

# A Unified Particle-Based Solver for Non-Newtonian Behaviors Simulation

Chunlei Li, Yang Gao , Jiayi He, Tianwei Cheng, Shuai Li, Aimin Hao, Hong Qin, *Member, IEEE*

**Abstract**—In this paper, we present a unified framework to simulate non-Newtonian behaviors. We combine viscous and elasto-plastic stress into a unified particle solver to achieve various non-Newtonian behaviors ranging from fluid-like to solid-like. Our constitutive model is based on a Generalized Maxwell model, which incorporates viscosity, elasticity and plasticity in one non-linear framework by a unified way. On the one hand, taking advantage of the viscous term, we construct a series of strain-rate dependent models for classical non-Newtonian behaviors such as shear-thickening, shear-thinning, Bingham plastic, etc. On the other hand, benefiting from the elasto-plastic model, we empower our framework with the ability to simulate solid-like non-Newtonian behaviors, i.e., visco-elasticity/plasticity. In addition, we enrich our method with a heat diffusion model to make our method flexible in simulating phase change. Through sufficient experiments, we demonstrate a wide range of non-Newtonian behaviors ranging from viscous fluid to deformable objects. We believe this non-Newtonian model will enhance the realism of physically-based animation, which has great potential for computer graphics.

**Index Terms**—Physically based animation, non-Newtonian material, SPH, viscous fluid, deformable solid.

## 1 INTRODUCTION

From melting ice cream and ketchup to human blood, non-Newtonian materials are everywhere in our daily lives. In contrast to Newtonian fluids, non-Newtonian fluids display a non-linear relationship between stress and strain rate, making them versatile and exhibiting variable properties that are different from those of traditional solids and fluids. In fact, non-Newtonian materials can be seen as materials in-between solids and fluids and therefore possess characteristics of both. By simulating these interesting non-Newtonian behaviors, we can create better physically-based animations for computer graphics applications.

Traditionally, there are two routes to simulate non-Newtonian materials in existing research. The first approach is to use a highly viscous fluid to mimic solid-like behaviour. This type of work dates back to Stora [1]. The key to success in this approach is to design a stable viscosity solver that can withstand drastic changes in viscosity [2], [3]. However, these methods are limited by viscous stress. Regardless of how large the viscous stress is, it solely influences the deformation rate and not the degree of deformation, meaning that elastic effects are not expected. Therefore, the non-Newtonian material produced by this method may not truly be “solid-like”. The second approach is the opposite, simulating materials from the elasticity theory and making them fluid-like. Recent works in Material Point Methods (MPM) fall under this category [4], [5]. In comparison to the traditional

Finite Element Method (FEM), MPM allows for the movement of Gaussian integral points, enabling the material to flow and giving it the ability to simulate various materials with complex, variable properties, including non-Newtonian materials. However, MPM faces a numerical adhesion problem. For low-resolution simulations with a small number of grid cells, the material may adhere together. This is because all information needs to be transferred to the grid to calculate the stress and update the deformation gradient, which limits the grid resolution and thus the resolution of the entire simulation. The material points belonging to different objects will be gathered into the same grid point and will never split again. Increasing the grid resolution will alleviate the problem, but the increase in complexity will limit the computational efficiency of the simulation.

In this research, we aim to effectively simulate non-Newtonian materials with physical properties that range from viscous fluids to deformable solids. We introduce the physics-based elastic stress to the Smoothed Particle Hydrodynamics (SPH) solver [6], thus combining the fluid-like resistance and the solid-like resistance within a unified framework. A notable merit of this practice compared to current viscous SPH methods is that our model is fully resistant to any non-Newtonian deformation. Moreover, we empower our method with the capability of simulating various strain rate-dependent materials, including shear-thinning, shear-thickening, Bingham plastic, and more. By utilizing our unified framework, which combines an elasto-plastic solver with a viscosity solver, we are able to achieve sufficient and realistic non-Newtonian phenomena.

Meanwhile, we also implement the diffusion equation to simulate the heat-based phase change. The pipeline is shown in Fig. 1. To summarise, our salient contributions are:

- C. Li, Y. Gao, J. He, T. Cheng, S. Li and A. Hao are with the State Key Laboratory of Virtual Reality Technology and Systems, and the Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing 100191, China, and also with the Research Unit of Virtual Body and Virtual Surgery (2019RU004), Chinese Academy of Medical Sciences, Beijing 100050, China.
  - S. Li and A. Hao are with Peng Cheng Laboratory, Shenzhen 518066, China.
  - S. Li is also with Zhongguancun Laboratory, Beijing 100094, China.
  - H. Qin is with the Department of Computer Science, Stony Brook University (SUNY at Stony Brook), New York 11794-2424, USA. USA
- We put the elastic stress and viscous stress together into the Navier-Stokes equation to develop a unified particle-based solver for simulating from fluid-like to solid-like non-Newtonian behaviors.

Corresponding author: gaoyangvr@buaa.edu.cn, qin@cs.stonybrook.edu.

- We implement strain rate dependent non-Newtonian models, and thus many classical non-Newtonian models can be reproduced in a flexible and physically meaningful way.
- We combine the diffusion model as well as the visco-elastic and visco-plastic coupled models into our framework using a Generalized Maxwell Model, thus further enriching the reality and the variety of non-Newtonian phenomena.

## 2 RELATED WORKS

### 2.1 Variable Viscosity for Non-Newtonian Material

One of the key characteristics of non-Newtonian material is that it has variable viscosity. Variable viscosity is a long-discussed issue in computer graphics. Stora et al. [1] pioneered in this field by simulating the lava with a high and variable viscosity in 1999. Carlson et al. [7] enhanced the numerical stability of algorithms under high viscosity for simulating phenomena, such as wax melting. Batty et al. [8] developed a variational principle-based Eulerian fluid solver that focuses on the accuracy of complex boundary conditions of viscous flow. Larionov et al. [9] designed a Stokes flow solver that is unconditionally stable when simulating highly viscous phenomena like buckling and coiling. Goldade et al. [3] designed an adaptive variational finite difference framework for octrees that is capable of simulating variable viscosity. Shao et al. [10] developed a high-resolution algebraic multigrid viscosity solver capable of simulating fluids with variable viscosity.

The above-mentioned research works are based on the Eulerian viewpoint or hybrid viewpoint. Additionally, there are several studies that utilize the Lagrangian solver [11], [12]. The SPH method can be traced back to Monaghan [13]. Muller et al. [14] introduced an equation-of-state based approach to enforce the incompressibility. Zhu et al. [15] put forward a codimensional non-Newtonian viscosity fluid simulator based on Fluid Implicit Particle (FLIP), where Carreau-Yasuda model is used to calculate shear thinning and shear thickening. Muller's model was further developed into a Weakly Compressible SPH (WCSPH) by Becker et al. [16]. Subsequently, numerous variants of SPH have been introduced to address incompressibility with greater efficiency, including the Predictive–Corrective Incompressible SPH (PCISPH) [17], Implicit Incompressible SPH (IISPH) [18], and Divergence-Free SPH (DFSPH) [19]. As for the viscosity in SPH, Koschier et al. [20] reviewed the recent works among the SPH viscous solvers. The XSPH [13], [21] is an explicit solver that takes advantage of the fact that viscosity is caused by the difference in velocities of fluid blobs. Takahashi et al. [22] used a second-ring neighbourhood to tackle the inconsistency issue of second-order derivatives of SPH. Peer et al. [2] decomposed the velocity gradient and reassembled it with a limited strain rate tensor to mimic the viscosity behavior. Similarly, Peer and Teschner [23] used the prescribed velocity gradient to simulate the viscous fluid. Bender and Koschier [19] constrained the strain rate by a user-defined parameter to simulate viscosity. And Weiler et al. [24] solved the Laplacian formula in an implicit scheme, taking into account the boundary particles and treating them with their conjugate gradient solver. Andrade et al. [25] utilized a SPH based method to simulate the non-Newtonian viscosity with Cross model and accelerated the performance with CUDA.

### 2.2 Visco-elasticity and Visco-plasticity

Another key characteristic of non-Newtonian material is the visco-elasticity and visco-plasticity. These materials possess properties

of both the solid and liquid states [26]. The study of visco-elastic and visco-plastic materials has long been a topic of interest among researchers. The interest in these materials can be traced back to Terzopoulos et al. in 1989 [27], when Terzopoulos simulated visco-elastic material by manipulating the integration point of FEM. Losasso et al. [28] tracked the level set of different materials, including solid, so their solver is capable of simulating multiple materials with distinct and variable physical properties. Similarly, Fujisawa et al. [29] adopted the Volume of Fluid method to track the boundary when simulating the ice melting. The concepts of viscosity and elasticity can be unified in terms of stress. Observing from the outcome, they both produce a limited degree of stress and hence inhibit mobility. Viscosity takes strain rate as input, whereas elasticity takes strain as input. Goktekin et al. [30] took this thought further by incorporating the elastic stress into the Navier-Stokes equations in order to simulate viscoelasticity. They used this method to simulate the jelly in the Marker And Cell (MAC) framework. Muller et al. [11] tackled the unified particle system to simulate elastic, plastic, and melting objects. They calculated the stress based on the traditional FEM but discretized it on particles. Becker et al. [31] followed this idea and implemented elasticity in a SPH framework.

