An Induce-on-Boundary Magnetostatic Solver for Grid-Based Ferrofluids

XINGYU NI*, School of CS & State Key Laboratory of General Artificial Intelligence, Peking University, China
RUICHENG WANG*, Yuanpei College, Peking University, China
BIN WANG†, State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI), China
BAOQUAN CHEN†, School of IST & State Key Laboratory of General Artificial Intelligence, Peking University, China

Fig. 1. Ferrofluid simulation using the Induce-on-Boundary (IoB) magnetostatic solver. A ferrofluid placed between two magnets is magnetized by the lower magnet initially, forming ground spikes. Then, a second magnet on top elevates a portion of the fluid, creating ceiling spikes. Afterward, the lower magnet is removed, causing the fluid to be attracted to the upper magnet. Finally, removing the top magnet allows the ferrofluid to fall back.

This paper introduces a novel Induce-on-Boundary (IoB) solver designed to address the magnetostatic governing equations of ferrofluids. The IoB solver is based on a single-layer potential and utilizes only the surface point cloud of the object, offering a lightweight, fast, and accurate solution for calculating magnetic fields. Compared to existing methods, it eliminates the need for complex linear system solvers and maintains minimal computational complexities. Moreover, it can be seamlessly integrated into conventional fluid simulators without compromising boundary conditions. Through extensive theoretical analysis and experiments, we validate both the convergence and scalability of the IoB solver, achieving state-of-the-art performance. Additionally, a straightforward coupling approach is proposed and executed to showcase the solver's effectiveness when integrated into a grid-based fluid simulation pipeline, allowing for realistic simulations of representative ferrofluid instabilities.

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

Additional Key Words and Phrases: Ferrofluid Simulation, Magnetostatics, Boundary Integral Equations, Single-Layer Potential, Free-Surface Flows

*joint first authors
†corresponding authors

Authors' addresses: Xingyu Ni, nixy@pku.edu.cn, School of CS & State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China; Ruicheng Wang, wrc0326@outlook.com, Yuanpei College, Peking University, Beijing, China; Bin Wang, binwangbuaa@gmail.com, State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI), Beijing, China; Baoquan Chen, baoquan@pku.edu.cn, School of IST & State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China.

0730-0301/2024/7-ART56 $15.00
https://doi.org/10.1145/3658124

ACM Reference Format:
Xingyu Ni, Ruicheng Wang, Bin Wang, and Baoquan Chen. 2024. An Induce-on-Boundary Magnetostatic Solver for Grid-Based Ferrofluids. *ACM Trans. Graph.* 43, 4, Article 56 (July 2024), 14 pages. https://doi.org/10.1145/3658124

1 INTRODUCTION

Ferrofluids, colloidal solutions of nanoscale magnetic particles, have attracted extensive attention in recent years due to their unique physical mechanisms and practical applications in materials [Liu et al. 2019; Zhang et al. 2019], robotics [Fan et al. 2020, 2022], and artistic creation [Kodama 2008; Yetisen et al. 2016]. Since the seminal work of Huang et al. [2019], both magnetostatics and hydrodynamics that govern ferrofluids have been effectively harnessed within the realm of computer graphics, as evidenced by a series of simulation frameworks. These frameworks effectively capture the dynamics of ferrofluids with the complex interplay of three distinct forces—gravity, magnetic force, and fluid surface tension—and successfully reproduce the formation of characteristic spike structures, a phenomenon known as *normal-field instability* [Rosensweig 1997].

However, the existing methodologies exhibit two primary shortcomings (see Table 1): first, the necessity of implementing sophisticated linear system solvers (e.g., GMGPCG [Ni et al. 2020] and preconditioned GMRES [Huang and Michels 2020]); second, the constrained scalability due to non-optimal modeling and discretization (e.g., volumetric force models [Huang et al. 2019; Shao et al. 2023], large number of variables [Ni et al. 2020], and unavoidable dense matrices [Huang and Michels 2020]). These two obstacles substantially hinder the seamless integration of magnetostatics as a plug-and-play module into the conventional pipeline of fluid simulation. Furthermore, certain approaches also face issues related to physical accuracy, such as the smearing of liquid-air interfaces [Huang et al. 2019; Ni et al. 2020; Shao et al. 2023], the failure to meet boundary conditions at infinity [Ni et al. 2020], and the neglect of fluid vorticity [Huang and Michels 2020].

Table 1. Comparisons between the IoB and existing magnetostatic solvers. The fluid simulation part of these frameworks is not compared here. In each column, the cells with properties that are preferred due to correctness or simplicity are colored light green.

Approach	Force type†	Physical accuracy		Linear system		Computational complexity‡	
		Sharp interface	BC at infinity	Sparsity	Method	Time	Space
SPH [2019; 2023]	Volumetric	No	Yes	Dense	CG squared	$O(KN\sqrt{N})$	$O(N\sqrt{N})$
Grid-based [2020]	Interfacial	No	No	Sparse	GMGPCG	$O(KN\sqrt{N})$	$O(N\sqrt{N})$
Surface-only [2020]	Interfacial	Yes	Yes	Dense	Preconditoned GMRES	$O(KN^2)$	$O(N^2)$
IoB (ours)	Interfacial	Yes	Yes	Dense	Fixed-point iterations	$O(KN)$	$O(N)$

† Magnetic interactions are ubiquitous. The terms "volume" and "interface" are used solely to designate the solution domain of the numerical method.
‡ K denotes the number of iterations, and N denotes the number of discrete points at the fluid–air interface. For SPH methods, $N\sqrt{N} \approx N_\mathrm{p}$, where N_p denotes the number of SPH particles; for the grid-based method, $N\sqrt{N} \approx n^3$, where n is the grid resolution.

We present a lightweight, fast, and accurate magnetostatic solver that iteratively computes the magnetic field upon fluid surfaces, inspired by recently proposed algorithms based on boundary integral equations (BIEs) [Miller et al. 2023; Sawhney et al. 2023; Sugimoto et al. 2023]. When subjected to an external magnetic field, our iterative computation scheme can be conceptually understood as a series of successive inductions of the internal field within infinitesimally small time intervals. Consequently, we term this solution the Induce-on-Boundary (IoB) approach, whose convergence has been theoretically proved in the paper. Our IoB solver is implemented in a matrix-free manner, relying solely on straightforward equations of the single-layer potential and without the need for preconditioning. In comparison to existing methods, the IoB solver, enhanced by a Fast Multipole Method (FMM), boasts the lowest computational complexity, making it exceptionally well-suited for handling large-scale simulation scenarios. Furthermore, thanks to the inherent advantages derived from our chosen BIEs, there is no requirement for non-physical approximations. This IoB magnetostatic solver is capable of efficiently and accurately determining the pressure jump at the liquid-air interface due to magnetic interactions, which is demonstrated to naturally couple with grid-based fluid simulation [Bridson 2015] via extensive experiments.

Contributions. Our technical contributions are

- The theory of Induce-on-Boundary discretization,
- Analysis on convergence of the numerical scheme,
- Implementation of the IoB solver for grid-based fluids,
- Scalability and accuracy validations through experiments.

2 RELATED WORK

Free-surface fluid simulation. Since the seminal work of Foster and Fedkiw [2001], the simulation of fluid phenomena with free surfaces has emerged as a popular research area in computer graphics. A broad spectrum of interfacial effects, such as foams and bubbles [Deng et al. 2022; Ishida et al. 2017; Ram et al. 2015; Wang et al. 2021; Wretborn et al. 2022], chemical reaction [Kang et al. 2007; Ren et al. 2014], viscous coiling [Larionov et al. 2017; Panuelos et al. 2023], and waves [Ando and Batty 2020; Huang et al. 2021; Jeschke and Wojtan 2023], etc., have been successfully simulated through the invention of many high-performance numerical solvers. According to the surface tracking methods employed, these solvers fall into two primary categories: grid-based and particle-based approaches. The former [Chentanez and Müller 2014; Goldade et al. 2016; Li et al. 2023; Narita and Ando 2022] typically rely on the level-set method [Osher and Fedkiw 2005] that implicitly captures smooth and consistent surfaces, while the latter, e.g., PBD [Macklin and Müller 2013; Xing et al. 2022], SPH [Koschier et al. 2022], PIC/FLIP [Chen et al. 2020; Ferstl et al. 2016; Kugelstadt et al. 2021; Wang et al. 2020; Zhu and Bridson 2005], and MPM [Chen et al. 2021; Hyde et al. 2020; Tampubolon et al. 2017], excel at preserving volume through surface reconstruction from particles. To deal with surface-tension-dominated fluids (e.g., ferrofluids), grid-based approaches are often easier choices because accurate, smooth curvatures are provided with little effort. Additionally, mesh-based approaches [Bojsen-Hansen and Wojtan 2013; Da et al. 2014, 2015, 2016; Wojtan et al. 2011; Zhu et al. 2015b, 2014] are critical alternatives due to their strengths in both surface tracking and volume preservation, but the relatively high computational complexity is the main weakness.

Magnetic simulation. In the computer graphics community, the simulation of magnetic effects is pioneered by Thomaszewski et al. [2008] through rigid body animations. After that, Kim et al. [2020; 2018] proposed magnetization dynamics inspired by micromagnetics to produce more realistic results, which further enables large-scale simulation of inducible magnets with fast stabilization techniques [Kim and Han 2022]. For non-rigid materials, Sun et al. [2021] extended MPM to simulate nonlinearly magnetized viscoelastic bodies, and Chen et al. [2022] successfully explored the simulation and optimization of hard-magnetic thin shells. All these frameworks incorporate volumetric magnetic interactions in their modeling. In term of ferrofluids, based upon procedural modeling of the normal-field instability [Ishikawa et al. 2013], Huang et al. [2019] made significant advancements by introducing an SPH-based solution for the first-principles simulation. This weakly compressible scheme was subsequently improved by the divergence-free SPH [Shao et al. 2023]. Both of the frameworks rely on volumetric force models in magnetostatics. In contrast, Ni et al. [2020] proposed the utilization of interfacial magnetic forces as an equivalent replacement of volumetric ones, whose conciseness and effectiveness are validated through grid-based ferrofluid simulations. The interfacial force model was also implemented in the work of Huang and Michels [2020], which is integrated into the surface-only liquid solver [Da et al. 2016] with mesh-based surface tracking.

BIE-based algorithms. Algorithms based on boundary integral equations have demonstrated their effectiveness in a variety of simulation applications, such as deformable bodies [James and Pai 1999; Sugimoto et al. 2022], brittle fractures [Hahn and Wojtan 2015, 2016;

Fig. 2. When exposed to a magnet approximated as a dipole positioned beneath it, a blob of ferrofluid steadily shape itself into radial spikes aligned with the orientation of the magnetic field.

Zhu et al. 2015a], liquids [Da et al. 2016; Huang and Michels 2020], and ocean waves [Huang et al. 2021; Keeler and Bridson 2015]. These algorithms generally convert the governing equations into BIEs and subsequently discretize object surfaces into boundary elements for numerical solution. Another interesting application is in acoustics [James et al. 2006; Umetani et al. 2016], where equivalent source methods resulting in least-squares BIE solves have succeeded in sound synthesis for both liquids and rigid-body fractures [Zheng and James 2009, 2010]. Apart from these, BIE-based algorithms have also been extensively studied for solving the rendering equation [Kajiya 1986], where stochastic methods, e.g., Monte–Carlo ray tracing [Pharr et al. 2023], are preferable due to their scalability. These methods provide a point of view to discover and exploit properties of the underlying BIEs, which enables invention of approaches in-between stochastic methods and boundary element methods (BEMs) like radiosity [Cohen et al. 1993; Keller 1997] and point-based global illumination (PBGI) [Christensen 2008]. Transferred from the rendering tasks, recent studies on Walk-on-Spheres (WoS) [Miller et al. 2023; Sawhney et al. 2023] and Walk-on-Boundary (WoB) [Sugimoto et al. 2023] reveal the possibility of using Monte–Carlo techniques for solving boundary value problems associated to Laplace's and Poisson's equations through BIEs.

3 BACKGROUND

In this section, we briefly review the background of ferrofluid simulation, which generally follows the work of Huang et al. [2019] and Ni et al. [2020] (see §A for details). Main symbols involved throughout the text is listed in Table 2.

