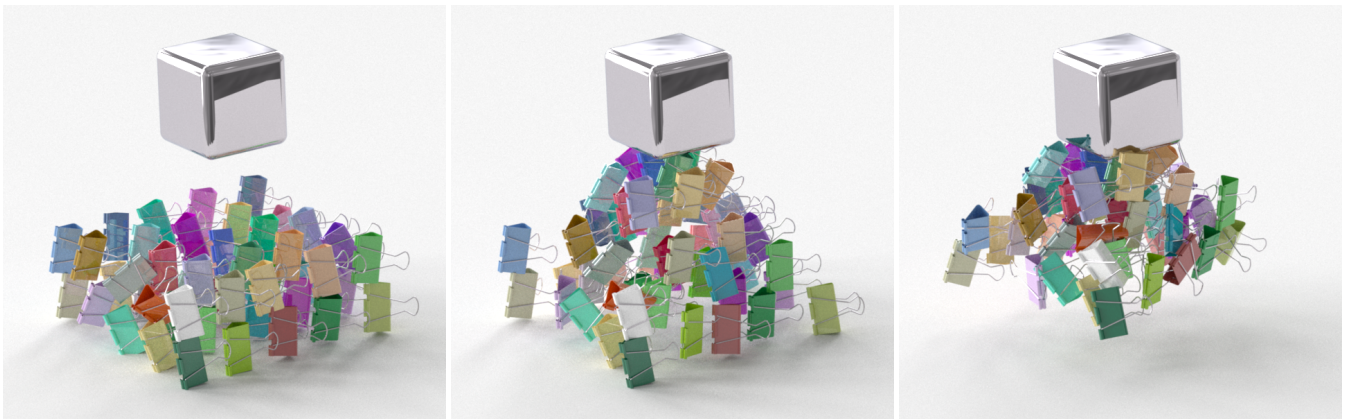# Simulation of Arbitrarily-shaped Magnetic Objects

Seung-wook Kim🆔 and JungHyun Han

Korea University

Figure 1: *A neodymium permanent magnet naturally attracts a number of binderclips, which have thin and sharp features. The magnet simulation method proposed in this paper enables arbitrarily-shaped magnetic objects to be simulated stably and efficiently.*

Abstract
We propose a novel method for simulating rigid magnets in a stable way. It is based on analytic solutions of the magnetic vector potential and flux density, which make the magnetic forces and torques calculated using them seldom diverge. Therefore, our magnet simulations remain stable even though magnets are in close proximity or penetrate each other. Thanks to the stability, our method can simulate magnets of any shapes. Another strength of our method is that the time complexities for computing the magnetic forces and torques are significantly reduced, compared to the previous methods. Our method is easily integrated with classic rigid-body simulators. The experiment results presented in this paper prove the stability and efficiency of our method.

CCS Concepts
• *Computing methodologies → Physical simulation;*

1. Introduction

Magnets are frequently used in our daily lives, including refrigerator magnets at home, push pin magnets at office, and neodymium magnets at science classes. In the field of computer graphics, magnet simulations have been earnestly investigated since the pioneering work of Thomaszewski et al. [TGPS08]. Most recently, Kim et al. [KPH18] presented rigid magnet simulations, Huang et al. [HHM19] presented magnetic fluid simulations, and Ni et al. [NZWC20] presented magnetic liquid-solid simulations.

Magnetic objects (henceforth, simply *magnets*) generate magnetic fields, and the other magnets may become magnetized by the

magnetic fields. Then, the magnetization and the magnetic fields determine magnetic forces and torques, which move the magnets.

To simulate interactions between rigid magnets, Thomaszewski et al. [TGPS08] and Kim et al. [KPH18] used *magnetic dipole moments*, which are called in short *magnetic moments*. Being point magnets, they are sampled from the magnet body volumes. Fig. 2-(a) depicts two magnets' configuration in a rigid-body simulation step. Each magnet's cross section is sampled with 9 magnetic moments. Two magnets are in close proximity and so are the magnetic moments, especially those enclosed with a dotted ellipse. Then, the magnetic forces and torques become stronger than the exact solutions, leading to unnatural simulation results. As experimentally
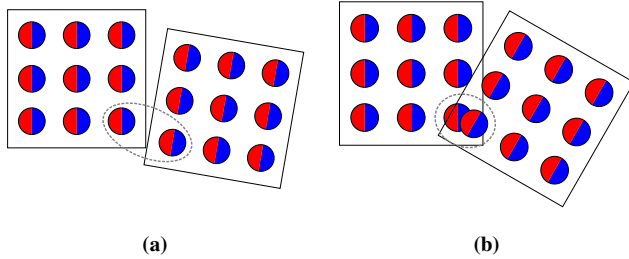
Figure 2: *Magnetic moments are regularly sampled from the rigid magnets. (a) The magnets are close to each other, and so are the magnetic moments. (b) Due to interpenetration, magnetic moments become too close, resulting in unstable rigid-body motions.*

proven by Thomaszewski et al. [TGPS08], this is due to numerical errors. A solution to this problem is to increase the number of magnetic moments. Unfortunately, this significantly increases the computational cost since volumetric evaluation over a pair of magnets is required.

The closer the magnetic moments are, the stronger the magnetic fields are. Fig. 2-(b) shows a subsequent simulation step, where the magnets penetrate each other. (Contact resolution in existing rigid body solvers cannot always completely resolve interpenetration, which is thus permitted in virtually all physics engines such as Bullet Physics [Cou15].) As the distance between magnetic moments is close to zero, the magnetic fields diverge and so do the magnetic forces and torques. Magnets frequently collide with each other due to the attractive forces, and the magnetic moment-based simulation method suffers from the *instability* problem.

Thomaszewski et al. [TGPS08] attempted to prevent the excessive forces and torques by imposing the lower limits on the distances between magnetic moments, *i.e.*, they directly controlled the magnetic forces and torques without being able to prevent excessive magnetic fields, which lead to strong magnetic moments. Such magnetic moments bring huge and wide impact on the other (even distant) magnetic moments. Kim et al. [KPH18] handled this problem by limiting the strengths of magnetic moments, still without being able to prevent excessive magnetic fields, which make the magnetic moments hard to converge.

In this paper, we propose a novel method for computing magnetic forces and torques by integrating numerically over a magnet body's *boundary*. We neither use the magnetic moments nor consider the magnet body's volume. The mathematical expressions of the forces and torques are developed upon "analytic" solutions of *magnetic vector potential* and *magnetic flux density*, which Fabbri [Fab08] proposed for uniformly magnetized objects. Because the time complexities of the analytic solutions are constant, the cost for computing the forces and torques is significantly reduced, compared to the magnetic moment-based method.

Because the analytic solutions take into account the continuity of magnetization within the magnet body, not only the magnetic vector potential and flux density but also the magnetic forces and torques calculated using them seldom diverge. Therefore, our mag-

net simulations remain mostly stable even if the magnets penetrate each other.

Another problem of the magnetic moment-based method is that it has difficulties in handling "magnets with sharp features" because not only do they require high resolutions of magnetic moments but also they are more likely to penetrate each other due to torques. To circumvent this difficulty, the magnets used in the experiments of Thomaszewski et al. [TGPS08] and Kim et al. [KPH18] are mostly limited to rounded shapes, such as rotation-invariant metal balls, because they are less likely to interpenetrate.

The theoretical and technical contributions of our method are summarized as follows:

- The magnetic forces and torques are integrated over the magnet boundaries, and therefore the time complexities of computing them are considerably reduced, which is a significant improvement over the magnetic moment-based method.
- We adopt closed-form expressions to compute the magnetic vector potential and flux density, which are then used for calculating the magnetic forces and torques. As well as the magnetic vector potential and flux density, the forces and torques seldom diverge, providing stable simulations of magnetic interactions.
- Our method can stably and efficiently handle any kinds of magnet shapes represented in polygon meshes. For example, complex shapes with thin and sharp features (such as binderclips and paperclips) can be properly simulated.

This paper is organized as follows. Section 2 reviews the related studies. Section 3 describes the magnetic moment-based method, and Section 4 presents our novel method. Section 5 proposes to combine the two methods for the sake of efficiency. Section 6 reports the experiment results, and Section 7 concludes the paper.

2. Related Work

In the field of magnetics, a number of methods for analytically calculating magnetic forces and torques have been proposed. Akoun and Yonnet [AY84] suggested a method for calculating the magnetic forces between two cubic magnets aligned in parallel. Given the same configuration, Allag and Yonnet [AY09] proposed a method that evaluates magnetic torques as well as magnetic forces. In the work of Vokoun et al. [VBHŠ09], magnetic forces for cylindrical magnets were considered. The analytic methods are accurate and fast but are difficult to generalize to various geometric shapes. In order to overcome the limitation, *i.e.*, to calculate the magnetic forces and torques between arbitrarily-shaped magnets, De Graef and Beleggia [DGB09] proposed to use 3D Fourier Transform. Unfortunately, it is necessary to use extremely high resolutions of the frequencies for the solution to be accurate.