Meanwhile, plasticity and elasticity are inseparable parts of Newtonian behaviors. O'Brien et al. [32] provided a solution to simulate the effect of plasticity. When the elasticity strain (or stress) exceeds a criterion, it will be constrained by the plastic law. Such a criterion is von Mises criterion in their work, and the limit is the elastic limit. A further increasing elastic strain will encounter another limit, the plastic limit. The object will fracture if the plastic limit is exceeded. A significant distinction between elasticity and plasticity is that plasticity is a history-dependent quantity that requires updating and accumulation at each time step. Muller et al. [11] and Solenthaler et al. [12] adopted this method to simulate plasticity in the particle method. Another common practise to treat plasticity is to separate the deformation gradient into two components, the elastic component and the plastic component. This practise is adopted by Stomakhin et al. [33] during the simulation of snow, where the snow shows both elasticity and plasticity. Similar to O'Brien's work, there are two thresholds for elasticity and plasticity. Yue et al. [34] used MPM to simulate the foam, which is a typical Bingham type non-Newtonian material. Their constitutive correlation is based on the Herschel-Bulkley model. More related works of MPM [5], [35]–[37] can be classified into this type. Gissler et al. [38] simulated the snow with SPH. Their work refers to the visco-elastic model developed by Stomakhin et al. [4] Bargteil et al. [39] used FEM to simulate the visco-plastic flow. To tackle the potential problem of the ill-conditioned mesh after large deformation, they used a re-mesh procedure. Wojtan and Turk [40] took a similar strategy with Bargteil et al. [39] to simulate the visco-elastic behavior. They carefully distinguished the situation for remeshing. With the spatial adaptivity technique, their algorithm achieves an order-of-magnitude speedup. Ozgen et al. [41] simulated the behavior of shear thickening fluids with SPH model. The key factor in their method is to use a spring to capture the nonlinearity of the non-Newtonian behavior.

For computer graphics applications, the key to depicting the visual effect of non-Newtonian behaviors is to demonstrate the material's dynamic characteristics, which can range from fluid-like to solid-like. Therefore, a simulation algorithm must take into account the following key features: 1) variable viscosity (incorporating classical shear-related models), 2) the inclusion

of elasticity, and 3) the integration of plasticity.

Existing research seldom covers all of the above-mentioned research issues. Early works, such as Carlson et al. [7] and Stora et al. [1], considered classical shear-related variable viscosity models. The work of Zhu et al. [15] is among the few papers dedicated to simulating non-Newtonian behaviors, taking into account variable viscosity and elasticity. However, it only offers one shear-related variable viscosity model and lacks plasticity simulation. In recent years, some works using the MPM, such as Su et al. [5] and Fang et al. [36], claim to be able to simulate non-Newtonian behaviors. However, their methods primarily focus on viscoelastic behavior. Solenthaler et al. [12] proposed a unified SPH-based approach that could simulate both fluids and solids, but it did not specifically address viscoelastic materials that lie between fluids and solids. Goktekin et al. [30] simulated semi-fluid, semi-solid materials by adding elastic stress in the momentum equation, which offered us some inspiration. However, we haven't seen specialized research that encompasses the majority of non-Newtonian behaviors, despite the fact that non-Newtonian behaviors are widely observed in real-world phenomena. As such, we believe that consolidating the three aforementioned behavioral characteristics (viscous, elastic, and plastic) into a unified framework will represent a significant contribution to the computer graphics community, at least at the application level.

Thus we design a flexible and versatile solver to incorporate all characteristics that simulating non-Newtonian behaviors required into one unified framework. With adjustable parameter tweaking, it is hopeful to develop more diverse strain rate-driven, strain-driven, and temperature-driven phenomena than with prior studies.

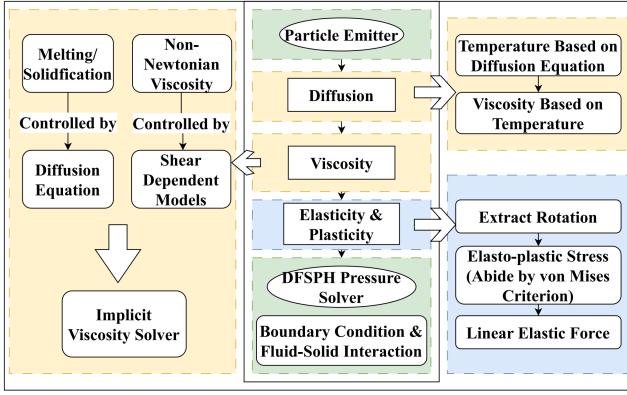


Fig. 1. The pipeline of our unified framework, where the yellow background represents related to viscous model, blue background represents related to elasto-plastic model, and green background represents others.

### 3 GOVERNING EQUATIONS AND GENERALIZED MAXWELL MODEL

We present our momentum balance equation by Eq. 1. The primary advancement lies in the modification of the original momentum equation through the inclusion of the elasto-plastic stress  $\sigma_{ep}$  within the equation, drawing inspiration from the work of Goktekin et al. [30].

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho} \nabla p + \nabla \cdot (\sigma_v + \sigma_{ep}) + \frac{\mathbf{f}_{ext}}{\rho}, \quad (1)$$

where  $\sigma_v$  represents viscous stress,  $p$  is the pressure,  $\rho$  is the density,  $\mathbf{f}_{ext}$  is the external force,  $\sigma_{ep}$  is the elasto-plastic stress.

There are various methods to discretize and solve Eq. 1. In this work, we adopt the DFSPH [19] for our particle-based solver, which is the most advanced SPH solver to this date [20].

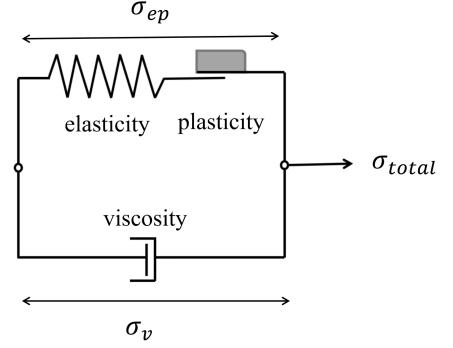


Fig. 2. Generalized Maxwell model to describe one-dimensional stress-strain (strain rate) relation.

To solve Eq. 1, we need to focus on the computation of  $\sigma_v$ ,  $\sigma_{ep}$  and their relation.

We present a Generalized Maxwell Model (GMM) to describe the constitutive relation underlying our unified non-Newtonian framework. Our inspiration is from the similar model from Fang et al. [36] for viscoelasticity simulation. The Generalized Maxwell Model can be conceptualized as a combination of a spring, a slider, and a dashpot, arranged both in series and parallel. This model comprises two branches: the upper branch features a spring and a slider in series, while the lower branch is composed solely of a dashpot. Each component of the model symbolizes a distinct physical property: the spring represents elasticity, the slider embodies plasticity, and the dashpot signifies viscosity.

Assuming that a material undergoes a specific deformation, it will inherently exhibit some resistance to this deformation. This total resistance is represented by the total stress, denoted as  $\sigma_{total}$ . In our framework, this total stress can be abstracted to the sum of the resistances provided by the upper and lower branches. The resistance from the upper branch is derived from elasto-plasticity, while the resistance from the lower branch arises due to viscosity.

$$\sigma_{total} = \sigma_{ep} + \sigma_v, \quad (2)$$

where  $\sigma_{ep}$  denotes the elasto-plastic stress, representing the resistance from the upper branch of Fig. 2. Similarly,  $\sigma_v$  signifies the viscous stress, corresponding to the resistance provided by the lower branch in Fig. 2. Subsequently, it is necessary to determine the individual computation methods for  $\sigma_{ep}$  and  $\sigma_v$ .

The stress,  $\sigma_{ep}$ , is related to the elasto-plastic strain  $\epsilon_{ep}$ , but this relation is not simply linear. The elasto-plastic strain can be further decomposed into the elastic part and the plastic part:

$$\epsilon_{ep} = \epsilon_e + \epsilon_p. \quad (3)$$

The relation between stresses and strains for elasto-plasticity is described by Eq. 4.

$$\begin{cases} \sigma_e = K : \epsilon_e \\ \sigma_p = 0 \end{cases}, \quad (4)$$

where the plastic strain  $\epsilon_p$  exhibits zero resistance to deformation, which results in  $\sigma_p$  always being zero. This is a characteristic of plasticity, where the material enters a state known as yielding. In

this state, the material offers no resistance or only mild resistance to deformation. This is the underlying rationale for using a slider to represent plasticity in our Generalized Maxwell model, the slider can move freely without any damping. On the other hand, the elastic constitutive relation is linear due to our adoption of linear elasticity theory, thus we use a spring to represent elasticity. In the upper row of Eq. 4,  $K$  represents a fourth-order stiffness tensor. A visual schematic diagram illustrating the stress-strain relation in one-dimensional case is presented in Fig.3. Please note that in this one-dimensional diagram,  $K$  is represented as a scalar coefficient, indicating that  $K$  represents the slope of the curve before reaching the elastic limit. Further details about elasto-plasticity are discussed in Section 4.2.