3.1 Magnetostatics

Given the ferrofluid volume Ω (as an open set in 3D Euclidean space) and its surface $\partial\Omega$, the magnetic field $\boldsymbol{H}(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^3$) is regarded as the summation of an applied external field $\boldsymbol{H}_{\mathrm{app}}$ and an induced internal field $\boldsymbol{H}_{\mathrm{ind}}$:

$$\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{H}_{\mathrm{app}}(\boldsymbol{x}) + \boldsymbol{H}_{\mathrm{ind}}(\boldsymbol{x}), \tag{1}$$

in which $\boldsymbol{H}_{\mathrm{app}}$ is known. Under the zero-current assumption, $\boldsymbol{H}_{\mathrm{ind}}$ is determined by a special case of Maxwell's equations as

$$\begin{cases} \nabla \cdot \mu_0(\boldsymbol{H}_{\mathrm{ind}} + \boldsymbol{M}) = 0, & (2) \\ \nabla \times \boldsymbol{H}_{\mathrm{ind}} = 0, & (3) \end{cases}$$

where μ_0 is the vacuum permeability, and the magnetization, denoted $\boldsymbol{M}(\boldsymbol{x})$, is determined by

$$\boldsymbol{M} = \begin{cases} \chi\boldsymbol{H}, & \boldsymbol{x} \in \Omega, \quad (4a) \\ \boldsymbol{0}, & \boldsymbol{x} \notin \Omega \cup \partial\Omega, \quad (4b) \end{cases}$$

Table 2. Main symbols involved throughout the text. Each denotes a geometric attribute or a physical quantity. For readers' convenience, whether an attribute or a quantity is to be solved within the IoB method or the conventional fluid simulation is indicated in the table, where two 'no's imply that it is a constant or a parameter.

Symbol†	Definition	To be solved	
		IoB	Fluid
$\boldsymbol{H}$	Magnetic field	Yes	No
$\boldsymbol{H}_{\mathrm{app}}$	Applied magnetic field	No‡	No‡
$\boldsymbol{H}_{\mathrm{ind}}$	Induced magnetic field	Yes	No
μ_0	Vacuum permeability	No	No
$\boldsymbol{M}$	Magnetization	Yes	No
χ	Susceptibility of ferrofluid	No	No
ψ	Magnetic scalar potential	Yes	No
$\boldsymbol{n}$	Normal of fluid surface	No	Yes
$\boldsymbol{u}$	Velocity of fluid	No	Yes
p	Pressure of fluid	No	Yes
ρ	Density of fluid	No	No
$\boldsymbol{g}$	Gravitational acceleration	No	No
γ	Surface tension of fluid	No	No
κ	Mean curvature of fluid surface	No	Yes
ϕ	Single layer potential density	Yes	No
α	Reduced permeability	No	No

† Scalars and vectors are symbolized by italic and bold letters, respectively.
‡ $\boldsymbol{H}_{\mathrm{app}}$ is given in advance but not necessarily constant.

provided that the ferrofluid is isotropic and linearly magnetized with a constant susceptibility χ ($\chi > 0$).

Since $\boldsymbol{H}_{\mathrm{ind}}$ is conservative (see (3)), we can define $\boldsymbol{H}_{\mathrm{ind}} = -\nabla\psi$, where $\psi(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^3$) is a continuous scalar potential, such that Gauss's law for magnetism (2) can be reformulated as

$$\begin{cases} \nabla^2\psi = 0, & \boldsymbol{x} \notin \partial\Omega, \quad (5) \\ \left.\dfrac{\partial\psi}{\partial n}\right|_{+} + \chi\boldsymbol{H}_{\mathrm{app}} \cdot \boldsymbol{n} = (1+\chi)\left.\dfrac{\partial\psi}{\partial n}\right|_{-}, & \boldsymbol{x} \in \partial\Omega, \quad (6) \\ \psi \to 0, & \|\boldsymbol{x}\| \to \infty, \quad (7) \end{cases}$$

which is Laplace's equation (5) subject to the boundary conditions at interface (6) and at infinity (7). According to (6), there is a jump of the derivative $\partial\psi/\partial n = \nabla\psi \cdot \boldsymbol{n}$ ($\boldsymbol{n}$ is the outward-pointing normal) across the interface, causing the normal components and further the values of $\boldsymbol{H}_{\mathrm{ind}}$ and $\boldsymbol{H}$ to be discontinuous at $\partial\Omega$. For the sake of brevity, we use the negative and positive signs to indicate the ferrofluid side and the outer side, respectively.

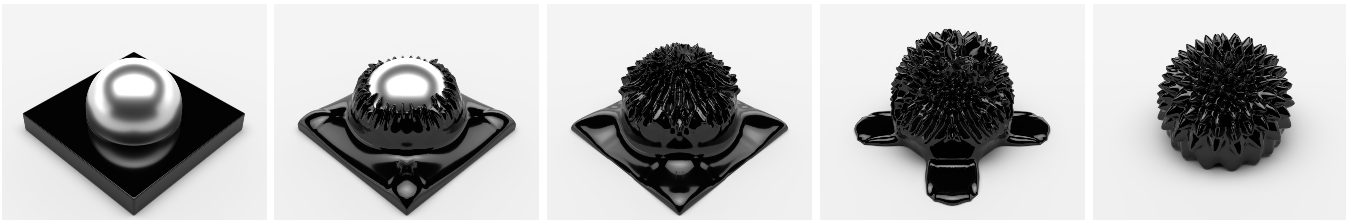


Fig. 3. A magnetized iron ball attracts the surrounding ferrofluid, causing it to form spike structures on its surface.

3.2 Hydrodynamics

Given the solved magnetic field $\boldsymbol{H}$, the free-surface flow of an incompressible inviscid ferrofluid is governed by the Euler equations

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)=-\nabla p+\rho\boldsymbol{g}, \qquad \boldsymbol{x}\in\Omega, \tag{8}$$

subject to $\nabla\cdot\boldsymbol{u}=0$ and the boundary conditions

$$\begin{cases} p-p_{\mathrm{air}}=p_{\mathrm{c}}-p_{\mathrm{m}}, & \boldsymbol{x}\in\partial\Omega\cap\partial\Omega_{\mathrm{a}}, & (9)\\ \boldsymbol{u}\cdot\boldsymbol{n}=\boldsymbol{u}_{\mathrm{solid}}\cdot\boldsymbol{n}, & \boldsymbol{x}\in\partial\Omega\cap\partial\Omega_{\mathrm{s}}, & (10)\end{cases}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ is the velocity, $p(\boldsymbol{x})$ is the pressure, ρ is the density, and $\boldsymbol{g}$ is the gravitational acceleration. Here we use $\partial\Omega_{\mathrm{a}}$ and $\partial\Omega_{\mathrm{s}}$ to denote boundaries of the air and solid regions, respectively, such that $\partial\Omega\cap\partial\Omega_{\mathrm{a}}$ indicates the free surface and $\partial\Omega\cap\partial\Omega_{\mathrm{s}}$ indicates the solid-liquid interface. Note that the air pressure p_{air} is zero in a free-surface model.

In (9), the capillary pressure p_{c} is determined by the surface tension γ and the mean curvature κ as $p_{\mathrm{c}}=2\gamma\kappa$, and the magnetic pressure p_{m}, as an alternative to the volumetric magnetic force of isotropic linear materials [Ni et al. 2020], is calculated by

$$p_{\mathrm{m}}=\frac{1}{2}\mu_0\chi\|\boldsymbol{H}\|^2+\frac{1}{2}\mu_0(\chi\boldsymbol{H}\cdot\boldsymbol{n})^2, \qquad \boldsymbol{x}\in\partial\Omega\cap\partial\Omega_{\mathrm{a}}. \tag{11}$$

Considering the discontinuity, we denote $\boldsymbol{H}$ and related symbols at the interface to be ferrofluid-sided by default.

4 THE INDUCE-ON-BOUNDARY SOLVER

The IoB solver is designed to address the boundary value problem of magnetostatics (5–7) in a numerical way. In this section, we present the continuous formulations underpinning IoB, along with the point-based numerical scheme. We then delve into an analysis of its convergence and offer comparisons regarding computational complexity.

4.1 The Single-Layer Potential

The solution $\psi(\boldsymbol{x})$ to Laplace's equation (5) can be expressed as the single-layer potential with an unknown density $\phi(x)$:

$$\psi(\boldsymbol{x})=\iint_{\partial\Omega}G(\boldsymbol{x},\boldsymbol{y})\,\phi(\boldsymbol{y})\,\mathrm{d}A_y, \tag{12}$$

where $G(\boldsymbol{x},\boldsymbol{y})=1/(4\pi\|\boldsymbol{x}-\boldsymbol{y}\|)$ is a Green's function for Poisson's equation associated with an infinite domain. The normal derivative of $\psi(\boldsymbol{x})$ is then obtained by

$$\left[\frac{\partial\psi}{\partial n}(\boldsymbol{x})\right]_{\pm}=\iint_{\partial\Omega}\frac{\partial G}{\partial n_x}(\boldsymbol{x},\boldsymbol{y})\,\phi(\boldsymbol{y})\,\mathrm{d}A_y\mp\frac{1}{2}\phi(\boldsymbol{x}), \quad \boldsymbol{x}\in\partial\Omega, \tag{13}$$

in which the directional derivative of $G(\boldsymbol{x},\boldsymbol{y})$ with respect to the normal at $\boldsymbol{x}$ is calculated as

$$\frac{\partial G}{\partial n_x}(\boldsymbol{x},\boldsymbol{y})=\boldsymbol{n}_x\cdot\nabla_x G(\boldsymbol{x},\boldsymbol{y}), \tag{14}$$

$$\nabla_x G(\boldsymbol{x},\boldsymbol{y})=\frac{1}{4\pi}\frac{\boldsymbol{y}-\boldsymbol{x}}{\|\boldsymbol{y}-\boldsymbol{x}\|^3}. \tag{15}$$

The boundary condition at infinity (7) has been naturally imposed here due to the particular form of Green's function.

Taking the magnetic boundary condition (6) into account, we can finally formulate a BIE [Lindholm 1980] that express the relationships among the values of ϕ over the ferrofluid boundary:

$$\frac{\phi(\boldsymbol{x})}{2\alpha}=-\iint_{\partial\Omega}\frac{\partial G}{\partial n_x}(\boldsymbol{x},\boldsymbol{y})\,\phi(\boldsymbol{y})\,\mathrm{d}A_y+\boldsymbol{H}_{\mathrm{app}}(\boldsymbol{x})\cdot\boldsymbol{n}, \quad \boldsymbol{x}\in\partial\Omega, \tag{16}$$

where $\alpha=\chi/(2+\chi)$ is the reduced permeability.

This BIE enables a rather simple strategy to calculate the normal component of the ferrofluid-sided magnetic field as

$$\begin{aligned}\boldsymbol{H}(\boldsymbol{x})\cdot\boldsymbol{n}&=\boldsymbol{H}_{\mathrm{app}}(\boldsymbol{x})\cdot\boldsymbol{n}-\left[\frac{\partial\psi}{\partial n}(\boldsymbol{x})\right]_{-}\\&=\boldsymbol{H}_{\mathrm{app}}(\boldsymbol{x})\cdot\boldsymbol{n}+\left(\frac{\phi(\boldsymbol{x})}{2\alpha}-\boldsymbol{H}_{\mathrm{app}}(\boldsymbol{x})\cdot\boldsymbol{n}\right)-\frac{\phi(\boldsymbol{x})}{2}\\&=\frac{1}{\chi}\phi(\boldsymbol{x}), \qquad \boldsymbol{x}\in\partial\Omega.\end{aligned} \tag{17}$$

In addition, the tangential derivative of ψ can be acquired by

$$\frac{\partial\psi}{\partial\boldsymbol{\tau}}(\boldsymbol{x})=\iint_{\partial\Omega}\left(\nabla_x-\boldsymbol{n}_x\frac{\partial}{\partial n_x}\right)G(\boldsymbol{x},\boldsymbol{y})\,\phi(\boldsymbol{y})\,\mathrm{d}A_y, \quad \boldsymbol{x}\in\partial\Omega, \tag{18}$$

because no discontinuity exists in the tangential direction, and then we are able to assemble the ferrofluid-sided magnetic field as

$$\boldsymbol{H}_{\mathrm{ind}}(\boldsymbol{x})=-\frac{1}{2}\phi(\boldsymbol{x})\boldsymbol{n}-\iint_{\partial\Omega}\nabla_x G(\boldsymbol{x},\boldsymbol{y})\,\phi(\boldsymbol{y})\,\mathrm{d}A_y, \tag{19}$$

$$\boldsymbol{H}(\boldsymbol{x})=\boldsymbol{H}_{\mathrm{app}}(\boldsymbol{x})+\boldsymbol{H}_{\mathrm{ind}}(\boldsymbol{x}), \qquad \boldsymbol{x}\in\partial\Omega, \tag{20}$$

which satisfies the requirement for the magnetic pressure (11). For detailed derivations, please refer to §B.1.

