THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MECHANICAL ENGINEERING

An Investigation into Complex Systems of Magnetic Particles Exposed to External Fields

JAMES ZOLA
SPRING 2023

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Mechanical Engineering
with honors in Mechanical Engineering

Reviewed and approved* by the following:

Paris R. Vonlockette
Professor of Mechanical Engineering
Thesis Supervisor

Margaret Byron
Assistant Professor of Mechanical Engineering
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

Understanding the collective motion of magnetic particles has become of interest in recent years because of their potential applications in targeted drug delivery, manipulation of material properties, and heat production. Current models mostly consider collective behavior driven by fluid flow induced by particle-fluid interaction. This thesis attempts to investigate the role of interparticle dipole interactions in collective behavior of magnetic particles, expanding a mathematical model of particle dynamics to simulate grids of rotating magnetic particles in an applied field. Simulations are run for different frequencies and magnitudes of the applied field to investigate the effect of these parameters on the behavior of the system. Additionally, a physical model is constructed and tested to corroborate the results of the simulation. Because of the ability of the system to retain information about past stimuli, magnetic particle arrays are of potential interest in neuromorphic materials, which adapt their properties based on prior input.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. von Lockette for his patience, enthusiasm and constant support. His thorough understanding of magnetics has been indispensable to this project, but also his research mindset has taught me to critically question my work and anticipate questions that others might ask. I would also like to thank Dr. Najem for his expertise with neuromorphic materials and his aid in classifying system memory. Finally, this work would be impossible without the support of my family, who have provided me with encouragement, intellectual stimulation and a lifelong love of learning.

Chapter 1

Introduction

Magnetic particles suspended in fluid media have become an area of interest in recent decades as a vehicle for drug delivery [1], [2], material processing [3], mixing tools [4], and even as heating elements [5] because of their ability to be actuated non-invasively using magnetic fields. Another advantage of magnetic particle suspensions is that they are reconfigurable, meaning the particles can be arranged into patterns without losing the ability to take on a new form based on different input conditions [6]. Thus, one material can move freely between several different states. To better be able to leverage the unique behaviors of these particles, significant research is dedicated to understanding how these particles move and interact with one another. The motion of such particles is relatively complex, as it depends on factors such as the applied magnetic field frequency and magnitude. It is also influenced by any fluid effects that arise because of the medium such as drag, and interparticle interactions both caused by the motion of the fluid and magnetic moments of the particles [7]. The complexity of the particle-fluid system can give rise to a number of interesting behaviors and patterns in the form of particle motion as a result of external activation.

Magnetic Particle Systems and Behavior

Kaiser et Al. characterized three unique responses for nickel particles dispersed in a fluid medium on a convex lens, using both computational and experimental methods [8]. They found that particles responded differently to different frequencies of applied field, leading to a tunable system whereby changing the frequency of the magnetic field could change the behavior of the

system to one of three responses they observed. All three different types of behavior are shown in the plots in Figure 1. In the first stage, at low frequencies, the particles moved with a chaotic motion, with no large-scale collective motion. At a higher frequency, the particles exhibited a flocking behavior, where large groups of particles moved together coherently in groups or flocks, splitting off and reforming into groups. This flocking behavior is similar to behavior observed in groups of birds flying together. A higher frequency band caused all the particles to begin following a similar pattern of circular motion around the same central axis, forming a vortex. Interestingly, when the frequency was increased further, the particles returned to flocking behavior and then finally to random behavior. These transitions suggest that at high frequencies, the particles were no longer able to keep up with the oscillation of the magnetic field and thus it no longer had an organizing effect, while at lower frequencies, the particles rolled in tandem with the field, creating increasingly strong levels of organization.

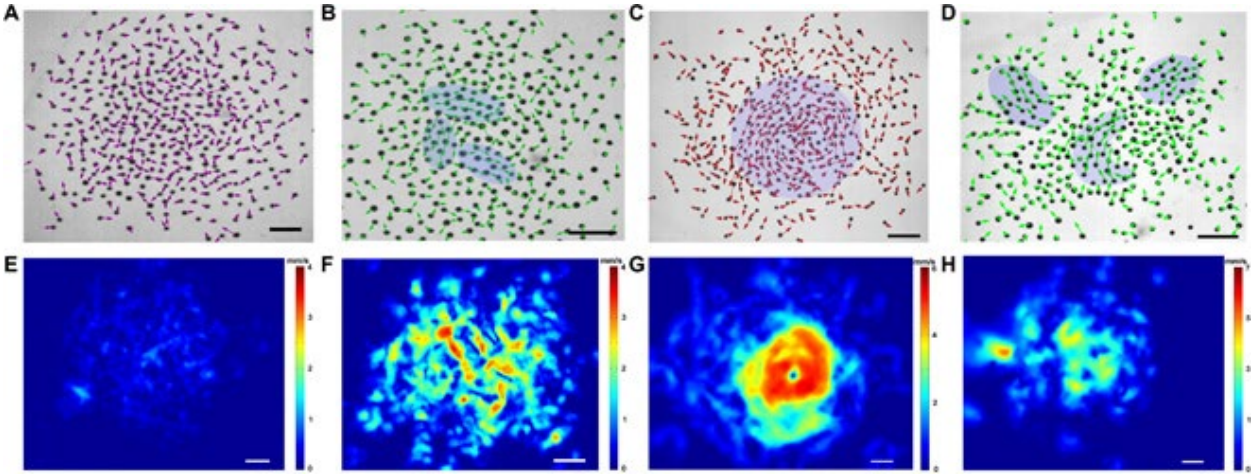

Figure 1. This figure shows the three different types of behavior found in the study by Kaiser et Al. Plots A-D show the velocity vectors of each different type of behavior, with bulk behavior indicated by a darker blue coloration in the background. In plots E-H, the magnitude of the velocity field is shown, with higher velocity represented as red in color while lower velocity is blue in color. Plots A and E show the gas behavior, plots B and F show the low frequency flocking, plots C and G show the vortex behavior, and plots D and H show the high frequency flocking behavior. CC BY-SA 4.0 Licensed, reproduced from [8].

Another study used low frequency magnetic fields to study the impact of chain formation in ferromagnetic fluids, which are suspensions of magnetic particles on a micro- scale [3]. The

study found that chains could be made both thicker and longer with a low frequency cycling field, as particles were able to change direction to align with the field, temporarily creating transverse assemblies where particles separated slightly and attracted more adjacent chains. Adjacent chains then merged to form longer and thicker chains as a result of the field oscillation, and Figure 2 shows the effects of frequency on these factors. The tunable parameters of magnetic field frequency and strength can be used to reliably manufacture materials with chains of magnetic particles of different length and thickness.

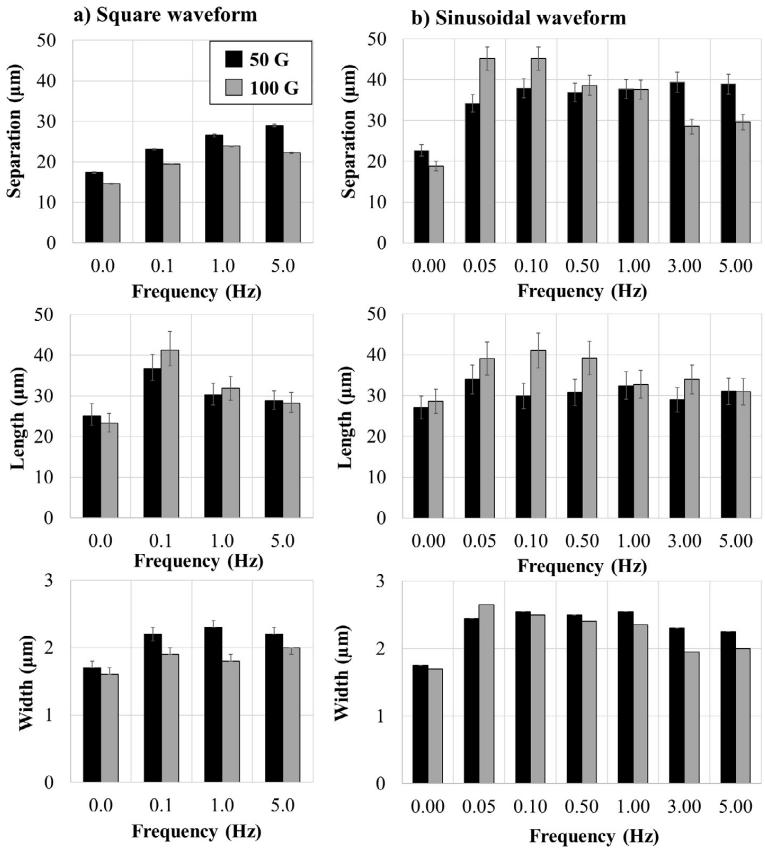

Figure 2. This figure shows the effects of frequency on separation distance, length, and width of chains of magnetic particles. Data is given for two different waveforms, square and sinusoidal, and two different field strengths, 50 and 100 Gauss. Reproduced with permission from [3].

Further studies have demonstrated the effect of rotating magnetic fields on the self-organization of magnetic particles suspended in a fluid. Gryzybowski et Al. performed

experiments with magnetic discs fabricated from iron oxide and PDMS which were suspended in a fluid and exposed to a rotating magnetic field [9]. Fluid interaction with the particles caused them to organize into several distinct structures depending on the number of particles present in the suspension. A later study developed a simulation of the fluid dynamics which cause this self-organizing behavior for spherical particles, and found their model to have good agreement with experimental results [10]. Expanding these results to greater numbers of particles, Kokot et Al. performed both simulation and experiment on large systems of magnetic particles at a liquid-air interface [11]. Their particles were caused to rotate by an oscillating magnetic field generated by an electromagnet, and while they were most interested in the fluid behavior as a result of these active particles, they did find that particles rotating in the same direction tended to attract one another and form groups of particles with like rotation direction, similar to the results found in previous studies.

While magnetic materials provide a promising option for these complex behaviors and controlled particle actuation, they are not the only particles that can be manipulated in this way. Electrically charged Janus particles, which are dipolar particles that are created by coating one hemisphere of a particle with metal to create an uneven electric dipole, can be actuated in a similar manner to the magnetic particles discussed previously. In a study by Jing et Al., these particles were studied under an AC electric field, with different frequencies inducing different behaviors in the particles due to interactions between the uneven dipoles [12]. Particles exhibited clustering behavior, where particles were grouped together by electric forces. This behavior arises due to the repulsive force for positive charges in the Janus particles being greater than the negative repulsion force, causing the positive repulsion to dominate and creating clusters with negative charge concentrated in the center. Additionally, particles exhibited chaining and flocking behavior, where

positively charged hemispheres on one particle were attracted to negatively charged particles on another, causing the particles to move collectively or form chains. The emergent behaviors in these solutions bear strong similarities to the magnetic particle solutions discussed earlier.

Despite all the research that has been done regarding magnetic particle motion, few studies investigate the effects of the magnetic interparticle interactions that occur in these suspensions. In fact, several studies ignore these interactions entirely [5], [8], either because they are assumed to be negligible for a given situation, or to simplify a model. Nonetheless, interparticle interactions provide an area of interest because they have a strong effect on the organization of active particles. Belovs and Cebers demonstrate how magnetic interaction between particles causes them to synchronize their rotation [13]. In [8], this synchronized rotation is typical of the vortex phase of particle organization. Particle interactions are also key in the case of Janus particles [12]. They also provide an opportunity for study, as relatively little work has been done concerning these interactions.

Interparticle Interactions and Complex Systems

Complex systems analysis is the study of systems of interrelated elements that give rise to emergent behavior, where the system as a whole behaves differently than the individual elements do. While the study of complex systems originated in physics, they are found across disciplines, and they can be found in social sciences, such as economics or sociology, as well as physical sciences, like chemistry or biology. Complex systems analysis can be used to assess behaviors such as flocking and locked rotation. An interesting consequence of complex systems, discussed in a paper by Siegenfeld and Yam, is that interaction between elements in a complex system can

lead the system to be more volatile [14]. Essentially, behavior that would be extremely unlikely for the individual constituents of a system becomes more likely to occur when the components of the system interact.

An example of the wide range of complex behaviors is a study by Nguyen et Al. where systems of rotating particles were investigated [15]. The particles rotate in two dimensions and are classified as either clockwise or counterclockwise rotators. The system is a symmetrical one, because the particles themselves behave exactly the same regardless of rotation direction. In the study, rotating particles self-organized by rotation direction, forming groups of particles with the same rotation. Because of the way particles interacted with each other when colliding, spending more time interlocked with particles of the same rotational direction than particles rotating in opposing directions, these areas of like behavior formed. Thus, even seemingly symmetrical systems can have interactions that promote self-organization and complex behavior. The magnetic particles discussed earlier easily fit into the complex system model and will have self-organizing behavior that make them worthy of study [6].

Aeroelastic Flutterers

Another area of research that studies complex interaction is found in aeroelastic flutterers, which are essentially cantilevered beams that are flexible enough to move in fluid flow. Aeroelastic flutterers were originally conceived as an idea for small quantities of energy generation needed to power sensors on airplanes or other small electronic devices exposed to fluid flow. These harvesters take advantage of piezoelectric effects, which convert strain to electric potential, and by deforming in fluid flow induce a small piezoelectric voltage. These harvesters are a well-

developed area of research [16], and a subset of studies investigate the interaction between harvesters, as multiple harvesters in a given configuration interfere with the flow they collectively experience. The flow interaction is analogous to interactions in the fluid that occur in magnetic particle suspensions, and magnetic particles have an additional area of interaction from the magnetic field.

