Surface-Only Ferrofluids

LIBO HUANG, KAUST
DOMINIK L. MICHELS, KAUST

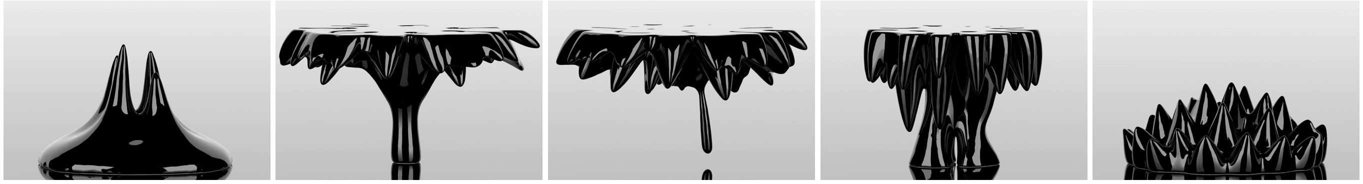


Fig. 1. Simulation of a magnetic fluid placed in between two electromagnets. Once the strength of the upper magnet is increased, the fluid previously placed on the ground is climbing up until it is fully attracted to the top. Then, the strength of the lower magnet is increased while the strength of the upper one is decreased getting the fluid attracted to the bottom.

We devise a novel surface-only approach for simulating the three dimensional free-surface flow of incompressible, inviscid, and linearly magnetizable ferrofluids. A Lagrangian velocity field is stored on a triangle mesh capturing the fluid's surface. The two key problems associated with the dynamic simulation of the fluid's interesting geometry are the magnetization process transitioning the fluid from a non-magnetic into a magnetic material, and the evaluation of magnetic forces. In this regard, our key observation is that for linearly incompressible ferrofluids, their magnetization and application of magnetic forces only require knowledge about the position of the fluids' boundary. Consequently, our approach employs a boundary element method solving the magnetization problem and evaluating the so-called magnetic pressure required for the force evaluation. The magnetic pressure is added to the Dirichlet boundary condition of a surface-only liquids solver carrying out the dynamical simulation. By only considering the fluid's surface in contrast to its whole volume, we end up with an efficient approach enabling more complex and realistic ferrofluids to be explored in the digital domain without compromising efficiency. Our approach allows for the use of physical parameters leading to accurate simulations as demonstrated in qualitative and quantitative evaluations.

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

Additional Key Words and Phrases: Boundary Element Method, Computational Electromagnetics, Eulerian Fluid Simulation, Ferrofluids, Fluid Mechanics, Galerkin Method, Helmholtz Decomposition, Large-scale Simulations, Magnetic Fluids, Maxwell's Equations, Natural Phenomena, Navier-Stokes Equations, Numerical Simulations, Surface-Only Fluids, Surface Tension, Validation Experiments.

ACM Reference Format:
Libo Huang and Dominik L. Michels. 2020. Surface-Only Ferrofluids. *ACM Trans. Graph.* 39, 6, Article 174 (December 2020), 17 pages. https://doi.org/10.1145/3414685.3417799

1 INTRODUCTION

Fluid simulations are one of the core topics of research within the computer graphics and visual computing community [Chern et al. 2016; Enright et al. 2002; Foster and Fedkiw 2001; Koschier et al. 2019; Larionov et al. 2017; Stam 1999; Weißmann and Pinkall 2010]. Researchers are attracted to this topic not only for its scientific value and because of several industrial application such as for example in the context of visual effects, but also simply for its own beauty.

In recent years, researchers started to devise new approaches and algorithms addressing fascinating phenomena related to fluids, for example for simulating water bells, droplets and jet collisions by just making use of the information on the fluid's surface [Da et al. 2016], by employing vortex sheets for simulating turbulent plumes [Pfaff et al. 2012] or soap films [Da et al. 2015], or even by employing variable thickness, viscous vortex filaments to simulate underwater bubble rings or "chandeliers" formed by ink dropping into liquid [Padilla et al. 2019].

In parallel to the development of new simulation methods, new physical models were devised progressively such as for surface tension [Akinci et al. 2013; Schroeder et al. 2012; Zheng et al. 2015], fluid viscosity [Bergou et al. 2010; Goldade et al. 2019; Larionov et al. 2017; Nagasawa et al. 2019], phase changes [Stomakhin et al. 2014], non-Newtonian fluids [Fang et al. 2019; Zhu et al. 2015b], and, e.g., the interaction of liquids with human hair [Fei et al. 2017].

Furthermore, recent work explores neural networks in order to represent details of fluids, for example in the context of temporal coherence [Xie et al. 2018], liquid splash modeling [Um et al. 2018], Lagrangian simulations [Ummenhofer et al. 2020], or even style-transfer [Kim et al. 2020].

Recently, the large-scale simulation of magnetic fluids was addressed within the SIGGRAPH community using smoothed-particle hydrodynamics (SPH) [Huang et al. 2019] and level-sets [Ni et al. 2020]. These so-called ferrofluids are colloidal liquids containing nanoscale magnetic particles that react to an external magnetic field without solidifying [Rosensweig 1988, 1997]. Ferrofluids have raised much interest because of their applications in medicine [Kafrouni and Savadogo 2016], energy harvesting [Alazemi et al. 2015], in the context of the optimization of loudspeakers [Rosensweig et al.

Authors' addresses: Libo Huang, KAUST, Visual Computing Center, Thuwal 23955, KSA; Dominik L. Michels, KAUST, Visual Computing Center, Thuwal 23955, KSA.

0730-0301/2020/12-ART174
https://doi.org/10.1145/3414685.3417799

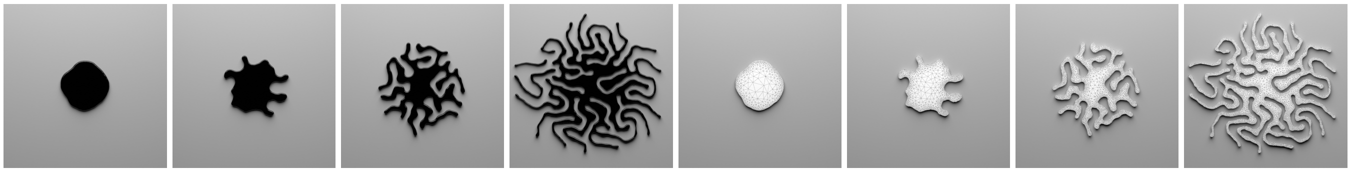


Fig. 2. The ferrofluid is trapped between two planes and exposed to an increasing magnetic field perpendicular to the planes. This lead to the formation of a characteristic labyrinth pattern [Dickstein et al. 1993; Rosensweig et al. 1983] as shown in this sequence (left: photorealistically rendered; right: mesh view).

2008] and adaptive deformable mirrors [Brousseau et al. 2007]. Moreover, as presented in SIGGRAPH's art gallery, the Japanese artist Sachiko Kodama [2008] employed ferrofluids to build organic shape-changing art forms. In contrast to regular (i.e. non-magnetic) fluids, ferrofluids seem to be a natural choice for building such structures since their dynamic behavior can easily be controlled by an external magnetic field as illustrated in Figure 1. Observing the use of ferrofluids in art is also not surprising from an aesthetics' perspective given their visually appealing appearance mainly caused by their interesting surface geometry including their characteristic spikes. While ferrofluids behave like regular Newtonian fluids in the absence of an external magnetic field, once a magnetic field is applied, the magnetic particles within the fluid arrange according to the field lines of the magnetic field so that the fluid starts forming these spikes. While a strong magnetic field strength increases the height of the spikes, the strength of the surface tension determines the wavelength (i.e. the distance between the individual spikes) and influences their smoothness.

The influence of magnetic effects is mostly reflected in the geometry of the free-surface for which reason the development of a surface-only approach [Da et al. 2016] for the simulation of ferrofluids is well motivated. Such an approach purely acts on the fluid's surface potentially saving computation time. In this contribution, we devise a novel surface-only approach for simulating the three dimensional free-surface flow of incompressible, inviscid, and linearly magnetizable ferrofluids. We incorporate the magnetic effects by only making use of the information available on the fluid's surface. For the phenomena we are interested in, we can assume that the movement of the fluid is slow compared to its response to the magnet, so that the fluid reaches its equilibrium within the external field instantly. Under such a quasi-static assumption, there are two key challenges we have to address. The first challenge is the magnetization problem: given a magnetic environment, how does the fluid react to the magnetic field? The second challenge is, once we know how the fluid is magnetized, how does this change its movement? To phrase it another rather way: how to apply the magnetic forces?

Most previous work [Cao and Ding 2014; Gollwitzer et al. 2007; Lavrova et al. 2006, 2008] use the finite element method (FEM) or couple a FEM with a boundary element method (BEM) to solve the magnetization problem. These approaches all require volume meshes which hinders the efficient simulation with dynamic topology changes. Particle methods [Huang et al. 2019; Ishikawa et al. 2012, 2013] do not require re-meshing, but still need to fill the whole volume with sample points. We observe that in the case of a linearly magnetizable fluid (which is a good approximation when the external magnetic field is weak), the BEM is sufficient to solve the magnetization problem. As a consequence, no volume discretization is required. Furthermore, we observe that, if the ferrofluid is incompressible, the magnetic forces can be applied by adding the gradient of a magnetic pressure field to the body force. Such a magnetic pressure field is consistent across a large number of theories in physics ranging from classic theories to their modern counterparts [Byrne 1977]. In addition, the magnetic pressure is similar to surface tension and can be applied by modifying the Dirichlet boundary condition at the fluid-air interface in the pressure-projection step of a fluid solver. Based on these insights, we are convinced that the surface mesh alone is sufficient to incorporate magnetic effects.

The surface-only liquid simulation assumes the fluid to be incompressible, thus divergence free, and inviscid, thus irrotational. Under these two assumptions the velocity of the interior of the fluid can be uniquely determined by the surface velocity field for a connected domain. Therefore, the information on the fluid's surface is sufficient to drive the whole simulation. The simulator leverages a surface tracking procedure that maintains a water tight triangle mesh of the liquid and employ the mesh for the dynamic simulation. We additionally solve the magnetization problem using a BEM based on the surface mesh, and calculate the magnetic pressure discontinuity on the surface. The magnetic pressure discontinuity at the fluid-air surface is added on top of the pressure discontinuity caused by surface tension. The two together are then fed into the surface-only fluid simulation to perform the simulation.

We aim for the development of a novel surface-only approach addressing the dynamical simulation of ferrofluids. In this regard, our main technical contribution are as follows.

- We incorporate magnetic phenomena within a surface-only fluid approach.
- We improve the Helmholtz decomposition step in surface-only fluid solvers by devising a more accurate analytic integration process.
- We improve the accuracy of the pressure projection step within surface-only fluid solvers by a Galerkin BEM.

Our method can accurately simulate the complex dynamics of ferrofluids such as shown in Figure 1, as well as the emergence of their characteristic patterns as demonstrated in Figure 2.

2 RELATED WORK

In this section, we provide an overview of related work. Our contribution primarily addresses magnetic phenomena for which reason we specifically focus on related work in this regard. In addition, we briefly discuss related fluid solvers and boundary element methods (BEM).

2.1 Simulation of Ferrofluids

In general, there are numerous ways to incorporate magnetic effects into fluid solvers depending on their specific types.

For Eulerian fluid solvers, recent work by Ni et al. [2020] employed level-sets to track the interface of the magnetic substance and solve the magnetic field problem in a Cartesian marker-and-cell (MAC) grid [Harlow and Welch 1965] by finite differences. They incorporated the magnetic effects by adding a magnetic pressure term to the Dirichlet boundary condition of the Poisson pressure solver within their fluid simulation framework. Their solver can successfully produce the dynamic motion of various magnetic substances including ferrofluids. However, they need to discretize both, the fluid and air voxels, leading to large number of unknowns. 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 adding the magnetic force as a Helmholtz body force. Zhu et al. [2011] employed a similar approach to simulate ferrofluid droplets, but used a level-set method instead.

In other contributions, researchers discretize the magnetic substance using particles. Based on this Lagrangian perspective, Huang et al. [2019] proposed to employ the concept of smoothed magnetic particles to sample the ferrofluid domain, and to calculate the magnetic forces on each of the magnetic particles. Their method handles the surface and body magnetic force uniformly, and does not require the explicit handling of the boundaries of the fluid domain. Although their method is augmented with fast multipole summations, the computation time is still proportional to the volume of the liquid. Prior to that, Ishikawa et al. [2012; 2013] used dipole magnets and smooth particle hydrodynamics (SPH) to simulation the motion of magnetic liquids. However, they found it necessary to augment it with a procedural modeling approach generating the spike pattern. Yoshikawa et al. [2010] presented another Lagrangian approach employing FEM to solve the magnetic part. However they were not able to create the spike pattern based on physical principles.

The Lagrangian view was also adopted in the context of magnetic rigid-body simulations. Thomaszewski et al. [2008] sampled the volume of a rigid body with magnetic dipoles and magnetized them only with the external field. Their approach is sufficient to capture some magnetic effects, however the forces between each magnetic dipole can be too large and crash a simulation. Later on, Kim et al. [2018] improved the mutual induction by generalizing microscopic magnetic equations to macroscopic scenarios. Both of these methods were designed for rigid bodies where the particles are always well separated. For fluid particles often closely located next to each other, these methods can potentially lead to instabilities.

The majority of earlier successes in the computational physics community with respect to simulating characteristic spike patterns of ferrofluids is limited to the equilibrium shape. Several authors [Boudouvis et al. 1987; Cao and Ding 2014; Gollwitzer et al. 2007; Lavrova et al. 2006, 2008] employed FEM, or coupled FEM-BEM techniques to solve the magnetic field problem. The FEM allows for simulating the nonlinear magnetization law, but it requires to apply a re-meshing procedure in every step of the dynamic simulation. Except these static simulations, Nochetto et al. [2016a; 2016b] developed a coupled Navier-Stokes and quasistatic Maxwell model, and solved it by using FEM in two dimensions.

