

Classical Magnetic Frustration

Eugene J. Tsao

*In lab of Professor E.A. Henriksen
Washington University in St. Louis, Department of Physics*

Abstract

We present a classical analogue to the spin glass state realized experimentally via a system of freely rotating bar magnets that is capable of exploring dipole wave dynamics and macroscopic frustration. We report low amplitude wave transport in a 1-dimensional chain, and observe a variety of metastable states in the square, triangular, kagome, and honeycomb lattices. We confirm features observed in the physical 1-dimensional chain and square lattice through a Monte Carlo simulation of interacting finite-length dipoles.

Keywords: Classical Frustration, Magnets, Spin Glass

1. Introduction

Macroscopic models of traditionally microscopic quantum phenomena have attracted recent interest in investigations of both frustrated magnetism [1] and topological edge transport [2]. As materials with geometrically frustrated spins are sought and studied for their topological properties [3], this search has expanded interest in spin glass and spin liquid physics. In this Senior Honors Thesis, we report on an experimental model of interacting dipoles, realized in lattices composed of freely rotating bar magnets, which serves as a classical analogue to the quantum spin glass and spin liquid.

In Sections 1.1 and 1.2, we discuss spin glasses and topological metamaterials as areas for which the experimental model may deliver physical intuition. In Section 2, we describe the system construction. In Section 3, we report on wave transport and lattice behavior through manual arrangement, randomization, and equilibration. In Section 4, we compare the 1-D chain and 2-D square lattice to a Monte Carlo simulation of interacting dipoles.

1.1. Spin Glasses

Beginning in the 1970s, research on spin glasses emerged to address questions surrounding “quenched disorder,” i.e. the absence of a typical phase transition between a liquid and a glass [4, 5]. In the following decade, scientists studying notions of complexity recognized that spin glasses shared several features common with complex systems in general. These include frustration—the presence of conflicting constraints that cannot be simultaneously satisfied—and relatedly, many low-lying metastable states.

Ordinary (non-spin) glasses are generally characterized by their spatially amorphous organization resulting in a lack of long range crystalline order (Figure 1). Spin glasses require an amorphous spin orientation rather than position orientation; they lack long range spin alignment order. Spin glasses may or may not be positionally crystalline, but they must be magnetically disordered over long ranges. Figure 2 shows an example, in the distinction between the spin glass and ferromagnetic (ordered) state.

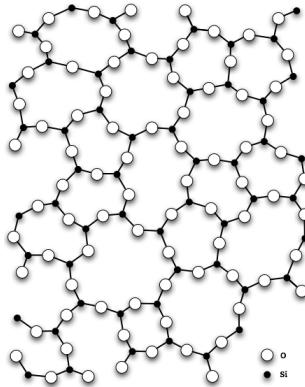


Figure 1: Window glass is composed primarily of amorphous silica. Picture from Wikipedia. Public Domain.

Metastable state degeneracy, which we define as the large number of local energy minima each corresponding to a different state of a system, emerges from this kind of disorder. To understand this degeneracy of spin glasses, one can imagine a collection of spins; depending on the particular positional arrangement of the spins, there are certain overall orientations that can minimize the global energy of the system. There are specific combinations of spin orientations that correspond to metastable states. The correspondence between spin orientation and metastability is commonly visualized in terms

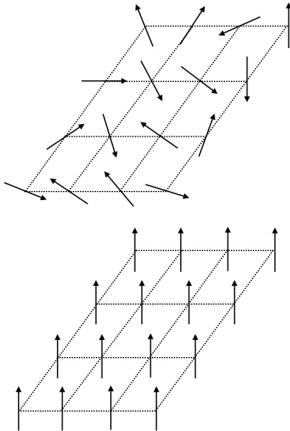


Figure 2: The upper drawing shows the disordered nature of spin-glass. The lower drawing shows the ferromagnetic state in which all spin dipoles orient in the same direction. This state characterizes permanent magnets.

of an energy landscape. For every collective permutation, the total energy of the system is computed and graphed as a function of each degree of freedom of each spin, i.e. as a function of the phase space. Figure 3 displays the energy landscape in a simplified form.

It is helpful to take a metaphorical walk through the energy landscape. One may imagine standing in a local minimum. What would it mean to step over to an adjacent minimum? In the phase space of an Ising spin model, a single step corresponds to a single flip of an arbitrary spin. This spin flip might increase the total energy, thereby partially lifting one out of the “valley” yet still standing on a slope pushing one back to where one started. But by choosing the correct spin to flip one might take another step up and over the hill, cresting the local maximum. Falling down the other side of the hill, more spins can be flipped to lower the total energy. One now sits in a different metastable combination of spins. This is one feature of the frustrated energy landscape.

Additionally, experimental observations have shown intriguing memory effects in spin glasses. Jonason et al. [6] demonstrated an ability to imprint magnetic susceptibility memories into spin glasses. Spin glasses exhibit features that are more complicated than those in ordered magnets, such as the simple lagging hysteresis found in ordinary paramagnets.

The extensive ground state degeneracy of spin glasses can be understood

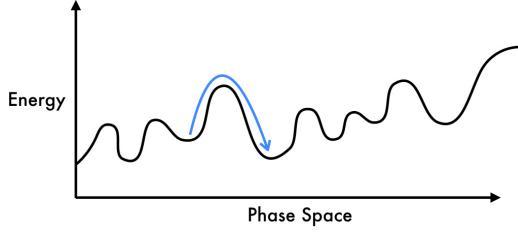


Figure 3: Low points or local minima represent meta-stable states of the system. These minima can represent multiple degenerate states of systems such as the frustrated kagome lattice.

as a consequence of frustration. While frustration often leads to complex and unpredictable effects, its origins are simple. One could imagine three Ising spins situated on a triangular lattice, all antiferromagnetically coupled; each spin has a tendency to point in the opposite direction of a neighboring spin (Figure 4).

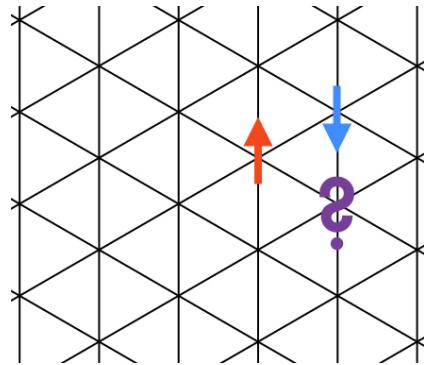


Figure 4: Placing an additional spin on the question mark, the spin would simultaneously satisfy the antiferromagnetic interaction of one spin and not satisfy the antiferromagnetic interaction of the other spin. This diagram is of a triangular lattice.

These three interacting spins are frustrated. They cannot obey all their antiferromagnetic constraints. One frustrated 2-D lattice is the antiferromagnetic kagome lattice. The kagome lattice is characterized by corner sharing triangles (Figure 5).

The unique 1-in 2-out or 2-in 1-out spin orientation rules of the kagome lattice have spurred investigation of magnetic monopoles and ordering. This orientation rule in the kagome lattice can be understood by imagining three magnetic charge dipoles, each pointing into the center point of a shared tri-

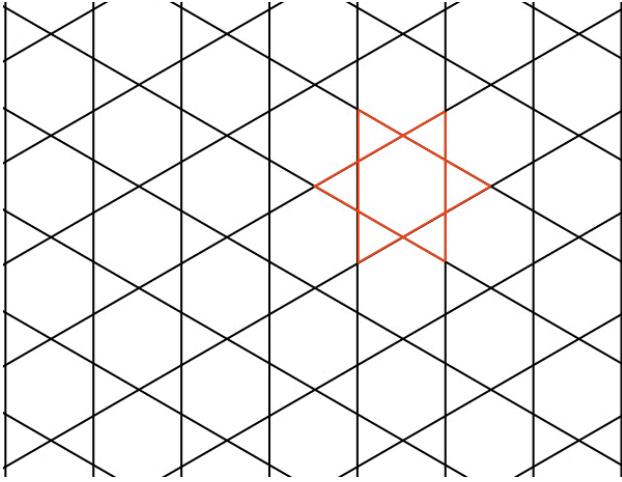


Figure 5: A kagome lattice is composed of corner sharing triangles.

angle. The dipoles should align such that there are two positive charges and one negative charge within the triangle, or two negative charges and one positive charge in the triangle. Work by Mellado et al. [1], from which we draw experimental inspiration, demonstrated two-in one-out rules in a macroscopic setting (after observation in the mesoscopic realm [7]). In the system of interacting classical magnets used by Mellado et al., each bar magnet contained one free axis of rotation allowing vertical motion into and out of the center of each triangle.

That the kagome lattice forms a net charge of either +1 or -1 within a triangle contrasts with other frustrated lattices like the 3-D pyrochlore lattice, which obeys a 2-in 2-out rule and thus contains a single lowest energy state of net charge 0. This property of the kagome lattice suggests that charge defects at one vertex may result from a low energy spin flipping chain.

The frustrated kagome lattice has also received great interest owing to the experimental observation of novel transport phenomena such as the thermal Hall effect [8] and topologically protected magnon bands [9].

1.2. Topological Meta-materials

Topology is the study of those properties of arbitrary shapes which are preserved under continuous deformation, absent operations like tearing or breaking. A common example of a topological property (invariant under deformation) is described by the hairy ball theorem, colloquially stated as “one

cannot comb a hairy ball without creating a cowlick,” (Figure 6). Indeed, in any hairy object with the topology of a ball, i.e. any object that can be molded from the shape of a ball such as a cube or cylinder, one could not comb that object without creating a cowlick. In contrast, because of the topology of a donut, one *can* comb a hairy donut without creating a cowlick (Figure 7), as well as any object which shares the topology of a donut, such as a coffee mug.

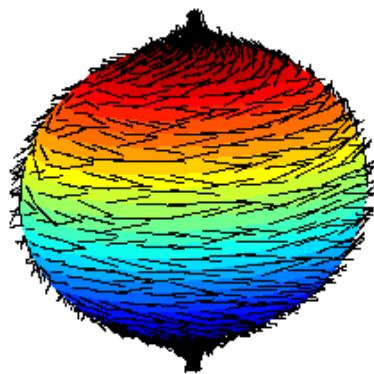


Figure 6: Two cowlicks appear at opposite poles of the sphere. Picture from Wikipedia. Public Domain.

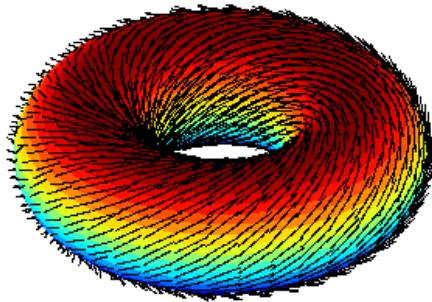


Figure 7: In the donut, the hair can be combed flat. The ball and the donut are topologically distinct—one cannot mold a ball from a donut without tearing or breaking it. Picture from Wikipedia. Public Domain.

This subject has real ramifications in quantum systems, for example,

topological understanding of the quantum Hall effect [10] has expanded understanding on particle transport in 2-D systems. Quantum Hall effect systems were always known to exhibit edge transport; however, theories from Laughlin and others [11] have yielded a topological understanding of high-mobility (low back-scattering) edge transport channels, leading to an understanding of so-called “topological insulators.” These have garnered great interest because of their edge-defect resilient (topologically protected) transport. This property suggests a new class of components possible for electronic (or magnonic) devices.

Moreover, topological effects are not inherently quantum mechanical. Recent experiments have shown topologically protected edge transport in classical systems. Suesstrunk and Huber [2] demonstrated helical edge transport in a system of coupled pendula, and L. M. Nash et al. [12] realized chiral edge transport in an array of coupled gyroscopes. As mechanical structures with band gaps of topological origin are studied in greater depth, we regard the following system as a candidate for such analysis.

2. System Description

The system studied here consists of 156 freely rotating magnets that can be arranged in a variety of one- and two-dimensional arrays. “Rare earth” rod magnets are press-fit into holes drilled in 1” spherical polypropylene balls, which float in an air-bearing formed by a 1” concave hemisphere in Delrin. The magnets inside the balls interact with one another, with rotational freedom in polar and azimuthal angles but no translational freedom. We arrange the balls in a lattice, then flow air through each Delrin cup thereby floating the balls, allowing them to interact and adopt a metastable equilibrium state.

Compressed air is first passed through a desiccating air filter and two plastic manifolds, then routed individually to each plastic cup. A single ball on its own will align with the local background magnetic field of the lab space, of strength approximately 0.5G. Using the magnetic field sensor on the iPhone 6S and the Magnetometer app, each ball exerts a field of approximately 2G at 2in. from the sensor, and 10G at 1in. from the sensor. The manufacturer lists the surface field for each magnet as 4400G.