As we conduct the boundary integral to compute magnetic forces and torques, our method is similar to boundary element method (BEM). With regard to BEM in magnetics, Koizumi et al. [KOU90] proposed the method for calculating magnetic flux density. Kuhn and Steinbach [KS02] used FEM-BEM coupling to evaluate magnetic flux density. Fabbri [Fab08] proposed a solution for calculating magnetic flux density through analytical integration over the boundary of a polyhedral magnet. However, these methods do not focus on the calculation of magnetic forces and torques.

In the computer graphics field, Thomaszewski et al. [TGPS08] proposed the first earnest work for magnetic interactions integrated with rigid-body simulations. In their work, the magnetic fields are generated by permanent magnets, then the magnetization is determined by the magnetic fields based on the linear constitutive relation, and finally the magnetization and the magnetic fields produce magnetic forces and torques. Kim et al. [KPH18] proposed magnetization dynamics to limit the strength of magnets, which stabilizes the rigid magnet simulations. However, these methods suffer from instability when magnets interpenetrate.

Several works have been reported in the domain of non-rigid magnet simulation. Using Smoothed Particle Hydrodynamics (SPH) method [MCG03], Ishikawa et al. [IYI*13] simulated magnetic fluids. They also expressed spikes at the fluid surfaces using an energy minimization scheme. Huang et al. [HHM19] simulated complex dynamics of magnetic fluids in an accurate and effective way. Their method provided unconditionally stable magnetic forces between magnetic SPH particles without a singularity of force. Ni et al. [NZWC20] proposed a level-set method based on Eulerian approach for simulating magnetic liquids and solids. Their method uses magnetic forces exerted on the material's boundary, which is different from our method that uses body forces.

There have been efforts for magnetostatic visualization. Based on molecular dynamics approach, Klein and Ertl [KE04] visualized magnetic field lines by simulating ellipsoidal-shaped particles made of ferromagnetic materials. Bachthaler et al. [BSW*12] used lines connecting magnetic moments to reveal the topological structure of the magnetic flux. Park et al. [PLH16] visualized 3D magnetic fields using view-dependent streamlines in the context of virtual science experiments. Yoon et al. [YLUH14] presented an iron filing art tool, where the user provides a silhouette of a 2D target shape and then the patterns of the iron filings are automatically generated through the principle of magnetism.

3. Equivalent Dipole Method

In order to calculate magnetic forces and torques, Thomaszewski et al. [TGPS08] and Kim et al. [KPH18] used *magnetic moments*, which are vector quantities. Every magnetic object, simply named *magnet*, has its own magnetic moments. For example, a neodymium magnet's magnetic moments are non-zero and typically large whereas an isolated metal ball's are zero or close to zero. *Magnetization* is defined as the magnetic moment density. It is also a vector quantity. Let $\mathbf{m}$ and $\mathbf{M}$ denote the magnetic moment and the magnetization, respectively.

A magnetic moment generates a *magnetic field*. The magnetic fields induce the other magnets. If a magnet can be easily induced by the magnetic fields that the other magnets generate, *i.e.*, if its magnetic moments are easily changed, it is called an *induced magnet*. In contrast, if its magnetic moments are not changed without an exceptionally strong magnetic field, it is called a *permanent magnet*. In our work, we do not consider such a strong magnetic field, and therefore the magnetic moments of a permanent magnet are assumed to remain unchanged.

A magnetic field is composed of several subfields, and the one that is related with magnetic forces and torques is *magnetic flux density*. It is denoted as $\mathbf{B}$. Given N magnetic moments, each denoted as $\mathbf{m}_i$, $\mathbf{B}$ at an arbitrary point $\mathbf{r}$ is defined as follows [Jac99]:

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \sum_i^N \left(\frac{3\mathbf{r}_i(\mathbf{r}_i \cdot \mathbf{m}_i) - (\mathbf{r}_i \cdot \mathbf{r}_i)\mathbf{m}_i}{\|\mathbf{r}_i\|^5} \right) \quad (1)$$

where μ_0 represents the permeability of the free space and $\mathbf{r}_i$ is the vector from the position of $\mathbf{m}_i$ to $\mathbf{r}$.

The magnetization, $\mathbf{M}$, of an induced magnet is determined by $\mathbf{B}$ produced by the other magnets. It is defined as follows:

$$\mathbf{M} = \frac{1}{\mu_0} \mathcal{D}^{-1} \mathbf{B} \quad (2)$$

where $\mathcal{D}$ represents the *self-demagnetizing tensor*. This tensor is determined by the shape of the magnet body and is constant. (In Section 4.3, we will discuss this in detail.)

The magnetic force, $\mathbf{F}$, and torque, $\mathbf{T}$, exerted on $\mathbf{m}$ are determined by $\mathbf{B}$. According to Ampere's law, $\mathbf{F}$ and $\mathbf{T}$ can be formulated as Taylor series expansions, where the leading terms are defined as follows:

$$\mathbf{F} = \mathbf{m} \cdot \nabla \mathbf{B} \quad (3)$$

$$\mathbf{T} = \mathbf{m} \times \mathbf{B} \quad (4)$$

Equations (3) and (4) can be rephrased using Equation (1):

$$\mathbf{F} = \frac{\mu_0}{4\pi} \sum_i^N \left[\frac{-15\mathbf{r}_i\big((\mathbf{m} \cdot \mathbf{r}_i) \cdot (\mathbf{m}_i \cdot \mathbf{r}_i)\big)}{\|\mathbf{r}_i\|^7} \right.$$
$$\left. + \frac{3\mathbf{r}_i(\mathbf{m} \cdot \mathbf{m}_i) + 3\big(\mathbf{m}(\mathbf{m}_i \cdot \mathbf{r}_i) + \mathbf{m}_i(\mathbf{m} \cdot \mathbf{r}_i)\big)}{\|\mathbf{r}_i\|^4} \right] \quad (5)$$

$$\mathbf{T} = \frac{\mu_0}{4\pi} \sum_i^N \left[\frac{3(\mathbf{m} \times \mathbf{r}_i)(\mathbf{m}_i \cdot \mathbf{r}_i)}{\|\mathbf{r}_i\|^5} - \frac{\mathbf{m} \times \mathbf{m}_i}{\|\mathbf{r}_i\|^3} \right] \quad (6)$$

This kind of method, which uses the magnetic moments to compute the physical quantities such as $\mathbf{F}$ and $\mathbf{T}$, is generally called *Equivalent Dipole Method* (EDM). Unfortunately, EDM suffers from serious problems. Above all, Equations (5) and (6) reveal that, when the magnetic moments are in close proximity, *i.e.*, when $\mathbf{r}_i$ is small, $\mathbf{F}$ and $\mathbf{T}$ become excessively strong. They are not even defined when the magnetic moments perfectly overlap due to interpenetration of magnets.

If magnets have sharp features, they are more likely to penetrate each other due to torques. As a result, magnetic moments are also more likely to become close, leading to unstable simulations. If we use a higher resolution of magnetic moments, the instability problem is aggravated because the magnetic moments are more likely to be in close proximity.

4. Magnetic Boundary Method

This section presents our method, which overcomes the drawbacks of EDM. First of all, Section 4.1 defines the *magnetic flux density* and *magnetic vector potential*. Then, Section 4.2 presents how to compute the magnetic forces, magnetic torques, and mechanical torques using them. Finally, Section 4.3 discusses how to implement magnetic induction.

4.1. Magnetic Vector Potential and Flux Density

Given a magnet with magnetization $\mathbf{M}$, Jackson [Jac99] formulated the magnetic vector potential, denoted as $\mathbf{A}$, at an arbitrary point $\mathbf{r}$:

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_V \mathbf{M}(\mathbf{r}') \times \frac{(\mathbf{r} - \mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|^3} \, d^3 r' \tag{7}$$

where $\mathbf{r}'$ represents a point within the magnet volume, V. Assuming that $\mathbf{M}$ is uniform across V, Fabbri [Fab08] proposed an analytic solution for $\mathbf{A}$. First, by applying an alternate form of Gauss' theorem introduced in Arfken and Weber [AW05], Equation (7) is expressed in the form of integral over the magnet's surface, ∂V:

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi} \mathbf{M} \times \oint_{\partial V} \frac{\mathbf{n}(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} \, d^2 r' \tag{8}$$

where $\mathbf{n}(\mathbf{r}')$ represents the surface normal at $\mathbf{r}'$.