4.2 Point-Based Discretization

Provided that the ferrofluid surface $\partial\Omega$ is smooth and represented by a point cloud, in which the i-th point has three attributes: position $\boldsymbol{x}_i$, normal $\boldsymbol{n}_i$, and control area ΔA_i, the IoB approach additionally assigns the single-layer potential density ϕ_i to each point. Since $\boldsymbol{n}_x\cdot\nabla_x G(\boldsymbol{x},\boldsymbol{y})$ tends to zero as $\boldsymbol{y}$ approaches $\boldsymbol{x}$, the integral in (16)

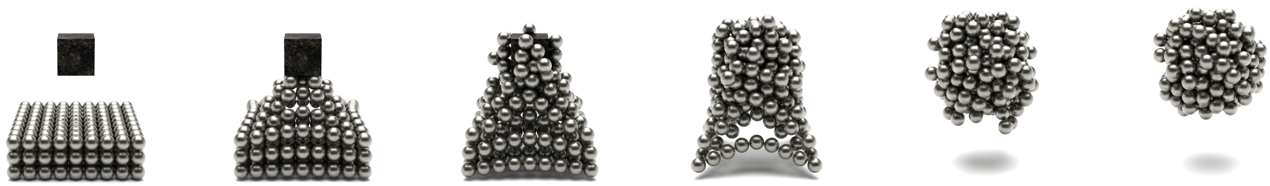


Fig. 4. A cubic magnet stably attracts 243 small metal balls, simulated by integrating our IoB solver into Bullet Physics SDK [Coumans 2015].

ALGORITHM 1: Point-based numerical scheme of the IoB solver

Input: the applied magnetic field $\boldsymbol{H}_{\text{app}}(\boldsymbol{x})$ and the point cloud with attributes $\{\boldsymbol{x}_i\}$, $\{\boldsymbol{n}_i\}$, and $\{\Delta A_i\}$.
Output: the magnetic field $\boldsymbol{H}(\boldsymbol{x}_i)$, evaluated at each point.
$k \leftarrow 0$;
for $i = 1 \to N$ **do** /* Initial guess */
  $\phi_i^k \leftarrow 2\alpha \boldsymbol{H}_{\text{app}}(\boldsymbol{x}_i) \cdot \boldsymbol{n}_i/(1+\alpha)$;
repeat
  $k \leftarrow k+1$;
  for $i = 1 \to N$ **do** /* Accelerated by FMM */
    $\phi_i^k \leftarrow 2\alpha \boldsymbol{H}_{\text{app}}(\boldsymbol{x}_i) \cdot \boldsymbol{n}_i$;
    for $j = 1 \to N$ **do**
      if $i = j$ **then continue**;
      $\phi_i^k \leftarrow \phi_i^k - 2\alpha \boldsymbol{n}_i \cdot \boldsymbol{\nabla}_i G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, \phi_j^{k-1}\, \Delta A_j$;
until $\sum_i \|\phi_i^k - \phi_i^{k-1}\| < \varepsilon$ /* ε is a tolerance */;
for $i = 1 \to N$ **do** /* Accelerated by FMM */
  $\boldsymbol{H}(\boldsymbol{x}_i) \leftarrow \boldsymbol{H}_{\text{app}}(\boldsymbol{x}_i) - \phi(x_i)\boldsymbol{n}/2$;
  for $j = 1 \to N$ **do**
    if $i = j$ **then continue**;
    $\boldsymbol{H}(\boldsymbol{x}_i) \leftarrow \boldsymbol{H}(\boldsymbol{x}_i) - \boldsymbol{\nabla}_i G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, \phi(\boldsymbol{x}_j)\, \Delta A_j$;

is not singular, such that (16) can be directly discretized as

$$\phi_i = -2\alpha \sum_{j=1, j\neq i}^{N} \frac{\partial G}{\partial n_i}(\boldsymbol{x}_i, \boldsymbol{x}_j)\, \phi_j\, \Delta A_j + 2\alpha \boldsymbol{H}_{\text{app}}(\boldsymbol{x}_i) \cdot \boldsymbol{n}_i, \qquad (21)$$

for all i from 1 to N where N denotes the number of points.

We propose to solve (21) by fixed-point iterations due to its intrinsic property (see §4.3). After $\{\phi_i\}$ is calculated, it is natural to acquire $\boldsymbol{H}_{\text{ind}}(\boldsymbol{x}_i)$ for each i by similarly discretizing (19) as

$$\boldsymbol{H}_{\text{ind}}(\boldsymbol{x}_i) = -\frac{1}{2}\phi_i \boldsymbol{n} - \sum_{j=1, j\neq i}^{N} \boldsymbol{\nabla}_i G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, \phi(\boldsymbol{x}_j)\, \Delta A_j. \qquad (22)$$

Note that the summation terms in (21) and (22) both can be accelerated by FMM. The algorithm is summarized in Alg. 1, which can be seen as a scheme of the Nyström method [Tong and Chew 2020].

In addition, overly close point pairs may cause numerical issues, an effective workaround is replacing $G(\boldsymbol{x}, \boldsymbol{y})$ by $\tilde{G}(\boldsymbol{x}, \boldsymbol{y}) = 1/[4\pi \max(\|\boldsymbol{x} - \boldsymbol{y}\|, \varepsilon_{\text{d}})]$, where $\varepsilon_{\text{d}} \ll 1$ is a threshold.

4.3 Convergence Analysis

First, we consider convergence for BIEs of the single-layer potential. It is convenient to define an operator $\mathcal{K}$ that satisfies

$$\mathcal{K}\phi(\boldsymbol{x}) = -\iint_{\partial\Omega} 2\frac{\partial G}{\partial n_x}(\boldsymbol{x}, \boldsymbol{y})\, \phi(\boldsymbol{y})\, \mathrm{d}A_y, \qquad (23)$$

and then (16) can be reformulated as a Fredholm integral equation of the second kind:

$$\phi(\boldsymbol{x}) = \alpha\mathcal{K}\phi(\boldsymbol{x}) + 2\alpha f(\boldsymbol{x}), \qquad (24)$$

where $f(\boldsymbol{x}) = \boldsymbol{H}_{\text{app}}(\boldsymbol{x}) \cdot \boldsymbol{n}$ is the boundary term. Such an equation is solved in a standard way by a Neumann series as follows:

$$\begin{aligned}\phi(\boldsymbol{x}) &= (\mathcal{I} - \alpha\mathcal{K})^{-1} 2\alpha f(\boldsymbol{x}) \\ &= \left(\mathcal{I} + \alpha\mathcal{K} + \alpha^2\mathcal{K}^2 + \alpha^3\mathcal{K}^3 + \cdots\right) 2\alpha f(\boldsymbol{x}),\end{aligned} \qquad (25)$$

in which $\mathcal{I}$ denotes the identity operator. Given that the characteristic values of $\alpha\mathcal{K}$, denoted $\lambda_1, \lambda_2, \ldots$ ($|\lambda_1| \leq |\lambda_2| \leq \ldots$), are such values of λ that the equation

$$(\mathcal{I} - \alpha\lambda\mathcal{K})\phi(\boldsymbol{x}) = 0 \qquad (26)$$

has non-zero solutions, the above series converges if and only if $|\lambda_1| > 1$. According to the property that $\lambda = -1$ is the non-zero solution with the least modules for $(\mathcal{I} - \lambda\mathcal{K})\phi = 0$ [Sabelfeld and Simonov 2016, pp. 14, 37, 44], it is easy to prove $\lambda_1 = -1/\alpha$. Thus we conclude that a sufficient condition for the convergence of (25) is $|\alpha| < 1$, which holds true as long as χ is positive.

Now we proceed to investigate the convergence of the numerical scheme proposed in §4.2. It is evident that (21) converges to (16) as the point cloud tends to infinitely dense. In fact, the points with attributes can be regarded as constant elements in the BEM [Sauter and Schwab 2011]. Therefore, the preceding analysis applies equally to the numerical scheme as well: one can naturally replace $\mathcal{K}$ with a discretized version and establish connections between the fixed-point iterations and the Neumann series. Generally speaking, the convergence rate of Alg. 1 linearly depends on the spectral radius $1/|\lambda_1| = \alpha = \chi/(\chi+2)$ ($\chi > 0$).

Intuitive explanation. Here we provide an intuitive explanation of the IoB solver. As noted by Pharr et al. [2023], when solving the rendering equation with ray tracing, each additional bounce of the ray corresponds to the inclusion of a higher-order term in the Neumann series, where the original illuminated surfaces are treated as virtual light sources. Similarly, during every iteration of the IoB solver, the original solution of $\boldsymbol{H}_{\text{ind}}$ (derived from ϕ) can be regarded as a virtual applied magnetic field. The material is magnetized by

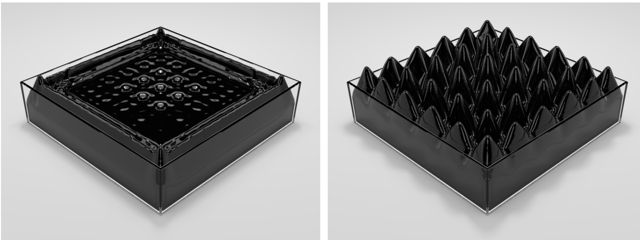
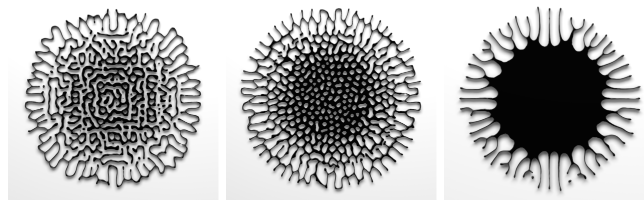


Fig. 5. In the presence of a uniform magnetic field, a ferrofluid within the tank undergoes a gradual transformation, developing distinctive spikes that align precisely along the direction of the magnetic field.



A. Thickness = 2.4 mm; B. Thickness = 3.6 mm; C. Thickness = 5.0 mm.

Fig. 6. A circular thin layer of ferrofluid is confined within a 5.0 mm gap formed by two glass plates. Under a uniform magnetic field perpendicular to the glass, the ferrofluid stretches and forms complex labyrinth structures. Growth pattern changes depending on its initial thickness.

this field to a higher-order term and subsequently induces a higher-order magnetic field that in turn contributes to the solution. This magnetization-and-induction process continues to occur within an infinitesimally small time interval and finally reaches equilibrium, which is numerically emulated by the solver.

