Analytical and Computational Study of Multi-temperature Kinetic Ising Models on Various Graphs

Sho Gibbs

Department of Physics and Engineering, Washington and Lee University

E-mail: gibbss21@mail.wlu.edu / shogibbs@gmail.com

Submitted in partial fulfillment of graduation requirements for University Honors

Abstract. We analyze a class of multi-temperature kinetic Ising models with Glauber dynamics on both one-dimensional lattices and Cayley Trees, focusing on the overall and sublattice magnetizations. Expectation values for one-dimensional systems are determined numerically via ordinary differential equation solvers, and Monte Carlo methods are applied for Cayley Trees. We describe both the relaxation dynamics and non-equilibrium steady state values arising from different temperature patterns and explore common behaviors exhibited by our systems. We also discuss interesting physical analogies arising from the numerical lines of best fit. Finally, we address potential future work on our multi-temperature kinetic Ising model.

Contents

1. Introduction

Systems in thermal equilibrium are well understood at the microscopic scale thanks to the formalism of statistical mechanics developed by Boltzmann, Gibbs, and many others [1]. For systems in equilibrium, the partition function is the key to computing physical quantities such as the internal energy, the free energy and the entropy of the system [2]. Every undergraduate physics major can confirm how tedious (and sometimes impossible) this process can be for large systems. However, this approach to most systems in thermal equilibrium is well established and forms a solid theoretical bedrock for the study of more exotic and interesting behaviors.

On the other hand, no such approach exists in general for non-equilibrium systems. There is not a clear strategy for how to tackle the problems, which poses a problem for researchers [1]. Developing a better understanding of systems far from equilibrium is recognized as a key goal within condensed matter and materials physics [3]. The importance of this challenge is apparent when one considers the vast range of systems which would fall under the umbrella of non-equilibrium models.

One popular method to dig into the mystery of systems out of thermal equilibrium is the use of toy models [1, 3, 4, 5, 6, 7, 8, 9]. The aim is to work with simple and more easily understood systems to gain insight into the broader category of non-equilibrium systems. In doing so, we hope to lay the foundation for a standard non-equilibrium formalism for approaching a wide range of problems. In this thesis we study a variation of the Ising model, which is a popular toy model due to its simplicity.

1.1. Model Background

The Ising model concerns a lattice of N particles that have spins of $\sigma_i(t) = \pm 1$, where i is an integer $1 \leq i \leq N$ [10]. In its simplest form, its energy is determined by an exchange constant J, a magnetic moment μ, a magnetic field h_i that depends on the site i, and the relative spin orientations of all interacting particles, with a Hamiltonian as shown below in Equation 1 where the sum is over all neighbor pairs [11]. Typically J is taken to be positive so that it costs less energy for a particle to be in the same spin orientation as its neighbors, and in some generalizations it can vary depending on the specific particle pair in question.

$$\hat{\mathrm{H}} = -J \sum_{\{i,j\}} \sigma_i \, \sigma_j + \mu \sum_{\{i\}} h_i \, \sigma_i \tag{1}$$

The multi-temperature kinetic Ising model presented in this thesis is a generalization of Glauber's work in 1963 [12]. He introduced a one-dimensional ring with N particles that have spins $\sigma_i(t) = \pm 1$. The system evolves in time by having a random particle flip at a probabilistic rate which is determined by the spins of its two neighbors, a temperature bath connected to the system, and a constant magnetic field. The transition rate for a particle in the Glauber model is given by the following equation:

$$w_i(\sigma_i) = \frac{1}{2}\,\alpha\,(1 - \frac{1}{2}\,\gamma\,\sigma_i\,(\sigma_{i-1} + \sigma_{i+1})) \cdot (1 - \beta\,\sigma_i) \tag{2}$$

From some analysis and via correspondence to the traditional Ising model, the parameters α, β, γ can be understood to correspond to a general spin flipping rate, the magnetic field effect, and the inverse temperature respectively. Although α does not have an analogue in the Ising model, Glauber explains that it varies the timescale of the spin flipping dynamics and must have a value $0 \leq \alpha \leq 1$, where 0 prevents any spin flipping. Meanwhile $\beta \equiv \tanh(\frac{\mu H}{k_B T})$ is defined in terms of a magnetic field H, temperature T, and constants μ and k_B. The final parameter $\gamma \equiv \tanh(\frac{2J}{k_B T})$ involves a coupling constant J and the Boltzmann constant k_B but is entirely a function of temperature.

The most complete description of this system is the set of probability functions $\{\mathrm{p}(\{\sigma\}, t)\}$. This is a set of 2^N functions varying in time, each tracking the probability that the spins will be in a specific state at the given time. As Glauber notes, this is generally far more information than is necessary or even useful in practice. As a result, he solves for the expectation values of spins $q_i(t) \equiv < \sigma_i(t) >$ and the correlation functions between spins $r_{i,j}(t) \equiv < \sigma_i(t)\sigma_j(t) >$, as these can be used to describe the system at a more reasonable scale of information.

We will illustrate the relevant parts of Glauber's analysis by looking at a toy model where the ring only has two particles. We will also take $\alpha = 1$ and $\beta = 0$, allowing spins to flip without resistance and removing any external field effects. This reduces our flip rate to $w_i = \frac{1}{2}(1 - \gamma\,\sigma_1\,\sigma_2)$, as periodic boundary conditions mean $\sigma_{i-1} = \sigma_{i+1}$. The explicit transition rates are shown below, where each of the four system states $(\{+1, +1\},\ \{+1, -1\},\ \{-1, +1\},\ \{-1, -1\})$ can transition to two different states:

$$
\begin{aligned}
w(\{+1,+1\} \to \{+1,-1\}) &= \frac{1}{2}(1-\gamma) \\
w(\{+1,+1\} \to \{-1,+1\}) &= \frac{1}{2}(1-\gamma) \\
w(\{+1,-1\} \to \{+1,+1\}) &= \frac{1}{2}(1+\gamma) \\
w(\{+1,-1\} \to \{-1,-1\}) &= \frac{1}{2}(1+\gamma) \\
w(\{-1,+1\} \to \{+1,+1\}) &= \frac{1}{2}(1+\gamma) \\
w(\{-1,+1\} \to \{-1,-1\}) &= \frac{1}{2}(1+\gamma) \\
w(\{-1,-1\} \to \{+1,-1\}) &= \frac{1}{2}(1-\gamma) \\
w(\{-1,-1\} \to \{-1,+1\}) &= \frac{1}{2}(1-\gamma)
\end{aligned}
\tag{3}
$$

A system of differential equations can be written for these probability functions using our knowledge of the transition rate and the constraint of one particle flipping at a time. Note that this system is normalized by $\sum_{\{\sigma\}} \mathrm{p}(\{\sigma\},t) = 1$, where $\{\sigma\}$ represents an arbitrary state of the system, meaning that all the probabilities add up to unity.

$$
\begin{aligned}
\frac{\partial}{\partial t}\mathrm{p}(\{+1,+1\},t) = &-(1-\gamma)\,\mathrm{p}(\{+1,+1\},t) \\
&+\frac{1}{2}(1+\gamma)\,\mathrm{p}(\{-1,+1\},t) + \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{+1,-1\},t) \\
\frac{\partial}{\partial t}\mathrm{p}(\{+1,-1\},t) = &-(1+\gamma)\,\mathrm{p}(\{+1,-1\},t) \\
&+\frac{1}{2}(1-\gamma)\,\mathrm{p}(\{+1,+1\},t) + \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{-1,-1\},t) \\
\frac{\partial}{\partial t}\mathrm{p}(\{-1,+1\},t) = &-(1+\gamma)\,\mathrm{p}(\{-1,+1\},t) \\
&+\frac{1}{2}(1-\gamma)\,\mathrm{p}(\{+1,+1\},t) + \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{-1,-1\},t) \\
\frac{\partial}{\partial t}\mathrm{p}(\{-1,-1\},t) = &-(1-\gamma)\,\mathrm{p}(\{-1,-1\},t) \\
&+\frac{1}{2}(1+\gamma)\,\mathrm{p}(\{-1,+1\},t) + \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{+1,-1\},t)
\end{aligned}
\tag{4}
$$

We now derive expressions for the expectation values. In general these are defined as $q_j(t) = \sum_{\{\sigma_i\}} \sigma_j(t)\,\mathrm{p}(\{\sigma_i\},t)$, where σ_j is a specific (the j^{th}) particle in the system.

$$
\begin{aligned}
q_1(t) &= +\,\mathrm{p}(\{+1,+1\},t) + \mathrm{p}(\{+1,-1\},t) - \mathrm{p}(\{-1,+1\},t) - \mathrm{p}(\{-1,-1\},t) \\
q_2(t) &= +\,\mathrm{p}(\{+1,+1\},t) - \mathrm{p}(\{+1,-1\},t) + \mathrm{p}(\{-1,+1\},t) - \mathrm{p}(\{-1,-1\},t)
\end{aligned}
\tag{5}
$$