Multiple studies have been conducted to optimize the positioning of these harvesters to benefit the system. Bryant et Al. investigated systems of two and four harvesters, finding that positioning the harvesters directly behind each other with no spacing led to about 10-25% more power generated by each harvester in the system than the individual harvesters could have generated alone [17]. Figure 3 shows the power generated by each harvester in a four-harvester array normalized by the power from a single harvester plotted against the separation between the tail of one harvester to the front of another normalized by the length of a single harvester. The figure also shows the efficiency of the arrays with each separation distance. The interactions of the harvesters elevated the overall energy state of the system, causing the individual harvesters to behave in ways they would not when separated. Applying this concept to the magnetic particles discussed earlier, it becomes clear that the particle interactions are a crucial part of the behavior of the system and could lead to net behaviors that would not occur if the particles did not interact at all. This result is well maintained in other optimization studies which investigate the effects of wake interactions on aeroelastic flutterer devices of various geometries [18], [19].

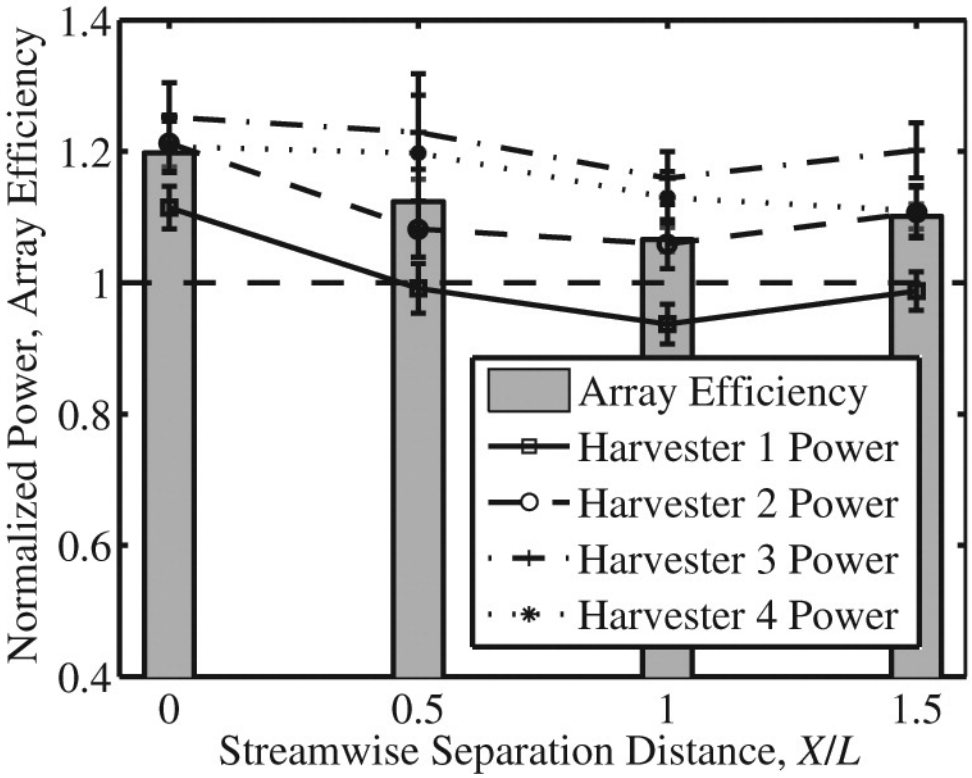

Figure 3. A figure showing the enhanced power generation of the array based on separation distance between flutterers. Reproduced with permission from [17].

Of particular interest to this study are the methods of quantifying the behavior of these aeroelastic fluttering systems, as they may be applied to this research to demonstrate any similar patterns that might occur in magnetic systems. Modal analysis [19], amplitude and frequency of plate motion [18], and power generation [17] were metrics used in the evaluation of these aeroelastic systems. While there is no power generated in the system of magnetic particles studied in this thesis, the other metrics may prove useful in providing methods to characterize the behavior of these magnetic systems and contextualize them in a similar way to the aeroelastic flutterers, complex systems which are already well understood.

Memristors and Neuromorphic Materials

Of particular interest as an application for these systems of magnetic particles are memristors and other neuromorphic materials. Memristors are a theorized circuit element that are variable resistors whose resistances change based on combined effects of past inputs [20]. Essentially, frequent inputs cause the resistance of a memristor to decrease, allowing more current to flow through. A lack of input or infrequent input causes the resistance of the memristor to rise again, allowing less current to pass. This theoretical circuit element mimics the behavior of neurons in the human brain. In a human brain, a charge builds up at the interface between neurons, which, when reaches a certain threshold, causes the neuron to fire. The charge required to induce this firing is dependent on the strength of the neurons in contact, so neurons that are not as closely wired together require more buildup before firing, and vice versa for close neurons.

Memristors can be used in neuromorphic materials, which are networks for artificial intelligence that could replace conventional binary computing models. One model of these computing architectures uses conducting bars in a grid formation connected by memristors at junctions between conductors, pictured in Figure 4 [21]. This structure allows multiple model "neurons" to be wired together, so that when one of the memristors is activated, the signal is transferred to all of the other memristors on a bar. This joined connectivity mimics how neurons are connected in the brain, allowing these networks of neurons to be developed simultaneously.

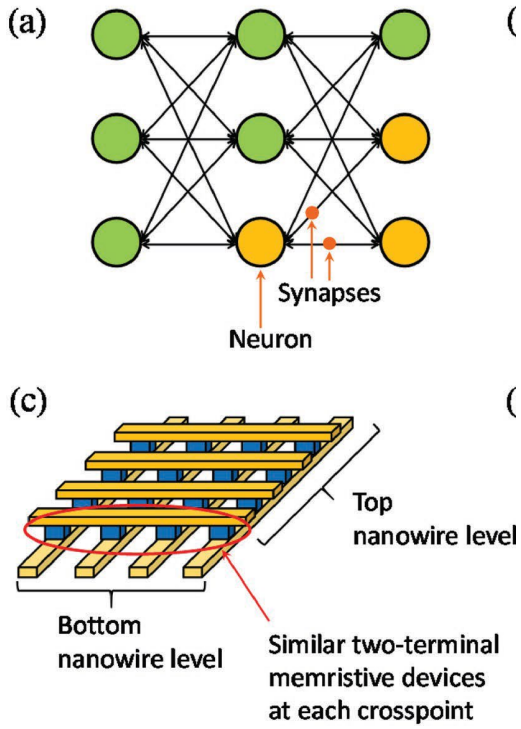

Figure 4. An image of a crossbar configuration that uses memristors to model neural networks. The image in (a) shows a model of how neurons are connected in the brain while (c) shows the model. Reproduced in part with permission from [21]

While in an early stage of development, neuromorphic materials provide advantages to more conventional computing techniques, namely in that they can both store memory and perform computations in the same location [22]. A typical computer would need to write values to memory and access them to perform computations in a separate location, but in a neuromorphic model, these processes can occur in the same place. Thus, neuromorphic materials are much more time-efficient than conventional computers and can be made to use a lot less power.

A loose analog of neuromorphic behavior can be identified in the magnetic systems discussed earlier. As the frequency of input changes, in this case the applied magnetic field, so does the response of the system. An ideal memristor can change continuously, unlike the magnetic systems which cycle through discrete forms, but they nonetheless provide an analog to memristive behavior that warrants investigation. Finally, some fluids and elastomers have shape-memory

behavior, where the deformation of the material is influenced by previous states, so the choice of medium for magnetic particle suspensions can produce effects on the particles that are dependent on past motion.

Summary

The goal of this thesis, then, is to create a computational and experimental model of a magnetic particle system that is able to showcase complex behavior akin to that seen in other systems of active particles discussed previously [1], [3], [8], [12]. Particles ought to be able to organize and reorganize into rotational patterns at different frequencies of applied field. In order to characterize this behavior, methods from the study of aeroelastic flutterers will be used, such as analyzing the modes of the system or the frequency and magnitude of response induced in the particles in relation to the applied field [17]–[19]. This created array may then be applicable as a memristor, as there will be several discrete states of behavior across a wide range of frequency input [21]. It may be beyond the scope of this study to translate these states to useful output, but it is worth noting that such outcomes are a possibility and indeed an end goal for future study in this lab.

Chapter 2

Methods

Adapting an in-house particle dynamics simulation, an array of 25 hard magnetic particles exposed to an external magnetic field were modeled. The simulation parameters were based on barium hexaferrite, a material which forms disc-shaped, hexagonal, microscale particles. These particles were chosen because of the strong, permanent magnetic moment they create, allowing for interaction between the particles regardless of the presence of an applied field. The particles are arranged into a grid structure, with initial angles of magnetic moment randomly determined, although the same random initial condition is kept constant across varied test cases with unique input parameters. Only rotational motion is investigated, both to cut down on simulation time, because of the computational power required to model translational motion, and also because the relative orientation of the particles' magnetic moments is what causes interaction between them, with distance between the particles only serving as a scaling factor. The roles of each of these factors on the magnetic torque on each particle are more clearly seen in equations (1) and (2), which are derived from torque sum on the nth particle and the local magnetic field on the nth particle created by summing the contribution of each of the other m particles. These equations are derived in the book Ferrohydrodynamics, and applied to the simulation framework in another thesis from this group [23], [24].

(1) $\dot{\boldsymbol{d}}_n = \frac{\mu_0 MV}{8\pi\eta R^3}(\boldsymbol{H}_n - \boldsymbol{d}_n(\boldsymbol{d}_n \cdot \boldsymbol{H}_n))$

(2) $\boldsymbol{H}_n = \boldsymbol{H}_0 + \sum_{m=1}^{N} \frac{M_m V_m}{4\pi r_m^3}[3(\boldsymbol{d}_m \cdot \hat{\boldsymbol{r}}_m)\hat{\boldsymbol{r}}_m - \boldsymbol{d}_m]$

In equation (1), $\boldsymbol{d}_n$ refers to a unit vector in the direction of the particle's magnetization vector, μ_0 is the permeability of free space, M is the magnitude of the particle's magnetization in A/m, V is the particle volume, R is the particle radius, η is the drag coefficient of the surrounding fluid, assuming drag linearly proportional to angular velocity, and $\boldsymbol{H}_n$ is the local magnetic field acting on the nth particle. In equation (2), $\boldsymbol{H}_0$ is the globally applied field, M_m and V_m are the corresponding magnitude of magnetization and volume of the mth particle, respectively, r_m is the magnitude of the direction vector from the nth particle to the mth particle, $\boldsymbol{d}_m$ is a unit vector in the direction of the mth particle magnetization, and $\hat{\boldsymbol{r}}_m$ is a unit vector in the direction from the nth particle to the mth particle.

Equation (1) assumes that the second order term in the differential equation of motion is zero. Because the particles are so small, they have small moments of inertia. This, coupled with the particles' strong magnetic moments, means that the acceleration of the particles can temporarily become quite large as they are exposed to strong torques from interparticle interactions or interactions with the applied field. Accelerations on the order of $10^{13}\ rad/s^2$ mean that the time step for any iterative approximation of the system must be prohibitively small to account for the massive acceleration the particles experience, but since an equivalent deceleration can quickly be reached once the particles rotate beyond alignment with the applied field, the large accelerations the particles experience decay quickly. A relatively good approximation of the system can still be achieved if the second order term is set to zero, assuming that the small moment of inertia means that any magnetic torque is large enough to accelerate or decelerate the particle instantaneously.

In the simulation, the particles are driven by a sinusoidal magnetic field H_0, which oscillates with frequency f, and whose peak strength scales with the strength of the particle magnetization M. The particles are assumed to be identical, with identical magnetizations and

volumes. The simulation uses a Runge-Kutta approximation to iterate over a time range and approximate the solution to the particle dynamics equations. Data is collected on particle orientation, speed, and the strength of the magnetic field at each time step. Selected parts of the code used to model particle motion can be found in Appendix A.

In a paper by Egri and Bihari, physical magnets were driven by an electromagnet to cause them to spin [25]. The study investigates the effect that one driving magnet had on an array of magnets, showing that the rotation of a driven magnet could ripple out and affect layers of magnets which were not exposed to an external magnetic field. Using a square wave generated from a function generator, the magnet moved erratically at low frequencies, but could be made to exhibit continuous rotation at higher frequencies. Cycling between behaviors at different frequencies is part of the desired behavior of the magnetic systems modelled in this thesis. Taking inspiration from the Egri study, a physical model of the simulated system was developed, whereby physical magnets were placed in the gap of an electromagnet to recreate the conditions in the simulation.

Table 1. Table of RMS values for field strength of custom electromagnet when supplied with given current and frequency

Peak to Peak Current (A)	*Field Strength @ 0.1 Hz (mT)*	*Field Strength @ 0.5 Hz (mT)*	*Field Strength @ 1 Hz (mT)*
1.25	11.909	10.058	10.228
2.50	23.422	21.372	21.834
3.75	33.714	32.015	33.875
5.00	44.771	46.198	47.848
6.25	57.653	56.698	63.578
7.50	65.652	67.164	67.891
8.75	78.098	76.222	73.105
10.00	87.903	87.271	75.410
11.25	91.975	93.572	78.207
12.50	93.054	94.817	75.934

A custom electromagnet was used to drive the rotation of a set of four permanent magnets. The specifications of the electromagnet can be seen in Table 1, which displays the measured RMS field strength of the electromagnet for input current at a variety of frequencies. The current was controlled by a pair of KEPCO BOP 50-20MG power supplies wired in series, which enabled them to supply periodic, computer controlled current. The permanent magnets were obtained from K&J magnetics, model B848CS, and were fixed in place by a PLA housing printed on a 3D printer. Because of limitations such as the gap in the electromagnet, the filament size on the 3D printer, and the size of the permanent magnets available for purchase, a 2 by 2 array of magnets was used instead of the 5 by 5 array used in the computer simulations. A picture of the experimental setup can be seen in Figure 5.

Figure 5. The experimental setup used to test the physical model. A pair of KEPCO power supplies wired in series, on the right, power custom electromagnet, on the left. The array of magnets is placed within the gap in the electromagnet. A camera records the magnet motion.