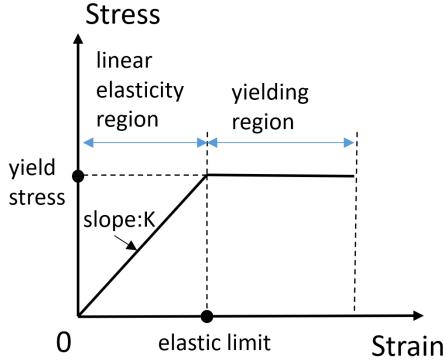


Fig. 3. Schematic diagram of elasto-plastic constitutive relation in one dimension.

The bottom branch of Fig. 2, i.e., the viscosity, is symbolized by a dashpot. This is because while viscosity has a resistance effect, it differs from the elastic component in that it responds to the rate at which the deformation is applied rather than the magnitude of the deformation. This concept is encapsulated in mathematical expressions, indicating that the magnitude of the viscous stress is related to the strain rate, the first derivative of the strain with respect to time. Here, the viscous stress,  $\sigma_v$ , is determined by the viscous strain rate,  $\dot{\epsilon}_v$ , but it's important to note that this relation is not necessarily linear. More details of the connection between viscous stress and viscous strain rate will be discussed in Section 4.1.

## 4 NON-NEWTONIAN MATERIALS MODELING

Non-Newtonian behaviors can generally be classified into two categories [42], [43]: 1) time-independent and 2) time-dependent. Our work falls under the time-dependent category due to the co-existence of viscosity and elasto-plasticity. However, we can simplify this by considering these components separately, as we have already discussed in section 3. After the separation, the viscosity model can be viewed as time-independent, and we can implement it by adopting a variety of classic non-Newtonian models based on the apparent viscosity coefficient. Section 4.1 will discuss viscosity, and section 4.2 will cover elasto-plasticity. We will explain in detail how to implement these elements in our solver.

### 4.1 Variable Viscosity

As the viscosity may undergo drastic changes, a stable implicit solver becomes necessary. Therefore, we adopt the implicit

viscosity solver from Weiler [24], in which the viscous term can be discretized as follows:

$$\nabla \cdot \sigma_v = 2(d+2)v \sum_j \frac{m_j}{\rho_j} \frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2 + 0.01h^2} \nabla W_{ij}, \quad (5)$$

where  $v = \mu/\rho$  is the kinematic viscosity, and  $\mu$  is the dynamic viscosity. And  $d$  represents the dimension.  $x_{ij}$  is the distance between particle  $i$  and its neighbor  $j$ .  $h$  is the support radius,  $W$  is the kernel function of SPH solver. Then we impose an implicit time integration scheme for Eq. 5:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}^* + \frac{\Delta t}{\rho} \nabla \cdot (\mu \nabla \mathbf{v}(t + \Delta t)). \quad (6)$$

The implicit viscosity solver necessitates a distinct matrix solver. In our simulation, we employ a matrix-free conjugate gradient solver with Jacobi preconditioning. Detailed information can be found in Weiler's work [24].

Thanks to the separation of viscosity and elasto-plasticity as discussed in Generalized Maxwell Model in section 3, we could view the viscosity as time-independent. This means the strain rate and stress can be described by a single-value function between strain rate  $\dot{\epsilon}_v$  and viscous stress  $\sigma_v$ :

$$\sigma_v = f(\dot{\epsilon}_v), \quad (7)$$

where  $\sigma_v$  is the viscous stress, and  $\dot{\epsilon}_v$  is the viscous strain rate. This relation is non-linear so we use the  $f$  to represent their relation. Since  $\mu = d\sigma_v/d\dot{\epsilon}_v$ , the effective viscosity can also be expressed as the function of the strain rate, i.e.,

$$\mu = g(\dot{\epsilon}_v). \quad (8)$$

In the following paragraphs, we will introduce six classic strain rate dependent viscosity models. Each of them can be considered a variant of Eq. 8. Note that both the strain rate and viscous stress are second-order tensors. However, for simplicity, we retain the convention that every time we take the power of the tensor, we first compute the Frobenius norm of that tensor.

Based on the response of stress with respect to strain rate, the time-independent non-Newtonian behaviors into three types: 1) shear-thinning body (also known as pseudo-plastic body); 2) Bingham body without shear-thinning or with shear-thinning (also known as Casson body); and 3) shear-thickening body (also known as dilatant body).

Fig. 4 illustrates the relation between strain rate and viscous stress for different time-independent non-Newtonian behaviors. The slope of the rheogram represents the viscosity. As it is shown, shear-thinning behavior exhibits a decreasing slope as the strain rate increases, while the opposite is observed for shear-thickening behavior. A rheogram curve of the Bingham body has a non-zero intercept on the stress axis, which is the yield stress. Shear-thinning is the most common type of time-independent, non-Newtonian behaviors. Daily necessities, such as ketchup, nail polish, and wall paint, exhibit shear-thinning behaviors.

We use the velocity gradient to calculate the strain rate, as shown in Eq. 9:

$$\dot{\epsilon}_v = \frac{1}{2} (\nabla \mathbf{v} + \nabla \mathbf{v}^T), \quad (9)$$

where  $v$  represents velocity, and the velocity gradient  $\nabla \mathbf{v}$  can be discretized as:

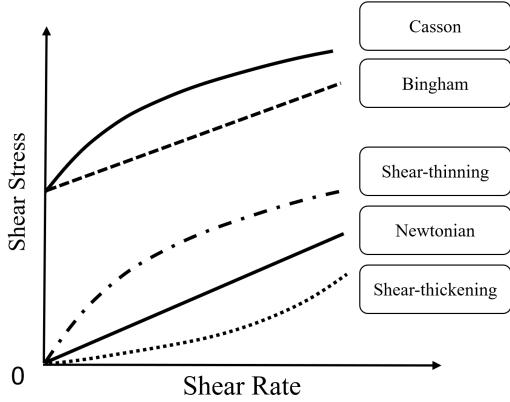


Fig. 4. Classification of time-independent non-Newtonian behaviors.

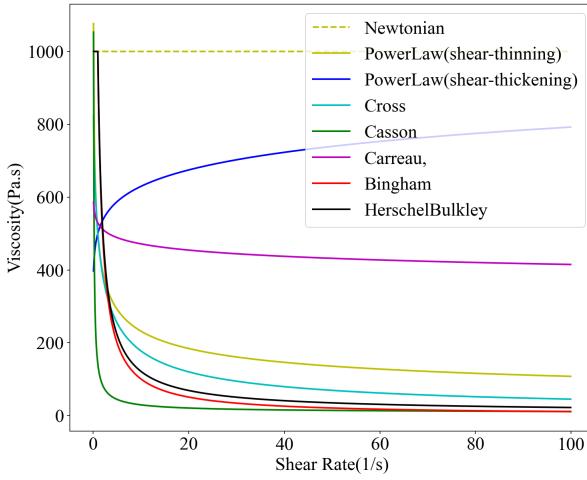


Fig. 5. Different strain rate dependent viscous models. The apparent viscosity changes with the shear rate. The parameters used in Fig. 5 are listed in Table 4.

$$\nabla \mathbf{v}_i = \frac{1}{\rho_i} \sum_j m_j (\mathbf{v}_j - \mathbf{v}_i) \nabla W_{ij}^\top. \quad (10)$$

In the following paragraphs, we will develop customized models of the six classic time-independent non-Newtonian behaviors and express them in the form of effective viscosity. Fig. 5 gives us an estimation of how the effective viscosity varies with the strain rate. The parameters used in Fig. 5 can refer to Table 4.

**Power Law.** The first and simplest model is the Power Law model (also known as the Ostwald De Waele model). As the name shows, it is simply a power-law relation, given by:

$$\mu(\dot{\epsilon}_v) = m(\dot{\epsilon}_v)^{n-1}, \quad (11)$$

where  $n$  is the power index and  $m$  is the consistency index.  $0 < n < 1$  corresponds to the shear-thinning case, and  $n > 1$  corresponds to the shear-thickening case. For most shear-thinning materials,  $n$  ranges from 0.3 to 0.7. One advantage of the Power Law model is that it is not only capable of simulating shear-thinning, but also capable of simulating shear-thickening.

**Cross Model.** The Cross model is superior to Power Law for representing shear thinning behaviour with three regions, including the start-up region, the middle region, and the ending region. As shown in Fig. 6, the curve representing the relationship between

viscosity and shear rate exhibits a reverse-S shape. The viscosity in the start-up region and the ending region approach  $\mu_\infty$  and  $\mu_0$ , respectively. The viscosity in the middle region has a linear relation between  $\mu$  and  $\dot{\epsilon}_v$ . The formula is shown in Eq. 12, where  $m$  is the relaxation time. And we set  $n = 2/3$ , which is a good default value for most materials [44].