For a polyhedral object, Equation (8) can be transformed into the sum of the integrals over all polygons:

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi} \mathbf{M} \times \sum_{p \in \partial V} W_p(\mathbf{r}) \mathbf{n}_p \tag{9}$$

where p represents a polygon, $\mathbf{n}_p$ is its normal, and $W_p(\mathbf{r})$ is expressed as follows:

$$W_p(\mathbf{r}) = \int_p \frac{d^2 r'}{\|\mathbf{r} - \mathbf{r}'\|} \tag{10}$$

The magnetic vector potential, $\mathbf{A}$, and the magnetic flux density, $\mathbf{B}$, are related via the curl operator: $\mathbf{B} = \nabla \times \mathbf{A}$. When ψ and $\mathbf{v}$ respectively represent a scalar and a vector, $\nabla \times (\psi \mathbf{v}) = \psi (\nabla \times \mathbf{v}) + \nabla \psi \times \mathbf{v}$. As $\mathbf{M}$ is uniform across the magnet volume, $\mathbf{B}$ can be obtained by applying the curl operator to Equation (9):

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \sum_{p \in \partial V} \nabla W_p(\mathbf{r}) \times (\mathbf{M} \times \mathbf{n}_p) \tag{11}$$

As detailed in **Appendix A**, $W_p(\mathbf{r})$ and $\nabla W_p(\mathbf{r})$ can be reformulated and expressed in the following closed forms:

$$W_p(\mathbf{r}) = \sum_{e \in \partial p} \omega_e(\mathbf{r}) \big(\mathbf{n}_p \times (\mathbf{r}_e - \mathbf{r}) \cdot \mathbf{u}_e \big) - \Omega_p(\mathbf{r})(\mathbf{r}_p - \mathbf{r}) \cdot \mathbf{n}_p \tag{12}$$

$$\nabla W_p(\mathbf{r}) = \sum_{e \in \partial p} \omega_e(\mathbf{r})(\mathbf{n}_p \times \mathbf{u}_e) + \Omega_p(\mathbf{r}) \mathbf{n}_p \tag{13}$$

The elements of Equations (12) and (13) are described as follows: (i) e represents an edge of p's boundary, ∂p. (ii) $\mathbf{r}_e$ and $\mathbf{r}_p$ are arbitrary points on e and p, respectively. (iii) $\mathbf{u}_e$ is the unit vector parallel to e. It always points counter-clockwise with respect to p. (iv) $\omega_e(\mathbf{r})$ is expressed as follows:

$$\omega_e(\mathbf{r}) = \ln \frac{\|\mathbf{r}_1 - \mathbf{r}\| + \|\mathbf{r}_2 - \mathbf{r}\| + \|\mathbf{r}_1 - \mathbf{r}_2\|}{\|\mathbf{r}_1 - \mathbf{r}\| + \|\mathbf{r}_2 - \mathbf{r}\| - \|\mathbf{r}_1 - \mathbf{r}_2\|} \tag{14}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the end points of e. (v) We decompose p into a set of triangles. Let t represent a triangle of p, and $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ represent three vertices of t. Then, $\Omega_p(\mathbf{r})$, which represents the

(a)   **(b)**
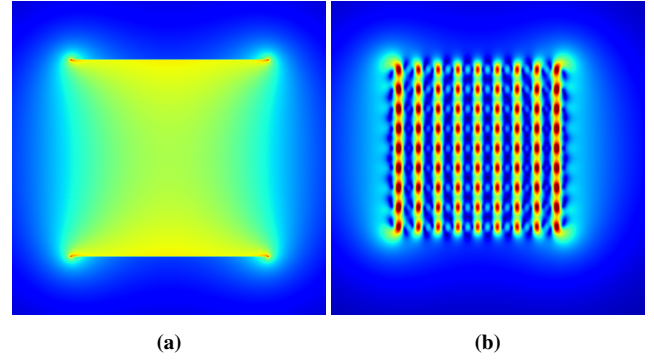
Figure 3: *Visualization of $\mathbf{B}$'s magnitudes over a cross section of a cubic magnet, which is uniformly magnetized along the upper direction. The magnitudes of $\mathbf{B}$ are in rainbow colors, **VIBGYOR**, where Violet is the weakest and **R**ed is the strongest. (a) $\mathbf{B}$ is computed using Equation (11). (b) $\mathbf{B}$ is computed using Equation (1).*

signed solid angle with respect to p, is defined as follows [VOS83]:

$$\Omega_p(\mathbf{r}) = \sum_{t \in p} \Omega_t(\mathbf{r})$$
$$= \sum_{t \in p} 2 \arctan \frac{(\mathbf{r}_1 - \mathbf{r}) \cdot (\mathbf{r}_2 - \mathbf{r}) \times (\mathbf{r}_3 - \mathbf{r})}{D(\mathbf{r})} \tag{15}$$
$$D(\mathbf{r}) = \|\mathbf{r}_1 - \mathbf{r}\| \|\mathbf{r}_2 - \mathbf{r}\| \|\mathbf{r}_3 - \mathbf{r}\| +$$
$$\|\mathbf{r}_1 - \mathbf{r}\| (\mathbf{r}_2 - \mathbf{r}) \cdot (\mathbf{r}_3 - \mathbf{r}) +$$
$$\|\mathbf{r}_2 - \mathbf{r}\| (\mathbf{r}_3 - \mathbf{r}) \cdot (\mathbf{r}_1 - \mathbf{r}) +$$
$$\|\mathbf{r}_3 - \mathbf{r}\| (\mathbf{r}_1 - \mathbf{r}) \cdot (\mathbf{r}_2 - \mathbf{r})$$

Readers are referred to Fabbri's work [Fab08] for the complete procedures of deriving the aforementioned equations.

Suppose that $\mathbf{B}$'s closed form in Equation (11) is computed for a cubic magnet. Fig. 3-(a) visualizes the result in the magnet's cross section. Now suppose that the cubic magnet is sampled by $9 \times 9 \times 9$ magnetic moments and EDM computes $\mathbf{B}$ using Equation (1). Fig. 3-(b) visualizes the result. Unlike the groundtruth shown in Fig. 3-(a), Fig. 3-(b) reveals excessively strong $\mathbf{B}$ near the magnetic moments. Consequently, $\mathbf{F}$ (in Equation (3)) and $\mathbf{T}$ (in Equation (4)) also become excessive, causing unstable magnetic interactions.

4.2. Magnetic Force and Torque

This subsection presents how to calculate the magnetic force and torque using the magnetic vector potential, $\mathbf{A}$, and the magnetic flux density, $\mathbf{B}$, defined above. The magnetic force, $\mathbf{F}$, exerted on a

magnet is derived as follows:

$$\mathbf{F} = \int_V \mathbf{f}(\mathbf{r})\, d^3 r$$

$$= \int_V \mathbf{M} \cdot \nabla \mathbf{B}(\mathbf{r})\, d^3 r$$

$$= \oint_{\partial V} \big(\mathbf{M} \cdot \mathbf{B}(\mathbf{r})\big) \mathbf{n}(\mathbf{r})\, d^2 r$$

$$= \sum_{p \in \partial V} \left(\mathbf{M} \cdot \int_p \mathbf{B}(\mathbf{r})\, d^2 r \right) \mathbf{n}_p \qquad (16)$$

where the second line is obtained using Equation (3), the third line is obtained through the alternate form of Gauss' theorem used to derive Equation (8), and the last line is derived as the force is integrated over the polyhedral magnet's surface.

The magnetic torque, **T**, is derived in a similar fashion:

$$\mathbf{T} = \int_V \mathbf{t}(\mathbf{r})\, d^3 r$$

$$= \int_V \big(\mathbf{M} \times \mathbf{B}(\mathbf{r})\big)\, d^3 r$$

$$= \mathbf{M} \times \int_V \mathbf{B}(\mathbf{r})\, d^3 r$$

$$= \mathbf{M} \times \int_V \big(\nabla \times \mathbf{A}(\mathbf{r})\big)\, d^3 r$$

$$= \mathbf{M} \times \oint_{\partial V} \mathbf{n}(\mathbf{r}) \times \mathbf{A}(\mathbf{r})\, d^2 r$$

$$= \mathbf{M} \times \sum_{p \in \partial V} \left(\mathbf{n}_p \times \int_p \mathbf{A}(\mathbf{r})\, d^2 r \right) \qquad (17)$$

where the second line is obtained using Equation (4), the fourth line is obtained using the definition of $\mathbf{B} = \nabla \times \mathbf{A}$, the fifth line is obtained using another alternate form of Gauss' theorem introduced in Arfken and Weber [AW05], and the last line is derived as the torque is integrated over the polyhedral magnet's surface.

The mechanical torque, $\mathcal{T}$, is also similarly derived:

$$\mathcal{T} = \int_V \tau(\mathbf{r})\, d^3 r$$

$$= \int_V (\mathbf{r} - \mathbf{r}_c) \times \mathbf{f}(\mathbf{r})\, d^3 r$$

$$= \int_V (\mathbf{r} - \mathbf{r}_c) \times \big(\nabla(\mathbf{M} \cdot \mathbf{B}(\mathbf{r}))\big)\, d^3 r$$

$$= -\int_V \nabla \times \big((\mathbf{M} \cdot \mathbf{B}(\mathbf{r}))\,(\mathbf{r} - \mathbf{r}_c)\big)\, d^3 r$$

$$= -\oint_{\partial V} \mathbf{n}(\mathbf{r}) \times \big((\mathbf{M} \cdot \mathbf{B}(\mathbf{r}))\,(\mathbf{r} - \mathbf{r}_c)\big)\, d^2 r$$

$$= -\sum_{p \in \partial V} \mathbf{n}_p \times \left(\int_p (\mathbf{M} \cdot \mathbf{B}(\mathbf{r}))\,(\mathbf{r} - \mathbf{r}_c)\, d^2 r \right) \qquad (18)$$

where $\mathbf{r}_c$ represents the center of mass of the magnet, the third line is obtained using Equation (3), the fourth line is obtained using the vector identity, $\nabla \times (\psi \mathbf{v}) = \psi(\nabla \times \mathbf{v}) + \nabla \psi \times \mathbf{v}$, the fifth line is obtained through the alternate form of Gauss' theorem used for Equation (17), and the last line is derived as the torque is integrated over the polyhedral magnet's surface.