Data collection for the physical model was achieved through use of a dot tracking program which analyzed a video recording of each experiment to collect particle orientation data at each frame of the video. Each magnet was marked with a red dot at one pole, so that the dot tracking program could identify the direction of the magnetic moment. The code for the dot tracking algorithm can be found in Appendix B. Magnets were coated with graphite powder to reduce the effect of friction on the rotation of the magnets, and to limit the possibility of magnet edges being caught on the housing, impeding motion.

Unlike the computer simulation, where a repeatable randomized initial particle configuration could be achieved by providing a seed to a random number generator, for the physical model, an initial particle configuration had to be selected to ensure repeatability across all case tests. The magnets naturally settled into a configuration where the magnetization moment of each particle was aligned at about a 45-degree angle with the field axis when placed in a 2 by 2 configuration. For this reason, this "diamond" arrangement was chosen as the common initial condition. The diamond configuration is shown in Figure 6.

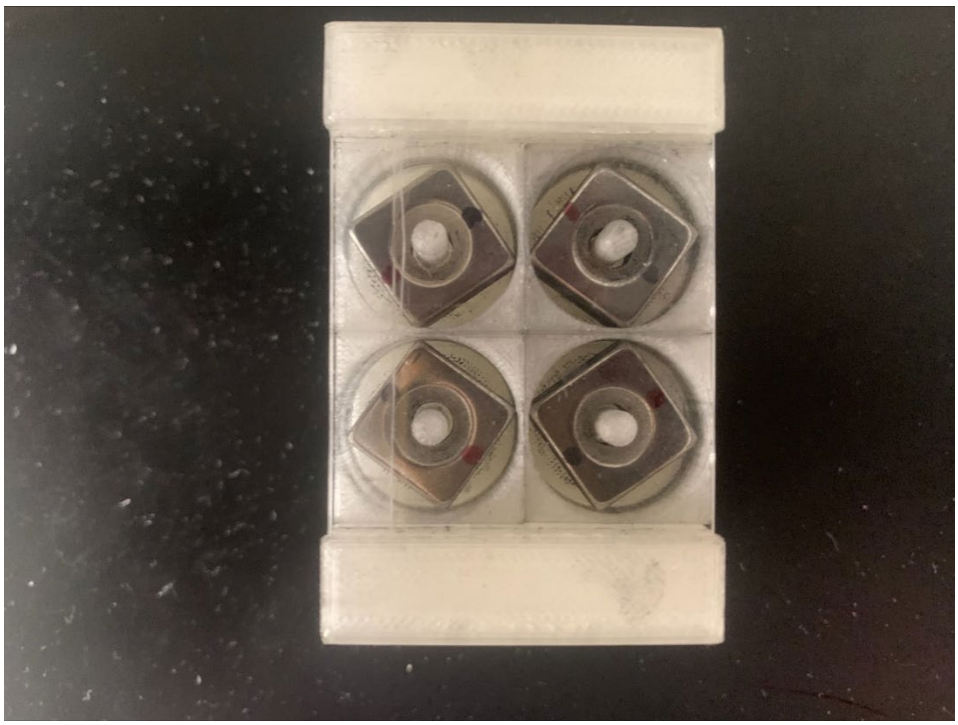

Figure 6. The diamond configuration of the magnets. The red dots indicate the direction of the magnetic moment of the particle, which is at an approximately 45-degree angle with the horizontal axis, where the field is applied.

Chapter 3

Results and Discussion

Simulation Results

At first glance, the system of magnetic particles seems rather simple. When the applied field switches, the particles move to align with the field completely, switching direction with the applied field to align opposite how they were initially. However, as the frequency and strength of the applied field are changed, some deviation to this behavior occurs, yielding systems with different organizational behavior. When the applied field strength is too low to completely overcome the interparticle interactions, particles enter a low energy state where each particle assumes an approximately 45-degree angle with the field axis, a diamond configuration analogous to the one used as a starting position in the physical model. The applied field causes the particles to oscillate around these fixed positions, with slight deviations as the particles move into a configuration of greater alignment depending on the direction and strength of the field in time. Similarly, as frequency increases, the particles can no longer move fast enough to fully align with the changing field, and so form these oscillating patterns. Plotting the particle angle with the field axis against field strength paints a picture of these oscillatory patterns, showing how the particle position is affected by field strength over multiple cycles.

At intermediate field strengths and frequencies, a different sort of pattern is formed, where the particles align in rows perpendicular to the direction of the applied field. Particles oscillate around these positions, not fully aligning with the applied field but moving closer to a direction of alignment. When the field is low, the particles returned to being perpendicular to the field axis. A representation of this pattern is shown in Figure 7.

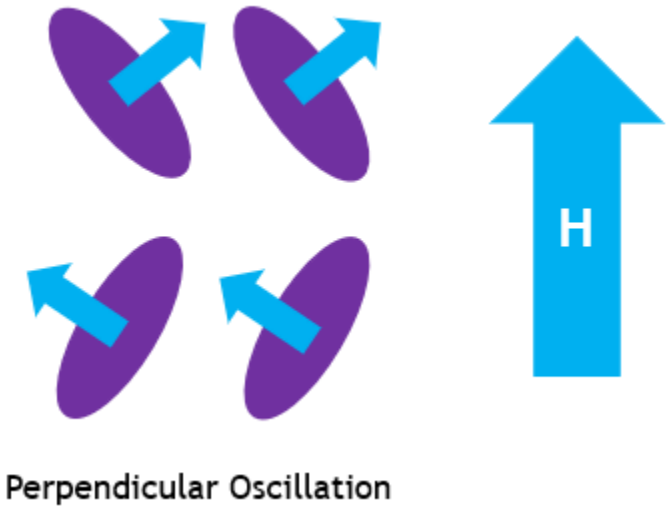

Figure 7. The perpendicular oscillation position. Blue arrows indicate the magnetic moment of the particles, which are represented by purple ellipses. Rows of aligned particles are formed, which move to align with the applied magnetic field H. Rows alternate direction, but all are perpendicular to the applied field.

At high frequencies and high field strengths, an initial preference is formed toward one alignment direction – the positive field direction. Particles cannot keep up with the rapidly switching field, but the field strength is enough to bias them in one direction, so that when the field direction switches, the majority of particles are at least partially aligned with the positive y. Then, the particles have to cover a greater distance to align with the negative y, a process which is not completed in its entirety by the time the field has reversed direction once more. Particles end up more aligned with the positive y after each iteration, eventually able to reach full alignment in this direction, but not the reverse, creating a hysteresis loop with a flattened bottom (representing the positive y) and curved top (representing the negative y). Figure 8 shows a close example of this pattern, which occurs at a maximum field strength of 1 times the magnitude of particle magnetization and a frequency of 100 Hertz.

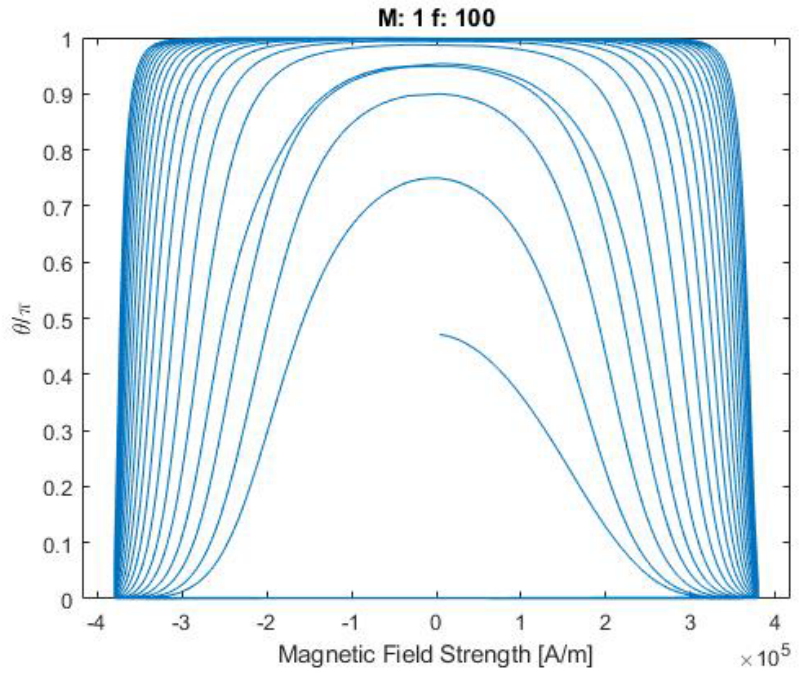

Figure 8. A close image of the hysteresis curve for a particle with the initial direction preference. This pattern occurs at a maximum external field equal to the particle magnetization and a frequency of 100 Hz

A major draw of these magnetic systems is their ability to display memory-like behavior. It is important to discuss two types of memory when considering systems like these. Volatile memory involves hysteresis loops that retrace the same basic path after each iteration. They appear as simple closed loops. Systems with involatile memory have increased response over time, typical of a memristive system. They appear as multiple inlaid loops, where the total area enclosed within the hysteresis loop changes over time. In a system without memory, the plot is a line, with no area enclosed. The path traced out by such a system is always the same, regardless of the direction it is moving, from positive y to negative y or vice versa. Both types of memory can be observed in Figure 9, which plots magnetic field strength versus average particle position for the simulated system. The particles require some minimum field strength in an opposite alignment to their own magnetic dipole to begin to switch alignment. Additionally, particles trace out gradually expanding paths before finally settling into a limit cycle, a system with saturated memory capacity.

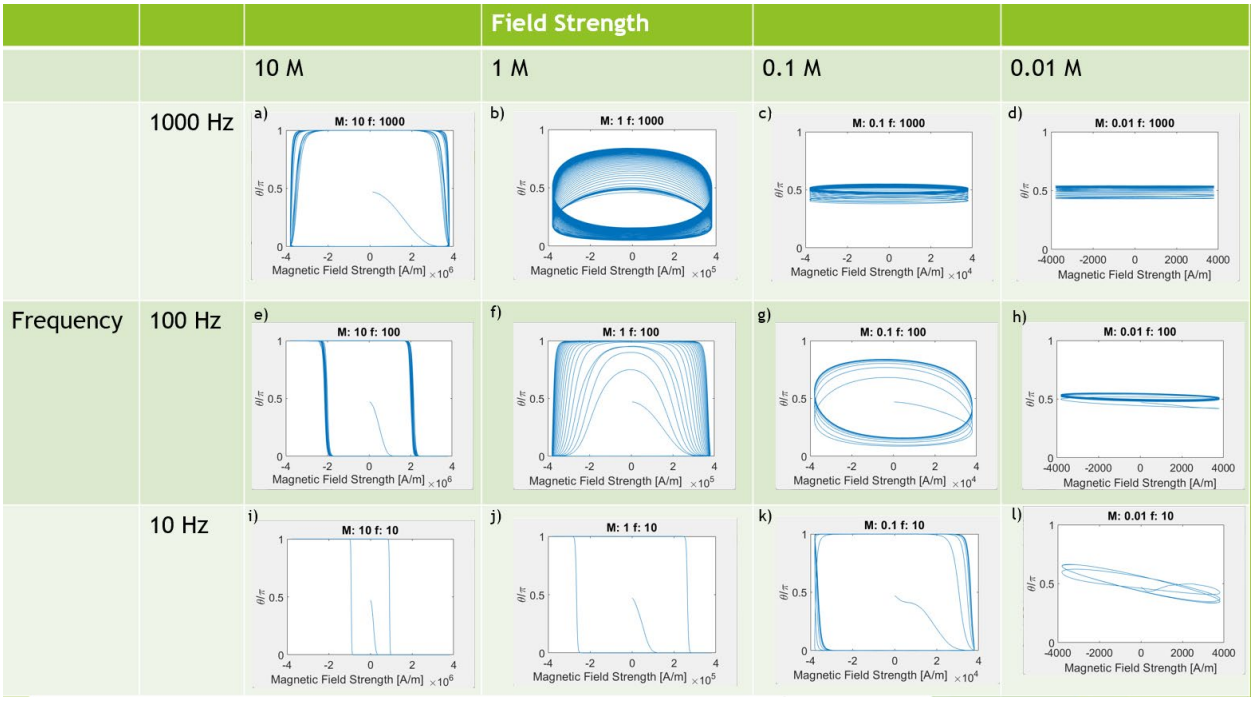

Figure 9. A figure showing the field strength versus average particle position hysteresis curves for a set of applied field strengths and frequencies. The field strength is varied from left to right, with high field strength on the left side. The frequency is varied from top to bottom, with high frequency at the type. The curves display both volatile memory, as a pattern is swept through multiple times, and involatile memory, as subsequent cycles lead to increased hysteresis.

Magnetic field strength and frequency represent two conflicting influences over the behavior of the system as a whole. A high field strength will be a driver of organization in the system, causing it to behave more collectively as torques caused by particle interaction are overwhelmed by the external field. A high frequency, on the other hand, is a disordering factor in the system, as the field strength cycles too quickly for the particles to fully align with the external field. In this circumstance, interparticle torques are more consistent than magnetic torques and thus dominate system behavior. Measuring the area beneath the hysteresis curve provides a measure of the disorder present in the system, as it represents the energy dissipated by the system during each cycle. Hysteresis values for given inputs of field strength and frequency are plotted in Figure 10, with log scales used for all three axes. Generally, the system behaves as described, with increased field strength at low frequencies leading to lower hysteresis values and increased frequencies across the spectrum of field strength leading to increased hysteresis. On this scale, it appears

that frequency dominates when determining hysteresis, as hysteresis is at its peak with high field strength and frequency.

