A Material Point Method for Nonlinearly Magnetized Materials

YUCHEN SUN*, CFCS, Peking University, China
XINGYU NI*, CFCS, Peking University, China
BO ZHU, Dartmouth College, United States of America
BIN WANG†, Beijing Institute for General Artificial Intelligence, China
BAOQUAN CHEN†, CFCS, Peking University, China

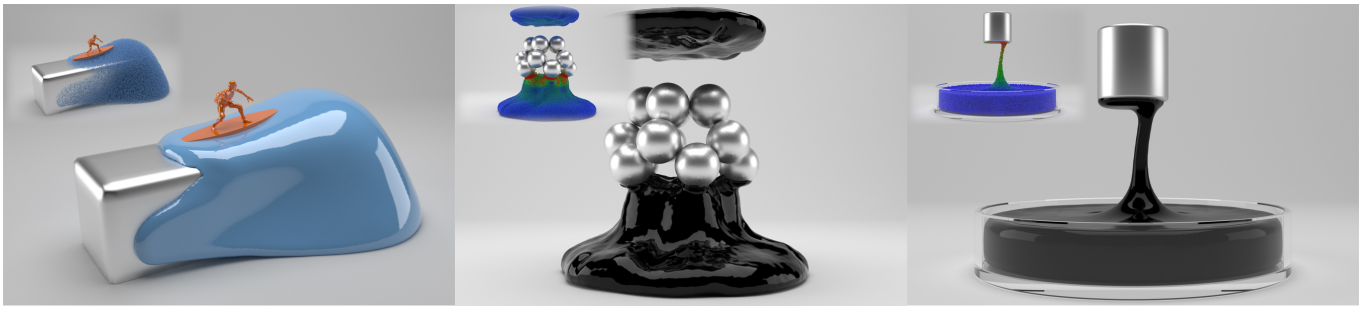

Fig. 1. Our unified numerical scheme can simulate the dynamics of a broad array of nonlinearly magnetized materials, ranging from magnetic deformable bodies to magnetic viscous fluids.

We propose a novel numerical scheme to simulate interactions between a magnetic field and nonlinearly magnetized objects immersed in it. Under our nonlinear magnetization framework, the strength of magnetic forces is effectively saturated to produce stable simulations without requiring any parameter tuning. The mathematical model of our approach is based upon Langevin's nonlinear theory of paramagnetism, which bridges microscopic structures and macroscopic equations after a statistical derivation. We devise a hybrid Eulerian-Lagrangian numerical approach to simulating this strongly nonlinear process by leveraging the discrete material points to transfer both material properties and the number density of magnetic micro-particles in the simulation domain. The magnetic equations can then be built and solved efficiently on a background Cartesian grid, followed by a finite difference method to incorporate magnetic forces. The multi-scale coupling can be processed naturally by employing the established particle-grid interpolation schemes in a conventional MLS-MPM framework. We demonstrate the efficacy of our approach with a host of simulation examples governed by magnetic-mechanical coupling effects, ranging from magnetic deformable bodies to magnetic viscous fluids with nonlinear elastic constitutive laws.

*joint first authors
†corresponding authors

Authors' addresses: Yuchen Sun, yuchen_sun@pku.edu.cn, CFCS, Peking University, China; Xingyu Ni, nixy@pku.edu.cn, CFCS, Peking University, China; Bo Zhu, bo.zhu@dartmouth.edu, Dartmouth College, United States of America; Bin Wang, binwangbuaa@gmail.com, Beijing Institute for General Artificial Intelligence, China; Baoquan Chen, baoquan@pku.edu.cn, CFCS, Peking University, China.

0730-0301/2021/12-ART205 $15.00
https://doi.org/10.1145/3478513.3480541

CCS Concepts: • **Computing methodologies** → **Physical simulation**; • **Applied computing** → *Physics*.

Additional Key Words and Phrases: magnetic simulation, nonlinear magnetization, Material Point Method (MPM)

ACM Reference Format:
Yuchen Sun, Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2021. A Material Point Method for Nonlinearly Magnetized Materials. *ACM Trans. Graph.* 40, 6, Article 205 (December 2021), 13 pages. https://doi.org/10.1145/3478513.3480541

1 INTRODUCTION

Physical systems in the natural world exhibit interactions at distinct spatial and temporal scales. Magnetism is a representative example of this long-distance and multi-scale interaction. In engineering communities, leveraging magnetic interactions has become an effective control mechanism for its instant interaction speed and fast magnetomechanical coupling via manipulating the electrically charged particles, opening up a broad horizon of applications such as remote robotic actuation, wireless control, and precise 3D printing. Devising computational tools to simulate and optimize such magnetic phenomena is essential for both engineering practices and scientific research. At the same time, these visually appealing material motions have also drawn attention from researchers in the computer graphics community. A large number of efforts have been devoted to the simulation of magnetic materials, ranging from magnetic rigid bodies [Kim and Han 2020; Kim et al. 2018; Thomaszewski et al. 2008] to ferrofluids [Huang et al. 2019; Huang and Michels 2020; Ishikawa et al. 2013; Ni et al. 2020]. Specifically, over the course of the ferrofluid simulation study, researchers have been tackling the challenges of handling the intricate, nonlinear interactions between the magnetic fields and the various material properties such as viscosity, incompressibility, and surface tension.

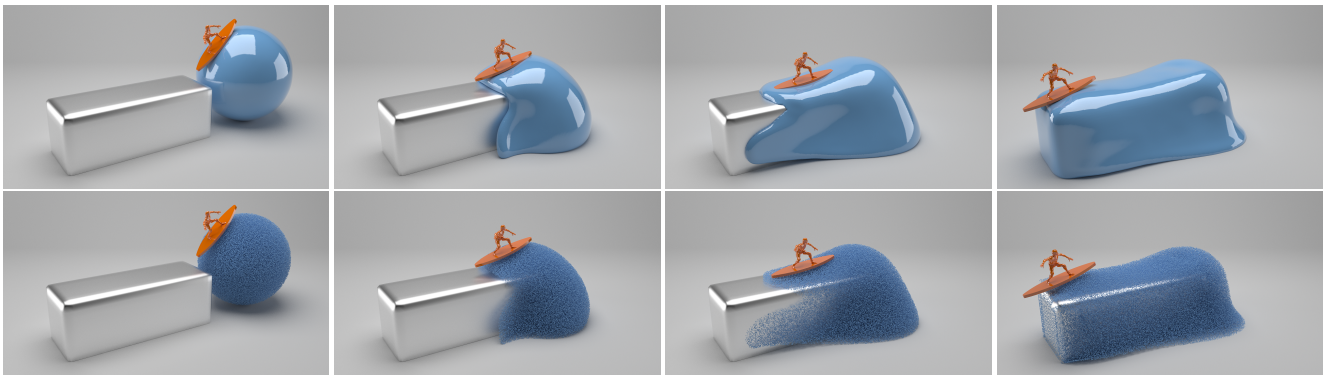

Fig. 2. A lump of magnetic viscous fluid is attracted by a magnet, bearing a rigid body.
Top: rendered by a reconstructed mesh; bottom: rendered by original particles.

Simulating nonlinearly magnetized materials in a physically correct way is tantalizing. A nonlinear model, as its many counterparts in physical simulation such as (nonlinear) elasticity, produces rich unique geometric and dynamic characteristics, which would have been infeasible to obtain using a linear model. Theoretically, a strong magnetic field does saturate magnetization in the real world that deviates from linear behaviors. However, among the many advancing simulation techniques in physics simulation, nonlinear magnetization has remained a largely unexplored research domain in physics simulation due to its multi-faceted computational challenges. First, it is difficult to directly discretize and solve nonlinear PDEs of magnetization using the existing high-performance linear solvers. Second, it is challenging to devise a data structure that can handle both microscopic properties and macroscopic quantities. Taking the ferrofluid as an example, when a magnet approaches, spikes on the liquid surface become sharper and denser. In the nonlinear model, the strength of magnetic forces can get saturated, thus slowing down the deformation rate when the magnet gets extremely close. On the contrary, in a linear model situation, magnetic forces increase infinitely, which will blow up the simulation eventually. To deal with the instability issue there, the only practical way is to carefully guide the trajectory of magnets.

Several pioneering work on exploring the nonlinearly magnetized materials focus on simulating nonlinear magnetic interactions among rigid bodies. In particular, Kim et al. [2020; 2018] defined the effective magnetic field as a sum of four components, and the evolution of individual components is governed by equations originated from micromagnetics. This method well resolved the instability issue for rigid bodies where the relative positions of internal particles do not change. Motivated by their work, we aim to devise a more comprehensive and versatile simulation framework to facilitate a broader range of magnetic phenomena ranging from fluids to deformable solids, without any parameters to tune.

We leverage Langevin's nonlinear theory of paramagnetism [Coey 2010], which was established by Paul Langevin to bridge the microscopic particles and macroscopic quantities in statistical physics. Intuitively, the microscopic information can be carried by Lagrangian granular discrete volumes. In constrast, the macroscopic field functions are more suitable to be defined over an Eulerian data structure such as a background grid. To this end, we adopt a hybrid Eulerian-Lagrangian discretization where a magnetic-mechanical system can be efficiently solved on a Cartesian grid, enhanced with a set of moving particles carrying the number density of magnetic microscopic particles. We choose to use the material point method (MPM) as our computational framework. MPM has received extensive attention in the computer graphics community in recent years [Jiang et al. 2016; Stomakhin et al. 2013]. Specifically, we leverage the inherent computational merits of a moving-least-squares (MLS) MPM scheme [Hu et al. 2018] to automatically process the objects' collision and unify multi-physics simulations under its Eulerian-Lagrangian framework. By incorporating the magnetic interactions into the MLS-MPM framework, we achieve an expressive numerical tool for simulating various kinds of magnetic viscoelastic phenomena, such as magnetic soft bodies and magnetic clay, which were not feasible for previous approaches.

We summarize our technical contributions as

- The first unified approach to modeling a broad range of nonlinear magnetic phenomena, with a particular focus on magnetic viscoelastic materials, based on Langevin's theory of paramagnetism;
- An efficient Newton-based algorithm to solve the nonlinear equations of magnetization on a Cartesian grid;
- An effective numerical scheme to incorporate the nonlinear magnetic forces into the MLS-MPM framework.

2 RELATED WORK

The existing methods of physically simulating magnetic effects can be divided into two categories: the Eulerian approaches and the Lagrangian approaches.

Eulerian Magnetic Simulation. Most of the Eulerian ways to simulate magnetic substance are developed for fluid solvers. Beginning with the pioneering work of Oldenburg et al. [2000], a surge of literature has been devoted to the development of Eulerian numerical schemes to simulate ferrofluids in a computational physical setting. Liu et al. [2011] simulated ferrofluid droplets by tracking the surface using a particle level-set method and solving the magnetic field on a Cartesian staggered grid by a finite volume approach. Zhu et al.