Our method deals with magnets represented in polygon meshes. **A** and **B** are defined over the magnet's boundary. Using **A** and **B**, **F**, **T** and $\mathcal{T}$ are integrated over the boundary. In these senses, we name our method *Magnetic Boundary Method* (MBM).

Equations (9) and (11) (for **A** and **B**, respectively) do not require sampling and are computed at $O(1)$ time. Note that, in Equations (16) through (18), the degrees of the integrals are all reduced from three to two. Due to difficulty in transforming the boundary integrals into closed-form expressions, they are calculated through numerical integration by sampling evaluation points on the boundary.

4.3. Magnetic Induction

Kim et al. [KPH18] proposed *magnetization dynamics*, where a magnet is decomposed into magnetic cells, two submagnetizations are computed in each cell, and their average is taken as the cell's magnetization. For computing submagnetizations, they adopt what is called *effective magnetic field*, $\mathbf{H}_{\text{eff}}$. Its simple form is expressed as follows:

$$\mathbf{H}_{\text{eff}} = \frac{1}{\mu_0} \mathbf{B} - \mathcal{D} \mathbf{M} \qquad (19)$$

where $\mathcal{D}$ is the self-demagnetizing tensor introduced in Equation (2). Recall that $\mathcal{D}$ depends on the magnet body's shape. Kim et al. [KPH18] assume spherical cells such that $\mathcal{D}$ is analytically defined to be $\frac{1}{3}\mathbf{I}$, where **I** is the identity matrix. Given $\mathbf{H}_{\text{eff}}$ evaluated at each cell's center, the Landau-Lifshitz-Gilbert (LLG) equation [Gil55] is used to simulate each submagnetization's dynamics:

$$\frac{d\mathbf{M}_i}{dt} = -\frac{\gamma}{1 + \alpha^2} \Big(\mathbf{M}_i \times \big(\mathbf{H}_{\text{eff}} + \alpha(\mathbf{M}_i \times \mathbf{H}_{\text{eff}})\big) \Big) \qquad (20)$$

where $\mathbf{M}_i$ represents a submagnetization, γ is the gyromagnetic ratio, and α is the material-specific damping constant.

In order to implement magnetic induction, we adopt the magnetization dynamics. Unlike Kim et al. [KPH18], however, we do not decompose a magnet into cells but assign two submagnetizations to the entire body of a magnet. Our MBM evaluates Equation (19) at the magnet's center of mass and then uses Equation (20) to compute submagnetizations. Note that the magnet may have an arbitrary shape. Then, how do we define $\mathcal{D}$ for the magnet? The answer is surprisingly simple.

In Kim et al. [KPH18], $\mathcal{D}$ is set to $\frac{1}{3}\mathbf{I}$, and then **M** is computed via the LLG equation. As proven by Kim et al., the LLG equation converges over time such that the resulting **M** satisfies Equation (2), *i.e.*, $\mathbf{M} = \frac{3}{\mu_0}\mathbf{B}$ for every cell. This shows that $\mathcal{D}$ works simply as a "control parameter" that relates **B** and **M**.

In MBM, the magnet that is assigned two submagnetizations

Algorithm 1 Magnet simulation

1: **for all** pairs of magnets **do**
2:    Compute *distance* between the magnets
3: **repeat**
4:    **for all** pairs of magnets **do**
5:       **if** *distance* < *threshold* **then**
6:          Compute **B** using Equation (11)
7:       **else**
8:          Compute **B** using Equation (1)
9:       **for all** magnets **do**
10:         Compute $\mathbf{H}_{\text{eff}}$ using Equation (19)
11:         Update magnetization via Equation (20)
12: **until** all magnetizations converge
13: **for all** pairs of magnets **do**
14:    **if** *distance* < *threshold* **then**
15:       Compute **A** using Equation (9)
16:       Compute **B** using Equation (11)
17:       Compute **F** using Equation (16)
18:       Compute **T** using Equation (17)
19:       Compute $\mathcal{T}$ using Equation (18)
20:    **else**
21:       Sample magnetic moments from the magnet
22:       Compute **F** using Equation (3)
23:       Compute **T** using Equation (4)
24:       Compute $\mathcal{T}$ in a classic manner

---

may have an arbitrary shape. If appropriately adjusted regardless of the shape, however, $\mathcal{D}$ can play the same role of the control parameter that relates **B** and **M**. In the current implementation, $\mathcal{D}$ is heuristically set to $\frac{1}{k}\mathbf{I}$, where k is a user-defined value. For example, k can be made larger to express stronger magnetic induction, *i.e.*, larger **M**.

5. Implementation

As discussed in Section 1, the magnetic forces and torques in EDM become overly stronger (than the exact solutions) if two magnets are in close proximity. If they are sufficiently far apart, however, EDM can calculate the magnetic forces and torques with a very small number of magnetic moments, and the calculated forces and torques are almost the same as MBM's, as will be shown in Section 6.1. This implies that the overall performance of EDM excels that of MBM *if* two magnets are not close. Therefore, our implementation chooses between MBM and EDM based on the distance between two magnets.

The distance used to choose between MBM and EDM is measured between two magnets' bounding spheres, whose centers are their centers of mass. If it is greater than the bounding sphere radius, EDM is chosen. Otherwise MBM is chosen.

Algorithm 1 shows our magnet simulation steps. We first compute the distance between two magnets' bounding spheres, whose centers are their centers of mass. If it is greater than the bounding sphere radius, i.e., if two magnets are close to each other, we compute the magnetic flux density, **B**, with MBM, which is more accu-

rate and robust than EDM. Otherwise, we compute **B** with EDM, which shows comparable accuracy and higher efficiency when the magnets are sufficiently far apart.

Algorithm 1 shows our magnet simulation steps. We first compute the distances between the magnets. If two magnets are close to each other, we compute the magnetic flux density, **B**, with MBM, which is more accurate and robust than EDM. Otherwise, we compute **B** with EDM, which shows comparable accuracy and higher efficiency when the magnets are sufficiently far apart. Then, the effective magnetic field, $\mathbf{H}_{\text{eff}}$, is computed using **B**, and all magnetizations are updated. This process is repeated until all magnetizations converge. Finally, we compute the magnetic forces (**F**), magnetic torques (**T**) and mechanical torques ($\mathcal{T}$) for every pair of magnets. For this, we also choose between MBM and EDM based on the distance between the magnets. (The distance used to choose between MBM and EDM is measured between two magnets' bounding spheres, whose centers are their centers of mass. If it is greater than the bounding sphere radius, EDM is chosen. Otherwise MBM is chosen.)

For a step of rigid-body simulation, **Algorithm 1** is first run to determine **F**, **T** and $\mathcal{T}$ for each magnet. Then, they are used as the external forces and torques exerted on the magnet.

Let us discuss more details on MBM implementation. When a magnet is represented in a closed polygon mesh, its edge is shared by two polygons. Computing **A** (in Equation (9)) and **B** (in Equation (11)) requires $W_p(\mathbf{r})$ (in Equation (12)) and $\nabla W_p(\mathbf{r})$ (in Equation (13)) to be evaluated. The summation terms in $W_p(\mathbf{r})$ and $\nabla W_p(\mathbf{r})$ are defined with respect to edges ($e \in \partial p$) and contain $\omega_e(\mathbf{r})$ (defined in Equation (14)). Therefore, $\omega_e(\mathbf{r})$ is evaluated twice when **A** is computed. It is also the case when **B** is computed. Since it is time-consuming to compute a logarithmic function in Equation (14), $\omega_e(\mathbf{r})$ evaluated for one polygon is stored and then reused for the other polygon.

Note that $\omega_e(\mathbf{r})$ may diverge if **r** gets very close to the edge. If **r** lies on the edge, $\omega_e(\mathbf{r})$ becomes singular. In these cases, we smooth $\omega_e(\mathbf{r})$ by adding a small positive value, ε, to the denominator of Equation (14). In reality, however, these cases seldom happen. In our experiments, smoothing with ε was not done at all.

6. Experiment Results and Discussions

Our experiments are made with Intel Core i7-8700 3.20 GHz CPU and 32 GB memory. The rigid-body simulation is based on the open-source library, Bullet Physics [Cou15], and all scenes are rendered with Mitsuba [Jak10].