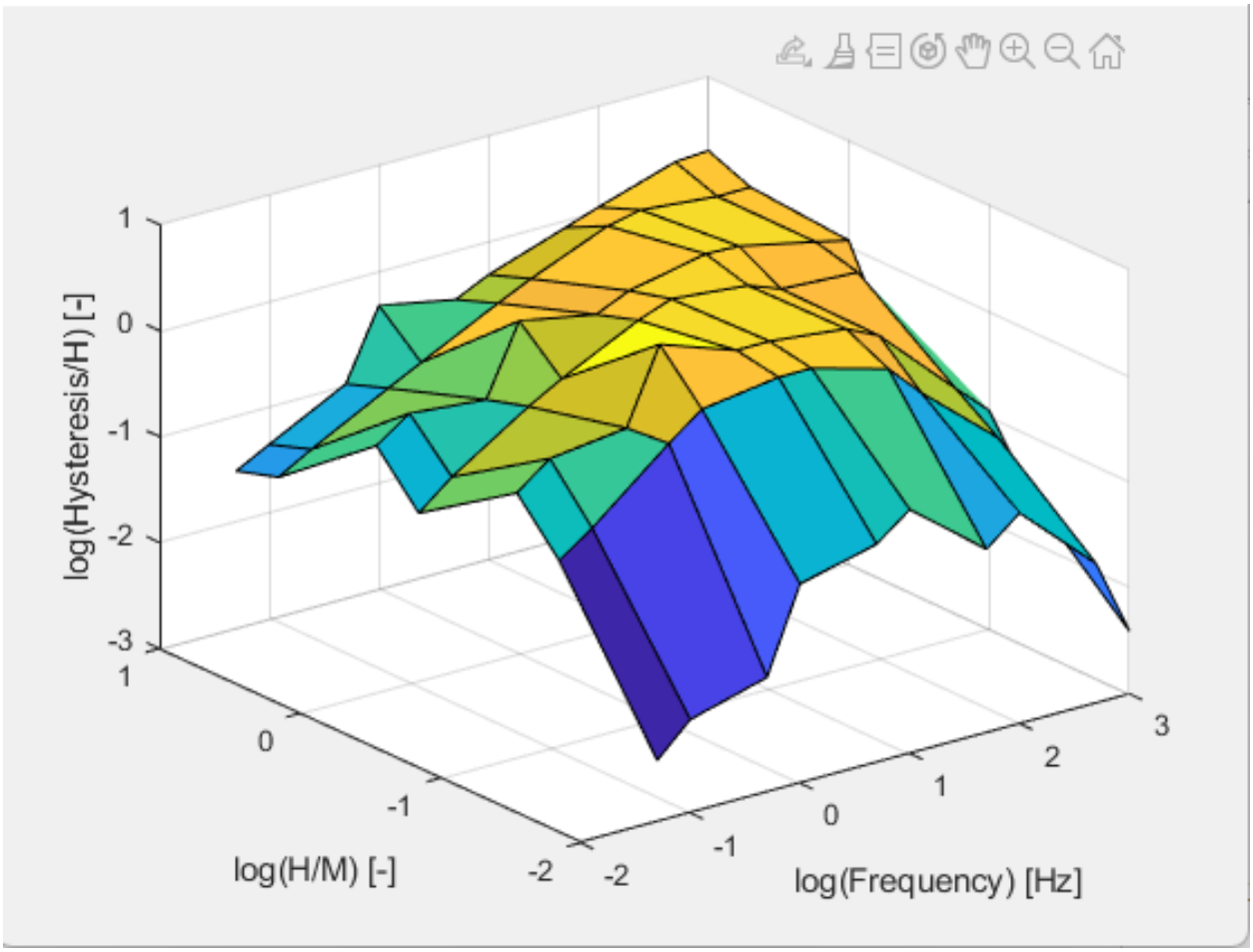

Figure 10. This plot shows hysteresis values on a logarithmic scale at different values of magnetic field strength and frequency, also on log scales. Hysteresis values are calculated as the area under the final hysteresis curve for each case, so only the final hysteresis is considered.

An exception to these ideas can be seen at very low field strength over the range of frequencies, with disorder increasing in the middle range of frequency, but falling once more as frequency increases further. This phenomenon can likely be attributed to the system's ability to keep up with the changes in magnetic field that occur because of the oscillation. In the first stage, at low frequency, the system is essentially memoryless, able to occupy the lowest possible energy state at every instance in time, because the rate of change of magnetic field strength is small enough for the system to respond to. As frequency increases, the system begins to lag the optimal state, which manifests as hysteresis in the field strength-

particle position plots. When the frequency becomes too great, however, the lag is such that the field switches before the particles have time to begin moving to occupy the optimal position, so the hysteresis is low once again because the particles do not have enough time to move out of the steady-state position and instead oscillate around it. Because the range of motion is limited, the position of the particle is also limited, so the hysteresis is low because the range of "swept-through" positions is low. This manifests as a flattened, long hysteresis loop seen in Figure 9, sections d, h and l.

At higher field strengths, this intermediate effect is less pronounced, because higher field strength means greater torque which means greater particle acceleration, so the particles can respond more quickly to changes and keep up better with the applied field. Nonetheless, the intermediate effect provides a way to see based on hysteresis when particles transition from one type of behavior to another. The peak hysteresis indicates a transition from some type of incomplete alignment with magnetic field to full alignment, because when particles are fully aligned, hysteresis increases with frequency and decreases with field strength. If this pattern changes, it indicates a change in behavior from the fully aligned motion.

Particles need a certain amount of energy from the external field to leave the low energy configuration that they form naturally when allowed to settle. The point at which the oscillating external field supplies this energy may be critical for determining the behavior of the system. The earlier this point is in the magnetic field cycle, the lower the hysteresis will be, as particles will be able to quickly move from their current state to the next optimal state and vice versa. For high field strength cases, the cycle reaches the critical field more quickly, as it is a property of the particles in the system and independent of field strength and frequency.

Magnetic systems in the simulation are able to occupy three different states, which are defined by critical values of field strength and frequency. The first, and simplest state, where particles align with the applied field, shows variation within it. Particles can approach this state, showcasing involatile memory as they grow closer and closer to full alignment over successive cycles of field. They can also show simply volatile memory, moving with the magnetic field to remain aligned, but always lagging it slightly. Involatile

memory is crucial for neuromorphic applications, as successive stimuli elicit a stronger response from the system.

The states of behavior can also be compared to those found in studies mentioned earlier, such as the flocking and vortex behavior observed in [8]. In addition to the fully aligned case, two other unique patterns can be observed in the simulation results. The perpendicular pattern, where particles oscillate in rows of aligned particles which are perpendicular to the applied field, and the diamond pattern, where particles settle into a steady state configuration and oscillate around this position with the applied field. This too could be important for neuromorphic applications, functioning as a coarse "switch" between different responses dependent on strength and frequency of stimuli.

Physical Model Results

The physical model similarly showed the ability for the magnetic system to transition between different types of behavior. These behaviors are organized by combination of field strength and frequency of the applied field, as can be seen in Figure 11. For the physical model, the energy total energy of the system at each moment in time was plotted. Total energy was composed of two components, the Zeeman energy, which is the energy that arises between a dipole and an applied field, and dipole energy, which is the energy caused by interactions between the particles.

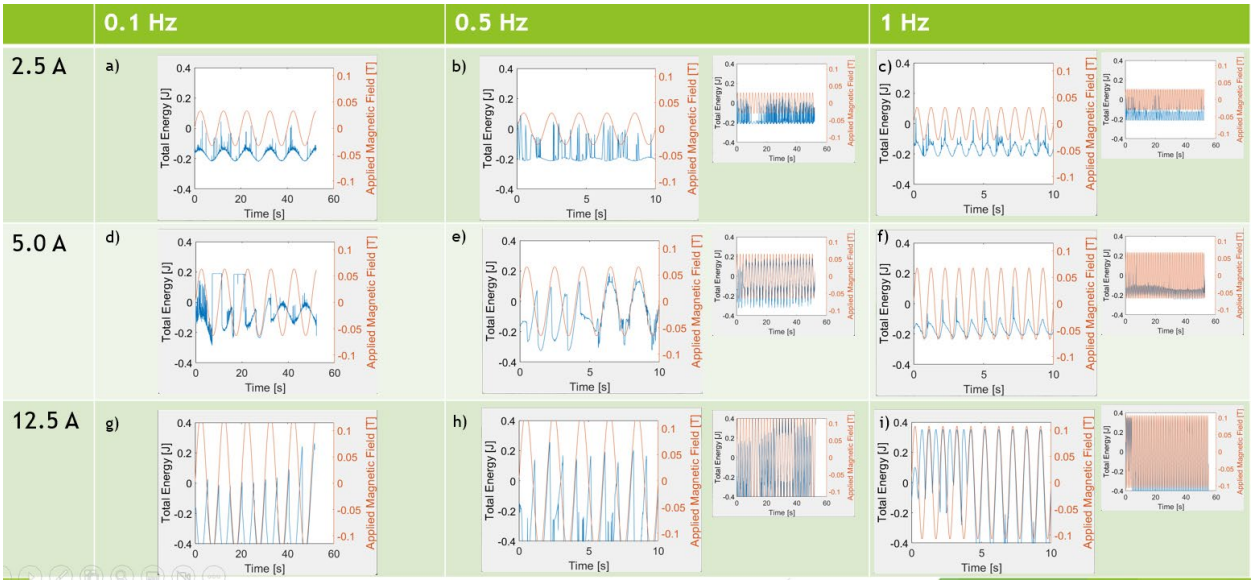

Figure 11. The total energy for each test of the physical model is plotted against time and represented in blue, with the applied field in red. The frequency is varied from left to right, with the low frequency on the left side. The field strength, represented by the peak current supplied to the electromagnet, is varied from top to bottom, with the low current at the top. Each test was run for the same amount of time, 50 seconds, so the inset graphs represent the full test, while the larger graphs zoom in on the first 10 seconds of each test so the behavior can be seen more clearly.

In contrast to the simulation results, the physical magnets maintained the initial position for low field strengths, exhibiting little to no response. The energy of the system is relatively constant, regardless of the applied field. The small oscillations in energy come from the Zeeman energy, which oscillates with the applied field. If the particles were perfectly arranged in a diamond pattern, this energy would be zero, because the particles are found in orientations which exhibit destructive interference with respect to the energy from the external field. Because they are not in exactly this pattern when seen by the image processing software, there is some slight oscillation in Zeeman energy, while the dipole energy remains constant as the particles remain in a set configuration.

As the field strength increased, however, the magnets began to behave in much the same way as the simulation, aligning in the direction of the applied magnetic field. At low frequencies, the particles were able to keep up with the applied field, aligning immediately with each directional change. In this case, the Zeeman energy dominates, causing the energy to oscillate in time with the magnetic field in a pattern similar to $\sin^2 x$. At intermediate field strengths and higher frequencies, the particles followed the motion of the

applied field less closely. At a certain point, the particles would no longer be able to keep up with the oscillation of the field and would instead rest in a parallel, unstable configuration while the field continued to oscillate. In terms of energy, this results in the transition of the total energy from behavior akin to the $\sin^2 x$ function to simple sinusoidal behavior, where the total energy oscillates in time with the applied field. Some intermediate frequency systems would recover, eventually returning to oscillate with the applied field, while in the high frequency cases, the particles would remain in the unstable position until the end of the test.

One final type of behavior observed in the physical modelling was that particles would oscillate around the diamond-shaped initial condition, never aligning fully with the applied field, but exhibiting some form of motion rather than remaining completely still. This behavior can be observed in the 5 Amp, 1 Hz case, where the energy exhibits an oscillation, but a non-sinusoidal one, indicating that the particles are moving with the applied field rather than stationary. The sum of oscillations in both the Zeeman energy and dipole energy causes this unusual behavior.

Critical Field Calculation and Significance

The critical field strength, a concept discussed in both the simulation and physical model sections of this thesis, is a conceptualization of the field strength at which it is more energetically favorable to move between states of the system. In the case of the physical system, the critical field can be calculated by setting the amount of energy of the particles in the aligned "parallel" configuration equal to the amount of energy the particles hold in the diamond configuration. The total energy in each configuration is the sum of the dipole energy of each particle and the sum of the Zeeman energy of each particle.

The dipole energy can be calculated from the equation:

(3) $U_{nm} = \frac{-\mu_0}{4\pi r^3}\left(3(MVd\boldsymbol{m}\cdot\hat{\boldsymbol{r}}) * (MVd\boldsymbol{n}\cdot\hat{\boldsymbol{r}}) - MV(d\boldsymbol{n}\cdot d\boldsymbol{m})\right)$,

where U_{nm} is the energy between particles n and m, $d\boldsymbol{n}$ is a direction vector in the direction of the nth particle magnetization, $d\boldsymbol{m}$ is a unit vector in the direction of the nth particle magnetization, and other variables are represented as defined in the method section calculations for local magnetic field. The Zeeman energy is calculated from the equation:

(4) $Z = -\mu_0 MV(d\boldsymbol{n} \cdot \boldsymbol{H})$,

where $\boldsymbol{H}$ is the applied field vector.

Thus, the total energy equivalency can be expressed with the following equation:

(5) $U_{para} + Z_{para} = U_{dia} + Z_{dia}$

Where equation (5) applies, the magnitude of the applied magnetic field in the equation for Zeeman energy can be designated as the critical field, as this energy is what causes the parallel state to become more energetically favorable than the diamond state. Therefore, by solving equation (5) for the applied field from the Zeeman energy equation, a "critical field" which marks the transition point between which state is energetically favorable can be found.

As discussed in the physical model section, the Zeeman energy for the diamond configuration is zero because of the rotationally symmetric particle distribution. Thus, the equation reduces to $Z_{para} = U_{dia} - U_{para}$. In the parallel configuration, the applied field and the particles are aligned, so the dot product in the equation for Zeeman energy becomes simple multiplication. Therefore, the critical field can be found from $H_{crit} = \frac{U_{para} - U_{dia}}{\mu_0 MV}$.

The equations for dipole energy are more complicated and thus will not be developed here, but by summing the dipole energy of each particle, which is in turn a summation of the dipole energy from every particle in the system to every other, the dipole energy for both configurations can be obtained. Since the dipole energy between two particles is the same for both particles, a simplified expression for dipole energy can be obtained. For a four-particle system, the total dipole energy of the system is equal to $U_{tot} = 2U_{12} + 2U_{13} + 2U_{14} + 2U_{23} + 2U_{24} + 2U_{34}$.