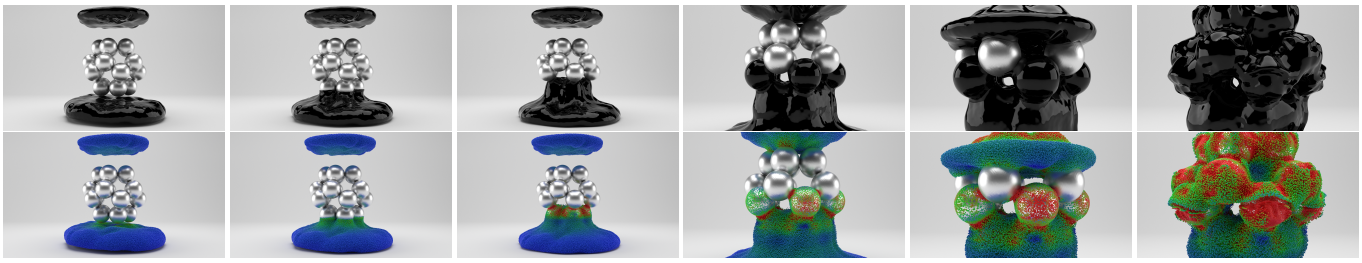

Fig. 3. Two lumps of magnetic viscous fluid are attracted by 20 buckyballs.
Top: rendered by a reconstructed mesh; bottom: rendered by original particles, and the color signifies the magnitude of the magnetization intensity.

[2011] employed a similar way to simulate ferrofluid droplets but used a level-set method instead, while the volume-of-fluid method was also attempted by researchers as an option [Ghaffari et al. 2015; Shi et al. 2014]. The state-of-the-art work was present by Ni et al. [2020] in visual computing community, which employed a level-set method to track the interface of the magnetic substance and solved the magnetic field in a Cartesian marker-and-cell (MAC) grid [Harlow and Welch 1965] by the finite difference method. Their solver can successfully produce the dynamic motion of various magnetic substances, including ferrofluids and magnetic solids, but may suffer instability because of its assumption of linear magnetization, just like most of the aforementioned Eulerian work.

Lagrangian Magnetic Simulation. Based on the Lagrangian perspective, Thomaszewski et al. [2008] first introduced the magnetic effects into the computer graphics field, integrated with rigid-body simulations. In their work, the magnetic fields are generated by permanent magnets, then the magnetization is determined by the magnetic fields based on the linear constitutive relation, with the induced magnetic field ignored. Kim et al. [2020; 2018] proposed magnetization dynamics inspired by micromagnetics to gain a more convincing physical property and to limit the strength of magnetic forces and torques, which allows for nonlinear magnetization and produces both stable and efficient simulations of magnetic rigid bodies. For non-rigid magnet simulation, Ishikawa et al. [2013] employed a smoothed-particle-hydrodynamics (SPH) approach to simulating ferrofluids by treating each particle as a magnetic dipole. A procedural method was devised to generate the spike structure on the surface. Huang et al. [2019] invented an accurate large-scale SPH simulation scheme by incorporating the fast multipole method (FMM) into the Lagrangian framework to model the magnetic evolution, which produces visually captivating effects. They also devised a novel surface-only approach [2020] to simulating incompressible, inviscid, and linearly magnetizable ferrofluids, which enables more complex and realistic magnetic simulations.

Material Point Method. The material point method (MPM) [Sulsky et al. 1995] is a hybrid Lagrangian/Eulerian discretization scheme for solid mechanics. It is recognized as a generalization of the fluid implicit particle (FLIP) method [Brackbill and Ruppel 1986], which is widely used for liquid animation [Zhu and Bridson 2005]. The framework of MPM has been used for simulating a wide range of materials, including snow [Stomakhin et al. 2013], foam [Ram et al. 2015; Yue et al. 2015], sand [Daviet and Bertails-Descoubes 2016; Klár et al. 2016], cloth [Jiang et al. 2017], fracture [Wretborn et al. 2017], cutting [Hu et al. 2018], and solid-fluid mixture [Gao et al. 2018; Stomakhin et al. 2014; Tampubolon et al. 2017]. Hu et al. [2018] incorporated a moving-least-squares (MLS) version of the MPM, which is consistent with a Galerkin-style weak form discretization of the governing equations. Due to its easy implementation and friendly performance optimization, the MLS-MPM scheme is becoming popular in the recent years as a conventional pipeline of the material point method.

3 PHYSICAL MODELS

In this section, we will introduce the physical models which lay the foundation of our method. The main quantities involved in the paper are presented in Table 1, the first part of which is for continuous quantities, and the second part of which is for discrete ones.

Naming convention. We symbolize vectors and second-order tensors using bold letters (such as $\boldsymbol{H}$ and $\boldsymbol{\sigma}$) and symbolize scalars using italic letters (such as H and μ_0). In particular, if a bold letter is used to stand for a vector, the corresponding italic letter indicates its magnitude, omitting the information of directions (e.g., $H = |\boldsymbol{H}|$).

3.1 Nonlinear Magnetization

As an essential part of our solver, we will expand on the nonlinear magnetization formula from the work of Ni et al. [2020]. To make the paper self-contained, we will rewrite a detailed derivation of the formula with the necessary expansion.

The magnetic theory studies the interactions among three fields – *magnetic field intensity* $\boldsymbol{H}$, *magnetization intensity* $\boldsymbol{M}$ and *magnetic induction intensity* $\boldsymbol{B}$. They are constrained by

$$\boldsymbol{B} = \mu_0(\boldsymbol{H} + \boldsymbol{M}), \tag{1}$$

in which μ_0 is the constant *vacuum permeability*.

Among the three fields, $\boldsymbol{M}$ is the special one, because of its distinct microscopic significance. Every microscopic particle, e.g., molecule, atom and ion, can be considered as a dipole with an invariant magnetic moment $\boldsymbol{m}_l$ indicating its magnetic performance. The macroscopic magnetic performance is described by the vector field $\boldsymbol{M}$:

$$\boldsymbol{M} = \boldsymbol{M}(\boldsymbol{x}) = \lim_{\Delta V \to 0} \frac{\sum_l \boldsymbol{m}_l}{\Delta V}, \tag{2}$$

which accumulates across all particles within an infinitesimal ΔV-volume domain neighboring the point $\boldsymbol{x}$.

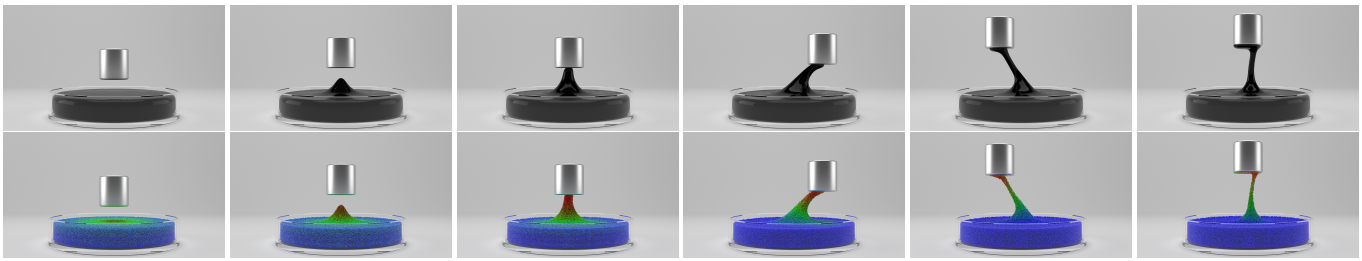

Fig. 4. A lump of magnetic viscous fluid in a container is attracted by a magnet, generating a spike.
Top: rendered by a reconstructed mesh; bottom: rendered by original particles, and the color signifies the magnitude of the magnetization intensity.

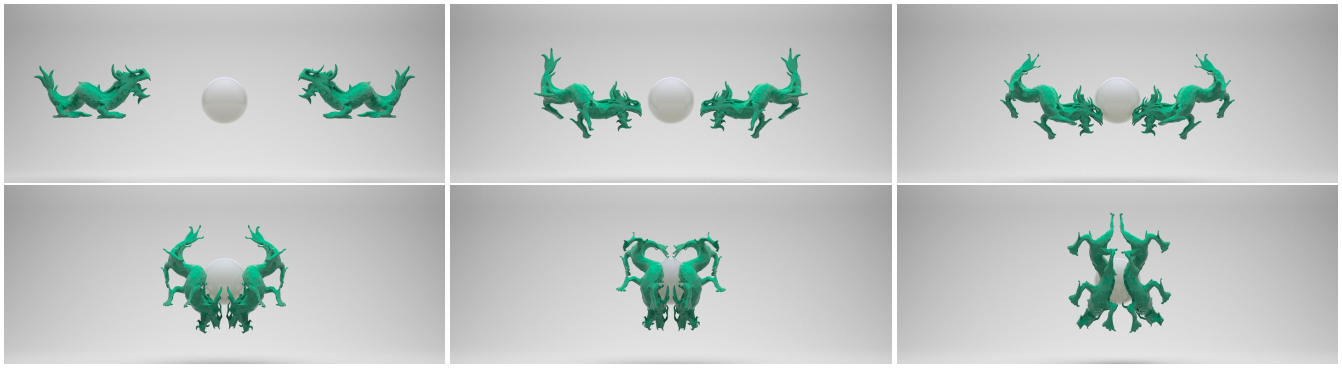

Fig. 5. Two magnetized elastic dragons are 'playing' with a bead-shaped magnet.

For paramagnetic materials, the orientation of microscopic particles is totally random if no magnetic field is applied (i.e., $H = 0$), which leads to $\boldsymbol{M} = \boldsymbol{0}$ everywhere. However, when $H > 0$, the micro-particles will rotate and their magnetic moments will tend to align, with high probability, with the direction of the magnetic field. Langevin's theory of paramagnetism [Rosensweig 1985], induced by statistical physics, states that

$$\overline{\boldsymbol{m}}_l = m_l \, L\left(\frac{\mu_0 m_l H}{k_\mathrm{B} T}\right) \frac{\boldsymbol{H}}{H}, \tag{3}$$

with $\overline{\boldsymbol{m}}_l$ as the expectation of $\boldsymbol{m}_l$ in equilibrium, k_B as the *Boltzmann constant* and T as the ambient temperature. Here, $L(\cdot)$, which denotes the *Langevin function*, has the following form:

$$L(\alpha) = \coth \alpha - \frac{1}{\alpha}. \tag{4}$$

It is not hard to prove that the Langevin function converges to 1 as α goes to infinity, which is shown in Figure 6.