6.1. Mathematical Evaluation

Consider two cubic magnets shown in Fig. 4-(a), each with the dimensions of $1\,\text{cm}^3$ and the magnetic flux density of 1 tesla. Given the distance, d, between their opposing faces, the magnetic forces are calculated by EDM and MBM respectively. For EDM, the algorithm of Thomaszewski et al. [TGPS08] is implemented, and each cube is regularly sampled by n^3 magnetic moments. We call n the *sampling rate*. In MBM, each of six faces of a magnet is regularly sampled by n^2 points. Thanks to Newton's third law of motion, the

(a)

(b)

(c)

(d)
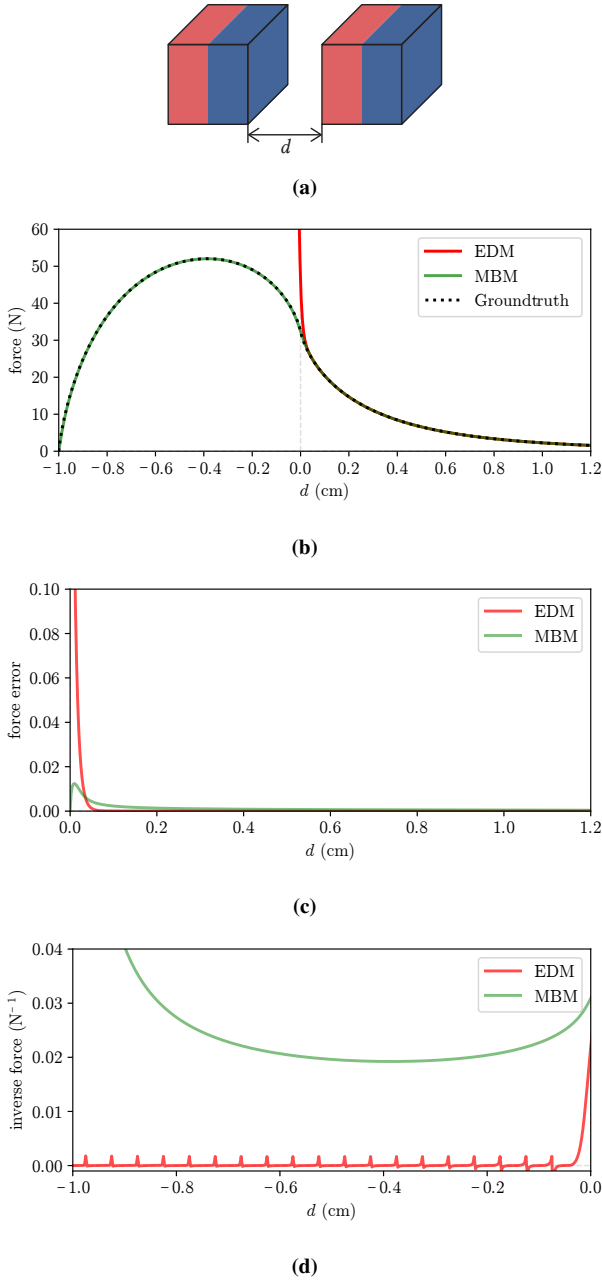
Figure 4: *Experiments with two aligned magnets. (a) The cubic magnets are separated by d. (b) The forces computed using EDM [TGPS08] and our method, MBM. The MBM force becomes zero (due to its symmetry) when two magnets perfectly overlap, i.e., when d = −1.0. (c) The errors incurred by EDM and MBM. (d) The inverse forces.*

force, **F**, computed for one magnet is negated to define −**F** exerted on the other magnet. In Fig. 4-(a), n is set to 20 for both EDM and MBM. Fig. 4-(b) compares the magnetic forces computed by EDM (in red) and those by MBM (in green).

Given the special configuration shown in Fig. 4-(a), where two cubic magnets are aligned in parallel, an analytic solution for the magnetic force is defined [AY84], and it is mathematically defined even when the magnets interpenetrate. In Fig. 4-(b), the groundtruth is depicted in dotted black. The MBM force curve almost completely overlaps with the groundtruth.

Fig. 4-(b) shows that the EDM forces are almost identical to the MBM forces when d is positive and large. When d is close to zero, however, the EDM forces are excessively strong. Fig. 5-(c) depicts the errors of the EDM and MBM forces against the groundtruth when $d \geq 0$.

Fig. 4-(d) shows the inverse forces when $d < 0$, *i.e.*, when two magnets penetrate each other. The EDM forces diverge due to close proximity of magnetic moments and the forces are not even defined at their singularities.

Fig. 5-(a) depicts how the sampling rate, n, affects the magnetic forces in MBM. It is observed that the MBM forces do not heavily depend on n, *i.e.*, the MBM forces can be computed properly with small n. Shown on the left in Fig. 5-(b) is the closeup view of the MBM force curves within the dotted rectangle in Fig. 5-(a). Compare the magnitudes of two forces computed when $d = 0.02$: $30.33N$ (when $n = 2$) vs. $28.68N$ (when $n = 20$). The former is 5.8% greater than the latter.

Fig. 5-(c) depicts the EDM forces. The closeup view shown on the right of Fig. 5-(b) reveals that the force computed when $n = 2$ is 25.4% greater than that computed when $n = 20$. EDM is more sensitive to the sampling rate than MBM. In order to obtain accurate forces in EDM, it is required to increase the sampling rate, *i.e.*, to use a larger number of magnetic moments. (Note however that it helps only when $d \geq 0$. It does not help if $d < 0$. As the sampling rate is increased, the magnetic moments are more likely to be close to each other, aggravating the instability problem.)
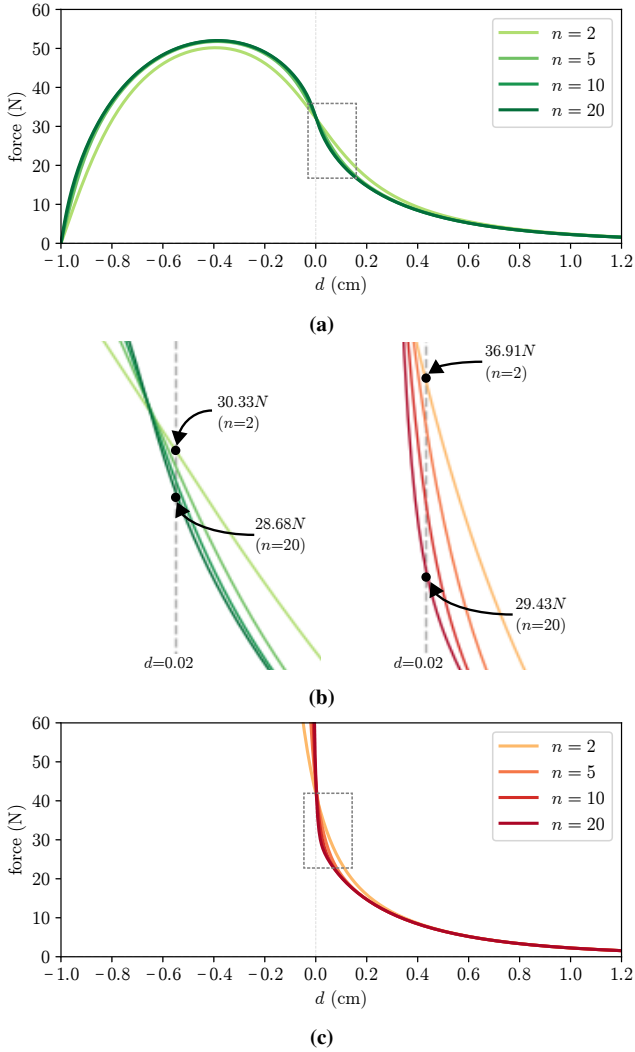
As presented in **Algorithm 1** of Section 5, our implementation chooses between MBM and EDM based on the distance between two magnets. In Fig. 5-(c), observe the EDM curves where d is large. The forces are not sensitive to the sampling rate and are accurately calculated even with a very small number of magnetic moments. This validates our implementation strategy of choosing EDM if the distance between magnets is large enough.

Recall that, to compute magnetic forces, the evaluation on the boundary of one magnet is required in MBM while the volumetric evaluation between two magnets is required in EDM. In our experiments, both EDM and MBM compute forces using a single thread. In the experiment results reported in Fig. 4, where $n = 20$, it takes about 4385 milliseconds to compute the magnetic forces in EDM whereas it takes about 7 milliseconds in MBM.

6.2. Simulation

This subsection compares the simulation results obtained by our implementation (which combines EDM and MBM) and the work of Kim et al. [KPH18], which is the state of the art in EDM. All experiment results can be also found in the attached video.

Kim et al.'s work is implemented to control the sampling

Figure 5: *Magnetic forces computed with various sampling rates. (a) The MBM forces are computed with n^2 points sampled from each square face. (b) The closeup views of MBM and EDM force curves around where $d = 0.02$. (c) The EDM forces are computed with n^3 magnetic moments sampled from each cube.*

Table 1: *Performance data of MBM in Fig. 6. T_m represents the elapsed time (in milliseconds) on magnet simulation, where the time step size of the rigid-body simulation is set to 10 milliseconds.*

n	# of samples in total	T_m
2	24	0.12
5	150	0.36
10	600	0.89
20	2400	3.32

out suffering from instability and properly pushes the blue magnet outwards.