3. Results

In order to characterize the experimental system, we investigated dynamics in the 1-D chain as well as magnet arrangement in both the 1-D chain

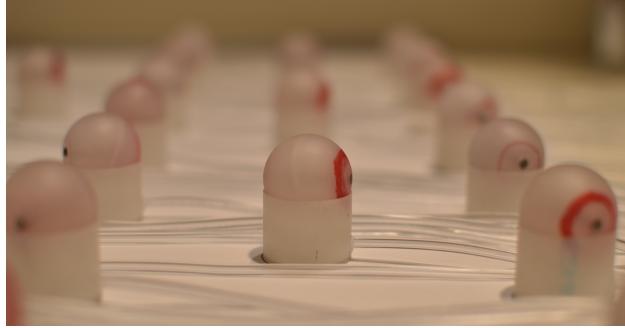


Figure 8: A polypropylene ball housing a rod magnet sits on a cushion of air formed from the Delrin cup. The red markings on the balls serve to identify like magnetic poles.

and 2-D lattices. In the 1-D chain, we arranged balls in a line and observed dynamics after both single impulses and driving impulses (i.e. angle perturbation of an end ball). In the 2-D lattices, before turning on the air, we set up each lattice either in a completely randomized (all out-of-plane) state, partially randomized (some out-of-plane) state, or manually arranged, ordered state. Turning on the air, we allowed the balls in the lattice to interact and reach a metastable state, turning off the air when ball motion had attenuated. For a 12×12 square lattice, this process was completed within 10s.

3.1. Wave Transport

We found that adjacent balls are capable of transmitting perturbations to each other. In longer chains, this transmitted perturbation takes the form of a traveling wave, in which a wave packet of larger angle deviations travels through adjacent balls. Wave packet transport is observed over maximum distances of 54 balls in the 1" edge-to-edge spacing case. In the 2" edge-to-edge spacing case, waves propagate over maximum distances of 45 balls. At 3" edge-to-edge spacing, waves propagate over maximum distances of 13 balls, unable to interact strongly enough to perturb the 14th ball.

While we were able to drive long distance wave transport through a single impulse at an end ball, we were also able to drive long distance wave transport through oscillating impulses, noting that only a single frequency for each spacing allowed long distance transport. At 3" edge-to-edge spacing and at 1" edge-to-edge spacing, this frequency is 1.8Hz and 4Hz respectively. At spacings of 1in. and 2in., wave front velocity is 33cm/s and 19cm/s respectively.

The equilibrium position for balls in a 1-D chain is pole-to-pole alignment, such that magnets orient together along the chain of alignment. Traveling wave packets are only observed for small angle oscillations away from this equilibrium position. Perturbations of large deviation angle quickly disperse into small angle oscillations. Thus waves cannot be driven further distances through stronger impulses, i.e. larger initial deviation angle.

3.2. Building Block Arrangements

Much can be learned by examining arrangements of only a few magnets, the behavior of which arises again in larger or more complex arrangements. In pairs, balls align end-to-end. In a triangle, two balls attempt to align end-to-end, while the third aligns oppositely to the others; all balls are canted slightly as if to also satisfy end-to-end alignment with both neighbors. In a square, balls align in directions perpendicular to the diagonal, again attempting end-to-end alignment with the two nearest neighbors, as well as counter alignment with the diagonal neighbor. Being classical ferromagnets, the driving order is to align north to south. Figure 9 illustrates these arrangements.

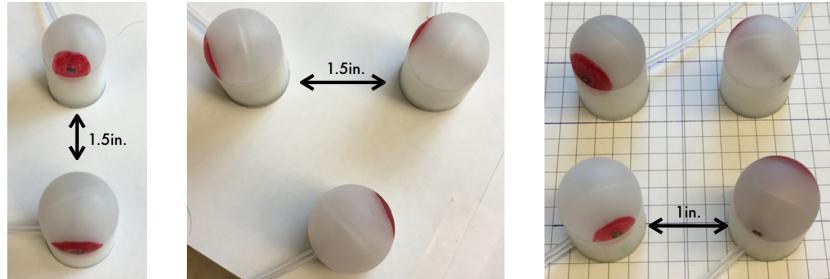


Figure 9: Balls arrange in ordered patterns whose degeneracy can be explained by rotational symmetry.

3.3. Square Lattice

Beginning with the 9-unit square lattice, balls arrange in regions of alignment or domains exhibiting multiple metastable array states. In all lattices investigated, we employed a series of randomization tests in which an initially metastable lattice state is partially randomized then re-equilibrated. The three types of randomization performed were 1) flipping one side of the metastable lattice such that flipped balls point out-of-plane, effectively erasing their previous, in-plane state (Figure 10), 2) flipping every other ball or

set of balls in a checkerboard-like pattern, and 3) using a coin flip algorithm such that each ball has a 50/50 probability of flipping into an erased state.

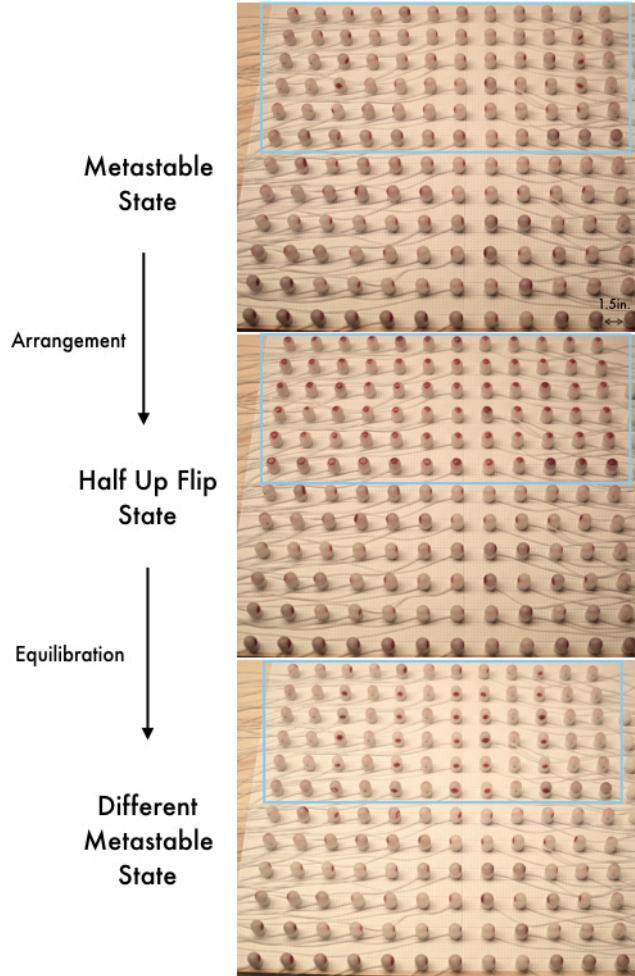


Figure 10: The top photo is of a metastable state. The upper half of the lattice was rearranged (middle photo) then re-equilibrated into a new metastable state.

Figure 10 displays randomization of the upper half of a 12×12 square lattice. In the top photo, crosswise chains of alignment form across the entire array. In the bottom photo, the region previously flipped displays new, vertical chains of alignment. The row below this region reveals slight angle shifts to accommodate the vertically aligned region. Therefore, through regional randomization and subsequent equilibration a new metastable state

was found.

Not all forms of partial randomization result in such distinctly different metastable states. Using the coin-flip algorithm for randomization, the 12×12 square lattice maintained its previous state with the exception of a small region in which balls orient along chains different from the initial metastable state, corresponding to a slight domain shift (Figure 11).

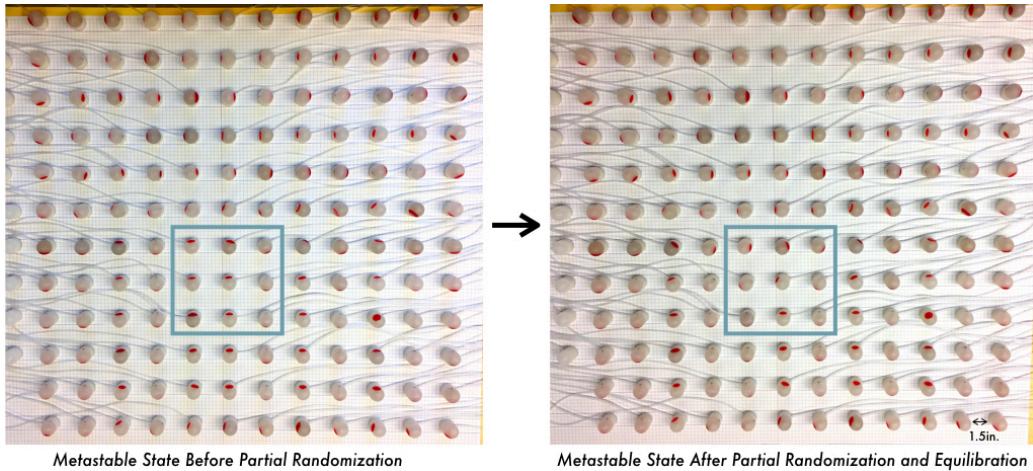


Figure 11: The photo on the left is of the original metastable state. The photo on the right is of a metastable state after partial randomization via coin-flip arrangement and equilibration. The overall position of balls are the same between states, with shifting in the blue square corresponding to shifting domain boundaries.

Features of the square lattice will be discussed in further detail in Section 4 as the physical lattice is compared to the simulated lattice.

3.4. Triangular Lattice

Like the square lattice, as well as kagome and honeycomb lattices, the triangular lattice displays multiple metastable states. Also like the square lattice, regional ball orientation erasure can result in a new metastable state in the triangular lattice (Figure 12).

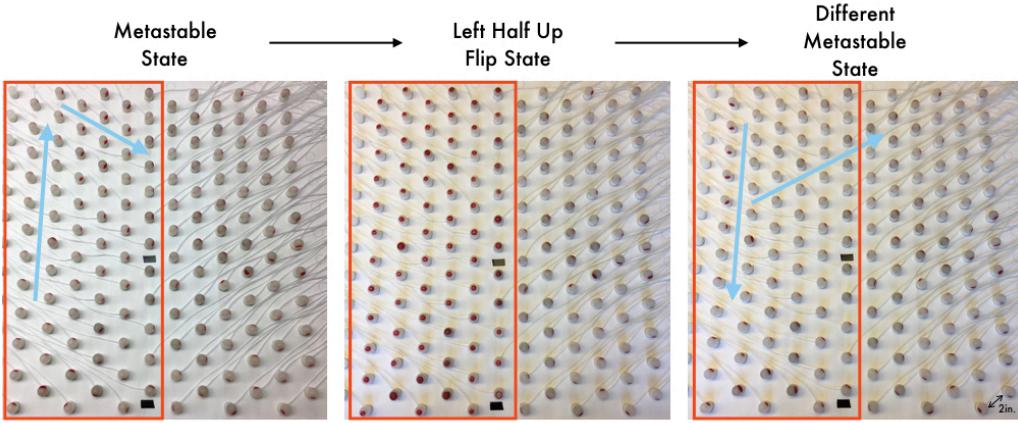


Figure 12: In the triangular lattice, regional orientation erasure also leads to changes in ball alignment in the re-equilibrated state. Blue arrows indicate alignment trends.

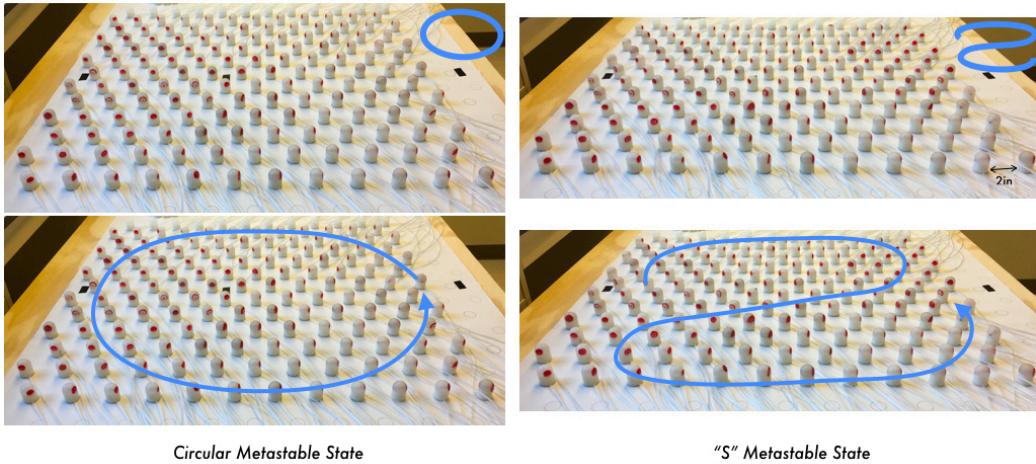


Figure 13: Looping motifs arise like the circular and S shape alignments shown above. Angle deviation between adjacent balls is sometimes sharp and sometimes smooth, resulting in hard and soft domain transitions.

The triangular lattice also displays a tendency to align in large loops (Figure 13). In the triangular lattice and square lattice, both soft domain transitions (in which angle deviation between adjacent balls is small) and hard domain transitions (large angle deviation between adjacent balls) occur depending on location within a loop or between counter-propagating adjacent

loops.

3.5. Kagome Lattice

Intriguingly, the kagome lattice favors alignment along closely spaced chains. Testing a ferromagnetic analogue, we equilibrated from an all in-plane alignment and found relaxation along chains not parallel with the original alignment.