Next we take a derivative with respect to time on both sides. This allows us to find an expression for the change in $q_i(t)$ with respect to the derivatives of the probability functions, which can then be substituted for an expression in terms of the probability functions. Below we have the derivatives of the expectation values in term of the probability functions before like terms have been collected. Each row on the right hand side is an expression which has been substituted using the probability function time derivatives found above.

$$\begin{aligned}
\frac{\partial}{\partial t} q_1(t) = &- (1-\gamma)\,\mathrm{p}(\{+1,+1\},t) + \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{-1,+1\},t) + \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{+1,-1\},t) \\
&- (1+\gamma)\,\mathrm{p}(\{+1,-1\},t) + \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{+1,+1\},t) + \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{-1,-1\},t) \\
&+ (1+\gamma)\,\mathrm{p}(\{-1,+1\},t) - \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{+1,+1\},t) - \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{-1,-1\},t) \\
&+ (1-\gamma)\,\mathrm{p}(\{-1,-1\},t) - \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{-1,+1\},t) - \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{+1,-1\},t) \\
\frac{\partial}{\partial t} q_2(t) = &- (1-\gamma)\,\mathrm{p}(\{+1,+1\},t) + \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{-1,+1\},t) + \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{+1,-1\},t) \\
&+ (1+\gamma)\,\mathrm{p}(\{+1,-1\},t) - \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{+1,+1\},t) - \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{-1,-1\},t) \\
&- (1+\gamma)\,\mathrm{p}(\{-1,+1\},t) + \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{+1,+1\},t) + \frac{1}{2}(1-\gamma)\,\mathrm{p}(\{-1,-1\},t) \\
&+ (1-\gamma)\,\mathrm{p}(\{-1,-1\},t) - \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{-1,+1\},t) - \frac{1}{2}(1+\gamma)\,\mathrm{p}(\{+1,-1\},t)
\end{aligned} \tag{6}$$

Collecting like terms results in the expressions below. They have been organized to make the subsequent substitutions of $q_1(t)$ and $q_2(t)$ clear.

$$\begin{aligned}
\frac{\partial}{\partial t} q_1(t) = &- 1 \cdot \big(\,\mathrm{p}(\{+1,+1\},t) + \mathrm{p}(\{+1,-1\},t) - \mathrm{p}(\{-1,+1\},t) - \mathrm{p}(\{-1,-1\},t)\,\big) \\
&+ \gamma \cdot \big(\,\mathrm{p}(\{+1,+1\},t) - \mathrm{p}(\{+1,-1\},t) + \mathrm{p}(\{-1,+1\},t) - \mathrm{p}(\{-1,-1\},t)\,\big) \\
\frac{\partial}{\partial t} q_2(t) = &- 1 \cdot \big(\,\mathrm{p}(\{+1,+1\},t) - \mathrm{p}(\{+1,-1\},t) + \mathrm{p}(\{-1,+1\},t) - \mathrm{p}(\{-1,-1\},t)\,\big) \\
&+ \gamma \cdot \big(\,\mathrm{p}(\{+1,+1\},t) + \mathrm{p}(\{+1,-1\},t) - \mathrm{p}(\{-1,+1\},t) - \mathrm{p}(\{-1,-1\},t)\,\big)
\end{aligned} \tag{7}$$

Finally, we can make this a system of equations in terms of the expectation values. Importantly, the equations below are a special case of Glauber's more general result of $\frac{\partial}{\partial t} q_i = -q_i + \frac{\gamma}{2}(q_{i-1} + q_{i+1})$, where Equation 8 is simplified because $\sigma_{i-1} = \sigma_{i+1}$ when the system size is two.

$$\begin{aligned} \frac{\partial}{\partial t}\, q_1(t) &= -\, q_1 + \gamma\, q_2 \\ \frac{\partial}{\partial t}\, q_2(t) &= -\, q_2 + \gamma\, q_1 \end{aligned} \tag{8}$$

Glauber also finds expressions for the spin correlations $r_{ij}(t)$ which are the average product of the spins. While he also treats time-delay correlations which are computed between two arbitrary particles at a constant time delay, we are most interested in the spin correlations computed at a single time in specific cases which invite simplifications. The translational invariance along the ring allows us to redefine them in terms of $m \equiv i - j$, and if the spins are also uniform and free of magnetic field influences they evolve in time as shown in the equation below for $m \neq 0$. Note that the derivative includes an α, which simply encodes any general resistance to flipping shared by the entire system.

$$\frac{\partial}{\partial(\alpha\, t)} r_m(t) = -2\, r_m(t) + \gamma\, (r_{m-1}(t) + r_{m+1}(t)) \tag{9}$$

Finally, Glauber also introduces an alternate approach to his system, formulated in terms of matrices. This takes the 2^N probability functions $p(\sigma_1, \sigma_2, , , , t)$ to each be components of a vector $\mathbf{p}$. He then defines a matrix M such that the master equation takes on the form below. Knowledge of the eigenvalues and eigenvectors of the associated transition matrix M leads to an exact solution for the probability distribution of the system.

$$\frac{\partial}{\partial t}\mathbf{p} = M\mathbf{p} \tag{10}$$

Glauber's kinetic Ising model was expanded by Racz and Zia in 1994, who introduced the study of a kinetic Ising model ring which has the spins alternate between two temperature baths as a heat source [13]. They break the ring into two sublattices split into even and odd numbered particles, with notation along the lines of T_e and T_o for variables or parameters which depend on the associated heat bath. Note that each particle in a sublattice is only neighbors with two particles which are both a part of the other sublattice. This simplifies to Glauber's model when the temperatures have the same value (i.e. $T_e = T_o$), but is a system out of thermal equilibrium in general. Energy flows through the system from the hotter heat bath to the colder one, meaning that the system evolves in time towards a non-equilibrium steady state.

They define their systems in terms of average energy (ε) and energy flux between the sublattices (j_ε) rather than the two temperatures of the sublattices. The expressions are given below in Equation 11, where J is our familiar coupling constant and $r_{n,n+1}$ is the two-spin correlation as defined by Glauber. The energy flux at a given particle n is shown in terms of $\gamma_n \equiv \tanh(\frac{2J}{k_B T_n})$ which depends on the particle's temperature T_n.

By considering cases when each sublattice is homogeneous they introduce translational invariance, removing the n-dependence and replacing it with a parity dependence. This formulation of the system in terms of energies is shown to be equivalent to using the two sublattice temperatures, and is preferred by Racz and Zia as it simplifies the expressions for most of their final results.

$$\begin{aligned} \varepsilon &= -J < \sigma_n \sigma_{n+1} >= -J\, r_{n,n+1} \\ j_\varepsilon(n) &= J\,\alpha\,(\gamma_n(1 + r_{n-1,n+1}) - (r_{n,n+1} + r_{n,n-1})) \end{aligned} \tag{11}$$

Their results include the two-spin correlations (with no time delay) for the non-equilibrium steady state and the analysis of sublattice magnetization dynamics. The correlations are found to form a closed set of differential equations ($i \neq j$):

$$\begin{aligned} \frac{2}{\alpha}\frac{\partial}{\partial t} < r_{i,j} >= &-4 < r_{i,j} > + \gamma_i(< r_{i-1,j} > + < r_{i+1,j} >) \\ &+ \gamma_j(< r_{i,j-1} > + < r_{i,j+1} >) \end{aligned} \tag{12}$$

Then they use the translational symmetry to categorize the equations of the system into three different types of correlations: odd-odd, even-even, and odd-even. In the 'thermodynamic limit' where the size of the system $N \to \infty$ these can be expressed in an exact manner as a function of γ_e, γ_o (which are both functions of the associated temperatures), and the relative displacement n where $0 \leq n$. They find that the correlations grow exponentially weaker as the displacement increases, an intuitively reasonable result for same-time correlations in the thermodynamic limit.