In Fig. 6-(d), the sampling rate n is set to 20. However, the cubic magnets can be naturally simulated with significantly decreased n, as can be found in the attached video. Table 1 reports the MBM performances measured with different sampling rates. When $n = 10$, for example, 10^2 points are regularly sampled from each of the cubic magnet's six faces and therefore the total sample count becomes 600 ($10^2 \times 6$). The elapsed time on magnet simulation, T_m, increases in general linearly with the sample count.
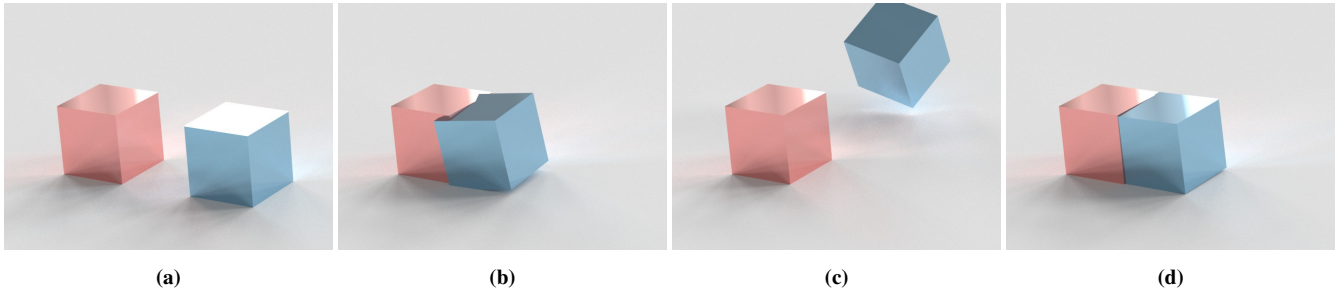
Fig. 7-(a) shows the initial state of a group of permanent magnets, which are uniformly separated in the air. The magnets are very strong to be stacked in the air, as shown in Fig. 7-(b), and penetrate each other. Fig. 7-(c) shows the unstable simulation result generated by EDM. In contrast, MBM produces stable motions of the stacked magnets, as shown in Fig. 7-(d), where the magnets have landed on the ground.

Fig. 8-(a) shows the initial state of a bar magnet (a permanent magnet) and paperclips (induced magnets). The thin and sharp paperclips are likely to penetrate each other (due to their torques) as well as the bar magnet. When the magnetic moments are very close to each other, EDM generates excessive magnetic fields. Consequently, the paperclips' magnetizations are hard to converge, and excessive magnetic forces and torques are generated. Fig. 8-(b) shows the resulting unstable simulation. In contrast, the magnetic fields generated by MBM are bounded enough for magnetizations to converge. Consequently, the magnetic forces and torques do not diverge. See our implementation's result in Fig. 8-(c).

Fig. 9 shows that a huge crane (a permanent magnet) attracts metal junks (induced magnets) of complex shapes. Despite the wide spectrum of the shape complexities, the metal junks are attracted to the crane and are also stuck to each other stably. (Fig. 1 already showed that the binderclips with thin, sharp and complex shape features are stably simulated.)

The performances measured in all experiments are listed in Table 2, where T_m is the elapsed time on magnet simulation, and T_r is that on rigid-body simulation with the time step size Δt. **Ours** indicates our implementation that combines EDM and MBM whereas **EDM** means the implementation of Kim et al. [KPH18]. The sample counts are also listed in the table. For example, "Fig. 6 (**EDM**)" uses 2000 samples as each of two cubes is sampled by 10^3 magnetic moments. Unlike all magnets in Fig. 6 and Fig. 7, the induced magnets in Fig. 1, Fig. 8 and Fig. 9 are irregularly sampled because it is impossible to sample their arbitrary shapes regularly.

rate based on the distance between magnets. As the distance increases, the sampling rate is decreased. When "our implementation" chooses EDM (between EDM and MBM) at a certain distance, its sampling rate is made identical to that of Kim et al.'s implementation used at the same distance. This is for fair comparisons of our and Kim et al.'s implementations.

Fig. 6-(a) shows the configuration presented in Fig. 4-(a), but the red magnet is fixed on the ground to attract the blue one. Initially, they are separated by 1*cm*. The magnets are very strong, and the attracted blue magnet penetrates the red one, as shown in Fig. 6-(b). Fig. 6-(c) shows the result of EDM. The instability problem is caused by excessive magnetic forces and torques. In contrast, Fig. 6-(d) shows that our MBM handles the penetration state with-

Figure 6: *Two strong permanent magnets. (a) Initially, the distance d is set to one. (b) The magnets are very strong to penetrate each other. (c) In EDM, the magnetic forces diverge, producing unstable simulation results. (d) Our MBM produces proper magnetic forces to handle the penetration stably.*
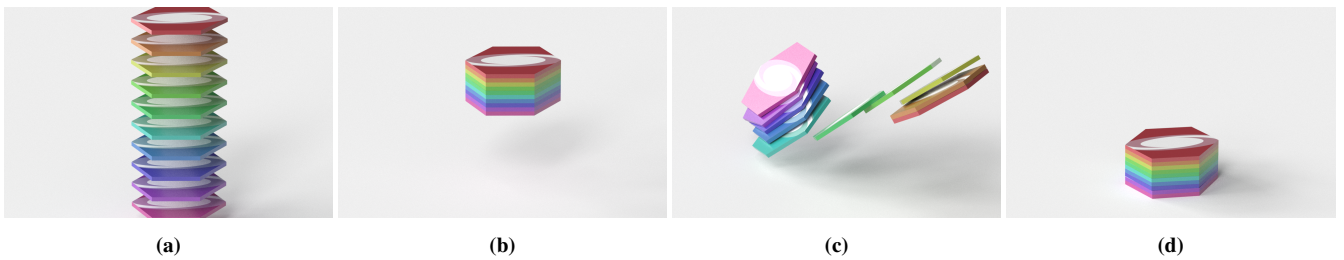
Figure 7: *Stacked permanent magnets. (a) Initial state. (b) An intermediate simulation result: The magnets are strong enough to be stacked in the air. (c) In EDM, the excessively strong forces and torques cannot be prevented. (d) MBM produces stable simulation results.*
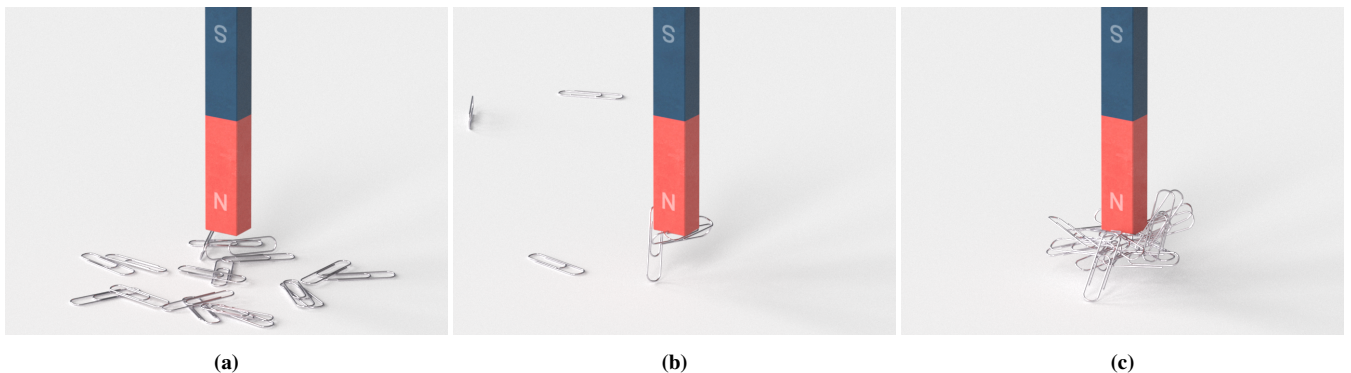
Figure 8: *Paperclips are magnetized and attracted by the bar magnet. The paperclips also attract each other. (a) Initial state. (b) In EDM, the excessive forces and torques cannot be prevented. (c) MBM produces stable simulation results.*

As presented earlier with Table 1 and the attached video, **Ours** can simulate naturally the cubic magnets in Fig. 6 with small n, e.g., even when $n = 2$. "Fig. 6 (**Ours**)" in Table 2 reports the performance measured when $n = 10$. Even though the sampling rate is unnecessarily high, T_m consumed by **Ours** (0.8 milliseconds) is about 29 times smaller than that consumed by **EDM** (23 milliseconds). It is mainly due to the time complexity differences: $O(n^2)$ in MBM vs. $O(n^6)$ in EDM. Similar discussions can be made between "Fig. 7 (**EDM**)" and "Fig. 7 (**Ours**)." T_m of **Ours** is 93 times smaller than that of **EDM**.

It is worth noting that a considerably large number of samples is necessary for "Fig. 8 (**Ours**)." If the sample count is overly de-

creased, numerical errors may be accumulated due to the irregular normals of the paperclips' high-frequency surfaces. (Our work's limitation related to this issue will be discussed in Section 7.) Despite the large sample count, however, T_m of "Fig. 8 (**Ours**)" is about 57 times smaller than that of "Fig. 8 (**EDM**)." Similar discussions can be made between "Fig. 1 (**EDM**)" and "Fig. 1 (**Ours**)."

