^{NL})_i + (1 - F_T) k \nabla^2 \tilde{T}_i}{(W_2 \Lambda_5 \Delta t + W_1 \Lambda_6 \tau)} & \\ - (\nabla \cdot \tilde{\mathbf{q}}^{NL})_i + (\nabla \cdot (\tilde{\sigma}_p \cdot \tilde{\mathbf{v}}_p))_i + \tilde{\mathbf{v}}_i \cdot \mathbf{g} + S_i & \end{aligned} \quad (39)$$

where S_i represents the external heat source/sink on grid nodes. After solving for $\Delta \dot{T}_i$, we compute $\Delta \dot{\mathbf{q}}_i^{NL}$ as follows:

$$\Delta \dot{\mathbf{q}}_i^{NL} = -\frac{\tau \tilde{\mathbf{q}}_i^{NL} + (1 - F_T) k \nabla \tilde{T}_i^*}{(\eta W_2 \Lambda_5 \Delta t + W_1 \Lambda_6 \tau)} \quad (40)$$

where $\tilde{T}_i^* = \tilde{T}_i + W_2 \Lambda_5 \Delta t \Delta \dot{T}_i$. We compute \dot{T}_i^{n+1} , T_i^{n+1} , $(\dot{q}_i^{NL})^{n+1}$ and $(q_i^{NL})^{n+1}$ as follows:

$$\begin{aligned}\dot{T}_i^{n+1} &= \dot{T}_i^n + \Delta \dot{T}_i, \quad (\dot{q}_i^{NL})^{n+1} = (\dot{q}_i^{NL})^n + \Delta \dot{q}_i^{NL} \\ T_i^{n+1} &= T_i^n + \lambda_4 \Delta t \dot{T}_i^n + \lambda_5 \Delta t \Delta \dot{T}_i, \\ (q_i^{NL})^{n+1} &= (q_i^{NL})^n + \lambda_4 \Delta t (\dot{q}_i^{NL})^n + \lambda_5 \Delta t \Delta \dot{q}_i^{NL}\end{aligned}\quad (41)$$

We update a_p^{n+1} , v_p^{n+1} , \dot{T}_p^{n+1} , T_p^{n+1} , $(\dot{q}_p^{NL})^{n+1}$, and $(q_p^{NL})^{n+1}$ considering the PIC and FLIP components as follows:

(1) PIC update ($\kappa_{\text{PIC}}^{n+1}$)

$$\begin{aligned}a_p^{n+1} &= \sum_i a_i^{n+1} \tilde{w}_{ip}, \quad v_p^{n+1} = \sum_i v_i^{n+1} \tilde{w}_{ip}, \\ \dot{T}_p^{n+1} &= \sum_i \dot{T}_i^{n+1} \tilde{w}_{ip}, \quad T_p^{n+1} = \sum_i T_i^{n+1} \tilde{w}_{ip}, \\ (\dot{q}_p^{NL})^{n+1} &= \sum_i (\dot{q}_i^{NL})^{n+1} \tilde{w}_{ip}, \quad (q_p^{NL})^{n+1} = \sum_i (q_i^{NL})^{n+1} \tilde{w}_{ip}.\end{aligned}\quad (42)$$

(2) FLIP update ($\kappa_{\text{FLIP}}^{n+1}$)

$$\begin{aligned}a_p^{n+1} &= a_p^n + \sum_i \Delta a_i \tilde{w}_{ip}, \\ v_p^{n+1} &= v_p^n + \lambda_4 \Delta t a_p^n + \lambda_5 \Delta t \sum_i \Delta a_i \tilde{w}_{ip}, \\ \dot{T}_p^{n+1} &= \dot{T}_p^n + \sum_i \dot{T}_i^{n+1} \tilde{w}_{ip}, \\ T_p^{n+1} &= T_p^n + \lambda_4 \Delta t \dot{T}_p^n + \lambda_5 \Delta t \sum_i \dot{T}_i^{n+1} \tilde{w}_{ip}, \\ (\dot{q}_p^{NL})^{n+1} &= (\dot{q}_p^{NL})^n + \sum_i \Delta \dot{q}_i^{NL} \tilde{w}_{ip}, \\ (q_p^{NL})^{n+1} &= (q_p^{NL})^n + \lambda_4 \Delta t (\dot{q}_p^{NL})^n + \lambda_5 \Delta t \sum_i \Delta \dot{q}_i^{NL} \tilde{w}_{ip},\end{aligned}\quad (43)$$

and the final updated formula is given by:

$$\kappa^{n+1} = (1 - \alpha) \kappa_{\text{PIC}}^{n+1} + \alpha \kappa_{\text{FLIP}}^{n+1}, \quad (44)$$

where κ is a generalized variable, and the parameter $\alpha \in [0, 1]$ allows for transitioning between fully PIC and FLIP update formulas. Once post-collision velocities v_p^{n+1} have been computed, we use them to update the particle positions x_p^{n+1} and F_p^{n+1} as follows:

$$\begin{aligned}x_p^{n+1} &= x_p^n + \lambda_1 \Delta t v_p^n + \lambda_2 \Delta t^2 a_p^n + \lambda_3 \Delta t^2 \sum_i \Delta a_i \tilde{w}_{ip} \\ F_p^{n+1} &= (\mathbf{I} + \lambda_1 \Delta t \nabla v_p^n + \lambda_2 \Delta t^2 \nabla a_p^n + \lambda_3 \Delta t^2 \sum_i \Delta a_i \nabla \tilde{w}_{ip}) F_p^n\end{aligned}\quad (45)$$

Once x_p is updated, one can also use $\nabla \tilde{w}_{ip}^{n+1}$ to update F_p^{n+1} .

6.2 Geometric Multigrid Solver

We follow the same strategy as outlined in [Xue et al. 2020] for the non-Fourier solve. The implicit system in equation (51) (see Appendix B) is still *elliptic* [Trottenberg et al. 2001]. However, it is much better-conditioned than a traditional Poisson system because of the presence of a scaled identity term on the diagonal. This allows iterative Krylov solvers such as preconditioned Conjugate Gradients (PCG) to converge in relatively fewer iterations. To allow for even

faster convergence rates, we design a geometric Multigrid preconditioner, closely following the description in [McAdams et al. 2010]. Specifically, the restriction and prolongation operators are the same as those described in [McAdams et al. 2010] for pressure projection. For implementing the smoother, we pass down all constants, such as τ , Δt , F_T , W_1 , W_2 , Λ_5 , Λ_6 and k to all levels in the Multigrid hierarchy for multiplying a vector with the system matrix in equation (51).

7 RESULTS

Accompanying this article, we open-source our code for running 3D examples with our unified constitutive model for viscosity. The noteworthy advantage emanating from such a unified model is the practicality and convenience of using the same numerical implementation when simulating different viscous fluids without having to switch from one model to another. In all our 3D examples, we utilize the fully explicit GS4 scheme for fluid simulations, while the semi-implicit GS4 non-Fourier solve (see Appendix B) is coupled for phase change problems by setting $F_T = 1$, $k = 0.01$, and $T_m = 0.1$.