They also find individual spin expectation values, forming a system of equations of the form shown below in Equation 13. This can be seen as a direct generalization for the two-particle system which was expanded upon in the summary of Glauber's work above.

$$\frac{\partial}{\partial t} q_i = -q_i + \frac{\gamma_i}{2}(q_{i-1} + q_{i+1}) \tag{13}$$

This individual expectation value equation is used to find the aggregate sublattice magnetizations. It is clear that if both sublattices have homogeneous initial conditions that their expectation values will remain homogeneous through time evolution. Thus, the system of equations for the entire spin system reduce to the equations below, where $\Gamma_\pm = \alpha(1 \pm \sqrt{\gamma_o \gamma_e})$, and $a_\pm^{o/e}$ are constants determined by the initial conditions.

$$\begin{aligned} m_o(t) &= a_+^o\, \mathrm{e}^{-\Gamma_+ t} + a_-^o\, \mathrm{e}^{-\Gamma_- t} \\ m_e(t) &= a_+^e\, \mathrm{e}^{-\Gamma_+ t} + a_-^e\, \mathrm{e}^{-\Gamma_- t} \end{aligned} \tag{14}$$

1.2. Thesis Outline

In this thesis we will be looking at a kinetic Ising model generalized to an arbitrary number of temperatures and applied to a number of graphs. We will be focusing on numerical simulations of expectation values, with some additional work on Monte Carlo simulations. The focus is on overall and sublattice magnetizations as our macroscopic variable. We will describe the various forms of the model in terms of the graph and the configuration of temperature baths connected to it. While a significant portion of the modern literature on kinetic Ising models is formulated in terms of energy and energy flux, we will describe the exact system at hand in terms of temperature. These are equivalent frames as noted by Racz and Zia [13]. When generalizing to an arbitrary number of temperatures, the frame of energy flux can become cumbersome, especially in higher dimensions. This is because in order to properly categorize the system, one would want to know the energy flux between each temperature bath, making the number of relevant fluxes potentially larger than the number of temperatures and adding the complication of how to best organize the information about the fluxes. This is a problem unique to multi-temperature systems with more than two temperatures, and therefore does not have much representation in existing literature.

The goal of this thesis is to begin probing the multi-temperature kinetic Ising model in a novel way across a variety of graphs and configurations. This contributes to the broader literature on models of systems out of thermal equilibrium, ideally adding something new to the efforts to develop a framework for working with such systems. It may also be useful in understanding potential applications of this model. Two-state models have been applied to a great variety of systems, ranging from surface kinetics [14] to problems in epidemics [15, 16] and voting behavior [17]. Novel experimental models of the kinetic Ising model have been proposed [8], which is another reason to strive for a better understanding of the dynamics.

The thesis is structured as follows. First we will look at the one-dimensional ring, in the spirit of Racz and Zia. However, we expand the temperature bath patterns past two temperatures and also look at the cases where there are 'blocks' of temperatures. We take particular note of how the analytical work is complicated by these additional variations as well as a magnetic field. We do numerical work on the expectation values to show the variety of dynamics available and note some potential physical analogies, primarily how the one-dimensional systems appear to act as 'thermal damped oscillators'. This is behavior which has been described for specific configurations of the kinetic Ising model but has not been generalized as such in existing literature. Then, we move our attention towards the kinetic Ising model on Cayley Trees, and describe how this new graph presents additional challenges. In this case we use Monte Carlo simulations as they allow us to avoid some approximations necessitated by the graph structure, highlighting future work to be done in this direction. Finally, we present a conclusion which summarizes some patterns observed across the different graphs, and pose some questions for future investigation.

2. One-Dimensional Graphs

For one-dimensional graphs, we consider a chain of N particles as in Racz and Zia's study as seen in Figure 1 below. However, we generalize their formulation such that we have an arbitrary number of sublattices $X \geq 1$, each with an identical number of particles N/X (where $N \bmod X \equiv 0$) and their own associated temperature T_x. Each particle is in exactly one sublattice. It should be noted that temperatures can theoretically take on any real value ($-\infty \leq T_x \leq \infty$), where a negative temperature can be interpreted in the traditional statistical mechanical manner (where negative temperatures are a property of systems with finitely many energy states and are "hotter" than a temperature of positive infinity). However, we will be studying the ferromagnetic and more physically motivated cases where temperatures take on positive values. We also include the magnetic field as discussed by Glauber, keeping in mind that its influence on the spins was dependent on the temperature of the heat bath. While there is some literature on the effects of magnetic fields that are dynamic and change in time, we will look at constant fields. For simplicity we will set $\alpha = 1$, which means there is no systematic resistance to flipping. We can also let the constants $\mu = J = k_B = 1$ by choosing convenient units.

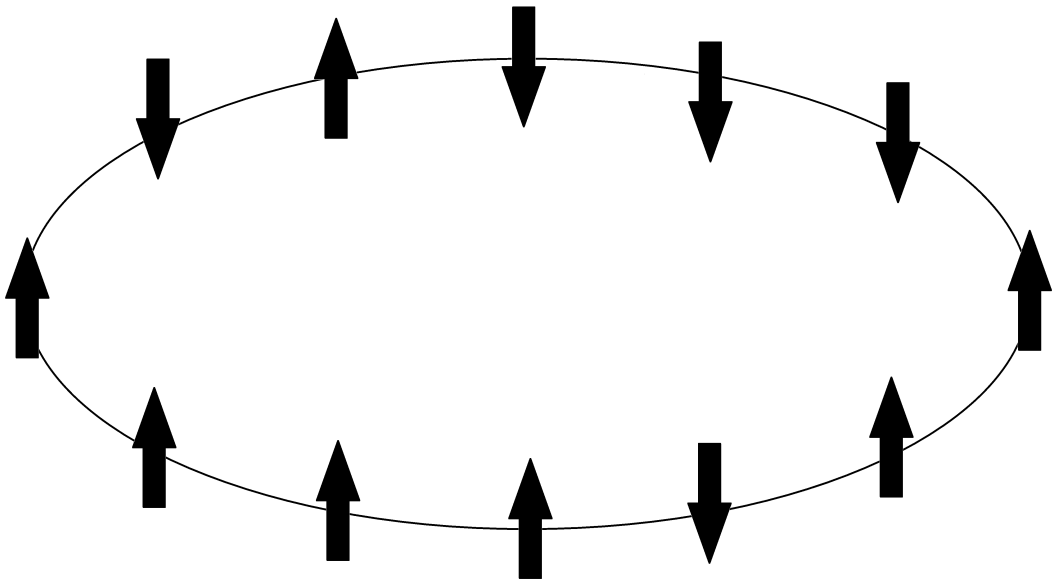

Figure 1: A chain of $N = 12$ particles with periodic boundaries.

Equation 15 below shows our transition rate for a given particle σ_i in terms of $\gamma_i = \tanh(\frac{2}{T_i})$ and $\beta_i = \tanh(\frac{H}{T_i})$. Recall from our summary of Glauber's work that H is the magnetic field's strength. While we define both γ and β for each particle σ_i, they are entirely determined by the sublattice temperature $T_i \equiv T_x$.

$$w_i(\sigma_i) = \frac{1}{2}(1 - \frac{1}{2}\,\gamma_i\,\sigma_i\,(\sigma_{i-1} + \sigma_{i+1})) \cdot (1 - \beta_i\,\sigma_i) \tag{15}$$

These transition rates are used in constructing the system of equations for the probability functions $\mathrm{p}(\{\sigma\})$. In order to keep the notation compact we introduce the idea of a particular alignment of spins $\{\sigma\}_i$ which differs from the alignment $\{\sigma\}$ only by the single spin σ_i flipping. We also allow the transition rate w_i to act upon the entire system rather than a singular particle, with the understanding that the transition rate is still the rate at which the single particle σ_i flips its spin. This results in Equation 16 below, which has the completely general change in a given state's probability function dependent on the rates of states flipping into and out of this given state at each particle site.

$$\frac{\partial\,\mathrm{p}(\{\sigma\},\,t)}{\partial t} = \sum_{i=1}^{N} \left[w_i(\{\sigma\}_i)\;\mathrm{p}(\{\sigma\}_i,\,t) - w_i(\{\sigma\})\;\mathrm{p}(\{\sigma\},\,t)\right] \tag{16}$$

While most of our analysis of these systems is done with periodic boundaries (i.e. σ_1 and σ_N are neighbors), they can also be constructed with hard wall boundaries. This would mean that the transition rate equations for the two end particles would each only have one neighboring particle. This introduces asymmetries in the system with regard to translation, providing an interesting additional angle of analysis.

Finally, we categorize the one-dimensional multi-temperature kinetic Ising models based on their sublattice structures. Even for the constrained case of a one-dimensional graph there are a variety of two-temperature models examined in the literature that we can draw from, starting with Racz and Zia's alternating temperatures to a clean half-and-half split into two "blocks", to a smaller window of a second temperature, to limiting cases involving zero or infinite temperatures. We broadly look into two categories, the alternating and block sublattice structures.

For an alternating structure with X sublattices, we can define the system by its X temperatures: $\{T_1,\,T_2,\,,\,,\,T_X\}$. Each spin σ_i in the first sublattice with T_1 is neighbors with spins σ_{i-1} and σ_{i+1} in the X^{th} and second sublattices respectively. The pattern continues with the second sublattice particles interacting with the first and third sublattice particles, and so on. In other words, the temperatures cycle through the set $\{T_1,\,T_2,\,,\,,\,T_X\}$ as one looks at the particles along the ring. This is visualized in Figure 2 below by coloring each spin based on its sublattice.