7. Conclusions and Future Work

We presented a novel method for stable simulations of rigid magnets. Our method adopts analytic solutions for the magnets' magnetic vector potentials and flux densities. Because the solutions are bounded, the magnetic forces, magnetic torques and mechanical

(a)   **(b)**   **(c)**
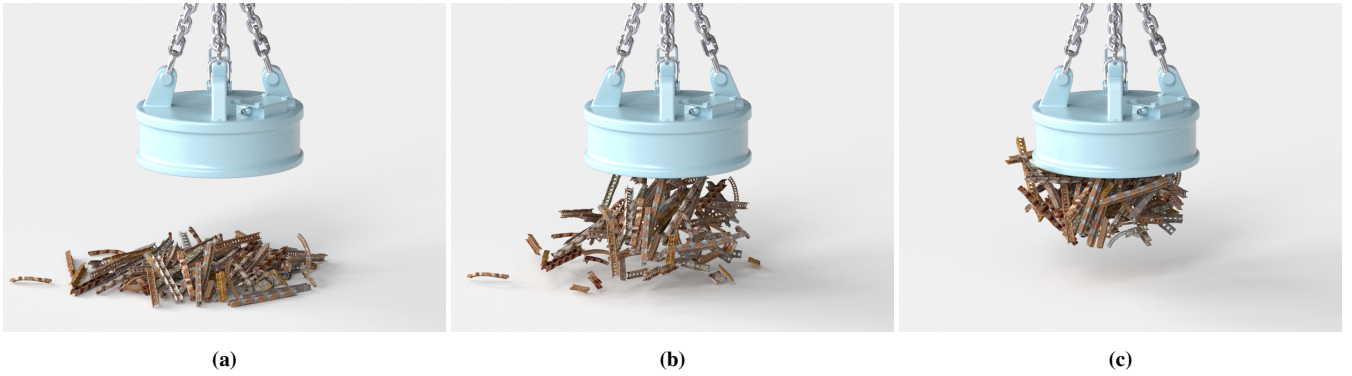
Figure 9: *The huge crane magnet pulls up the metal junks. The thin, sharp and complex materials are stably simulated by our implementation.*

Table 2: *Performance data. T_m, T_r and Δt are all in milliseconds.*

Scene	# of samples in total	T_m	T_r	Δt
Fig. 1 (**EDM**)	29696	1961	21.8	1
Fig. 1 (**Ours**)	660595	242	22.1	1
Fig. 6 (**EDM**)	2000 ($n = 10$)	23	1.3	10
Fig. 6 (**Ours**)	600 ($n = 10$)	0.8	1.4	10
Fig. 7 (**EDM**)	31500	1302	6.0	1
Fig. 7 (**Ours**)	14400	14	6.0	1
Fig. 8 (**EDM**)	24435	510	7.0	1
Fig. 8 (**Ours**)	21312	9	6.9	1
Fig. 9 (**Ours**)	768000	564	37.1	1

torques calculated using them are also bounded. Consequently, the resulting simulations remain stable and are robust to interpenetration of the magnets. In addition, our method integrates the forces and torques over the magnet boundaries, and it is a significant improvement with respect to the time complexity compared to the existing methods.

A few possible improvements over the current implementation of our method are listed as follows. Our method assumes that magnetization is uniform across the body of a magnet. If necessary, however, a magnet can be decomposed into multiple shapes such that a distinct magnetization can be computed for each shape. On the other hand, our current implementation uses the distance between the magnets' bounding spheres when choosing between EDM and MBM. This can be replaced, for example, by the GJK distance algorithm [GJK88] to increase the accuracy.

Our method has also limitations. The computation times for the magnetic vector potential and flux density are proportional to the edge count of the magnet's polygon mesh for a polyhedral magnet. Given e edges, their time complexities are $O(e)$. In order to apply our method to real-time applications, it may often be necessary to decrease the edge count, *i.e.*, the polygon mesh resolution should be decreased. This level-of-detail approach needs to be carefully designed so as not to compromise the simulation accuracy.

As discussed in Section 6.2, it is safe for our method to use a small number of samples to simulate magnets of low-frequency surfaces, but a considerably large number of samples is necessary for stably simulating magnets of high-frequency surfaces. Fortunately, the large sample count does not seriously degrade the performances, as shown in Table 2. Unfortunately, the appropriate sampling count should be determined by trial and error. For our method to be applied to a wide range of applications, an auto-setting technique needs to be developed to determine the optimal sample count. These limitations make up our future work.

Acknowledgements

**References**

[AW05] ARFKEN G., WEBER H.: *Mathematical methods for physicists*. Elsevier, 2005. 4, 5

[AY84] AKOUN G., YONNET J.-P.: 3d analytical calculation of the forces exerted between two cuboidal magnets. *IEEE Transactions on magnetics 20*, 5 (1984), 1962–1964. 2, 7

[AY09] ALLAG H., YONNET J.: 3-d analytical calculation of the torque and force exerted between two cuboidal magnets. *IEEE Transactions on Magnetics 45*, 10 (2009), 3969–3972. 2

[BSW*12] BACHTHALER S., SADLO F., WEEBER R., KANTOROVICH S., HOLM C., WEISKOPF D.: Magnetic flux topology of 2d point dipoles. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 955–964. 3

[Cou15] COUMANS E.: Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses* (2015), ACM, p. 7. 2, 6

[DGB09] DE GRAEF M., BELEGGIA M.: General magnetostatic shape–shape interaction forces and torques. *Journal of Magnetism and Magnetic Materials 321*, 15 (2009), L45–L51. 2

[Fab08] FABBRI M.: Magnetic flux density and vector potential of uniform polyhedral sources. *IEEE Transactions on Magnetics 44*, 1 (2008), 32–36. 2, 4

[Gil55] GILBERT T.: A lagrangian formulation of the gyromagnetic equation of the magnetization field. *Phys. Rev. 100* (1955), 1243. 5

[GJK88] GILBERT E. G., JOHNSON D. W., KEERTHI S. S.: A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation 4*, 2 (1988), 193–203. 10

[HHM19] HUANG L., HÄDRICH T., MICHELS D.: On the accurate large-scale simulation of ferrofluids. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 93. 1, 3

[IYI*13] ISHIKAWA T., YUE Y., IWASAKI K., DOBASHI Y., NISHITA T.: Visual simulation of magnetic fluid using a procedural approach for spikes shape. In *Computer Vision, Imaging and Computer Graphics. Theory and Application*. Springer, 2013, pp. 112–126. 3

[Jac99] JACKSON J.: Classical electrodynamics. *American Journal of Physics 67*, 9 (1999), 841–842. 3, 4

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 6

[KE04] KLEIN T., ERTL T.: Illustrating magnetic field lines using a discrete particle model. In *VMV* (2004), vol. 4, pp. 387–394. 3

[KOU90] KOIZUMI M., ONISAWA M., UTAMURA M.: Three-dimensional magnetic field analysis method using scalar potential formulated by boundary element method. *IEEE transactions on magnetics 26*, 2 (1990), 360–363. 2

[KPH18] KIM S., PARK S., HAN J.: Magnetization dynamics for magnetic object interactions. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 121. 1, 2, 3, 5, 7, 8

[KS02] KUHN M., STEINBACH O.: Fem-bem coupling for 3d exterior magnetic field problems. *Math. Meth. Appl. Sci 25*, 5 (2002), 357–371. 2

[Mac58] MACMILLAN W.: The theory of the potential. 36–37. 12

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Eurographics Association, pp. 154–159. 3

[NZWC20] NI X., ZHU B., WANG B., CHEN B.: A level-set method for magnetic substance simulation. *ACM Transactions on Graphics (TOG) 39*, 4 (2020). 1, 3

[PLH16] PARK J., LEE K., HAN J.: Interactive visualization of magnetic field for virtual science experiments. *Journal of Visualization 19*, 1 (2016), 129–139. 3

[TGPS08] THOMASZEWSKI B., GUMANN A., PABST S., STRASSER W.: Magnets in motion. In *ACM SIGGRAPH Asia 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH Asia '08, ACM, pp. 162:1–162:9. URL: http://doi.acm.org/10.1145/1457515.1409115, doi:10.1145/1457515.1409115. 1, 2, 3, 6, 7

[VBHŠ09] VOKOUN D., BELEGGIA M., HELLER L., ŠITTNER P.: Magnetostatic interactions and forces between cylindrical permanent magnets. *Journal of magnetism and Magnetic Materials 321*, 22 (2009), 3758–3763. 2

[VOS83] VAN OOSTEROM A., STRACKEE J.: The solid angle of a plane triangle. *IEEE transactions on Biomedical Engineering*, 2 (1983), 125–126. 4

[WS96] WERNER R., SCHEERES D.: Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 castalia. *Celestial Mechanics and Dynamical Astronomy 65*, 3 (1996), 313–344. 12

[YLUH14] YOON W., LEE N., UM K., HAN J.: Computer-generated iron filing art. *The Visual Computer 30*, 6-8 (2014), 889–895. 3