The way we incorporate magnetic effects is mostly related to the work of Boudouvis et al. [1987], Cao and Ding [2014], Gollwitzer et al. [2007], and Lavrova et al. [2006; 2008]. They enforce the balance between surface tension, gravity potential and magnetic pressure discontinuity on the ferrofluid-air boundary and update the position of surface meshes according to the calculated magnetic pressure. The major difference between our method and theirs is that we use a simplified linear magnetization model, which allows us to use only the surface to calculate the magnetic field, and apply the magnetic pressure discontinuity. This is the key concept difference that leads to higher efficiency allowing for simulating larger problems with higher complexities.

2.2 Mesh-based Fluid Simulation

Since at least the seminal work of Stam [1999], the simulation of fluids is an established research focus within computer graphics and these days an ongoing topic within the whole visual computing community. For example, Bridson and Müller provided an introduction to fluid simulation from a computer graphics' perspective in their SIGGRAPH course [2007].

Since the magnetic part of our method requires a triangulated mesh representing the fluid's boundary, methods explicitly tracking [Müller 2009; Wojtan et al. 2011] the surface are mostly related. In general, plenty of surface tracking methods have been presented in the literature. Brochu and Bridson [2009] advect the explicit triangle meshes, and employ continuous collision detection to detect and resolve collisions. Topology changes and mesh improvements are supported. This method was improved by Da et al. [2014] to support multi-material tracking. Wojtan et al. [2009] make use of a signed distance field to assist topology changes while maintaining a Lagrangian mesh. Chentanez et al. [2015] achieved a balance between speed and accuracy by tracking manifold triangle meshes, deleting intersecting triangles, and filling the holes. Li et al. [2016] track multiple materials using signed distance and indicator functions, and reconstruct the meshes between materials at every frame. Bojsen-Hansen et al. [2013] addressed the problem of tracking high resolution surface meshes with low resolution grids, reducing artifacts in the meshes by smoothing or augmented dynamics on the surfaces.

Aside from methods dedicated to tracking the surface, several other work address coupled fluid simulation with surface meshes or interior meshes. Chentanez et al. [2007] first generated a surface of the liquid, then filled the inside with lattice-based tetrahedral meshes. Thurey et al. [2010] devised a multi-scale approach to simulate surface tension effects by decomposing surface tension into a grid part employing Eulerian grids and a sub-grid part employing Lagrangian surface meshes. Zhang et al. [2012] used a fast deformable surface to approximate the motion of droplets and their merging behavior. Ando et al. [2013] developed a highly adaptive tetrahedral mesh approach solving the pressure projection problem in fluid simulation. Clausen et al. [2013] proposed to use tetrahedral Lagrangian meshes to simulate liquids and solids in a unified way. In order to

ALGORITHM 1: Numerical integration procedure updating the state (i.e. vertex positions and velocities) of the magnetic fluid.

```
Input: Current fluid state.
Output: Updated fluid state.
Advection; Section 4.1.
Enforce harmonic velocity by Helmholtz decomposition; Section 4.2.
Calculate surface tension and gravity potential. Section 4.3.
Calculate magnetic pressure using Algorithm 2.
Solve the pressure using a BEM; Section 4.5.
Apply the negative gradient of the pressure to the velocity.
```

ALGORITHM 2: Evaluation procedure of the magnetic pressure.

```
Input: Vertex positions.
Output: Magnetic pressure at vertices.
Update the external magnetic field.
Evaluate the external magnetic scalar potential; Section 4.4.1.
Calculate magnetic double layer charges at vertices; Section 4.4.2.
Evaluate magnetic pressure discontinuities at vertices; Section 4.4.4.
```

take into account the fluid dynamics in different codimensional spaces, Zhu et al. [2014] introduced free-surface flow on simplicial complexes. Zheng et al. [2015] used adaptive tetrahedral meshes to simulate the liquid with an implicit solver accounting for surface tension effects. Da et al. [2016] simulated the fluid dominated by inertia and surface tension using only the Lagrangian surface mesh.

2.3 Boundary Integral

Solving the magnetic field problem in our approach relies heavily on boundary integrals. Boundary integrals have been used extensively in computer graphics; e.g. for soft body simulation [James and Pai 1999], ocean waves [Keeler and Bridson 2014], fractures [Hahn and Wojtan 2015, 2016; Zhu et al. 2015a], and mesh generation [Solomon et al. 2017; Wang et al. 2013]. Boundary integral and boundary element methods only require the information at the surface of the object saving the effort to discretize and handle the interior of the object or even the whole domain.

3 OVERVIEW

We devise a novel surface-only approach for simulating the three dimensional free-surface flow of incompressible, inviscid, and linearly magnetizable ferrofluids. In this section, we provide an overview of the approach by first explaining the general structure of a surface-only fluid simulator and then specifically addressing the main steps of the magnetic part. The simulation framework is summarized in Algorithm 1, and specifically the magnetic part in Algorithm 2.

The surface-only fluid concept [Da et al. 2016] is dedicated to fluid scenes dominated by surface tension and inertia such as fluid droplets and water-jets where viscosity effects are negligible. Assuming inviscid liquids, Da et al. [Da et al. 2016] concluded that the velocity field enclosed by the liquid boundary is divergence-free and curl-free (denoted by them as a harmonic velocity field). Therefore the velocity in the interior of the liquid is uniquely determined by the velocity of the boundary mesh vertices (for a contractible domain). The surface mesh is driven by such a harmonic velocity field, and the external forces (such as surface tension and gravity) apply harmonic updates to the enclosed velocity field. The surface-only fluid solver mainly comprises three main steps: advection, Helmholtz decomposition, and solving the pressure based on a BEM. In the advection step, vertices move to new positions based on their velocities. After the advection step, because the positions of vertices have changed while their velocities are remained, the new velocity field is no longer harmonic. Then a Helmholtz decomposition based on a boundary integral is applied to correct the normal component of the velocity. In this regard, the tangential velocities on the surface remain unchanged. This step ensures that the velocity field is incompressible. In the last step, the effects of surface tension, gravity, and solid impact are added. The pressure discontinuities caused by surface tension are added as Dirichlet boundary conditions, and solid impacts are incorporated as Neumann boundary conditions on the solid surface. A collocation BEM is used to solve the mixed boundary value problem. Finally the negative gradient of the solved pressure field is added to the velocities.

In Da et al. [2016], the only body force (gravity) is directly added to each vertex. We realize that there is another way to incorporate the body force, although the liquid is only represented by a closed surface. Please note, that in the last step, the effects of all forces are incorporated by a gradient of a harmonic scalar potential field, which guarantees that the velocity update results in a harmonic vector field. The gravity update is actually a constant vector field pointing downwards. Such a constant vector field is the negative gradient of the gravity potential $\varphi_g = -\rho \boldsymbol{g} \cdot \boldsymbol{x}$, where $\boldsymbol{g}$ is the gravity acceleration vector and $\boldsymbol{x}$ is the position vector. Hence, to apply the gravity body force, we simply need to add the gravity potential φ_g to the Dirichlet boundary conditions.

This view can be generalized to other body forces including magnetic body forces. Just like the gravity potential is given by the integral of the gravity body force along a path from a reference level, the magnetic pressure is also given by the integral of the magnetic force along an arbitrary path from a reference point. Note that the magnetic force consists of both, body and surface forces. Hence, the integral path needs to include the surface as well. Figure 3 illustrates the definition of the magnetic pressure discontinuity p_{mag} at the surface. As pointed out by Byrne [1977], there are numerous classic and modern theories on magnetic body forces and corresponding surface forces. They have different forms of magnetic body pressure and magnetic surface pressure discontinuity. However, when these two terms are joint together, all these theories are consistent for incompressible ferrofluids. This definition unifies gravity potential where there is only the body part, and surface tension where there is only the surface part.

There are two key problems to calculate the magnetic pressure discontinuity at the surface. The first one is a magnetization problem and the second one is to evaluate the magnetic pressure discontinuity based on the calculated magnetic field after magnetization.

In the magnetization process, ferrofluids interact with external magnetic fields. We take a quasi-static approach assuming the magnetic effect reaches the equilibrium instantly within a single simulation frame. We further assume the external field is weak, so that the material responses to the magnetic field linearly. We solve the

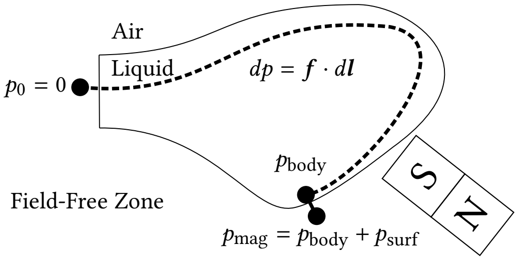

Air
Liquid
$dp = \boldsymbol{f} \cdot d\boldsymbol{l}$
$p_0 = 0$
p_{body}
Field-Free Zone
$p_{\text{mag}} = p_{\text{body}} + p_{\text{surf}}$
S
N

Fig. 3. Illustration of the magnetic pressure adapted from Byrne [1977]. From a reference point where pressure p_0 is zero, we integrate the magnetic body force $\boldsymbol{f} \cdot d\boldsymbol{l}$ along a path inside the ferrofluid to the interior limit of the surface to get the body part of the magnetic pressure. Because the magnetic stress tensor is discontinuous across the surface, we need to add a surface part to obtain the final magnetic pressure discontinuity.

magnetization problem by the Galerkin BEM with the indirect double layer potential formulation presented by Andjelic et al. [2011]. It consists of three steps: evaluating the magnetic scalar potential of the external magnetic field, solving the double-layer charges on the surface, and evaluating the magnetic field with the solved double layer charges. The details will be explained in Section 4.4.

After we solve the magnetization problem, we know the magnetic field. Based on that we can calculate the magnetic pressure discontinuity. The magnetic pressure discontinuity is reviewed by Byrne [1977] and already used in literature [Boudouvis et al. 1987; Cao and Ding 2014; Gollwitzer et al. 2007; Lavrova et al. 2006, 2008]. The magnetic pressure is evaluated at the surface and only require quantities on the surface although it includes a line integral through the body.

To summarize, our surface-only ferrofluid simulator consists of two parts: the surface-only fluid simulator and the magnetic solver which takes the mesh geometry and returns the magnetic pressure discontinuities on the surface. The magnetic solver consists of a BEM magnetostatic field solver [Andjelic et al. 2011], and the second part computing the magnetic pressure discontinuity is based on well-established theory [Byrne 1977].

4 METHODOLOGY

After the high-level overview of our surface-only approach for simulating ferrofluids presented in the previous section, this section provides a detailed description of our methodology. The subsections are arranged according to Algorithm 1. In Figure 8, we visualize the main quantities within the computation process. For an introduction to the Galerkin BEM we refer to our supplemental material and relevant literature [Rjasanow and Steinbach 2007; Sauter and Schwab 2010; Steinbach 2007].

4.1 Advection Procedure

In the advection step, each vertex moves to its new position using its current velocity. Collisions of triangles are resolved by surface tracking also handling topology changes and mesh improvements [Da et al. 2014]. Vertex velocities are not changed in the advection step.

4.2 Helmholtz Decomposition

After the advection procedure, one can calculate the total divergence and curl of the velocity field in the liquid domain based on the velocity on its boundary. Generally, the velocity field is no longer harmonic (i.e. divergence-free and curl-free). In order to enforce a harmonic velocity field, a Helmholtz decomposition is applied decomposing a vector field into two vector fields which are divergence-free and curl-free respectively.

4.2.1 Helmholtz Decomposition in Continuous Form. Da et al. [2016] proposed a constructive Helmholtz decomposition only keeping the harmonic velocity. Assume after the advection, the fluid occupies the domain Ω, whose boundary is $\Gamma := \partial\Omega$, and the fluid velocity at the boundary is $\boldsymbol{u}$. First, a scalar field ϕ and a vector field $\boldsymbol{A}$ that only rely on the velocities $\boldsymbol{u}$ on the boundary Γ are defined:

$$\begin{aligned} \phi(\boldsymbol{x}) &= -\int_\Gamma \boldsymbol{n}(\boldsymbol{y}) \cdot \boldsymbol{u}(\boldsymbol{y}) \frac{1}{4\pi} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} ds_y\,, &(1)\\ \boldsymbol{A}(\boldsymbol{x}) &= -\int_\Gamma \boldsymbol{n}(\boldsymbol{y}) \times \boldsymbol{u}(\boldsymbol{y}) \frac{1}{4\pi} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} ds_y\,, &(2)\end{aligned}$$

where $\boldsymbol{n}$ is the outward normal on the surface. The new divergence-free and curl-free velocity field $\bar{\boldsymbol{u}}$ is constructed as follows:

$$\begin{aligned} \bar{\boldsymbol{u}} &= -\nabla\phi + \nabla \times \boldsymbol{A} \\ &= \frac{1}{4\pi}\int_\Gamma \boldsymbol{n}(\boldsymbol{y}) \cdot \boldsymbol{u}(\boldsymbol{y}) \nabla_{\boldsymbol{x}} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} ds_y \\ &\quad -\nabla_{\boldsymbol{x}} \times \frac{1}{4\pi}\int_\Gamma \boldsymbol{n}(\boldsymbol{y}) \times \boldsymbol{u}(\boldsymbol{y}) \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} ds_y\,. \qquad (3)\end{aligned}$$

Da et al. [2016] proposed to keep the tangential component of the original velocity field $\boldsymbol{u}$ and only update the normal component of the velocity to that of the constructed velocity $\bar{\boldsymbol{u}}$. By keeping the tangential velocity field, no constrains on the curl of the velocity are applied. In reality, many of the phenomena are closely related to vortices. Therefore only correcting the normal component could create more phenomena that relies on vortices. However, numerical noise accumulated in the tangential velocity field is not corrected leading to more noisy simulation as shown in Section 5. Therefore we decide to use the full constructed velocity $\bar{\boldsymbol{u}}$ which satisfies the assumption of the surface-only fluid simulator.