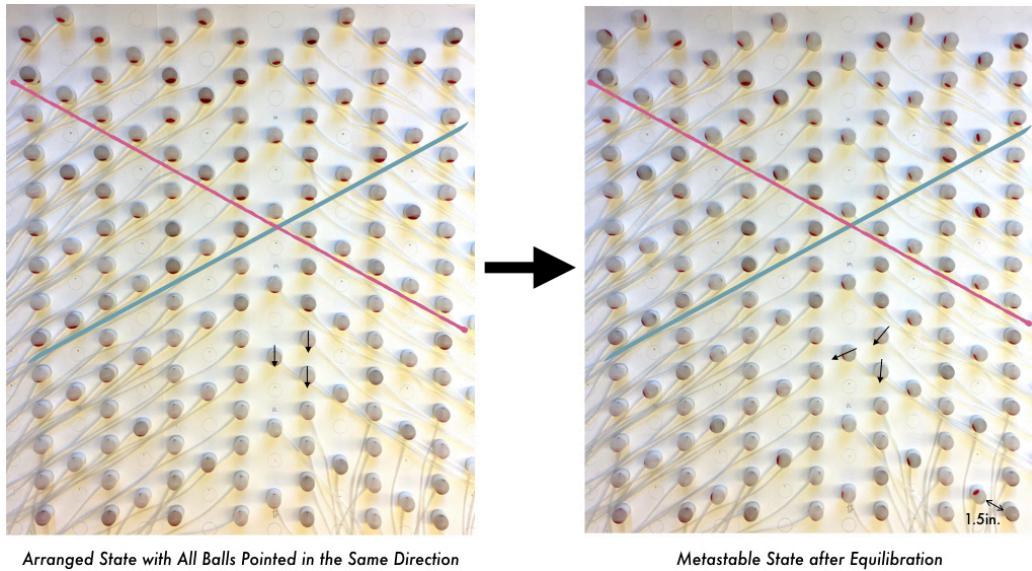


Figure 14: The kagome lattice was originally arranged such that all balls pointed in-plane, in the same direction. Diagonal lines aid the eye in observing partial realignment along diagonal chains. Arrows show ball orientation in a triangle.

Relaxation out of this “ferromagnetic” state brought the system closer to satisfying the 2-in 1-out and 2-out 1-in rules observed by Mellado et. al [1]. Nevertheless, balls do not orient strictly into and out of corner sharing triangles, even in bulk triangles.

3.6. Honeycomb Lattice

While the honeycomb lattice also exhibits many metastable states that lack a clear orientation, one metastable state was found through a dipole loop tiling. We observed that in arbitrary metastable states, balls tended to align such that chains formed around hexagons as balls attempted to align end-to-end. Whereas a coherent ordering rule in these arbitrary metastable

states was not obvious to the eye, we hypothesized an ordering rule illustrated in Figure 15, which would satisfy the tendency to form chains or “dipole loops.” We tested this state by first arranging balls in hexagonal dipole loops, then equilibrated, finding that the orientation of bulk balls remained virtually unchanged, with minor and expected shifting of edge balls. A similar tiling pattern was tested for the kagome lattice and was found not to be a metastable state.

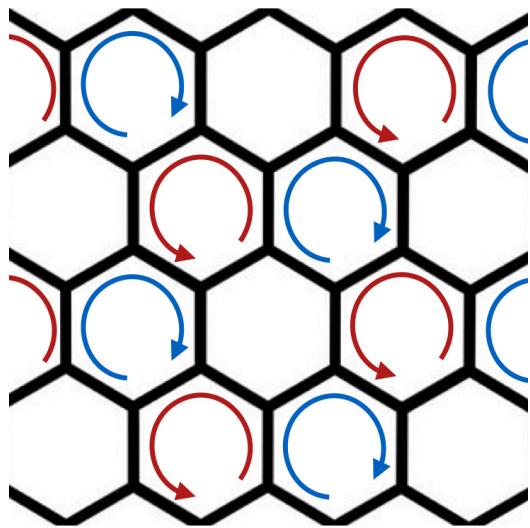


Figure 15: The hexagonal “dipole loop” tiling pattern superimposes a honeycomb lattice over the existing lattice. This tiling leaves hexagons forming the center of the superimposed hexagon unable to satisfy a “dipole loop”; however, hexagons forming the border of the superimposed hexagon do satisfy the “dipole loop.”

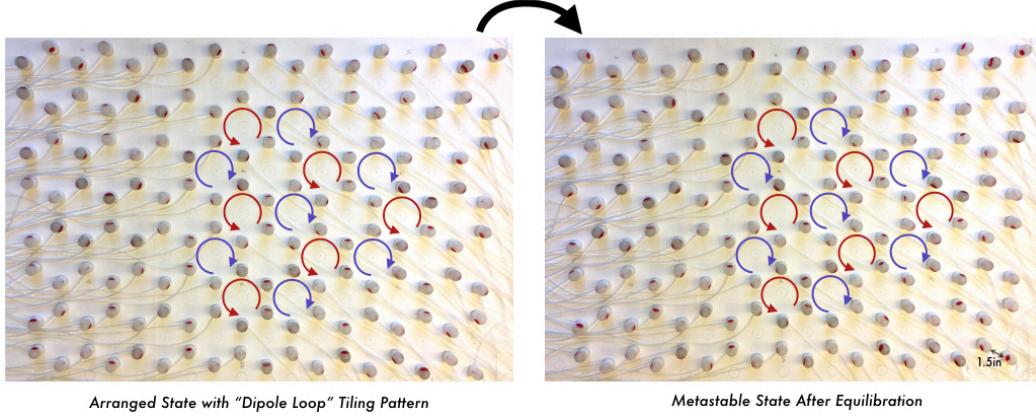


Figure 16: Corresponding circles to Figure 15 display the tiling pattern in the physical system. The entire lattice was tiled this way, and upon equilibration noticeable deviation occurred only along edges.

4. Comparison with Monte Carlo Simulation

A basic Monte Carlo method was used to simulate the 1-D chain and square lattice. Monte Carlo methods are often used in computationally intensive problems and generally rely on random sampling rather than exhaustive calculation. In the following simulation, we randomly select dipoles in a lattice and alter their orientation, comparing their past energy to their new energy, keeping their new orientation if the energy of the dipole is lower. This is a simplification of the interactions present in the experimental lattice, in which all dipoles are simultaneously attempting to find low energy orientations. Indeed, the following simulation would more closely resemble the experimental procedure of feeding air into one randomly chosen cup at a time and executing this task many times over.

In greater detail, the chain and lattice are either randomly populated or manually populated in an initial state. One dipole in the ensemble is randomly selected and assigned new, random polar (ϑ) and azimuthal (φ) angles. The new energy of the dipole in the ensemble is computed and compared with the energy of the dipole in the ensemble before angle reassignment. If the new energy is lower than the old energy, the new angle coordinates are kept by the dipole. The simulations were written in Mathematica and are included in the Appendix, as well as available upon request.

In the simulation, bar magnets are modeled as “magnetic charge” dipoles. Because the length of the rod magnets is not much smaller than the distance

between rod magnets, the ideal dipole-dipole interaction energy is not an accurate model for the physical system. Indeed, we attempted simulation using the ideal dipole-dipole interaction energy, with results that deviated greatly from experiment. Therefore, as though the rod magnets were composed of oppositely charged magnetic monopoles separated by a distance d , we compute the interaction energy by calculating distance between dipole charges, where the charges are spatially constrained by the dipole length and the lattice spacing (diagram in Figure 17):

$$U = \alpha \left(\frac{1}{d_{13}} + \frac{1}{d_{24}} - \frac{1}{d_{14}} - \frac{1}{d_{23}} \right) \quad (1)$$

where α is an arbitrary constant encoding permeability and charge strengths (all of equal magnitude) that is set to 1, and the sign of each term above comes from the positive and negative dipole charges. Distance between charges is determined by,

$$d_{ij} = \sqrt{(\Delta x_{ij})^2 + (\Delta y_{ij})^2 + (\Delta z_{ij})^2} \quad (2)$$

such that,

$$\Delta x_{ij} = \Delta x_{spacing} + \frac{l}{2}(\sin(\vartheta_j) \cos(\varphi_j) - \sin(\vartheta_i) \cos(\varphi_i)) \quad (3)$$

$$\Delta y_{ij} = \Delta y_{spacing} + \frac{l}{2}(\sin(\vartheta_j) \sin(\varphi_j) - \sin(\vartheta_i) \sin(\varphi_i)) \quad (4)$$

$$\Delta z_{ij} = \frac{l}{2}(\cos(\vartheta_j) - \cos(\vartheta_i)) \quad (5)$$

where $\Delta z_{spacing} = 0$ in a 2-D lattice in the x-y plane.

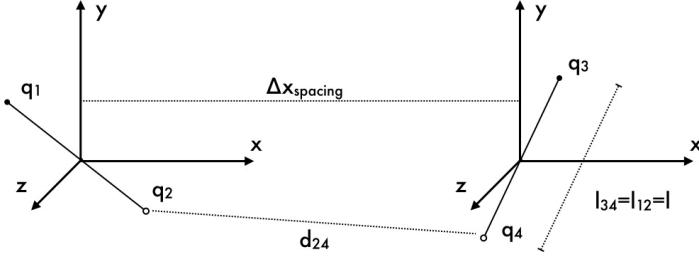


Figure 17: The energy of two dipoles was computed by modeling two magnetic charges separated by a distance l , the length of the rod magnet.

4.1. 1-D Chain

The simulated 1-D chain displays nearly identical features to the statics of the physical system. We randomly populate the chain then allow the Monte Carlo simulation to run. With many iterations (>1000) and fewer dipoles (<20) the simulation converges to uniform dipole alignment along the axis of the chain (Figure 18).

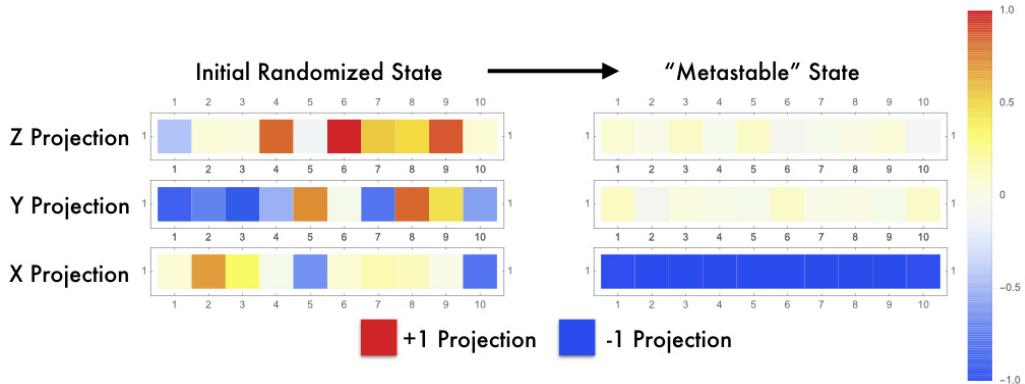


Figure 18: This simulation of 10 initially randomized dipoles in a 1-D chain underwent 10,000 iterations through the Monte Carlo algorithm. In this simulation, $l = d/2$ corresponding to the physical chain with 1“ edge-to-edge spacing.

As the number of iterations decreases or the number of dipoles increases, the chain may host multiple regions of counter alignment (Figure 19). This feature matches observation in the physical system, where longer chains can host metastable states in which there exist several regions of alignment and counter-alignment (always in the axis of the chain); however, because of computational limitations it is unclear if, with more iterations, regions of counter-alignment would attenuate until there is only north-south alignment, or if they are stable given the testing parameters of the Monte Carlo simulation, i.e. if chains such as the one shown in Figure 19 would converge to complete alignment over a longer simulation.

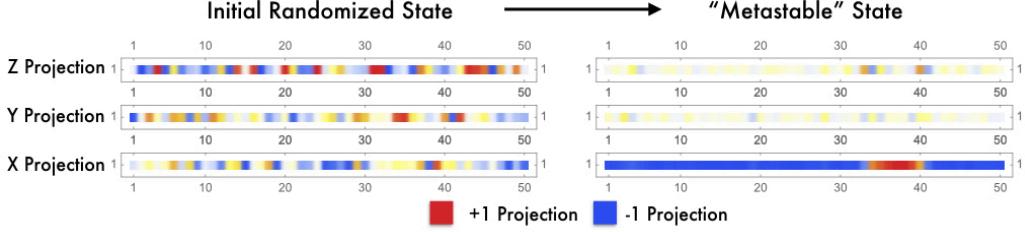


Figure 19: This simulation of 50 initially randomized dipoles in a 1-D chain also underwent 10,000 iterations through the Monte Carlo algorithm. In this simulation, $l = d/2$.

4.2. Square Lattice

The simulated square lattice also shares key features with the physical system. These features include the absence of out-of-plane alignment once equilibrated (with the central tile discussed below), as well as domain formation in which the dipoles are oriented in-plane either in the $+x$, $-x$, $+y$, or $-y$ direction.

Figure 20 displays a simulation result demonstrating the overall lack of z projection as well as domain formation oriented along the x and y axes. We have roughly translated the simulated result into the physical system by arranging balls to match the projections specified through the simulation. The metastable state produced through subsequent equilibration displays shifting from the arranged state. While there remain regions of alignment that match the arranged state, balls have tended to align in much straighter lines along the in-plane axes. Domain transitions appear much sharper in the physical system.