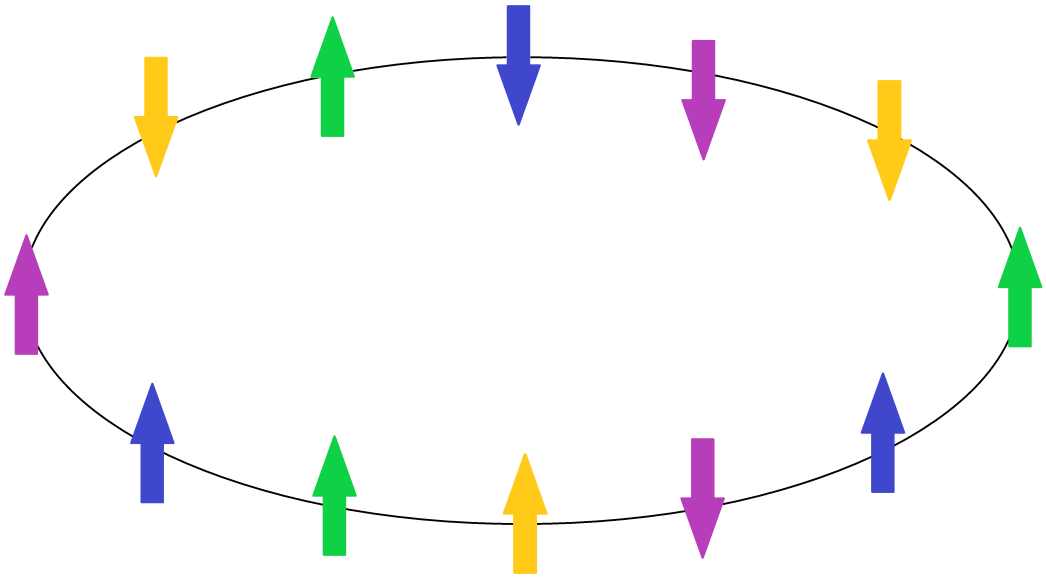

Figure 2: A chain of 12 particles with periodic boundaries. There are four sublattices, represented by blue, green, yellow, and purple arrows respectively. Note how the sublattice pairs of blue-yellow and green-purple do not have particles that interact with one another due to the alternating pattern and limited edges on our graph.

On the other hand, the temperature block structure has the particles in each sublattice clumped together into blocks. Each spin σ_i in the first sublattice with temperature T_1 is neighbors with spins connected to the same temperature bath, with the exception of "boundary" spins which each have one neighbor in the second or X^{th} sublattices. This is illustrated in Figure 3 below which has three sublattices.

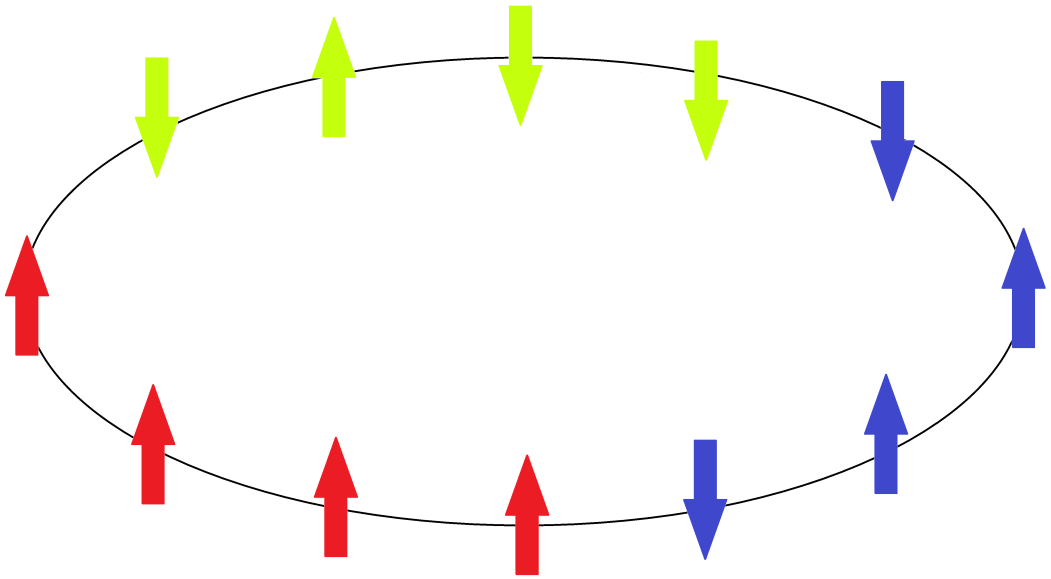

Figure 3: A chain of 12 particles with periodic boundaries. There are three sublattices, represented by red, blue and lime colored arrows respectively. Note how the two central spins in each sublattice only interact with spins in the same sublattice. Meanwhile, the "boundary" spins have a single neighbor of another sublattice.

2.1. Our Approaches

We analyze these systems in a variety of ways. First is the analytic approach, along the lines of Glauber or Racz and Zia [12, 13]. We look at the properties of the one-dimensional graph at hand and mathematically derive results for the kinetic Ising model's behaviors. In addition to broad analyses, we will focus on specific cases which simplify the system at hand and provide broader insight. This kind of limiting case analysis is also seen in the literature in the form of consideration of cases with homogeneous sublattices, the analysis of the thermodynamic limit, or the limiting of the scope to steady state behavior.

A significant part of our analysis comes in the form of numerical simulations. This is an approach which is more prevalent in the more recent work in the area, as computational power and accessibility has grown with improvements in technology. We take the analytically derived systems of differential equations and have a computer program solve them numerically. This allows us to get results which would be far too time consuming to work out by hand, and easily adjust the parameters of the system to see larger patterns within our results. Crucially, the systems of differential equations lead to results of expectation values, or what we would expect to see on average. These expectation values are a useful description of the system even if they are not the most complete one (recall Glauber's argument about the probability functions being far too much information for most practical purposes).

We focus our analysis on the overall system magnetization, which is computed by averaging the individual spin expectation values. This is done for one-dimensional

systems using a Python module we created which can be seen in Appendix A. We have functions for various temperature sublattice organizations and run them with parameters with arbitrary and/or convenient units for the temperature pattern, system size, time scale, initial spin pattern, boundary type, and magnetic field. The number of time steps can be adjusted in the code to fit our needs. This information is used to populate arrays of appropriate values. We then use a ordinary differential equation function from SCIPY, a Python library for scientific computing, to numerically solve for the individual particle expectation values over time. The total lattice or sublattice magnetizations over time are a simple averaging along the appropriate array axis, and and we can find a line of best fit of a given form using a function from the SCIPY library. When looking at a system with a magnetic field, there are complications which we simplify by using a mean field approximation. This is fleshed out in more detail in the magnetic field subsection.

A second method of computationally-assisted analysis is via Monte Carlo simulations. With these we can capture the stochastic nature of the system, as the computer takes a random particle σ_i and flips its spin based on the transition rate $w_i(\sigma_i)$. This allows us to see one possible evolutionary path of the system in full detail, and we can average these results over a large number of trials to piece together results that mirror the expectation value numerical simulations. In this way Monte Carlo simulations are an important alternative method that allow for a sanity check of our other computational work. It is important to note that these stochastic simulations use the transition rates rather than differential equations derived from them, which means that they are free of some approximations we will make for the numerical solutions.

2.2. Alternating Temperatures

We approach the one-dimensional kinetic Ising model with an alternating temperature pattern along the lines of Racz and Zia [13]. For now, we set the magnetic field to zero and employ periodic boundary conditions. Multiply both sides of the probability function system of equations with σ_i and sum over the configuration $\{\sigma\}$, and the result is a system of differential equations for the expectation value $q_i \equiv < \sigma_i >$. This is the system of equations which are solved numerically in the Python module, and each particle in the system has an equation as in Equation 17.

$$\frac{\partial}{\partial t} q_i(t) = -q_i + \frac{\gamma_i}{2}(q_{i-1} + q_{i+1}) \tag{17}$$

To better understand what this means for the overall system, we then assume initial spin homogeneity within sublattices. This means that each spin σ_i in a given sublattice with temperature T_x begins with the same spin value. Therefore, the alternating structure means that all particles within a sublattice x have their spins evolve in an identical manner. We define a sublattice magnetization as $m_x = \frac{X}{N} \sum_{i=1}^{N/X} q_i$ where the sum is restricted to spins within the sublattice, and let the total system

magnetization be $m = \frac{1}{X}\sum_{x=1}^{X} m_x$. The homogeneity of the sublattices means that each sublattice magnetization obeys an equation that looks very similar to the individual spin expectation values:

$$\frac{\partial}{\partial t} m_x(t) = -m_x + \frac{\gamma_x}{2}(m_{x-1} + m_{x+1}) \tag{18}$$

Recall that in the two-temperature case explored by Racz and Zia we have $m_{x-1} \equiv m_{x+1}$, and this allows them to find the individual sublattice magnetizations as a sum of two decaying exponentials. The exponentials' powers are the same, so the total system magnetization can also be expressed in the same way, which we recover in our generalized numerical simulations. This can be seen in Figures 4 - 6 below.