After performing the calculations for the physical system, the critical field was determined to be around 11,000 A/m, or converted to units of magnetic field, 13.7 mT. In the case of the custom electromagnet, this corresponds to an amperage supply of around 1.25 A, although in experiment, particles required more than 2.5 A of current to respond to the applied magnetic field, as evidenced by the total energy graphs pictured in Figure 11. This difference is likely attributable to factors not included in the simple energy analysis done in the critical field calculation, such as the effect of friction on the particles, which could cause additional energy to be added to the system before the particles will move from one state to another.

A similar set of calculations can be performed for the set of particles in the simulation, with a few key differences. Because the particle matrix in the simulation is 5 by 5, there is Zeeman energy associated with the diamond configuration, because the pattern cannot be repeated symmetrically throughout an odd matrix length. Critical field can still be obtained by factoring it out of the terms for Zeeman energy. As the size of the matrix increases, the dipole energy calculations become more complicated and would be too complex to reproduce here.

Calculations for critical field were performed using MATLAB, code for which can be found in Appendix C. The critical field for the particles in the simulation was found to be 9561.4 A/m, which is equivalent to 0.0252 times the particle magnetization. Applying this analysis to the simulation results, it is notable that none of the tests where the magnetic field strength is 0.01 times the magnetization exhibit full alignment, which seems to confirm critical field as a useful measure for determining the point at which the system behavior changes. At high frequency and higher field strengths, full alignment is also not achieved, so it is not the only determinant of the change in behavior.

Additionally, the point at which the cycle reaches the critical field could be important to determining the response of the system to the applied field. In low field strength cases, the critical field is reached later in the sinusoidal cycle, which means that the time the system has to respond to a change in energy is less than in higher field strength cases. For example, in the simulation case where frequency is

1000 Hz and field strength is 0.1 times the particle magnetization, particles do not reach full alignment, instead, oscillating around their fixed positions and cycling to a local pattern. In this case, the proportion of the sinusoid where the field strength is above the critical field strength is found through the following equations. Critical field strength is found at 0.0252 times particle magnetization regardless of case or frequency. Normalizing this value by the peak field strength, we get that the critical field is reached when $\sin 2\pi f t = \frac{0.0252}{0.1}$, or when $t = \frac{\text{asin}(0.252)}{2000\pi} = 0.4054 * 10^{-4}\ s$. The peak field strength is reached at $t = \frac{1}{4f} = \frac{1}{4000} = 2.5 * 10^{-4}$. Therefore, the time above critical field is $2 * (2.5 * 10^{-4} - 0.405 * 10^{-4}) = 4.2 * 10^{-4}\ s$. From these calculations, it is clear that the time above critical field is inversely proportional to the frequency of the applied field. In order to undergo complete alignment, there may be a minimum time above critical field required, which may explain the effect of frequency on the different behaviors seen in these experiments. The effect of maximum field strength on time above critical field is nonlinear, which may explain the nonlinear relationship between maximum field strength, frequency, and hysteresis.

It must be noted that this critical field analysis ignores one of the cases apparent in the simulation which is common at low field strengths and frequencies. The perpendicular case is not accounted for in these calculations, but it is possible that the critical field for the perpendicular case is lower than the critical field for the parallel case which can cause it to appear in low field cases, or that the more complicated dynamics in larger systems cause this behavior to emerge.

Physical Model and Simulation Comparison

The cases where the particles completely rotate to align with the magnetic field and the case where the particles rotate around a fixed diamond configuration are found in both the physical model and the simulation. One configuration which is not found in the physical model is one where particles align in rows perpendicular to the applied field. It is possible that because of the limited number of particles in the

physical model, this configuration is less likely, requiring multiple rows of particles to be able to enter that pattern.

The physical model does not have any fluid drag as is found in the simulation, instead with some friction occurring between the magnets in the housing. These types of energy dissipation manifest in different ways, which could explain some of the behavior in the physical model. For instance, in some of the higher frequency cases, the particles stop after following the field for a few cycles. Because of static friction, if the magnets become out of phase with the oscillating field, it could require more force to return them to motion, resulting in a higher critical field than previously calculated.

When particles are anti-parallel to the applied field, a conditionally stable state occurs, because magnetic torque is a cross product between the particle magnetization and the applied field. If the particles are anti-parallel, the cross product is zero, even though the state is high energy. Interparticle interaction can provide the force necessary for the particles to move from this conditionally stable state, aligning once more with the external field. Thus, any hysteresis and oscillation observed in this study is due to interparticle interaction overcoming the conditionally stable state and destabilizing the parallel configuration enough to allow the applied field to take effect. Since the interparticle repulsion can occur over a longer period of time in the lower frequency case, it may be possible for lower frequencies to return to the particles to oscillation, essentially returning the system to “in phase” behavior, whereas in the higher frequency case, the interparticle interaction is not able to overcome the static friction and destabilize the particles before the field has reversed directions and is an aligning field once more.

Finally, the inertia of the magnets is much higher than the inertia of the physical particles. It is therefore not accurate to assume that the particles will respond instantaneously to changes in magnetic field as it is in the simulation case, which is another reason that the particles may become out of phase with the applied field in the physical system, but not the simulation.

Chapter 4

Conclusions and Future Work

The magnetic systems investigated in this thesis showcased distinct behaviors depending on the field strength and frequency of the input driving magnetic field introduced to the system. The magnetic interactions between particles were responsible for creating some of these patterns, such as the oscillatory diamond pattern and patterns of particles perpendicular to the applied field. In addition, interparticle interaction was critical to allowing the particles to fully align with the applied field and then subsequently switch directions when the field switches directions.

A smaller, physical model confirmed some of the behavior observed in the simulation, with slight differences as a result in differences in the energy dissipating elements in each system as well as the differences in inertia and magnetization strength between the magnets and particles.

These systems demonstrate characteristics of neuromorphic materials, showing both volatile and involatile hysteresis in the simulation case. These hysteresis loops can be manipulated based on the input field, which demonstrates that the memory capacity of the system can be changed with differences in input.

Models like these could be used to aid in research regarding causes of dementia. Dementia occurs when connections between neurons break down and cause brain cells to no longer be able to communicate with one another. The memory capacity of the magnetic system in this thesis is dependent on the frequency and magnitude of the input magnetic field, which correlate to the frequency and strength of the electrical signals in biological systems. This model provides a potential methodology for understanding what causes the memory capacity in biological systems to diminish – the strength or frequency of the electrical signal decreases. By increasing our understanding of systems like these, we can better understand what causes them to fail, transition

from one state to another, or work together in networks. This knowledge can be used to create therapies for dementia patients with a better understanding of what is the core problem, or provide a way for compensate for a weakness in one aspect of memory capacity with another which may not have suffered the same damage, like medications which can enhance the frequency of neuron signals when their strength has decreased, or vice versa.

While this system models an individual neuron with an externalized input and output, neurons in the brain work together to accomplish the thinking process that is writing this paper. A single neuron is easy to understand – it is the complex network of neurons that still makes the inner workings of the brain a mystery. Another potential application for neuromorphic materials is to create larger-scale physical networks where individual pockets of neuromorphic material work together to create a network. Such networks would have the advantage of being able to store information and perform calculations in the same location, moving beyond the digital computer architecture that makes up the basis of our current infrastructure. Passing a signal from one neuromorphic core to another is the next step in this type of architecture and coordinating multiple of these cores like true neurons is another application of the phenomena outlined here, made possible because of the demonstrated memory capacity the system shows and its ability to change behavior based on the input signal.

While the systems show promise for memristive applications, work still needs to be done to transform the output of the system in terms of particle direction into an input that can be used in neuromorphic materials. Ideally, the particle oscillations would correlate to an input field and frequency to another system of magnetic particles, so that chains of these systems could be established to drive one another and form stronger connections between them, like neurons do.

Additionally, while there is relatively good agreement between them, not all of the behaviors observed in the simulation occurred in the physical model. Introducing more magnets to the physical model could help clarify the role of particle count in these systems, as well as verify the perpendicular pattern seen in the simulation.

The relationship between inertia and frequency could also provide an interesting direction for study. While the same behaviors can be seen in both the physical model and simulation, these behaviors do not occur at the same field strengths and frequencies. Finding a way to relate the two systems is crucial to being able to apply the simulation results to any physical system, as well as scale the model for different applications.

Finally, the role of drag in the behavior of the system has not been tested. Applying some type of viscous resistance to the physical model would help to clarify the effect of drag on key system behaviors such as hysteresis. Increasing or decreasing the amount of drag could similarly have effects on the amount of hysteresis the system exhibits and which formations become more likely. Different amounts of drag could also change which patterns present themselves at a given field strength or frequency, which could be useful in scaling the system to work with physical electromagnets, which cannot provide the infinite variety of field strengths and frequencies that a simulation can.