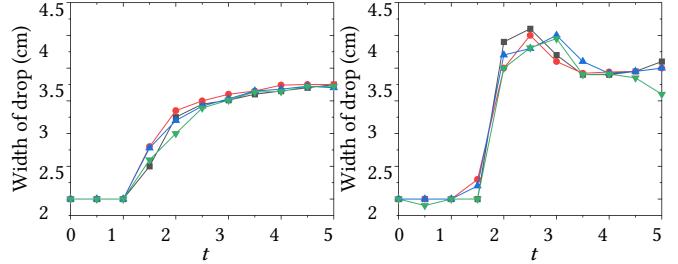


Fig. 9. Width of a droplet vs. time. (Left) Newtonian droplet at $R_e = 5.0$ and $F_r = 2.26$. (Right) non-Newtonian droplet at $Re = 5.0$ using semi-implicit GS4 (red circle), partitioned GS4 (blue up-triangle), fully explicit GS4 (green down-triangle), and the method of [Oishi et al. 2012] (black square). Comparing the width of the droplet during impact confirms that the present study quantitatively matches the results in [Oishi et al. 2012].

7.1 Quantitative Numerical Evaluation

7.1.1 Validation. To validate our proposed method, we computed the width of the Newtonian and non-Newtonian fluid droplets obtained using the proposed fully explicit GS4 scheme and the method of [Oishi et al. 2012]. Figure 9 shows that our results are in good agreement with the results published in [Oishi et al. 2012]. Besides, by setting $(\rho_\infty = \rho_s^\infty = 1)$, our proposed partitioned GS4 scheme recovers the numerical method in [Oishi et al. 2012], which is the Crank-Nicolson scheme for integrating the solution forward in time.

7.1.2 Volume preservation. We ran several simulations of falling droplets to compare our implementation based on different values of c_0 in the Tait's pressure equation (explicit GS4 schemes), and with partitioned and semi-implicit GS4 schemes. We measure the total volume on grids by $V = \sum_i m_i / \rho_i$, where ρ_i is updated at each time step. Figure 10 shows a severe volume loss in cases with small c_0 , while the case with $c_0 = 10$ preserves its volume fairly close to the reference. Figure 7 shows the comparison of droplet deformations with $c_0 = 1$ and $c_0 = 10$. The volume of the droplet with small c_0 decreases and increases depending on the impact, while the case with large $c_0 = 10$ maintains a fairly constant volume. As evident from Figure 10, the partitioned and semi-implicit GS4 schemes are much better in terms of their volume preservation ability.

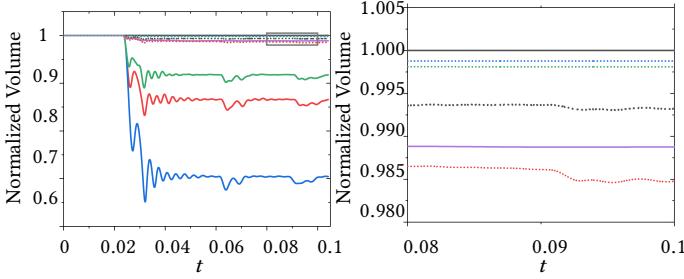


Fig. 10. Volume preservation of the falling droplet with different algorithms. Solid black guidance line indicates ideal volume preservation. **Explicit GS4** (solid lines): $c_0 = 0.1$: blue line; $c_0 = 0.5$: red line; $c_0 = 1$: green line, and $c_0 = 10$: purple line. **Partitioned GS4** (dotted lines): tolerance= 10^{-3} : black line and tolerance= 10^{-4} : blue line. **Semi-implicit GS4** (dotted lines): tolerance= 10^{-3} : red line and tolerance= 10^{-4} : green line. The volume of the droplet with explicit GS4 ($c_0 = 10$), partitioned GS4, and semi-implicit GS4 schemes preserve its volume. (Right): Zoom-in over $t \in [0.08, 0.1]$

7.1.3 Time accuracy. We quantitatively evaluate the temporal accuracy of our method on a falling droplet (see accompanying code) by varying the time step Δt . We utilized numerical results with $\Delta t = 10^{-8}$ as the benchmark solution and ran sampling examples with $\Delta t = 10^{-i}$, $i \in \{1, \dots, 6\}$ up to a total simulation time of 2s, so that the absolute difference between the benchmark solution and the sampling solution provided us with the error value. Figure 11 shows the convergence plots for the particle displacement and velocity for the semi-implicit backward Euler scheme, fully explicit and semi-implicit GS4 schemes, as described in Section 5. In general, high-order time integration schemes can produce more accurate simulations than first-order accurate Euler-based schemes with the same time step. Our discretizations using GS4 exhibit lower error and higher convergence order than semi-implicit backward Euler. We observe that the fully explicit GS4 scheme loses these advantages with large Δt since it requires a time step restriction for stability.

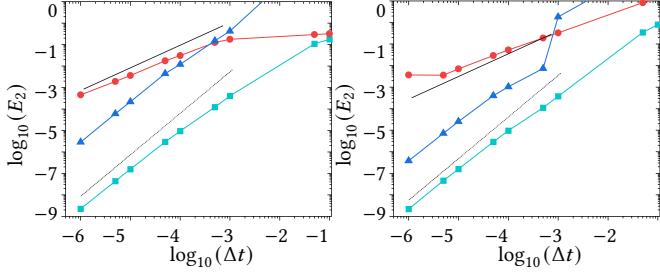


Fig. 11. Log-log plots of L_2 error (labelled E_2) vs. time interval size Δt . (Left) L_2 error of the displacement. (Right) L_2 error of the velocity using semi-implicit backward Euler (red circle), fully explicit GS4 (blue triangle), and semi-implicit GS4 (green square). Solid and dotted guide lines indicate ideal first and second order slopes. Comparing the slopes and solution accuracy confirms that the second-order GS4 schemes significantly improve the convergence and solution accuracy for cases with small Δt .

7.1.4 Efficiency. We use the same droplet falling case to evaluate the cost of each scheme. We use $\Delta t = 0.001$ in the backward Euler/GS4 semi-implicit schemes and $\Delta t = 0.0001$ in the fully explicit GS4 scheme, and run each case for the same target simulation time. As shown in the first three rows of Table 3, the computational cost of the semi-implicit schemes is significant because of the size of the linear systems in equations (54) and (55), which is $2197n^3$ in

3D, where n is the number of degrees of freedom. In contrast, the partitioned scheme (26) only needs to solve an $n \times n$ Poisson equation for pressure and the fully explicit scheme does not require any solve. In all our 3D examples, we observed that the fully explicit GS4 scheme was stable. Table 3 summarizes the specific timings for all examples, and the parameters used are listed in Table 4. Δt is chosen in an adaptive manner restricted by the CFL conditions [Caiden et al. 2001] $\Delta t = 0.5 \min\{\Delta t_1, \Delta t_2\}$, where $\Delta t_1 |\mathbf{u}_{max}| / \Delta x \leq 1$ and $\Delta t_2 [2\mu / (\rho \Delta x^2) + |\mathbf{u}_{max}| / \Delta x] \leq 1$, where \mathbf{u}_{max} is the maximum fluid velocity.