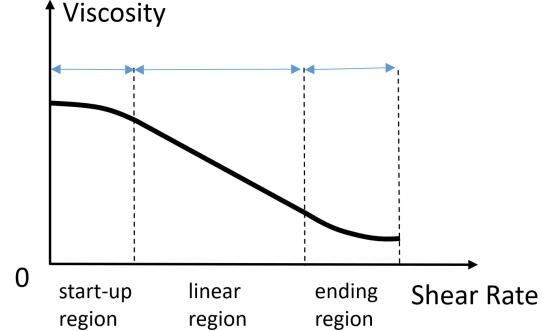


Fig. 6. The reverse-S shape viscosity-shear-rate curve for Cross Model and Carreau Model

$$\mu(\dot{\epsilon}_v) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{1 + (m\dot{\epsilon}_v)^n}. \quad (12)$$

**Carreau Model.** Carreau model also improves the fitting of start-up region and ending region like the Cross model [42]. As Eq. 13 defined,

$$\mu(\dot{\epsilon}_v) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1 + (m\dot{\epsilon}_v)^\alpha)^{(1-n)/\alpha}}, \quad (13)$$

where the default value for  $\alpha$  is 2. Notably, this formula simplifies to the Newtonian model when the strain rate is extremely low and into the power law when the strain rate is extremely large.

**Bingham model.** The Bingham model is the most straightforward model for describing a fluid of the Bingham type. These fluids are distinguished by the presence of a yield stress. When viscous stress is less than yield stress, the fluid behaves like a rigid body. When viscous stress is greater than yield stress, the Bingham fluid behaves either like Newtonian fluid or like shear-thinning fluid [15]. We adopt the Eq. 14 to formulate the Bingham plastic behavior without shear-thinning:

$$\mu(\dot{\epsilon}_v) = \begin{cases} \mu_0, & \dot{\epsilon}_v \leq \dot{\epsilon}_{v,c} \\ \mu_\infty + \tau_0/\dot{\epsilon}_v, & \dot{\epsilon}_v > \dot{\epsilon}_{v,c} \end{cases}, \quad (14)$$

where  $\mu_0$  is the initial viscosity,  $\tau_0$  is the yield stress. The  $\dot{\epsilon}_{v,c}$  is the critical strain rate corresponding to the yield stress. It must satisfy  $\tau_0 = \dot{\epsilon}_{v,c}(\mu_0 - \mu_\infty)$  to make the formula continuous. A smaller  $\tau_0$  means the viscosity change is more drastically.

**Casson Model.** Casson model is a model to describe the Bingham behavior with shear-thinning. The Casson model is given by:

$$\mu(\dot{\epsilon}_v) = \left( \sqrt{\mu_c} + \sqrt{\tau_0/\dot{\epsilon}_v} \right)^2. \quad (15)$$

The Casson model's significant difference from the Bingham model is that there is a square root relation between effective viscosity and the strain rate.

**Herschel-Bulkley Model** The Herschel–Bulkley model also describes Bingham's related behavior with shear thinning. It can be

seen as a combination of the Power Law model and the Bingham model. The formula is given by Eq. 16:

$$\mu(\dot{\epsilon}_v) = \begin{cases} \mu_0, & \dot{\epsilon}_v \leq \dot{\epsilon}_{v,c} \\ \tau_0/\dot{\epsilon}_v + m\dot{\epsilon}_v^{n-1}, & \dot{\epsilon}_v \geq \dot{\epsilon}_{v,c} \end{cases}, \quad (16)$$

where it satisfies  $\tau_0 = \mu_0 \dot{\epsilon}_{v,c} - m \dot{\epsilon}_{v,c}^n$  to ensure the continuity of the function.

## 4.2 Elasto-plasticity

In this section, we will implement the elastic and plastic models to empower our solver with the capability to solve non-Newtonian fluids to capture the time-dependent behaviors. One important characteristic of non-Newtonian material is that it neither belongs to a fluid nor a solid. It exhibits both fluid-like behavior and solid-like behavior.

Our method of simulating plasticity is improved from that of O'Brien et al. [32] while the implementation of elasticity is based on the study of Becker et al. [31].

**Plasticity.** When a material undergoes deformation (measured by strain) that exceeds its elastic limit, the deformation will be permanently preserved and material exhibits zero resistance to the excessive part of deformation, a phenomenon commonly referred to as yielding. The elasto-plastic strain adheres to a summation law for plastic and elastic strain, as represented in Eq. 3. Our primary objective is to determine the plastic strain.

First, the elastic strain deviation is calculated. It represents the strain stripped off the hydrostatic pressure, which has no impact on the plasticity.

$$\epsilon' = \epsilon_{ep} - \frac{\text{Tr}(\epsilon_{ep})}{3} \mathbf{I}, \quad (17)$$

where  $\epsilon_{ep}$  is the elasto-plastic strain,  $I$  is the identity matrix and  $\text{Tr}$  is the trace.

Then we need to compare the strain deviation with the elastic limit  $\gamma_1$ . Only when the strain deviation exceeds  $\gamma_1$ , the plasticity has effect. We adopt the Von Mises criterion to judge yielding. If  $\|\epsilon'\| > \gamma_1$  is satisfied, the yielding happens.

Additionally, the plastic strain increment is determined. Plastic strain, unlike elastic strain, accumulates from zero because plasticity is dependent on its past. Hence, we calculate the strain increment:

$$\Delta\epsilon^p = \frac{\|\epsilon'\| - \gamma_1}{\|\epsilon'\|} \epsilon'. \quad (18)$$

Finally, we update the plastic strain:

$$\epsilon^p = (\epsilon^p + \Delta\epsilon^p) \min \left( 1, \frac{\gamma_2}{\|\epsilon^p + \Delta\epsilon^p\|} \right), \quad (19)$$

where  $\epsilon^p$  represents plastic strain. The plastic limit is denoted as  $\gamma_2$ . Plastic strain is bounded by the plastic limit, and beyond the limit, it will tear the object into separate parts.

**Elasticity.** Referring to Eq. 3 as we discussing the Generalized Maxwell Model, the elastic strain is obtained by subtracting the plastic strain from the elasto-plastic strain. So this time our goal is to first calculate the elasto-plastic strain, and elastic force subsequently. We implement a linear corotated elasticity as Becker's [31].

First, we calculate the rotation matrix from the particle positions of the current configuration and initial configuration. We follow the method given by Muller et al. [45], as shown in Eq.20:

---

### Algorithm 1 Calculate elastic force with the plasticity

---

**Input:** rest and current positions  $x_i^0, x_i$   
 1: initial neighbors and initial volumes  
 2: (initial preparation)  
 3: **for** all particles  $i$  **do**  
 4:     store rest position and initial neighbors  
 5:     set  $\epsilon_p$  to zero  
 6: **end for**  
 7:  
 8: (for every timestep)  
 9: **for** all particles  $i$  **do** ▷ extract rotation  
 10:     compute rotation matrix  $R_i$  by Eq.20  
 11: **end for**  
 12: **for** all particles  $i$  **do** ▷ compute stress  
 13:     **for** all  $i$ 's neighbor  $j$  **do**  
 14:         calculate the displacement  $\mathbf{u}_{ji}$  by Eq. 22  
 15:     **end for**  
 16:     calculate displacement gradient  $\nabla \mathbf{u}_i$  by Eq. 23  
 17:     calculate elasto-plastic strain  $\epsilon_{ep}$  by Eq. 25  
 18:     calculate strain deviation  $\epsilon'$  by Eq. 17  
 19:     **if**  $|\epsilon'| > \gamma_1$  **then** ▷ von Mises plasticity  
 20:         calculate the plastic strain increment  $\Delta\epsilon_p$  by Eq. 18  
 21:         update the plastic strain  $\epsilon_p$  by Eq. 19  
 22:     **end if**  
 23:     calculate elastic strain by  $\epsilon_e = \epsilon_{ep} - \epsilon_p$   
 24:     transfer strain to the stress:  $\sigma_e = \mathbf{K} : \epsilon$   
 25: **end for**  
 26: **for** all particles  $i$  **do** ▷ compute forces  
 27:     **for** all  $i$ 's neighbor  $j$  **do**  
 28:         calculate  $\mathbf{f}_{ji}$  by Eq. 27  
 29:     **end for**  
 30:     calculate elastic force  $\mathbf{f}_i^e$  by Eq. 29  
 31: **end for**

**Output:** elastic force with plasticity  $\mathbf{f}_i^e$

---

$$\mathbf{R}_i = \mathbf{A}_{pq} \mathbf{S}_i^{-1}, \quad (20)$$

where  $S$  is the symmetric part of deformation matrix,  $S = \sqrt{\mathbf{A}_{pq}^T \mathbf{A}_{pq}}$ . And the  $\mathbf{A}_{pq}$  is:

$$\mathbf{A}_{pq} = \sum_j m_j W(\mathbf{x}_{ij}^0, h) \left( (\mathbf{x}_j - \mathbf{x}_i) (\mathbf{x}_j^0 - \mathbf{x}_i^0)^T \right), \quad (21)$$

where  $\mathbf{x}_j^0$  is the initial position of  $j$ th particle,  $x_i^0$  refers to the initial position of particle  $i$ , and  $h$  is the kernel radius.

Secondly, it is necessary to calculate the displacement  $\mathbf{u}$  and its gradient  $\nabla \mathbf{u}$ :

$$\mathbf{u}_{ji} = \mathbf{R}_i^{-1} (\mathbf{x}_j - \mathbf{x}_i) - (\mathbf{x}_j^0 - \mathbf{x}_i^0), \quad (22)$$

$$\nabla \mathbf{u}_i = \sum_j \tilde{v}_j \mathbf{u}_{ji} \nabla W(\mathbf{x}_{ij}^0, h)^T, \quad (23)$$

where  $\tilde{v}$  denotes the initial volume of particle. Eq. 22 can be viewed as a restoring of the rigid body motion. Thus, the extracted part is the displacement.