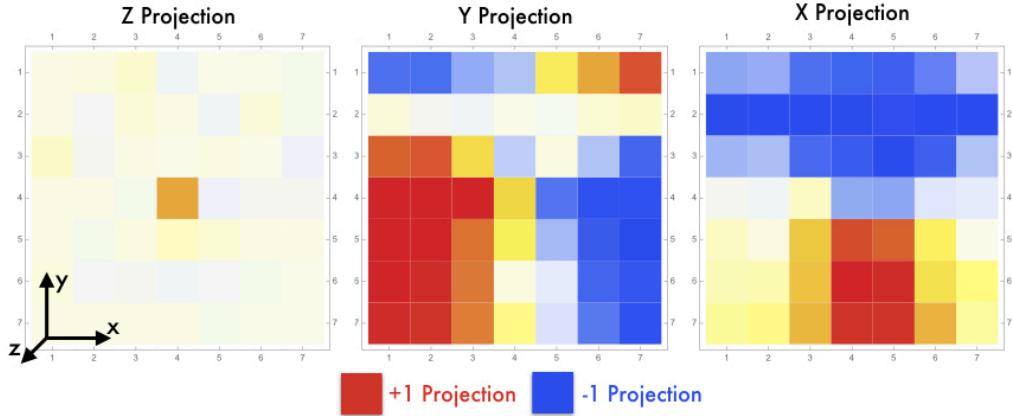
We also perform this type of comparison in reverse, where we translate a metastable state in the physical system into the simulation and then perform the Monte Carlo algorithm. In running this comparison, we again observe equilibration differences between the physical system and the simulated system. Figure 21 displays the means by which we approximated the angular coordinates of the balls. We note large measurement uncertainty due to image distortion and report the angle only to the nearest $\pi/16$ radians; however, we find that this approach allows meaningful comparison between the physical and simulated state.

In Figure 22 we display the metastable state of the physical system in the projection representation of the simulation, and we also display the subsequent state produced by running the original physical state through the

Monte Carlo simulation. Comparing these two states, we find that the Monte Carlo simulation eliminates the striping pattern found in the physical system and produces larger regions with shared alignment. We also find that the Monte Carlo simulation allows a dipole at the center to project slightly out of plane as it sits at a domain transition.

Thus, while simulated metastable states and physical metastable states share key features, metastability is different in the physical and simulated systems. Understood from the perspective of the frustrated energy landscape, in which there exist many local minima throughout phase space, there should be a difference in the metastability of the physical and simulated systems; because the physical model allows all dipoles to interact and equilibrate simultaneously, whereas the simulation only allows one dipole to interact and equilibrate, the physical model and simulated model allow very different movement through phase space and therefore access different metastable states. In other words, in phase space the simulated model allows movement only in the two dimensions of the polar and azimuthal coordinates of a single dipole (for each step), whereas the physical model allows unconstrained movement in the high-dimensional phase space defined by all polar and azimuthal coordinates of every dipole. As a consequence different energy maxima and minima may be encountered depending on the means by which the phase space is traversed, therefore the simulated and physical models may tend to settle into differing local minima and differing metastable states. However, it is informative that a simple model retains some key features of the physical system, and the variance between the physical and simulated models suggest that there may be phenomena observable in the physical system and not in simulation, especially in the dynamics of 2-D lattices.

Monte Carlo Simulation Results



Arranged Representation of Simulation Metastable State after Equilibration

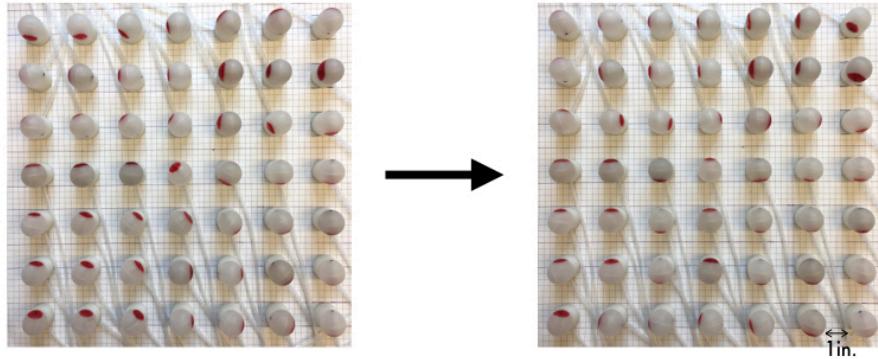


Figure 20: The projection representation above describes an approximately "metastable" state in the Monte Carlo simulation (i.e. the state has been allowed to equilibrate over many steps). In the simulation, $l = d/2$ and the number of steps is 50,000. On the lower left, we arranged the physical system to match the simulation result. This state was not maintained after equilibration, and the equilibrated physical metastable result is shown on the lower right.

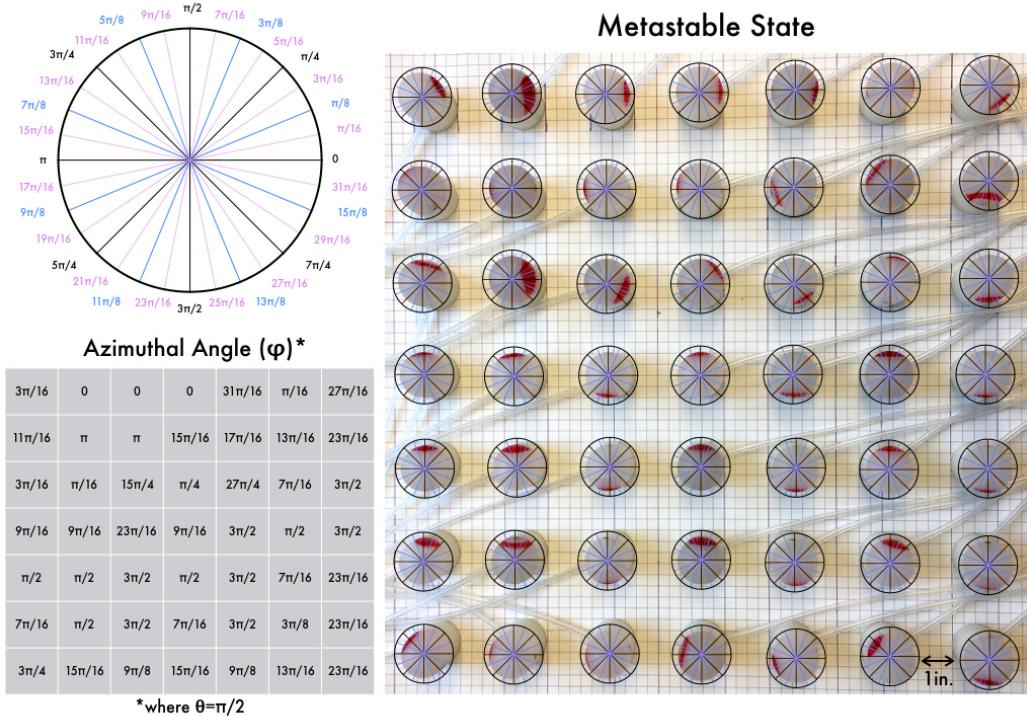
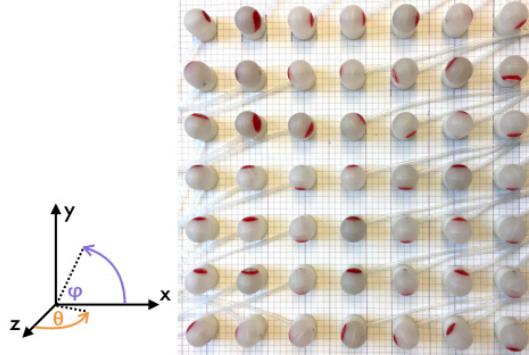
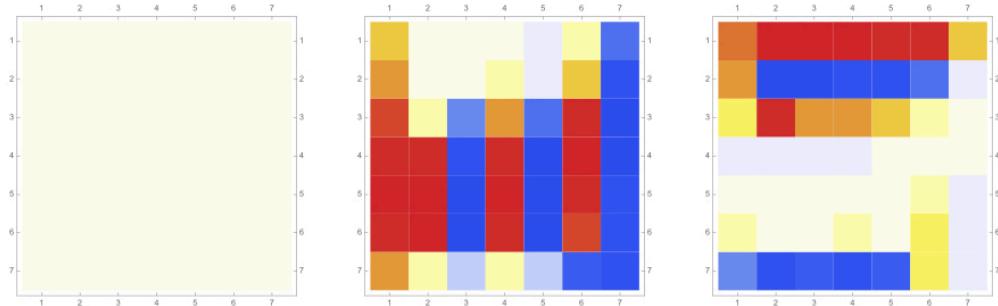


Figure 21: This figure describes the method by which the angle was extracted from the physical system and inputted into the simulation. The circle on the upper left marks angles in radians, to which the circles superimposed on the right image correspond. The table on the lower left displays the angles extracted for matching lattice sites.

Metastable State of Physical System



Metastable State of Physical System in Simulation Form



State after Equilibration in Monte Carlo Simulation

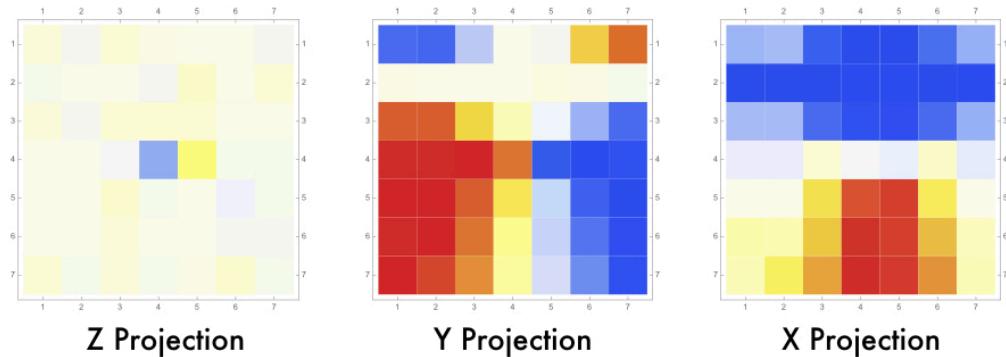


Figure 22: The upper image displays the same state as in Figure 21. The central row displays the physical metastable state in projection form. The lower row displays the state after Monte Carlo simulation. In the simulation, $l = d/2$ and the number of steps is 50,000.

5. Conclusion

In constructing an ensemble of freely rotating magnets, we classically explore wave transport through a 1-D dipole chain and metastability in various 2-D lattices. In the 1-D chain, we observe low amplitude wave transport, with varying driving frequencies and wave velocities for different spacings. We identify domain-like ordering in the many metastable states of the square and triangular lattices. In the kagome lattice, we find a tendency to align along closely spaced chains, which partially satisfies 2-in 1-out ordering rules. In the honeycomb lattice, out of many metastable states, we obtain one metastable state through an ordered “dipole loop” filling pattern.

In addition, we identify key areas of contrast and commonality between the physical square lattice and the Monte Carlo simulated square lattice. The presence of multiple metastable states in the 2-D lattices indicates a complex energy landscape in phase space, which is a hallmark of frustration. We believe that there is applicability for our experimental system in the search for macroscopic topological insulators, hysteresis studies of magnetically frustrated lattices, and wave localization measurements with varying degrees of lattice disorder. We look forward to exploring lattice dynamics, and in doing so, probing the nature of disorder, topology, and frustration in 2-D classical wave transport.

6. Acknowledgments

The author is grateful for mentorship and guidance from Professor Henriksen and the Henriksen Lab, as well as for experimental help from Jordan Pack, Shannon Levin, and many others. He is also appreciative of Professor Murch and the Murch Lab for allowing such ample space use, as well Professor Nussinov for helpful and intriguing discussions. The author thanks the Delos Family for funding provided through the Greg Delos Summer Fellowship.

- [1] P. Mellado, A. Concha, L. Mahadevan, Macroscopic Magnetic Frustration, *Physical Review Letters* 109 (2012) 257203.
- [2] R. Suessstrunk, S. Huber, Observation of Phononic Helical Edge States in a Mechanical Topological Insulator, *Science* 349 (2015) 47–50.

- [3] A. Banerjee, C. A. Bridges, J. Q. Yan, A. A. Aczel, L. Li, M. B. Stone, G. E. Granroth, M. D. Lumsden, Y. Yiu, J. Knolle, S. Chattacharjee, D. L. Kovrizhin, R. Moessner, D. A. Tennant, D. G. Mandrus, S. E. Nagler, Proximate Kitaev quantum spin liquid behaviour in a honeycomb magnet, *Nature Materials* (2016).
- [4] D. L. Stein, C. M. Newman, *Spin Glasses and Complexity*, Princeton University Press, 1st edition, 2013.
- [5] J. A. Mydosh, *Spin Glasses: an Experimental Introduction*, Taylor and Francis, 1st edition, 1993.
- [6] K. Jonason, E. Vincent, J. Hammann, J. Bouchaud, P. Nordblad, Memory and Chaos Effects in Spin Glasses, *Physical Review Letters* 81 (1998) 3243–3246.
- [7] Y. Qi, T. Brintlinger, J. Cumings, Direct Observation of the Ice Rule in an Artificial Kagome Spin Ice, *Physical Review Letters B*. 77 (2008) 09458.
- [8] M. Hirschberger, R. Chisnell, Y. S. Lee, N. P. Ong, Thermal Hall Effect in a Kagome Magnet, *Physical Review Letters* 115 (2015) 106603.
- [9] R. Chisnell, J. S. Helton, D. E. Freedman, R. I. Singh, R. I. Bewley, D. G. Nocera, Y. S. Lee, Topological Magnon Bands in a Kagome Lattice Ferromagnet, *Physical Review Letters* 114 (2015) 147201.
- [10] K. v. Klitzing, G. Dorda, M. Pepper, New Method for High-Accuracy Determination of the Fine-Structure Constant Based on Quantized Hall Resistance, *Physical Review Letters* 45 (1980) 494–497.
- [11] J. E. Avron, D. Osadchy, R. Seiler, A Topological Look at the Quantum Hall Effect, *Physics Today* 56 (2003) 38.
- [12] L. M. Nash, D. Kleckner, A. Read, V. Vitelli, A. M. Turner, W. T. M. Irvine, Topological Mechanics of Gyroscopic Metamaterials, *Applied Physical Sciences PNAS* 112 (2015) 14495–14500.