7.2 Droplet Impact of Shear Thickening Liquids

Our method produces an appealing shear thickening behavior, similar to the observed behavior in the real world¹ via tuning the Weissenberg number (W_e) and non-local effect (F_v). At the moment of impact, the rate of shear strain increases abruptly, which triggers different mechanisms of viscosity. The classical Newtonian cube shows linear relationship between the shear stress and shear strain such that it expands smoothly and becomes a thin layer, as shown in Figure 3(top). In contrast, the non-Newtonian cube (see Figure 3(bottom)) oscillates due to the increased elastic portion of the viscous stress and rebounds back immediately since viscosity increases with the rate of shear strain, exhibiting solid-like behavior. Moreover, as W_e increases, we observe stronger solid-like behavior of the viscous cube. We also simulated a falling Stanford bunny (see Figure 5), showing the bouncing back due to shear thickening viscous forces.

Table 3. All simulations were run on Machine 1: Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHZ and Machine 2: Intel(R) Xeon(R) CPU E5-1620 v4 @ 3.50GHz. Simulation time is measured in average seconds per time step. **Grid:** The number of occupied voxels in the background sparse grid. **Particle:** The total number of MPM particles in the simulation.

Simulation	Time	Machine	Grid	Particle
Droplet (Euler Semi-implicit)	9.05	1	8K	57K
Droplet (GS4 Semi-implicit)	12.28	1	8K	57k
Droplet (GS4 Explicit)	1.25	1	8K	57K
Cube impact(Fig.3(1st row))	0.147	2	2M	35K
Cube impact(Fig.3(2nd row))	0.153	2	2M	35K
Cube impact(Fig.3(3rd row))	0.159	2	2M	35K
Cube impact(Fig.3(4th row))	0.155	2	2M	35K
Coiling 1 (Fig.2 (top))	0.225	2	524K	190K
Coiling 2 (Fig.2 (middle))	0.191	2	524K	188K
Coiling 3 (Fig.2 (bottom))	0.272	2	524K	188K
Buckling 1 (Fig.6 (top))	3.3679	2	33M	2.4M
Buckling 2 (Fig.6 (bottom))	3.0178	2	33M	2.8M
Ball impact 1 (Fig.7 (top))	0.232	2	2M	562K
Ball impact 2 (Fig.7 (bottom))	0.229	2	2M	562K
Bunny impact (Fig.5)	0.1715	2	33M	52K
Hourglass 1 (Fig.12 (left))	0.2165	2	2M	82K
Hourglass 2 (Fig.12 (right))	0.1939	2	2M	82K
Bunny piling (Fig.4)	0.5758	2	33M	420K
Stamping bunny (Fig.1 (top right))	1.0448	2	33M	420K
Bunny melting 1 (Fig.14 (top))	0.0928	2	2M	26K
Bunny melting 2 (Fig.14 (bottom))	0.0918	2	2M	26K
3D printing (Fig.15)	0.0013	2	4.2M	156K

¹https://www.youtube.com/watch?v=Ol6bBB3zuGc&ab_channel=BarryBelmont



Fig. 12. Under the effects of gravity, a Newtonian liquid flows out uniformly (left), while a non-Newtonian liquid exhibits viscoelasticity, forming clumps. The average Δt during simulations is 10^{-5} s.

Table 4. We provide a list of the main parameters used in the simulations. In the Newtonian case ($F_v = 1$), λ_b , λ_{0s} , G_0 , α , and Q are dummy parameters.

Simulation	F_v	λ_b	λ_{0s}	μ	G_0	α	Q
Cube impact 1	1	N/A	N/A	4.9	N/A	N/A	N/A
Cube impact 2	0.01	0.98	0.8820	4.9	4.95	100	8
Cube impact 3	0.01	0.098	0.0882	4.9	49.5	100	8
Cube impact 4	0.01	0.0098	0.0088	4.9	495	100	8
Ball impact	0.01	0.0098	0.0088	4.9	495	100	8
Bunny impact	0.01	0.613	0.0551	306.25	4950	1	8
Coiling 1	1	N/A	N/A	61.25	N/A	N/A	N/A
Coiling 2	0.02	12.25	12.25	3063	245	0	16
Coiling 3	0.02	12.25	11.025	3063	245	1	16
Buckling 1	1	N/A	N/A	30.625	N/A	N/A	N/A
Buckling 2	0.01	3.0625	2.7563	3063	990	0.01	16
Hourglass 1	1	N/A	N/A	0.49	N/A	N/A	N/A
Hourglass 2	0.01	1.96	1.764	49	24.75	1	16
Bunny piling	1	N/A	N/A	N/A	N/A	N/A	N/A
Stamping bunny	1	N/A	N/A	1.225	N/A	N/A	N/A
Bunny melting 1	1	N/A	N/A	0	N/A	N/A	N/A
Bunny melting 2	1	N/A	N/A	39.2	N/A	N/A	N/A
3D printing	0.5	39.2	39.2	1.96	0	0	1

7.3 Jet Coiling and Buckling

We simulated the jet buckling of different materials using our generalized constitutive model, as described in Section 4. By varying the non-local effects, we can control the fluidity of the simulated matter. Figure 2 (top) shows that our method successfully reproduces the liquid rope coiling effect for classical Newtonian viscous liquids, such as honey, using our generalized viscous model with local viscosity ($F_v = 1$). In contrast to the Newtonian viscous liquid, our model also reproduces the vivid coiling behavior of viscoelastic liquids, such as shaving cream (see Figure 2 (middle) and (bottom)). Moreover, our video shows the *secondary coiling* phenomenon in the non-local coiling cases. In particular, the non-local viscous flow first performs a rapid coiling rope up to a great height, forming a fluid columnar structure whose length greatly exceeds the rope diameter. When the

height of the column exceeds a critical value, it collapses under its own weight, and the process then repeats itself with a well-defined period that greatly exceeds the coiling period [Habibi et al. 2006]. To further highlight differences due to non-local effects, we simulated a layer of viscous sheet folding onto a wafer, as shown in Figure 6. Our method captures the characteristic folding behaviors for different materials. The local model ($F_v = 1$) produces smooth merging and symmetric buckling patterns of liquids such as chocolate (see Figure 6 (top)). However, the non-local models $F_v \neq 1$ introduce elasticity which tends to cause larger folds and retains the change of shapes for fluids such as ice cream (see Figure 6 (bottom)).



Fig. 13. A closer view of die swell in the shear thinning liquid example from Figure 12. After exiting the throat of hourglass, the shear thinning stream swells (right), while the Newtonian liquid (left) does not. The average Δt during simulations is 10^{-5} s.

7.4 Die Swelling

To demonstrate the shear thinning behavior, we simulated the viscous liquid hourglass. Gravity forces extrude the viscous liquid through the narrow throat of the hourglass, as shown in Figure 12. Our non-local viscous stream (see Figure 13 (right)) demonstrates the appealing *die swell* [Tanner 1970] where the stream “swells” back to the former shape substantially over a period of time after the shear

stress has been removed, while the classical Newtonian viscous stream (which is local and non-elastic) does not show any swelling. This is a typical shear thinning phenomena since the viscosity decreases as the shear strain increases. Moreover, our generalized non-local viscosity can also exhibit Bingham-Plastic behavior [Zhu et al. 2015], showcasing the break-up of a fluid stream, such as ketchup (see here² for a real Bingham-Plastic flow).