Acceleration. Furthermore, we point out that it is possible to accelerate the convergence of iterations by rewriting (25) as

$$\begin{aligned}\phi(\boldsymbol{x}) &= \frac{1}{1+\alpha}\phi(\boldsymbol{x}) + \frac{\alpha}{1+\alpha}\phi(\boldsymbol{x}) \\ &= \frac{1}{1+\alpha}\left(\mathcal{I} + \alpha(\mathcal{I}+\mathcal{K}) + \alpha^2(\mathcal{K}+\mathcal{K}^2) + \cdots\right)2\alpha f(\boldsymbol{x}) \\ &= \left(\frac{\mathcal{I}}{1+\alpha} + \lim_{m\to\infty}\sum_{k=1}^{m}\frac{\alpha(\mathcal{I}+\mathcal{K})}{1+\alpha}(\alpha\mathcal{K})^{k-1}\right)2af(\boldsymbol{x}). \qquad (27)\end{aligned}$$

This reformulation eliminates $\lambda_1 = -1/\alpha$ from the characteristic values of the summands, such that the new series converges at the rate of $1/|\lambda_2| \propto \alpha$, which is consistently faster than the original version since $|\lambda_2| \neq |\lambda_1|$ [Sabelfeld and Simonov 2016, pp. 37, 44]. For iterative solutions that truncate the series of (27) at some finite m as follows:

$$\begin{aligned}\phi(\boldsymbol{x}) &= \left(\frac{\mathcal{I}}{1+\alpha} + \sum_{k=1}^{m}\frac{\alpha(\mathcal{I}+\mathcal{K})}{1+\alpha}(\alpha\mathcal{K})^{k-1}\right)2\alpha f(\boldsymbol{x}) \\ &= \sum_{k=0}^{m-1}(\alpha\mathcal{K})^k 2\alpha f(\boldsymbol{x}) + (\alpha\mathcal{K})^m\frac{2\alpha}{1+\alpha}f(\boldsymbol{x}), \qquad (28)\end{aligned}$$

it is clear that the only modification is the inclusion of the factor $1/(1+\alpha)$ in the term with the highest order. This is reflected in the selection of the initial guess for our IoB solver (Alg. 1). A similar strategy has been adopted in solving Dirichlet problems by a Walk-on-Boundary method [Sugimoto et al. 2023].

4.4 Complexity Comparisons

The IoB approach has an outstanding advantage in that its equations are aligned with the assignments of an FMM [Beatson and Greengard 1997]: both (21) and (22) can be seen as N-body problems, with the kernels depending solely on the positions of the source points. Accelerated by the FMM, the time complexity of Alg. 1 is $O(N\log\varepsilon\log\varepsilon')$, where ε and ε' represent the numerical tolerances for fixed-point iterations and multipole expansions, respectively. Since only a tree structure needs to be constructed, the space complexity is also limited to $O(N)$.

Compared to the existing methods, the time and space complexities of our approach are both the lowest, which is illustrated in Table 1. Here, we assume $\log\varepsilon'$ is a constant and let $K = \log\varepsilon^{-1}$ be the number of iterations. For the purpose of fair comparisons, the following quantities are considered to be of the same order of magnitude: $\sqrt[3]{N_\mathrm{p}}$, $\sqrt{N}$, and n, where N_p is the number of particles used in SPH methods and n is the grid resolution of an Eulerian approach. Note that the iteration numbers are roughly the same (~ 10) according to reported experiments.

5 COUPLING WITH FLUID SIMULATION

The IoB magnetostatic solver introduced in §4 is agnostic to the underlying fluid simulation framework. In this section, we use a conventional grid-based fluid simulation as a representative example to showcase the straightforward integration of our IoB solver into an existing simulation pipeline.

5.1 The Coupling Method

Grid-based fluid simulation employs the level-set method for surface tracking. Specifically, a signed distance field (SDF) $\varphi(\boldsymbol{x})$ is defined as the level-set function and updated through advection and reinitialization at each time step. For any point on the surface, the outward-pointing normal and the mean curvature are calculated by $\boldsymbol{n} = \nabla\varphi/\|\nabla\varphi\|$ and $\kappa = \nabla\cdot\boldsymbol{n}$, respectively.

First, it is a logical step to discretize (9) by the ghost fluid method, because as the only new introduced term, the magnetic pressure p_m serves a similar role to the capillary pressure p_c in the boundary condition. Suppose that the grid cell (i,j,k) is in the ferrofluid (i.e., $\varphi_{i,j,k} \le 0$) and an adjacent cell, e.g., $(i+1,j,k)$, is in the air (i.e., $\varphi_{i+1,j,k} > 0$), as proposed by Kang et al. [2002; 2000], the location of the interface is at $(i+\theta,j,k)$ where

$$\theta = \frac{\varphi_{i,j,k}}{\varphi_{i,j,k} - \varphi_{i+1,j,k}}. \qquad (29)$$

Subsequently, with κ and $\boldsymbol{H}$ evaluated at $(i+\theta,j,k)$, the boundary condition for fluid pressure is discretized as

$$p_{i+1,j,k} = \frac{2\gamma\kappa - \frac{1}{2}\mu_0\chi\|\boldsymbol{H}\|^2 - \frac{1}{2}\mu_0(\chi\boldsymbol{H}\cdot\boldsymbol{n})^2 - (1-\theta)p_{i,j,k}}{\theta}, \quad (30)$$

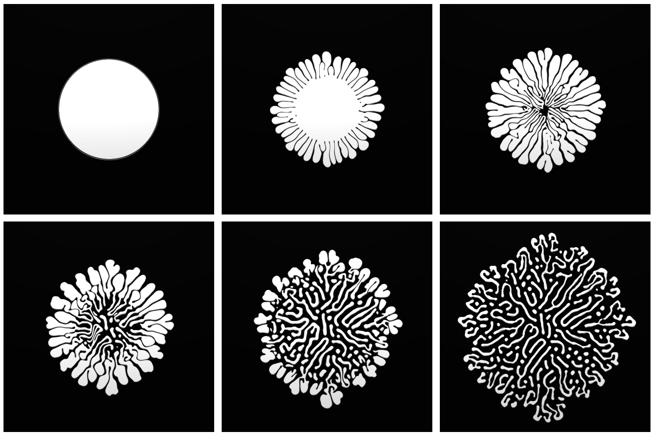


Fig. 7. A thin layer of ferrofluid with a circular cavity is confined between two glass panes. Driven by a uniform magnetic field perpendicular to the panes, it extends tentacles converge at the center. Subsequently, these intricate patterns diffuse like ink spreading through water.

of which the left-hand side is a ghost pressure.

Second, the marching cubes algorithm [Lorensen and Cline 1987], commonly used as a preprocessing step for rendering in a conventional pipeline, can be used to generate the input point cloud of Alg. 1. The triangular mesh extracted from the level-set function φ using the marching cubes algorithm possesses several essential properties for our purposes:

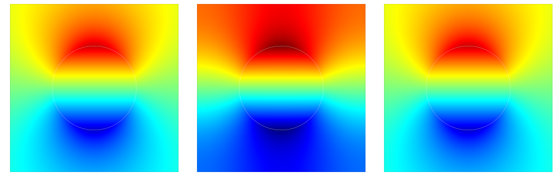
- It is fine-grained that the distance between neighbouring vertices is less than $\sqrt{3}\Delta x$, where Δx is the grid spacing.
- Its vertex positions align precisely with all the interface points required for evaluation in the ghost fluid method.
- Local vertex areas can be readily determined based on its topological characteristics.

Therefore, in practical implementation, we employ all the vertices from the marching-cubes mesh as the input point cloud of Alg. 1. For each point within this cloud, its position corresponds to the vertex position, and the control area is estimated as one-third of the total area of all adjacent faces connected to the vertex. Simultaneously, we compute the point's normal vector by applying finite difference to the level-set function φ, thereby achieving smoother results.

5.2 The Full Pipeline

For the sake of reproducibility, we present the pipeline of grid-based ferrofluid simulation during each time step as follows:

(1) Determine the time-step size Δt as $c\Delta x/u_{\max}$, where c is the CFL number and $u_{\max}$ is the maximum velocity;
(2) Advect φ and $\boldsymbol{u}$ by $\varphi \leftarrow \varphi - \Delta t(\boldsymbol{u} \cdot \nabla\varphi)$ and $\boldsymbol{u} \leftarrow \boldsymbol{u} - \Delta t(\boldsymbol{u} \cdot \nabla\boldsymbol{u})$ using a semi-Lagrangian method [Staniforth and Côté 1991];
(3) Reinitialize φ by solving $\partial\varphi/\partial\tau + \operatorname{sgn}(\varphi)(|\nabla\varphi| - 1) = 0$ with HJ WENO discretization [Jiang and Peng 2000];
(4) Extract a triangular mesh $\mathcal{M}$ from φ by marching cubes;
(5) Generate the input point cloud of the IoB solver based on $\mathcal{M}$ and then perform Alg. 1 to calculate point-based values of $\boldsymbol{H}$;
(6) Apply gravity by $\boldsymbol{u} \leftarrow \boldsymbol{u} + \Delta t\boldsymbol{g}$;



(a) Ground truth; (b) Grid-based ($\text{AE}_{\max} \approx 0.1222$); (c) **IoB (FMM)** ($\text{AE}_{\max} \approx 0.0014$).

Fig. 8. Comparisons of different solvers with respect to accuracy in modeling magnetic potential generated by a magnetized circle within a square domain. Higher potential is mapped to warmer color in the visualization, and the maximum absolute error of each solver is reported in the captions.

(7) Solving Poisson's equation

$$\begin{cases} \nabla \cdot \left(\boldsymbol{u} - \dfrac{\Delta t}{\rho}\nabla p\right) = \beta, & \boldsymbol{x} \in \Omega, \quad (31a) \\ p = p_{\text{air}} + p_{\text{c}} - p_{\text{m}}, & \boldsymbol{x} \in \partial\Omega \cap \partial\Omega_{\text{a}}, \quad (31b) \\ \dfrac{\Delta t}{\rho}(\boldsymbol{n} \cdot \nabla p) = \boldsymbol{n} \cdot (\boldsymbol{u} - \boldsymbol{u}_{\text{solid}}), & \boldsymbol{x} \in \partial\Omega \cap \partial\Omega_{\text{s}}, \quad (31c) \end{cases}$$

for p, where β is determined by volume control [Kim et al. 2007], (31b) is discretized using the ghost fluid method, and (31c) is discretized with the method of Batty et al. [2007];
(8) Project $\boldsymbol{u}$ by $\boldsymbol{u} \leftarrow \boldsymbol{u} - \Delta t\nabla p/\rho$;
(9) Damp $\boldsymbol{u}$ by $\boldsymbol{u} \leftarrow \mathrm{e}^{-\eta\Delta t}\boldsymbol{u}$, where η is the damping coefficient;
(10) Extrapolate $\boldsymbol{u}$ into the air and solid regions;
(11) Export $\mathcal{M}$ for rendering if necessary.

It is worth emphasizing that we have expanded the grid-based fluid simulation pipeline by introducing just one additional step (Step (5)), and made slight modification of the free-surface boundary condition (31b). Given that fluid simulation is not our primary focus, we employ an artificial viscosity approach by damping the velocity (Step (9)) as a temporary solution. This can certainly be substituted with a physically accurate viscosity solver [Batty and Bridson 2008].

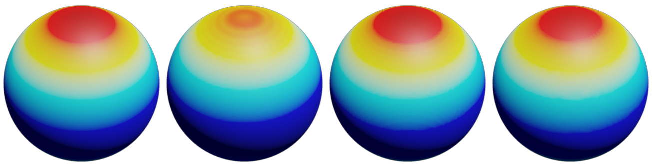
The current implementation is based on an underlying marker-and-cell (MAC) grid [Harlow and Welch 1965]. For details of methods used in the above steps, please refer to the books of Bridson [2015] or Osher and Fedkiw [2005].

6 EXPERIMENTAL RESULTS

We have validated the IoB magnetostatic solver via a parallel CPU implementation, with the assistance of the FMMTL library [Cecka and Layton 2015] to accelerate summations in Alg. 1. With a minor modification to substitute G with $\tilde{G}$ (see §4.2), we directly use the `LaplaceSpherical` kernel built in FMMTL for this task. The order of the multipole expansion is set to 6. In this section, we begin by showcasing benchmark tests of the solver for magnetostatic fields computations, followed by experiments where it is integrated into the grid-based fluid solver to simulate ferrofluids.

6.1 Solving Magnetostatic Fields

In comparison experiments, we include the existing solvers that adopt the interfacial force model, i.e., the grid-based [Ni et al. 2020] and surface-only [Huang and Michels 2020] approaches, alongside



(a) Ground truth; (b) Grid-based ($\mathrm{RE}_{\max} \approx 8.39\%$); (c) Surface-only ($\mathrm{RE}_{\max} \approx 1.80\%$); (d) **IoB (FMM)** ($\mathrm{RE}_{\max} \approx 2.44\%$).

Fig. 9. Comparison of different solvers with respect to accuracy of the magnetic pressure on a sphere surface. Higher pressure is mapped to warmer color in the visualization, and the maximum relative error of each solver is reported in the captions.