7. Appendix

Section 7.1 contains Mathematica code for the long 1-D finite-length dipole chain. Section 7.2 contains Mathematica code for the 12×12 square lattice simulation, in which the angular coordinates of an experimental metastable state were manually inputted into Mathematica, then run through the Monte Carlo simulation. All other simulations are available upon request, including simulations of ideal dipoles, rather than finite-length dipoles. Please email tsao@wustl.edu for the code or with questions regarding this research. Note, when viewing this document as a PDF there may be distortion of simulation images below, which may be due to image anti-aliasing from the PDF viewer. The images contained in the body of this thesis display the true Mathematica output.

7.1. *Mathematica Code for Long 1-D Dipole Chain*

```

In[759]:= f[x_, y_, 1] := RandomReal[2 π] (*The first position is theta.*)
f[x_, y_, 2] := RandomReal[π] (*The second position is phi.*)
d = 1/2;
l = 50;
steps = 10000; (*Number of iterations in Monte Carlo test*)
array = Array[f, {l, 1, 2}] // MatrixForm
Out[763]//MatrixForm=
( ( 4.60212 ) ( 3.23475 ) ( 2.24416 ) ( 0.183333 ) ( 3.55076 ) ( 1.13999 ) ( 2.4012 ) ( 2.11848 ) ( 4
1.30402 ) ( 1.38547 ) ( 1.23439 ) ( 1.91573 ) ( 2.01499 ) ( 0.810265 ) ( 2.11848 ) ( 2
)

```

```

In[764]:= (*array[[1,1,1,1]]=π/2;
array[[1,1,1,2]]=0;
array[[1,1,2,1]]=π/2;
array[[1,1,2,2]]=0;
array[[1,1,3,1]]=π/2;
array[[1,1,3,2]]=0;
array[[1,1,4,1]]=π/2;
array[[1,1,4,2]]=0;
array[[1,1,5,1]]=π/2;
array[[1,1,5,2]]=0;
array[[1,1,6,1]]=π/2;
array[[1,1,6,2]]=π;
array[[1,1,7,1]]=π/2;
array[[1,1,7,2]]=π;
array[[1,1,8,1]]=π/2;
array[[1,1,8,2]]=π;
array[[1,1,9,1]]=π/2;
array[[1,1,9,2]]=π;
array[[1,1,10,1]]=π/2;
array[[1,1,10,2]]=π;
array*)

zmatrix0 = ConstantArray[0, {1, 1}];

Do[Part[zmatrix0, 1, j] = Cos[array[[1, 1, j, 1]]], {j, 1}]
zmatrix0 // MatrixForm
MatrixPlot[zmatrix0,
 ColorFunction → (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling → False]

ymatrix0 = ConstantArray[0, {1, 1}];

Do[Part[ymatrix0, 1, j] = Sin[array[[1, 1, j, 1]]] * Sin[array[[1, 1, j, 2]]], {j, 1}]
ymatrix0 // MatrixForm
MatrixPlot[ymatrix0,
 ColorFunction → (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling → False]

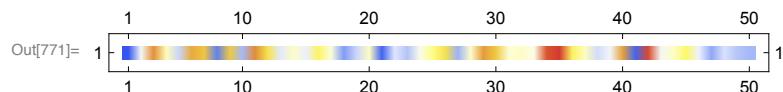
xmatrix0 = ConstantArray[0, {1, 1}];

Do[Part[xmatrix0, 1, j] = Sin[array[[1, 1, j, 1]]] * Cos[array[[1, 1, j, 2]]], {j, 1}]
xmatrix0 // MatrixForm
MatrixPlot[xmatrix0,
 ColorFunction → (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling → False]

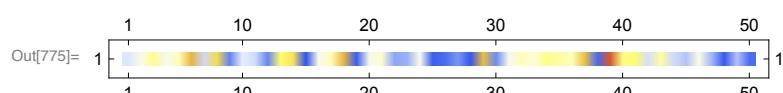
Out[766]//MatrixForm=
 (- 0.110043 - 0.995664 - 0.623615 0.983242 - 0.917451 0.417607 - 0.738205 - 0.440824 -
 1
Out[767]= 1
 1 10 20 30 40 50
 1 10 20 30 40 50

```

```
Out[770]//MatrixForm=
```

$$\begin{pmatrix} -0.958767 & -0.091427 & 0.737914 & 0.171569 & -0.35924 & 0.658274 & 0.575908 & -0.747931 & 0.565 \end{pmatrix}$$


```
Out[774]//MatrixForm=
```

$$\begin{pmatrix} -0.262022 & -0.0171402 & 0.258044 & -0.0616449 & 0.170966 & 0.626323 & -0.35126 & 0.496259 & -0 \end{pmatrix}$$


```
In[776]:=
```

```
In[777]:=
```

```
In[778]:=
```

```
In[779]:=
```

```
In[780]:=
```

```

In[78]:= xdisp1[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] :=
  (x2 - x1) + (d / 2) (Sin[θ2] Cos[φ2] - Sin[θ1] Cos[φ1]);
xdisp2[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] :=
  (x2 - x1) + (d / 2) (-Sin[θ2] Cos[φ2] + Sin[θ1] Cos[φ1]);
xdisp3[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (x2 - x1) + (d / 2) (-Sin[θ2] Cos[φ2] - Sin[θ1] Cos[φ1]);
xdisp4[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (x2 - x1) + (d / 2) (Sin[θ2] Cos[φ2] + Sin[θ1] Cos[φ1]);

ydisp1[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] :=
  (y2 - y1) + (d / 2) (Sin[θ2] Sin[φ2] - Sin[θ1] Sin[φ1]);
ydisp2[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (y2 - y1) + (d / 2) (-Sin[θ2] Sin[φ2] + Sin[θ1] Sin[φ1]);
ydisp3[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (y2 - y1) + (d / 2) (-Sin[θ2] Sin[φ2] - Sin[θ1] Sin[φ1]);
ydisp4[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (y2 - y1) + (d / 2) (Sin[θ2] Sin[φ2] + Sin[θ1] Sin[φ1]);

zdisp1[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] := (d / 2) (Cos[θ2] - Cos[θ1]);
zdisp2[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := (d / 2) (-Cos[θ2] + Cos[θ1]);
zdisp3[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := (d / 2) (-Cos[θ2] - Cos[θ1]);
zdisp4[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := (d / 2) (Cos[θ2] + Cos[θ1]);

11[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  √(xdisp1[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + ydisp1[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  zdisp1[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);
12[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  √(xdisp2[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + zdisp2[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  ydisp2[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);
13[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  √(xdisp3[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + zdisp3[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  ydisp3[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);
14[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  √(xdisp4[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + zdisp4[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  ydisp4[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);

energy[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := Piecewise[{{0, x1 == x2 && y1 == y2}, {1,
  11[x1, y1, x2, y2, θ1, φ1, θ2, φ2] + 12[x1, y1, x2, y2, θ1, φ1, θ2, φ2] - 13[x1, y1, x2, y2, θ1, φ1, θ2, φ2] - 14[x1, y1, x2, y2, θ1, φ1, θ2, φ2]}, {1, 1}}];

```

```
In[798]:= Do[randx = RandomInteger[{1, 1}];  
randy = 1;  
θp = Part[array, 1, 1, randx, 1];  
φp = Part[array, 1, 1, randx, 2];  
θn = RandomReal[2 π];  
φn = RandomReal[π];  
xi = 1;  
yi = 1;  
totalenergynew = 0;  
totalenergypast = 0;  
While[xi < 1, xi++;  
    totalenergynew = totalenergynew + (If[xi == randx, 0, 1]) * energy[randx, randy,  
        xi, yi, θn, φn, Part[array, 1, yi, xi, 1], Part[array, 1, yi, xi, 2]]];  
xi = 1;  
While[xi < 1, xi++;  
    totalenergypast = totalenergypast + (If[xi == randx, 0, 1]) * energy[randx, randy,  
        xi, yi, θp, φp, Part[array, 1, yi, xi, 1], Part[array, 1, yi, xi, 2]]];  
If[totalenergynew < totalenergypast, Part[array, 1, randy, randx, 1] = θn, 0];  
If[totalenergynew < totalenergypast, Part[array, 1, randy, randx, 2] = φn, 0], steps]
```

```
In[799]:= zmatrix = ConstantArray[0, {1, 1}];

Do[Part[zmatrix, 1, j] = Cos[array[[1, 1, j, 1]]], {j, 1}]
zmatrix // MatrixForm
MatrixPlot[zmatrix,
 ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling -> False]

ymatrix = ConstantArray[0, {1, 1}];

Do[Part[ymatrix, 1, j] = Sin[array[[1, 1, j, 1]]] * Sin[array[[1, 1, j, 2]]], {j, 1}]
ymatrix // MatrixForm
MatrixPlot[ymatrix,
 ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling -> False]

xmatrix = ConstantArray[0, {1, 1}];

Do[Part[xmatrix, 1, j] = Sin[array[[1, 1, j, 1]]] * Cos[array[[1, 1, j, 2]]], {j, 1}]
xmatrix // MatrixForm
MatrixPlot[xmatrix,
 ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling -> False]

Out[801]//MatrixForm=
(-0.0573379 0.119849 -0.00692288 0.337679 -0.335327 -0.0851159 -0.0870841 0.0763
 1 10 20 30 40 50 -1
Out[802]= 1
 1 10 20 30 40 50 -1
 1 10 20 30 40 50 -1
Out[805]//MatrixForm=
(0.155095 -0.107124 0.368497 -0.303225 -0.0290435 0.123334 -0.0341074 0.156136 -
 1 10 20 30 40 50 -1
Out[806]= 1
 1 10 20 30 40 50 -1
 1 10 20 30 40 50 -1
Out[809]//MatrixForm=
(-0.986234 -0.986996 -0.929603 -0.891082 -0.941654 -0.988708 -0.995617 -0.984781
 1 10 20 30 40 50 -1
Out[810]= 1
 1 10 20 30 40 50 -1
In[811]:= 
In[812]:=
```

```
In[813]:= zmatrix0 // MatrixForm
MatrixPlot[zmatrix0,
ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
ColorFunctionScaling -> False]
ymatrix0 // MatrixForm
MatrixPlot[ymatrix0,
ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
ColorFunctionScaling -> False]
xmatrix0 // MatrixForm
MatrixPlot[xmatrix0,
ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
ColorFunctionScaling -> False]

Out[813]//MatrixForm=
(-0.110043 -0.995664 -0.623615 0.983242 -0.917451 0.417607 -0.738205 -0.440824 -0.958767 -0.091427 0.737914 0.171569 -0.35924 0.658274 0.575908 -0.747931 0.561222 -0.262022 -0.0171402 0.258044 -0.0616449 0.170966 0.626323 -0.35126 0.496259 -0.000122)

Out[814]=

A horizontal color bar with a scale from 1 to 50. The color transitions from white at 1 to dark red at 50, with intermediate colors including yellow, orange, and blue. The label "1" is at both ends of the scale, and tick marks are at 10, 20, 30, 40, and 50.

Out[815]//MatrixForm=
(-0.958767 -0.091427 0.737914 0.171569 -0.35924 0.658274 0.575908 -0.747931 0.561222 -0.262022 -0.0171402 0.258044 -0.0616449 0.170966 0.626323 -0.35126 0.496259 -0.000122)

Out[816]=

A horizontal color bar with a scale from 1 to 50. The color transitions from white at 1 to dark blue at 50, with intermediate colors including yellow, orange, and red. The label "1" is at both ends of the scale, and tick marks are at 10, 20, 30, 40, and 50.

Out[817]//MatrixForm=
(-0.262022 -0.0171402 0.258044 -0.0616449 0.170966 0.626323 -0.35126 0.496259 -0.000122)

Out[818]=

A horizontal color bar with a scale from 1 to 50. The color transitions from white at 1 to dark blue at 50, with intermediate colors including yellow, orange, and red. The label "1" is at both ends of the scale, and tick marks are at 10, 20, 30, 40, and 50.
```

7.2. Mathematica Code for 2-D Square Lattice with Monte Carlo Equilibration from Experimental Metastable State

```
In[820]:= f[x_, y_, 1] := RandomReal[2 π] (*The first position is theta.*)
f[x_, y_, 2] := RandomReal[π] (*The second position is phi.*)
d = 1;
2
l = 7; (*width*)
k = 7; (*height*)
steps = 50000; (*Number of iterations in Monte Carlo test*)
array = Array[f, {k, l, 2}] // MatrixForm
```