7.5 Stamping Bunny

We first dropped eight copies of the viscous Stanford bunny with random orientations into a pile, as shown in Figure 4. Our MPM-GS4 method can easily handle the moving boundary problem while preserving fine details. Once bunnies are piled up, the bunny pile was stamped by a scripted moving SIGGRAPH label (see Figure 1(top right)). A standard no-slip condition applied to the discrete viscosity equations forces the fluid velocity to match that of the solid, causing the viscous liquid to be dragged alongside moving solids. During simulations, the average Δt is 5×10^{-6} s.

7.6 Melting Bunny

Next, we melt a bunny over a hot pan, showcasing our phase change computational framework. We simulated heat transfer and fluid dynamics simultaneously using our proposed MPM-GS4 scheme, as described in Section 6. The viscosity coefficient is defined as a function of the particle temperatures and heat diffusion is also affected by the dissipation due to viscous stress. Our careful treatment of

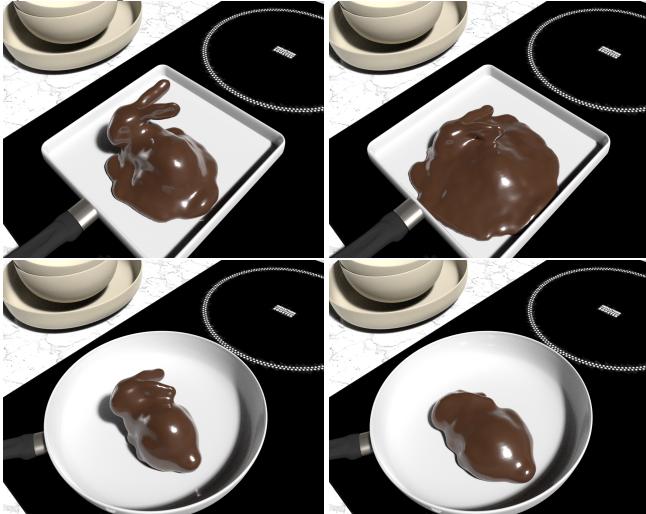


Fig. 14. Simulation of a chocolate bunny melting over a hot pan. An inviscid bunny (top) melts smoothly showing wave-like spread while a viscous bunny (bottom) melts slowly and gradually, exhibiting high viscosity at the end. The average Δt during simulations is 10^{-4} s.

phase transition allows our method to maintain sharp, yet stable, interfaces between the solid and fluid phases, as shown in Figure 14. We observe wave-like spread and micro ripples of the fluid phase in the inviscid bunny melting example (see Figure 14(top)), while the viscous bunny melts slowly, turning tacky and then liquid over a fairly broad temperature range (see Figure 14(bottom)).

²https://www.youtube.com/watch?v=X_cLJvUBLxw&t=31s&ab_channel=NathanPalmer

7.7 3D Printing

Our method can also be used to simulate realistic additive manufacturing, such as 3D printing. As shown in Figure 1(bottom right), a polymer filament is heated and deposited from a moving nozzle along a pre-designed path [Gajjar et al. 2017] of the SIGGRAPH logo. Our non-local viscous model captures the vivid viscoelastic behavior of polymer melts. The melting polymer filament solidifies gradually on the cold platform. Once a layer has been completed, the nozzle is raised, and another layer is laid down, thus building attractive layered shapes. After each layer completely solidifies, we drop the logo exhibiting its elasticity and solidity (see Figure 15). We also 3D printed the SIGGRAPH letters (in simulation) and dropped them on the printer bed, as shown in Figure 16. The letters exhibit different dynamic responses upon hitting the ground, highlighting that the material stiffness is a function of the printing process.

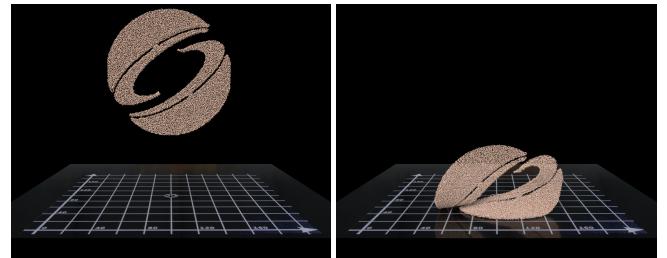


Fig. 15. Our method allows for realistic simulation of additive manufacturing. The 3D printed SIGGRAPH logo from Figure 1(bottom right) deforms when dropped on the printer bed, demonstrating its elasticity and solidity. The average Δt during simulations is 5×10^{-5} s.

8 CONCLUSION AND FUTURE WORK

We proposed a unified second order accurate in time MPM formulation for simulating viscous phenomena from the perspective of *non-local modeling*, which is a strict generalization of Newtonian viscosity. To simulate the proposed model on Cartesian MAC grids, we introduced the second-order accurate GS4 schemes [Tamma et al. 2000; Zhou and Tamma 2004] to computer graphics for greater accuracy. Our method easily integrates with MPM and phase change modeling, and allows for realistic simulation of 3D printing.

Our method presents a non-local perspective of the basic viscous phenomena in nature, which leads to the unification of different models. We demonstrated that our method captures a wide range of viscoelastic liquid behaviors, such as liquid rope coiling, buckling, shear thinning/thickening, melting/solidification processes and produces more detailed and realistic effects in practical applications than the classical Newtonian (local) model, such as die swelling effect and solid-like elasticity of a liquid droplet during impact.

Future work can explore many interesting avenues. A lot of our parameters were tuned by hand, and it would be interesting to calibrate them to measured models. We also neglected interactions with the air while modeling phase change, which are important for investigating the leathery behavior of polymers [Borisenkova et al. 1982]. It would also be interesting to extend our work by considering non-locality in both space *and* time. This would undoubtedly allow for a wider range of rheologies, such as *shear banding* [Lu et al. 2000]. Due to the high complexity of our unified constitutive model,

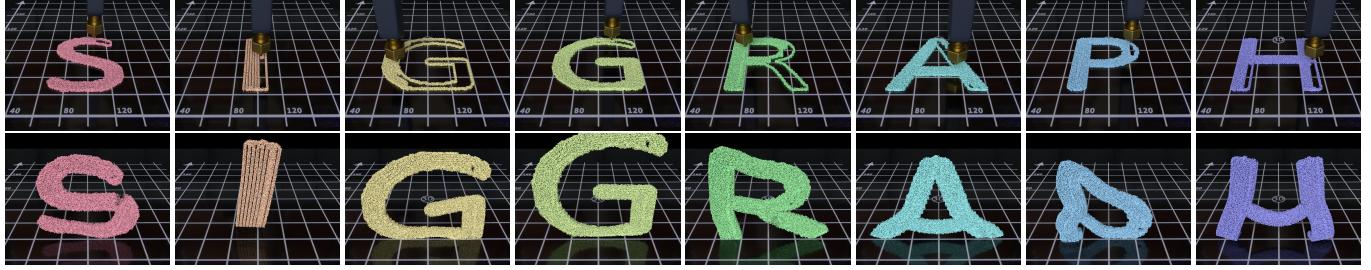


Fig. 16. (Top) The individual letters of SIGGRAPH were 3D printed (in simulation) using our method that provides a unified framework for thermo-viscoelastic effects. (Bottom) The letters deform differently when dropped on the printer bed, illustrating that the material stiffness depends on the fabrication process. The average Δt during simulations is 6×10^{-5} s.