To facilitate the computation, it is helpful to modify the curl term:

$$\begin{aligned} \nabla \times \boldsymbol{A} &= -\nabla_{\boldsymbol{x}} \times \frac{1}{4\pi}\int_\Gamma \boldsymbol{n}(\boldsymbol{y}) \times \boldsymbol{u}(\boldsymbol{y}) \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} ds_y \\ &= \frac{1}{4\pi}\int_\Gamma (\boldsymbol{n}(\boldsymbol{y}) \times \boldsymbol{u}(\boldsymbol{y})) \times \nabla_{\boldsymbol{x}} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} ds_y\,. \qquad (4)\end{aligned}$$

As pointed out by Da et al. [2016], the gradient part Eq. (3) and the curl part Eq. (4) are difficult to evaluate at sharp corners (vertices) because the gradient of the Green's function $\nabla_{\boldsymbol{x}}(4\pi\|\boldsymbol{x}-\boldsymbol{y}\|)^{-1}$ is strongly singular. It is easier to evaluate them on flat triangles. We first calculate the constructed velocity $\bar{\boldsymbol{u}}$ in Eq. (3) at the interior limit of symmetric quadrature points on each triangle. Then we convert the quantities on faces to quantities defined on vertices using the following approach.

4.2.2 Face Quadrature to Vertex Process. Assume we have a scalar function f defined on the boundary Γ, our goal is to find its discrete version $f_h = \sum f_{h,i}\psi_i^1$. Here, ψ_i^1 is the continuous piece-wise linear function associated with vertex i (see Figure 4, left) defined on the surface; $f_{h,i}$ is the coefficient of the discrete approximation f_h. The discrete approximation f_h equals the continuous function in a weak sense:

$$\langle \psi_i^1, \sum_{j=1}^{N_v} f_{h,j}\psi_j^1 \rangle_\Gamma \quad = \quad \langle \psi_i^1, f \rangle_\Gamma \, , \tag{5}$$

where $\langle \cdot, \cdot \rangle_\Gamma$ denotes the inner product defined on the mesh boundary Γ, N_v is the number of vertices, and ψ_i^1 is the linear basis function defined on vertex i. The above equation is set up for each vertex leading to N_v equations. In a more compact form, we can write

$$\widehat{M}_h[m,n] \quad = \quad \langle \psi_m^1, \psi_n^1 \rangle_\Gamma \, , \tag{6}$$

$$\widehat{M}_h \underline{f_h} \quad = \quad \langle \psi_i^1, f \rangle_\Gamma \, , \tag{7}$$

where $\underline{f_h}$ denotes the vector formed by the coefficients $f_{h,i}$, $\widehat{M}_h$ is a sparse matrix denoted as the mass matrix, and $\widehat{M}_h[m,n]$ refers to its entry at row m and column n.

The right hand side of Eq. (7) can be calculated by a weighted summation of the function f at quadrature points on each triangle. In order to get the sharp value $\underline{f_h}$, we need to solve Eq. (7). We could also approximate the mass matrix in Eq. (6) by summing all entries in a row or column to the diagonal entry. Each diagonal entry becomes the area for each vertex:

$$\widetilde{M}_h[m,m] = \sum_n \langle \psi_m^1, \psi_n^1 \rangle_\Gamma = \langle \psi_m^1, 1 \rangle_\Gamma = \mathsf{VertexArea}(m) \, . \tag{8}$$

If we use the approximated mass matrix $\widetilde{M}_h$ to solve Eq. (7), we obtain the smoothed version of $\underline{f_h}$:

$$\underline{\tilde{f_h}} = \widetilde{M}_h^{-1} \langle \psi_i^1, f \rangle_\Gamma \, . \tag{9}$$

The amount of smoothing can be controlled by a linear combination of sharp $\underline{f_h}$ and smooth $\underline{\tilde{f_h}}$:

$$\underline{f_{h,\text{final}}} = (1-\eta)\underline{f_h} + \eta \underline{\tilde{f_h}} \, , \tag{10}$$

where $\underline{f_{h,\text{final}}}$ is the final value on each vertex, and $\eta \in [0,1]$ is the artificial smoothing coefficient. The smoothed $\underline{\tilde{f_h}}$ can be considered as the diffused sharp $\underline{f_h}$ on the surface manifold. The smoothed $\underline{\tilde{f_h}}$ introduces errors, so it is only incorporated in the Helmholtz decomposition to achieve damping. In other cases, we use the sharp $\underline{f_h}$ for accuracy. For vector-value functions, we can apply this face quadrature to vertex conversion for each component.

4.2.3 Integration over the Triangle Surface Mesh. We evaluate Eq. (3) and (4) for $\boldsymbol{x}$ at the quadrature points on each triangle. For each quadrature point, we need to integrate over the whole boundary Γ, where the velocity is linearly interpolated on triangles. Hence the integral can be applied as a summation over all triangles. Both integrals (3) and (4) rely on a common part which is the gradient of a single layer potential with linearly varying charges at vertex i:

$$\frac{1}{4\pi} \int_\Gamma \psi_i^1(\boldsymbol{y}) \nabla_{\boldsymbol{x}} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} ds_y \, . \tag{11}$$

Such an integral over the boundary Γ can be decomposed to an integral over the triangles $T(i)$ adjacent to vertex i. The term (11) has an analytical solution if $\boldsymbol{x}$ is not on the triangle (see Eq. (3) and Eq. (9) in Eibert and Hansen [1995], and Appendix A). If the quadrature point $\boldsymbol{x}$ is located outside the integral triangle, we directly use the analytical solution. If the quadrature point is located on the integral triangle, we need a special handling. The gradient of the single layer potential (Eq. (11)) can be decomposed into the triangle normal component and the triangle tangential component (Eq. (11) and (12) in Eibert and Hansen [1995]). The triangle normal component of Eq. (11) is discontinuous across the triangle, and must be evaluated from the interior side of the triangle (inside the fluid domain) taking the limit to the quadrature point on the triangle. The tangential component is continuous on both sides of the triangle, so we can evaluate it exactly on the quadrature point. Part of the tangential term can be combined with the face quadrature to vertex procedure (Eq. (7)) leading to a double integral over the same triangle, and can benefit from analytical results [Sievers et al. 2005]. The other part of the tangential component is still evaluated on quadrature points.

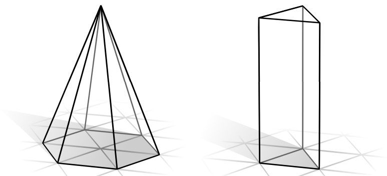


Fig. 4. Continuous piece-wise linear basis ψ^1 defined on vertices (left) and discontinuous piece-wise constant basis ψ^0 defined on triangles (right). The superscript (1 and 0) denotes the maximal order of the polynomials. The grid on the floor indicates the triangle meshes, the vertical direction indicates the strength of the basis function.

Finally, we convert the evaluated integral at quadrature points to the vertex values by the previous face quadrature to vertex process for each component of the constructed velocity $\overline{\boldsymbol{u}}$ and blend it with the smoothed version (Eq. (10)). We denote this by smooth damping. Alternatively, we can multiply the sharpened velocity uniformly by a scalar called the vacuum damping constant since the motion is damped as if it is slowed down by the vacuum. The smoothed velocity is not strictly harmonic because it is the low-pass-filtered sharp, harmonic velocity. It introduces minor volume errors in the simulation. About 50% of the smoothed velocity leads to visible volume changes over hundreds of frames, while about 10% smoothed velocity usually does not matter much.

We compare the partial Helmholtz decomposition (only update normal velocity) using the Duffy transform quadrature integration [Da et al. 2016], partial Helmholtz decomposition using analytical integration, and full Helmholtz decomposition using analytical integration. This is illustrated in the context of the simulation of two colliding fluid jets as shown in Figure 5. The partial decomposition with quadrature rules (top) is noisy (left) unless we use smooth damping (middle column). Using the analytical integration (middle) the partial Helmholtz decomposition behaves well. Full Helmholtz decomposition (bottom) is the most stable one but sacrifices some velocity modes. Vacuum damping (right column) leads to significantly different dynamic results.

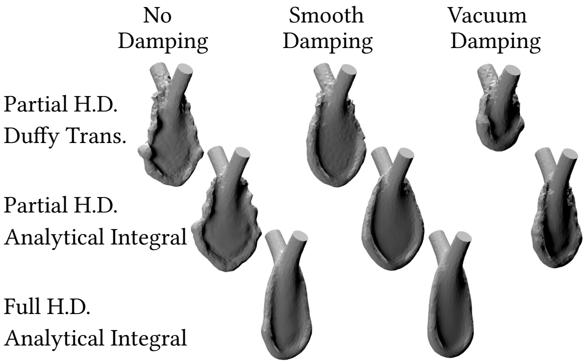

No Damping
Smooth Damping
Vacuum Damping
Partial H.D. Duffy Trans.
Partial H.D. Analytical Integral
Full H.D. Analytical Integral

Fig. 5. Simulation of two colliding fluid jets using different Helmholtz decomposition techniques. Top row: partial Helmholtz decomposition that only updates the normal component of the velocity using the Duffy transform [1982] adopted by Da et al. [2016]. Middle row: partial Helmholtz decomposition using the analytical integration of the Green's function over triangles. Bottom row: full Helmholtz decomposition using the analytical integration of the Green's function over triangles. Left column: no damping. Middle column: smooth damping with 20% smooth velocity and 80% sharp velocity. Right column: vacuum damping with sharp velocity scaled by a factor of 0.995. The partial Helmholtz decomposition with quadrature integration is noisy unless we use smooth damping. Using the analytical integration the partial Helmholtz decomposition behaves well. Full Helmholtz decomposition is the most stable one but sacrifices some velocity modes.

4.3 Surface Tension and Gravity Potential

The surface tension [Cohen-Steiner and Morvan 2003] is evaluated per vertex as done by Da et al. [2016]. The gravity potential is given by $\phi_g = -\rho \boldsymbol{g} \cdot \boldsymbol{x}$ where $\boldsymbol{x}$ is the vertex position, ρ is the fluid density, and $\boldsymbol{g} = (0, 0, -g)^\top$ denotes the gravity acceleration. This is illustrated in Figure 8.

4.4 Magnetic Pressure Calculation

To calculate the magnetic pressure, we first need to solve the magnetization problem, where the total magnetic field is different from the external field due to the interaction of the magnetized ferrofluids. Then based on the total magnetic field, we can determine the magnetic pressure discontinuity on the fluid boundary, and add it to the boundary conditions of the pressure solver.

To solve the magnetization problem, we follow the method of Andjelic et al. [2011], who compared Galerkin boundary element methods with three different formulations with respect to precision and efficiency. We choose the double layer potential formulation to achieve the balance.

In the double layer potential formulation, the total magnetic field after the ferrofluid has been magnetized is the linear combination of the given external magnetic field vector $\boldsymbol{H}_{\text{ext}}$ and the gradient of a double layer scalar potential field $\nabla W v$ generated by some unknown double layer charges v on the surface of the object. The linear integral operator W is the double layer potential operator (see supplemental material, and Rjasanow and Steinbach [2007], page 4). The operator W transforms scalar-value double layer charges v defined on the boundary into a scalar-value potential field defined in space. Then a boundary integral equation about v (see Andjelic et al. [2011], Eq. (3.19)) is set up to ensure the transmission property of the magnetic field across the object boundary:

$$\frac{1}{2}\frac{\mu_1+\mu_0}{\mu_1-\mu_0}v(\boldsymbol{x}) + (Kv)(\boldsymbol{x}) = -\mu_0\phi_{\text{ext}}(\boldsymbol{x}), \quad \text{for } \boldsymbol{x}\in\Gamma, \tag{12}$$

$$(Kv)(\boldsymbol{x}) = \frac{1}{4\pi}\lim_{\varepsilon\to 0}\int_{\boldsymbol{y}\in\Gamma:\|\boldsymbol{x}-\boldsymbol{y}\|>\varepsilon} \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{y})}{\|\boldsymbol{x}-\boldsymbol{y}\|^3}v(\boldsymbol{y})ds_{\boldsymbol{y}}, \tag{13}$$

where K is the double layer operator (see supplemental material; Rjasanow and Steinbach [2007], page 5), ϕ_{ext} is the external magnetic scalar potential (see Section 4.4.1), $\mu_0 = 4\pi\cdot 10^{-7}$ N/A^2 is the vacuum permeability, $\mu_1 = (1+\chi)\mu_0$ is the permeability inside the ferrofluid, and χ is the volume magnetic susceptibility of the ferrofluid. A larger χ means it is more likely to be magnetized.

The unknown double layer charges v are approximated by the discrete version $v_h = \sum v_{h,i}\psi_i^1$, where ψ_i^1 is the linear basis function associated with vertex i (see Figure 4, left). For the discrete unknown $v_{h,i}$, the Galerkin BEM is used to transform the continuous Eq. (12) into the following linear algebraic equation (see Andjelic et al. [2011], Section 4.2):

$$\left(\frac{1}{2}\frac{\mu_1+\mu_0}{\mu_1-\mu_0}\widehat{M}_h + \widehat{K}_h\right)\underline{v_h} = -\mu_0\widehat{M}_h\underline{\phi_{\text{ext}}}, \tag{14}$$

in which $\underline{v_h}$ denotes the vector of the unknown coefficients $v_{h,i}$ of double layer charges on vertices i. Eq. (14) can be reduced to

$$\left(\frac{1}{2}\frac{2+\chi}{\chi}\widehat{M}_h + \widehat{K}_h\right)\underline{v_h} = -\mu_0\widehat{M}_h\underline{\phi_{\text{ext}}}, \tag{15}$$

in which $\underline{\phi_{\text{ext}}}$ is the magnetic scalar potential on each vertex position. The calculation is addressed in Section 4.4.1. The meaning of the matrices $\widehat{M}_h$ and $\widehat{K}_h$, and how to efficiently solve Eq. (15) will be explained in Section 4.4.2.