Third, we calculate the elasto-plastic strain  $\epsilon_{ep}$  based on the gradient of displacement  $\nabla \mathbf{u}$ . Here we use the the Cauchy strain(aka. linear strain or small strain) to describe the linear elasticity:

$$\boldsymbol{\varepsilon}_{\text{e}} = \frac{1}{2} (\mathbf{F} + \mathbf{F}^T) - \mathbf{I}, \quad (24)$$

where the  $\mathbf{F}$  denotes the deformation gradient, which is the Jacobian matrix of displacement.  $\mathbf{I}$  denotes the identity matrix. The  $\mathbf{F}$  can be also regarded the gradient of the deformation map, i.e.,  $\mathbf{F} = \nabla \mathbf{x}^0 + \nabla \mathbf{u}^T = \mathbf{I} + \nabla \mathbf{u}^T$ .

We can further simplify the Eq.24 by substituting  $\mathbf{F}$  into it. Then we get the relation between elasto-plastic strain  $\boldsymbol{\varepsilon}_{\text{ep}}$  and gradient of displacement  $\nabla \mathbf{u}$ :

$$\boldsymbol{\varepsilon}_{\text{ep}} = \frac{1}{2} \left( \nabla \mathbf{u}^T + (\nabla \mathbf{u}^T)^T \right). \quad (25)$$

After getting the elasto-plastic strain, the subsequent step is to calculate the plastic strain, which has been described previously. Then, we get the elastic strain by subtracting the elastic strain from the elasto-plastic strain.

The next step is to apply Hook's Law, which can transfer the strain to the stress. The Hook's Law describes a stress-strain relation as

$$\boldsymbol{\sigma}_{\text{e}} = \mathbf{K} : \boldsymbol{\varepsilon}_{\text{e}}, \quad (26)$$

where  $\mathbf{K}$  is the stiffness tensor, which is a fourth order tensor.

If the material is isotropic and homogeneous, it can be expressed by only two material parameters. The two parameters have multiple equivalent choices. A common choice is the Poisson ratio  $\nu$  and Young's modulus  $E$ . The Poisson ratio is in  $[0, 0.5]$ , which measures the incompressibility of a material. The Young's modulus ranges from thousands to billions, which measures the stretch resistance.

Then, we can get the elastic force  $\mathbf{f}_{ij}$  based on the stress:

$$\mathbf{f}_{ji} = -\tilde{v}_i \boldsymbol{\sigma}_{e,i} \mathbf{d}_{ij}, \quad (27)$$

where  $\mathbf{f}_{ji}$  is the elastic force originating from  $j$  and acting on  $i$ . The  $d_{ij}$  is calculated by

$$\mathbf{d}_{ij} = \tilde{v}_j \nabla W(\mathbf{x}_{ij}^0, h), \quad (28)$$

where  $\tilde{v}$  denotes the initial volume of the particle. Finally, the elastic force particle  $i$  received is:

$$\mathbf{f}_i^e = \sum_j \frac{-\mathbf{R}_i \mathbf{f}_{ji} + \mathbf{R}_j \mathbf{f}_{ij}}{2}. \quad (29)$$

The full algorithm to calculate elasticity and plasticity is summarized in Algorithm 1.

### 4.3 Diffusion Model

By implementing a diffusion equation, we could further achieve the melting and solidifying phenomena. In contrast to our existing non-Newtonian viscosity models, where viscosity is controlled by strain rate, the diffusion model can be viewed as a non-Newtonian phenomenon whose viscosity is controlled by temperature [5], [46]. Thanks to the pure Lagrangian framework, we do not need to tackle the convection because it is naturally satisfied. All we need is to implement a diffusion equation. The governing equation that controls the evolution of a scalar field (e.g. temperature) [47], [48] is:

$$\frac{\partial T_i}{\partial t} = D \sum_{j \neq i} \frac{m_j}{\rho_j \rho_i} (T_j - T_i) |\nabla W_{ij}| + R, \quad (30)$$

where  $T$  is the temperature of the particle.  $D$  is the diffusion coefficient, and it controls the diffusion speed. And  $R$  is the source term, which controls the heat generation. Eq.30 says that the diffusion is driven by the temperature difference of the field.

## 5 EXPERIMENTAL RESULTS AND COMPARISONS

### 5.1 Comparison with Existing Works



Fig. 7. Comparisons of visco-elastic-plastic behaviors (particles view). From left to right, there are three toy cows modeled by MPM [36], FLIP [10], and our unified framework, respectively. The particle color represents magnitude of the instantaneous velocity.

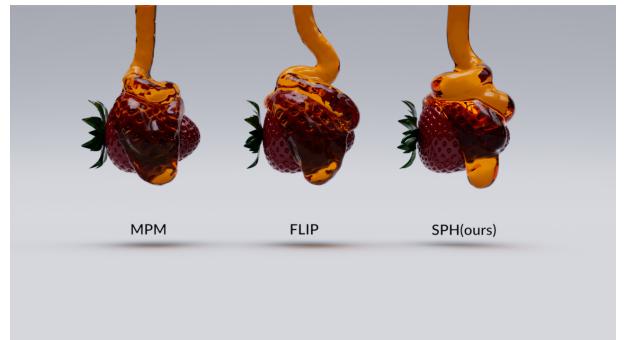


Fig. 8. Comparisons of highly viscous behaviors. From left to right, there are three phenomena of honey dripping modeled by MPM (Fang et al.) [36], FLIP (Shao et al.) [10] and our framework, respectively.

TABLE 1  
Performance comparison with existing works.

Method	Time(s)/frame		Particles number	
	Cow	Honey	Cow	Honey
FLIP [10]	<u>0.44</u>	0.06	275,556	13,815
MPM [36]	23.4	11.6	274,462	13,860
SPH (ours)	22.7	<u>0.04</u>	276,625	13,780

To validate the effectiveness and show the superiority of our framework, we draw comparisons with three state-of-the-art frameworks concerning highly viscous fluid and visco-elastic fluid simulation, including MPM by Fang et al. [36], FLIP by Shao et al. [10], and MPM by Su et al. [5].

Fig. 7 illustrates three toy cows with similar particles number falling on the ground, each modeled by FLIP [10], MPM [36], and our unified framework, respectively. When compared with the

TABLE 2  
Supported features compared with existing works.

Framework	FLIP [10]	MPM [36]	SPH (Ours)
high viscosity	✓	✓	✓
variable viscosity	✗	✗	✓
visco-elasticity	✗	✓	✓
plasticity	✗	✓	✓
cutting	✗	✗	✓

MPM-based model of [36], which exhibits similar visco-elastic-plastic behavior to our non-Newtonian model, our framework demonstrates superior performance. While the FLIP model, being a classical viscosity model, naturally exhibits the best time performance because it can not handle visco-elastic-plastic modeling. However, as shown in Table 2, our method can deal with more abundant non-Newtonian phenomena.

When simulating highly viscous honey, as depicted in Fig. 8, with similar particle number, our method performs on par with FLIP [10] and outperforms MPM [36].



Fig. 9. Comparison with [5]. Our method (left) is capable of simulating melting and cutting, with higher allowable time step size (5e-3s vs. 1e-4s).

Also, we have performed a comparison with Su21 [5] while simulating the classical melting bunny scene (as shown in Fig. 9). In contrast to this scene, our framework accommodates a larger time step size (our average time step size is 5e-3, compared to 1e-4 in [5]). Moreover, owing to the pure particle framework, our work effectively handles cutting without any additional treatment. Thanks to our unified particle system, cutting an object is a seamless process without any further complications related to topology changes.

## 5.2 Phase Change of Non-Newtonian Materials

**Ice-cream melting.** Fig.10 shows the process of non-Newtonian viscous ice-cream melting. We first set a heat source at the surface of the ice cream, and then the heat diffuses gradually into the inside, which can be controlled by Eq.30. To implement this, we define the surface particles as those having ten or fewer neighbours. The viscosity of the ice cream decreases as the temperature rises. We introduce a user-controllable parameter  $d$  (decay) to slow down the viscosity drop. The relation between viscosity coefficient and temperature is given by:

$$\mu_i = \mu_0 e^{-dT}, \quad (31)$$

where  $T$  is the temperature, heat source  $R = 1.0$  (only at surface),  $\mu_0 = 1000$  is the initial viscosity,  $D = 30.0$  is the diffusivity, and decay factor  $d = 0.1$ .



Fig. 10. Ice-cream melting.

To texture a topologically changed animation, the texture is first mapped onto the point and carried throughout the simulation before being mapped back to the surface during rendering.

**Cutting and melting.** The left figure in Fig. 9 illustrates the process of melting. The figure demonstrates that our solver is capable of handling object cutting as well. The cutting phenomenon is realized by the insertion of the cutting board, which is composed of particles and has a certain thickness. The board divides the bunny particles to disperse, leading them to travel beyond the neighbourhood search threshold, thereby creating the separating effect. In this particular scene, the heat emanates from the ground (set by a bounding box with [-1, 0, 1] to [1, 0.05, 1]) where a heat source  $R = 1.0$  is present, with a diffusivity of 100.0. And initial viscosity  $\mu_0 = 20.0$ . We use the same decaying factor  $d = 0.1$ .