we only briefly investigated a fully implicit treatment of the second order accurate solve in Section 5. While we did not experience a need for excessively small time steps given the low stiffness of the materials we considered, melting materials with high wave speed, such as steel, could benefit from a fully implicit discretization of phase change. Besides, another interesting direction for future work would be to investigate the influence of setting the parameter $F_v \neq 1$ in the non-Fourier solve in equation (51), which may be important for simulating practical phase change problems, such as micro-scale solidification during casting of copper. Finally, while our focus was on second-order temporal accuracy, it would be interesting to investigate second-order spatial accuracy, as in practice, this is much more crucial in 3D, and not self-evident for MPM approaches.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback. This research was supported in part by the Rutgers University start-up grant, NSF CAREER (IIS-1943199), CCF-1813624 and ECCS-2023780.

REFERENCES

- Mridul Aanjaneya, Chengguizi Han, Ryan Goldade, and Christopher Batty. 2019. An Efficient Geometric Multigrid Solver for Viscous Liquids. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–21.
- Stefan Band, Christoph Gissler, Andreas Peer, and Matthias Teschner. 2018. MLS Pressure Extrapolation for the Boundary Handling in Divergence-Free SPH. In *Proceedings of the 14th Workshop on Virtual Reality Interactions and Physical Simulations (VRIPHYS '18)*. Eurographics Association, 55–63.
- Héctor Barreiro, Ignacio García-Fernández, Iván Alduán, and Miguel A. Otaduy. 2017. Conformation Constraints for Efficient Viscoelastic Fluid Simulation. *ACM Trans. Graph.* 36, 6, Article 221 (2017), 11 pages.
- Cx K Batchelor and GK Batchelor. 2000. *An introduction to fluid dynamics*. Cambridge university press.
- Christopher Batty and Robert Bridson. 2008. Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids.. In *Symposium on Computer Animation*. 219–228.
- Christopher Batty and Ben Houston. 2011. A simple finite volume method for adaptive viscous liquids. In *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 (TOG)* 31, 4 (2012), 1–7.
- Jan Bender and Dan Koschier. 2017. Divergence-Free SPH for Incompressible and Viscous Fluids. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2017), 1193–1206.
- J. Bender, Tassilo Kugelstadt, M. Weiler, and Dan Koschier. 2020. Implicit Frictional Boundary Handling for SPH. *IEEE Transactions on Visualization and Computer Graphics* 26 (2020), 2982–2993.
- Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Trans. Graph.* 29, 4, Article 116 (2010), 10 pages.
- Volodymyr Valeriyovich Bilovol. 2003. Mould filling simulations during powder injection moulding. (2003).
- G Bishko, TCB McLeish, OG Harlen, and RG Larson. 1997. Theoretical molecular rheology of branched polymers in simple and complex flows: The pom-pom model. *Physical Review Letters* 79, 12 (1997), 2352.
- Javier Bonet and Richard D Wood. 1997. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press.
- EK Borisenkova, VE Dreval, GV Vinogradov, MK Kurbanaliev, VV Moiseyev, and VG Shagalanova. 1982. Transition of polymers from the fluid to the forced high-elastic and leathery states at temperatures above the glass transition temperature. *Polymer* 23, 1 (1982), 91–99.
- Robert Bridson. 2015. *Fluid simulation for computer graphics*. CRC Press.
- George E Brown, Matthew Overby, Zahra Forootaninia, and Rahul Narain. 2018. Accurate rate dissipative forces in optimization integrators. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14.
- Rachel Caiden, Ronald P Fedkiw, and Chris Anderson. 2001. A numerical method for two-phase flow consisting of separate compressible and incompressible regions. *J. Comput. Phys.* 166, 1 (2001), 1–27.
- 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.
- Carlo Cattaneo. 1948. Sulla conduzione del calore. *Atti Sem. Mat. Fis. Univ. Modena* 3 (1948), 83–101.
- CI Christov. 2009. On frame indifferent formulation of the Maxwell–Cattaneo model of finite-speed heat conduction. *Mechanics Research Communications* 36, 4 (2009), 481–486.
- Pascal Clausen, Martin Wicke, Jonathan R Shewchuk, and James F O'brien. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Transactions on Graphics (TOG)* 32, 2 (2013), 1–15.
- Nicolas Clemeur and Benoit Debaut. 2007. A pragmatic approach for deriving constitutive equations endowed with pom–pom attributes. *Rheologica acta* 46, 9 (2007), 1187–1196.
- 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.
- A Cemal Eringen. 1992. Vistas of nonlocal continuum physics. *International journal of engineering science* 30, 10 (1992), 1551–1565.
- A Cemal Eringen and DGB Edelen. 1972. On nonlocal elasticity. *International journal of engineering science* 10, 3 (1972), 233–248.
- 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.
- Adolph Fick. 1855. V. On liquid diffusion. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 10, 63 (1855), 30–39.
- Wilhelm Flügge. 2013. *Viscoelasticity*. Springer Science & Business Media.
- Jean-Baptiste Joseph Fourier. 1878. *Théorie analytique de la chaleur*. Paris 1822. *Engl. Übersetzung: "Theory of Heat Transfer"*, Cambridge (1878).
- Albrecht Fröhlich and R Sack. 1946. Theory of the rheological properties of dispersions. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 185, 1003 (1946), 415–430.
- Sumit Gajjar, Jaydeep Bhadani, Pramit Dutta, and Naveen Rastogi. 2017. Complete coverage path planning algorithm for known 2d environment. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 963–967.
- Ming Gao, Xinlei Wang, Kui Wu, Andre Pradhana, Eftychios Sifakis, Cem Yuksel, and Chenfanfu Jiang. 2018. GPU optimization of material point methods. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12.
- Tolga Goktekin, Adam W Bargteil, and James F O'Brien. 2004. A method for animating viscoelastic fluids. In *ACM SIGGRAPH 2004 Papers*. 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.