4.4.1 Evaluation of the External Magnetic Scalar Potential at Vertices. We express the external magnetic field as a combination of a constant vector field and the field generated by a cluster of magnetic dipole sources. The magnetic field $\boldsymbol{H}_{\text{ext}}$, and corresponding magnetic scalar potential ϕ_{ext} can be represented as follows:

$$\begin{aligned} \boldsymbol{H}_{\text{ext}}(\boldsymbol{x}) &= \sum_{i=1}^{N}\frac{1}{4\pi}\frac{3(\boldsymbol{m}_i\cdot\hat{\boldsymbol{r}}_i)\hat{\boldsymbol{r}}_i - \boldsymbol{m}_i}{r_i^3}, && (16)\\ \phi_{\text{ext}}(\boldsymbol{x}) &= -\sum_{i=1}^{N}\frac{1}{4\pi}\frac{\boldsymbol{r}_i\cdot\boldsymbol{m}_i}{r_i^3}, && (17)\end{aligned}$$

where N is the number of dipole sources, $\boldsymbol{r}_i = \boldsymbol{x} - \boldsymbol{y}_i$ is the vector pointing from the source location $\boldsymbol{y}_i$ to the field point $\boldsymbol{x}$, $r_i = \|\boldsymbol{r}_i\|$ is the distance, $\hat{\boldsymbol{r}}_i = \boldsymbol{r}_i/r_i$ is the normalized version, and $\boldsymbol{m}_i$ is the dipole moment for the source indexed by i. The magnetic scalar potential ϕ_{ext} is visualized in Figure 8.

4.4.2 Calculation of the Magnetic Double Layer Charges at Vertices. Assume, we have N_{v} vertices, then the dimension of the unknown double layer charges $\underline{v_h}$ is N_{v}. The matrix $\widehat{M}_h$ is the mass matrix defined in Eq. (6). The $\widehat{K}_h$ is another N_{v}-dimensional square matrix. Let m be the row index and let n be the column index, the values of the entries are

$$\widehat{K}_h[m,n] \quad = \quad \langle\psi_m^1, K\psi_n^1\rangle_\Gamma\,. \tag{18}$$

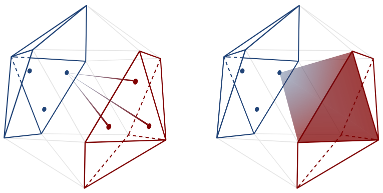


Fig. 6. Double integral over the support of the source (trial) function (red triangles in the lower right), and the support of target (test) function (blue triangles in the upper left), see e.g. Eq. (20). The outer integral (over test function) uses a symmetric three-point quadrature rule. The quadrature points are marked as blue dots. For each target quadrature point, the inner integral is evaluated using the three-point quadrature rule illustrated by three red dots on the left, and using analytic integration over the whole source triangle illustrated by the red triangle on the right.

As a concrete example, if we denote all the triangles adjacent to the vertex m by $T(m)$, and all the triangles adjacent to the vertex n by $T(n)$, we expand the above expression as follows:

$$\widehat{M}_h[m,n] = \sum_{p\in T(m)} \int_i \psi_n^1(\boldsymbol{y})\psi_m^1(\boldsymbol{y})ds_y\,, \tag{19}$$

$$\widehat{K}_h[m,n] = \sum_{q\neq p} \int_{p\in T(m)} \psi_m^1(\boldsymbol{x}) \int_{q\in T(n)} \frac{\boldsymbol{n}(\boldsymbol{y})\cdot(\boldsymbol{x}-\boldsymbol{y})\psi_n^1(\boldsymbol{y})}{4\pi\|\boldsymbol{x}-\boldsymbol{y}\|^3} ds_y ds_x\,. \tag{20}$$

The mass matrix $\widehat{M}_h$ is obtained analytically and stored as a sparse matrix. The $\widehat{K}_h$ matrix is dense and requires double integrals over the source triangle q and the target triangle p. The inner integral with ψ_n^1 (double layer potential with linearly varying charges over a triangle) has an analytical expression (see Eibert and Hansen [1995], Eq. (11)). The result is smooth if $\boldsymbol{x}$ is outside of the source triangle q. Therefore, for the outer integral with ψ_m^1 over target triangle p, it is sufficient to use quadrature rules (see Figure 6, triangles in the upper-left corner in both left and right sub-figures). We use three symmetric quadrature points for the outer integration over the target triangle p. Using symmetric quadrature points on a triangle instead of a Duffy transform with tensor quadrature rules allows us to reuse the value of the inner integral at these quadrature points for three basis functions associated with the three vertices of the target triangle.

We visualize one column of $\widehat{K}_h$ in Figure 7. It indicates that in each column of $\widehat{K}_h$, the values of entries decay quickly with increasing distance. In addition, the difference between the analytical expression and the quadrature integration for the inner integral is small if the evaluation point $\boldsymbol{x}$ is far away from the integral triangle q. Hence, we switch to three point quadrature integration (see Figure 6, left) if $\boldsymbol{x}$ is beyond some distance. In our experience, four times the average triangle edge length of the source triangle is sufficient.

To allow for large-scale simulations, we only implement the matrix-vector multiplication, rather than explicitly store all matrices. First, each source triangle is assigned three weighted linear basis functions corresponding to the double layer charges at the three vertices. Then for each quadrature point on the target triangle the contribution from all source triangles are collected, analytically for near triangles (see Figure 6, right), and numerically for far triangles (see Figure 6, left). Finally for each target triangle, the double layer potentials at quadrature points are multiplied with the corresponding linear basis function and contribute to one of the vertices of the triangle. This concludes the calculation of the outer integral.

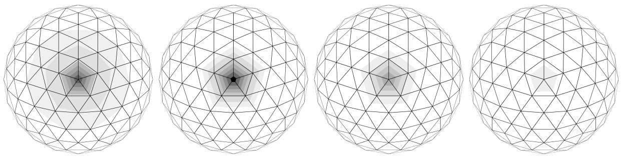


Fig. 7. From left to right: one column of the matrix $\widehat{K}_h$ with analytical inner integral; the relative error of $\widehat{K}_h$ is evaluated with a symmetric {3,7,16}-point quadrature rule [Zhang et al. 2009] for the inner integral compared to the reference analytical integral result. We clearly observe that the values of the entries in one column of $\widehat{K}_h$ quickly decay with increasing distance from the source (diagonal entry). The relative error of the quadrature integral is small for triangles that are far away. The maximum relative errors are 15.6%, 10.4%, and 3.5% respectively for the {3,7,16}-point quadrature rule.

The double layer charges (Eq. (15)) are solved by the preconditioned generalized minimal residual method (GMRES) [Saad and Schultz 1986]. The solved double layer charges are visualized in Figure 8. We use CUDA kernels to execute the matrix-vector multiplication. Each thread is responsible for one target quadrature point and collects all influences from triangle sources. To maximize performance, the CUDA kernels evaluate all triangle sources using the three-point quadrature rule. To account for the analytic integral for triangles in the near field, we apply another correction summation. For each target triangle we find its list of near source triangles over which the analytic integral differs significantly from the three-point quadrature rule. For example, if the distance between them is smaller than four times the mean edge length of the source triangle. With such a list, we can partially assemble the high precision matrix using analytical integration, as well as the low precision matrix using the three-point quadrature rule. The difference is a sparse correction matrix. The sparse correction matrix is stored explicitly and reused. Such a two-step scheme is more efficient than a single CUDA kernel evaluating both, analytical and quadrature rule, depending on the distance because it leads to less register usage and instructions on the GPU.

4.4.3 *Preconditioner.*

Adaptive triangle sizes result in smaller triangle numbers. However, the non-uniform distribution of triangle sizes leads to ill-conditioned Galerkin BEM equations because each equation is weighted differently by the outer integration over the support of test functions with different area. We tried different precondition strategies. The first preconditioner called area preconditioner multiplies each equation by the inverse of the area of the support of the test function. The second preconditioner is a symmetric diagonal preconditioner [Ainsworth et al. 1999; Graham and McLean 2006]. The third preconditioner is based on the observation that the left hand side (LHS) matrix is dominated by elements near each other (Figure 7). We can partially assemble the LHS matrix using analytical integration for the Galerkin BEM Eq. (15). However, the partially assembled matrix can be singular with close-to-zero singular values. Inverting this matrix can lead to a noisy result. To invert it stably, we limit the iteration number in a GMRES solver. We call this a partial-LHS preconditioner. In the end we choose the symmetric diagonal preconditioner because it is more effective than the area preconditioner, and introduces less errors than the partial-LHS preconditioner.

4.4.4 Evaluation of the Magnetic Pressure Discontinuity at Vertices. After we solve the double layer charges Eq. (15), we obtain v_h on the surface. To calculate the magnetic field, the gradient of the double layer potential generated by these double layer charges is required. The total magnetic field $\boldsymbol{H}$ is the linear combination of the external magnetic field $\boldsymbol{H}_{\text{ext}}$ and the gradient of the double layer potential $\nabla(Wv_h)$:

$$\boldsymbol{H}(\boldsymbol{x}) = \frac{\mu_0}{\mu_1}\boldsymbol{H}_{\text{ext}}(\boldsymbol{x}) + \frac{1}{\mu_1}\nabla(Wv_h)(\boldsymbol{x})\,, \quad \boldsymbol{x} \in \Omega\,. \tag{21}$$

For a complete derivation we refer to Andjelic et al. [2011] and Eq. (3.20) therein. Eq. (21) describes the magnetic field everywhere inside the ferrofluid. Once we know the magnetic field, we use it to calculate the magnetic pressure discontinuity on the boundary. In the work of Byrne [1977] and specifically in Table 2 therein, many modern and classic theories on the magnetic forces in ferrofluids are analyzed. Byrne concluded that all of the theories surveyed lead to the same magnetic pressure discontinuity given by

$$P_{\text{mag}} = \mu_0 \frac{1}{2}\chi\|\boldsymbol{H}(\boldsymbol{x})\|^2 + \frac{1}{2}\mu_0(\chi\boldsymbol{H}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}))^2\,, \quad \boldsymbol{x} \in \Gamma\,. \tag{22}$$

This equation was also employed in a series of successful numerical simulations for the static shapes of ferrofluids described in the computational physics literature (see, e.g., Cao and Ding [2014], Eq. (10); Gollwitzer et al [2007], Eq. (3.5); Lavrova et al. [2006], Eq. (10); Lavrova et al. [2008], Eq. (3)). Please note, that $\boldsymbol{H}(\boldsymbol{x})$ should be considered as the limit of the interior field to the boundary (instead of directly measuring at the boundary) since $\boldsymbol{H}$ is discontinuous across the boundary and not well defined on the boundary. We evaluate the magnetic pressure discontinuity P_{mag} at the face quadrature points and convert them to vertex quantities by the procedure described in Section 4.2.2.

The normal on each quadrature point is well defined. The only technical difficulty in Eq. (21) and Eq. (22) is the gradient of the double layer potential at the interior limit of the triangle quadrature point given by

$$\nabla(Wv)(\boldsymbol{x}) := \lim_{\Omega\ni\tilde{\boldsymbol{x}}\to\boldsymbol{x}\in\Gamma} \nabla(Wv)(\tilde{\boldsymbol{x}})\,. \tag{23}$$

We simplify the expression on the right hand side (see Steinbach [2007], page 135 and Lemma 6.16)[1]:

$$\begin{aligned} \frac{\partial}{\partial\tilde{\boldsymbol{x}}}(Wv)(\tilde{\boldsymbol{x}}) &= \frac{1}{4\pi}\int_\Gamma v(y)\frac{\partial}{\partial\tilde{\boldsymbol{x}}}\left(\frac{\partial}{\partial n_y}\frac{1}{\|\tilde{\boldsymbol{x}}-\boldsymbol{y}\|}\right)ds_y \\ &= -\frac{1}{4\pi}\int_\Gamma\left(\text{curl}_{\Gamma,y}\, v(y)\times\nabla_{\tilde{\boldsymbol{x}}}\frac{1}{\|\tilde{\boldsymbol{x}}-\boldsymbol{y}\|}\right)ds_y\,, \end{aligned} \tag{24}$$

in which the definition of the surface curl of a scalar function is given by

$$\text{curl}_\Gamma v = \boldsymbol{n}(\boldsymbol{x}) \times \nabla(\tilde{v})\,. \tag{25}$$

Here, $\tilde{v}$ is an arbitrary extension of the function v defined on the surface Γ to $\mathbb{R}^3$. The surface curl of a linear basis function is piecewise constant, so that we only need to evaluate the gradient of a single layer potential with constant charges on a triangle. Fortunately, the integral of the gradient of a single layer potential of constant charges over a triangle has a close-form solution (see Eibert and Hansen [1995], Eq. (4); Graglia [1993], Eq. (34)). The negative magnetic pressure is visualized in Figure 8.

4.5 BEM Pressure Solve

The goal of the pressure solver is to evaluate the negative gradient of the pressure and apply it as the velocity update. We improve the accuracy of the pressure solver by replacing the original collocation BEM with a Galerkin BEM for mixed boundary conditions.