## 5.3 Shear Dependent Viscosity

**Cornstarch Mixture (Golf Smash).** As shown in Fig. 12, we use the Power Law model in Eq. 11 with a power index of 1.1 and a consistency index of 20 to simulate the scenario of a golf ball dropping into a cornstarch-water mixture. These values are chosen to ensure that the ball can naturally sink in the fluid at an initial velocity of zero.

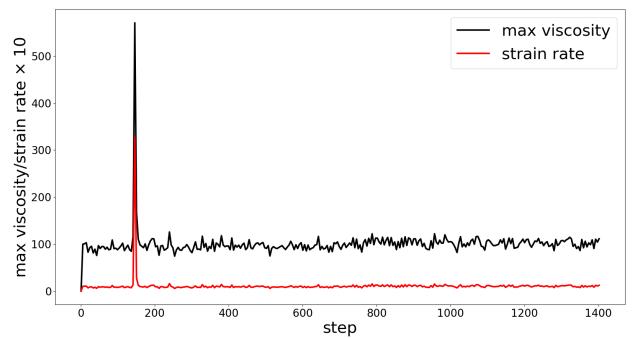


Fig. 11. Max viscosity and strain rate norm in every step. Through the change of viscosity and strain rate, we can also clearly see the rigidity change at the moment of golf ball contact.

In this experiment(Fig. 12), the classic shear-thickening phenomenon is replicated. The viscosity of the fluid will increase with the strain rate during the simulation, resulting in a large resistance when the ball smashes into the fluid (as shown in Fig.11). As a result, the ball will initially slow down and then gradually drop underwater. We compare our simulation with a real-world video of a similar scenario, and the motion patterns are quite similar (Fig. 12).

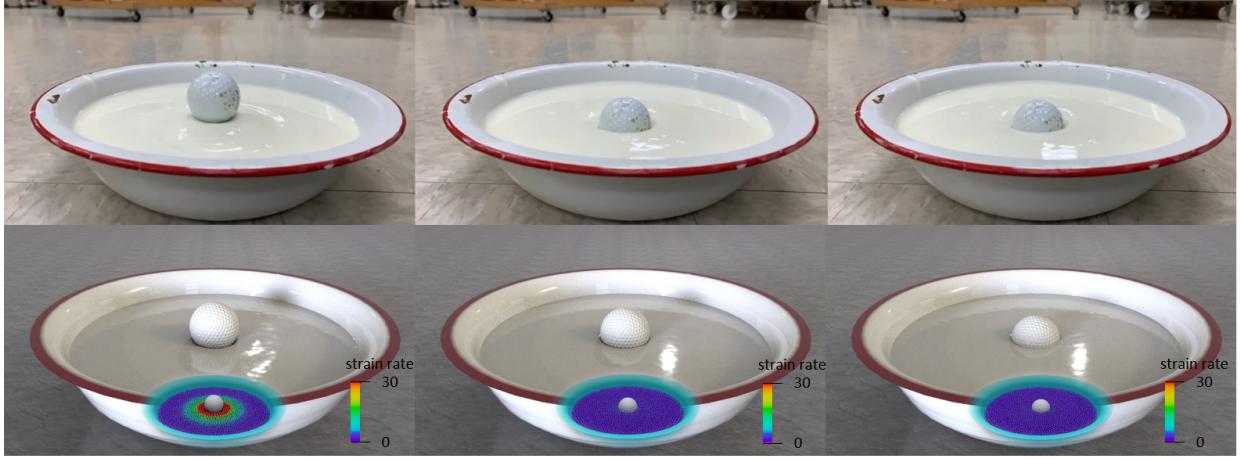


Fig. 12. Real and simulated classic non-Newtonian fluid scenarios. A golf ball drops into a shear-thickening fluid of cornstarch-water mixture. The upper-row pictures are real photos and the lower ones are our simulated rendering results and particles views, where the color corresponds to the strain rate of Fig. 11.



Fig. 13. Change the fluid to ketchup (which is shear-thinning) and keep other setups the same with golf smashing cornstarch-water mixture case (Fig. 12).

To compare the shear-thinning and shear-thickening behaviors, we substitute the fluid with ketchup (a shear-thinning fluid) and a golf ball falls into ketchup, while retaining other setup parameters. We adapt its parameters from [49], using a power law model ( $n=0.26$ ,  $m=5.86$ , parameters can be found in Table 3). Due to its shear-thinning nature, the viscosity decreases at the moment of collision, resulting in a splashing effect.

**Non-Newtonian Armadillos.** Fig. 14 depicts viscous armadillos with various time-independent non-Newtonian models. Parameters are listed in Table 4.

In this experiment, we evaluate seven distinct viscosity models that are related to shear rate. The viscosity models depicted in Fig. 14 include Carreau, Casson, Cross, PowerLaw2 (shear-thickening), PowerLaw1 (shear-thinning), Newtonian, Herschel-Bulkley, and Bingham models. These models can be classified into four categories: 1) Shear-thinning behaviors, which include Carreau, Cross, and PowerLaw1; 2) Shear-thickening behavior, primarily represented by PowerLaw2; 3) Bingham behavior, including Herschel-Bulkley, Bingham, and Casson; and 4) Newtonian behavior [42], [43].

As shown in Fig. 14, it's clear that PowerLaw2 (shear-thickening) exhibits the optimal flowability. This is due to the fact that it simulates shear-thickening behavior, where viscosity escalates with the increase of shear rate. In our experiment, the high-viscosity fluid begins flowing from zero velocity under the influence of gravity, at a relatively low shear rate. At low shear rates, the viscosity remains low. In contrast, PowerLaw1 (shear-thinning) exhibits the least flowability. This is due to its simulation of shear-thinning behavior under the same parameters as PowerLaw2 (with

the exception of  $n$ ). Consequently, at low shear rate, the viscosity remains high.

There are two models used for simulating Bingham behavior: the Herschel-Bulkley model and the Casson model. Beyond the yield point, the Herschel-Bulkley model adheres to PowerLaw behavior, which could manifest as shear-thinning, shear-thickening, or Newtonian. Consequently, it offers more flexibility and has a greater number of parameters. As demonstrated in Fig. 14, this model exhibits behavior similar to that of a Newtonian fluid. Conversely, the Casson model is only capable of simulating shear-thinning Bingham behavior. Therefore, in agreement with our analysis of PowerLaw1 (shear-thinning), the fluid flowability simulated by the Casson model is relatively poor at low shear rate.

The Carreau model and the Cross model [44] are used to simulate shear-thinning behaviors. These two models are somewhat similar, with their stress-strain curves displaying a reverse S-shaped pattern. Specifically, at both low and high shear rates, the constitutive curves tend to level out. Therefore, even though both models are used to simulate shear-thinning behavior similar to PowerLaw1, the viscosity at low shear rates doesn't reach the high levels observed with PowerLaw1, resulting in moderate flowability.

#### 5.4 Visco-elasticity and Visco-plasticity

**Falling of tomatoes.** Fig. 15 shows different visco-elastic and visco-plastic tomatoes dropping on the ground.

In this scene, the Youngs modulus is  $1e7$ , the Poisson ratio is 0.42, the plastic limit is 1.0, and the elastic limit is 0.001, 0.05, 0.1, and infinity, respectively. The upper row shows different visco-elasticity/plasticity and the bottom row shows different viscosity

TABLE 3  
Key parameters of our experiments.

Scene	Parameters
Golf(slowly sink, Fig. 12)	Power law n=1.1, m=20.
Golf(ketchup, bounce back, Fig. 13)	Power law n=0.26, m=5.86 [49]
Armadillo (Fig. 14)	refer to Table 4.
Ice Cream (Fig. 10)	$\mu_i = \mu_0 e^{-dT}$ , $\mu_0 = 1000, D = 30, R = 1, d = 0.1$
Cut and Melt (Fig. 9)	$\mu_i = \mu_0 e^{-dT}$ , $\mu_0 = 20, D = 1000, R = 1, d = 0.1$
Tomato(top row first, Fig. 15)	$E = 1e7, v = 0.42, \gamma_l = 0.001, \gamma_2 = 1.0$
Tomato(top row second, Fig. 15)	$E = 1e7, v = 0.42, \gamma_l = 0.05, \gamma_2 = 1.0$
Tomato(top row third, Fig. 15)	$E = 1e7, v = 0.42, \gamma_l = 0.1, \gamma_2 = 1.0$
Tomato(top row fourth, Fig. 15)	$E = 1e7, v = 0.42, \gamma_l = \infty, \gamma_2 = 1.0$
Tomato(Newtonian, Fig. 15)	$\mu_0 = 0.01$
Tomato(Non-Newtonian1, Fig. 15)	Herschel-Bulkley $m = 1.0, n = 0.667, \mu_0 = 0.1, \dot{\epsilon}_{v,c} = 10^{-5}$
Tomato(Non-Newtonian2, Fig. 15)	Herschel-Bulkley $m = 20, n = 1.1, \mu_0 = 10, \dot{\epsilon}_{v,c} = 10^{-3}$
Tomato(Non-Newtonian3, Fig. 15)	Herschel-Bulkley $m = 50, n = 1.1, \mu_0 = 100, \dot{\epsilon}_{v,c} = 100$
Honey (our SPH, Fig. 8)	$\mu = 5$
Honey (FLIP, middle Fig. 8)	$\mu = 5000$
Honey (MPM, left Fig. 8)	$\lambda_E = 1428, \mu_E = 1428, \lambda_N = 28, \mu_N = 7140, dx = 0.05$
Cow (our SPH, right Fig. 7)	$HB, m = 1.0, n = 0.9, \mu_0 = 80, \dot{\epsilon}_{v,c} = 5 \times 10^{-4}, \gamma_l = 0.001$
Cow (FLIP, left Fig. 7)	$\mu = 500$
Cow (MPM, middle Fig. 7)	$\lambda_E = 1428.6, \mu_E = 35.7, \lambda_N = 28.6, \mu_N = 7.1$