Since all the microscopic particles are identical, we can ignore subscript and use m and $\overline{\boldsymbol{m}}(\boldsymbol{x})$ to represent the magnitude of each magnetic moment and the average magnetic moment within a particular infinitesimal domain, respectively. With $n = n(\boldsymbol{x})$ indicating the number density of micro-particles, we will acquire

$$\begin{aligned} \boldsymbol{M} &= n\overline{\boldsymbol{m}} \\ &= nm \, L\left(\frac{\mu_0 m H}{k_\mathrm{B} T}\right) \frac{\boldsymbol{H}}{H}. \end{aligned} \tag{5}$$

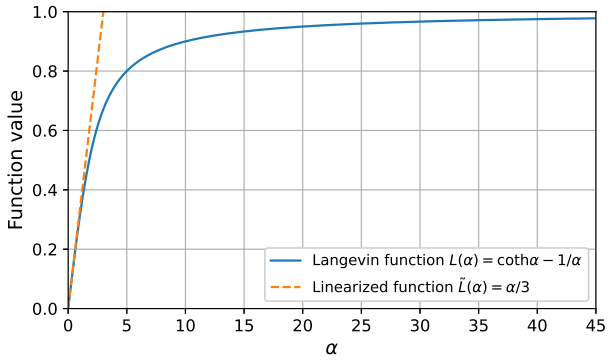

Fig. 6. The curve of the Langevin function $L(\alpha)$ and its linearized version $\tilde{L}(\alpha) = \alpha/3$. We can see that $\tilde{L}(\alpha)$ deviates a lot as α gets larger.

Linearization. The number density n remains constant when a material is incompressible. Given constants $k_1 = nm$ and $k_2 = \mu_0 m / k_\mathrm{B} T$, (5) can be rewritten as

$$M = k_1 \left(\coth k_2 H - \frac{1}{k_2 H}\right) \tag{6}$$

without regard to the direction. If the intensity of the magnetic field is not that high, which suggests $k_2 H \ll 1$, expanding the coth

Table 1. The main quantities involved in describing a magnetic field.

Notation†	Order‡	Meaning
$\boldsymbol{x}$	1	The position vector
$\boldsymbol{H}(\boldsymbol{x})$	1	The *magnetic field intensity*
$\boldsymbol{M}(\boldsymbol{x})$	1	The *magnetization intensity*
$\boldsymbol{B}(\boldsymbol{x})$	1	The *Magnetic induction intensity*
μ_0	0	The *vacuum permeability*
k_B	0	The *Boltzmann constant*
T	0	The ambient temperature
$L(\alpha)$	0	The *Langevin function*
m	0	The magnitude of magnetic moments
$n(\boldsymbol{x})$	0	The micro-particle number density
$\boldsymbol{\sigma}(\boldsymbol{x})$	2	The *Cauchy stress tensor*
$\boldsymbol{T}_\mathrm{m}(\boldsymbol{x})$	2	The *Maxwell stress tensor*
Δx	0	The grid spacing
Δt	0	The time interval
$\boldsymbol{x}_i$	1	The position of the i-th grid node
$\boldsymbol{x}_p$	1	The position of the p-th particle
$N_i(\boldsymbol{x})$	0	The *B-spline basis function* centered at $\boldsymbol{x}_i$
$\boldsymbol{H}_i$	1	$\boldsymbol{H}$ saved at grid node i
$\boldsymbol{M}_i$	1	$\boldsymbol{M}$ saved at grid node i
$\boldsymbol{B}_i$	1	$\boldsymbol{B}$ saved at grid node i
$\boldsymbol{T}_i$	2	$\boldsymbol{T}_\mathrm{m}$ computed at grid node i
$\boldsymbol{t}_i$	1	$\nabla \cdot \boldsymbol{T}_\mathrm{m}$ computed at grid node i
n_i	0	The number density saved at grid node i
m_i	0	The mass saved at grid node i
$\boldsymbol{v}_i$	1	The velocity saved at grid node i
n_p	0	# of micro-particles carried by particle p

† Alternatively, the subscript i can be replaced by j.
‡ The order of tensors, with 0 indicating scalars, 1 indicating vectors and 2 indicating matrices.

function in (6), we will obtain

$$\begin{aligned} M &= k_1 \left[\frac{1}{k_2 H} + \frac{k_2 H}{3} + o\left((k_2 H)^3 \right) \right] - \frac{k_1}{k_2 H} \\ &= \frac{k_1 k_2}{3} H + k_1\, o\left((k_2 H)^3 \right) \\ &\approx \frac{k_1 k_2}{3} H, \end{aligned} \tag{7}$$

which rolls back to the linear assumption, just as the curves of $L(\alpha)$ and $\tilde{L}(\alpha)$ show in Figure 6, with α set as $k_2 H$. Now we can gain insight into the effect of k_1 and k_2. While k_1 determines the saturation magnetization, the product of k_1 and k_2 determines the slope of the magnetizing curve's approximately linear part.

Despite that the convergence of the Langevin function can be achieved by just clamping the curve in the linearized situation, the latter will involve non-smoothness in magnetization, which is not only non-physical, but also makes the magnetic system hard to solve because of non-differentiability.

3.2 Magnetic Field Evolution

The evolution of a magnetic field is governed by Maxwell's equations [Feynman et al. 2011]:

$$\begin{cases} \nabla \cdot \boldsymbol{B} = 0, & \text{(8a)} \\ \nabla \times \boldsymbol{H} = \boldsymbol{j}_\mathrm{f} + \dfrac{\partial \boldsymbol{D}}{\partial t}. & \text{(8b)} \end{cases}$$

Here $\boldsymbol{j}_\mathrm{f}$ is the *electric current density* of free charges and $\boldsymbol{D}$ is the *electric displacement field* affecting the magnetic field by electromagnetic induction. With (1) substituted in, (8a) is further written as

$$\nabla \cdot (\boldsymbol{H} + \boldsymbol{M}) = \boldsymbol{0}. \tag{9}$$

In practice, we usually extract the external part from the total magnetic field, which is denoted by the subscript 'ext'. The external magnetic field may be induced by some permanent magnets and electric currents, and it is determined during runtime. This field should also satisfy Maxwell's equations:

$$\begin{cases} \nabla \cdot \boldsymbol{B}_\mathrm{ext} = 0, & \text{(10a)} \\ \nabla \times \boldsymbol{H}_\mathrm{ext} = \boldsymbol{j}_{\mathrm{f}\,\mathrm{ext}} + \dfrac{\partial \boldsymbol{D}_\mathrm{ext}}{\partial t}. & \text{(10b)} \end{cases}$$

Therefore, the quantities of the internal magnetic field, denoted by subscript 'int', can be defined as

$$\begin{cases} \boldsymbol{H}_\mathrm{int} = \boldsymbol{H} - \boldsymbol{H}_\mathrm{ext}, & \text{(11a)} \\ \boldsymbol{B}_\mathrm{int} = \boldsymbol{B} - \boldsymbol{B}_\mathrm{ext}, & \text{(11b)} \\ \boldsymbol{j}_{\mathrm{f}\,\mathrm{int}} = \boldsymbol{j}_\mathrm{f} - \boldsymbol{j}_{\mathrm{f}\,\mathrm{ext}}, & \text{(11c)} \\ \boldsymbol{D}_\mathrm{int} = \boldsymbol{D} - \boldsymbol{D}_\mathrm{ext}. & \text{(11d)} \end{cases}$$

For (nearly) non-conductive magnetic objects, it is reasonable to assume steady-state electric displacement $\partial \boldsymbol{D}_\mathrm{int}/\partial t = 0$ and zero free current, $\boldsymbol{j}_{\mathrm{f}\,\mathrm{int}} = 0$, which leads to

$$\nabla \times \boldsymbol{H}_\mathrm{int} = 0 \tag{12}$$

by subtracting (10b) from (8b). Since $\boldsymbol{H}_\mathrm{int}$ is conservative, we can let $\boldsymbol{H}_\mathrm{int} = -\nabla\varphi$, with φ as a potential function, which, like (11a), can be further substituted into (9) to obtain

$$\nabla \cdot (\boldsymbol{H}_\mathrm{ext} - \nabla\varphi + \boldsymbol{M}) = \boldsymbol{0}, \tag{13}$$

where $\boldsymbol{M}$ is a nonlinear function of $\boldsymbol{H} = \boldsymbol{H}_\mathrm{ext} - \nabla\varphi$ as formulated by (5). Solving (13) for φ, the total magnetic field (i.e., $\boldsymbol{H}$, $\boldsymbol{M}$ and $\boldsymbol{B}$) can be computed directly.

Magnetic force. We compute the magnetic force $\boldsymbol{f}_\mathrm{m}$ by taking divergence of the *Maxwell stress tensor* $\boldsymbol{T}_\mathrm{m}$ in matter. The Einstein–Laub form [1908] of this tensor is

$$\boldsymbol{T}_\mathrm{m} = \boldsymbol{B} \otimes \boldsymbol{H} - \frac{\mu_0}{2} H^2 \boldsymbol{I}, \tag{14}$$

in which $\boldsymbol{I}$ is the second-order unit tensor, so the precise formula of the magnetic force is

$$\boldsymbol{f}_\mathrm{m} = \nabla \cdot \boldsymbol{T}_\mathrm{m}, \tag{15}$$

known as the *Kelvin force*.

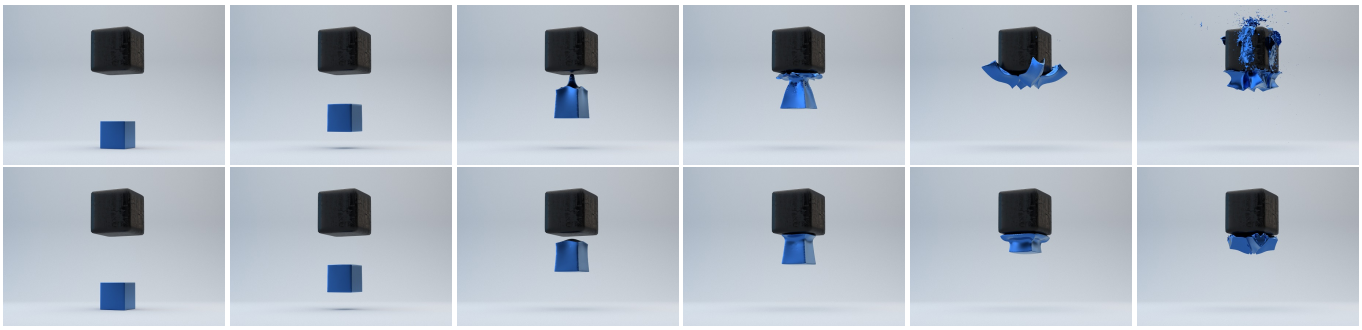

Fig. 7. Top: a linearly magnetized cube; below: a nonlinearly magnetized cube, in which the cubes are attracted by a magnet.