Appendix A

Numerical Simulation Code

```
function [data] =
DPM3DconvE_NoTranslation_NoAcceleration_Summer2021(defaultN, volFrac, frac,
freq, folder, saveFile, seed, varargin)
%% Input Parsing

% This function was created to accept different types of input:
tic
p = inputParser;

% Any input that is not defined will take its corresponding following
% value:
defaultN    = defaultN;%5 ; % numberof particles
defaultvol  = volFrac;
defaultHtoM = 5; % ratio of magnetic to electric fields
defaultR    = 0.5*10^-6;
defaultrhok = 5.28*(1/1000)*(100/1)^3;% density of particles
defaultM    = 72*defaultrhok; % magnetization
defaulteta  = 3500*(1/1000); % Dynamic viscosity of the medium [Poise]
defaulttmax = 6000000*3;
defaultmaxit= 10000;

% This function can take as input, its own output. If this option is used,
% the simulation will continue wherever it left off the previous run.
defaultdata = 0;

addOptional(p,'N',defaultN,@isnumeric);
addOptional(p,'vol',defaultvol,@isnumeric);
addOptional(p,'R',defaultR,@isnumeric);
addOptional(p,'M',defaultM,@isnumeric);
addOptional(p,'HtoM',defaultHtoM,@isnumeric);
addOptional(p,'eta',defaulteta,@isnumeric);
addOptional(p,'tmax',defaulttmax,@isnumeric);
addOptional(p,'maxit',defaultmaxit,@isnumeric);
addOptional(p,'data',defaultdata);

parse(p,varargin{:});

%% Initialization

datain = p.Results.data;
tmax = p.Results.tmax;
maxit= p.Results.maxit;

if isnumeric(datain)==0
```

```
    Param = datain.Parameters;
    N     = Param(1);
    vol   = Param(2);
    HtoM  = Param(3);
    R     = Param(4);
    dim   = Param(5);
    M     = Param(6);
    eta   = Param(7);

    t = datain.t1(end);

    X = datain.X(:,:,end);
    D = datain.D(:,:,end);

    Xdata       = datain.X;
    Ddata       = datain.D;
    tdata       = datain.t1;
    g           = datain.g;
    f           = datain.f;
    previous_i  = length(datain.t1);

else

    N    = p.Results.N;
    vol  = p.Results.vol;
    HtoM = p.Results.HtoM;
    R    = p.Results.R;
    dim  = ceil(sqrt(N))*R*1.5; %*3; %*1.5; %(N*(4/3)*pi*R^3/vol)^(1/3)/2;
    M    = p.Results.M;
    eta  = p.Results.eta;

    t      = 0;

    fid = fopen('initData.txt','wt');
    fprintf(fid,'%g\t %g\t %g\t' ,N,R,dim);
    fprintf(fid,'\n');
    fclose(fid);

    % Randomly generate direction vectors for each particle
    rng(seed)
    D = zeros(3, N);
    D(1, :) = rand(1, N);
    D(2, :) = sqrt(1 - D(1, :).^2);
    D = D./(ones(3,1)*sum(D.^2).^(1/2));

    % Space each particle ~1 particle radius away from each other in a grid
    X = ones(3, N);
    interval = ceil((N)^(1/2));
    step = 2*dim/(interval);
    index = 1;
    for i = 1:interval
        for j = 1:interval
                if index <= N
```

```
                    X(:, index) = [-dim + step/2 + step*(j-1), -dim + step/2
+ step*(i-1), 0];
                    index = index + 1;
                end
            end
        end


    % Store initial data for output
    Xdata(:,:,1)  = X;
    Ddata(:,:,1)  = D;
    g(1)          = zeros(1);
    f(1)          = zeros(1);
    tdata(1)      = 0;
    previous_i    = 1;
end

% Plotting conditions are defined
plotmult = 1000; % this determines the multiples of it at which the program
will plot the state
times = zeros(maxit/plotmult,1);

%%%%%%%% APPLY ELELCTRIC AND MAGNETIC EXTERNAL FIELDS %%%%%%%%%%
E0x = 0; E0y = 0; E0z = 0; %100*10^6; %applied ELECTRIC field [V/m]
H0x = 0; H0y = 0; H0z = 0; %HtoM*M; %applied MAGNETIC field [A/m]

% Declare initial condition for angular velocity
omega = zeros(3, N);

mu0   = 4*pi*10^(-7); %permeability of free space [N/A^2]
B     = mu0*H0z; %induced field

%E = [E0x, E0y, E0z]; % initial field directions
% Initial field setup
    Ex = E0x*ones(1,N); % total field in x
    Ey = E0y*ones(1,N); % total field in y
    Ez = E0z*ones(1,N); % total field in z
    E = [Ex; Ey; Ez];

%Particle and magnetic field properties:
Vequi    = 4/3*pi*R^3; %particle volume [m^3]
a = ((Vequi*(3/4)*3.5)/(pi))^(1/3); % Particle long axis
b = a;
c = a/3.5; % Particle short axis

V    = 4/3*pi*a*b*c; % [m^3] ellipsoidal particle volume

Lx = (a^2*c)/(2*(a^2 - c^2))*(((pi/2)/sqrt(a^2-c^2))-(c/(a^2))); %
depolarizing factor in x-dir
Ly = Lx; % depolarizing factor in y-dir
Lz = 1 - 2*Lx; % depolarizing factor in z-dir

eps0 = 8.85*10^-12; % [F/m] electric permittivity of vacuum
```

```
eps1 = 2*eps0; % electric permittivity of medium
eps2 = 10*eps0; % electric permittivity of particles

% Magnetic field
Ds    = 6*pi*eta*R; %Drag coefficient
eta   = p.Results.eta; %viscosity of the fluid [Pa*s]

Q    = mu0*(norm(M)*V)^2/(4*pi); %Magnetic force constant
QQQ  = mu0*M*V/(8*pi*eta*R^3); % Magnetic Torque constant
QQ   = 3*Q*norm(mu0*HtoM*M)^2/(8*R^4); %Repelling force constant
QQQe = 1/(8*pi*eta*R^3); % Electric torque
beta = 40; %Repelling parameter
J(1) = 1/5*(b^2 + c^2); %Moment of inertia of particles along a axis
J(2) = J(1); %Moment of inertia of particles along b axis
J(3) = 1/5*(a^2 + b^2); %Moment of inertia of particles along c axis
% frac = 1;
% freq = 1;
lineGrid = ones(1, N);

% Useful operators
corrector  = ones(N,N)-2*tril(ones(N,N),-1);
corrector2 = diag(inf(1,N));

dim2 = dim;
diffx=@(XI,XJ) diag(([XI(1)-XJ(:,1),XI(1)-XJ(:,1)+2*dim,XI(1)-XJ(:,1)-
2*dim])*(([abs(XI(1)-XJ(:,1))<=dim2,abs(XI(1)-XJ(:,1)+2*dim)<dim2,abs(XI(1)-
XJ(:,1)-2*dim)<dim2])'));
diffy=@(XI,XJ) diag(([XI(2)-XJ(:,2),XI(2)-XJ(:,2)+2*dim,XI(2)-XJ(:,2)-
2*dim])*(([abs(XI(2)-XJ(:,2))<=dim2,abs(XI(2)-XJ(:,2)+2*dim)<dim2,abs(XI(2)-
XJ(:,2)-2*dim)<dim2])'));
diffz=@(XI,XJ) diag(([XI(3)-XJ(:,3),XI(3)-XJ(:,3)+2*dim,XI(3)-XJ(:,3)-
2*dim])*(([abs(XI(3)-XJ(:,3))<=dim2,abs(XI(3)-XJ(:,3)+2*dim)<dim2,abs(XI(3)-
XJ(:,3)-2*dim)<dim2])'));

xunit=[ones(1,N);zeros(1,N);zeros(1,N)];
yunit=[zeros(1,N);ones(1,N);zeros(1,N)];
zunit=[zeros(1,N);zeros(1,N);ones(1,N)];

i = previous_i+1;
it= 1;
tf= 0;
ti= cputime;
h = 0; % ADDED - REMOVE LATER
maxDeflection = 0;
avg = zeros(1, 6);
std = zeros(1, 6);
%[avg(1), std(1)] = zAngles(D, 1);
Tz = zeros(1, N);
fib = [1, 1, 2, 3];
p = 1;
%% Iterative Solver

%Boundary condition implementation
```

```
Ixplus=find(dot(X,xunit)>dim);
Ixminus=find(dot(X,xunit)<-dim);
Iyplus=find(dot(X,yunit)>dim);
Iyminus=find(dot(X,yunit)<-dim);
Izplus=find(dot(X,zunit)>dim);
Izminus=find(dot(X,zunit)<-dim);

X(1,Ixplus)=X(1,Ixplus)-2*dim;
X(1,Ixminus)=X(1,Ixminus)+2*dim;
X(2,Iyplus)=X(2,Iyplus)-2*dim;
X(2,Iyminus)=X(2,Iyminus)+2*dim;
X(3,Izplus)=X(3,Izplus)-2*dim;
X(3,Izminus)=X(3,Izminus)+2*dim;

%Force and field calculations - Moved to accomodate new integration

rx  = squareform(pdist(X',diffx));
ry  = squareform(pdist(X',diffy));
rz  = squareform(pdist(X',diffz));

rx  = (rx.*corrector)';
ry  = (ry.*corrector)';
rz  = (rz.*corrector)';

r   = (rx.^2+ry.^2+rz.^2).^(1/2)+eye(N);
phi = r/2/dim;

rhatx = rx./r;
rhaty = ry./r;
rhatz = rz./r;

while t<1/freq*5 %20 %it<maxit && tf<tmax*60 && t<0.5

while t<i*0.0001 && t<1/freq*5 %20

    % Create a set of 4 k-values (pages) for the Runge-Kutta 4 approximation
of the particle direction, with a
    % vector (cols) for each of the N particles (rows)
    kD = zeros(3, N, 4);

    % Create a set of 4 k-values (pages) for the Runge-Kutta 4
    % approximation for the angular velocity,
    % with a vector (cols) for each of the N particles (rows)
    %kW = zeros(3, N, 4);

    % Declare the time step
    %h = abs(t-i*0.0001);

    if it == 1
        h = 10^-11;
    else
        h = min([10^-5/max(max(T)),abs(t-i*0.0001)]);
    end
```

```
    for j = 1:4

        % Determine a scaling factor for time step for direction Runge Kutta
approximation
        if j == 1
            timeScale = 0;
        elseif j < 4
            timeScale = 0.5;
        else
            timeScale = 1;
        end

        % Create a temporary direction for use in calculation of k values for
        % Runge Kutta
        tempD = D + h*timeScale*kD(:, :, fib(j));
        tempD = tempD./(ones(3,1)*sum(tempD.^2).^(1/2));

        Dnx = ones(N,1)*tempD(1,:);
        Dny = ones(N,1)*tempD(2,:);
        Dnz = ones(N,1)*tempD(3,:);

        Dmx = tempD(1,:)'*ones(1,N);
        Dmy = tempD(2,:)'*ones(1,N);
        Dmz = tempD(3,:)'*ones(1,N);

        DnR = rhatx.*Dnx+rhaty.*Dny+rhatz.*Dnz;
        DmR = rhatx.*Dmx+rhaty.*Dmy+rhatz.*Dmz;

        DmDn = Dnx.*Dmx + Dny.*Dmy+ Dnz.*Dmz;


        H0y = frac*M*sin((t+timeScale*h)*2*pi*freq + (freq == 0)*pi/2);

        % Magnetic Field calculations
        Hddx = (M*V/4/pi)*sum((1./r.^3-eye(N)).*(3*DmR.*rhatx-Dmx));
        Hddy = (M*V/4/pi)*sum((1./r.^3-eye(N)).*(3*DmR.*rhaty-Dmy));
        Hddz = (M*V/4/pi)*sum((1./r.^3-eye(N)).*(3*DmR.*rhatz-Dmz));

        Hx = H0x*lineGrid + Hddx;
        Hy = H0y*lineGrid + Hddy;
        Hz = H0z*lineGrid + Hddz;

        % The dot product of D and H
        DnH = Dnx(1,:).*Hx + Dny(2,:).*Hy+ Dnz(2,:).*Hz;

        %Eh = -mu0*M*V*(Hx.*D(1,:)+Hy.*D(2,:)+Hz.*D(3,:));

        % Calculate angular velocity due to magnetic field assuming zero
angular acceleration
        Wx = (QQQ)*(Hx-Dnx(1,:).*DnH);
        Wy = (QQQ)*(Hy-Dny(1,:).*DnH);
```

```
        Tx = Wx; % + Wex;
        Ty = Wy; % + Wey;

        T = [Tx;Ty;Tz];

        kD(:, :, j) = T;
    end

    % Calculate the new omega

    %Calculate the new D
    D = D + 1/6*h*(kD(:, :, 1) + 2*kD(:, :, 2) + 2*kD(:, :, 3) + kD(:, :,
4));
    D = D./(ones(3,1)*sum(D.^2).^(1/2));
    t  = t+h;

    maxDeflection = 180/pi*max(acos(sum(D(:,:).*Ddata(:,:,1),
2)./(sqrt(sum(D(:,:).^2,2)).*sqrt(sum(Ddata(:,:,1).^2,2)))));

        %Increasing the loop parameters
    it = it+1;

        % Keeping track of running time
    tf = cputime-ti;

    % Determine if the iteration is one that should be plotted
    plotcond = rem(it,plotmult);
    if plotcond == 0 && it ~= 1
%         it
%        t
        % Gather the standard deviation and average angle and velocity for
the given
        % iteration
       [xAvg(it/plotmult), xStd(it/plotmult)] = xAngles(D, 0);
       times(it/plotmult) = t;
       [yAvg(it/plotmult), yStd(it/plotmult)] = yAngles(D, 0);
       times(it/plotmult) = t;

       yPos(it/plotmult, :) = acos(D(2, :));

       xOm(it/plotmult) = mean(T(1, :));
       yOm(it/plotmult) = mean(T(2, :));
       Ddata(:, :, it/plotmult + 1) = D;
    end

end

    tdata(i) = t;

    DDdata(:,:) =  ((1./phi.^3-eye(N)).*(3*DmR.*rhatx-Dmx)).*Dnx...
                    +((1./phi.^3-eye(N)).*(3*DmR.*rhaty-Dmy)).*Dny...
                    +((1./phi.^3-eye(N)).*(3*DmR.*rhatz-Dmz)).*Dnz;
```

```
    f(i) = (1/4/pi/N^2)*sum(sum(DDdata(:,:)));
    g(i) = (1/N)*sum(D(3,:));

    i = i+1;
end

    Xdata(:,:,2)  = X;
    Ddata(:,:,length(Ddata(1, 1, :)) + 1) = D;

    % Determine the radial frequency
    radFreq = freq*2*pi;

    % Plot the average particle orientation over time +- 1 stdev
    f(1) = figure(3);
    plot(times, yAvg./pi)
    xlabel('Time')
    axis([0 1/freq*5 -0.01 1.01])
    hold on

    plot(times, yAvg./pi + yStd./pi)
    plot(times, yAvg./pi - yStd./pi)

    totalField = frac*M*sin((times)*2*pi*freq + (freq == 0)*pi/2);

    yyaxis right
    plot(times, totalField)
    hold off

    % Plot normalized x-position versus x-velocity
    f(2) = figure(4);
    figure(4)

    plot(xAvg./pi, xOm./radFreq)
    xlabel('Average particle angle wrt x-axis')
    ylabel('Average particle velocity in x direction')

    % Plot normalized y-position versus y-velocity
    f(3) = figure(5);
    figure(5)

    plot(yAvg./pi, yOm./radFreq)
    xlabel('Average particle angle wrt y-axis')
    ylabel('Average particle velocity in y direction')

    % Plot x- and y- position versus total velocity
    f(4) = figure(6);
    figure(6)

    %[A, B] = meshgrid(xAvg, yAvg);
    plot3(xAvg./pi, yAvg./pi, sqrt(xOm.^2 + yOm.^2)./radFreq);
    xlabel('Average particle angle wrt x-axis');
    ylabel('Average particle angle wrt y-axis');
```

```
    zlabel('Average magnitude of particle velocity');

    % Plot field strength versus particle orientation
    f(5) = figure(7);
    figure(7);
    plot(totalField, yAvg./pi)
    xlabel('Magnetic Field Strength (A/m)')
    ylabel('Average particle angle wrt y-axis');
    % Create a cell array of file names to save all the figures
    file = cell(5);
    file{1} = '_OrientationOverTime_5Cycles.fig';
    file{2} = '_XPos_XVel.fig';
    file{3} = '_YPos_YVel.fig';
    file{4} = '_XYPos_MagVel.fig';
    file{5} = '_Hysterisis.fig';
%     file{5} = '_Tile.fig';

%    Save all figures
    for i = 1:5
        saveas(f(i), [pwd, folder, saveFile, file{i}]);
        close(f(i));
    end

    % Save the data
    save([pwd, folder, 'Magnetization', num2str(frac), '_Frequency',
num2str(freq), '-10_', num2str(N), 'Particles.mat'], 'N', 'X', 'D', 'yAvg',
'yStd', 'times', 'totalField', 'xOm', 'yOm', 'xAvg', 'yAvg', 'yPos');

    H0 = [H0x; H0y; H0z];
    E0 = [E0x; E0y; E0z];


data=struct('Parameters',[N,vol,HtoM,R,dim,M,eta],'X',Xdata,'D',Ddata,'g',g,'
f',f,'t1',tdata,'E0',E0,'H0',H0,'dim',dim, 'totalField', totalField);
end
```