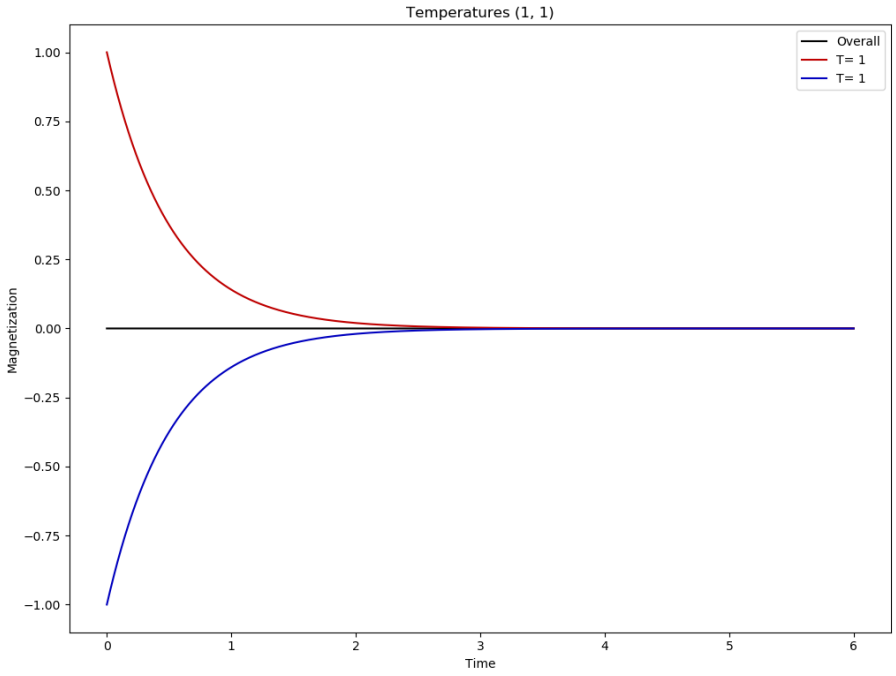

Figure 4: A two-temperature ring's magnetizations graphed versus time with arbitrary units. The red line is the first sublattice's magnetization, the blue line is the second, and the black line is the total system magnetization. We see that equal temperatures leads to perfectly symmetrical dynamics to the steady state magnetization of zero.

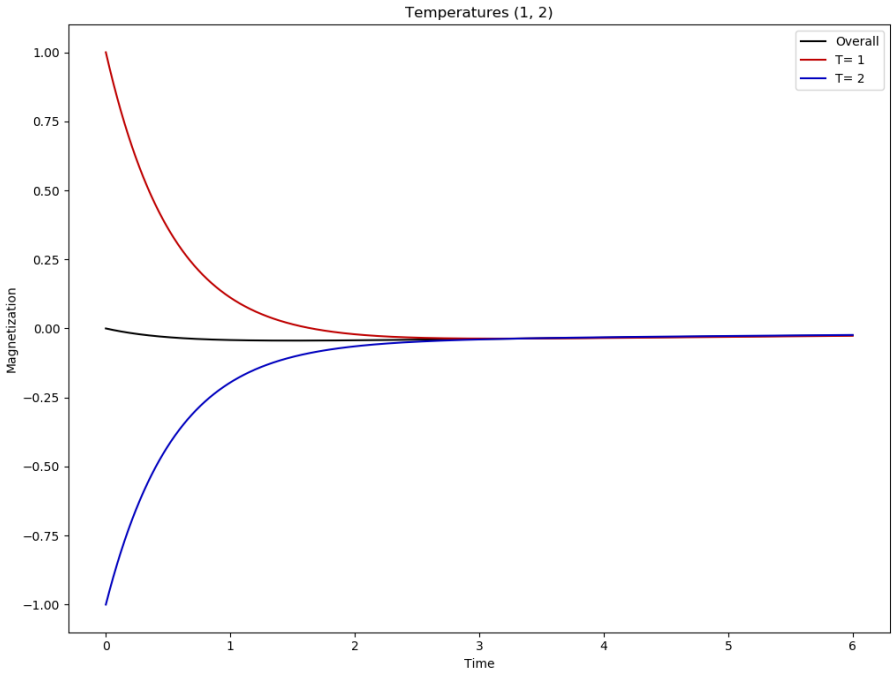

Figure 5: Here we have temperatures (1, 2), and the total system magnetization visibly drops before returning to the equilibrium value of zero. This drop comes from the higher temperature sublattice relaxing more quickly.

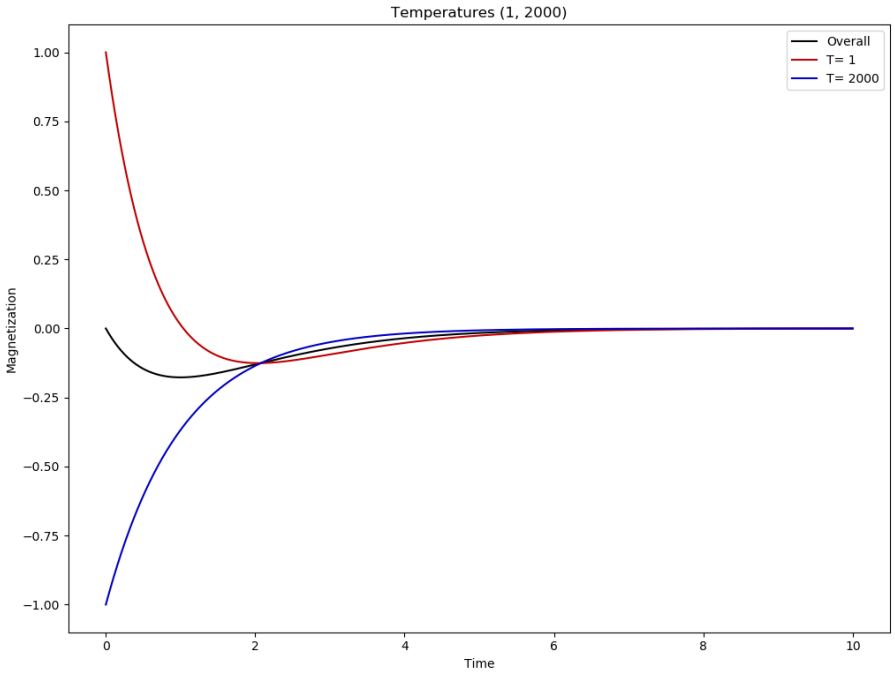

Figure 6: This time the temperatures are (1, 2000). Now the second sublattice relaxes far more quickly, and the first sublattice magnetization drops below zero. The lines cross at around the Time = 2 mark, and the greater temperature gap's effects can be seen in the overall system magnetization moving in a more significant manner.

In each case the red line is the sublattice magnetization with a larger temperature, and the two sublattices start with opposite spin orientations before moving to an evenly split steady state. The higher temperature sublattices relax more quickly towards their non-equilibrium steady state, which is what causes the overall magnetization to temporarily move away from the ultimate equilibrium magnetization of 0. Using curve fitting functions allows us to check and see that we do recover the behavior outlined by Racz and Zia.

In generalizing this system to more than two sublattices, we see that the same sort of sublattice behaviors are maintained. This is due to the structure of the equations staying the same, although the increased complexity does make the explicit form more challenging. The familiar relaxational dynamic can be seen in all multi-temperature systems, which are demonstrated by the plots below.

First we have an alternating system with three sublattices. The odd number of sublattices means we have a total system magnetization which begins from a non-zero value. This is due to the sublattice initial conditions being homogeneous. The time evolution of one such system's magnetizations is shown in Figure 7 below. We again see the higher temperature sublattice relaxing more quickly, and a lower temperature sublattice's magnetization over-correcting.