Once we obtain the magnetic pressure at the vertices, we can combine it with surface tension and gravity potential to formulate the final pressure as the Dirichlet boundary condition at the fluid-air boundary:

$$P_{\text{total}}(\boldsymbol{x}) = -P_{\text{mag}}(\boldsymbol{x}) + \sigma\kappa(\boldsymbol{x}) - \rho\boldsymbol{g}\cdot\boldsymbol{x}\,, \quad \boldsymbol{x} \in \Gamma_{\text{air}}\,, \tag{26}$$

where $\boldsymbol{x}$ is the vertex position on the fluid-air boundary Γ_{air}, P_{total} is the total pressure, P_{mag} is the magnetic pressure, κ is the signed mean curvature, ρ is the ferrofluid density, and $\boldsymbol{g}$ is the gravity acceleration. Please note, that there is a negative sign in front of the magnetic pressure which is the increase of hydrostatic pressure [Byrne 1977] corresponding to the loss of the magnetic energy density. All quantities in this equation are visualized in Figure 8.

To get the pressure at solid boundaries, the following Laplace equation is solved in three dimensions [Da et al. 2016]:

$$\begin{aligned} \nabla^2 P &= 0\,, & (27)\\ P|_{\text{air}} &= P_{\text{total}}\,, & (28)\\ \frac{\partial P}{\partial n}|_{\text{solid}} &= \frac{\rho}{\Delta t}(\overline{\boldsymbol{u}} - \boldsymbol{u}_{\text{solid}})\cdot\boldsymbol{n}\,. & (29) \end{aligned}$$

After the pressure P is computed, the velocity $\overline{\boldsymbol{u}}$ after Helmholtz decomposition is updated according to

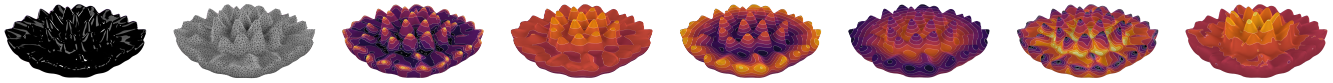
$$\boldsymbol{u}_{\text{new}} = \overline{\boldsymbol{u}} - \frac{\Delta t}{\rho}\nabla P\,. \tag{30}$$

Da et al. [2016] solve the aforementioned equation system using the collocation BEM. It outputs the pressure and normal component of the pressure derivatives at the vertex positions. The tangential component of the pressure gradient is calculated per triangle based on the pressure value on its vertices and combined together with the normal part on the vertex to get the full gradient at the vertex positions.

However, it is discovered that using such a collocation pipeline, if one plugs in the gravity potential as the boundary condition, the negative gradient does not correspond to gravity, not even to a constant vector as illustrated in Figure 9. This motivated us to use a higher-precision Galerkin BEM.

In the following, we summarize the main idea of our Galerkin BEM pressure solver. For further details regarding the BEM we refer to Rjasanow and Steinbach [2007] and specifically to Chapter 2.33 and Eq. (2.41) therein.

[1] Please note, that Steinbach's [2007] definition of the double layer potential is negative compared to ours.



(a) Final Render (b) Surface Mesh (c) Surface Tension (d) Gravity (e) ϕ_{ext} (f) D.L. Charges (g) $-P_{\text{mag}}$ (h) P_{total}

Fig. 8. Different stages of the pressure calculation. Given the surface mesh, we calculate the surface tension and gravity potential based on the curvature of the mesh and vertex positions (see Section 4.3). To calculate the magnetic pressure (see Section 4.4), we first evaluate the external magnetic scalar potential ϕ_{ex} (see Section 4.4.1), and then solve the boundary integral equation to get the double layer charges (see Section 4.4.2). The double layer (D.L.) charges are used to evaluate the negative magnetic pressure $-P_{\text{mag}}$ (see Section 4.4.4). Combining surface tension, gravity potential, and negative magnetic pressure we obtain the total pressure P_{total} (see Eq. (26)) whose negative gradient is added to the velocity field (see Section 4.5).

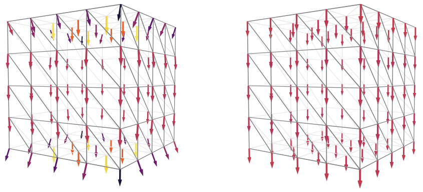


Fig. 9. Negative gradient of the gravity potential solved on a cubic domain. Left: collocation BEM pipeline used by Da et al. [2016] leading to visible errors. Right: our Galerkin BEM pipeline providing results close to the analytical solution.

The liquid is enclosed by manifold triangle meshes. Due to the requirement of smoothness of the solution, the pressure is approximated using continuous piece-wise linear bases defined on the vertices (see Figure 4, left). On the other hand, the normal derivative of the pressure is approximated using piece-wise constant bases defined on the triangles (see Figure 4, right). The pressure inside the liquid is the combination of the single layer potential generated by piece-wise constant charges representing boundary pressure normal derivatives on the triangles, and the double layer potential generated by continuous piece-wise linear charges representing the boundary pressure value on the vertices.

When the liquid is in contact with solids, the normal derivative of the pressure on each contact triangle is known since it must cancel the normal component of the velocity of the triangle. Therefore, these solid-contact triangles are marked as Neumann triangles. The rest of the triangles in the liquid-air boundary are marked as Dirichlet triangles, because the normal derivative of pressure on them is unknown. Similarly, the vertices in the liquid-air boundary, and those in the liquid-air-solid triple junction have known pressure values, and are marked as Dirichlet vertices. The remaining vertices that lie in the interior of the liquid-solid boundary are marked as Neumann vertices.

In summary, the double layer charges (pressure) on Dirichlet vertices are given by gravity, surface tension, magnetic forces (see Eq. (26)). The single layer charges (normal derivatives) on Neumann triangles are determined by solid velocities (see Eq. (29)). The unknown double layer charges (pressure) are associated with the Neumann vertices in the liquid-solid interface, and unknown single layer charges (normal derivatives) are associated with the Dirichlet triangles in the liquid-air interface. After we solve the linear equation of the Galerkin BEM by preconditioned GMRES (Section 4.4.3), we obtain the complete double layer charges on all vertices, and complete single layer charges on all triangles. Once we get all the linear double layer charges and constant single layer charges for each triangle, we can use the analytical solution (see Eq. (24) in this work for the gradient of the double layer potential and Eq. (4) in the work of Eibert and Hansen [1995] for the single layer potential) to calculate their gradient at the interior limit gradients at the quadrature points on each triangle. Finally we convert the gradient on quadrature points to the gradient on vertices by the face quadrature to vertex process described in Section 4.2.2. The gradient evaluation pipeline resembles what we described in Section 4.4.4.

Table 1. The table provides an overview of the numerical examples presented in this work. Max. N_v denotes the number of vertices, Δt the time step size measured in seconds, χ the magnetic susceptibility, and σ the surface tension measured in N/m. Min. E.L. denotes the minimum edge length defined in meters as a measure for the mesh resolution. Max. R.T. measured in seconds denotes the maximum run time per frame (i.e. one cycle of Algorithm 1). The damping parameter listed below is the portion of the smoothed velocity field $\eta \in [0, 1]$ in Eq. (10). The damping parameter (D.) is not set up in a way that damping eliminates instabilities, but rather in a way that the simulation quickly reaches the equilibrium. A constant density of 1 300 kg/m^3 is used in all simulations.

Scene	**Max.** $N_{\mathbf{v}}$	**Steps**	Δt/s	χ	σ/(N/m)	**Min. E.L.**/m	**Max. R.T.**/s	**D.**
5.1: Sphere Test	41k	–	–	1.0	0.00	$2.5 \cdot 10^{-4}$	14.4	0.0
5.2: Wavenumber Test	22k	2 400	$5 \cdot 10^{-4}$	1.0	[0.02, 0.07]	$6.0 \cdot 10^{-4}$	22.0	0.7
5.3: Ablation Study	6.6k	2 000	$3 \cdot 10^{-4}$	1.0	0.045	$7.8 \cdot 10^{-4}$	5.8	0.1
5.4: Resolution Test	8.7k	2 000	$3 \cdot 10^{-4}$	1.0	0.045	$7.0 \cdot 10^{-4}$	8.1	0.1
5.5: Rotating Magnet	5.6k	5 000	$5 \cdot 10^{-4}$	2.0	0.02	$8.0 \cdot 10^{-4}$	5.8	0.7
5.6: Labyrinth Pattern	28k	2 200	$5 \cdot 10^{-4}$	1.5	0.02	$5.0 \cdot 10^{-4}$	45.0	0.5
5.7: Climbing Ferrofluid	8.2k	7 080	$3 \cdot 10^{-4}$	1.0	0.036	$6.5 \cdot 10^{-4}$	10.0	0.5

5 NUMERICAL EXAMPLES

Based on the methodology described in the previous section, we implemented our surface-only ferrofluid solver according to Algorithm 1 using C++/CUDA. In this section, we present a variety of numerical examples evaluating our solver with respect to accuracy, capability and performance. Table 1 provides an overview of the scenes presented throughout this section including the corresponding physical parameters and computation time.

5.1 Accuracy of the Magnetic Solver

To validate the correctness of the magnetic solver, we compare the analytical solution of a sphere filled with linearly magnetizable material placed in a homogeneous external magnetic field. We gradually refine the mesh to test the convergence of the boundary element solver. We set the external magnetic field to $\boldsymbol{H}_{\text{ext}} = (0, 0, 1)^{\mathsf{T}}$. For a homogeneous sphere with susceptibility χ, the theoretical magnetic

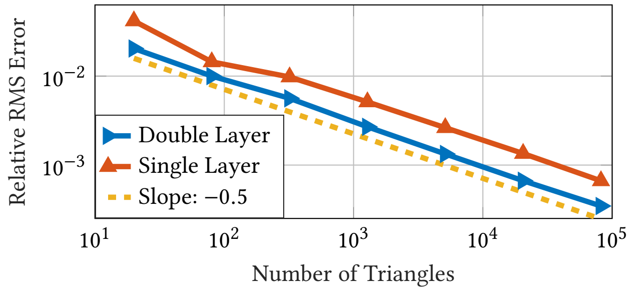


Fig. 10. Relative RMS error of the sphere magnetization calculated by double layer and single layer formulations vs. number of triangles discretizing the sphere. Both axes are logarithmically scaled. The reference line has a slope of −0.5. For each vertex the difference to the reference solution is calculated and the norm of the difference divided by the reference solution defines the error.

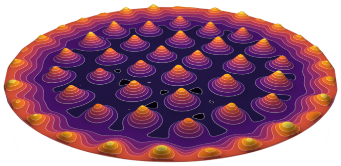


Fig. 11. The pattern used in our wavenumber analysis. Surface tension of $\sigma = 0.03$ N/m, density of $\rho = 1300$ kg/m^3, gravity of $g = 9.81$ m/s^2, and a disk diameter of $D = 8$ cm are applied. The height contour distance is 0.5 mm.

field inside the sphere is a constant field $(0, 0, 3/(3 + \chi))^\intercal$. In Figure 10, we can see that the magnetic field computed by the double layer potential formula adopted in this work and the magnetic field computed with the single layer formula converge to the analytical solution. Moreover the double layer is more accurate given the same amount of triangles.

5.2 Surface Tension and Magnetic Field Strength

The influence of surface tension and field strength is systematically investigated. Fluids of different surface tension are placed in a vertical homogeneous magnetic field with varying strengths; see Figure 13. The surface tension coefficients are 0.02 N/m, 0.025 N/m, 0.03 N/m, and 0.05 N/m (from bottom to top); the density is ρ = 1300 kg/m^3 and the gravity $g = 9.81$ m/s^2. Given these parameters we can calculate the wavenumber of the pattern as $k = \sqrt{\rho g/\sigma}$ and the corresponding wavelength as $\lambda = 2\pi/k$ (see Rosensweig [1987], Eq. (45)). From bottom to top, the theoretical wavelengths are 0.79 cm, 0.88 cm, 0.96 cm, and 1.24 cm. Four white marks illustrating the theoretical wavelengths are shown in the plot providing a visual comparison. Clearly, our simulation matches the theoretical wavelength. Although the previous work of Huang et al. [2019] also shows a similar trend in its analysis, there is no way to compare their result to the theoretical results because their surface tension parameter does not correspond to a physical parameter.

A higher surface tension coefficient results in a higher critical external magnetic field strength beyond which the pattern starts to form. This is also reflected in the first two columns on the left side of Figure 13. The quantitative correctness of our solver is analyzed in terms of the dominant wavenumber of the pattern in the Rosensweig instability (i.e. the surface forms a regular pattern of peaks and valleys). We place ferrofluids of different surface tension parameters σ in a cylindrical container with a diameter of 8 cm within a homogeneous magnetic field. We measure the dominant wavenumber of the pattern by transforming the height field to the Fourier domain. The measured result is plotted in Figure 12. Our result agrees well with the theoretical prediction $k = \sqrt{\rho g/\sigma}$ (Rosensweig [1987], Eq. (45)).

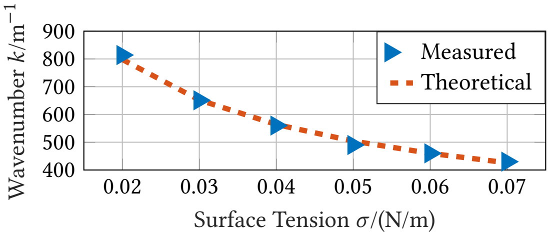


Fig. 12. Dominant wavenumber k of the pattern vs. surface tension σ. The ferrofluid is placed in a homogeneous magnetic field. The red curve shows the theoretical prediction of the dominant wavenumber $k(\sigma) = \sqrt{\rho g/\sigma}$ with $\rho = 1300$ kg/m^3 and $g = 9.81$ m/s^2. The blue dots indicate dominant wavenumbers in the Fourier domain of our simulation results.