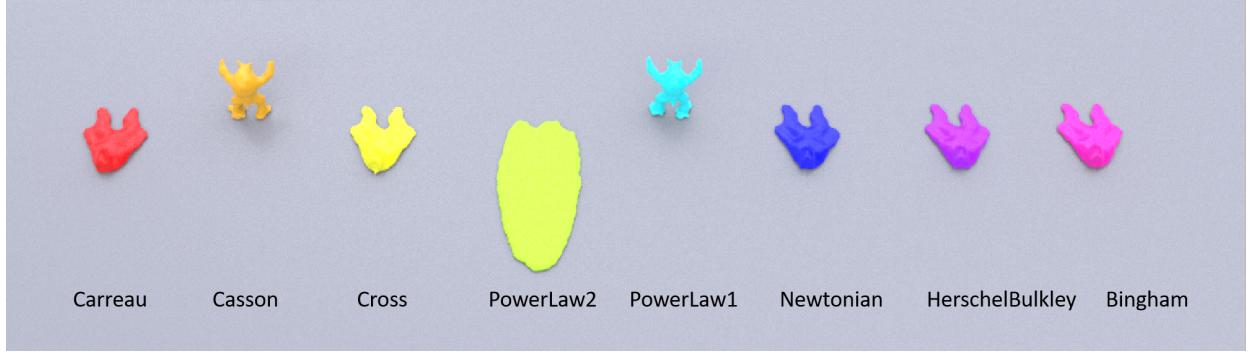


Fig. 14. Armadillos with different strain-rate-dependent non-Newtonian viscosity model. From left to right: Carreau, Casson, Cross, PowerLaw2, PowerLaw1, Newtonian, Herschel-Bulkley, Bingham. Detailed parameters are listed in Table 4.

TABLE 4  
Parameters of the armadillo sliding scenario.

	n	m	$\mu_c$	$\tau_0$	$\dot{\gamma}_c$	$\mu_\infty$	$\mu_0$
Newtonian	-	-	-	-	-	-	10
PowerLaw1	0.667	4.5	-	-	-	0.1	10
PowerLaw2	1.5	1.0	-	-	-	0.1	10
Cross	0.667	1.0	-	-	-	0.1	10
Casson	-	-	1.0	10	-	0.1	10
Carreau	0.1	0.2	-	-	-	0.1	10
Bingham	-	-	-	-	1.0	0.1	10
HerschelBulkley	0.667	10	-	-	10	0.1	10

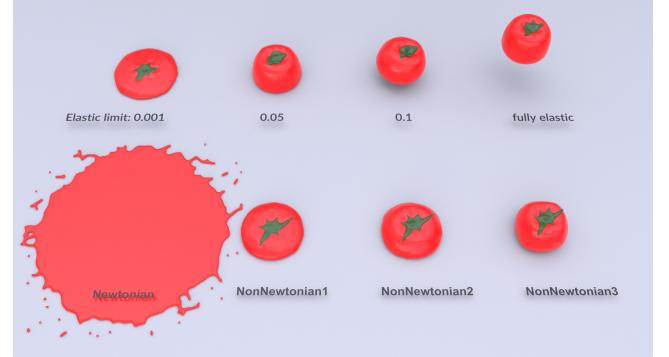


Fig. 15. Various non-Newtonian tomatoes dropping on the ground. The upper row displays different levels of visco-elasticity and plasticity, ranging from more plastic to more elastic. The lower row illustrates different viscous models, progressing from less viscous to more viscous on the right. See Table 3 for parameters.

models. The upper rightmost one is fully elastic and contains no plastic, therefore, the elastic limit can be seen as infinity. The existence of plasticity will make the tomato less energetic and reduce its ability to restore its original shape. As we increase the elastic limit in the upper row, which means the plastic region is easier to reach, the plastic effect will be more obvious. This gives users an easy way to control the plasticity and generate different effects as they want.

The lower row of Fig. 15 shows non-Newtonian viscous model to exhibit that our solver is also capable of solving viscous fluid together. From left to right, the parameters of the lower row are

as follows. Newtonian: 0.01 viscosity; Non-Newtonian1: Herschel-Bulkley model with  $m = 1.0, n = 0.667, \mu_0 = 0.1, \mu_\infty = 0.001, \dot{\gamma}_c = 1e-5$ , Non-Newtonian2:  $m = 20, n = 1.1, \mu_0 = 10, \mu_\infty = 0.01, \dot{\gamma}_c = 1e-3$ , Non-Newtonian3:  $m = 50, n = 1.1, \mu_0 = 100, \mu_\infty = 1, \dot{\gamma}_c = 100$ . Readers can refer to table 3 for all parameters.

We have noticed an unnatural trembling when the elastic scene has a large number of particles (over 100K). We suspect that this trembling is inherent to the particle-based elastic simulation itself. Given that particles are only locally connected through neighborhood searches, the information they store is confined to their immediate surroundings.

## 6 CONCLUSION AND LIMITATIONS

In this paper, we develop a physically meaningful solution for various non-Newtonian behaviors modeling. The variable physical viscous models and our empirical parameter system enable most of the non-Newtonian materials. We design a non-linear time-dependent model based on the Generalized Maxwell Model. The stress is divided into an elasto-plastic part and a viscous part, which allows us to separately tackle them. The viscous and elasto-plastic stresses are combined into the Navier-Stokes equations to achieve various rheological behaviours. Meanwhile, the introduced diffusion models can support many interesting cases, such as phase change phenomenon.

There are several limitations to our method that can be addressed in future research. First, we face the particle deficiency problem that is common to all SPH-based methods. It affects not only the quality of our simulation but also imposes numerical stability issues, especially when we need a drastic change in viscosity. We address this issue by prolonging the number of iterations of the viscosity solver, but this is an ad-hoc solution that sacrifices performance. We expect better solutions to this problem to emerge from future research. Second, we have not yet optimised the efficiency of our solver. We use the conjugate gradient as our viscosity solver and DFSPH as the algorithm to solve pressure, both of which are implemented on the CPU. No extra effort has been made to boost performance. Third, We also observed that unreasonable extreme parameters may lead to the non-convergence of numerical simulations. For example, rapidly changing viscosity and excessively large elastic modulus can cause issues such as numerical explosion. For instance, the moment the golf ball collides with the fluid in the falling golf ball scenario will trigger a sudden change in viscosity, which may result in a breakdown. Similarly, in the case of a falling elastic tomato, if the elastic modulus is set extremely high, the program will crash immediately. We speculate that these extreme situations may be attributed to numerical stiffness issues. Therefore, a stable viscosity and elastoplastic solver may be the key to resolving these problems. We anticipate further exploration in this area in future work.

## ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2023YFC3604504), National Natural Science Foundation of China (No. 62002010), CAMS Innovation Fund for Medical Sciences (CIFMS) under Grant (2019-I2M-5-016), and the Beijing Science and Technology Project (No. Z221100007722001, Z231100005923039). This work was also partially supported by USA NSF IIS-1715985 and USA NSF IIS-1812606 (awarded to Hong Qin).