Out[824]//MatrixForm=

(1.05585)	(4.48431)	(5.79864)	(3.82348)	(2.48689)	(3.29857)	(4.47982)
(2.41317)	(2.837)	(0.518465)	(2.13684)	(1.4167)	(1.5024)	(1.621)
(4.78324)	(3.37635)	(5.64079)	(1.50534)	(5.35146)	(5.39741)	(2.64371)
(1.71164)	(1.16945)	(2.19427)	(2.63773)	(1.15909)	(2.41645)	(1.42475)
(2.29203)	(5.05255)	(5.01807)	(0.316856)	(3.20407)	(2.19508)	(6.15622)
0.0509193	(2.83943)	(0.198741)	(1.87955)	(0.340752)	(1.40515)	(0.648321)
(2.00266)	(1.58216)	(4.14662)	(4.92539)	(5.74984)	(0.499857)	(3.25048)
(0.933029)	(1.95719)	(2.44119)	(2.39329)	(2.08227)	(2.36641)	(2.29644)
(2.07505)	(5.50416)	(0.501597)	(4.4509)	(3.75605)	(3.94242)	(0.493489)
(0.63344)	(0.48262)	(1.66291)	(0.906069)	(0.622282)	(0.250594)	(2.36576)
(1.95014)	(5.94407)	(6.04432)	(4.4008)	(5.45802)	(5.30659)	(3.47308)
(0.882984)	(1.66929)	(0.982727)	(1.09854)	(2.94302)	(0.287934)	(1.65457)
(1.35047)	(5.99962)	(3.74365)	(3.60152)	(5.48962)	(2.65934)	(4.54014)
(1.48714)	(1.41772)	(2.03165)	(1.87378)	(1.54234)	(0.735606)	(1.50091)

```
In[825]:= array[[1, 1, 1, 1]] = π/2;
array[[1, 1, 1, 2]] = 3 π/16;
array[[1, 1, 2, 1]] = π/2;
array[[1, 1, 2, 2]] = 0;
array[[1, 1, 3, 1]] = π/2;
array[[1, 1, 3, 2]] = 0;
array[[1, 1, 4, 1]] = π/2;
array[[1, 1, 4, 2]] = 0;
array[[1, 1, 5, 1]] = π/2;
array[[1, 1, 5, 2]] = 31 π/16;
array[[1, 1, 6, 1]] = π/2;
array[[1, 1, 6, 2]] = π/16;
array[[1, 1, 7, 1]] = π/2;
array[[1, 1, 7, 2]] = 27 π/16;

array[[1, 2, 1, 1]] = π/2;
array[[1, 2, 1, 2]] = 4 π/16;
array[[1, 2, 2, 1]] = π/2;
array[[1, 2, 2, 2]] = π;
array[[1, 2, 3, 1]] = π/2;
array[[1, 2, 3, 2]] = π;
array[[1, 2, 4, 1]] = π/2;
array[[1, 2, 4, 2]] = 15 π/16;
array[[1, 2, 5, 1]] = π/2;
array[[1, 2, 5, 2]] = 17 π/16;
array[[1, 2, 6, 1]] = π/2;
array[[1, 2, 6, 2]] = 13 π/16;
array[[1, 2, 7, 1]] = π/2;
array[[1, 2, 7, 2]] = 23 π/16;

array[[1, 3, 1, 1]] = π/2;
array[[1, 3, 1, 2]] = 3 π/8;
array[[1, 3, 2, 1]] = π/2;
```

```
array[[1, 3, 2, 2]] =  $\pi / 16$ ;
array[[1, 3, 3, 1]] =  $\pi / 2$ ;
array[[1, 3, 3, 2]] =  $15\pi / 4$ ;
array[[1, 3, 4, 1]] =  $\pi / 2$ ;
array[[1, 3, 4, 2]] =  $\pi / 4$ ;
array[[1, 3, 5, 1]] =  $\pi / 2$ ;
array[[1, 3, 5, 2]] =  $27\pi / 16$ ;
array[[1, 3, 6, 1]] =  $\pi / 2$ ;
array[[1, 3, 6, 2]] =  $7\pi / 16$ ;
array[[1, 3, 7, 1]] =  $\pi / 2$ ;
array[[1, 3, 7, 2]] =  $3\pi / 2$ ;

array[[1, 4, 1, 1]] =  $\pi / 2$ ;
array[[1, 4, 1, 2]] =  $9\pi / 16$ ;
array[[1, 4, 2, 1]] =  $\pi / 2$ ;
array[[1, 4, 2, 2]] =  $9\pi / 16$ ;
array[[1, 4, 3, 1]] =  $\pi / 2$ ;
array[[1, 4, 3, 2]] =  $23\pi / 16$ ;
array[[1, 4, 4, 1]] =  $\pi / 2$ ;
array[[1, 4, 4, 2]] =  $9\pi / 16$ ;
array[[1, 4, 5, 1]] =  $\pi / 2$ ;
array[[1, 4, 5, 2]] =  $3\pi / 2$ ;
array[[1, 4, 6, 1]] =  $\pi / 2$ ;
array[[1, 4, 6, 2]] =  $\pi / 2$ ;
array[[1, 4, 7, 1]] =  $\pi / 2$ ;
array[[1, 4, 7, 2]] =  $3\pi / 2$ ;

array[[1, 5, 1, 1]] =  $\pi / 2$ ;
array[[1, 5, 1, 2]] =  $\pi / 2$ ;
array[[1, 5, 2, 1]] =  $\pi / 2$ ;
array[[1, 5, 2, 2]] =  $\pi / 2$ ;
array[[1, 5, 3, 1]] =  $\pi / 2$ ;
array[[1, 5, 3, 2]] =  $3\pi / 2$ ;
array[[1, 5, 4, 1]] =  $\pi / 2$ ;
array[[1, 5, 4, 2]] =  $\pi / 2$ ;
array[[1, 5, 5, 1]] =  $\pi / 2$ ;
array[[1, 5, 5, 2]] =  $3\pi / 2$ ;
array[[1, 5, 6, 1]] =  $\pi / 2$ ;
array[[1, 5, 6, 2]] =  $7\pi / 16$ ;
array[[1, 5, 7, 1]] =  $\pi / 2$ ;
array[[1, 5, 7, 2]] =  $23\pi / 16$ ;

array[[1, 6, 1, 1]] =  $\pi / 2$ ;
array[[1, 6, 1, 2]] =  $7\pi / 16$ ;
array[[1, 6, 2, 1]] =  $\pi / 2$ ;
array[[1, 6, 2, 2]] =  $\pi / 2$ ;
array[[1, 6, 3, 1]] =  $\pi / 2$ ;
array[[1, 6, 3, 2]] =  $3\pi / 2$ ;
array[[1, 6, 4, 1]] =  $\pi / 2$ ;
array[[1, 6, 4, 2]] =  $7\pi / 16$ ;
array[[1, 6, 5, 1]] =  $\pi / 2$ ;
array[[1, 6, 5, 2]] =  $3\pi / 2$ ;
array[[1, 6, 6, 1]] =  $\pi / 2$ ;
array[[1, 6, 6, 2]] =  $3\pi / 8$ ;
array[[1, 6, 7, 1]] =  $\pi / 2$ ;
```

```

array[[1, 6, 7, 2]] = 23 π / 16;

array[[1, 7, 1, 1]] = π / 2;
array[[1, 7, 1, 2]] = 3 π / 4;
array[[1, 7, 2, 1]] = π / 2;
array[[1, 7, 2, 2]] = 15 π / 16;
array[[1, 7, 3, 1]] = π / 2;
array[[1, 7, 3, 2]] = 9 π / 8;
array[[1, 7, 4, 1]] = π / 2;
array[[1, 7, 4, 2]] = 15 π / 16;
array[[1, 7, 5, 1]] = π / 2;
array[[1, 7, 5, 2]] = 9 π / 8;
array[[1, 7, 6, 1]] = π / 2;
array[[1, 7, 6, 2]] = 13 π / 8;
array[[1, 7, 7, 1]] = π / 2;
array[[1, 7, 7, 2]] = 23 π / 16;
(*array[[1,1,1,1]]=π/2;
array[[1,1,1,2]]=π/2;
array[[1,1,2,1]]=π/2;
array[[1,1,2,2]]=π/2;
array[[1,1,3,1]]=π/2;
array[[1,1,3,2]]=−π/2;
array[[1,1,4,1]]=π/2;
array[[1,1,4,2]]=−π/2;
array[[1,2,1,1]]=π/2;
array[[1,2,1,2]]=π/2;
array[[1,2,2,1]]=π/2;
array[[1,2,2,2]]=π/2;
array[[1,2,3,1]]=π/2;
array[[1,2,3,2]]=−π/2;
array[[1,2,4,1]]=π/2;
array[[1,2,4,2]]=−π/2;
array[[1,3,1,1]]=π/2;
array[[1,3,1,2]]=π/2;
array[[1,3,2,1]]=π/2;
array[[1,3,2,2]]=π/2;
array[[1,3,3,1]]=π/2;
array[[1,3,3,2]]=π/2;
array[[1,3,4,1]]=π/2;
array[[1,3,4,2]]=π/2;
array[[1,4,1,1]]=π/2;
array[[1,4,1,2]]=−π/2;
array[[1,4,2,1]]=π/2;
array[[1,4,2,2]]=−π/2;
array[[1,4,3,1]]=π/2;
array[[1,4,3,2]]=−π/2;
array[[1,4,4,1]]=π/2;
array[[1,4,4,2]]=−π/2;*)
array

zmatrix0 = ConstantArray[0, {k, l}];

Do[Do[Part[zmatrix0, i, j] = Cos[array[[1, i, j, 1]]], {j, l}], {i, k}]
zmatrix0 // MatrixForm
MatrixPlot[zmatrix0,

```

```

ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
ColorFunctionScaling -> False]

ymatrix0 = ConstantArray[0, {k, 1}];

Do[Do[Part[ymatrix0, i, j] =
  Sin[array[[1, i, j, 1]]] * Sin[array[[1, i, j, 2]]], {j, 1}], {i, k}]
ymatrix0 // MatrixForm
MatrixPlot[ymatrix0,
  ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
  ColorFunctionScaling -> False]

xmatrix0 = ConstantArray[0, {k, 1}];

Do[Do[Part[xmatrix0, i, j] =
  Sin[array[[1, i, j, 1]]] * Cos[array[[1, i, j, 2]]], {j, 1}], {i, k}]
xmatrix0 // MatrixForm
MatrixPlot[xmatrix0,
  ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
  ColorFunctionScaling -> False]

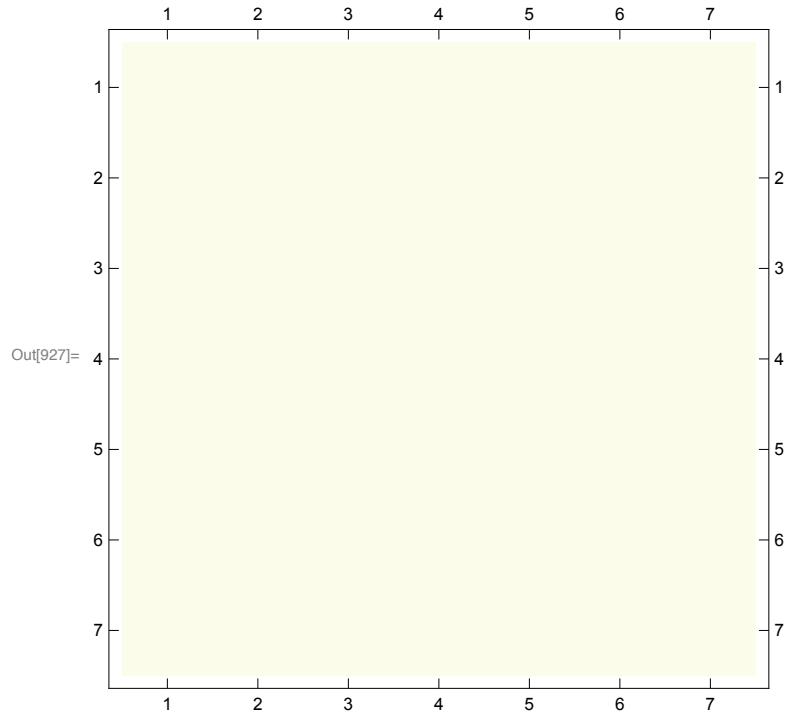
```

Out[923]//MatrixForm=

$$\begin{pmatrix} \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ 0 \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ 0 \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ 0 \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{31\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{27\pi}{16} \end{array}\right) \\ \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{\pi}{4} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \pi \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \pi \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{15\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{17\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{13\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{23\pi}{16} \end{array}\right) \\ \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{8} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{15\pi}{4} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ 4 \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{27\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{7\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{2} \end{array}\right) \\ \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{9\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{9\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{23\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{9\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{2} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ 2 \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{2} \end{array}\right) \\ \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{\pi}{2} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{\pi}{2} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{2} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ 2 \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{2} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{7\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{23\pi}{16} \end{array}\right) \\ \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{7\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{\pi}{2} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{2} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{7\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ 2 \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{8} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{23\pi}{16} \end{array}\right) \\ \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{3\pi}{4} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{15\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{9\pi}{8} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{15\pi}{16} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{9\pi}{8} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{13\pi}{8} \end{array}\right) & \left(\begin{array}{c} \frac{\pi}{2} \\ \frac{23\pi}{16} \end{array}\right) \end{pmatrix}$$