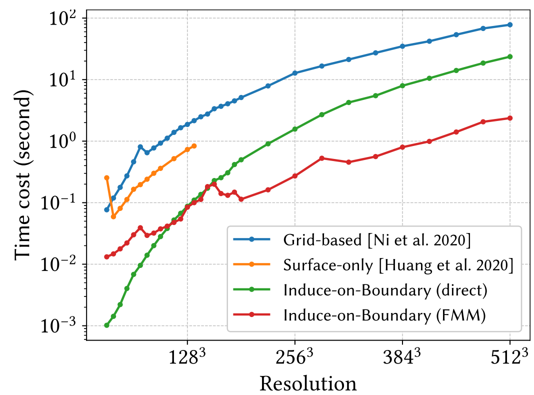


Fig. 10. Comparison of different solvers in terms of computation time on Eulerian grids with various resolutions. The timing measurement starts when a solver achieves a relative residual reduction below 10^{-6}. Note that the surface-only solver is implemented with GPU acceleration.

the IoB solver. For the sake of fairness, all comparisons are performed on the same grid for each solver. The background grid serves the purpose of discretizing the equations and approximating the boundary condition at infinity through magnetic shielding for the grid-based solver. It also provides the input surface mesh or point cloud for the other two solvers through marching cubes.

The timings here are measured on an AMD EPYC 9654 processor without GPU acceleration (for the IoB and grid-based solvers) and an AMD Ryzen 9 7950X processor with an Nvidia Geforce RTX 4090 graphics card (for the surface-only solver).

Accuracy (2D). A magnetic circle with a radius of 1 m and a susceptibility of 1 is centered at the origin. When an external field $\boldsymbol{H}_{\text{app}} = (0, 1\,\mathrm{A\,m^{-1}})$ is applied, the magnetic scalar potential of the induced field can be analytically determined by

$$\psi(\boldsymbol{x}) = \begin{cases} \dfrac{y}{3}\,\mathrm{A\,m^{-1}}, & \|\boldsymbol{x}\| \le 1\,\mathrm{m}, \quad (32a) \\[2mm] \dfrac{y}{3(x^2+y^2)}\,\mathrm{A\,m}, & \|\boldsymbol{x}\| > 1\,\mathrm{m}. \quad (32b) \end{cases}$$

Given a grid with a resolution of 1024^2, and coordinates confined to the range of $[-2\,\mathrm{m}, 2\,\mathrm{m}]$, we employ both the grid-based and IoB approaches to compute ψ and compare the solutions with the ground truth within the domain occupied by the grid. Fig. 8 visualizes results of the accuracy test, with the potential of each solver aligned at the center. Compared to the grid-based approach, the maximum absolute error of the IoB solver measured at grid points is two orders of magnitude smaller, attributed to the correct application of the boundary conditions.

Accuracy (3D). Similar to the 2D case, a magnetic sphere ($r = 1\,\mathrm{m}$, $\chi = 1$) is centered at the origin. Under the action of an external field $\boldsymbol{H}_{\text{app}} = (0, 1\,\mathrm{A\,m^{-1}}, 0)$, the total magnetic field within the sphere is $\boldsymbol{H} = (0, 0.75\,\mathrm{A\,m^{-1}}, 0)$, such that the magnetic pressure has an accurate solution

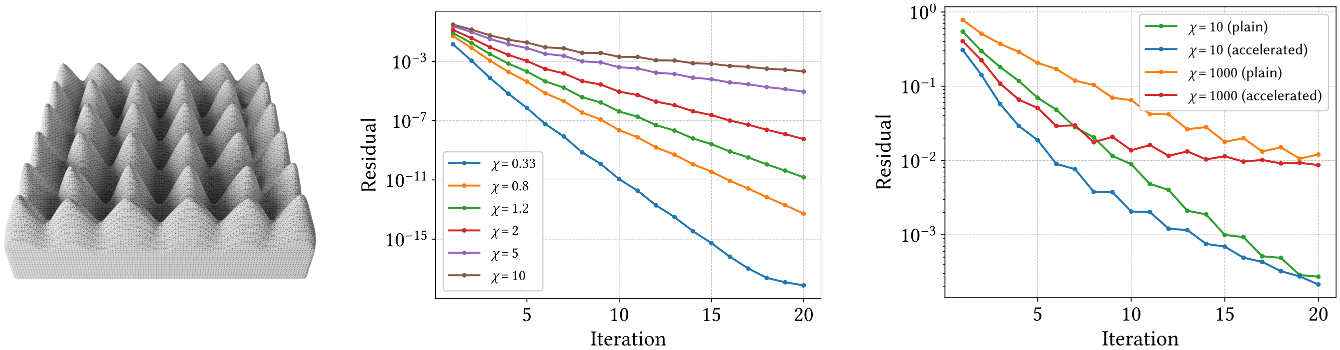
$$p_{\mathrm{m}} = \frac{9\mu_0}{32}\left(1 + \frac{y^2}{\|\boldsymbol{x}\|^2}\right)\mathrm{A^2\,m^{-2}}, \quad \|\boldsymbol{x}\| = 1\,\mathrm{m}. \tag{33}$$

We employ the grid-based, surface-only, and IoB solvers to compute p_{m} upon a grid with a resolution of 128^3, whose occupied domain is $-2\,\mathrm{m} \le x, y, z \le 2\,\mathrm{m}$. As illustrated in Fig. 9, the magnetic pressure calculated by the grid-based solver obviously differs from the ground truth especially around the poles of the sphere, due to its compromise on physical accuracy. The surface-only and IoB solvers produce similar results, although the former exhibits a smaller maximum relative error compared to the latter, thanks to the sophisticated Galerkin BEM.

Performance. To assess the performance of different solvers, we expand the 3D accuracy test scenario to grids with a range of resolutions, spanning from 32^3 to 512^3. During the experiment, we record the time required for each solve to reduce the relative residual to less than 10^{-6}. As illustrated in Fig. 10, the grid-based solver generally takes the longest solving time for the magnetic field. The surface-only solver is limited to handling resolutions no greater than 136^3 due to memory constraints arising from the assembly of dense matrices, whereas the IoB solver demonstrates the exceptional scalability thanks to its lower complexities. Note that there is a large constant factor in the time complexity of the FMM-accelerated IoB solver. A direct summation version can be a better choice with resolutions lower than 100^3 ($N \approx 12\,\mathrm{k}$).

Convergence. In Fig. 11, we present experimental results to validate the convergence of the IoB solver. The test scenario involves a complex mesh with nearly 10^5 vertices, as depicted in Fig. 11a. It is clear to see from Fig. 11b that the convergence rate approximately follows a linear pattern with respect to $\alpha = \chi/(\chi+2)$, as proved in §4.3, and the IoB solver only takes about ten iterations to reduce the residual to a fairly low magnitude, when the susceptibility remains in a moderate range ($\chi < 5$). Furthermore, we examine the convergence of IoB solver with and without the acceleration strategy described in §4.3. As demonstrated in Fig. 11c, the plain scheme, which excludes the factor $1/(1+\alpha)$ from the initial guess in Alg. 1, displays less favorable convergence behavior compared to the accelerated scheme, especially when dealing with high susceptibility values (e.g., $\chi = 1000$).

Ablation study. Moreover, we conduct ablation experiments comparing the IoB solver with conventional solvers for solving the system of linear equations (21). Since the system is neither symmetric nor positive definite, we select the biconjugate gradient stabilized (BiCGSTAB) and generalized minimal residual (GMRES) methods



(a) The test mesh (73 868 vertices); (b) Residual plots with different susceptibilities; (c) Residual plots with and without acceleration.

Fig. 11. Convergence validation of the IoB solver, upon a complex mesh presenting the normal-field instability of ferrofluids (Fig. 11a). The experimental results verify the conclusions derived in §4.3: the convergence rate of the IoB solver linearly depends on α (Fig. 11b), and the convergence can be accelerated by the inclusion of the factor $1/(1+\alpha)$ in the initial guess (Fig. 11c).

as the conventional counterparts, where diagonal and smoothed aggregation multigrid preconditioners are utilized. To ensure a fair comparison, no FMM acceleration is employed for any approach. Based on the results shown in Figs. 12 and 13, we conclude that the IoB solver delivers superior performance across various levels of discretization as long as χ is less than 50, which aligns well with the requirements for simulating large-scale ferrofluids.

6.2 Simulating Ferrofluids

By integrating the IoB solver into a grid-based fluid simulation pipeline, we can easily create vivid animations of ferrofluids. During our implementation, we utilize the AMGCL library [Demidov 2019] to solve Poisson's equation for the pressure. The specifications and statistics of the simulations are reported in Table 3.

6.2.1 Normal-field instability. The normal-field instability in ferrofluids arises due to the interplay of magnetic, gravitational, and capillary forces, leading to the formation of regular patterns or spikes aligned parallel to the magnetic field direction.

Uniform field. A ferrofluid is initially stable inside a tank. Upon the application of a uniform magnetic field, the ferrofluid is magnetized and induces an internal field that triggers the formation of spikes parallel to the magnetic field (Fig. 5).

Dipole source. A blob of ferrofluid is placed on the tabletop, with a magnet approximated as a dipole located beneath it. After magnetization, the ferrofluid clusters and shaped into (more than one hundred) radial spikes (Fig. 2).

Magnetized ball. A highly magnetized iron ball (with a susceptibility of 5000) is positioned in the center of a ferrofluid. The ball behaves as a magnet, attracting the ferrofluid and causing it to form spike structures around the sphere (Fig. 3).

Lifting ferrofluid. This experiment is divided into four phases. Initially, a ferrofluid is magnetized by a magnet below, shaped into spikes on the ground. Second, another magnet is placed above. A portion of the fluid is lifted and forms spike structures on the ceiling. Later, the magnet underneath is removed, so that all the fluid is attracted to the above one. In the end, the top magnet is also moved away, and thus the ferrofluid falls back (Fig. 1).

6.2.2 Labyrinthine instability. When magnetic fields are applied perpendicular to trapped thin layers of ferrofluids, fascinating patterns emerge, resembling maze or labyrinthine structures, with walls and pathways formed by the ferrofluid.

Patterns in relief. A thin layer of circular ferrofluid is sandwiched between two glass panes, and a uniform field is applied perpendicular to the panes. By adjusting the initial thickness of ferrofluids, our simulations produce various maze-like relief patterns (Fig. 6).

Patterns in intaglio. A thin layer of ferrofluid with a circular cavity in the middle is sandwiched between two glass panes. A uniform field is applied perpendicular to the panes, giving rise to maze-like patterns in intaglio (Fig. 7).

6.2.3 Extension to solids. The IoB solver is agnostic to the underlying numerical simulation framework, making it highly adaptable to various simulation systems. By implementing the solver as a plugin of Bullet Physics SDK [Coumans 2015], an animation where a magnet attracts 243 iron balls is obtained (Fig. 4). In spite of the relatively high susceptibility ($\chi = 50$), the IoB solver still shows excellent convergence, whose residual generally decreases from 10^4 to 10^{-2}, 10^{-3}, and 10^{-5} after 10, 15, and 20 iterations, respectively.

7 CONCLUSION AND DISCUSSION

We have introduced IoB, an innovative magnetostatic solver based on the single-layer potential, which can be readily adapted to simulating ferrofluids on Eulerian grids. At the heart of our solver, highly efficient fixed-point iterations are utilized to address the magnetostatic governing equations. We have not only conducted a theoretical analysis of the algorithm's convergence but also carried out extensive experiments to showcase the IoB solver's lightweight, fast, and accurate characteristics. It emerges as a valuable tool for determining the magnetic pressure of ferrofluids and can be seamlessly integrated into grid-based fluid simulation pipelines.

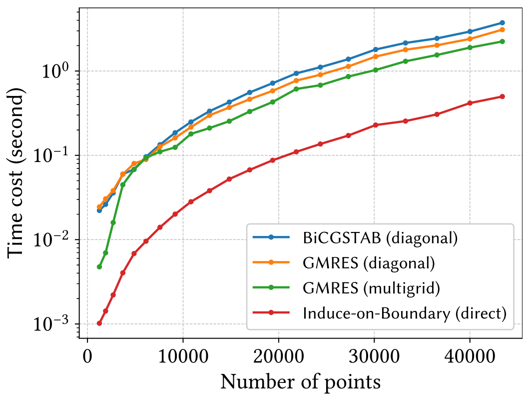


Fig. 12. Comparison of different solvers in terms of computation time in the scenario of a magnetic sphere with $\chi = 1$ (see Fig. 9). The experiments are conducted on grids with resolutions ranging from 32^3 to 192^3, and the number of points is reported in the figure. For each experiment, the time cost is measured once the relative residual is reduced to less than 10^{-6}.