3.3 Magnetic Substance Evolution

The magnetic substance evolution is governed by the basic momentum equation:

$$\rho \frac{\mathrm{D}}{\mathrm{D}t}\boldsymbol{v} = \nabla \cdot \boldsymbol{\sigma} + \nabla \cdot \boldsymbol{T}_{\mathrm{m}} + \rho \boldsymbol{g} \tag{16}$$

with ρ as the mass density, $\boldsymbol{v}$ as the velocity, $\boldsymbol{\sigma}$ as the *Cauchy stress tensor*, $\boldsymbol{T}_{\mathrm{m}}$ as the Maxwell stress tensor and $\boldsymbol{g}$ as the gravitational acceleration. While the Cauchy stress tensor varies from one material to another, the Maxwell stress tensor is distributed everywhere and not explicitly dependent on material properties. Here, we discuss the Cauchy stress tensor of two kinds of materials we simulate: elastic solid and viscoelastic fluid.

Elastic solid. Elastic solid only has elasticity. Therefore, the Cauchy stress tensor for elastic solid $\boldsymbol{\sigma}^{\mathrm{S}}$ only has the elastic component $\boldsymbol{\sigma}^{\mathrm{E}}$, i.e.,

$$\boldsymbol{\sigma}^{\mathrm{S}} = \boldsymbol{\sigma}^{\mathrm{E}}. \tag{17}$$

We adopt the *Neo–Hookean constitutive model* [Sifakis and Barbic 2012] for elasticity, which expresses the elastic component $\boldsymbol{\sigma}^{\mathrm{E}}$ as

$$\boldsymbol{\sigma}^{\mathrm{E}} = \mu J^{-1}\left(\boldsymbol{F}\boldsymbol{F}^{T} - \boldsymbol{I}\right) + \lambda J^{-1}\log J, \tag{18}$$

where μ and λ are the Lamé's parameters, and $\boldsymbol{F}$ is the deformation gradient with $J = |\boldsymbol{F}|$ indicating the rate of volume change.

Viscoelastic Fluid. Viscoelastic fluid has both viscosity and elasticity. Therefore, the Cauchy stress tensor $\boldsymbol{\sigma}^{\mathrm{F}}$ for viscoelastic fluid has two components: the viscous component $\boldsymbol{\sigma}^{\mathrm{N}}$ and the elastic component $\boldsymbol{\sigma}^{\mathrm{E}}$:

$$\boldsymbol{\sigma}^{\mathrm{F}} = \boldsymbol{\sigma}^{\mathrm{N}} + \boldsymbol{\sigma}^{\mathrm{E}}. \tag{19}$$

We adopt the Newtonian model for viscosity, which expresses the viscous component σ^{N} as

$$\boldsymbol{\sigma}^{\mathrm{N}} = \mu^{\mathrm{N}}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}^{\mathrm{T}}\right), \tag{20}$$

where μ^{N} is the viscous coefficient. The elastic component $\boldsymbol{\sigma}^{\mathrm{E}}$ for fluid is given by

$$\boldsymbol{\sigma}^{\mathrm{E}} = -K(1-J)\boldsymbol{I}, \tag{21}$$

in which K is the bulk modulus. Therefore, we only need to take care of the rate of volume change J without the deformation gradient [Tampubolon et al. 2017].

4 NUMERICAL ALGORITHMS

4.1 Magnetic Field

We discretize the concerned domain by a node-centered collocated Cartesian grid, just identical with the grid of MPM as shown in Section 4.3. The magnetic field on the grid boundary is set to be $\nabla\varphi \cdot \boldsymbol{n} = 0$, where $\boldsymbol{n}$ is the normal vector of the boundary. This Neumann boundary condition, also known as *magnetic shielding*, which is commonly adopted to replace the boundary condition $\varphi \longrightarrow 0$ at infinity.

Supposed that the number of grid nodes is s, the scalar field φ can be discretized as an s-dimensional column vector denoted $\hat{\boldsymbol{\varphi}}$. Similarly, $\boldsymbol{H}_{\mathrm{ext}}$, $\boldsymbol{M}$ and other vector fields can be discretized as $\hat{\boldsymbol{H}}_{\mathrm{ext}}$, $\hat{\boldsymbol{M}}$ and so on, which are ds-dimensional column vectors with d as the dimension of original vector. Then, (13) can be discretized as

$$\boldsymbol{F}(\hat{\boldsymbol{\varphi}}) = -\boldsymbol{G}^{\mathrm{T}}(\hat{\boldsymbol{H}}_{\mathrm{ext}} - \boldsymbol{G}\hat{\boldsymbol{\varphi}} + \hat{\boldsymbol{M}}) = \boldsymbol{0}, \tag{22}$$

in which $\boldsymbol{G} \in \mathbb{R}_{ds\times s}$ is the discretized form of the gradient operator with its transpose $\boldsymbol{G}^{\mathrm{T}}$ as one of the (negative) divergence operator (see the work of Robinson-Mosher et al. [2008]), and $\hat{\boldsymbol{M}}$ is a nonlinear function of $\hat{\boldsymbol{H}} = \hat{\boldsymbol{H}}_{\mathrm{ext}} - \boldsymbol{G}\hat{\boldsymbol{\varphi}}$.

We employ the Newton–Raphson method to solve this nonlinear equation. We start with the $\hat{\boldsymbol{\varphi}}^{0}$ from the last frame as an initial guess. For the n-th Newton iteration, we first calculate the derivative of $\boldsymbol{F}(\hat{\boldsymbol{\varphi}})$ at $\hat{\boldsymbol{\varphi}}^{n-1}$, then we hope to decrease the current value of $\boldsymbol{F}(\hat{\boldsymbol{\varphi}})$ at $\hat{\boldsymbol{\varphi}}^{n}$:

$$\begin{cases} \boldsymbol{F}'(\hat{\boldsymbol{\varphi}}^{n-1})\,\Delta\hat{\boldsymbol{\varphi}}^{n} = -\boldsymbol{F}(\hat{\boldsymbol{\varphi}}^{n-1}), & \text{(23a)} \\ \hat{\boldsymbol{\varphi}}^{n} = \hat{\boldsymbol{\varphi}}^{n-1} + \lambda^{n}\,\Delta\hat{\boldsymbol{\varphi}}^{n}, & \text{(23b)} \end{cases}$$

where λ^{n} is the step size determined by line search, and

$$\boldsymbol{F}'(\hat{\boldsymbol{\varphi}}) = \boldsymbol{G}^{\mathrm{T}}\left(\boldsymbol{I} + \frac{\partial \hat{\boldsymbol{M}}}{\partial \hat{\boldsymbol{H}}}\right)\boldsymbol{G}. \tag{24}$$

Here, $\partial\hat{\boldsymbol{M}}/\partial\hat{\boldsymbol{H}}$ is a block diagonal matrix with s blocks. Each block is a $d \times d$ square matrix. Taking consideration of quantities saved at grid nodes, we use $\boldsymbol{x}_i$ to denote the position of the i-th grid node and define $\boldsymbol{M}_i$, $\boldsymbol{H}_i$ and n_i to be corresponding quantities saved at grid node i. Then $\partial\boldsymbol{M}_i/\partial\boldsymbol{H}_i$ is just the i-th block of $\partial\hat{\boldsymbol{M}}/\partial\hat{\boldsymbol{H}}$. Therefore, We can get $\mathrm{d}\hat{\boldsymbol{M}}/\mathrm{d}\hat{\boldsymbol{H}}$ by computing $\partial\boldsymbol{M}_i/\partial\boldsymbol{H}_i$ for every grid node i. First, we obtain

$$\boldsymbol{M}_i = n_i m\, L\left(\frac{\mu_0 m H_i}{k_{\mathrm{B}}T}\right)\frac{\boldsymbol{H_i}}{H_i} \tag{25}$$

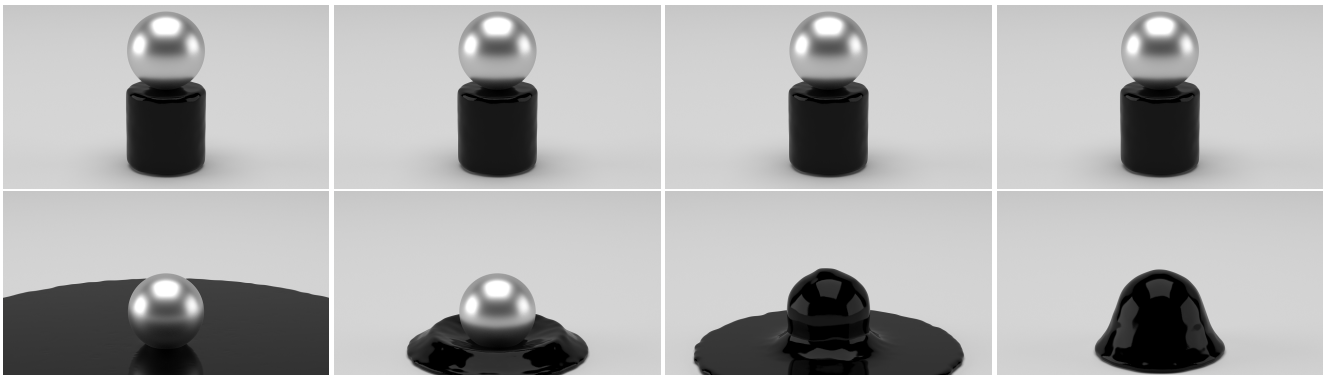

Fig. 8. Four pairs of magnetic viscous fluid and magnets. From left to right: low viscosity-low intensity of magnetization; high viscosity-low intensity of magnetization; low viscosity-high intensity of magnetization; high viscosity-high intensity of magnetization;

according to (5). Using the chain rule in differentiation, the derivative of the α-th component of $\boldsymbol{M}_i$ with respect to the β-th component of $\boldsymbol{H}_i$ can be calculated as follows:

$$\begin{aligned}\frac{\partial \boldsymbol{M}_{i\alpha}}{\partial \boldsymbol{H}_{i\beta}} &= \frac{n_i \mu_0 m^2}{k_\mathrm{B} T} L'\left(\frac{\mu_0 m H_i}{k_\mathrm{B} T}\right) \frac{H_{i\alpha} H_{i\beta}}{H_i^2} \\ &+ n_i m L\left(\frac{\mu_0 m H_i}{k_\mathrm{B} T}\right) \frac{\delta_{\alpha\beta} H_i^2 - H_{i\alpha} H_{i\beta}}{H_i^3}.\end{aligned} \tag{26}$$

Thanks to the monotony of the Langevin function, the matrix $\partial \hat{\boldsymbol{M}} / \partial \hat{\boldsymbol{H}}$ is symmetric positive semi-definite, and the matrix $\boldsymbol{F}'(\hat{\boldsymbol{\varphi}})$ is symmetric positive definite. The proof is given in Appendix A.1. Therefore, we can use the preconditioned conjugate gradient solver in each iteration of the Newton–Raphson method.