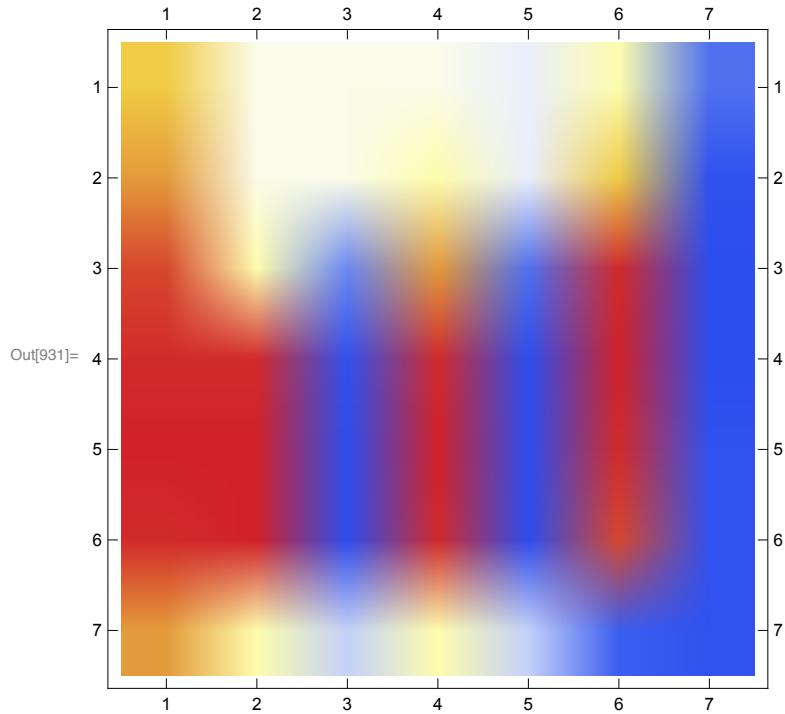
Out[926]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



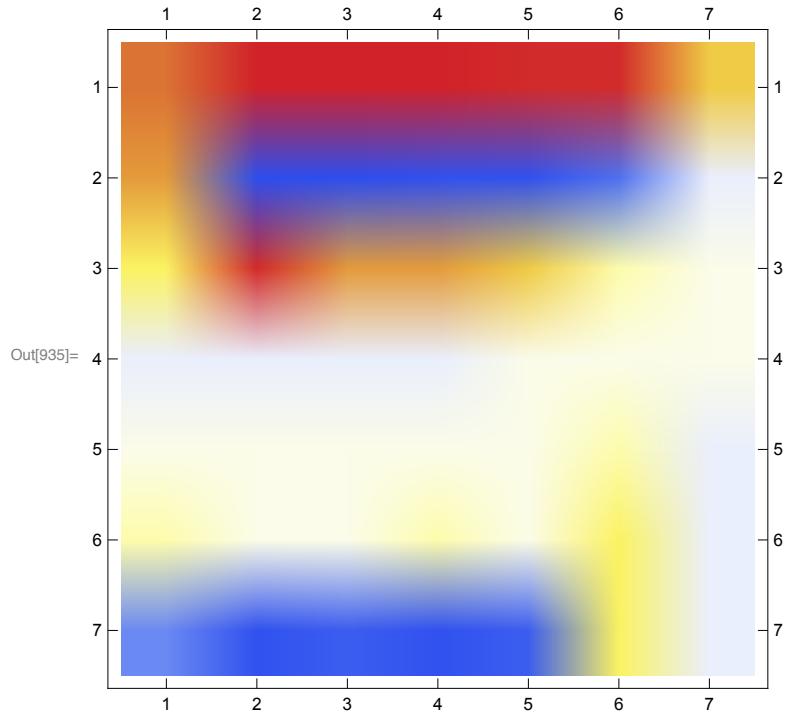
Out[930]:= MatrixForm=

$$\left(\begin{array}{ccccccc} \sin\left[\frac{3\pi}{16}\right] & 0 & 0 & 0 & -\sin\left[\frac{\pi}{16}\right] & \sin\left[\frac{\pi}{16}\right] & -\cos\left[\frac{3\pi}{16}\right] \\ \frac{1}{\sqrt{2}} & 0 & 0 & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & \sin\left[\frac{3\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] \\ \cos\left[\frac{\pi}{8}\right] & \sin\left[\frac{\pi}{16}\right] & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\cos\left[\frac{3\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & -1 \\ \cos\left[\frac{\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & \cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] \\ \cos\left[\frac{\pi}{16}\right] & 1 & -1 & \cos\left[\frac{\pi}{16}\right] & -1 & \cos\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{16}\right] \\ \frac{1}{\sqrt{2}} & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{8}\right] & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{16}\right] \end{array} \right)$$



Out[934]//MatrixForm=

$$\left(\begin{array}{ccccccc} \cos\left[\frac{3\pi}{16}\right] & 1 & 1 & 1 & \cos\left[\frac{\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & \sin\left[\frac{3\pi}{16}\right] \\ \frac{1}{\sqrt{2}} & -1 & -1 & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{3\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] \\ \sin\left[\frac{\pi}{8}\right] & \cos\left[\frac{\pi}{16}\right] & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \sin\left[\frac{3\pi}{16}\right] & \sin\left[\frac{\pi}{16}\right] & 0 \\ -\sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] \\ \sin\left[\frac{\pi}{16}\right] & 0 & 0 & \sin\left[\frac{\pi}{16}\right] & 0 & \sin\left[\frac{\pi}{8}\right] & -\sin\left[\frac{\pi}{16}\right] \\ -\frac{1}{\sqrt{2}} & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{8}\right] & \sin\left[\frac{\pi}{8}\right] & -\sin\left[\frac{\pi}{16}\right] \end{array} \right)$$



In[936]:=

In[937]:=

In[938]:=

In[939]:=

In[940]:=

```

In[941]:= xdisp1[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] :=
  (x2 - x1) + (d / 2) (Sin[θ2] Cos[φ2] - Sin[θ1] Cos[φ1]);
xdisp2[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] :=
  (x2 - x1) + (d / 2) (-Sin[θ2] Cos[φ2] + Sin[θ1] Cos[φ1]);
xdisp3[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (x2 - x1) + (d / 2) (-Sin[θ2] Cos[φ2] - Sin[θ1] Cos[φ1]);
xdisp4[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (x2 - x1) + (d / 2) (Sin[θ2] Cos[φ2] + Sin[θ1] Cos[φ1]);

ydisp1[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] :=
  (y2 - y1) + (d / 2) (Sin[θ2] Sin[φ2] - Sin[θ1] Sin[φ1]);
ydisp2[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (y2 - y1) + (d / 2) (-Sin[θ2] Sin[φ2] + Sin[θ1] Sin[φ1]);
ydisp3[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (y2 - y1) + (d / 2) (-Sin[θ2] Sin[φ2] - Sin[θ1] Sin[φ1]);
ydisp4[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] :=
  (y2 - y1) + (d / 2) (Sin[θ2] Sin[φ2] + Sin[θ1] Sin[φ1]);

zdisp1[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] := (d / 2) (Cos[θ2] - Cos[θ1]);
zdisp2[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := (d / 2) (-Cos[θ2] + Cos[θ1]);
zdisp3[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := (d / 2) (-Cos[θ2] - Cos[θ1]);
zdisp4[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := (d / 2) (Cos[θ2] + Cos[θ1]);

11[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] :=
  √(xdisp1[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + ydisp1[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  zdisp1[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);
12[x1_, y1_, x2_, y2_, θ1_, θ2_, φ1_, φ2_] := √(xdisp2[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  ydisp2[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + zdisp2[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);
13[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := √(xdisp3[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  ydisp3[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + zdisp3[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);
14[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := √(xdisp4[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 +
  ydisp4[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2 + zdisp4[x1, y1, x2, y2, θ1, φ1, θ2, φ2]^2);
energy[x1_, y1_, x2_, y2_, θ1_, φ1_, θ2_, φ2_] := Piecewise[{{0, x1 == x2 && y1 == y2}},

  1
  -----
  11[x1, y1, x2, y2, θ1, φ1, θ2, φ2]
  1
  -----
  12[x1, y1, x2, y2, θ1, φ1, θ2, φ2]
  1
  -----
  13[x1, y1, x2, y2, θ1, φ1, θ2, φ2]
  1
  -----
  14[x1, y1, x2, y2, θ1, φ1, θ2, φ2]
];

```

```
In[958]:= Do[randx = RandomInteger[{1, 1}];
  randy = RandomInteger[{1, k}];
  θp = Part[array, 1, randy, randx, 1];
  φp = Part[array, 1, randy, randx, 2];
  θn = RandomReal[2 π];
  φn = RandomReal[π];
  xi = 1;
  yi = 1;
  totalenergynew = 0;
  totalenergypast = 0;
  While[yi < k, yi++;
    While[xi < 1, xi++];
    totalenergynew = totalenergynew + (If[xi == randx, 0, 1]) * energy[randx, randy, xi,
      yi, θn, φn, Part[array, 1, yi, xi, 1], Part[array, 1, yi, xi, 2]]];
    xi = 1;
    yi = 1;
    While[yi < k, yi++;
      While[xi < 1, xi++];
      totalenergypast = totalenergypast + (If[xi == randx, 0, 1]) * energy[randx, randy,
        xi, yi, θp, φp, Part[array, 1, yi, xi, 1], Part[array, 1, yi, xi, 2]]];
    ];
  If[totalenergynew < totalenergypast, Part[array, 1, randy, randx, 1] = θn, 0];
  If[totalenergynew < totalenergypast, Part[array, 1, randy, randx, 2] = φn, 0];
, steps]

In[959]:= zmatrix = ConstantArray[0, {k, 1}];

Do[Do[Part[zmatrix, i, j] = Cos[array[[1, i, j, 1]]], {j, 1}], {i, k}]
zmatrix // MatrixForm
MatrixPlot[zmatrix,
 ColorFunction → (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling → False]

ymatrix = ConstantArray[0, {k, 1}];

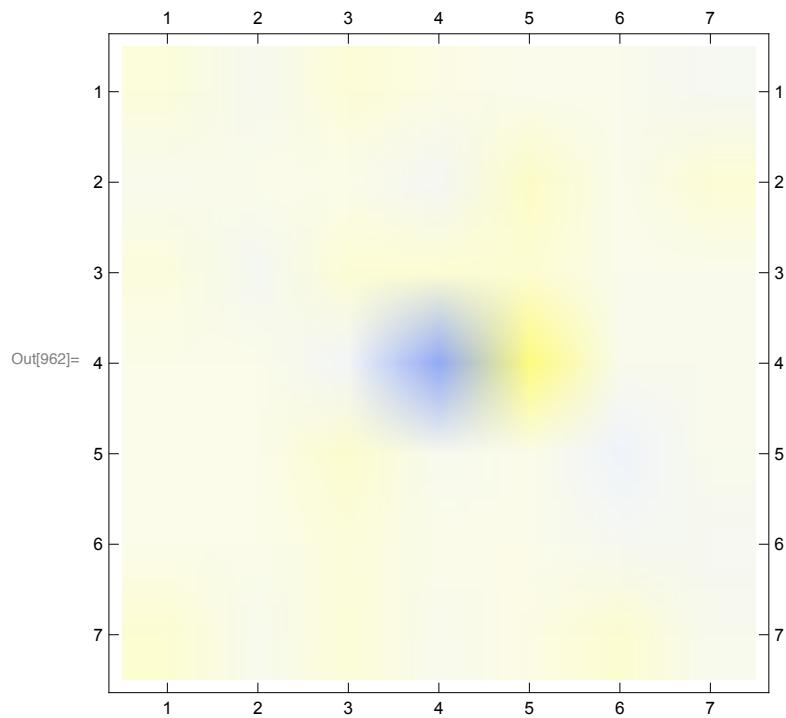
Do[Do[Part[ymatrix, i, j] = Sin[array[[1, i, j, 1]]] * Sin[array[[1, i, j, 2]]], {j, 1}],
 {i, k}]
ymatrix // MatrixForm
MatrixPlot[ymatrix,
 ColorFunction → (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling → False]

xmatrix = ConstantArray[0, {k, 1}];

Do[Do[Part[xmatrix, i, j] = Sin[array[[1, i, j, 1]]] * Cos[array[[1, i, j, 2]]], {j, 1}],
 {i, k}]
xmatrix // MatrixForm
MatrixPlot[xmatrix,
 ColorFunction → (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
 ColorFunctionScaling → False]
```