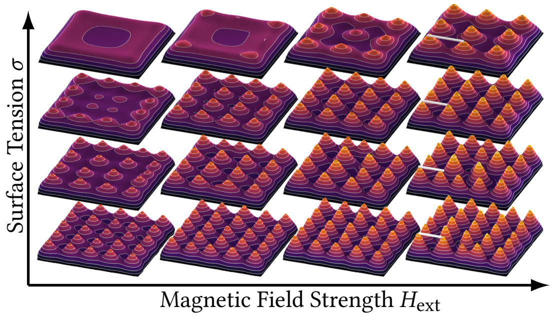


Fig. 13. The pattern of the ferrofluid's surface for different surface tensions and field strengths. The ferrofluid has a density of $\rho = 1300$ kg/m^3. From bottom to top, the surface tension σ is 0.02 N/m, 0.025 N/m, 0.03 N/m, and 0.05 N/m. From left to right, the field strength H_{ext} is 13 kA/m, 14 kA/m, 15 kA/m, and 16 kA/m. The white marks on the right illustrate the theoretical wavelength predicted by the analytical formula $\lambda(\sigma) = 2\pi/\sqrt{\rho g/\sigma}$ with $g = 9.81$ m/s^2.

5.3 Ablation Study

In this section, we compare the importance of the different components of our approach. The complete algorithm uses the full (i.e. both, the tangential and normal component of the re-constructed velocity field) Helmholtz decomposition with analytical integration, Galerkin BEM pressure solver, double-layer potential formulation for the magnetic field problem, and the semi-analytical approach to calculate the matrix in the Galerkin BEM. We show that the most important component is the double-layer formulation and the semi-analytical approach to evaluate the matrix. Detailed comparisons providing an overview of the critical components of our approach can be found in Figures 14, 15, 16, and 17.

5.3.1 Partial vs. Full Helmholtz Decomposition. In the work of Da et al. [2016], the tangential velocity on the mesh is kept in the

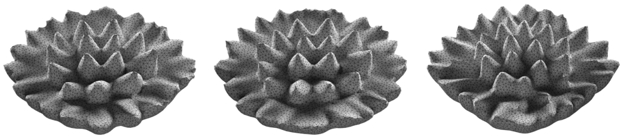

(a) Partial + Duffy. (b) Partial + Analytical. (c) Full + Analytical.

Fig. 14. Two partial Helmholtz decompositions (a) and (b) are compared vs. the full Helmholtz decomposition (c) using analytical integration; (a) uses the quadrature rules with Duffy transform, (b) uses analytical integration. While the partial Helmholtz decomposition just has a minor influence on the static shape, during the dynamical simulation, it induces more velocity fluctuations and re-meshing operations.

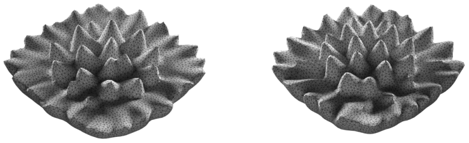

(a) Collocation BEM. (b) Galerkin BEM.

Fig. 15. In (a), a collocation BEM is used to solve the mixed boundary value problem for the pressure while in (b), a Galerkin BEM is employed. The collocation method dissipates more energy despite similarly looking static shapes.

Helmholtz decomposition step. We denote this as a partial Helmholtz decomposition. In Figure 14, we compare the complete algorithm against the partial Helmholtz decomposition using quadrature integration with Duffy transform and analytical integration. Such a partial Helmholtz decomposition leads to noisy simulations, but the shapes of the spikes are not much affected.

The noise could originate from less accurate quadrature integration [Da et al. 2016], or from the unattended tangential component of the velocity. The analytical integration helps to reduce the noise, especially when no artificial damping is added (not shown in Figure 14). However, when some artificial damping is included, it does not outperform the quadrature integration technique, and the noise is still perceptible compared to the full Helmholtz decomposition with analytical integration.

5.3.2 Collocation / Galerkin BEM Pressure Solver. The collocation BEM in the pressure solver does not show a noisy result as illustrated in Figure 15. However, during the simulation, energy quickly decreases. In contrast, the Galerkin BEM can preserve the energy satisfying the inviscid ferrofluid assumption.

5.3.3 Single vs. Double Layer Potential Formulation. An alternative formulation for the magnetic field problem was proposed by Andjelic et al. [2011] called a single-layer potential formulation. We tried this formulation, but the numerical error of the magnetic part is too large as shown in Figure 16. We observe little performance advantage in our single layer implementation as argued by Andjelic et al. [2011]. Hence, in our case the double layer potential formulation is more beneficial compared to the single layer formulation.

5.3.4 Vertex / Face Magnetic Pressure Calculation. In our approach, the magnetic pressure is first evaluated at the quadrature point on each face using the face normal. An alternative approach is to use vertex normals and the magnetic field strength at vertex positions. We observed that the vertex normal approach (see Figure 17) leads to inaccurate magnetic pressure and crashes the simulation.

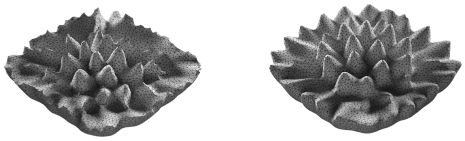

(a) Single Layer. (b) Double Layer.

Fig. 16. Single (a) and double (b) layer potential formulations are used to compute the magnetic field. It can be clearly observed that the single layer formulation is not sufficient to generate correct shapes.

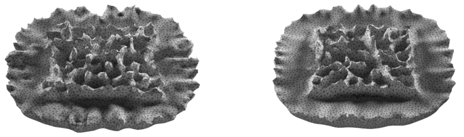
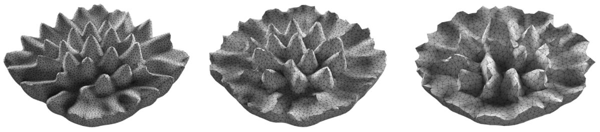

(a) Vertex. (b) Face to Vertex.

Fig. 17. In (a), we convert the magnetic field from face quadrature positions to vertex positions and evaluate the magnetic pressure using vertex normals. In (b), we evaluate the magnetic pressure on quadrature points and convert the pressure to vertex values. The last frame before the vertex normal approach crashes is shown.

5.3.5 Symmetric Triangle Quadrature / Semi-analytical Correction. The solution of the double layer potential problem requires the application of a matrix $\widehat{K}_h$. Each entry in this matrix consists of an inner integral and an outer integral over different triangles. To show the sensitivity of our algorithm with respect to the accuracy of the integration techniques, we change the number of quadrature points for the inner integral. In the three examples, three, seven and sixteen quadrature points [Zhang et al. 2009] are used respectively to evaluate the inner integral of the $\widehat{K}_h$ matrix. We evaluate the outer integral using a three point quadrature rule, and evaluate the gradient using analytical integration. All three simulations crashed after three, five, eighteen steps respectively. One might argue that the failure is due to the mismatch of the integration methods between solving the double layer charges and evaluating the gradient of it. The fact is that even if we use the three point quadrature rule for both, the evaluation of $\widehat{K}_h$ and the evaluation of the double layer charges, it still crashes after three steps.

5.4 Resolution Test

We analyze the importance of the mesh resolution on the simulations as shown in Figure 18. The minimum edge length is increased to reduce the resolution. A visible loss of quality is observed if no sufficient amount of triangles is available.



(a) 0.7 mm, 6.7k. (b) 1 mm, 3.3k. (c) 1.3 mm, 2.1k.

Fig. 18. Simulations with different resolutions and corresponding triangle numbers. From left to right we set the minimum edge length to be 0.7 mm, 1.0 mm, 1.3 mm corresponding to 6.7k, 3.3k, 2.1k triangles respectively. Naturally, a correct simulation requires a sufficiently high resolution to capture the spikes' geometry.

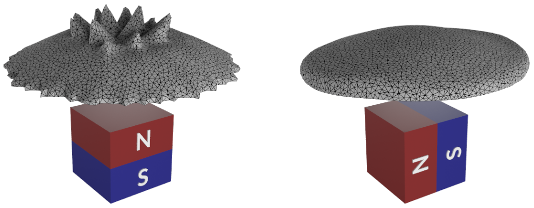


Fig. 19. The fluid is approached by an external magnet which leads to the formation of spikes. The orientation of the magnet is relevant as demonstrated here.

5.5 Rotating Magnet Experiment

We approach the fluid with a cubic magnet from below leading to the formation of the characteristic spikes. We furthermore simulate the impact of the magnet's orientation on the formation of spikes. If the magnet is rotated in a way that its field lines are arranged in parallel to the fluid's surface, naturally no spikes can be generated. Our simulation reproduces this effect as demonstrated in Figure 19.

5.6 Labyrinth Pattern

Ferrofluids exhibit other nonlinear patterns beyond their familiar spikes. When the fluid is located in between a narrow space formed by two parallel planes (in a so-called Hele-Shaw cell [Saffman 1986]) and is exposed to a magnetic field perpendicular to the planes, a labyrinth structure is generated [Rosensweig et al. 1983]. This effect can successfully be reproduced with our approach. With increasing magnetic field strength, the labyrinth pattern develops as shown in Figure 2. The gap between the two parallel planes is 2 mm. The visible expansion of the fluid is an artifact associated with a strong smooth damping coefficient $\eta = 0.5$. The artificially smoothed velocity after the Helmholtz decomposition is not strictly divergence-free introducing volume errors. Since the surface-only fluid concept does not include viscosity, the volume error induced by artificial smooth damping seems inevitable unless an extra Helmholtz decomposition step is applied to enforce incompressibility again.

5.7 Climbing Ferrofluid

As a more complex example, we use multiple magnets to control the movement of ferrofluids. We position the fluid in a closed space between the floor and the ceiling. Moreover, two cylindrical electromagnets are located below the floor and above the ceiling, and change their relative strength to control the fluid. Initially, the fluid is located on the ground. We increase the strength of the upper magnet and attract the ferrofluid to the top forming a bridge between the floor and the ceiling finally getting fully attracted to the ceiling. Later on, we turn on the lower electromagnet to attract the ferrofluid down to the floor and reduce the strength of the upper electromagnet until the fluid rests completely on the floor. This simulation is shown in Figures 1 and 20.

5.8 Effectiveness of the Preconditioner

We analyze the effectiveness of the preconditioner (see Section 4.4.3) in the context of the magnetization problem (Section 4.4.2) as shown in Figure 21 and in the context of the pressure solver (see Section 4.5) as shown in Figure 22. We test our solver using the equilibrium shape of the wavenumber test with $\sigma = 0.04$ and $H_{\text{ext}} = 1.5 \cdot 10^4$ A/m because the diverse triangle sizes and large numbers of vertices lead to challenging ill-conditioned system. We solve the system allowing different maximum iterations and measure the computation time. The original system converges very slowly. If we use the inverse element area (support area for ψ^0, one third of support area for ψ^1, see Figure 4) as a preconditioner, the residuals are reduced significantly faster. The diagonal preconditioner [Ainsworth et al. 1999; Graham and McLean 2006] converges faster compared to the area preconditioner and the partial-LHS preconditioner. Since the partial-LHS is not solved exactly, it introduces errors to the final solution, see Figure 22.

5.9 Runtime Analysis

The performance of our simulator is analyzed by running the same fraction of the reference ablation study (Section 5.3) with different mesh resolutions as presented in Figure 23. The runtime of our Galerkin BEM magnetic solver and pressure projection is comparable to the surface tracking procedure [Da et al. 2014]. Although we implemented a naive summation with quadratic complexity, the GPU resources are not fully utilized unless the problem size is sufficiently large. The runtime scales linearly with respect to vertex numbers below 48k. All of our test cases are within this linear range.

6 DISCUSSION

Our surface-only ferrofluid solver is able to simulate a broad range of the characteristic dynamics of ferrofluids including the Rosensweig instability and labyrinth patterns [Rosensweig et al. 1983]. The linear magnetization assumption allows us to only use the surface to solve the magnetization problem, whereas the incompressibility assumption allows us to apply the magnetic forces as the gradient of the magnetic pressure solved with a BEM that only rely on the surface mesh as well.

Since only the information on the surface of the fluid is used, this solver is advantageous compared to volume methods for a large fluid volume. The boundary mesh can accurately capture the interface of fluid and air making it convenient to use a physically accurate surface tension model. The magnetic solver employing the Galerkin BEM has proven accuracy making our solver a strong candidate to study the dynamic free-surface flow of ferrofluids at large-scale. The

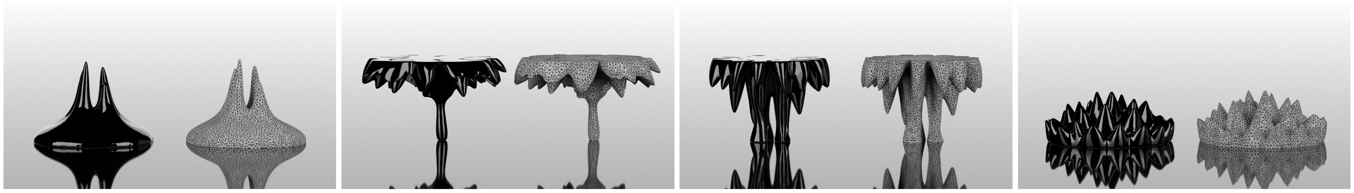


Fig. 20. Two electromagnets, one at the top and another one at the bottom, are employed controlling the movement of the fluid. First, the strength of the upper magnet is increased attracting the fluid to the top. Then, the strength of the lower magnet is increased while gradually reducing the strength of the upper magnet so that the fluid gets finally attracted to the bottom. A time sequence is shown here (left: photorealistically rendered; right: mesh view).