- Mehdi Habibi, Maniya Maleki, Ramin Golestanian, Neil M Ribe, and Daniel Bonn. 2006. Dynamics of liquid rope coiling. *Physical Review E* 74, 6 (2006), 066306.
- Francis H. Harlow and J. Eddie Welch. 1965. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids* 8, 12 (1965), 2182–2189.
- Kei Iwasaki, Hideyuki Uchida, Yoshinori Dobashi, and Tomoyuki Nishita. 2010. Fast particle-based visual simulation of ice melting. In *Computer graphics forum*, Vol. 29. Wiley Online Library, 2215–2223.
- Daniel D Joseph and Luigi Preziosi. 1989. Heat waves. *Reviews of Modern Physics* 61, 1 (1989), 41.
- J Murali Krishnan, Abhijit P Deshpande, and PB Sunil Kumar. 2010. *Rheology of complex fluids*. Springer.
- E Kröner. 1967. Elasticity theory of materials with long range cohesive forces. *International Journal of Solids and Structures* 3, 5 (1967), 731–742.
- 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.
- Ronald G Larson. 1999. *The structure and rheology of complex fluids*. Vol. 150. Oxford university press New York.
- C-Y David Lu, Peter D Olmsted, and RC Ball. 2000. Effects of non-local stress on the determination of shear banding flow. *Physical Review Letters* 84, 4 (2000), 642.
- 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.
- TCB McLeish and RG Larson. 1998. Molecular constitutive equations for a class of branched polymers: The pom-pom polymer. *Journal of Rheology* 42, 1 (1998), 81–110.
- Matthias Müller, Simon Schirrm, and Matthias Teschner. 2004. Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. *Technology and Health Care* 12, 1 (2004), 25–31.
- Meraj Mustafa. 2015. Cattaneo-Christov heat flux model for rotating flow and heat transfer of upper-convected Maxwell fluid. *Aip Advances* 5, 4 (2015), 047109.
- Kentaro Nagasawa, Takayuki Suzuki, Ryohei Seto, Masato Okada, and Yonghao Yue. 2019. Mixing sauces: A viscosity blending model for shear thinning fluids. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–17.
- 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*, 137–146.
- Raymond W Ogden. 1997. *Non-linear elastic deformations*. Courier Corporation.
- CM Oishi, FP Martins, MF Tomé, and MA Alves. 2012. Numerical simulation of drop impact and jet buckling problems using the eXtended Pom-Pom model. *Journal of Non-Newtonian Fluid Mechanics* 169 (2012), 91–103.
- CM Oishi, FP Martins, Murilo F Tome, JA Cuminato, and Sean McKee. 2011. Numerical solution of the eXtended Pom-Pom model for viscoelastic free surface flows. *Journal of Non-Newtonian Fluid Mechanics* 166, 3–4 (2011), 165–179.
- James G Oldroyd. 1950. On the formulation of rheological equations of state. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 200, 1063 (1950), 523–541.
- Afonso Paiva, Fabiano Petronetto, Thomas Lewiner, and Geovan Tavares. 2009. Particle-based viscoplastic fluid/solid simulation. *Computer-Aided Design* 41, 4 (2009), 306–314.
- Peyman Pakdel and Gareth H McKinley. 1996. Elastic instability and curved streamlines. *Physical Review Letters* 77, 12 (1996), 2459.
- Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. 2015. An implicit viscosity formulation for SPH fluids. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.
- 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.
- Kayan Sadeghy, Amir-Hosain Najafi, and Meghdad Saffaripour. 2005. Sakiadis flow of an upper-convected Maxwell fluid. *International Journal of Non-Linear Mechanics* 40, 9 (2005), 1220–1228.
- Mohamed Shaat, Esmaeal Ghavanloo, and S Ahmad Fazelzadeh. 2020. Review on nonlocal continuum mechanics: physics, material applicability, and mathematics. *Mechanics of Materials* (2020), 103587.
- M Shimada and Kumar K Tamma. 2012. Conserving/dissipative algorithms and designs for a system of N particles: Total energy framework and single-field form. *Computers & structures* 112 (2012), 380–405.
- SL Sobolev. 2014. Nonlocal diffusion models: Application to rapid solidification of binary mixtures. *International Journal of Heat and Mass Transfer* 71 (2014), 295–302.
- Arun R Srinivasa and JN Reddy. 2017. An overview of theories of continuum mechanics with nonlocal elastic response and a general framework for conservative and dissipative systems. *Applied Mechanics Reviews* 69, 3 (2017).
- Jos Stam. 1999. Stable Fluids. In *Proc. of ACM SIGGRAPH (SIGGRAPH '99)*, 121–128.
- 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.
- Dan Stora, Pierre-Olivier Agliati, Marie-Paule Cani, Fabrice Neyret, and Jean-Dominique Gascuel. 1999. Animating lava flows.
- 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).
- Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C Lin. 2015. Implicit formulation for SPH-based viscous fluids. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 493–502.
- T. Takahashi and M. Lin. 2019. A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling. *Computer Graphics Forum* 38 (2019).
- Kumar K Tamma, X Zhou, and D Sha. 2000. The time dimension: a theory towards the evolution, classification, characterization and design of computational algorithms for transient/dynamic applications. *Archives of Computational Methods in Engineering* 7, 2 (2000), 67–290.
- RI Tanner. 1970. A theory of die-swell. *Journal of Polymer Science Part A-2: Polymer Physics* 8, 12 (1970), 2067–2078.
- Demetris Terzopoulos and Kurt Fleischer. 1988. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, 269–278.
- Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schuller. 2001. *Multigrid*. Academic Press.
- BM Tymrak, Megan Kreiger, and Joshua M Pearce. 2014. Mechanical properties of components fabricated with open-source 3-D printers under realistic environmental conditions. *Materials & Design* 58 (2014), 242–246.
- Da Yu Tzou. 2014. *Macro-to microscale heat transfer: the lagging behavior*. John Wiley & Sons.
- Wilco MH Verbeeten, Gerrit WM Peters, and Frank PT Baaijens. 2001. Differential constitutive equations for polymer melts: The extended Pom-Pom model. *Journal of Rheology* 45, 4 (2001), 823–843.
- Vaughan R Voller and CR Swaminathan. 1991. ERAL Source-based method for solidification phase change. *Numerical Heat Transfer, Part B Fundamentals* 19, 2 (1991), 175–189.
- Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum (Eurographics)* 37, 2 (2018).
- Chris Wojtan and Greg Turk. 2008. Fast viscoelastic behavior with thin features. In *ACM SIGGRAPH 2008 papers*, 1–8.
- 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.
- Tao Xue, Kumar K Tamma, and Xiaobing Zhang. 2016. A consistent moving particle system simulation method: applications to parabolic/hyperbolic heat conduction type problems. *International Journal of Heat and Mass Transfer* 101 (2016), 365–372.
- Tao Xue, Xiaobing Zhang, and Kumar K Tamma. 2018. Generalized heat conduction model involving imperfect thermal contact surface: application of the GSSSS-1 differential-algebraic equation time integration. *International Journal of Heat and Mass Transfer* 116 (2018), 889–896.
- 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.
- X Zhou and Kumar K Tamma. 2004. Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *Internat. J. Numer. Methods Engrg.* 59, 5 (2004), 597–668.
- Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015. Codimensional non-Newtonian fluids. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

A ALGORITHMIC PARAMETERS IN GS4 INTEGRATOR

$$\begin{aligned} W_1\Lambda_6 &= \frac{3 + \rho_\infty + \rho_\infty^S - \rho_\infty\rho_\infty^S}{2(1 + \rho_\infty)(1 + \rho_\infty^S)}, & W_1 &= \frac{1}{1 + \rho_\infty}, & W_1\Lambda_4 &= \frac{1}{1 + \rho_\infty}, \\ W_2\Lambda_5 &= \frac{1}{(1 + \rho_\infty)(1 + \rho_\infty^S)}, & W_1\Lambda_1 &= W_1\Lambda_4, & W_2\Lambda_2 &= \frac{W_1\Lambda_4}{2}, \\ W_3\Lambda_3 &= \frac{W_2\Lambda_5}{2}, & \lambda_5 &= \frac{1}{1 + \rho_\infty^S}, & \lambda_4 &= 1, & \lambda_3 &= \frac{1}{\lambda_5}, & \lambda_2 &= \frac{1}{2}, & \lambda_1 &= 1, \end{aligned} \quad (46)$$

where $(\rho_\infty^S \leq \rho_\infty)$ are the principal and spurious roots at the high frequency limit satisfying $0 \leq \rho_\infty^S \leq \rho_\infty \leq 1$. All algorithms in the GS4 framework are second-order accurate, unconditionally stable, zero-order overshoot, and controllable numerically dissipative, with

the option of the selective control feature, which is a practically useful new feature to yield physically representative solutions of both variables. The existing and new algorithms contained in this framework is given in Table 5.