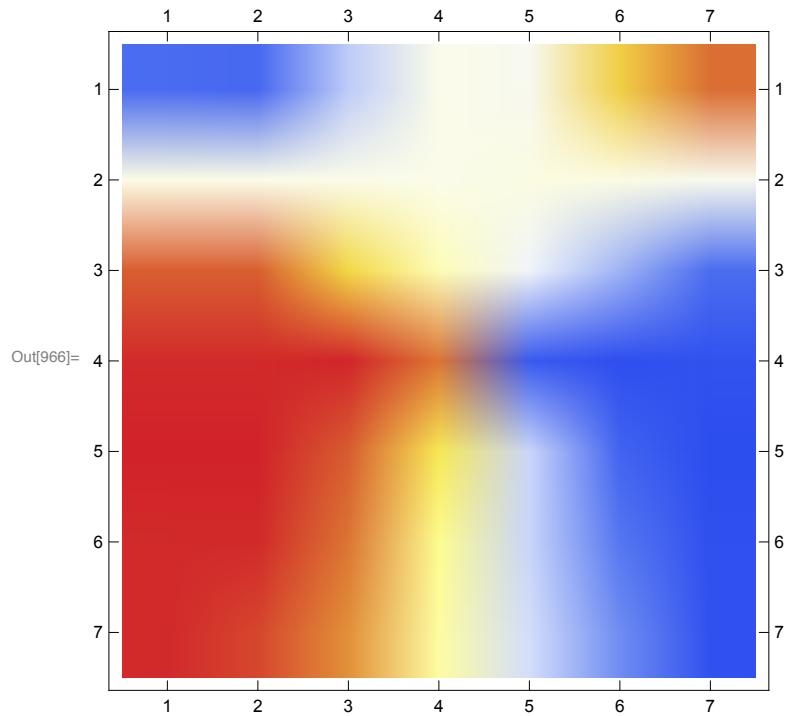
Out[961]//MatrixForm=

$$\begin{pmatrix} 0.0485737 & -0.0541072 & 0.0637119 & 0.0140646 & -0.0157757 & -0.0259063 & -0.072103 \\ -0.0354037 & 0 & 0 & -0.0890094 & 0.114947 & -0.0165234 & 0.0653483 \\ 0.0418292 & -0.0810818 & 0.0691154 & 0.0732179 & 0.0775651 & -0.0133037 & -0.0159738 \\ 0 & 0 & -0.11664 & -0.55913 & 0.291734 & -0.029736 & -0.0363627 \\ 0 & 0 & 0.0941922 & -0.0469433 & -0.011405 & -0.152455 & -0.0267837 \\ 0 & -0.00923867 & 0.0508361 & 0.000767105 & -0.0119266 & -0.063608 & -0.0697282 \\ 0.0810393 & -0.0411175 & 0.0615391 & -0.0360342 & 0.0135371 & 0.0855262 & -0.0407769 \end{pmatrix}$$



Out[965]//MatrixForm=

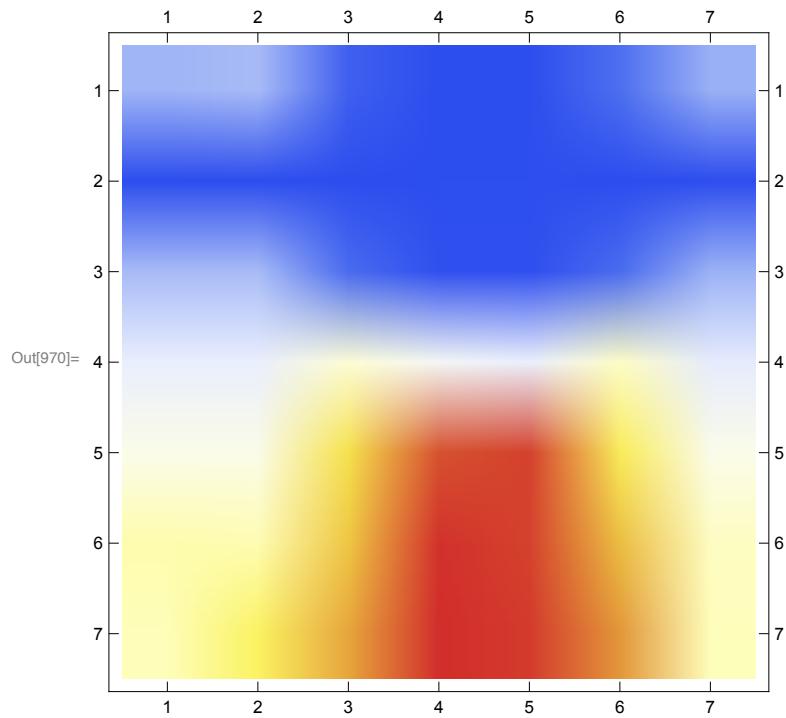
$$\begin{pmatrix} -0.857637 & -0.872139 & -0.404511 & -0.0060006 & -0.0528916 & 0.543383 & 0.842044 \\ 0.0136599 & 0 & 0 & -0.0118866 & 0.0259257 & 0.0159606 & -0.042586 \\ 0.873616 & 0.872729 & 0.509758 & 0.172242 & -0.1361 & -0.515728 & -0.850129 \\ \text{Cos} \left[\frac{\pi}{16} \right] & \text{Cos} \left[\frac{\pi}{16} \right] & 0.990844 & 0.822854 & -0.937845 & -0.992208 & -0.975858 \\ 1 & 1 & 0.877085 & 0.433909 & -0.359434 & -0.895683 & -0.999628 \\ \text{Cos} \left[\frac{\pi}{16} \right] & 0.98343 & 0.822224 & 0.256891 & -0.363146 & -0.799332 & -0.987846 \\ 0.983018 & 0.924389 & 0.738296 & 0.217103 & -0.317056 & -0.687441 & -0.985595 \end{pmatrix}$$



```
Out[969]//MatrixForm=

$$\begin{pmatrix} -0.511956 & -0.486257 & -0.912311 & -0.999883 & -0.998476 & -0.839085 & -0.534569 \\ -0.99928 & -1 & -1 & -0.99596 & -0.993033 & -0.999736 & -0.996953 \\ -0.484814 & -0.481426 & -0.857537 & -0.98233 & -0.987654 & -0.856649 & -0.526333 \\ -\text{Sin}\left[\frac{\pi}{16}\right] & -\text{Sin}\left[\frac{\pi}{16}\right] & 0.068001 & -0.101412 & -0.187984 & 0.120989 & -0.21536 \\ 0 & 0 & 0.47101 & 0.899733 & 0.933101 & 0.417744 & -0.00518485 \\ \text{Sin}\left[\frac{\pi}{16}\right] & 0.18105 & 0.566889 & 0.96644 & 0.931656 & 0.597514 & 0.138917 \\ 0.164646 & 0.379228 & 0.671663 & 0.975483 & 0.94831 & 0.721187 & 0.164131 \end{pmatrix}$$

```



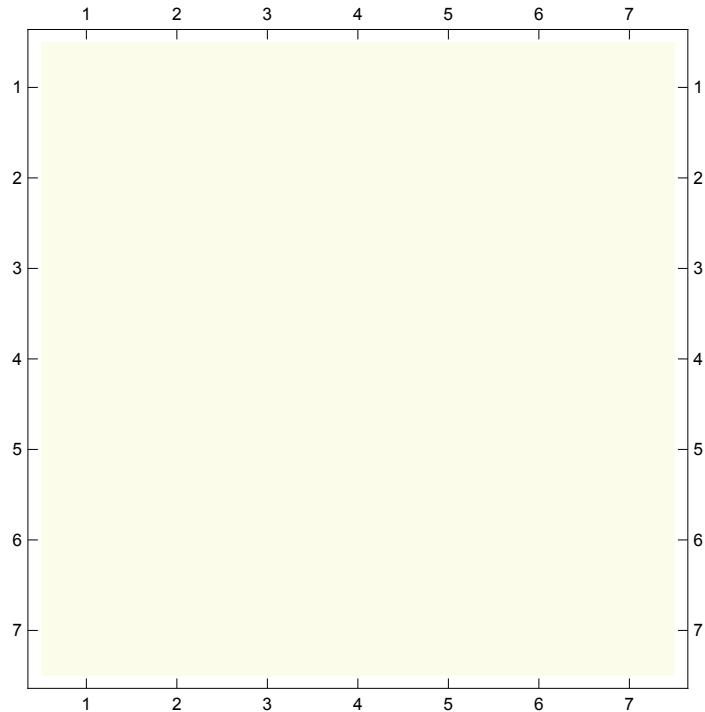
```
In[971]:= 

In[972]:= 

In[973]:= zmatrix0 // MatrixForm
MatrixPlot[zmatrix0,
  ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
  ColorFunctionScaling -> False]
ymatrix0 // MatrixForm
MatrixPlot[ymatrix0,
  ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
  ColorFunctionScaling -> False]
xmatrix0 // MatrixForm
MatrixPlot[xmatrix0,
  ColorFunction -> (ColorData["TemperatureMap"] [Rescale[#, {-1, 1}]] &),
  ColorFunctionScaling -> False]
```

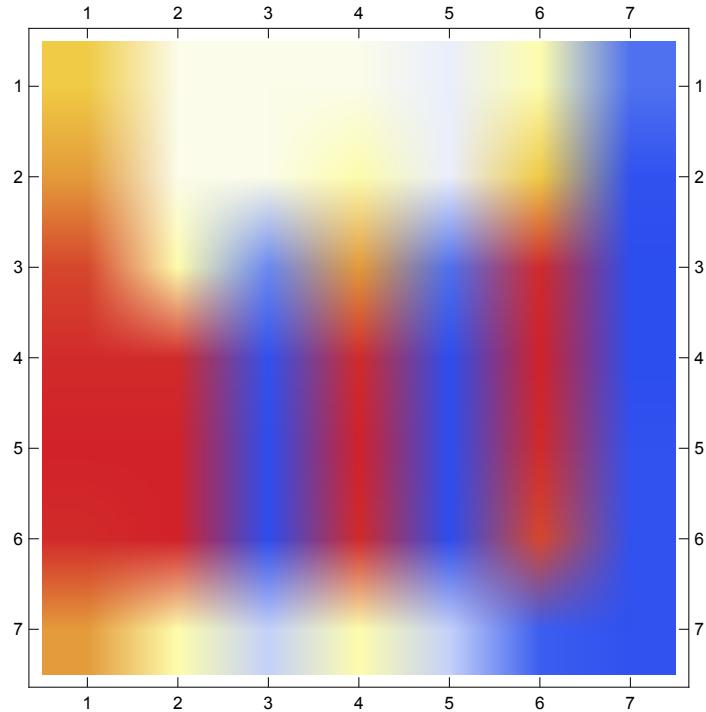
Out[973]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Out[975]//MatrixForm=

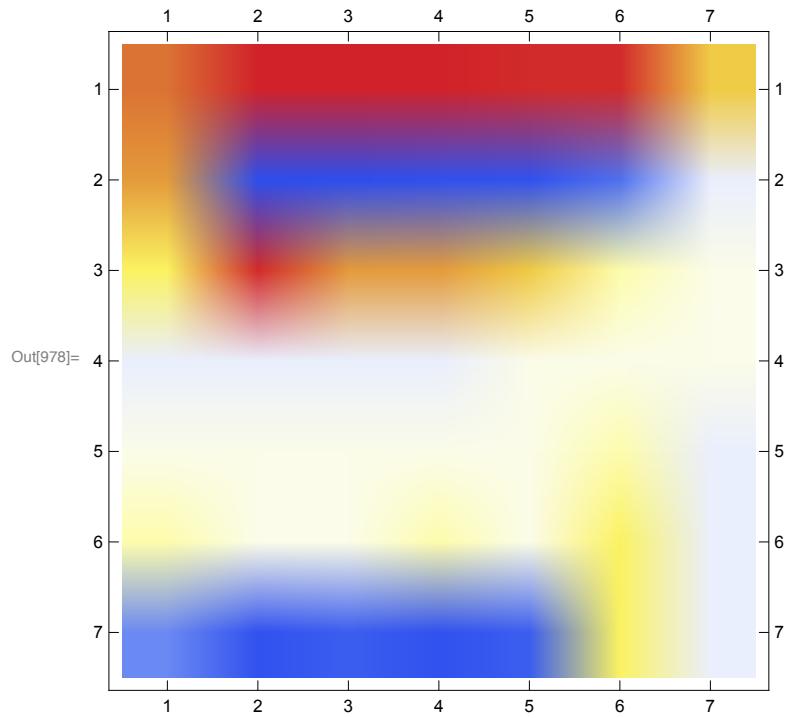
$$\begin{pmatrix} \sin\left[\frac{3\pi}{16}\right] & 0 & 0 & 0 & -\sin\left[\frac{\pi}{16}\right] & \sin\left[\frac{\pi}{16}\right] & -\cos\left[\frac{3\pi}{16}\right] \\ \frac{1}{\sqrt{2}} & 0 & 0 & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & \sin\left[\frac{3\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] \\ \cos\left[\frac{\pi}{8}\right] & \sin\left[\frac{\pi}{16}\right] & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\cos\left[\frac{3\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & -1 \\ \cos\left[\frac{\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & \cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] \\ \cos\left[\frac{\pi}{16}\right] & 1 & -1 & \cos\left[\frac{\pi}{16}\right] & -1 & \cos\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{16}\right] \\ \frac{1}{\sqrt{2}} & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{8}\right] & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{16}\right] \end{pmatrix}$$



Out[976]=

Out[977]//MatrixForm=

$$\begin{pmatrix} \cos\left[\frac{3\pi}{16}\right] & 1 & 1 & 1 & \cos\left[\frac{\pi}{16}\right] & \cos\left[\frac{\pi}{16}\right] & \sin\left[\frac{3\pi}{16}\right] \\ \frac{1}{\sqrt{2}} & -1 & -1 & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{3\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] \\ \sin\left[\frac{\pi}{8}\right] & \cos\left[\frac{\pi}{16}\right] & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \sin\left[\frac{3\pi}{16}\right] & \sin\left[\frac{\pi}{16}\right] & 0 \\ -\sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sin\left[\frac{\pi}{16}\right] & -\sin\left[\frac{\pi}{16}\right] \\ \sin\left[\frac{\pi}{16}\right] & 0 & 0 & \sin\left[\frac{\pi}{16}\right] & 0 & \sin\left[\frac{\pi}{8}\right] & -\sin\left[\frac{\pi}{16}\right] \\ -\frac{1}{\sqrt{2}} & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{8}\right] & -\cos\left[\frac{\pi}{16}\right] & -\cos\left[\frac{\pi}{8}\right] & \sin\left[\frac{\pi}{8}\right] & -\sin\left[\frac{\pi}{16}\right] \end{pmatrix}$$



In[979]:=