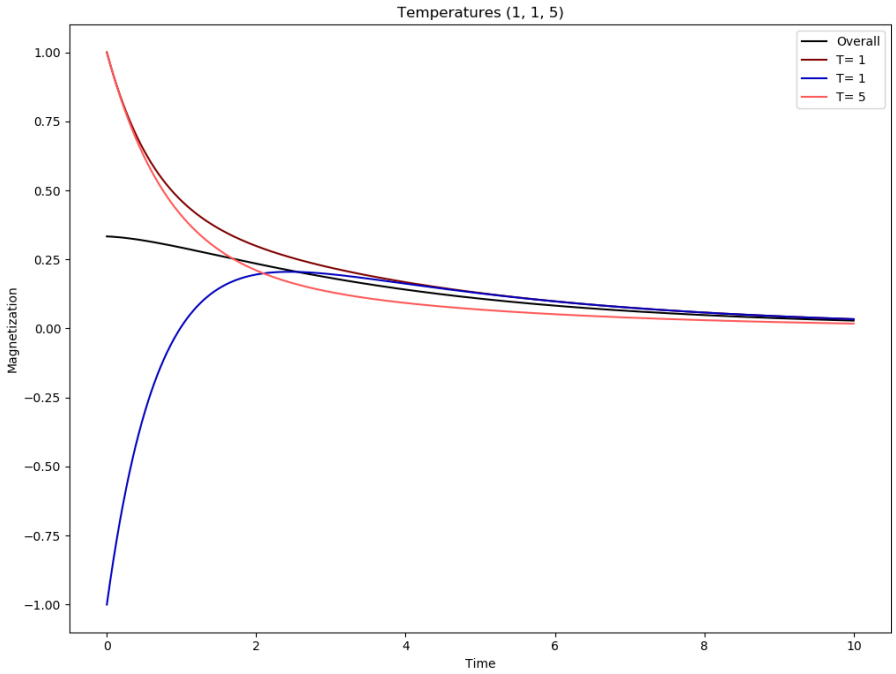

Figure 7: The overall and sublattice magnetizations for temperatures (1, 1, 5). Note the initial conditions of each, as well as their shared steady state magnetization of 0. The overall magnetization starts from $+\frac{1}{3}$.

Next we have an alternating system with four sublattices. Our choice of initial conditions means that the system magnetization starts at zero. We demonstrate how temperatures of different orders of magnitude behave by having temperatures of $(1, 10, 100, 1000)$. The temperatures influence the magnetization equations via $\gamma_x = \tanh(\frac{2}{T_x})$, so it follows that we see greater differences between 1 and 100 than 10 and 1000. The particular example of Figure 8 below highlights this by showing a greater gap in the red colored sublattice magnetizations than the blue colored ones. We also observe how the $T = 10$ sublattice relaxes more quickly than the $T = 100$ sublattice despite having a lower temperature due to the particular temperature ordering.

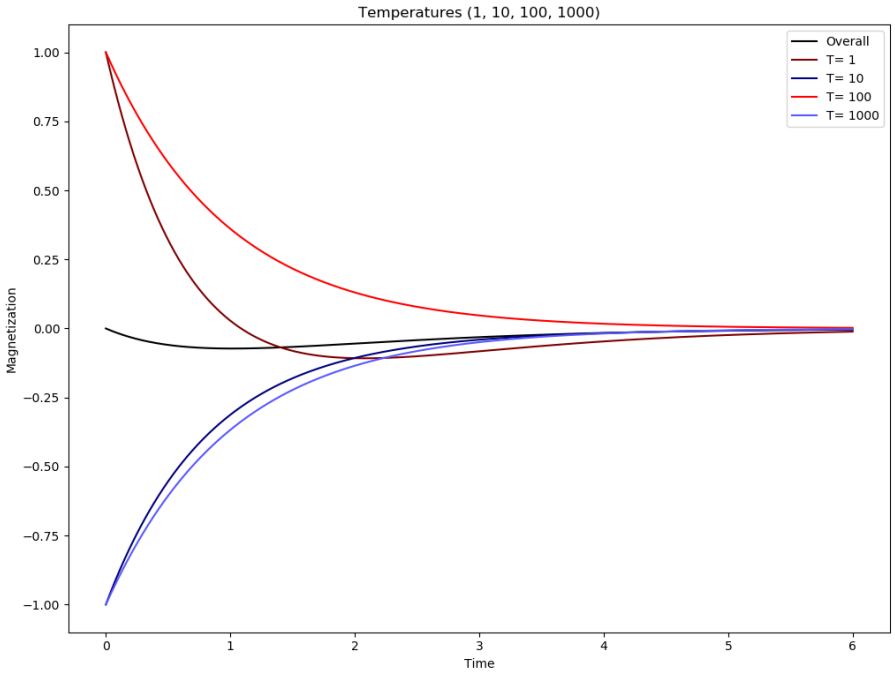

Figure 8: Here we have temperatures (1, 10, 1000, 1000). The black line shows the total system magnetization dipping slightly below zero before the relaxational dynamics dominate.

Finally, we give an example with five temperatures. This could continue indefinitely, and the translational symmetries of this specific system setup mean the expectation values can be computed with minimal computational cost. However, what we see is merely a continuation of previously described behaviors, and increasingly complex temperature patterns within our constraints do not appear to give more insight into multi-temperature kinetic Ising models.

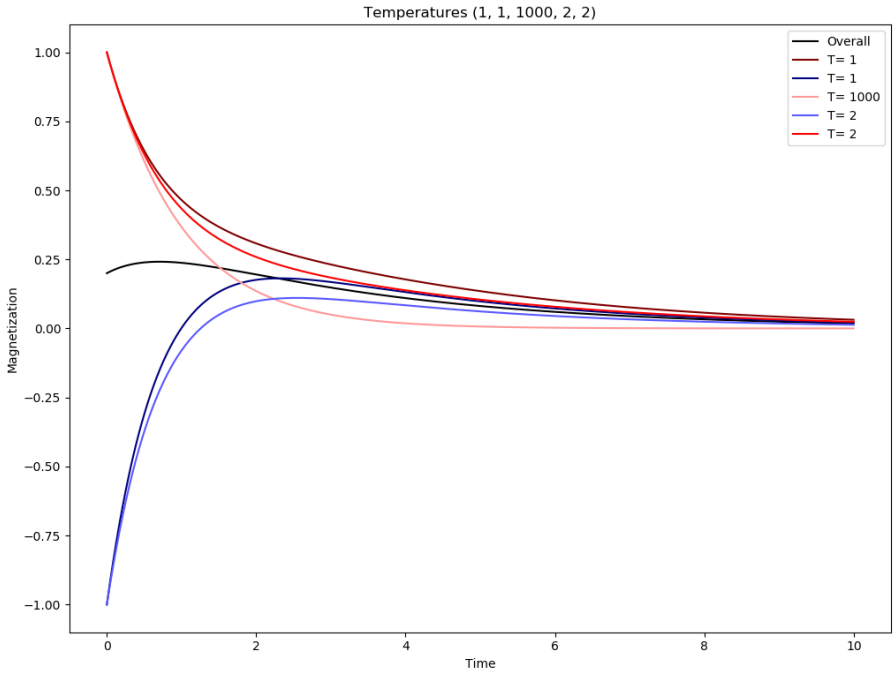

Figure 9: The temperature pattern is (1, 1, 1000, 2, 2), and we see the light red line relax significantly faster than the lower temperature sublattices. The overall magnetization starts from $+\frac{1}{5}$.

In all these cases the sublattices each relax to a magnetization of zero. This behavior is predictable from the form of the system of equations $\frac{\partial}{\partial t}m_x(t) = -m_x + \frac{\gamma_x}{2}(m_{x-1} + m_{x+1})$. The "$-m_x$" means that the sublattices each tend towards zero magnetization. The remaining terms are what cause the interesting short term dynamics, but merely reinforce the total magnetization approaching zero for the steady state system. This is therefore a completely general behavior for kinetic Ising models with no magnetic field, and the particulars of the magnetic field influence on the non equilibrium steady state are expanded upon in a later section.

The above work has been using periodic boundary conditions, which introduce translational symmetry and make the relevant system behaviors a matter of how many sublattices there are and not dependent on the system size N. However, it is also possible to have a one-dimensional kinetic Ising model with hard boundaries, which we may visualize as a linear chain of spins rather than a ring:

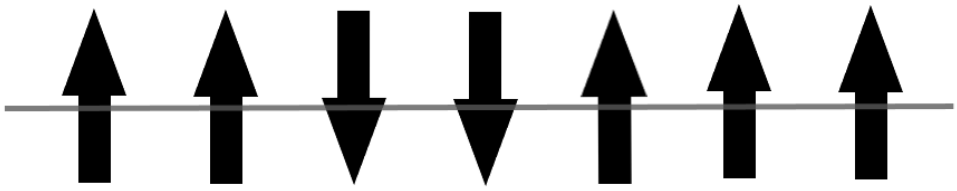

Figure 10: A chain of particles with hard boundary conditions, showing seven particles from an arbitrary part that does not include the ends.