The above procedure is summarized in Algorithm 1. If we perform linearization as mentioned in Section 3.1, it seems to suggest that the algorithm will converge in one iteration, but it will suffer from instability.

ALGORITHM 1: Magnetic Field Computation

Input: The discretized number density n_i of magnetic microscopic particles and the discretized external magnetic field, denoted by $\hat{\boldsymbol{H}}_\mathrm{ext}$.
Output: The magnetic field quantities, i.e., $\boldsymbol{H}_i$, $\boldsymbol{M}_i$ and $\boldsymbol{B}_i$, at each grid node.

```
φ̂ ← 0 or that from the last frame;
ε ← the convergence threshold of each Newton step;
while |F(φ̂)| > ε do
    Assemble matrix ∂M̂/∂Ĥ from blocks ∂M_i/∂H_i;
    F'(φ̂) ← G^T (I + ∂M̂/∂Ĥ) G;
    Solve F'(φ̂) Δφ̂ = −F(φ̂) for Δφ̂ by the PCG method;
    λ ← 1;
    while |F(φ̂ + λ Δφ̂)| ≥ |F(φ̂|) do
        λ ← λ/2;
    end
    φ̂ ← φ̂ + λ Δφ̂;
end
Ĥ ← Ĥ_ext + G φ̂;
Split Ĥ for H_i and compute M_i and B_i by corresponding equations;
```

4.2 Magnetic Force

The governing equation of magnetic substance evolution is given by (16). In practice, We concurrently solve elastic force and magnetic force toward an equilibrium state. The numerical solution for the elastic force is given by the traditional MPM. We now discuss how to apply the magnetic force numerically.

By taking the weak form and using spatial and temporal grid discretization, (16) can be written as

$$\begin{aligned}\frac{\left[(m\boldsymbol{v})_i^{n+1} - (m\boldsymbol{v})_i^n\right]}{\Delta t} &= \int_{\partial\Omega^{t^n}} N_i(\boldsymbol{x}) \boldsymbol{\sigma}(\boldsymbol{x}, t^n)\, \mathrm{d}\boldsymbol{s}(\boldsymbol{x}) \\ - \int_{\Omega^{t^n}} \boldsymbol{\sigma}(\boldsymbol{x}, t^n) \nabla N_i(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} &+ \int_{\Omega^{t^n}} N_i(\boldsymbol{x}) \nabla \cdot \boldsymbol{T}_\mathrm{m}(\boldsymbol{x}, t^n)\, \mathrm{d}\boldsymbol{x} + m_i \boldsymbol{g}\end{aligned} \tag{27}$$

with Δt as the time interval, t^n as the n-th time point, Ω as the material domain, and $\mathrm{d}\boldsymbol{s}$ as the area element. The grid basis function $N_i(x)$ is the dyadic product of one-dimensional quadratic B-spline as in the work of Steffen et al. [2008]:

$$N_i(\boldsymbol{x}) = N\left(\frac{x_1 - x_{i1}}{\Delta x}\right) N\left(\frac{x_2 - x_{i2}}{\Delta x}\right) N\left(\frac{x_3 - x_{i3}}{\Delta x}\right), \tag{28}$$

$$N(x) = \begin{cases} \frac{3}{4} - |x|^2, & 0 \le |x| < \frac{1}{2}, \\ \frac{1}{2}\left(\frac{3}{2} - |x|^2\right)^2, & \frac{1}{2} \le |x| < \frac{3}{2}, \\ 0, & \frac{3}{2} \le |x|, \end{cases} \tag{29}$$

with $\boldsymbol{x} = (x_1, x_2, x_3)$ as the evaluation position, i as the grid index, $\boldsymbol{x} = (x_{i1}, x_{i2}, x_{i3})$ as the position of the grid node i and Δx as the grid spacing. The discretization of the first term $\int_{\partial\Omega^{t^n}} N_i(\boldsymbol{x}) \boldsymbol{\sigma}(\boldsymbol{x}, t^n)\, \mathrm{d}\boldsymbol{s}(\boldsymbol{x})$ and second term $\int_{\Omega^{t^n}} \boldsymbol{\sigma}(\boldsymbol{x}, t^n) \nabla N_i(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$ is explained in the work of Jiang et al. [2016] in detail. Here we show how to discretize the third term, namely the magnetic force $\int_{\Omega^{t^n}} N_i(\boldsymbol{x}) \nabla \cdot \boldsymbol{T}_\mathrm{m}(\boldsymbol{x}, t^n)\, \mathrm{d}\boldsymbol{x}$.

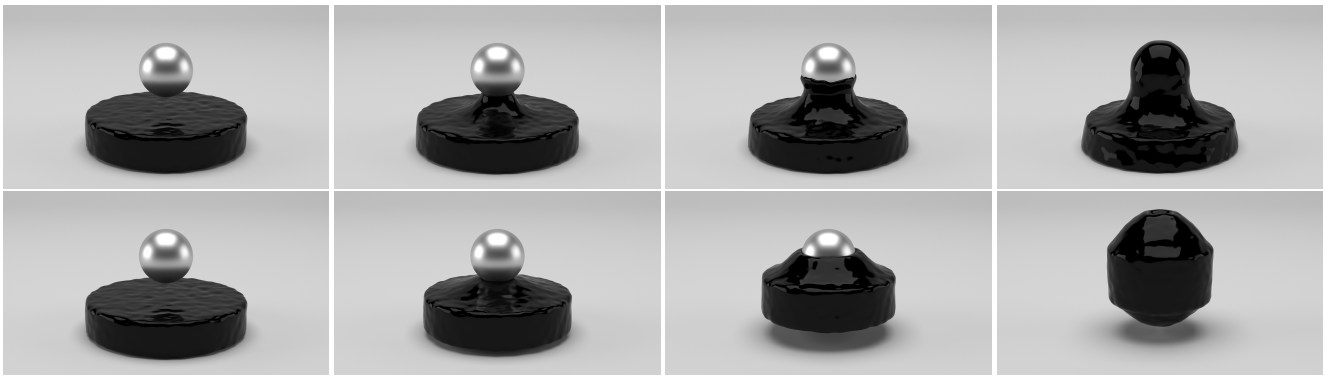

Fig. 9. Top: attraction caused by magnetic force; below: attraction caused by artificial force.

First, we split up the continuous material space Ω^{t^n} into the discrete grid cells Ω_j which are saturated by material points:

$$\int_{\Omega^{t^n}} N_i(\boldsymbol{x}) \nabla \cdot \boldsymbol{T}_{\mathrm{m}}(\boldsymbol{x}, t^n)\, \mathrm{d}\boldsymbol{x} \approx \sum_j \int_{\Omega_j} N_i(\boldsymbol{x}) \nabla \cdot \boldsymbol{T}_{\mathrm{m}}(\boldsymbol{x}, t^n)\, \mathrm{d}\boldsymbol{x}. \quad (30)$$

Second, within a grid cell j, the Maxwell stress tensor $\boldsymbol{T}_{\mathrm{m}}$ is viewed as a constant matrix $\boldsymbol{T}_j$ and the divergence of the Maxwell stress tensor $\nabla \cdot \boldsymbol{T}_{\mathrm{m}}$ is viewed as a constant vector $\boldsymbol{t}_j$. We can use $\boldsymbol{B}_j$ and $\boldsymbol{H}_j$ obtained in Algorithm 1 to calculate $\boldsymbol{T}_j$.

$$\boldsymbol{T}_j = \boldsymbol{B}_j \otimes \boldsymbol{H}_j - \frac{\mu_0}{2} H_j^2 \boldsymbol{I} \quad (31)$$

We use grid-based finite difference to calculate $\boldsymbol{t}_j$. To denote the neighbours of grid node j, we define two symbols $r_{\beta j}$ and $l_{\beta j}$, to represent as the indices of the right and left nodes of node j in the β-th axis respectively. The α-th component of $\boldsymbol{t}_j$ is given by

$$(\boldsymbol{t}_j)_\alpha = \sum_\beta \frac{(\boldsymbol{T}_{r_{\beta j}})_{\alpha\beta} - (\boldsymbol{T}_{l_{\beta j}})_{\alpha\beta}}{2\,\Delta x}, \quad (32)$$

where $\boldsymbol{T}_{r_{\beta j}}$ is the Maxwell stress tensor on the grid node $r_{\beta j}$, and $\boldsymbol{T}_{l_{\beta j}}$ is the one on the grid node $l_{\beta j}$. After getting $\boldsymbol{t}_j$, we make an approximation of our target:

$$\sum_j \int_{\Omega_j} N_i(\boldsymbol{x}) \nabla \cdot \boldsymbol{T}_{\mathrm{m}}(\boldsymbol{x}, t^n)\, \mathrm{d}\boldsymbol{x} \approx \sum_j \boldsymbol{t}_j \int_{\Omega_j} N_i(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \quad (33)$$

Third, we do cell-wise integration over the grid basis function $N_{\boldsymbol{i}}(\boldsymbol{x})$, with R_{ij} denoting the integration value of $\int_{\Omega_j} N_i(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$. We note that only if cell i adjoins cell j, is R_{ij} non-zero, since the value of $N_i(\boldsymbol{x})$ varnishes in Ω_j if cell i is not adjacent to cell j. Therefore, there are only finite cases for the spatial relation between node i and node j when R_{ij} is non-zero. Since the positions of grid nodes are constant, R_{ij} can be computed in advance. The calculation of R_{ij} is trivial, and we give the results in Appendix A.2.

Finally, the third term of the right-hand-side in (27) can be approximated by

$$\sum_j \boldsymbol{t}_j \int_{\Omega_j} N_i(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \sum_j R_{ij} \boldsymbol{t}_j. \quad (34)$$

4.3 The Full Method

We incorporate the magnetic field computation into the algorithm of MLS-MPM [Hu et al. 2018]. Here, the term 'particle' indicates a material point, which controls billions of microscopic particles. For the p-th particle, we add a new property n_p on it to indicate the number of magnetic micro-particles carried by it. With the new property assigned, we can start our magnetic field computation. First, we transfer n_p from particles to grid in the P2G step of MPM and calculate the corresponding n_i saved at the i-th grid node for each i. Then, we use Algorithm 1 to compute the magnetic field. After that, we calculate magnetic forces for all the grid nodes, and apply them in the later grid operation.

The full method is summarized in the following.