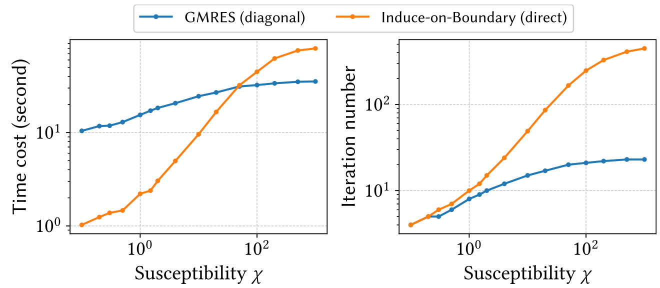


Fig. 13. Comparisons between GMRES (diagonal) and IoB (direct) in terms of both time cost and iteration number required to reduce the relative residual below 10^{-6}. The experiments are conducted upon a complex mesh (Fig. 11a) with various susceptibilities. Note that the multigrid-preconditioned GMRES is not compared here due to the impracticality to allocate memory for multigrid structures of such high-dimensional ($\sim 10^5$) dense systems.

Relation with CG methods. The underpinning equation of the IoB solver is a Fredholm integral equation of the second kind, bearing resemblance to the rendering equation [Pharr et al. 2023], as well as those in WoS [Qi et al. 2022] and WoB [Sugimoto et al. 2023]. As a point-based deterministic approach, the IoB solver can conceptually relate to rendering techniques such as radiosity [Cohen et al. 1993] and PBGI [Christensen 2008]. Moreover, the solver's computational paradigm, which traces from source points rather than field points, also aligns with the principles of adjoint estimators found in both rendering [Christensen 2003] and WoB [Sabelfeld and Simonov 1994]. Such similarities inspire us to design the presented IoB solver and also broaden our horizons to develop further improvements.

Choice of the single-layer potential. As pointed out by Huang and Michels [2020], the double-layer potential gives a more accurate solution than the single-layer one when calculating magnetic fields by a Galerkin BEM. However, there are some shortcomings of the double-layer potential that restrict us to invent a lightweight solver like IoB. First, the scalar potential of the applied field must be provided to calculate the potential density, while the single-layer one only requires the field itself. Second, the kernel of the double-layer BIEs depends on the field points rather than the source point (see §B.2), which makes it indirect to accelerate summations by FMM.

Limitations and future work. Our current simulation results exhibit minor jittering artifacts. Our observations suggest that, beyond the inherent shortcomings of the single-layer potential and point-based discretization, the explicit surface tension scheme and the volume loss issue associated with the level-set method significantly contribute to these artifacts. To mitigate these effects, potential solutions include (a) adopting implicit methods for surface tension, (b) utilizing more effective volume control techniques, and (c) developing an accurate and robust viscosity solver to stabilize surface movements properly. In this paper, we have exclusively validated the effectiveness of the IoB solver within a fully Eulerian framework, where the level-set method is employed to track the fluid surface. We are greatly interested in incorporating the solver into hybrid frameworks, e.g., PIC/FLIP, where the fluid surface is reconstructed based on advected particles [Yu and Turk 2013]. There is also potential for improving the accuracy and performance of the proposed coupling method (§5.1), for instance, by optimizing the marching-cubes mesh or exploiting temporal coherence. Furthermore, as demonstrated in §6.2, leveraging IoB solver for simulating magnetic solids presents another promising avenue. The point-based nature of the solver opens up the possibility of devising an implicit numerical scheme for magnetic forces, which can be seamlessly integrated with IPC [Li et al. 2020] and implicit elasticity [Sifakis and Barbic 2012].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This work was supported in part by National Key R&D Program of China (2022ZD0160801) and Shenzhen Collaborative Innovation Program (CJGJZD2021048092601003). We would like to acknowledge Libo Huang from miHoYo Co., Ltd. for help on comparison with the surface-only ferrofluid solver. The Houdini Education license is credited for the video generations.

REFERENCES

Ryoichi Ando and Christopher Batty. 2020. A Practical Octree Liquid Simulator with Adaptive Surface Resolution. *ACM Trans. Graph.* 39, 4, Article 32 (aug 2020), 17 pages.

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A Fast Variational Framework for Accurate Solid-Fluid Coupling. In *ACM SIGGRAPH 2007 Papers* (San Diego, California) *(SIGGRAPH '07)*. Association for Computing Machinery, New York, NY, USA, 100–es.

Christopher Batty and Robert Bridson. 2008. Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland) *(SCA '08)*. Eurographics Association, Goslar, DEU, 219–228.

Rick Beatson and Leslie Greengard. 1997. *A short course on fast multipole methods*. Oxford University Press, New York, NY, USA, 1–37.

Morten Bojsen-Hansen and Chris Wojtan. 2013. Liquid Surface Tracking with Error Compensation. *ACM Trans. Graph.* 32, 4, Article 68 (jul 2013), 13 pages.

Robert Bridson. 2015. *Fluid simulation for computer graphics* (2nd ed.). AK Peters/CRC Press, Boca Raton, FL, USA.

Cris Cecka and Simon Layton. 2015. FMMTL: FMM Template Library A Generalized Framework for Kernel Matrices. In *Numerical Mathematics and Advanced Applications - ENUMATH 2013*, Assyr Abdulle, Simone Deparis, Daniel Kressner, Fabio Nobile, and Marco Picasso (Eds.). Springer International Publishing, Cham, 611–620.

Table 3. Specifications and statistics of the simulations. For each case, the simulation runs on either Machine A (with an AMD EPYC 9654 processor) or Machine B (with an AMD Ryzen 9 7950X processor). The computation time of both the IoB solver and the full pipeline is measured per time step.

Case	General settings				Simulation parameters†				Problem size		Computation time			
	Resolution	Mach.	Frame‡		CFL #	Spacing	Suscept.	Damping	Time step	Max. point	IoB solver		Full pipeline	
			Interval	Count	c	$\Delta x/\text{mm}$	χ	η/s^{-1}	count	number	Maximum	Average	Average	Total
Uniform	$192 \times 144 \times 192$	A	2 ms	401	0.5	0.638	0.33	8	2318	249.7 k	6.756 s	3.187 s	6.091 s	3.92 h
Dipole	$256 \times 128 \times 256$	B	2 ms	400	0.5	0.397	0.45	8	3134	128.6 k	14.317 s	6.090 s	9.621 s	8.38 h
Ball	$192 \times 192 \times 192$	A	2 ms	400	1.0	0.426	0.8	16	2325	168.9 k	5.504 s	2.719 s	5.120 s	3.31 h
Lifting	$320 \times 224 \times 320$	A	2 ms	651	1.0	0.316	1.0	16	8297	423.5 k	11.385 s	4.400 s	9.689 s	22.33 h
Relief A	$768 \times 20 \times 768$	A	250 µs	401	1.0	0.314	0.8	8	645	1.253 M	17.028 s	7.052 s	11.994 s	2.15 h
Relief B	$768 \times 20 \times 768$	A	250 µs	401	1.0	0.314	0.8	8	702	1.047 M	10.576 s	6.059 s	11.049 s	2.15 h
Relief C	$1024 \times 25 \times 1024$	A	250 µs	401	1.0	0.235	0.8	8	697	761.0 k	6.300 s	2.553 s	9.588 s	1.86 h
Intaglio	$768 \times 20 \times 768$	A	500 µs	601	1.0	0.314	0.8	8	1744	1.594 M	10.738 s	6.378 s	16.385 s	7.94 h
Rigid	–	B	6 ms	501	–	–	50	–	1500	2.916 k	104 ms	57 ms	419 ms	10.47 min

† All cases apply realistic physical values, with $\boldsymbol{g} = -9.8\,\text{m}\,\text{s}^{-2}\hat{\boldsymbol{e}}_y$, $\sigma = 7.28 \times 10^{-2}\,\text{N}\,\text{m}^{-1}$, $\rho_{\text{fluid}} = 1 \times 10^3\,\text{kg}\,\text{m}^{-3}$, and $\rho_{\text{iron}} = 7.8 \times 10^3\,\text{kg}\,\text{m}^{-3}$.

‡ The frame interval here indicates the simulated time interval per frame. The frame rate of the supplementary video is fixed at 30 FPS.

Jingyu Chen, Victoria Kala, Alan Marquez-Razon, Elias Gueidon, David A. B. Hyde, and Joseph Teran. 2021. A Momentum-Conserving Implicit Material Point Method for Surface Tension with Contact Angles and Spatial Gradients. *ACM Trans. Graph.* 40, 4, Article 111 (jul 2021), 16 pages.

Xuwen Chen, Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2022. Simulation and Optimization of Magnetoelastic Thin Shells. *ACM Trans. Graph.* 41, 4, Article 61 (jul 2022), 18 pages.

Yi-Lu Chen, Jonathan Meier, Barbara Solenthaler, and Vinicius C. Azevedo. 2020. An Extended Cut-Cell Method for Sub-Grid Liquids Tracking with Surface Tension. *ACM Trans. Graph.* 39, 6, Article 169 (nov 2020), 13 pages.

Nuttapong Chentanez and Matthias Müller. 2014. Mass-Conserving Eulerian Liquid Simulation. *IEEE Transactions on Visualization and Computer Graphics* 20, 1 (Jan 2014), 17–29.

Per H. Christensen. 2003. Adjoints and importance in rendering: an overview. *IEEE Transactions on Visualization and Computer Graphics* 9, 3 (2003), 329–340.

Per H. Christensen. 2008. Point-based approximate color bleeding. In *Pixar Technical Notes*. Pixar Animation Studios, online, 9 pages.

Michael F. Cohen, John Wallace, and Pat Hanrahan. 1993. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Inc., USA.

Erwin Coumans. 2015. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*. Association for Computing Machinery, New York, NY, USA.

Fang Da, Christopher Batty, and Eitan Grinspun. 2014. Multimaterial Mesh-Based Surface Tracking. *ACM Trans. Graph.* 33, 4, Article 112 (July 2014), 11 pages.

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double Bubbles sans Toil and Trouble: Discrete Circulation-Preserving Vortex Sheets for Soap Films and Foams. *ACM Trans. Graph.* 34, 4, Article 149 (July 2015), 9 pages.

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-Only Liquids. *ACM Trans. Graph.* 35, 4, Article 78 (jul 2016), 12 pages.

Denis Demidov. 2019. AMGCL: An Efficient, Flexible, and Extensible Algebraic Multigrid Implementation. *Lobachevskii Journal of Mathematics* 40, 5 (2019), 535–546.

Yitong Deng, Mengdi Wang, Xiangxin Kong, Shiying Xiong, Zangyueyang Xian, and Bo Zhu. 2022. A Moving Eulerian-Lagrangian Particle Method for Thin Film and Foam Simulation. *ACM Trans. Graph.* 41, 4, Article 154 (jul 2022), 17 pages.

Xinjian Fan, Xiaoguang Dong, Alp C. Karacakol, Hui Xie, and Metin Sitti. 2020. Reconfigurable multifunctional ferrofluid droplet robots. *Proceedings of the National Academy of Sciences* 117, 45 (2020), 27916–27926.

Xinjian Fan, Yihui Jiang, Mingtong Li, Yunfei Zhang, Chenyao Tian, Liyang Mao, Hui Xie, Lining Sun, Zhan Yang, and Metin Sitti. 2022. Scale-reconfigurable miniature ferrofluidic robots for negotiating sharply variable spaces. *Science Advances* 8, 37 (2022), eabq1677.

Florian Ferstl, Ryoichi Ando, Chris Wojtan, Rüdiger Westermann, and Nils Thuerey. 2016. Narrow Band FLIP for Liquid Simulations. *Computer Graphics Forum* 35, 2 (2016), 225–232.

Nick Foster and Ronald Fedkiw. 2001. Practical Animation of Liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 23–30.

Frederic Gibou, Ronald P. Fedkiw, Li-Tien Cheng, and Myungjoo Kang. 2002. A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains. *J. Comput. Phys.* 176, 1 (2002), 205–227.