Appendix B

Dot Tracking Code

```
% This script analyzes an image to get the x-y position of all red dots
clc, clear

I = 12.5;
f = 1;
v = VideoReader(['PhysicalModel_Current', num2str(I), '_Frequency',
num2str(f), '.mp4']); %'_Zoomed.mp4']);
loop = v.numFrames;
dotAnim = cell(1, loop);
dot1 = zeros(loop, 2);
dot2 = zeros(loop, 2);
dot3 = zeros(loop, 2);
dot4 = zeros(loop, 2);

% This is the mask for an iPhone video
% xi1 = [512, 160, 160, 512];
% yi1 = [613, 613, 252, 252];
% xi2 = [512, 512, 886, 886];
% yi2 = [613, 252, 252, 613];
% xi3 = [512, 512, 160, 160];
% yi3 = [613, 978, 978, 613];
% xi4 = [512, 886, 886, 512];
% yi4 = [613, 613, 978, 978];

% This is the mask for a video from the camera.
xi1 = [651, 651, 459, 459];
yi1 = [367, 167, 167, 367];
xi2 = [651, 851, 851, 651];
yi2 = [367, 367, 167, 167];
xi3 = [651, 459, 459, 651];
yi3 = [367, 367, 570, 570];
xi4 = [651, 651, 851, 851];
yi4 = [367, 570, 570, 367];

mask1 = poly2mask(xi1, yi1, 720, 1280);
mask2 = poly2mask(xi2, yi2, 720, 1280);
mask3 = poly2mask(xi3, yi3, 720, 1280);
mask4 = poly2mask(xi4, yi4, 720, 1280);

for p = 1:loop
    frame = readFrame(v);
    [~, maskedFrame] = createMask(frame);

    for s = 1:4
        if s == 1
            mask = mask1;
        elseif s == 2
            mask = mask2;
        elseif s == 3
```

```
            mask = mask3;
        else
            mask = mask4;
        end
        redChannel = maskedFrame(:, :, 1);
        redChannel = redChannel & mask;
        clear redDots
        redDots = [];
        index = 1;
        for r = 1:length(redChannel(:, 1))
            for c = 1:length(redChannel(1, :))
                if redChannel(r, c) ~= 0
                    redDots(index, :) = [c, r];
                    index = index + 1;
                end
            end
        end

        % Here, take pixels from redDots and use to determine x-y locations
of
        % the red dots in the video
        dots = {};
        del = [1];
        k = 1;
        while length(redDots) > 2
            dots{k} = redDots(1, :);
            for m = 2:length(redDots)
                if abs(redDots(1, 1) - redDots(m, 1)) < 45 && abs(redDots(1,
2) - redDots(m, 2)) < 45
                    dots{k} = [dots{k}; redDots(m, :)];
                    del = [del, m];
                end
            end
            redDots(del, :) = [];
            del = [1];
            k = k + 1;
        end

        if length(dots) == 0
            dots{1} = [0, 0];
        end

        ind = 1;
        for w = 2:length(dots)
            if length(dots{w}) > length(dots{ind})
                ind = w;
            end
        end

        if s == 1
            dot1(p, 1) = sum(dots{ind}(:, 1))/length(dots{ind}(:, 1));
            dot1(p, 2) = sum(dots{ind}(:, 2))/length(dots{ind}(:, 2));
        elseif s == 2
            dot2(p, 1) = sum(dots{ind}(:, 1))/length(dots{ind}(:, 1));
            dot2(p, 2) = sum(dots{ind}(:, 2))/length(dots{ind}(:, 2));
```

```
        elseif s == 3
            dot3(p, 1) = sum(dots{ind}(:, 1))/length(dots{ind}(:, 1));
            dot3(p, 2) = sum(dots{ind}(:, 2))/length(dots{ind}(:, 2));
        else
            dot4(p, 1) = sum(dots{ind}(:, 1))/length(dots{ind}(:, 1));
            dot4(p, 2) = sum(dots{ind}(:, 2))/length(dots{ind}(:, 2));
        end
    end

    close
end
fps = 60;
time = (1:loop)./fps;

circle1 = dot1(1, :);
circle2 = dot2(1, :);
circle3 = dot3(1, :);
circle4 = dot4(1, :);

for q = 2:loop
    if abs(dot1(q, 1) - dot1(q - 1, 1)) > 15 || abs(dot1(q, 2) - dot1(q - 1,
2)) > 15
        if dot1(q, 1) ~= 0 || dot1(q, 2) ~= 0
            circle1 = [circle1; dot1(q, :)];
        end
    end
    if abs(dot2(q, 1) - dot2(q - 1, 1)) > 15 || abs(dot2(q, 2) - dot2(q - 1,
2)) > 15
        if dot2(q, 1) ~= 0 || dot2(q, 2) ~= 0
            circle2 = [circle2; dot2(q, :)];
        end
    end
    if abs(dot3(q, 1) - dot3(q - 1, 1)) > 15 || abs(dot3(q, 2) - dot3(q - 1,
2)) > 15
        if dot3(q, 1) ~= 0 || dot3(q, 2) ~= 0
            circle3 = [circle3; dot3(q, :)];
        end
    end
    if abs(dot4(q, 1) - dot4(q - 1, 1)) > 15 || abs(dot4(q, 2) - dot4(q - 1,
2)) > 15
        if dot4(q, 1) ~= 0 || dot4(q, 2) ~= 0
            circle4 = [circle4; dot4(q, :)];
        end
    end
end

%% Curve fit the points gathered using area fitting.

index1 = find(dot1 == [0, 0]);
index2 = find(dot2 == [0, 0]);
index3 = find(dot3 == [0, 0]);
index4 = find(dot4 == [0, 0]);

index1 = index1(1:(length(index1)/2));
```

```
index2 = index2(1:(length(index2)/2));
index3 = index3(1:(length(index3)/2));
index4 = index4(1:(length(index4)/2));

index = [index1; index2; index3; index4];
index = sort(index);

w = 2;
while w <= length(index)
    if index(w) == index(w - 1)
        index(w) = [];
    else
        w = w + 1;
    end
end

dot1(index, :) = [];
dot2(index, :) = [];
dot3(index, :) = [];
dot4(index, :) = [];
time(index) = [];

for t = 1:4
     figure(t)
    if t == 1
        x0 = 553;
        y0 = 270;
    elseif t == 2
        x0 = 753;
        y0 = 270;
    elseif t == 3
        x0 = 553;
        y0 = 465;
    else
        x0 = 753;
        y0 = 465;
    end

    if t == 1
        shifted1 = dot1 - [x0, y0];
        angle1 = atan2(shifted1(:, 2), shifted1(:, 1));
        plot(time, angle1);
        axis([0 loop/fps -pi pi])
    elseif t == 2
        shifted2 = dot2 - [x0, y0];
        angle2 = atan2(shifted2(:, 2), shifted2(:, 1));
        plot(time, angle2);
        axis([0 loop/fps -pi pi])
    elseif t == 3
        shifted3 = dot3 - [x0, y0];
        angle3 = atan2(shifted3(:, 2), shifted3(:, 1));
        plot(time, angle3);
        axis([0 loop/fps -pi pi])
    else
        shifted4 = dot4 - [x0, y0];
```

```
        angle4 = atan2(shifted4(:, 2), shifted4(:, 1));
        plot(time, angle4);
        axis([0 loop/fps -pi pi])
    end

end
%% Calculate dipole energy
% Parameters for M, V, and a will need to be defined. angles 1-4 can be
% converted to unit vectors in the direction of the angle to find dm and dn
% r will be different for each angle.
dist = 18.4*10^-3; % [m]
mu0 = 4*pi*10^-7; % [N/A^2]
M = 13200*10^-4/mu0; % [A/m]
V = ((0.5*25.1).^2*(0.25*25.1) - pi/4*3.3^2*0.25*25.1)*10^-9; % Countersunk
magnet volume [m^3]

dipoleEnergy1 = zeros(length(angle1), 1);
dipoleEnergy2 = dipoleEnergy1;
dipoleEnergy3 = dipoleEnergy1;
dipoleEnergy4 = dipoleEnergy1;

for z = 1:length(angle1)
    dipoleEnergy1(z) = -mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle2(z)),
sin(angle2(z))], [1, 0])*dot(M*V*[cos(angle1(z)), sin(angle1(z))], [1, 0]) -
dot(M*V*[cos(angle2(z)), sin(angle2(z))], M*V*[cos(angle1(z)),
sin(angle1(z))]));
    dipoleEnergy1(z) = dipoleEnergy1(z) + -
mu0/(4*pi*(dist*sqrt(2)).^3)*(3*dot(M*V*[cos(angle4(z)), sin(angle4(z))],
[1/(sqrt(2)), 1/(sqrt(2))])*dot(M*V*[cos(angle1(z)), sin(angle1(z))],
[1/(sqrt(2)), 1/(sqrt(2))]) - dot(M*V*[cos(angle4(z)), sin(angle4(z))],
M*V*[cos(angle1(z)), sin(angle1(z))]));
    dipoleEnergy1(z) = dipoleEnergy1(z) + -
mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle3(z)), sin(angle3(z))], [0,
1])*dot(M*V*[cos(angle1(z)), sin(angle1(z))], [0, 1]) -
dot(M*V*[cos(angle3(z)), sin(angle3(z))], M*V*[cos(angle1(z)),
sin(angle1(z))]));

    dipoleEnergy2(z) = -mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle1(z)),
sin(angle1(z))], [-1, 0])*dot(M*V*[cos(angle2(z)), sin(angle2(z))], [-1, 0])
- dot(M*V*[cos(angle1(z)), sin(angle1(z))], M*V*[cos(angle2(z)),
sin(angle2(z))]));
    dipoleEnergy2(z) = dipoleEnergy2(z) + -
mu0/(4*pi*(dist*sqrt(2)).^3)*(3*dot(M*V*[cos(angle3(z)), sin(angle3(z))],
[1/(sqrt(2)), -1/(sqrt(2))])*dot(M*V*[cos(angle2(z)), sin(angle2(z))],
[1/(sqrt(2)), -1/(sqrt(2))]) - dot(M*V*[cos(angle3(z)), sin(angle3(z))],
M*V*[cos(angle2(z)), sin(angle2(z))]));
    dipoleEnergy2(z) = dipoleEnergy2(z) + -
mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle4(z)), sin(angle4(z))], [0,
1])*dot(M*V*[cos(angle2(z)), sin(angle2(z))], [0, 1]) -
dot(M*V*[cos(angle4(z)), sin(angle4(z))], M*V*[cos(angle2(z)),
sin(angle2(z))]));

    dipoleEnergy3(z) = -mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle4(z)),
sin(angle4(z))], [1, 0])*dot(M*V*[cos(angle3(z)), sin(angle3(z))], [1, 0]) -
```

```
dot(M*V*[cos(angle4(z)), sin(angle4(z))], M*V*[cos(angle3(z)),
sin(angle3(z))]));
    dipoleEnergy3(z) = dipoleEnergy3(z) + -
mu0/(4*pi*(dist*sqrt(2)).^3)*(3*dot(M*V*[cos(angle2(z)), sin(angle2(z))],
[1/(sqrt(2)), -1/(sqrt(2))])*dot(M*V*[cos(angle3(z)), sin(angle3(z))],
[1/(sqrt(2)), -1/(sqrt(2))]) - dot(M*V*[cos(angle2(z)), sin(angle2(z))],
M*V*[cos(angle3(z)), sin(angle3(z))]));
    dipoleEnergy3(z) = dipoleEnergy3(z) + -
mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle1(z)), sin(angle1(z))], [0, -
1])*dot(M*V*[cos(angle3(z)), sin(angle3(z))], [0, -1]) -
dot(M*V*[cos(angle1(z)), sin(angle1(z))], M*V*[cos(angle3(z)),
sin(angle3(z))]));

    dipoleEnergy4(z) = -mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle2(z)),
sin(angle2(z))], [0, -1])*dot(M*V*[cos(angle4(z)), sin(angle4(z))], [0, -1])
- dot(M*V*[cos(angle2(z)), sin(angle2(z))], M*V*[cos(angle4(z)),
sin(angle4(z))]));
    dipoleEnergy4(z) = dipoleEnergy4(z) + -
mu0/(4*pi*(dist*sqrt(2)).^3)*(3*dot(M*V*[cos(angle1(z)), sin(angle1(z))], [-
1/(sqrt(2)), -1/(sqrt(2))])*dot(M*V*[cos(angle4(z)), sin(angle4(z))], [-
1/(sqrt(2)), -1/(sqrt(2))]) - dot(M*V*[cos(angle1(z)), sin(angle1(z))],
M*V*[cos(angle4(z)), sin(angle4(z))]));
    dipoleEnergy4(z) = dipoleEnergy4(z) + -
mu0/(4*pi*(dist).^3)*(3*dot(M*V*[cos(angle3(z)), sin(angle3(z))], [-1,
0])*dot(M*V*[cos(angle4(z)), sin(angle4(z))], [-1, 0]) -
dot(M*V*[cos(angle3(z)), sin(angle3(z))], M*V*[cos(angle4(z)),
sin(angle4(z))]));
end

figure
plot(time, dipoleEnergy1)
hold on
plot(time, dipoleEnergy2)
plot(time, dipoleEnergy3)
plot(time, dipoleEnergy4)
xlabel('Time [s]')
ylabel('Dipole Energy [J]')
title('Dipole Energy as a Function of Time')
legend('Particle 1', 'Particle 2', 'Particle 3', 'Particle 4')


magnetStats(1, :) = [-93.054, -91.975, -87.903, -78.098, -65.652, -57.653, -
44.771, -33.714, -23.422, -11.9087, 0, 11.9087, 23.422, 33.714, 44.771,
57.653, 65.652, 78.098, 87.903, 91.975, 93.054];
magnetStats(2, :) = [-94.817, -93.572, -87.271, -76.222, -67.164, -56.698, -
46.198, -32.015, -21.372, -10.058, 0, 10.058, 21.372, 32.015, 46.198, 56.698,
67.164, 76.222, 87.271, 93.572, 94.817];
magnetStats(3, :) = [-75.934, -78.207 -75.410, -73.105, -67.891, -63.578, -
47.848, -33.875, -21.834, -10.228, 0, 10.228, 21.834, 33.875, 47.848, 63.578,
67.891, 73.105, 75.410, 78.207, 75.934];
magnetStats(4, :) = [-20.300, -20.250, -20.199, -20.039, -19.607, -20.183, -
19.790, -20.051, -15.621, -7.476, 0, 7.476, 15.621, 20.051, 19.790, 20.183,
19.607, 20.039, 20.199, 20.250, 20.300];

magnetStats = magnetStats*10^-3;
```

```
currentStats = [-12.5, -11.25, -10.0, -8.75, -7.50, -6.25, -5.0, -3.75, -
2.50, -1.25, 0, 1.25, 2.5, 3.75, 5.0, 6.25, 7.50, 8.75, 10.00, 11.25, 12.5];

i_ind = find(currentStats == I);

switch f
    case 0.1
        freq_ind = 1;
    case 0.5
        freq_ind = 2;
    case 1
        freq_ind = 3;
    case 5
        freq_ind = 4;
end

H = magnetStats(freq_ind, i_ind)*sqrt(2.0)*sin(2*pi*f*time);

Z1 = -M*V*H'.*cos(angle1);
Z2 = -M*V*H'.*cos(angle2);
Z3 = -M*V*H'.*cos(angle3);
Z4 = -M*V*H'.*cos(angle4);

figure
hold on
plot(time, Z1)
plot(time, Z2)
plot(time, Z3)
plot(time, Z4)
xlabel('Time [s]');
ylabel('Zeeman Energy [J]');
title('Zeeman Energy over time');
yyaxis right
plot(time, H)
ylabel('Magnetic Field Strength [T]');
legend('Particle 1', 'Particle 2', 'Particle 3', 'Particle 4', 'Magnetic
Field');

Z = Z1 + Z2 + Z3 + Z4;

dipoleEnergy = dipoleEnergy1 + dipoleEnergy2 + dipoleEnergy3 + dipoleEnergy4;

totalEnergy = Z + dipoleEnergy;

figure
plot(time, totalEnergy)
xlabel('Time [s]');
ylabel('Total Energy [J]');
axis([0 60 -0.4 0.4]);
yyaxis right
plot(time, H)
ylabel('Applied Magnetic Field [T]');
axis([0 60 -0.115 0.115]);
```

```
externalToDipole = Z./dipoleEnergy;

figure
plot(time, externalToDipole)
xlabel('Time [s]');
ylabel('Ratio of Zeeman energy to dipole energy Z/U [-]');
title('Ratio of System Energy');
axis([0 60 -10 10]);

yyaxis right
plot(time, H)
ylabel('Applied Magnetic Field [T]');
axis([0 60 -0.115 0.115]);
```