(1) **Particles to grid.** Use APIC [Jiang et al. 2015] or PolyPIC [Fu et al. 2017] to transfer momentum from particles to grid. Use the grid basis functions $N_{\boldsymbol{i}}(\boldsymbol{x})$ to transfer mass and microscopic particle number from particles to grid.
(2) **Calculate magnetic field.** Use the transferred number of microscopic particles and the volume of cells to calculate the number density of micro-particles saved at every grid node. Then execute the Algorithm 1 to compute the magnetic field.
(3) **Calculate magnetic force.** Use the magnetic field to calculate the Maxwell stress tensor at every grid node. Then use the grid-based finite difference to calculate the divergence of Maxwell stress tensor, namely the magnetic force at every grid node.
(4) **Update grid momentum.** Use the forward or backward Euler scheme to update grid momentum. In this work, we adopt an explicit integrator for implementation. An implicit one can reduce the number of time steps but requires more effort on solving linear systems.
(5) **Grid to particles.** Use APIC or PolyPIC to transfer velocities and affine coefficients from grid to particles.
(6) **Particle deformation gradient.** Use the MLS approximation [Hu et al. 2018] to update the particle deformation gradient.
(7) **Particle advection.** Update particle positions with their new velocites.

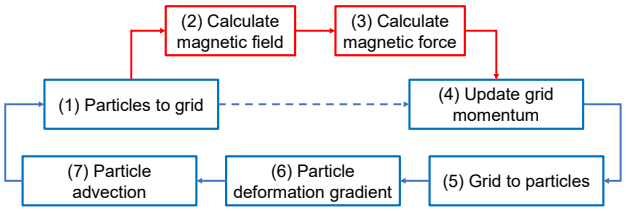

Fig. 10. The pipeline sketch of the temporal evolution of a magnetic substance. New components (painted red) are added on the basis of the standard material point method (painted blue).

5 SIMULATION RESULTS

We evaluate the efficacy of our method by a set of examples for magnetic phenomena simulation. The magnetic field is solved once per frame and several MPM steps are taken within each frame[1]. Parameter settings used in the simulations are summarized in Table 2. These experiments are performed by three threads on an Intel(R) Xeon(R) W-3175X processor, with the PCG solver implemented by CUDA and run on a NVIDIA Quadro RTX 8000 graphics card.

Convergence of the Newton based algorithm. First of all, we evaluate the convergence of Algorithm 1 for different settings. The objects all rest at the original positions, and the initial guess of the Newton–Raphson method is set as $\mathbf{0}$. From Figure 12, we can see that the numerical algorithm for magnetic field computation converges rapidly after several iterations, and the relative residual, i.e., $|\boldsymbol{F}(\hat{\boldsymbol{\varphi}})|/|\boldsymbol{F}(\mathbf{0})|$, almost approaches machine precision after 6 iterations for all the settings.

Surfing. As shown in Figure 2, we simulate a lump of magnetic viscous fluid, which bears a rigid surfer. The viscosity of the fluid is much higher than ferrofluid, so it behaves more like a lump of clay. The fluid is nonlinearly magnetized and attracted by a magnet. During the process, the surfer is carried forward by the fluid and behaves like surfing. We highlights the automatic handling of fluid-rigid body coupling in the magnetization induced motion. This is a merit of using the MPM to simulate magnetized materials.

Buckyballs. As shown in Figure 3, we simulate two lumps of magnetic viscous fluid. They are nonlinearly magnetized and attracted by 20 magnetic buckyballs. The process consists of two stages. In the former, the fluid at the bottom crawls on the buckyballs. In the latter, the fluid at the top falls down, then collides with the buckyballs and the fluid at the bottom. Through this example, we demonstrates that our method can handle complex geometries, topological changes and collisions.

Spike. As shown in Figure 4, we simulate a lump of magnetic viscous fluid in a container. The fluid generates a spike in the attraction of a magnet whose motion is scripted. The spike gradually grows up and finally contacts the magnet. After that the magnet begins to rotate and rise. The spike follows the magnet in the process.

[1]This is reasonable because the displacement distance of a MPM particle within a frame is far less than the grid spacing Δx in our simulation settings.

Two dragons. To test our scheme in coupling between multiple objects and a magnet, we set a scene of two magnetized dragons playing with a bead (a big one), as shown in Figure 5. The two dragons are elastic solid, and the bead is a spherical permanent magnet. The two dragons are attracted to the bead and collide with each other. The collision is automatically handled by the material point method.

Magnetic cube. We conduct a comparison study to demonstrate the stability of nonlinear magnetization during the interaction between a magnet and a magnetized object. As shown in Figure 7, We simulate two elastic cubes. One is linearly magnetized, and the other is nonlinearly magnetized. As a variable controlled in this experiment, the linear magnetization function is set to be equivalent to the first order Taylor's expansion term of the nonlinear Langevin function. We use a strong magnet to attract the two cubes. Because of our settings, the two cubes perform dynamically and similarly when they are far from the magnet. However, when they contact the magnet, the linearly magnetized cube explodes while the nonlinearly magnetized cube keeps stable. The comparison experiment highlights the importance of nonlinear magnetization, as it relieves a conflict in simulating the interaction between a magnet and a magnetized object. Generally, we want a magnetized object to perform some dynamic response to the magnet when it is far from the magnet, but still keeps stable when it contacts the magnet. This is challenging for a linearly magnetized object. The reason lies in the uneven spatial distribution of the magnetic field excited by the magnet. The intensity of the magnetic field damps rapidly as the distance from the magnet increases. If we use a weak magnet, a linearly magnetized object may not perform any observable response when it is far from the magnet. If we use a strong magnet, the linearly magnetized object can not contact the magnet stably. By contrast, the nonlinearly magnetized cube can tolerate a strong magnet thanks to its bounded magnetization intensity.

Parameter study. We simulate four pairs of magnetic viscous fluids and magnets. The scene setting is identical for each pair: putting a spherical magnet on a lump of magnetic viscous fluid. The viscosity of fluids and the magnetization intensity of magnets vary from one pair to another. We use the four pairs to demonstrate the role of fluid viscosity and the role of magnet's intensity of magnetization in the interaction between magnetic viscous fluids and magnets. When the magnet's intensity of magnetization is high, the magnetic force is greater than gravity, thus the fluid will crawl on the magnet. When the fluid viscosity is high, the fluid tends to gather. We demonstrate the different equilibrium states in Figure 8.

Attraction comparison. Both a magnetic force and universal gravitation can lead to attraction between objects. The intensity of universal gravitation is so small that it is negligible if celestial bodies are not involved. In computer graphics, the intensity of universal gravitation is often proportionally magnified to generate observable attraction between non-celestial objects. We refer to the magnified force as an artificial force here. We make a comparison between attractions generated by magnetic forces and artificial forces. We conduct two experiments. First, a lump of magnetic viscous fluid is attracted by a spherical magnet. Second, the existence of an artificial

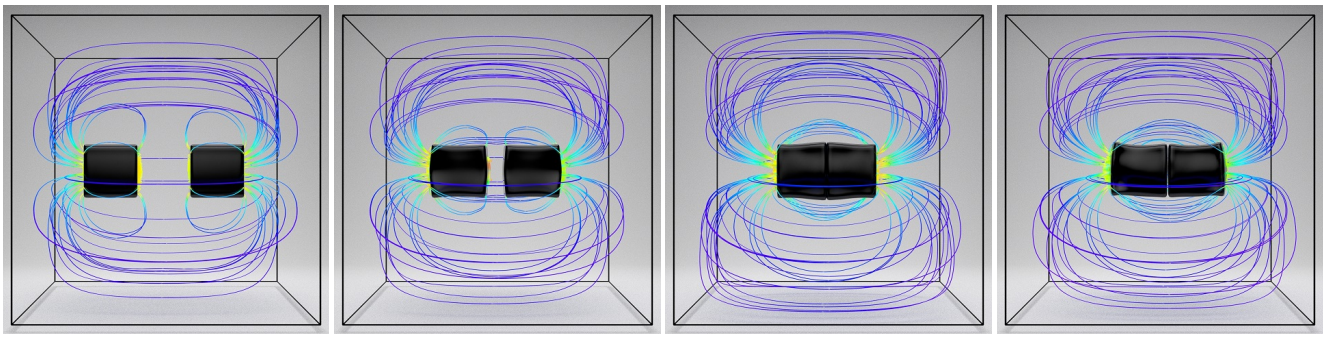

Fig. 11. Two magnetic cubes are nonlinearly magnetized by a horizontal uniform magnetic field and then attracted by each other because of the opposite polarities of confronting faces. The induction lines of the (induced) internal magnetic field are drawn.

Table 2. Simulation parameters for the examples.

Figure	Scene Description†	External Field	M of the Magnet	# of Particles	# of Particles Per Cell	Total Time‡	Alg. Time‡
2	Surfing	Magnet	2.0×10^6 A/m	2.8×10^5	32	88.3 s	21.4 *s*
3	Buckyballs	Magnet	3.0×10^6 A/m	1.1×10^6	32	239.3 *s*	25.9 *s*
4	Spike	Magnet	1.0×10^6 A/m	4.2×10^5	32	123.0 *s*	18.5 *s*
5	Two dragons	Magnet	4.0×10^5 A/m	4.0×10^5	8	118.2 *s*	16.8 *s*
7	Cube (linear)	Magnet	2.0×10^6 A/m	1.1×10^5	8	45.3 *s*	16.3 *s*
7	Cube (nonlinear)	Magnet	2.0×10^6 A/m	1.1×10^5	8	48.6 *s*	19.6 *s*
8	Parameter study (1)	Magnet	1.0×10^5 A/m	5.3×10^4	8	36.2 *s*	14.6 *s*
8	Parameter study (2)	Magnet	1.0×10^5 A/m	5.3×10^4	8	36.6 *s*	14.6 *s*
8	Parameter study (3)	Magnet	4.0×10^5 A/m	5.3×10^4	8	37.4 *s*	15.7 *s*
8	Parameter study (4)	Magnet	4.0×10^5 A/m	5.3×10^4	8	37.9 *s*	15.7 *s*
9	Attraction (magnetic)	Magnet	1.5×10^6 A/m	7.1×10^4	8	41.7 *s*	19.8 *s*
9	Attraction (artificial)	\	\	7.1×10^4	8	23.3 *s*	\
11	Induced attraction	Uniform	\	2.5×10^5	8	74.1 *s*	10.5 *s*
13	Ferrofluid (2D)	Uniform	\	1.6×10^4	16	2.2 *s*	0.1 *s*
14	Ferrofluid (3D)	Uniform	\	4.7×10^5	8	137.4 *s*	18.7 *s*

† All these simulations use realistic physical values, including $\mu_0 = 4\pi \times 10^{-7}\,\mathrm{N/A^2}$, $k_\mathrm{B} = 1.380649 \times 10^{-23}\,\mathrm{J/K}$ and $T = 300\,\mathrm{K}$, $\rho = 2500\,\mathrm{kg/m^3}$. The grid spacing Δx is 7.8×10^{-3} m. The grid resolution is $128 \times 128 \times 128$ for 3D cases, and it is 128×128 for the 2D case. The time step Δt is 5×10^{-5}s, and there are 40 steps per frame. The magnitude of a micro-particle's magnetic moment $\boldsymbol{m}$ is $1.09 \times 10^{-19}\mathrm{A \cdot m^2}$. Initially there are 3.60×10^{17} micro-particles in a cell filled with magnetized materials, which is evenly distributed to the MPM particles within it.