Table 5. The new and existing algorithms contained in the GS4 framework

No.	Parameters	Algorithms
1	$\rho_\infty = \rho_s^\infty = 1$	Crank-Nicolson method
2	$\rho_\infty = \rho_s^\infty = 0$	Gear's method/BDF2
3	$\rho_\infty = \rho_s^\infty$	Existing algorithms without selective control feature
4	$\rho_\infty \neq \rho_s^\infty$	New algorithms with selective control feature

B SECOND-ORDER GS4 NON-FOURIER SOLVER

Follow the GS-4 framework, we now revise the Non-Fourier solver in Section 5, we first explicitly update the convection terms

$$\begin{aligned} (\dot{T}_n + W_1 \Lambda_6 \Delta \dot{T}_*) + \mathbf{u}_n \cdot \nabla T_n &= W_n, \\ (\dot{\mathbf{q}}_n^{NL} + W_1 \Lambda_6 \Delta \dot{\mathbf{q}}_*^{NL}) + \mathbf{u}_n \cdot \nabla \mathbf{q}_n^{NL} &= 0, \end{aligned} \quad (47)$$

and solving for $\Delta \dot{T}_*$ and $\Delta \dot{\mathbf{q}}_*^C$ enables us to get

$$\begin{aligned} \tilde{T}_* &= \dot{T}_n + W_1 \Lambda_6 \Delta \dot{T}_*, \quad \tilde{T}_* = T_n + W_1 \Lambda_4 \dot{T}_n \Delta t + W_2 \Lambda_5 \eta \Delta \dot{T}_* \Delta t, \\ \tilde{\mathbf{q}}_* &= \dot{\mathbf{q}}_n^{NL} + W_1 \Lambda_6 \Delta \dot{\mathbf{q}}_*^{NL}, \quad \tilde{\mathbf{q}}_*^L = -F_T k \nabla \tilde{T}_*, \\ \tilde{\mathbf{q}}_*^{NL} &= \mathbf{q}_n^{NL} + W_1 \Lambda_4 \dot{\mathbf{q}}_n^{NL} \Delta t + W_2 \Lambda_5 \eta \Delta \dot{\mathbf{q}}_*^{NL} \Delta t, \end{aligned} \quad (48)$$

followed by an implicit projection solve:

$$\begin{aligned} (\tilde{T}_* + W_1 \Lambda_6 \Delta \dot{T}_{**}) + \nabla \cdot (\tilde{\mathbf{q}}_*^{NL} + W_2 \Lambda_5 \eta \Delta \dot{\mathbf{q}}_*^{NL} \Delta t + \tilde{\mathbf{q}}_{**}^L) &= 0, \\ \tau \left(\dot{\tilde{\mathbf{q}}}_*^{NL} + W_1 \Lambda_6 \Delta \dot{\mathbf{q}}_{**}^{NL} \right) + \left(\tilde{\mathbf{q}}_*^{NL} + \eta W_2 \Lambda_5 \Delta t \Delta \dot{\mathbf{q}}_{**}^{NL} \right) \\ + (1 - F_T) k \nabla \left(\tilde{T}_* + \eta W_2 \Lambda_5 \Delta t \Delta \dot{T}_{**} \right) &= 0, \end{aligned} \quad (49)$$

where $\tilde{\mathbf{q}}_{**}^L = -F_T k \nabla \left(\tilde{T}_* + \eta W_2 \Lambda_5 \Delta t \Delta \dot{T}_{**} \right)$. Taking the divergence of equation (49)₂ gives us the followings equations:

$$\begin{aligned} &\left(W_1 \Lambda_6 - \eta W_2 \Lambda_5 \Delta t F_T k \nabla^2 \right) \Delta \dot{T}_{**} + \eta W_2 \Lambda_5 \Delta t \nabla \cdot (\Delta \dot{\mathbf{q}}_{**}^{NL}) \\ &= (F_T k \nabla^2 - 1) \tilde{T}_* - \nabla \cdot \tilde{\mathbf{q}}_*^{NL}, \\ &(\eta W_2 \Lambda_5 \Delta t + W_1 \Lambda_6 \tau) \nabla \cdot (\Delta \dot{\mathbf{q}}_{**}^{NL}) + \eta W_2 \Lambda_5 \Delta t (1 - F_T) k \nabla^2 (\Delta \dot{T}_{**}) \\ &= -\tau \nabla \cdot \dot{\tilde{\mathbf{q}}}_*^{NL} - \nabla \cdot \tilde{\mathbf{q}}_*^{NL} - (1 - F_T) k \nabla^2 \tilde{T}_*. \end{aligned} \quad (50)$$

After mathematical manipulations, we have the following expression for $\Delta \dot{T}_{**}$:

$$\begin{aligned} \left[W_1 \Lambda_6 \mathbf{I} - \frac{\Delta t \eta W_2 \Lambda_5 (F_T \tau W_1 \Lambda_6 + \Delta t \eta W_2 \Lambda_5)}{(\eta W_2 \Lambda_5 \Delta t + W_1 \Lambda_6 \tau)} k \nabla^2 \right] \Delta \dot{T}_{**} = \\ (F_T k \nabla^2 - \mathbf{I}) \tilde{T}_* - \nabla \cdot \tilde{\mathbf{q}}_*^{NL} + \frac{\tau \nabla \cdot \dot{\tilde{\mathbf{q}}}_*^{NL} + \nabla \cdot \tilde{\mathbf{q}}_*^{NL} + (1 - F_T) k \nabla^2 \tilde{T}_*}{(\eta W_2 \Lambda_5 \Delta t + W_1 \Lambda_6 \tau)}, \end{aligned} \quad (51)$$

which can be efficiently solved with PCG. After solving for $\Delta \dot{T}_{**}$, we compute $\Delta \dot{\mathbf{q}}_{**}^{NL}$ based on the second equation in (49).

$$\Delta \dot{\mathbf{q}}_{**}^{NL} = -\frac{\tau \dot{\tilde{\mathbf{q}}}_*^{NL} + (1 - F_T) k \nabla \tilde{T}_{**}}{(\eta W_2 \Lambda_5 \Delta t + W_1 \Lambda_6 \tau)}, \quad (52)$$

where $\tilde{T}_{**} = \tilde{T}_* + \eta W_2 \Lambda_5 \Delta t \Delta \dot{T}_{**}$.