This system is governed by the same dynamics, with a slight alternation for the end points. There are several ways this can be done, such as simply removing one of the terms for the end point equations, doubling the influence of the penultimate particles on the ends, or letting the ends also interact with a fixed spin to anchor them. We will choose to focus on the second case, but each of these is itself a reasonable system to study. In this system, the single particle expectation values are governed by the system of equations given below where i is an integer $1 < i < N$.

$$\begin{aligned} \frac{d}{dt}q_1 &= -q_1 + \gamma_1\ q_2 \\ \frac{d}{dt}q_i &= -q_i + \frac{\gamma_i}{2}(q_{i-1} + q_{i+1}) \\ \frac{d}{dt}q_N &= -q_N + \gamma_N\ q_{N-1} \end{aligned} \tag{19}$$

The loss of translational invariance also introduces another parameter of interest: our system size. The end points σ_1 and σ_N introduce new dynamics, which spread through the system with each time step. It is therefore reasonable to expect both system size effects as well as more stable behavior as the system approaches the thermodynamic limit of $N \to \infty$. This is what we see in the example systems whose overall magnetizations are plotted in Figure 11 below.

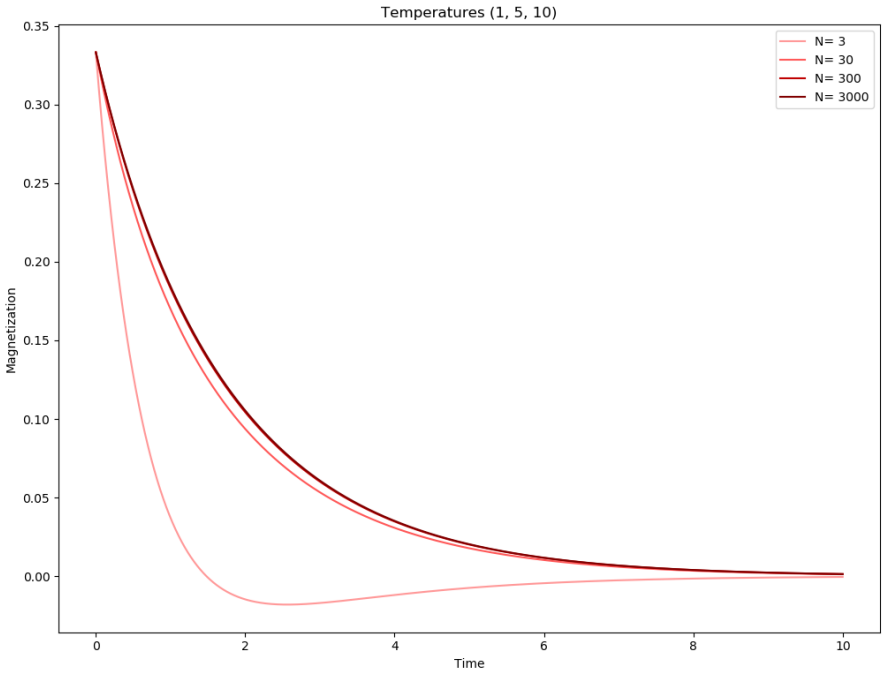

Figure 11: Plotting the overall system magnetization for temperatures (1, 5, 10) with hard boundaries. The systems have sizes 3, 30, 300, and 3000, with the smallest systems relaxing most quickly. Note how the two largest systems are almost indistinguishable, effectively following the same line.

Where is this significant change in overall magnetization coming from? We find that the end points do not change much with the system size. This is shown in Figure 12 below by looking at the two end points of the same systems as in Figure 11.

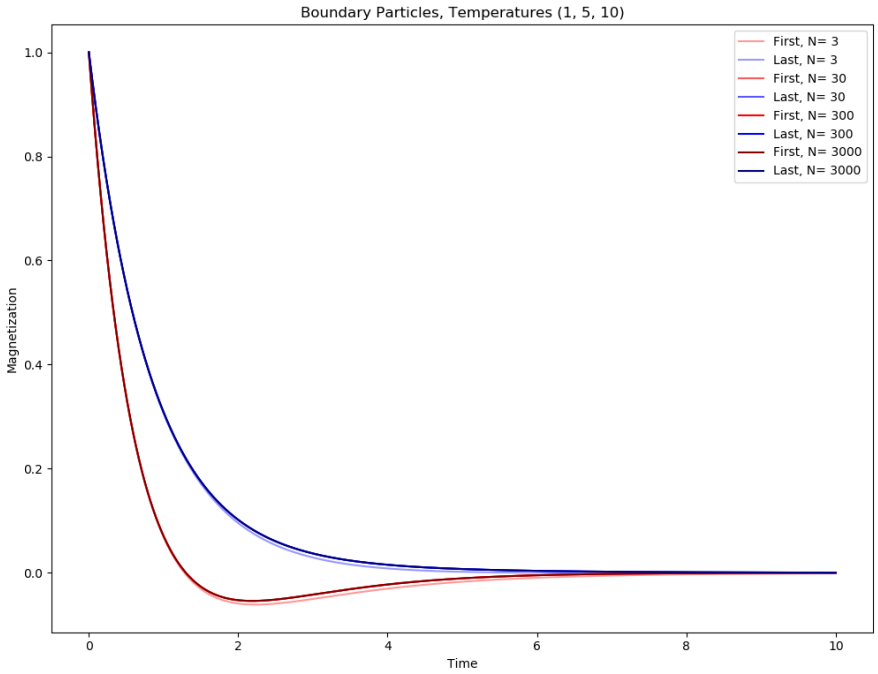

Figure 12: Plotting the boundary spin expectation values for temperatures (1, 5, 10) with hard boundaries. The dark red lines are the first sublattice end boundary while the blue lines are for the end of the third sublattice. The systems have sizes 3, 30, 300, and 3000, although only the most lightly colored size 3 lines are barely distinguishable.

With the individual end points acting nearly independently of the system size, we can view the system size effects as a result of more particles within each sublattice diluting the novel behaviors at the hard boundaries. As the number of sublattice repetitions grows, we see more and more of the familiar periodic boundary behaviors. We see that the systems with hard boundaries are indistinguishable at a large scale from systems with periodic boundaries, as evidenced by Figure 13 below.

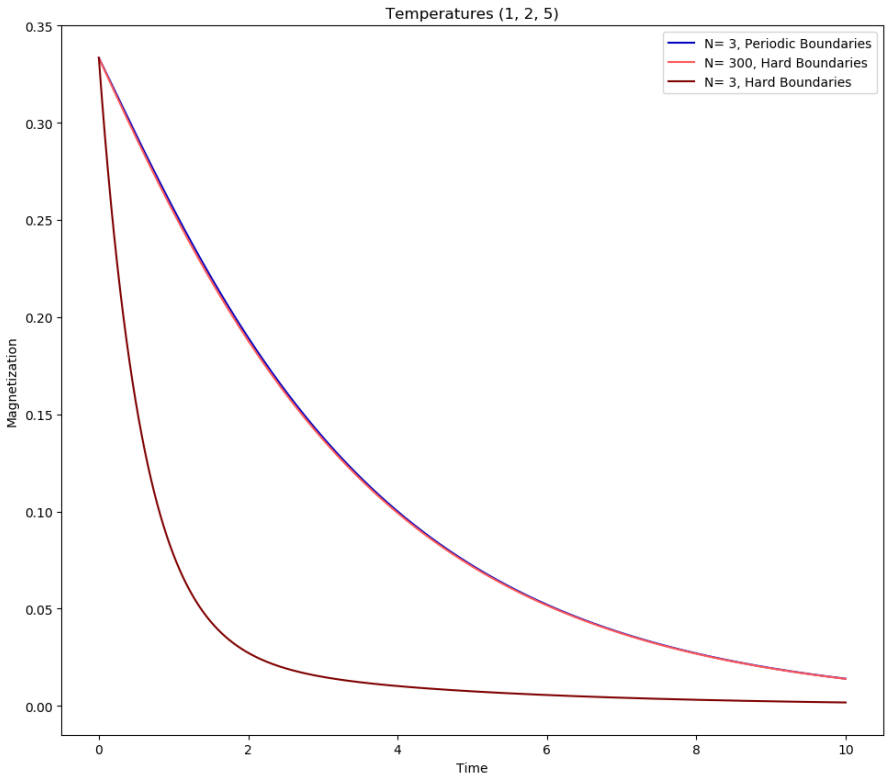

Figure 13: Overall magnetization plots for a system with temperatures (1, 2, 5). A hard boundary system with 3 spins is clearly distinct from a hard boundary system with 300 particles and a periodic boundary system with 3 particles, while the latter two are nearly indistinguishable and from the darker line.