Appendix C

Critical Field Calculation Code

```
R = 0.5*10^-6; % Particle Radius [m]
M = 72*5.28*(1/1000)*(100/1)^3; % Particle magnetization strength [N/m^3]
mu0 = 4*pi*10^(-7); %permeability of free space [N/A^2]
Vequi = 4/3*pi*R^3; %particle volume [m^3]
a = ((Vequi*(3/4)*3.5)/(pi))^(1/3);
b = a;
c = a/3.5;
V = 4/3*pi*a*b*c; % [m^3] ellipsoidal particle volume

% Determine x-y locations of particles
dim = 5*R*1.5;
step = 2*dim/5;

X = (-dim + step/2):step:(dim - step/2);
X = [X, X, X, X, X];

p1 = [];
p2 = [];
p3 = [];
p4 = [];
p5 = [];

for i = 1:5
    p1 = [p1, -dim + step/2];
    p2 = [p2, -dim + 3*step/2];
    p3 = [p3, 0];
    p4 = [p4, dim - 3*step/2];
    p5 = [p5, dim - step/2];
end

X(2, :) = [p1, p2, p3, p4, p5];

% Find the distance between each particle
r = squareform(pdist(X'));

% Find the unit vector pointing from one particle to another
xdist = [];
ydist = [];
for i = 1:25
    for j = (i + 1):25
        xdist = [xdist, X(1, j) - X(1, i)];
        ydist = [ydist, X(2, j) - X(2, i)];
    end
end

xdist = squareform(xdist);
ydist = squareform(ydist);
```

```
rhat(:, :, 1) = xdist;
rhat(:, :, 2) = ydist;

for i = 1:25
    for j = 1:25
        if i ~= j
            mag = sqrt(rhat(i, j, 1)^2 + rhat(i, j, 2)^2);
            rhat(i, j, :) = rhat(i, j, :)./mag;
        end
    end
end

% Define the two states at which dipole energy will be calculated, diamond
% and parallel
D_dia = [sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2, -
sqrt(2)/2, -sqrt(2)/2, -sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, sqrt(2)/2,
sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2, -sqrt(2)/2, -sqrt(2)/2, -
sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, sqrt(2)/2,
sqrt(2)/2];

D_dia(2, :) = [sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2,
sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, -
sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2,
sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2, sqrt(2)/2, -
sqrt(2)/2, sqrt(2)/2];

D_para = zeros(1, 25);

D_para(2, :) = ones(1, 25);

% Sum dipole energy between each particle
U_dia = 0;
U_para = 0;

for i = 1:25
    for j = 1:25
        if i ~= j
            U_dia = U_dia + -mu0/(4*pi*(r(i, j))^3)*(3*M^2*V^2*(dot(D_dia(:,
i), [rhat(i, j, 1), rhat(i, j, 2)]))*(dot(D_dia(:, j), [rhat(i, j, 1),
rhat(i, j, 2)])) - M^2*V^2*(dot(D_dia(:, i), D_dia(:, j))));
            U_para = U_para + -mu0/(4*pi*(r(i,
j))^3)*(3*M^2*V^2*(dot(D_para(:, i), [rhat(i, j, 1), rhat(i, j,
2)]))*(dot(D_para(:, j), [rhat(i, j, 1), rhat(i, j, 2)])) -
M^2*V^2*(dot(D_para(:, i), D_para(:, j))));
        end
    end
end

% Find the Zeeman energy for each state
Z_dia = 0;
for i = 1:25
    Z_dia = Z_dia + -mu0*M*V*(dot(D_dia(:, i), [0, 1]));
end
Z_para = -25*mu0*M*V;
```

```
% Calculate critical field
H_crit = (U_dia - U_para)/(Z_para - Z_dia);

% Express critical field as a proportion of particle magnetization
ratio = H_crit/M;
```

BIBLIOGRAPHY

[1] C. W. Shields and O. D. Velev, "The Evolution of Active Particles: Toward Externally Powered Self-Propelling and Self-Reconfiguring Particle Systems," *Chem*, vol. 3, no. 4, pp. 539–559, Oct. 2017, doi: 10.1016/J.CHEMPR.2017.09.006.

[2] K. Han, C. W. Shields, N. M. Diwakar, B. Bharti, G. P. López, and O. D. Velev, "Sequence-encoded colloidal origami and microbot assemblies from patchy magnetic cubes," *Sci Adv*, vol. 3, no. 8, Aug. 2017, doi: 10.1126/SCIADV.1701108/SUPPL_FILE/1701108_SM.PDF.

[3] M. P. Spencer, D. Gao, and N. Yamamoto, "Tunable one-dimensional assembly of magnetic nanoparticles using oscillating magnetic fields at low frequencies for polymer nanocomposite fabrication," *J Magn Magn Mater*, vol. 468, pp. 200–208, Dec. 2018, doi: 10.1016/J.JMMM.2018.08.006.

[4] O. Gravel, J. Lauzon-Gauthier, C. Duchesne, and F. Larachi, "Inception of vortical coherent structures from spinning magnetic nanoparticles in rotating magnetic fields – New nanofluid microscale mixing tool," *Chemical Engineering Journal*, vol. 260, pp. 338–346, Jan. 2015, doi: 10.1016/J.CEJ.2014.08.100.

[5] D. Soto-Aquino and C. Rinaldi, "Nonlinear energy dissipation of magnetic nanoparticles in oscillating magnetic fields," *J Magn Magn Mater*, vol. 393, pp. 46–55, Nov. 2015, doi: 10.1016/J.JMMM.2015.05.009.

[6] G. M. Whitesides and B. Grzybowski, "Self-assembly at all scales," *Science (1979)*, vol. 295, no. 5564, pp. 2418–2421, Mar. 2002, doi: 10.1126/SCIENCE.1070821/ASSET/D22255A9-68B3-47BD-AA26-06264D9DE927/ASSETS/GRAPHIC/SE1220326003.JPEG.

[7] G. Kokot and A. Snezhko, "Manipulation of emergent vortices in swarms of magnetic rollers," *Nat Commun*, vol. 9, no. 1, Dec. 2018, doi: 10.1038/S41467-018-04765-W.

[8] A. Kaiser, A. Snezhko, and I. S. Aranson, “Flocking ferromagnetic colloids,” *Sci Adv*, vol. 3, no. 2, Feb. 2017, doi: 10.1126/SCIADV.1601469/SUPPL_FILE/1601469_SM.PDF.

[9] B. A. Grzybowski, H. A. Stone, and G. M. Whitesides, “Dynamic self-assembly of magnetized, millimetre-sized objects rotating at a liquid–air interface,” *Nature 2000 405:6790*, vol. 405, no. 6790, pp. 1033–1036, Jun. 2000, doi: 10.1038/35016528.

[10] E. Climent, K. Yeo, M. R. Maxey, and G. E. Karniadakis, “Dynamic Self-Assembly of Spinning Particles,” *J Fluids Eng*, vol. 129, no. 4, pp. 379–387, Apr. 2007, doi: 10.1115/1.2436587.

[11] G. Kokot, S. Das, R. G. Winkler, G. Gompper, I. S. Aranson, and A. Snezhko, “Active turbulence in a gas of self-assembled spinners,” *Proc Natl Acad Sci U S A*, vol. 114, no. 49, pp. 12870–12875, Dec. 2017, doi: 10.1073/PNAS.1710188114/-/DCSUPPLEMENTAL.

[12] J. Yan, M. Han, J. Zhang, C. Xu, E. Luijten, and S. Granick, “Reconfiguring active particles by electrostatic imbalance,” *Nature Materials 2016 15:10*, vol. 15, no. 10, pp. 1095–1099, Jul. 2016, doi: 10.1038/nmat4696.

[13] S.-J. Guo *et al.*, “Synchronization of magnetic dipole rotation in an ac magnetic field,” *J Phys A Math Theor*, vol. 44, no. 29, p. 295101, Jun. 2011, doi: 10.1088/1751-8113/44/29/295101.

[14] A. F. Siegenfeld and Y. Bar-Yam, “An Introduction to Complex Systems Science and Its Applications,” *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/6105872.

[15] N. H. P. Nguyen, D. Klotsa, M. Engel, and S. C. Glotzer, “Emergent collective phenomena in a mixture of hard shapes through active rotation,” *Phys Rev Lett*, vol. 112, no. 7, p. 075701, Feb. 2014, doi: 10.1103/PHYSREVLETT.112.075701/FIGURES/3/MEDIUM.

[16] J. M. McCarthy, S. Watkins, A. Deivasigamani, and S. J. John, “Fluttering energy harvesters in the wind: A review,” *J Sound Vib*, vol. 361, pp. 355–377, Jan. 2016, doi: 10.1016/J.JSV.2015.09.043.

[17] M. Bryant, R. L. Mahtani, and E. Garcia, “Wake synergies enhance performance in aeroelastic vibration energy harvesting:,” *http://dx.doi.org/10.1177/1045389X12443599*, vol. 23, no. 10, pp. 1131–1141, May 2012, doi: 10.1177/1045389X12443599.

[18] S. Olivieri, C. Boragno, R. Verzicco, and A. Mazzino, “Constructive interference in a network of elastically-bounded flapping plates,” *J Fluids Struct*, vol. 90, pp. 334–353, Oct. 2019, doi: 10.1016/J.JFLUIDSTRUCTS.2019.07.009.

[19] E. Beltramo, M. E. P. Segura, B. A. Roccia, M. F. Valdez, M. L. Verstraete, and S. Preidikman, “Constructive Aerodynamic Interference in a Network of Weakly Coupled Flutter-Based Energy Harvesters,” *Aerospace 2020, Vol. 7, Page 167*, vol. 7, no. 12, p. 167, Nov. 2020, doi: 10.3390/AEROSPACE7120167.

[20] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature 2008 453:7191*, vol. 453, no. 7191, pp. 80–83, May 2008, doi: 10.1038/NATURE06932.

[21] Z. Wang, L. Wang, M. Nagai, L. Xie, M. Yi, and W. Huang, “x,” *Adv Electron Mater*, vol. 3, no. 7, p. 1600510, Jul. 2017, doi: 10.1002/AELM.201600510.

[22] V. K. Sangwan and M. C. Hersam, “Neuromorphic nanoelectronic materials,” *Nat Nanotechnol*, vol. 15, no. 7, pp. 517–528, Jul. 2020, doi: 10.1038/S41565-020-0647-Z.

[23] R. Rosenweig, *Ferrohydrodynamics*. Cambridge: Cambridge University Press, 1985.

[24] M. Rodriguez, “3-D Simulations of Hard Magnetic Particles In Fluid Media,” Thesis, The Pennsylvania State University, University Park, PA, 2016.

[25] S. Egri and G. Bihari, “A simple magnetic oscillator-rotator model made of magnetic balls, and its examination by video-analysis,” *Eur J Phys*, vol. 42, no. 3, p. 035002, Mar. 2021, doi: 10.1088/1361-6404/ABD25C.

ACADEMIC VITA

James M. Zola

Online: https://www.linkedin.com/in/jameszola/

Education

Bachelor of Science in Mechanical Engineering with Honors May 2023
The Pennsylvania State University, Schreyer Honors College, University Park, PA

Study Abroad at Tecnun University of Navarra Fall 2021
Tecnun University of Navarra, San Sebastian, Gipuzkoa, Spain

Work Experience

Pennsylvania State University State College, PA
Research Assistant to Dr. Paris von Lockette 2019-2023

- Simulated motion of magnetic particles under applied fields
- Fabricated experimental models using multi-material 3D printer
- Presented weekly progress updates
- Thesis title: "An Investigation into Complex Systems of Magnetic Particles Exposed to External Fields"

Pennsylvania State University State College, PA
Grader Spring 2021

- Reviewed student assignments to check for errors
- Communicated with peers to ensure fair grading

Pennsylvania State University State College, PA
Research Assistant to Dr. Joseph Najem Fall 2022

- Imported data from Excel to MATLAB to define particle geometry
- Performed surface optimization in Surface Evolver

Leadership and Activities

Schreyer Student Council State College, PA

- Service Committee Member 2019-2023
- Raised money for Schreyer Emergency Fund

Penn State Club Cross Country State College, PA, 2019-2023

- Diversity Forum Moderator Spring 2022

Engineering Leadership Development Club 2020-2023

- Two-time annual design competition winner, with students from Switzerland

Computer and Technical Skills

- Basic Knowledge of Java
- Thorough understanding of MATLAB

Non-Technical Skills

- Fluent in Spanish

Awards and Honors

- Dean's List