Finally, we update T and $\dot{\mathbf{q}}^{NL}$ at $t = t_{n+1}$ as follows:

$$\begin{aligned} \Delta \dot{T} &= \Delta \dot{T}_* + \Delta \dot{T}_{**}, \quad \Delta \dot{\mathbf{q}}^{NL} = \Delta \dot{\mathbf{q}}_*^{NL} + \Delta \dot{\mathbf{q}}_{**}^{NL}, \\ \dot{T}_{n+1} &= \dot{T}_n + \Delta \dot{T}, \quad \dot{\mathbf{q}}_{n+1}^{NL} = \dot{\mathbf{q}}_n^{NL} + \Delta \dot{\mathbf{q}}^{NL}, \\ T_{n+1} &= T_n + \lambda_4 \Delta t \dot{T}_n + \lambda_5 \Delta t \Delta \dot{T}, \\ \mathbf{q}_{n+1}^{NL} &= \mathbf{q}_{n+1}^{NL} + \lambda_4 \Delta t \dot{\mathbf{q}}_n^{NL} + \lambda_5 \Delta t \Delta \dot{\mathbf{q}}^{NL}. \end{aligned} \quad (53)$$

C SCHUR COMPLEMENTS COMPUTATION

C.1 Semi-implicit Backward Euler Scheme

Eliminating velocity in equation (17) by applying a Schur-complement to the center block yields a smaller SPD linear system as follows:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{13}^T & \mathbf{A}_{23}^T & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma}^L \\ \boldsymbol{\sigma}^{NL} \\ p \end{bmatrix}_{n+1} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \mathbf{B}_3 \end{bmatrix} \quad (54)$$

where

$$\begin{aligned} \mathbf{A}_{11} &= \frac{1}{2F_v \Delta t} \mathbf{I} + \mathbf{D}\mathbf{D}^T, \quad \mathbf{A}_{12} = \mathbf{D}\mathbf{D}^T, \quad \mathbf{A}_{13} = \mathbf{D}\mathbf{G}, \\ \mathbf{A}_{22} &= \mathbf{D}\mathbf{D}^T + \frac{(\tau_v + \Delta t)}{2(1 - F_v) \Delta t^2} \mathbf{I}, \quad \mathbf{A}_{23} = \mathbf{D}\mathbf{G}, \quad \mathbf{A}_{33} = \mathbf{G}^T \mathbf{G}, \\ \mathbf{B}_1 &= \mathbf{D}\mathbf{b} + \frac{\rho}{\Delta t} \mathbf{D}\mathbf{u}_*, \quad \mathbf{B}_2 = \frac{\tau_v}{2(1 - F_v) \Delta t^2} \boldsymbol{\sigma}_*^{NL} + \mathbf{D}\mathbf{b} + \frac{\rho}{\Delta t} \mathbf{D}\mathbf{u}_*, \\ \mathbf{B}_3 &= \mathbf{G}^T \mathbf{b} + \frac{\rho}{\Delta t} \mathbf{G}^T \mathbf{u}_*. \end{aligned}$$

C.2 Second Order Semi-Implicit GS4 Time Integration

Eliminating velocity $\Delta \dot{\mathbf{u}}_{**}$ from equation (21) by applying a Schur-complement yields a smaller SPD linear system as follows:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{13}^T & \mathbf{A}_{23}^T & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\sigma}_{**}^L \\ \Delta \dot{\boldsymbol{\sigma}}_{**}^{NL} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \mathbf{B}_3 \end{bmatrix} \quad (55)$$

where

$$\begin{aligned} \mathbf{A}_{11} &= \frac{\rho W_1 \Lambda_6 W_1^2 \eta}{2F_v W_2 \Lambda_5 \Delta t} \mathbf{I} + (\eta W_1)^2 \mathbf{D}\mathbf{D}^T, \quad \mathbf{A}_{12} = \eta^2 W_1 W_2 \Lambda_5 \Delta t \mathbf{D}\mathbf{D}^T, \\ \mathbf{A}_{22} &= (\Delta t \eta W_2 \Lambda_5)^2 \mathbf{D}\mathbf{D}^T + \frac{\rho W_1 \Lambda_6 (W_1 \Lambda_6 + \Delta t \eta W_2 \Lambda_5)}{2(1 - F_v)} \mathbf{I}, \\ \mathbf{A}_{13} &= \eta W_1 \mathbf{D}\mathbf{G}, \quad \mathbf{A}_{23} = \Delta t \eta W_2 \Lambda_5 \mathbf{D}\mathbf{G}, \quad \mathbf{A}_{33} = \mathbf{G}^T \mathbf{G}, \\ \mathbf{B}_1 &= -\eta W_1 (\mathbf{D}\mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^L + \mathbf{D}\mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^{NL} + \rho \mathbf{D}\tilde{\mathbf{u}}_*) - \frac{\rho W_1 W_1 \Lambda_6}{2F_v \Delta t W_2 \Lambda_5} (\tilde{\boldsymbol{\sigma}}_*^L - 2F_v \mu \mathbf{D}\tilde{\mathbf{u}}_*), \\ \mathbf{B}_2 &= \frac{\rho W_1 \Lambda_6 (2\mu(F_v - 1) \mathbf{D}\tilde{\mathbf{u}}_* + \tilde{\boldsymbol{\sigma}}_*^{NL} + \tau \tilde{\boldsymbol{\sigma}}_*^{NL})}{2(F_v - 1)} \\ &- \eta \Delta t W_2 \Lambda_5 (\mathbf{D}\mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^L + \mathbf{D}\mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^{NL} + \rho \mathbf{D}\tilde{\mathbf{u}}_*), \\ \mathbf{B}_3 &= \frac{\rho W_1 \Lambda_6 \mathbf{G}^T \tilde{\mathbf{u}}_*}{\eta \Delta t W_2 \Lambda_5} - (\mathbf{G}\mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^L + \mathbf{G}\mathbf{D}^T \tilde{\boldsymbol{\sigma}}_*^{NL} + \rho \mathbf{G}\tilde{\mathbf{u}}_*). \end{aligned}$$

D PRESSURE EVALUATION

We use the Tait's equation (equation (27)) for pressure evaluation. The proposed GS4 time integration is also used to discretize the following continuity equation for the evaluation of density:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0. \quad (56)$$

Following the GS4 framework, we can formulate the solve for ρ as:

$$(W_1 \Lambda_6 \Delta \dot{\rho} + W_2 \Lambda_5 \Delta t \nabla \cdot \tilde{\mathbf{u}}) \Delta \dot{\rho} = -\dot{\rho}_n - (\rho_n + W_1 \Lambda_4 \dot{\rho} \Delta t) \nabla \cdot \tilde{\mathbf{u}}, \quad (57)$$

where $\tilde{\mathbf{u}} = \mathbf{u}_n + W_1 \Lambda_4 \Delta t \mathbf{u}_n$ and solving for $\Delta \dot{\rho}$ yields:

$$\dot{\rho}_{n+1} = \dot{\rho}_n + \Delta \dot{\rho}, \quad \rho_{n+1} = \rho_n + \lambda_4 \Delta t \dot{\rho}_n + \lambda_5 \Delta t \Delta \dot{\rho}. \quad (58)$$