‡ These values are averaged with respect to a single frame, with the algorithm time measured for the Newton–Raphson method (Algorithm 1).

force between the viscous fluid and the sphere results in the attraction. We adjust the related parameters so that the magnetic force and the artificial force are comparable in our numerical experiments. We demonstrate the difference in Figure 9.

Induced attraction. As shown in Figure 11, we put two elastic cubes in a horizontal uniform external magnetic field. After being magnetized, the cubes' confronting faces have opposite polarities, so the cubes will attract each other. We demonstrate the fact and visualize the $\boldsymbol{H}_\mathrm{int}$ within the entire space.

Ferrofluid. We test the response of ferrofluid immersed in a uniform vertical magnetic field. Under the action of gravity, magnetic and fluid viscosity forces, our simulator can generate spiky features on a fluid surface in both 2D and 3D situations, as shown in Figure 13 and 14. As can be also gleaned from the supplementary video, due to the lack of capillary forces in our current implementation, such cone structures were not as stable as in the work of Ni et al. [2020] .

6 CONCLUSIONS AND DISCUSSIONS

We proposed a novel numerical method to simulate interactions between magnetic fields and *nonlinearly* magnetized objects. Our main contribution lies in an MPM formulation of the nonlinear Langevin model. The underlying Langevin's nonlinear theory of paramagnetism bridges microscopic structures and macroscopic equations through a statistical derivation. The material point method takes its responsibility to handle the multi-scale magnetic-mechanical coupling, in which the nonlinear equation of magnetization is efficiently solved with the Newton–Raphson method. Compared with previous work from Ni et al. [2020] and Liu et al. [2011], our nonlinear magnetization framework could precisely capture the magnetic saturation effect in the real world without tuning parameters, which

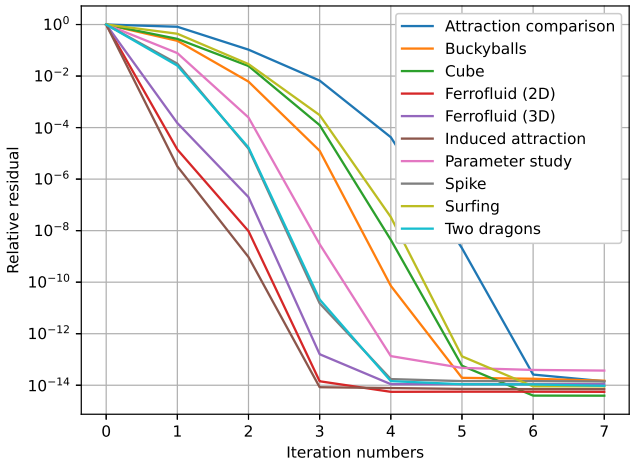

Fig. 12. Relative residuals of the Newton–Raphson method as the iteration number increases for different settings listed in Table 2, with the nonlinear model of magnetization taken.

will enable efficient and stable simulations of various magnetic phenomena in a unified way. Moreover, our particle-based domain discretization provides a more natural way for handling contact and carrying microscopic quantities than the level-set-based methods. As an indispensable complement of the numerical schemes proposed in Zhu et al. [2011] and Liu et al. [2011], we provide a complete numerical recipe on both the discretization formula and the nonlinear model extension. Due to its conciseness, our method can be easily integrated into any conventional MLS-MPM framework or installed as plugins of VFX commercial softwares which support MPM-based simulations to create stable and novel magnetic phenomena simulations in both engineering and entertainment.

However, due to the lack of surface tension in the current framework, our simulator can not stably manifest a rich set of the characteristic spike phenomenon as the ferrofluid is immersed in a magnetic field. Performance can be regarded as another limitation of our method. Through comparison with the results of Hu et al. [2018], we found most of the computation loads (shown in the last two columns of Table 2) were inherited from the original MPM solver rather than from our magnetic force evaluation. Our current implementation is a CPU-GPU hybrid version, where the MPM pipeline is fully performed on the CPU. Immediate next work is to move all these procedures onto GPU with the help of the sparse paged grid (SPGrid) [Setaluri et al. 2014], to gain improved performance, as proposed by Gao et al. [2018]. Besides, as in the conventional pipeline of Eulerian fluid simulation, replacing the collocated grid with a staggered grid, such as a marker-and-cell (MAC) grid [Harlow and Welch 1965], can promote the accuracy of finite difference and produce more numerically stable simulations. Thus, how to discretize the nonlinear equation of magnetization upon a staggered grid is left as a further study.

Our simulations output positions of particles. For mesh-based rendering, we need to reconstruct a surface that wraps around the particles. We use the surface reconstruction function in Houdini. Flickering occurs where the density of particles is relatively low. Future work may alleviate the problem by developing some more advanced surface reconstruction skills.

Another interesting direction for future work is to incorporate surface tension into the framework, which features spikes on the liquid surface even if inviscid. There are some work devoted to simulating surface tension effects within an MPM pipeline [Hyde et al. 2020], but it may be subtle to consider when magnetic forces present, in that producing ferrofluid spikes requires high-quality surface tracking techniques. Furthermore, the current scheme solves the magnetic field as an optimization problem upon the background Cartesian grid. Nevertheless, electromagnetic systems are also widely computed by the fast multipole method (FMM) [Beatson and Greengard 1997]. This method does not need any artificial boundary conditions and can easily resolve problems with open boundaries. Whether the FMM solver accelerates our numerical scheme and how to combine it with the MPM framework are the questions that warrant further research.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This work was supported in part by National Key R&D Program of China (2019YFF0302902). Bo Zhu acknowledges the funding supports from Neukom Institute CompX Faculty Grant, Burke Research Initiation Award, and Toyota TEMA North America Inc. We would like to thank Liangwang Ruan and Xingqiao Li from Peking University for their help on editing the supplementary video. The Houdini Education license is credited for the video generations.

REFERENCES

Rick Beatson and Leslie Greengard. 1997. *A short course on fast multipole methods.* Oxford University Press, Oxfordshire, England, UK, 1–37.

J. U. Brackbill and H. M. Ruppel. 1986. FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions. *J. Comput. Phys.* 65, 2 (Aug. 1986), 314–343.

J. M. D. Coey. 2010. *Magnetism and Magnetic Materials.* Cambridge University Press, Cambridgeshire, England, UK.

Gilles Daviet and Florence Bertails-Descoubes. 2016. A Semi-Implicit Material Point Method for the Continuum Simulation of Granular Materials. *ACM Trans. Graph.* 35, 4, Article 102 (July 2016), 13 pages.

Albert Einstein and Jakob Laub. 1908. Über die im elektromagnetischen Felde auf ruhende Körper ausgeübten ponderomotorischen Kräfte. *Annalen der Physik* 331, 8 (1908), 541–550.

Richard P. Feynman, Robert B. Leighton, and Matthew Sands. 2011. *The Feynman Lectures on Physics* (the new millennium ed.). Vol. 2. Basic Books, New York, NY, USA.

Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. 2017. A Polynomial Particle-in-Cell Method. *ACM Trans. Graph.* 36, 6, Article 222 (Nov. 2017), 12 pages.

Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. 2018. Animating Fluid Sediment Mixture in Particle-Laden Flows. *ACM Trans. Graph.* 37, 4, Article 149 (July 2018), 11 pages.

Ali Ghaffari, Seyed Hassan Hashemabadi, and Mansour Bazmi. 2015. CFD simulation of equilibrium shape and coalescence of ferrofluid droplets subjected to uniform magnetic field. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 481 (2015), 186–198.

Francis H. Harlow and J. Eddie Welch. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids* 8, 12 (1965), 2182–2189.

Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A Moving Least Squares Material Point Method with Displacement Discontinuity and Two-Way Rigid Body Coupling. *ACM Trans. Graph.* 37, 4, Article 150 (July 2018), 14 pages.

Libo Huang, Torsten Hädrich, and Dominik L. Michels. 2019. On the Accurate Large-scale Simulation of Ferrofluids. *ACM Trans. Graph.* 38, 4, Article 93 (July 2019), 15 pages.

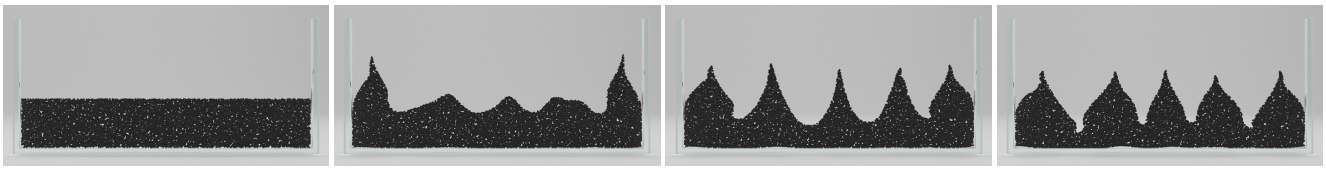

Fig. 13. Two-dimensional ferrofluid is magnetized by a uniform external field. The fluid surface is shaped into spikes.

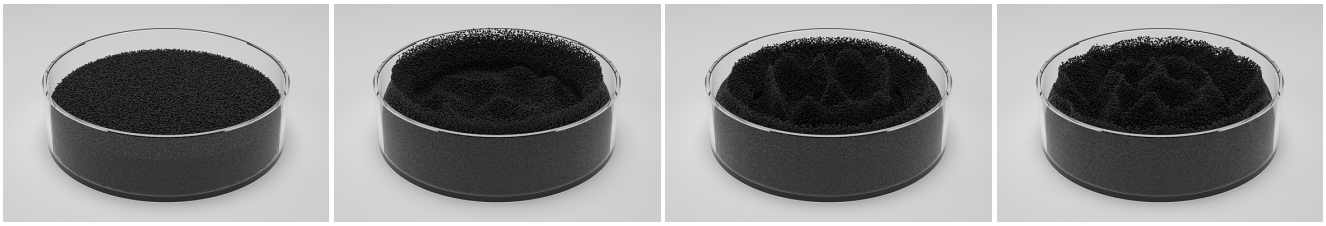

Fig. 14. Three-dimensional ferrofluid is magnetized by a uniform external field. The spikes cannot be stable, and the shape of them is also unclear.

Libo Huang and Dominik L. Michels. 2020. Surface-Only Ferrofluids. *ACM Trans. Graph.* 39, 6, Article 174 (Nov. 2020), 17 pages.