Ryan Goldade, Christopher Batty, and Chris Wojtan. 2016. A Practical Method for High-Resolution Embedded Liquid Surfaces. *Computer Graphics Forum* 35, 2 (2016), 233–242.

David Hahn and Chris Wojtan. 2015. High-Resolution Brittle Fracture Simulation with Boundary Elements. *ACM Trans. Graph.* 34, 4, Article 151 (jul 2015), 12 pages.

David Hahn and Chris Wojtan. 2016. Fast Approximations for Boundary Element Based Brittle Fracture Simulation. *ACM Trans. Graph.* 35, 4, Article 104 (jul 2016), 11 pages.

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.

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

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

Libo Huang, Ziyin Qu, Xun Tan, Xinxin Zhang, Dominik L. Michels, and Chenfanfu Jiang. 2021. Ships, Splashes, and Waves on a Vast Ocean. *ACM Trans. Graph.* 40, 6, Article 203 (dec 2021), 15 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.

Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka. 2017. A Hyperbolic Geometric Flow for Evolving Films and Foams. *ACM Trans. Graph.* 36, 6, Article 199 (nov 2017), 11 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.

Doug L. James, Jernej Barbič, and Dinesh K. Pai. 2006. Precomputed Acoustic Transfer: Output-Sensitive, Accurate Sound Generation for Geometrically Complex Vibration Sources. *ACM Trans. Graph.* 25, 3 (jul 2006), 987–995.

Doug L. James and Dinesh K. Pai. 1999. ArtDefo: Accurate Real Time Deformable Objects. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 65–72.

Stefan Jeschke and Chris Wojtan. 2023. Generalizing Shallow Water Simulations with Dispersive Surface Waves. *ACM Trans. Graph.* 42, 4, Article 83 (jul 2023), 12 pages.

Guang-Shan Jiang and Danping Peng. 2000. Weighted ENO Schemes for Hamilton–Jacobi Equations. *SIAM Journal on Scientific Computing* 21, 6 (2000), 2126–2143.

James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (aug 1986), 143–150.

Byungkwon Kang, Yoojin Jang, and Insung Ihm. 2007. Animation of Chemically Reactive Fluids Using a Hybrid Simulation Method. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '07)*. Eurographics Association, Goslar, DEU, 199–208.

Myungjoo Kang, Ronald P. Fedkiw, and Xu-Dong Liu. 2000. A Boundary Condition Capturing Method for Multiphase Incompressible Flow. *Journal of Scientific Computing* 15, 3 (2000), 323–360.

Todd Keeler and Robert Bridson. 2015. Ocean Waves Animation Using Boundary Integral Equations and Explicit Mesh Tracking. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark) *(SCA '14)*. Eurographics Association, Goslar, DEU, 11–19.

Alexander Keller. 1997. Instant Radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 49–56.

Byungmoon Kim, Yingjie Liu, Ignacio Llamas, Xiangmin Jiao, and Jarek Rossignac. 2007. Simulation of Bubbles in Foam with the Volume Control Method. In *ACM SIGGRAPH 2007 Papers* (San Diego, California) *(SIGGRAPH '07)*. Association for Computing Machinery, New York, NY, USA, 98–es.

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

Seung-wook Kim and JungHyun Han. 2022. Fast Stabilization of Inducible Magnet Simulation. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 52, 8 pages.

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.

Sachiko Kodama. 2008. Dynamic Ferrofluid Sculpture: Organic Shape-Changing Art Forms. *Commun. ACM* 51, 6 (jun 2008), 79–81.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2022. A Survey on SPH Methods in Computer Graphics. *Computer Graphics Forum* 41, 2 (2022), 737–760.

Tassilo Kugelstadt, Andreas Longva, Nils Thuerey, and Jan Bender. 2021. Implicit Density Projection for Volume Conserving Liquids. *IEEE Transactions on Visualization and Computer Graphics* 27, 4 (2021), 2385–2395.

Egor Larionov, Christopher Batty, and Robert Bridson. 2017. Variational Stokes: A Unified Pressure-Viscosity Solver for Accurate Viscous Liquids. *ACM Trans. Graph.* 36, 4, Article 101 (jul 2017), 11 pages.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4, Article 49 (aug 2020), 20 pages.

Xingqiao Li, Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2023. GARM-LS: A Gradient-Augmented Reference-Map Method for Level-Set Fluid Simulation. *ACM Trans. Graph.* 42, 6, Article 192 (dec 2023), 20 pages.

Dennis A. Lindholm. 1980. Notes on boundary integral equations for three-dimensional magnetostatics. *IEEE Transactions on Magnetics* 16, 6 (1980), 1409–1413.

Xubo Liu, Noah Kent, Alejandro Ceballos, Robert Streubel, Yufeng Jiang, Yu Chai, Paul Y. Kim, Joe Forth, Frances Hellman, Shaowei Shi, Dong Wang, Brett A. Helms, Paul D. Ashby, Peter Fischer, and Thomas P. Russell. 2019. Reconfigurable ferromagnetic liquid droplets. *Science* 365, 6450 (2019), 264–267.

William E. Lorensen and Harvey E. Cline. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.* 21, 4 (aug 1987), 163–169.

Miles Macklin and Matthias Müller. 2013. Position Based Fluids. *ACM Trans. Graph.* 32, 4, Article 104 (jul 2013), 12 pages.

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Trans. Graph.* 42, 4, Article 82 (jul 2023), 11 pages.

Fumiya Narita and Ryoichi Ando. 2022. Tiled Characteristic Maps for Tracking Detailed Liquid Surfaces. *Computer Graphics Forum* 41, 8 (2022), 231–242.

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 (aug 2020), 15 pages.

Stanley Osher and Ronald P. Fedkiw. 2005. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, New York, NY, USA.

Jonathan Panuelos, Ryan Goldade, Eitan Grinspun, David Levin, and Christopher Batty. 2023. PolyStokes: A Polynomial Model Reduction Method for Viscous Fluid Simulation. *ACM Trans. Graph.* 42, 4, Article 124 (jul 2023), 13 pages.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2023. *Physically Based Rendering, fourth edition: From Theory to Implementation* (4th ed.). The MIT Press, Cambridge, MA, USA.

Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A bidirectional formulation for Walk on Spheres. *Computer Graphics Forum* 41, 4 (2022), 51–62.

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.

Bo Ren, Chenfeng Li, Xiao Yan, Ming C. Lin, Javier Bonet, and Shi-Min Hu. 2014. Multiple-Fluid SPH Simulation Using a Mixture Model. *ACM Trans. Graph.* 33, 5, Article 171 (sep 2014), 11 pages.

Ronald E. Rosensweig. 1997. *Ferrohydrodynamics.* Dover Publications, Inc., Mineola, NY, USA.

Karl K. Sabelfeld and Nikolai A. Simonov. 1994. *Random Walks on Boundary for Solving PDEs.* De Gruyter, Berlin, Boston.

Karl K. Sabelfeld and Nikolai A. Simonov. 2016. *Stochastic Methods for Boundary Value Problems: Numerics for High-dimensional PDEs and Applications.* De Gruyter, Berlin, Boston.

Stefan A. Sauter and Christoph Schwab. 2011. *Boundary Element Methods.* Springer Berlin Heidelberg, Berlin, Heidelberg, 183–287.

Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2023. Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions. *ACM Trans. Graph.* 42, 4, Article 80 (jul 2023), 20 pages.

Han Shao, Libo Huang, and Dominik L. Michels. 2023. A Current Loop Model for the Fast Simulation of Ferrofluids. *IEEE Transactions on Visualization and Computer Graphics* 29, 12 (2023), 5394–5405.

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.

Andrew Staniforth and Jean Côté. 1991. Semi-Lagrangian Integration Schemes for Atmospheric Models – A Review. *Monthly Weather Review* 119, 9 (1991), 2206–2223.

Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. 2022. Surface-Only Dynamic Deformables using a Boundary Element Method. *Computer Graphics Forum* 41, 8 (2022), 75–86.

Ryusuke Sugimoto, Terry Chen, Yiti Jiang, Christopher Batty, and Toshiya Hachisuka. 2023. A Practical Walk-on-Boundary Method for Boundary Value Problems. *ACM Trans. Graph.* 42, 4, Article 81 (jul 2023), 16 pages.

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 (dec 2021), 13 pages.

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

Mei Song Tong and Weng Cho Chew. 2020. *The Nyström Method in Electromagnetics.* John Wiley & Sons, Singapore.

Nobuyuki Umetani, Athina Panotopoulou, Ryan Schmidt, and Emily Whiting. 2016. Printone: Interactive Resonance Simulation for Free-Form Print-Wind Instrument Design. *ACM Trans. Graph.* 35, 6, Article 184 (dec 2016), 14 pages.

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional Surface Tension Flow Using Moving-Least-Squares Particles. *ACM Trans. Graph.* 39, 4, Article 42 (aug 2020), 16 pages.

Mengdi Wang, Yitong Deng, Xiangxin Kong, Aditya H. Prasad, Shiying Xiong, and Bo Zhu. 2021. Thin-Film Smoothed Particle Hydrodynamics Fluid. *ACM Trans. Graph.* 40, 4, Article 110 (jul 2021), 16 pages.

Chris Wojtan, Matthias Müller-Fischer, and Tyson Brochu. 2011. Liquid Simulation with Mesh-Based Surface Tracking. In *ACM SIGGRAPH 2011 Courses* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, Article 8, 84 pages.

Joel Wretborn, Sean Flynn, and Alexey Stomakhin. 2022. Guided Bubbles and Wet Foam for Realistic Whitewater Simulation. *ACM Trans. Graph.* 41, 4, Article 117 (jul 2022), 16 pages.

Jingrui Xing, Liangwang Ruan, Bin Wang, Bo Zhu, and Baoquan Chen. 2022. Position-Based Surface Tension Flow. *ACM Trans. Graph.* 41, 6, Article 244 (nov 2022), 12 pages.

Ali K. Yetisen, Ahmet F. Coskun, Grant England, Sangyeon Cho, Haider Butt, Jonty Hurwitz, Mathias Kolle, Ali Khademhosseini, A. John Hart, Albert Folch, and Seok Hyun Yun. 2016. Art on the Nanoscale and Beyond. *Advanced Materials* 28, 9 (2016), 1724–1742.

Jihun Yu and Greg Turk. 2013. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.* 32, 1, Article 5 (feb 2013), 12 pages.

Xiaoxuan Zhang, Lingyu Sun, Yunru Yu, and Yuanjin Zhao. 2019. Flexible Ferrofluids: Design and Applications. *Advanced Materials* 31, 51 (2019), 1903497.

Changxi Zheng and Doug L. James. 2009. Harmonic fluids. *ACM Trans. Graph.* 28, 3, Article 37 (jul 2009), 12 pages.

Changxi Zheng and Doug L. James. 2010. Rigid-Body Fracture Sound with Precomputed Soundbanks. *ACM Trans. Graph.* 29, 4, Article 69 (jul 2010), 13 pages.

Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015b. Codimensional Non-Newtonian Fluids. *ACM Trans. Graph.* 34, 4, Article 115 (July 2015), 9 pages.

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional Surface Tension Flow on Simplicial Complexes. *ACM Trans. Graph.* 33, 4, Article 111 (July 2014), 11 pages.

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

Yufeng Zhu, Robert Bridson, and Chen Greif. 2015a. Simulating Rigid Body Fracture with Surface Meshes. *ACM Trans. Graph.* 34, 4, Article 150 (jul 2015), 11 pages.

A MAGNETOSTATICS

Generally speaking, the evolution of the magnetic field is governed by Maxwell's equations as follows:

$$\begin{cases} \nabla \cdot \boldsymbol{B} = 0, & (34) \\ \nabla \times \boldsymbol{H} = \boldsymbol{j}_{\mathrm{f}} + \dfrac{\partial \boldsymbol{D}}{\partial t}, & (35) \end{cases}$$

where $\boldsymbol{B}(\boldsymbol{x})$ is the magnetic induction, $\boldsymbol{H}(\boldsymbol{x})$ is the magnetic field, $\boldsymbol{j}_{\mathrm{f}}(\boldsymbol{x})$ is the electric current density of free charges, and $\boldsymbol{D}(\boldsymbol{x}, t)$ is the electric displacement. Given the ferrofluid volume Ω and its surface $\partial\Omega$, the zero-current assumption of non-conductive ferrofluids states that $\boldsymbol{j}_{\mathrm{f}} = \boldsymbol{0}$ ($\boldsymbol{x} \in \Omega \cap \partial\Omega$) and $\partial \boldsymbol{D}/\partial t = \boldsymbol{0}$ ($\forall \boldsymbol{x}$). That is why we use the term *magnetostatics* instead of *magnetics*.