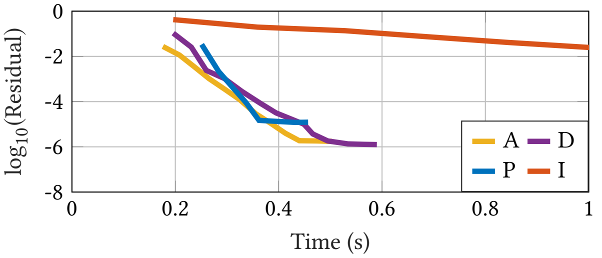


Fig. 21. Normalized residual vs. computation time using the Galerkin BEM magnetic field solver (see Section 4.4.2). A: inverse element area preconditioner. D: diagonal preconditioner. P: partial-LHS assembled with near elements preconditioner. I: reference identity preconditioner. Please note, that the partial-LHS preconditioner shows a steep slope but the error is larger. All solvers have around 0.2 s overhead.

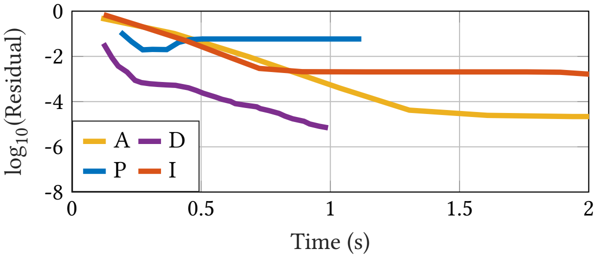


Fig. 22. Normalized residual vs. computation time using the Galerkin BEM pressure solver (see Section 4.5). A: inverse element area preconditioner. D: diagonal preconditioner. P: partial-LHS preconditioner. I: reference identity preconditioner. The partial-LHS preconditioner shows a large error due to its incomplete solve. The fastest convergence shows the diagonal preconditioner.

performance of the whole system is well optimized and the runtime of the magnetic part along with the pressure part is comparable to the surface-tracking procedure [Da et al. 2014].

6.1 State-of-the-Art Comparison

Existing methods simulating ferrofluids include Lagrangian [Huang et al. 2019] and Eulerian approaches [Ni et al. 2020]. Both methods solve equations for volume unknowns. Our BEM approach solves equations for surface unknowns. We discuss their differences in terms of accuracy and computation time. In terms of accuracy, Huang et al. [2019] approximate the ferrofluid with uniformly distributed radial basis functions. As a result, the ferrofluid-air interface is a smooth transition, instead of a sharp transition in reality. Ni et al. [2020] need to discretize both, the fluid and the air using a Cartesian grid. They need to approximate the sharp fluid-air interface with a smooth level-set function. In case a vanishing boundary is required, the whole domain must be sufficiently large to avoid unpleasant effects at the boundary. Our BEM approach can capture the sharp fluid-air interface and naturally support vanishing boundaries.

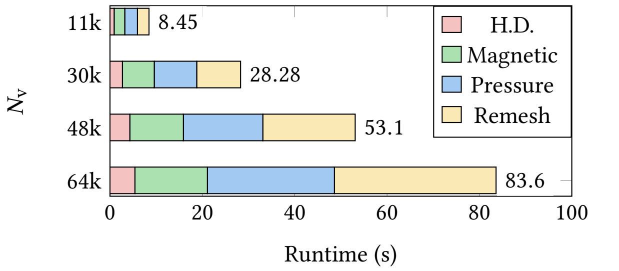


Fig. 23. Runtime analysis and decomposition into the different parts (i.e. Helmholtz decomposition, magnetic part, pressure part, and remeshing part) for different problem sizes. The number of vertices is shown of the vertical axis. The magnetic part includes all steps to turn vertex positions into magnetic pressures. The pressure part contains the whole projection step using the Galerkin BEM. The remeshing part includes a single mesh advection step, and five cycles of topology changes and mesh improvements. The computation time is measured in seconds for one simulation step with no sub-steps. Remeshing is running sequentially on a single thread. The near-field high-precision $\widehat{K}_h$ (see Section 4.4.2) is assembled on the CPU with two 12-core Intel® E5-2687w-v4 @3.0 GHz. All other parts are running on an NVIDIA® GeForce® RTX2080Ti.

In terms of computation time, we analyze the magnetic part conceptually and experimentally. The BEM solves equations for less unknowns, but each vertex unknown is associated with on average six quadrature points. Solving the system requires iterative pair-wise interactions between all quadrature points. Our current naive implementation performs well for small problems, but will fail in large cases. The Lagrangian method [Huang et al. 2019] solves equations for substantially more volume unknowns for the same resolution, and also requires iterative pair-wise interactions between all particles. Hence, a naive implementation of Huang et al. is likely to be inefficient. Their fast multipole implementation is necessary. The Eulerian method [Ni et al. 2020] involves more volume unknowns to the work of Huang et al. because the air needs to be discretized. They solve Poisson's problem on a Cartesian grid, which can benefit

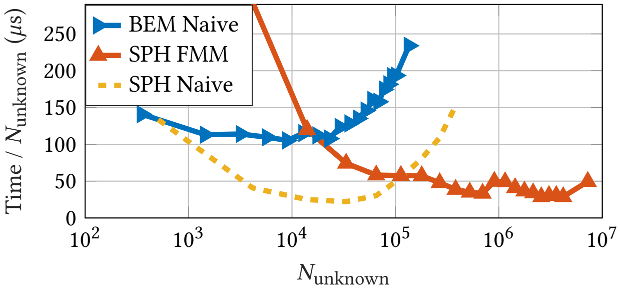


Fig. 24. Computation time to solve the magnetic part per unknown vs. the number of unknowns of our BEM with naive summation, SPH [Huang et al. 2019] with naive summation and fast multipole method (FMM). Both naive summation methods have stable costs per number of unknowns before the thousands of GPU cores are fully occupied. After the GPU is fully utilized, the cost increases. BEM costs twice as much as SPH for the same number of unknowns. The cost of the FMM is generally stable for large problem sizes.

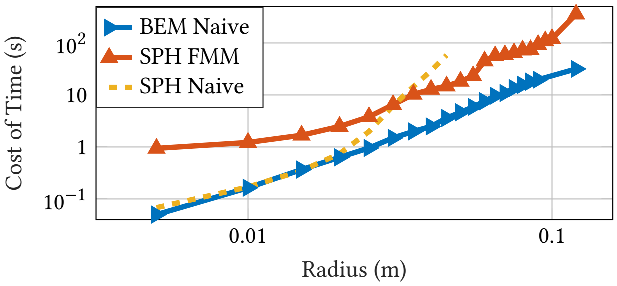


Fig. 25. Computation time to solve the magnetic part vs. the radii of the sphere with a fixed space resolution (1 mm). The BEM is most effective for this test because of the advantage of less equation sizes compared to the volume approach, although the cost of each unknown is slightly higher (see Figure 24).

from the existing Eulerian fluid simulation framework. Conceptually, for small and intermediate scenes, the method of Ni et al. [2020] may be the most promising one.

The work of Ni et al. is quite recent, and no source code is available. Hence, we compare the result of Huang et al. [2019] to our approach using their released source code. We test both algorithms in a sphere magnetization test (Section 5.1) with identical spatial resolution (1 mm) but with increasing radii from 5 mm to 12 cm. For the largest radius, the number of unknowns using the BEM is 130K, while the number using SPH is 7.3M. Since their ratio depend on the geometry, a more representative quantity is the cost of time per number of unknowns. If the cost per unknown is similar, then a smaller number of unknowns would lead to a more efficient algorithm. We plot the computation time divided by the problem size in Figure 24. It shows that our method is more expensive per unknown, but still comparable to SPH [Huang et al. 2019]. Since the BEM solves much less unknowns, it may outperform SPH. In the ideal spherical case, our method outperforms SPH [Huang et al. 2019] with FMM integrated; see Figure 25. In the future, FMM can be incorporated within our method to reduce the cost per unknown at large scale, potentially enabling more speed-up.

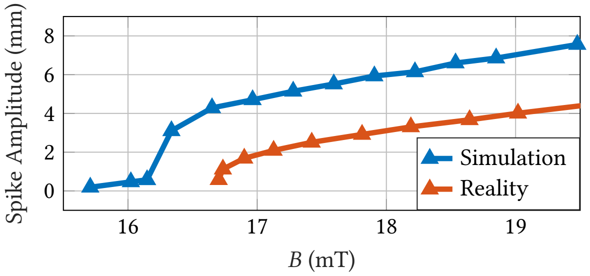


Fig. 26. Illustration of the magnetic induction vs. the spike amplitudes (i.e. the distance between the peak and valley of the pattern) in the center region of a disk. Real measurements [Gollwitzer et al. 2007] are compared to the results of our simulations.

6.2 Limitations

Our approach can accurately simulate complex patterns which occur at the ferrofluids' surfaces. Next to qualitative accuracy of the whole scene, the correct wavelengths can be reproduced quantitatively accurate. However, the surface-only concept comes with limitations in two aspects. First, it neglects viscosity and the presence of vortices within the fluid's volume. Second, BEM solvers for the magnetization problem can naturally only handle linearly magnetizable materials. As a consequence, the influence of the external magnetic field on the height of the spikes is quantitatively less accurate as shown in Figure 26. Although we use the exact physical parameters of the fluid reported by Gollwitzer et al. [2007] for this comparison, we can only qualitatively match their measurements. This is caused by the linear magnetization model in our work approximating their nonlinear model ($\chi = 1$). Taking non-linear magnetic responses of real ferrofluids into account would require volume discretization. Another problem is the handling of complex solid geometries. This is associated with the surface-only fluid solver and the surface tracking program. This necessitates further developments with respect to solid geometry interaction, multi-threading, etc.

6.3 Future Work

The current solver only implements naive summations. For larger problem sizes, the FMM [Beatson and Greengard 1997] or the particle-particle particle-mesh (P^{3}M) method [Zhang and Bridson 2014] could be incorporated to accelerate the summation. Limited by the explicit time integrator, our solver must use relatively small time steps. A possible improvement could be to incorporate implicit integration involving surface tension and magnetic energy.

Moreover, we would like to point out that the way we incorporate magnetic effects into fluids is not limited to a surface-only approach opening multiple avenues for future work. It is possible to extract the triangle meshes from the free-surface of an Eulerian fluid solver or a mixed Eulerian-Lagrangian fluid solver, and calculate the magnetic pressure everywhere on the mesh. The magnetic force and the surface tension force can then be applied by adding them to the Dirichlet boundary condition on the fluid-air surface.

ACKNOWLEDGEMENTS

This work has been supported by KAUST (individual baseline funding). We thank Albert Chern, Marcel Padilla, and Ulrich Pinkall for the initial discussion that gave birth to the inspiration of surface-only ferrofluids. The helpful discussions with Torsten Hädrich, Wolfgang Heidrich, Franziska Lissel, Dmitry A. Lyakhov, Sören Pirk, Jing Ren, Ravi Samtaney, and Han Shao as well as the valuable comments of the anonymous reviewers are gratefully acknowledged.

REFERENCES

Mark Ainsworth, William Mclean, and Thanh Tran. 1999. The Conditioning of Boundary Element Equations on Locally Refined Meshes and Preconditioning by Diagonal Scaling. *SIAM J. Numer. Anal.* 36, 6 (1999), 1901–1932.

Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2013. Versatile surface tension and adhesion for SPH fluids. *ACM Trans. Graph.* 32, 6, Article 182 (Nov. 2013), 8 pages.

Saad Alazemi, A. Bibo, and M.F. Daqaq. 2015. A ferrofluid-based energy harvester: An experimental investigation involving internally-resonant sloshing modes. *The European Physical Journal Special Topics* 224 (11 2015), 2993–3004. https://doi.org/10.1140/epjst/e2015-02602-9

Zoran Andjelic, Günther Of, Olaf Steinbach, and Peter Urthaler. 2011. Boundary element methods for magnetostatic field problems: a critical view. *Computing and visualization in science* 14, 3 (2011), 117–130.

Ryoichi Ando, Nils Thürey, and Chris Wojtan. 2013. Highly Adaptive Liquid Simulations on Tetrahedral Meshes. *ACM Trans. Graph. (Proc. SIGGRAPH 2013)* (July 2013).

Rick Beatson and Leslie Greengard. 1997. A short course on fast multipole methods. In *Wavelets, Multilevel Methods and Elliptic PDEs.* Oxford University Press, 1–37.

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 1–10.

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

Andreas G. Boudouvis, Jason L. Puchalla, Laurence E. Scriven, and Ronald E. Rosensweig. 1987. Normal field instability and patterns in pools of ferrofluid. *Journal of Magnetism and Magnetic Materials* 65, 2 (1987), 307 – 310.

Robert Bridson and Matthias Müller-Fischer. 2007. Fluid simulation: SIGGRAPH 2007 course notes. (2007), 1–81.

Tyson Brochu and Robert Bridson. 2009. Robust Topological Operations for Dynamic Explicit Surfaces. *SIAM Journal on Scientific Computing* 31, 4 (2009), 2472–2493.

Denis Brousseau, Ermanno F. Borra, and Simon Thibault. 2007. Wavefront correction with a 37-actuator ferrofluid deformable mirror. *Opt. Express* 15, 26 (Dec 2007), 18190–18199.

James V. Byrne. 1977. Ferrofluid hydrostatics according to classical and recent theories of the stresses. *Proceedings of the Institution of Electrical Engineers* 124, 11 (November 1977), 1089–1097.

Yuan Cao and Ze J. Ding. 2014. Formation of hexagonal pattern of ferrofluid in magnetic field. *Journal of Magnetism and Magnetic Materials* 355 (2014), 93–99.

Nuttapong Chentanez, Bryan E. Feldman, François Labelle, James F. O'Brien, and Jonathan R. Shewchuk. 2007. Liquid Simulation on Lattice-Based Tetrahedral Meshes. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2007.* 219–228.