David A. B. Hyde, Steven W. Gagniere, Alan Marquez-Razon, and Joseph Teran. 2020. An Implicit Updated Lagrangian Formulation for Liquids with Large Surface Energy. *ACM Trans. Graph.* 39, 6, Article 183 (Nov. 2020), 13 pages.

Tomokazu Ishikawa, Yonghao Yue, Kei Iwasaki, Yoshinori Dobashi, and Tomoyuki Nishita. 2013. Visual Simulation of Magnetic Fluid Using a Procedural Approach for Spikes Shape. In *Computer Vision, Imaging and Computer Graphics. Theory and Application*. Springer Berlin Heidelberg, Berlin, Heidelberg, 112–126.

Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact. *ACM Trans. Graph.* 36, 4, Article 152 (July 2017), 14 pages.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The Affine Particle-in-Cell Method. *ACM Trans. Graph.* 34, 4, Article 51 (July 2015), 10 pages.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The Material Point Method for Simulating Continuum Materials. In *ACM SIGGRAPH 2016 Courses* (Anaheim, California) *(SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, Article 24, 52 pages.

Seung-wook Kim and JungHyun Han. 2020. Simulation of Arbitrarily-shaped Magnetic Objects. *Computer Graphics Forum* 39, 7 (2020), 119–130.

Seung-Wook Kim, Sun Young Park, and Junghyun Han. 2018. Magnetization Dynamics for Magnetic Object Interactions. *ACM Trans. Graph.* 37, 4, Article 121 (July 2018), 13 pages.

Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-Prager Elastoplasticity for Sand Animation. *ACM Trans. Graph.* 35, 4, Article 103 (July 2016), 12 pages.

Jing Liu, Yit Fatt Yap, and Nam-Trung Nguyen. 2011. Numerical study of the formation process of ferrofluid droplets. *Physics of Fluids* 23, 7 (2011), 072008.

Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2020. A Level-Set Method for Magnetic Substance Simulation. *ACM Trans. Graph.* 39, 4, Article 29 (July 2020), 15 pages.

Curtis M. Oldenburg, Sharon E. Borglin, and George J. Moridis. 2000. Numerical simulation of ferrofluid flow for subsurface environmental engineering applications. *Transport in Porous Media* 38, 3 (2000), 319–344.

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* (Los Angeles, California) *(SCA '15)*. Association for Computing Machinery, New York, NY, USA, 157–163.

Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. 2008. Two-Way Coupling of Fluids to Rigid and Deformable Solids and Shells. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–9.

Ronald E. Rosensweig. 1985. *Ferrohydrodynamics*. Cambridge University Press, Cambridgeshire, England, UK.

Rajsekhar Setaluri, Mridul Aanjaneya, Sean Bauer, and Eftychios Sifakis. 2014. SPGrid: A Sparse Paged Grid Structure Applied to Adaptive Smoke Simulation. *ACM Trans. Graph.* 33, 6, Article 205 (Nov. 2014), 12 pages.

Dongxiao Shi, Qincheng Bi, and Rongqi Zhou. 2014. Numerical simulation of a falling ferrofluid droplet in a uniform magnetic field by the VOSET method. *Numerical Heat Transfer, Part A: Applications* 66, 2 (2014), 144–164.

Eftychios Sifakis and Jernej Barbic. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California) *(SIGGRAPH '12)*. Association for Computing Machinery, New York, NY, USA, Article 20, 50 pages.

Michael Steffen, Robert M. Kirby, and Martin Berzins. 2008. Analysis and reduction of quadrature errors in the material point method (MPM). *Internat. J. Numer. Methods Engrg.* 76, 6 (2008), 922–948.

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A Material Point Method for Snow Simulation. *ACM Trans. Graph.* 32, 4, Article 102 (July 2013), 10 pages.

Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for Phase-Change and Varied Materials. *ACM Trans. Graph.* 33, 4, Article 138 (July 2014), 11 pages.

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

Andre Pradhana Tampubolon, Theodore Gast, Gergely Klár, Chuyuan Fu, Joseph Teran, Chenfanfu Jiang, and Ken Museth. 2017. Multi-Species Simulation of Porous Sand and Water Mixtures. *ACM Trans. Graph.* 36, 4, Article 105 (July 2017), 11 pages.

Bernhard Thomaszewski, Andreas Gumann, Simon Pabst, and Wolfgang Straßer. 2008. Magnets in Motion. *ACM Trans. Graph.* 27, 5, Article 162 (Dec. 2008), 9 pages.

Joel Wretborn, Rickard Armiento, and Ken Museth. 2017. Animation of Crack Propagation by Means of an Extended Multi-Body Solver for the Material Point Method. *Comput. Graph.* 69, C (Dec. 2017), 131–139.

Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2015. Continuum Foam: A Material Point Method for Shear-Dependent Flows. *ACM Trans. Graph.* 34, 5, Article 160 (Nov. 2015), 20 pages.

Gui-Ping Zhu, Nam-Trung Nguyen, R. V. Ramanujan, and Xiao-Yang Huang. 2011. Nonlinear Deformation of a Ferrofluid Droplet in a Uniform Magnetic Field. *Langmuir* 27, 24 (2011), 14834–14841.

Yongning Zhu and Robert Bridson. 2005. Animating Sand as a Fluid. *ACM Trans. Graph.* 24, 3 (July 2005), 965–972.

A PHYSICAL ANALYSIS

A.1 Proof of Positive Definiteness

We will prove the derivative matrix $\boldsymbol{F}'(\hat{\boldsymbol{\varphi}})$ to be symmetric positive definite. First, we rewrite (24) as follows:

$$\boldsymbol{F}'(\hat{\boldsymbol{\varphi}}) = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + \boldsymbol{G}^{\mathrm{T}}\frac{\partial \hat{\boldsymbol{M}}}{\partial \hat{\boldsymbol{H}}}\boldsymbol{G}. \tag{35}$$

After setting the Dirichlet boundary condition, $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$ is already symmetric positive definite. Therefore, semi-positive definiteness of $\partial \hat{\boldsymbol{M}}/\partial \hat{\boldsymbol{H}}$ leads to positive definiteness of $\boldsymbol{F}'(\hat{\boldsymbol{\varphi}})$. Noting that $\partial \hat{\boldsymbol{M}}/\partial \hat{\boldsymbol{H}}$

is a block-diagonal matrix, we only need to prove $\partial \boldsymbol{M}_i / \partial \boldsymbol{H}_i$ is semi-positive definite for every i.

Without loss of generality, we take the two-dimensional case as an example. Given that $\boldsymbol{M}_i = (M_{i1}, M_{i2})$, $\boldsymbol{H}_i = (H_{i1}, H_{i2})$, we obtain

$$\begin{aligned}\frac{\partial \boldsymbol{M}_i}{\partial \boldsymbol{H}_i} = &\frac{\mu_0 n_i m^2}{k_\mathrm{B} T H_i^2} L'\left(\frac{\mu_0 m H_i}{k_\mathrm{B} T}\right)\begin{pmatrix} H_{i1}^2 & H_{i1}H_{i2} \\ H_{i1}H_{i2} & H_{i2}^2 \end{pmatrix} \\ &+\frac{n_i m}{H_i^3} L\left(\frac{\mu_0 m H_i}{k_\mathrm{B} T}\right)\begin{pmatrix} H_i^2 - H_{i1}^2 & -H_{i1}H_{i2} \\ -H_{i1}H_{i2} & H_i^2 - H_{i2}^2 \end{pmatrix}.\end{aligned} \tag{36}$$

Since $L'(\mu_0 m H_i / k_\mathrm{B} T)$ and $L(\mu_0 m H_i / k_\mathrm{B} T)$ are both positive, we only need to prove the two matrices in (36) to be semi-positive definite. We take the quadratic form of the two symmetric matrices with respect to an arbitrary vector $\boldsymbol{x} = (x_1, x_2)^\mathrm{T}$. It is easy to validate

$$\boldsymbol{x}^\mathrm{T}\begin{pmatrix} H_{i1}^2 & H_{i1}H_{i2} \\ H_{i1}H_{i2} & H_{i2}^2 \end{pmatrix}\boldsymbol{x} = (x_1 H_{i1} + x_2 H_{i2})^2 \geq 0 \tag{37}$$

and

$$\boldsymbol{x}^\mathrm{T}\begin{pmatrix} H_i^2 - H_{i1}^2 & -H_{i1}H_{i2} \\ -H_{i1}H_{i2} & H_i^2 - H_{i2}^2 \end{pmatrix}\boldsymbol{x} = (x_1 H_{i2} - x_2 H_{i1})^2 \geq 0. \tag{38}$$

This completes the proof.

A.2 Calculation of Cell-Wise Integration

The task is to calculate the following integration:

$$R_{ij} = \int_{\Omega_j} N_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{39}$$

Similarly, we take a two-dimensional case as an example. Provided that $\boldsymbol{x} = (x_1, x_2)$ is the evaluation position, $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ is the position of node i, and $\boldsymbol{x}_j = (x_{j1}, x_{j2})$ is the position of node j, we can acquire

$$\Omega_j = \left[x_{j1} - \frac{\Delta x}{2}, x_{j1} + \frac{\Delta x}{2}\right] \times \left[x_{j2} - \frac{\Delta x}{2}, x_{j2} + \frac{\Delta x}{2}\right] \tag{40}$$

and

$$\begin{aligned} R_{ij} &= \int_{\Omega_j} N\left(\frac{x_1 - x_{i1}}{\Delta x}\right) N\left(\frac{x_2 - x_{i2}}{\Delta x}\right)\mathrm{d}\boldsymbol{x} \\ &= \int_{x_{j1}-\frac{\Delta x}{2}}^{x_{j1}+\frac{\Delta x}{2}} N\left(\frac{x_1 - x_{i1}}{\Delta x}\right)\mathrm{d}x_1 \cdot \int_{x_{j2}-\frac{\Delta x}{2}}^{x_{j2}+\frac{\Delta x}{2}} N\left(\frac{x_2 - x_{i2}}{\Delta x}\right)\mathrm{d}x_2. \end{aligned} \tag{41}$$

The integration for the first axis can be calculated by

$$\int_{x_{j1}-\frac{\Delta x}{2}}^{x_{j1}+\frac{\Delta x}{2}} N\left(\frac{x_1 - x_{i1}}{\Delta x}\right)\mathrm{d}x_1 = \begin{cases} \frac{2}{3}\Delta x, & |x_{j1} - x_{i1}| = 0, \\ \frac{1}{6}\Delta x, & |x_{j1} - x_{i1}| = 1, \\ 0, & |x_{j1} - x_{i1}| > 1. \end{cases} \tag{42}$$

So do the other axes, and R_{ij} can be obtained by taking the product of the integration for each axis.