Considering that the $\boldsymbol{B}$-field is determined by the $\boldsymbol{H}$-field and the magnetization $\boldsymbol{M}(\boldsymbol{x})$ as $\boldsymbol{B} = \mu_0(\boldsymbol{H} + \boldsymbol{M})$, where μ_0 is the vacuum permeability, it is natural to expand (34–35) as

$$\begin{cases} \nabla \cdot \mu_0(\boldsymbol{H}_{\mathrm{app}} + \boldsymbol{H}_{\mathrm{ind}} + \boldsymbol{M}) = 0, & (36) \\ \nabla \times (\boldsymbol{H}_{\mathrm{app}} + \boldsymbol{H}_{\mathrm{ind}}) = \boldsymbol{j}_{\mathrm{f}}, & (37) \end{cases}$$

in which the total magnetic field $\boldsymbol{H}$ has been written as $\boldsymbol{H}_{\mathrm{app}} + \boldsymbol{H}_{\mathrm{ind}}$. Note that the applied magnetic field $\boldsymbol{H}_{\mathrm{app}}$ alone should hold Maxwell's equations as well in the concerned domain[1]:

$$\begin{cases} \nabla \cdot \mu_0 \boldsymbol{H}_{\mathrm{app}} = 0, & (38) \\ \nabla \times \boldsymbol{H}_{\mathrm{app}} = \boldsymbol{j}_{\mathrm{f}}, & (39) \end{cases}$$

and therefore we can obtain governing equations of $\boldsymbol{H}_{\mathrm{ind}}$ by subtracting (38–39) from (36–37):

$$\begin{cases} \nabla \cdot \mu_0(\boldsymbol{H}_{\mathrm{ind}} + \boldsymbol{M}) = 0, & (40) \\ \nabla \times \boldsymbol{H}_{\mathrm{ind}} = 0. & (41) \end{cases}$$

As described in §3.1, we define $\boldsymbol{H}_{\mathrm{ind}} = -\nabla\psi$, where $\psi(\boldsymbol{x})$ is a continuous scalar potential, to replace (41), such that Gauss's law for magnetism (40) can be reformulated as

$$\nabla^2\psi = \nabla \cdot \boldsymbol{M}. \tag{42}$$

Since the magnetization $\boldsymbol{M}(\boldsymbol{x})$ is determined by

$$\boldsymbol{M} = \begin{cases} \chi \boldsymbol{H}, & \boldsymbol{x} \in \Omega, & (43a) \\ \boldsymbol{0}, & \boldsymbol{x} \notin \Omega \cup \partial\Omega, & (43b) \end{cases}$$

we can further analyze (42) at different locations:

- $\boldsymbol{x} \notin \Omega \cup \partial\Omega$. It is clear to show $\nabla^2\psi = 0$ because of $\boldsymbol{M} = \boldsymbol{0}$.
- $\boldsymbol{x} \in \Omega$. The right-hand side of (42) is calculated by

$$\nabla \cdot \boldsymbol{M} = \chi(\nabla \cdot \boldsymbol{H}_{\mathrm{app}} - \nabla^2\psi) = -\chi\nabla^2\psi, \tag{44}$$

  such that $\nabla^2\psi = -\chi\nabla^2\psi$ holds true, which means $\nabla^2\psi = 0$.
- $\boldsymbol{x} \in \partial\Omega$. It is not hard to prove

$$\left.\frac{\partial\psi}{\partial n}\right|_+ - \left.\frac{\partial\psi}{\partial n}\right|_- = (\boldsymbol{M}_+ - \boldsymbol{M}_-) \cdot \boldsymbol{n} = -\boldsymbol{M}_- \cdot \boldsymbol{n} \tag{45}$$

  by applying Gauss's theorem on both sides of (42). Given that $\boldsymbol{M}_- \cdot \boldsymbol{n}$ can be reformulated by

$$\boldsymbol{M}_- \cdot \boldsymbol{n} = \chi \boldsymbol{H}_- \cdot \boldsymbol{n} = \chi \boldsymbol{H}_{\mathrm{app}} \cdot \boldsymbol{n} - \chi \left.\frac{\partial\psi}{\partial n}\right|_-, \tag{46}$$

  (45) is equivalent to the following formulation:

$$\left.\frac{\partial\psi}{\partial n}\right|_+ + \chi \boldsymbol{H}_{\mathrm{app}} \cdot \boldsymbol{n} = (1+\chi) \left.\frac{\partial\psi}{\partial n}\right|_-. \tag{47}$$

Thus we can conclude that the potential ψ is the solution to Laplace's equation $\nabla^2\psi = 0$ ($\boldsymbol{x} \notin \partial\Omega$), subject to the boundary conditions (47) ($\boldsymbol{x} \in \partial\Omega$) and $\lim_{\|\boldsymbol{x}\|\to\infty} \psi(\boldsymbol{x}) = 0$. The latter condition is conventional for potentials in an infinite domain.

[1]The free current density is only associated with external magnetic sources, according to the zero-current assumption of ferrofluids.

B THE POTENTIAL THEORY

B.1 The Single-Layer Potential

In addition to §4.1, here we show a concrete definition of the single-layer potential.

Different from that in §A, for $\boldsymbol{x} \in \partial\Omega$, we apply Gauss's theorem only on the right-hand side of (42) and acquire

$$\nabla^2\psi = -(\boldsymbol{M}_- \cdot \boldsymbol{n})\delta_{\partial\Omega}, \qquad \forall \boldsymbol{x}, \tag{48}$$

where $\delta_{\partial\Omega}(\boldsymbol{x})$ is the generalized Dirac delta function on $\partial\Omega$, which is a Poisson's equation, instead of Laplace's equation mentioned above. Considering that the solution to

$$\begin{cases} \nabla_x^2 G(\boldsymbol{x}, \boldsymbol{y}) = -\delta(\|\boldsymbol{x} - \boldsymbol{y}\|), & \forall \boldsymbol{x}, & (49) \\ G(\boldsymbol{x}, \boldsymbol{y}) \to 0, & \|\boldsymbol{x}\| \to \infty & (50) \end{cases}$$

is $G(\boldsymbol{x}, \boldsymbol{y}) = 1/(4\pi\|\boldsymbol{x} - \boldsymbol{y}\|)$, it is natural to define

$$\phi(\boldsymbol{x}) = \boldsymbol{M}_- \cdot \boldsymbol{n}, \qquad \boldsymbol{x} \in \partial\Omega, \tag{51}$$

such that the solution to (48) can be formulated as

$$\psi(\boldsymbol{x}) = \iint_{\partial\Omega} G(\boldsymbol{x}, \boldsymbol{y})\, \phi(\boldsymbol{y})\, \mathrm{d}A_y, \qquad \forall \boldsymbol{x}, \tag{52}$$

which is termed the *single-layer potential*.

Considering that the normal derivative of ψ is discontinuous across the interface:

$$\left.\frac{\partial\psi}{\partial n}\right|_+ - \left.\frac{\partial\psi}{\partial n}\right|_- = -\phi(\boldsymbol{x}), \qquad \boldsymbol{x} \in \partial\Omega, \tag{53}$$

by taking the normal derivative of (52), the symmetry of deciding the inner and outer sides results in

$$\left.\frac{\partial\psi}{\partial n}\right|_+ + \left.\frac{\partial\psi}{\partial n}\right|_- = 2\iint_{\partial\Omega} \frac{\partial G}{\partial n_x}(\boldsymbol{x}, \boldsymbol{y})\, \phi(\boldsymbol{y})\, \mathrm{d}A_y, \quad \boldsymbol{x} \in \partial\Omega, \tag{54}$$

and thus we can obtain

$$\left[\frac{\partial\psi}{\partial n}(\boldsymbol{x})\right]_\pm = \iint_{\partial\Omega} \frac{\partial G}{\partial n_x}(\boldsymbol{x}, \boldsymbol{y})\, \phi(\boldsymbol{y})\, \mathrm{d}A_y \mp \frac{1}{2}\phi(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega, \tag{55}$$

by combining (53) and (54). Substituting (47) (or (51)) into (55), it is easy to prove

$$\frac{\phi(\boldsymbol{x})}{2\alpha} = -\iint_{\partial\Omega} \frac{\partial G}{\partial n_x}(\boldsymbol{x}, \boldsymbol{y})\, \phi(\boldsymbol{y})\, \mathrm{d}A_y + \boldsymbol{H}_{\mathrm{app}}(\boldsymbol{x}) \cdot \boldsymbol{n}, \quad \boldsymbol{x} \in \partial\Omega, \tag{56}$$

where $\alpha = \chi/(2+\chi)$ is the reduced permeability.

B.2 The Double-Layer Potential

Besides the single-layer one, there is another potential named the *double-layer potential* for magnetostatics, which is adopted in the previous work of Huang and Michels [2020].

Suppose that the magnetic scalar potential of the applied field is known, denoted $\psi_{\mathrm{app}}(\boldsymbol{x})$ ($\boldsymbol{H}_{\mathrm{app}} = -\nabla\psi_{\mathrm{app}}$), a double-layer potential $\psi^*(\boldsymbol{x})$ is defined as

$$\psi^* = \begin{cases} (1+\chi)\mu_0\psi + \chi\mu_0\psi_{\mathrm{app}}, & \boldsymbol{x} \in \Omega\,, & (57a) \\ \mu_0\psi, & \boldsymbol{x} \notin \Omega \cup \partial\Omega. & (57b) \end{cases}$$

Considering that $-\nabla\psi^* = \mu_0(\boldsymbol{H}_{\mathrm{ind}} + \boldsymbol{M})$ both inside and outside the magnetic material, we use $\boldsymbol{B}_{\mathrm{ind}}$ to denote $-\nabla\psi^*$ because it has the same physical dimension as $\boldsymbol{B}$. Moreover, derived from Gauss's law for magnetism, $\boldsymbol{B}_{\text{ind}}$, as well as $\boldsymbol{B}$, is continuous along the normal direction across $\partial\Omega$, which means

$$\left.\frac{\partial\psi^*}{\partial n}\right|_+ = \left.\frac{\partial\psi^*}{\partial n}\right|_-, \qquad \boldsymbol{x} \in \Omega. \tag{58}$$

Since ψ^* is a potential without jumps of the normal derivative across the interface, we can define a density ϕ^* attached on the boundary [Sugimoto et al. 2023] such that

$$\psi^*(\boldsymbol{x}) = \iint_{\partial\Omega} \frac{\partial G}{\partial n_y}(\boldsymbol{x},\boldsymbol{y})\,\phi^*(\boldsymbol{y})\,\mathrm{d}A_y, \qquad \boldsymbol{x} \notin \partial\Omega. \tag{59}$$

By taking the limit, the above boundary integral equations suggests that the double-layer potential itself is not continuous:

$$\psi^*_\pm(\boldsymbol{x}) = \iint_{\partial\Omega} \frac{\partial G}{\partial n_y}(\boldsymbol{x},\boldsymbol{y})\,\phi^*(\boldsymbol{y})\,\mathrm{d}A_y \pm \frac{1}{2}\phi^*(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{60}$$

Substituting (57) into (60), the density ϕ^* is given by

$$\phi^* = -\chi\mu_0(\psi + \psi_{\text{app}}), \qquad \boldsymbol{x} \in \Omega, \tag{61}$$

and another equation is obtained as follows:

$$\frac{\phi^*(\boldsymbol{x})}{2\alpha} = -\iint_{\partial\Omega} \frac{\partial G}{\partial n_y}(\boldsymbol{x},\boldsymbol{y})\,\phi^*(\boldsymbol{y})\,\mathrm{d}A_y - \mu_0\psi_{\text{app}}. \tag{62}$$

Note that (62), similar to (56), can be utilized to determine ϕ^* on the boundary.