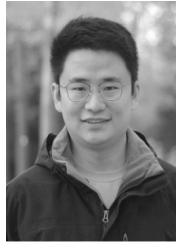
## REFERENCES

- [1] D. Stora, P. O. Agliati, M. P. Cani, F. Neyret, and J. D. Gascuel, “Animating lava flows,” in *Proceedings of the Graphics Interface 1999 Conference*, 1999, pp. 203–210.
- [2] A. Peer, M. Ihmsen, J. Cornelis, and M. Teschner, “An implicit viscosity formulation for SPH fluids,” *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 1–10, 2015.
- [3] R. Goldade, Y. Wang, M. Aanjaneya, and C. Batty, “An adaptive variational finite difference framework for efficient symmetric octree viscosity,” *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–14, 2019.
- [4] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle, “Augmented MPM for phase-change and varied materials,” *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 1–11, 2014.
- [5] H. Su, T. Xue, C. Han, C. Jiang, and M. Aanjaneya, “A unified second-order accurate in time mpm formulation for simulating viscoelastic liquids with phase change,” *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–18, 2021.
- [6] D. Koschier, J. Bender, B. Solenthaler, and M. Teschner, “A survey on SPH methods in computer graphics,” *Computer Graphics Forum*, vol. 41, no. 2, pp. 737–760, 2022.
- [7] M. Carlson, P. J. Mucha, R. B. Van Horn, and G. Turk, “Melting and flowing,” in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002, pp. 167–174.
- [8] C. Batty and R. Bridson, “Accurate viscous free surfaces for buckling, coiling, and rotating liquids,” in *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation*, 2008, pp. 219–228.
- [9] E. Larionov, C. Batty, and R. Bridson, “Variational stokes: A unified pressure-viscosity solver for accurate viscous liquids,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–11, 2017.
- [10] H. Shao, L. Huang, and D. L. Michels, “A fast unsmoothed aggregation algebraic multigrid framework for the large-scale simulation of incompressible flow,” *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–18, 2022.
- [11] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa, “Point based animation of elastic, plastic and melting objects,” in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2004, pp. 141–151.
- [12] B. Solenthaler, J. Schläfli, and R. Pajarola, “A unified particle model for fluid-solid interactions,” *Computer Animation and Virtual Worlds*, vol. 18, no. 1, pp. 69–82, 2007.
- [13] J. J. Monaghan, “Smoothed particle hydrodynamics,” *Annual Review of Astronomy and Astrophysics*, vol. 30, no. 1, pp. 543–574, 1992.
- [14] M. Müller, D. Charypar, and M. Gross, “Particle-based fluid simulation for interactive applications,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003, pp. 154–159.
- [15] B. Zhu, M. Lee, E. Quigley, and R. Fedkiw, “Codimensional non-newtonian fluids,” *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 1–9, 2015.
- [16] M. Becker and M. Teschner, “Weakly compressible SPH for free surface flows,” in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007, pp. 209–217.
- [17] B. Solenthaler and R. Pajarola, “Predictive-corrective incompressible SPH,” in *Proceedings of the 2009 ACM SIGGRAPH*, 2009, pp. 1–6.
- [18] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, “Implicit incompressible SPH,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 426–435, 2014.
- [19] J. Bender and D. Koschier, “Divergence-free SPH for incompressible and viscous fluids,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1193–1206, 2017.
- [20] D. Koschier, J. Bender, B. Solenthaler, and M. Teschner, “Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids,” in *Eurographics 2019*, 2019, pp. 1–41.
- [21] H. Schechter and R. Bridson, “Ghost SPH for animating water,” *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–8, 2012.
- [22] T. Takahashi, Y. Dobashi, I. Fujishiro, T. Nishita, and M. C. Lin, “Implicit formulation for SPH-based viscous fluids,” *Computer Graphics Forum*, vol. 34, no. 2, pp. 493–502, 2015.
- [23] A. Peer and M. Teschner, “Prescribed velocity gradients for highly viscous SPH fluids with vorticity diffusion,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 12, pp. 2656–2662, 2017.
- [24] M. Weiler, D. Koschier, M. Brand, and J. Bender, “A physically consistent implicit viscosity solver for SPH fluids,” *Computer Graphics Forum*, vol. 37, no. 2, pp. 145–155, 2018.
- [25] L. F. de Souza Andrade, M. Sandim, F. Petronetto, P. Pagliosa, and A. Paiva, “Particle-based fluids for viscous jet buckling,” *Computers and Graphics*, vol. 52, pp. 106–115, 2015.
- [26] Y. Gao, S. Li, A. Hao, and H. Qin, “Simulating multi-scale, granular materials and their transitions with a hybrid euler-lagrange solver,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 12, pp. 4483–4494, 2021.

- [27] D. Terzopoulos, J. Platt, and K. Fleischer, “Heating and melting deformable models,” *The Journal of Visualization and Computer Animation*, vol. 2, no. 2, pp. 68–73, 1991.
- [28] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw, “Multiple interacting liquids,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 812—819, 2006.
- [29] M. Fujisawa and K. T. Miura, “Animation of ice melting phenomenon based on thermodynamics with thermal radiation,” in *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, 2007, pp. 249—256.
- [30] T. G. Goktekin, A. W. Bargteil, and J. F. O’Brien, “A method for animating viscoelastic fluids,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 463–468, 2004.
- [31] M. Becker, M. Ihmsen, and M. Teschner, “Corotated SPH for deformable solids,” in *Proceedings of the Fifth Eurographics Conference on Natural Phenomena*, 2009, pp. 27—34.
- [32] J. F. O’Brien, A. W. Bargteil, and J. K. Hodgins, “Graphical modeling and animation of ductile fracture,” in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, 2002, pp. 291–294.
- [33] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, “A material point method for snow simulation,” *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–10, 2013.
- [34] Y. Yue, B. Smith, C. Batty, C. Zheng, and E. Grinspun, “Continuum foam: A material point method for shear-dependent flows,” *ACM Transactions on Graphics*, vol. 34, no. 5, pp. 1–20, 2015.
- [35] Y. Gao, S. Li, H. Qin, Y. Xu, and A. Hao, “An efficient FLIP and shape matching coupled method for fluid–solid and two-phase fluid simulations,” *The Visual Computer*, vol. 35, no. 12, pp. 1741–1753, 2019.
- [36] Y. Fang, M. Li, M. Gao, and C. Jiang, “Silly rubber: An implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids,” *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–13, 2019.
- [37] M. Gao, X. Wang, K. Wu, A. Pradhana, E. Sifakis, C. Yuksel, and C. Jiang, “GPU optimization of material point methods,” *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–12, 2018.
- [38] C. Gissler, A. Henne, S. Band, A. Peer, and M. Teschner, “An implicit compressible SPH solver for snow simulation,” *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 1–16, 2020.
- [39] A. W. Bargteil, C. Wojtan, J. K. Hodgins, and G. Turk, “A finite element method for animating large viscoplastic flow,” *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 1–8, 2007.
- [40] C. Wojtan and G. Turk, “Fast viscoelastic behavior with thin features,” *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–8, 2008.
- [41] O. Ozgen, M. Kallmann, and E. Brown, “An SPH model to simulate the dynamic behavior of shear thickening fluids,” *Computer Animation and Virtual Worlds*, vol. 30, no. 5, pp. 1–18, 2019.
- [42] N. Phan Thien and N. Mai Duy, *Understanding Viscoelasticity: An Introduction to Rheology*. Springer International Publishing, 2017, pp. 95–111.
- [43] R. P. Chhabra, *Non-Newtonian Fluids: An Introduction*. Springer New York, 2010, pp. 3–34.
- [44] M. M. Cross, “Rheology of non-Newtonian fluids: A new flow equation for pseudoplastic systems,” *Journal of Colloid Science*, vol. 20, no. 5, pp. 417–437, 1965.
- [45] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, “Meshless deformations based on shape matching,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 471–478, 2005.
- [46] Y. Gao, S. Li, L. Yang, H. Qin, and A. Hao, “An efficient heat-based model for solid-liquid-gas phase transition and dynamic interaction,” *Graphical Models*, vol. 94, pp. 14–24, 2017.
- [47] P. W. Cleary and J. J. Monaghan, “Conduction modelling using smoothed particle hydrodynamics,” *Journal of Computational Physics*, vol. 148, no. 1, pp. 227–264, 1999.
- [48] J. Orthmann and A. Kolb, “Temporal blending for adaptive SPH,” *Computer Graphics Forum*, vol. 31, no. 8, pp. 2436–2449, 2012.
- [49] P. Bottiglieri, F. De Sio, G. Fasanaro, G. Mojoli, M. Impembo, and D. Castaldo, “Rheological characterization of ketchup,” *Journal of Food Quality*, vol. 14, no. 6, pp. 497–512, 1991.



**Chunlei Li** received the master’s degree in Power Engineering and Engineering Thermophysics from Beihang University, in 2021. He is working toward the Ph.D. degree at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include computer graphics and physics-based simulation.



**Yang Gao** received the Ph.D. degree in computer science from Beihang University in 2019. He is currently an associate professor at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing. His research interests include computer graphics, VR/AR interaction, physics-based modeling and simulation. In particular, he is focusing on fluid simulation.



**Jiayi He** received her B.S. degree in computer science from Beihang University in 2023. Her research interests include computer graphics and physics-based simulation.



**Tianwei Cheng** received his master’s degree in the State Key Laboratory of Virtual Reality Technology and Systems from Beihang University in 2023. His research interests include computer graphics and physics-based simulation. In particular, he is focusing on viscous fluid simulation.



**Shuai Li** received the Ph.D. degree in computer science from Beihang University. He is currently a professor at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include computer graphics, physics-based modeling and simulation, virtual surgery simulation, computer vision, and medical image processing.



**Aimin Hao** is a professor in Computer Science School and the Associate Director of State Key Laboratory of Virtual Reality Technology and Systems at Beihang University. He received his B.S., M.S., and Ph.D. in Computer Science at Beihang University. His research interests are in virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.



**Hong Qin** received his B.S. and M.S. degrees in computer science from Peking University, and his Ph.D. degree in computer science from the University of Toronto. He is a professor of computer science in Department of Computer Science at Stony Brook University. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer-aided geometric design, human computer interaction, visualization, and scientific computing. He is a senior member of the IEEE and the IEEE Computer Society.