A. Closed-form Expressions of $W_p(\mathbf{r})$ and $\nabla W_p(\mathbf{r})$

In Fig. 10-(a), the polygon, p, is parallel to the xy-plane, and $\mathbf{r}$ is set to $(\Delta x, \Delta y, \Delta z)^T$. Let r' denote the length of $\mathbf{r}$. Then, $W_p(\mathbf{r})$ can
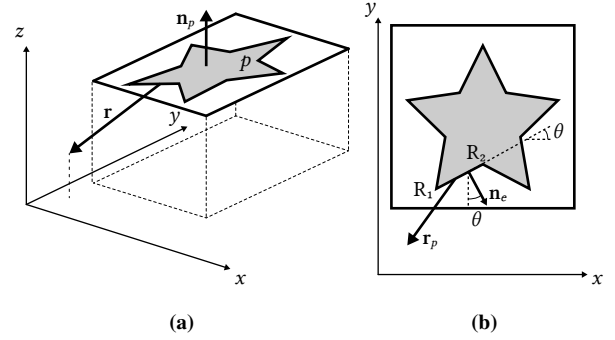
Figure 10: *A polygon parallel to the xy-plane. (a) The polygon normal is denoted as $\mathbf{n}_p$, and $\mathbf{r}$ is arbitrarily set to $(\Delta x, \Delta y, \Delta z)^T$. (b) Top view of the polygon, where $\mathbf{r}_p$ is the projection of $\mathbf{r}$ onto p's plane, and R_1 and R_2 are the ends of the polygon's edge, e.*

be written as follows:

$$
\begin{aligned}
W_p(\mathbf{r}) &= \int_p \frac{dp}{r'} \\
&= \int_p \left(\frac{r'^2 - \Delta x^2}{r'^3} + \frac{r'^2 - \Delta y^2}{r'^3} \right) dp - \Delta z \int_p \frac{\Delta z}{r'^3} dp \\
&= \int_p \left(\frac{\partial}{\partial \Delta x} \frac{\Delta x}{r'} + \frac{\partial}{\partial \Delta y} \frac{\Delta y}{r'} \right) - \Delta z \int_p \frac{\Delta z}{r'^3} dp \\
&= \int_p \left(\frac{\partial}{\partial \Delta x} \frac{\Delta x}{r'} + \frac{\partial}{\partial \Delta y} \frac{\Delta y}{r'} \right) - (\mathbf{r}_p - \mathbf{r}) \cdot \mathbf{n}_p \int_p \frac{\Delta z}{r'^3} dp \quad (21)
\end{aligned}
$$

where the second integral at the last line represents the signed solid angle, $\Omega_p(\mathbf{r})$, with respect to p, which was defined in Equation (15). On the other hand, the first integral at the last line can be transformed into the integral over the closed curve, using Green's theorem. Then, $W_p(\mathbf{r})$ is rewritten as follows:

$$
\begin{aligned}
W_p(\mathbf{r}) &= \oint_{\partial p} \frac{1}{r'} \left(\Delta x\, d\Delta y - \Delta y\, d\Delta x \right) - \Omega_p(\mathbf{r})(\mathbf{r}_p - \mathbf{r}) \cdot \mathbf{n}_p \\
&= \sum_{e \in \partial p} \int_e \left(\frac{1}{r'} \left(\Delta x\, d\Delta y - \Delta y\, d\Delta x \right) \right) - \Omega_p(\mathbf{r})(\mathbf{r}_p - \mathbf{r}) \cdot \mathbf{n}_p
\end{aligned}
$$

$$(22)$$

Let $W_e(\mathbf{r})$ denote the integral at the last line of Equation (22). In Fig. 10-(b), e connects R_1 and R_2. Let s denote the unit vector from R_1 to R_2. Then, $W_e(\mathbf{r})$ can be rewritten as follows:

$$
\begin{aligned}
W_e(\mathbf{r}) &= \int_e \frac{ds}{r'} \left((\Delta x + s\cos\theta)\sin\theta - (\Delta y + s\sin\theta)\cos\theta \right) \\
&= \left(\Delta x \sin\theta - \Delta y \cos\theta \right) \int_e \frac{ds}{r'} \\
&= (\mathbf{r}_p \cdot \mathbf{n}_e) \int_e \frac{ds}{r'} \quad (23)
\end{aligned}
$$

where $\mathbf{n}_e$ is orthogonal to s and points out of p. It is clear that $\mathbf{r}_p \cdot \mathbf{n}_e$

is identical to $\mathbf{n}_p \times (\mathbf{r}_e - \mathbf{r}) \cdot \mathbf{u}_e$ introduced in Equation (12). The integral at the last line of Equation (23) is the same expression as $\omega_e(\mathbf{r})$ introduced in Equation (14), as shown in **Appendix B**. This proves that Equation (21) equals Equation (12).

Next, $\nabla W_p(\mathbf{r})$ can be written as follows:

$$
\begin{aligned}
\nabla W_p(\mathbf{r}) &= \int_p \nabla \frac{d^2 r'}{r'} \\
&= \int_p \begin{pmatrix} -\dfrac{\partial}{\partial \Delta x}\dfrac{1}{r'} \\ -\dfrac{\partial}{\partial \Delta y}\dfrac{1}{r'} \\ -\dfrac{\partial}{\partial \Delta z}\dfrac{1}{r'} \end{pmatrix} d^2 r' \\
&= -\int_p \begin{pmatrix} \dfrac{\partial}{\partial \Delta x}\left(\dfrac{1}{r'}\right) + \dfrac{\partial}{\partial \Delta y}(0) \\ \dfrac{\partial}{\partial \Delta x}(0) + \dfrac{\partial}{\partial \Delta y}\left(\dfrac{1}{r'}\right) \\ -\dfrac{\Delta z}{r'^3} \end{pmatrix} d^2 r' \quad (24)
\end{aligned}
$$

Using Green's theorem, Equation (24) can be rephrased as follows:

$$
\begin{aligned}
\nabla W_p(\mathbf{r}) &= \begin{pmatrix} -\displaystyle\oint_{\partial p} \dfrac{d\Delta y}{r'} \\ \displaystyle\oint_{\partial p} \dfrac{d\Delta x}{r'} \\ \displaystyle\int_p \dfrac{\Delta z}{r'^3} d^2 r' \end{pmatrix} \\
&= \begin{pmatrix} -\displaystyle\sum_{e\in\partial p}\sin\theta \int_e \dfrac{ds}{r'} \\ \displaystyle\sum_{e\in\partial p}\cos\theta \int_e \dfrac{ds}{r'} \\ \Omega(\mathbf{r}) \end{pmatrix} \quad (25)
\end{aligned}
$$

The x and y components of the last line can be rewritten as follows:

$$
\sum_{e\in\partial p}(-\sin\theta,\cos\theta)\cdot\omega_e(\mathbf{r}) = \sum_{e\in\partial p}-\omega_e(\mathbf{r})\mathbf{n}_e \quad (26)
$$

where $-\mathbf{n}_e$ is identical to $\mathbf{n}_p \times \mathbf{u}_e$ in Equation (13). This proves that Equation (25) equals Equation (13).

Derivations of $W_p(\mathbf{r})$ and $\nabla W_p(\mathbf{r})$ are based on the concept of gravitation introduced in Werner and Scheeres [WS96], to which readers are referred for more detail.

B. Closed-form Expression of $\omega_e(\mathbf{r})$

Recall that the integral at the last line of Equation (23) is identical to $\omega_e(\mathbf{r})$. Then, using Fig. 11, $\omega_e(\mathbf{r})$ can be written as follows:

$$
\begin{aligned}
\omega_e(\mathbf{r}) &= \int_0^l \frac{d\xi}{\rho} \\
&= \int_{\theta_1}^{\theta_2} \csc\theta\, d\theta \\
&= \ln\frac{\tan(\theta_2/2)}{\tan(\theta_1/2)} \quad (27)
\end{aligned}
$$

where the second line is obtained as $d\xi\sin\theta = \rho\,d\theta$. When $\xi = 0$, $\theta = \theta_1$. When $\xi = l$, $\theta = \theta_2$. Because θ is limited to the range
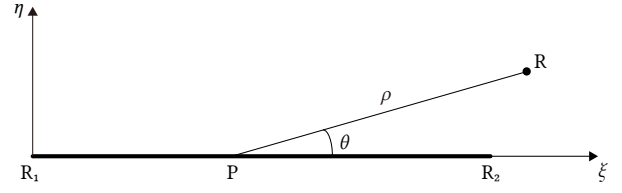
Figure 11: *On the $\xi\eta$-plane, $R_1 = (0,0)$, $R_2 = (l,0)$, $R = (u,v)$, and the arbitrary point $P = (\xi,0)$. P is on $\overline{R_1 R_2}$, and the length of $\overline{PR}$ equals ρ.*

between 0 and π, it is clear that the following trigonometric identity holds:

$$
\tan\frac{1}{2}\theta = \sqrt{\frac{1}{\tan^2\theta}+1} - \frac{1}{\tan\theta} \quad (28)
$$

In addition, we can express the tangents as follows:

$$
\begin{aligned}
\tan\theta_1 &= \frac{v}{u} \\
\tan\theta_2 &= \frac{v}{u-l}
\end{aligned} \quad (29)
$$

Using Equations (28) and (29), Equation (27) can be rephrased as follows:

$$
\begin{aligned}
\omega_e(\mathbf{r}) &= \ln\frac{\tan(\theta_2/2)}{\tan(\theta_1/2)} \\
&= \ln\frac{\sqrt{(u-l)^2+v^2}-(u-l)}{\sqrt{u^2+v^2}-u} \\
&= \ln\frac{\sqrt{(u-l)^2+v^2}+\sqrt{u^2+v^2}+l}{\sqrt{(u-l)^2+v^2}+\sqrt{u^2+v^2}-l} \\
&= \ln\frac{\overline{R_1 R}+\overline{R_2 R}+\overline{R_1 R_2}}{\overline{R_1 R}+\overline{R_2 R}-\overline{R_1 R_2}} \quad (30)
\end{aligned}
$$

Derivation of the closed-form expression of $\omega_e(\mathbf{r})$ is based on the concept of potential introduced in Macmillan [Mac58], to which readers are referred for more detail.