Nuttapong Chentanez, Matthias Müller, Miles Macklin, and Tae-Yong Kim. 2015. Fast Grid-Free Surface Tracking. *ACM Trans. Graph.* 34, 4, Article 148 (July 2015), 11 pages.

Albert Chern, Felix Knöppel, Ulrich Pinkall, Peter Schröder, and Steffen Weißmann. 2016. Schrödinger's Smoke. *ACM Trans. Graph.* 35, 4, Article 77 (July 2016), 13 pages.

Pascal Clausen, Martin Wicke, Jonathan R. Shewchuk, and James F. O'Brien. 2013. Simulating Liquids and Solid-Liquid Interactions with Lagrangian Meshes. *ACM Transactions on Graphics* 32, 2 (April 2013), 17:1–15. Presented at SIGGRAPH 2013.

David Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry.* 312–321.

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 (July 2016), 12 pages.

Akiva J. Dickstein, Shyamsunder Erramilli, Raymond E. Goldstein, David P. Jackson, and Stephen A. Langer. 1993. Labyrinthine Pattern Formation in Magnetic Fluids. *Science* 261, 5124 (1993), 1012–1015.

Michael G. Duffy. 1982. Quadrature Over a Pyramid or Cube of Integrands with a Singularity at a Vertex. *SIAM J. Numer. Anal.* 19, 6 (1982), 1260–1262.

Thomas F. Eibert and Volkert Hansen. 1995. On the calculation of potential integrals for linear source distributions on triangular domains. *IEEE transactions on antennas and propagation* 43, 12 (1995), 1499–1502.

Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. 2002. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics* 183, 1 (2002), 83–116.

Yu Fang, Minchen Li, Ming Gao, and Chenfanfu Jiang. 2019. Silly rubber: an implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–13.

Yun (Raymond) Fei, Henrique Teles Maia, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2017. A Multi-Scale Model for Simulating Liquid-Hair Interactions. *ACM Trans. Graph.* 36, 4, Article 56 (July 2017), 17 pages.

Nick Foster and Ronald Fedkiw. 2001. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques.* 23–30.

Ryan Goldade, Yipeng Wang, Mridul Aanjaneya, and Christopher Batty. 2019. An Adaptive Variational Finite Difference Framework for Efficient Symmetric Octree Viscosity. *ACM Trans. Graph.* 38, 4, Article 94 (July 2019), 14 pages.

Christian Gollwitzer, Gunar Matthies, Reinhard Richter, Ingo Rehberg, and Lutz Tobiska. 2007. The surface topography of a magnetic fluid: a quantitative comparison between experiment and numerical simulation. *Journal of Fluid Mechanics* 571 (2007), 455–474.

Roberto D. Graglia. 1993. On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle. *IEEE transactions on antennas and propagation* 41, 10 (1993), 1448–1455.

Ivan G. Graham and William McLean. 2006. Anisotropic Mesh Refinement: The Conditioning of Galerkin Boundary Element Matrices and Simple Preconditioners. *SIAM J. Numer. Anal.* 44, 4 (2006), 1487–1513.

David Hahn and Chris Wojtan. 2015. High-Resolution Brittle Fracture Simulation with Boundary Elements. *ACM Transactions on Graphics* 34 (07 2015), 151:1–151:12.

David Hahn and Chris Wojtan. 2016. Fast Approximations for Boundary Element Based Brittle Fracture Simulation. *ACM Trans. Graph.* 35, 4, Article 104 (July 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 (July 2019), 15 pages.

Tomokazu Ishikawa, Yonghao Yue, Kei Iwasaki, Yoshinori Dobashi, and Tomoyuki Nishita. 2012. Visual simulation of magnetic fluid taking into account dynamic deformation in spikes. In *Image Electronics and Visual Computing Workshop.*

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*, Gabriela Csurka, Martin Kraus, Robert S. Laramee, Paul Richard, and José Braz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 112–126.

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.

Lina Kafrouni and Oumarou Savadogo. 2016. Recent progress on magnetic nanoparticles for magnetic hyperthermia. *Progress in Biomaterials* 5 (09 2016). https://doi.org/10.1007/s40204-016-0054-6

Todd Keeler and Robert Bridson. 2014. Ocean Waves Animation Using Boundary Integral Equations and Explicit Mesh Tracking. In *ACM SIGGRAPH 2014 Posters (SIGGRAPH '14).* Association for Computing Machinery, New York, NY, USA, Article 11, 1 pages.

Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. 2020. Lagrangian Neural Style Transfer for Fluids. *ACM Transaction on Graphics (SIGGRAPH)* (2020).

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 (June 2008), 79–81.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2019. EUROGRAPHICS Tutorial on Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. (2019).

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 (July 2017), 11 pages.

Olga Lavrova, Gunar Matthies, Teodora Mitkova, Viktor Polevikov, and Lutz Tobiska. 2006. Numerical treatment of free surface problems in ferrohydrodynamics. *Journal of Physics: Condensed Matter* 18, 38 (2006), S2657.

Olga Lavrova, Gunar Matthies, and Lutz Tobiska. 2008. Numerical study of soliton-like surface configurations on a magnetic fluid layer in the Rosensweig instability.

Communications in Nonlinear Science and Numerical Simulation 13, 7 (2008), 1302–1310.

Xiaosheng Li, Xiaowei He, Xuehui Liu, , Baoquan Liu, and Enhua Wu. 2016. Multiphase Interface Tracking with Fast Semi-Lagrangian Contouring. *IEEE Transactions on Visualization and Computer Graphics* 22, 8 (2016), 1973–1986.

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.

Matthias Müller. 2009. Fast and Robust Tracking of Fluid Surfaces. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '09)*. Association for Computing Machinery, New York, NY, USA, 237–245.

Kentaro Nagasawa, Takayuki Suzuki, Ryohei Seto, Masato Okada, and Yonghao Yue. 2019. Mixing sauces: a viscosity blending model for shear thinning fluids. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–17.

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

Ricardo H. Nochetto, Abner J. Salgado, and Ignacio Tomas. 2016a. A diffuse interface model for two-phase ferrofluid flows. *Computer Methods in Applied Mechanics and Engineering* 309 (2016), 497–531.

Ricardo H. Nochetto, Abner J. Salgado, and Ignacio Tomas. 2016b. The equations of ferrohydrodynamics: modeling and numerical methods. *Mathematical Models and Methods in Applied Sciences* 26, 13 (2016), 2393–2449.

Marcel Padilla, Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. 2019. On Bubble Rings and Ink Chandeliers. *ACM Trans. Graph.* 38, 4, Article 129 (July 2019), 14 pages.

Tobias Pfaff, Nils Thürey, and Markus Gross. 2012. Lagrangian Vortex Sheets for Animating Fluids. *ACM Trans. Graph.* 31, 4, Article 112 (July 2012), 8 pages.

Sergej Rjasanow and Olaf Steinbach. 2007. *The fast solution of boundary integral equations.* Springer Science & Business Media.

Ronald E. Rosensweig. 1987. Magnetic Fluids. *Annu. Rev. Fluid Mech.* 19 (Jan 1987), 437–463.

Ronald E. Rosensweig. 1988. An Introduction To Ferrohydrodynamics. *Chemical Engineering Communications* 67, 1 (1988), 1–18.

Ronald E. Rosensweig. 1997. *Ferrohydrodynamics.* Dover Publications.

Ronald E. Rosensweig, Yuki Hirota, Sayaka Tsuda, and K. Russel Raj. 2008. Study of audio speakers containing ferrofluid. *Journal of physics. Condensed matter : an Institute of Physics journal* 20 (05 2008), 204147. https://doi.org/10.1088/0953-8984/20/20/204147

Ronald E. Rosensweig, Markus Zahn, and Raymond Shumovich. 1983. Labyrinthine instability in magnetic and dielectric fluids. *Journal of Magnetism and Magnetic Materials* 39, 1-2 (1983), 127–132.

Yousef Saad and Martin H Schultz. 1986. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* 7, 3 (July 1986), 856–869.

Philip G. Saffman. 1986. Viscous fingering in Hele-Shaw cells. *Journal of Fluid Mechanics* 173 (1986), 73–94.

Stefan A. Sauter and Christoph Schwab. 2010. Boundary element methods. In *Boundary Element Methods.* Springer, 183–287.

Craig Schroeder, Wen Zheng, and Ronald Fedkiw. 2012. Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid. *J. Comput. Phys.* 231, 4 (2012), 2092–2115.

Denis Sievers, Thomas F Eibert, and Volkert Hansen. 2005. Correction to "On the calculation of potential integrals for linear source distributions on triangular domains". *IEEE Transactions on Antennas and Propagation* 53, 9 (2005), 3113–3113.

Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary Element Octahedral Fields in Volumes. *ACM Trans. Graph.* 36, 4, Article 114b (May 2017), 16 pages.

Jos Stam. 1999. Stable Fluids. *Proc. of ACM SIGGRAPH* (1999), 121–128.

Olaf Steinbach. 2007. *Numerical approximation methods for elliptic boundary value problems: finite and boundary elements.* Springer Science & Business Media.

Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

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

Nils Thürey, Chris Wojtan, Markus Gross, and Greg Turk. 2010. A Multiscale Approach to Mesh-Based Surface Tension Flows. In *ACM SIGGRAPH 2010 Papers (SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 48, 10 pages.

Kiwon Um, Xiangyu Hu, and N. Thüerey. 2018. Liquid Splash Modeling with Neural Networks. *CGF* 37, 8 (2018), 171–182.

Benjamin Ummenhofer, Lukas Prantl, Nils Thüerey, and Vladlen Koltun. 2020. Lagrangian Fluid Simulation with Continuous Convolutions. In *ICLR*.

He Wang, Kirill A. Sidorov, Peter Sandilands, and Taku Komura. 2013. Harmonic Parameterization by Electrostatics. *ACM Trans. Graph.* 32, 5, Article 155 (Oct. 2013), 12 pages.

Steffen Weißmann and Ulrich Pinkall. 2010. Filament-Based Smoke with Vortex Shedding and Variational Reconnection. *ACM Trans. Graph.* 29, 4, Article 115 (July 2010), 12 pages.

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

Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2009. Deforming Meshes That Split and Merge. In *ACM SIGGRAPH 2009 Papers (SIGGRAPH '09)*. Association for Computing Machinery, New York, NY, USA, Article 76, 10 pages.

You Xie, Erik Franz, Mengyu Chu, and Nils Thürey. 2018. TempoGAN: A Temporally Coherent, Volumetric GAN for Super-Resolution Fluid Flow. *ACM Trans. Graph.* 37, 4, Article 95 (2018), 15 pages.

Gaku Yoshikawa, Katsuhiro Hirata, Fumikazu Miyasaka, and Yu Okaue. 2010. Numerical analysis of transitional behavior of ferrofluid employing MPS method and FEM. In *Digests of the 2010 14th Biennial IEEE Conference on Electromagnetic Field Computation.* 1–1.

Linbo Zhang, Tao Cui, and Hui Liu. 2009. A set of symmetric quadrature rules on triangles and tetrahedra. *Journal of Computational Mathematics* (2009), 89–96.

Xinxin Zhang and Robert Bridson. 2014. A PPPM fast summation method for fluids and beyond. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 1–11.

Yizhong Zhang, Huamin Wang, Shuai Wang, Yiying Tong, and Kun Zhou. 2012. A Deformable Surface Model for Real-Time Water Drop Animation. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (Aug. 2012), 1281–1289.

Wen Zheng, Bo Zhu, Byungmoon Kim, and Ronald Fedkiw. 2015. A new incompressibility discretization for a hybrid particle MAC grid representation with surface tension. *J. Comput. Phys.* 280, C (Jan. 2015), 96–142.

Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015b. Codimensional non-Newtonian fluids. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

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.

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

A NUMERICAL CORNER CASES

The analytical integration of Green's function G and its gradient ∇G with a linear or constant weight function over a triangle is described in the literature [Eibert and Hansen 1995; Graglia 1993]. The results are valid for evaluation points within $\mathbb{R}^3$. However, corner cases can cause numerical instabilities (e.g. due to divisions by small numbers). In this work, these instabilities are associated with the auxiliary f_{2i} function defined in Eq. (11) in Graglia [1993] and β_i defined in Eq. (13). The subscript $i \in \{1, 2, 3\}$ indicates the associated vertex indices in a triangle. The definition of f_{2i} and β_i is given by

$$f_{2i} = \ln \frac{R_i^+ + s_i^+}{R_i^- + s_i^-}\,, \tag{31}$$

$$\beta_i = \tan^{-1} \frac{t_i^0 s_i^+}{(R_i^0)^2 + |w_0| R_i^+} - \tan^{-1} \frac{t_i^0 s_i^-}{(R_i^0)^2 + |w_0| R_i^-}\,. \tag{32}$$

For the definition of the variables, we refer to Graglia [1993]. Large numerical errors can happen when $R_i^- + s_i^- = 0$ and $R_i^0 = 0$ respectively in two functions. To handle these corner cases in f_{2i}, we expand it around the singular point where $R_i + s_i \approx 0$ by using $R_i^\pm = \sqrt{(R_i^0)^2 + (s_i^\pm)^2}$:

$$f_{2i} = \begin{cases} \ln \dfrac{R_i^- \left(1 + \frac{(R_i^0)^2}{4(R_i^+)^2}\right)}{R_i^+ \left(1 + \frac{(R_i^0)^2}{4(R_i^-)^2}\right)} & \text{for } R_i^0 < (0.03\, R_i^+) \text{ and } s_i^+ < 0\,, \\ \ln \frac{R_i^+ + s_i^+}{R_i^- + s_i^-} & \text{else}\,. \end{cases} \tag{33}$$

To eliminate divisions by zero when computing β_i we add a small positive number $\varepsilon := 10^{-16}$ to the denominators in Eq. (33).