Finally, we must note the introduction of hard boundaries introduces additional complexity with regard to the sublattice temperature ordering. We noted early on that for $N \leq 4$ we have some temperatures which do not directly touch one another, meaning that (T_1, T_2, T_3, T_4) corresponds to a different system than (T_1, T_2, T_4, T_3). This is shown explicitly below in Figure 14.

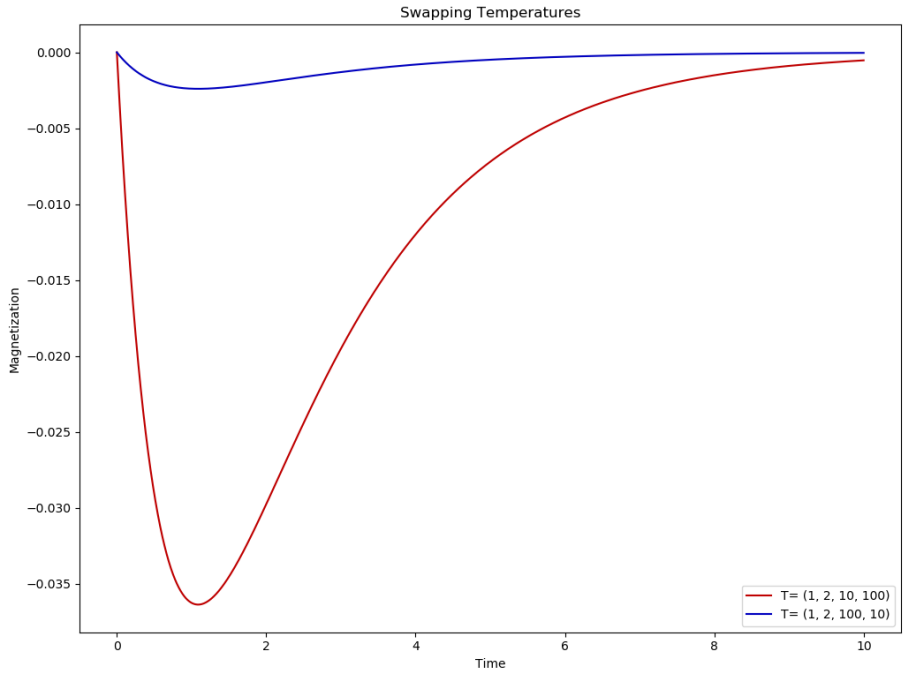

Figure 14: System magnetization for systems with temperatures of (1, 2, 10, 100) and (1, 2, 100, 10) and periodic boundaries. An example of how swapping temperatures changes the system even with periodic boundaries.

However, the temperature pattern (T_1, T_2, T_3, T_4) is equivalent to (T_4, T_1, T_2, T_3) with periodic boundaries. This is no longer the case with hard boundaries, as demonstrated in Figure 15 below.

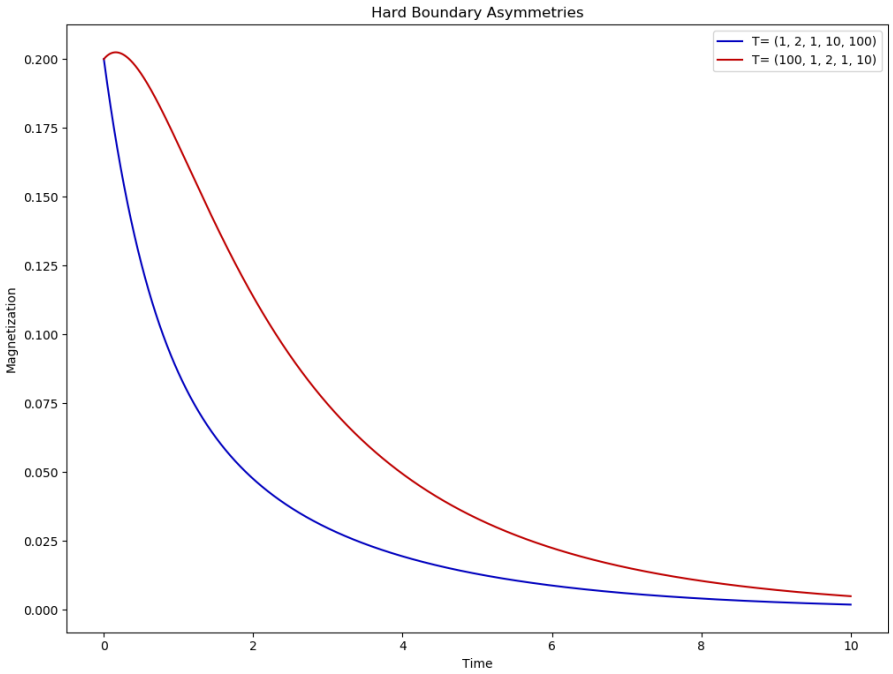

Figure 15: System magnetization for systems with temperatures of (1, 2, 1, 10, 100) and (100, 1, 2, 1, 10) and hard boundaries. The system size is 500 to reduce finite size effects. An example of how cycling through temperatures changes the system given hard boundaries.

2.3. Blocks of Temperatures

Now let us consider our other sublattice pattern, the blocks of temperatures. The figure below is shown again to illustrate how the sublattices are organized. Unlike the alternating pattern with periodic boundaries, it is clear that the number of particles within each sublattice is relevant for its dynamics.

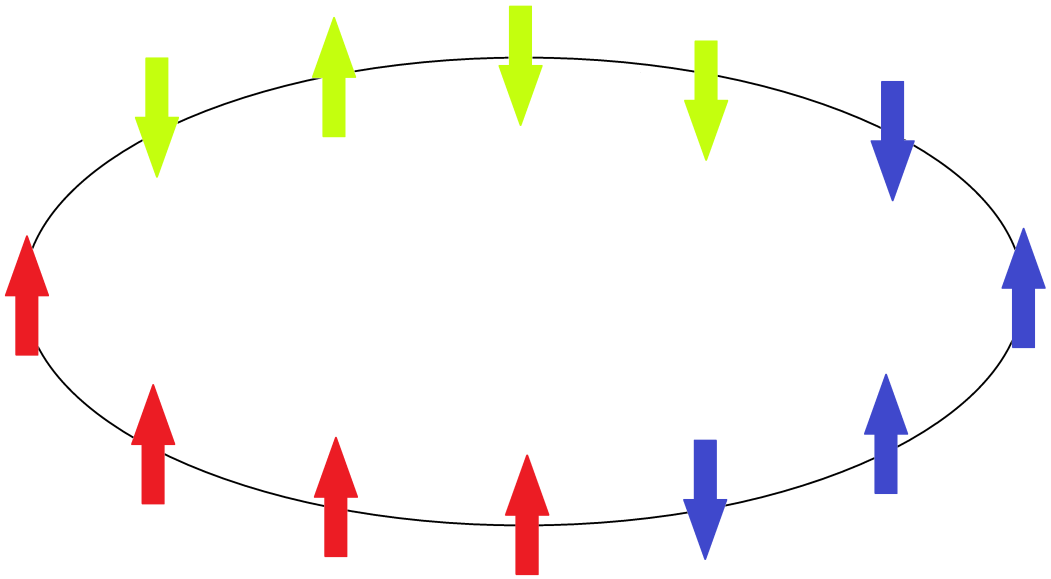

Figure 16: A chain of 12 particles with periodic boundaries. There are three sublattices, represented by red, blue and lime colored arrows respectively. Note how the two central spins in each sublattice only interact with spins in the same sublattice. Meanwhile, the "boundary" spins have a single neighbor of another sublattice.

For the specific case where the sublattices each have one particle, we have the exact same system as one with an alternating temperature pattern. This is trivially shown by looking at the system of equations by each particle. But as the block size (defined as the system size over the number of sublattices N/X) increases, the system exhibits some different behaviors.

It is important to note that beyond this trivial limit of $N/X = 1$ it can be difficult to make comparisons between the systems. This is because the specific time evolution of magnetizations are highly dependent on initial conditions, which are hard to align between the two sublattice patterns. The most natural initial conditions—which we adopt in thesis—are those which have each sublattice's spins starting in the same position. This means that the ferromagnetic dynamics of the kinetic Ising model cause the time evolution of magnetizations to generally be slower for temperature blocks because most pairs of neighbors start in the same orientation and resist the general relaxational dynamics of the system.

We can see the extreme influence of system size of sublattice and overall magnetization by looking at several very simple systems (with periodic boundaries). First, we consider the $N/X = 1$ case which is identical to the alternating sublattice pattern. The magnetizations are plotted in Figure 17 below.

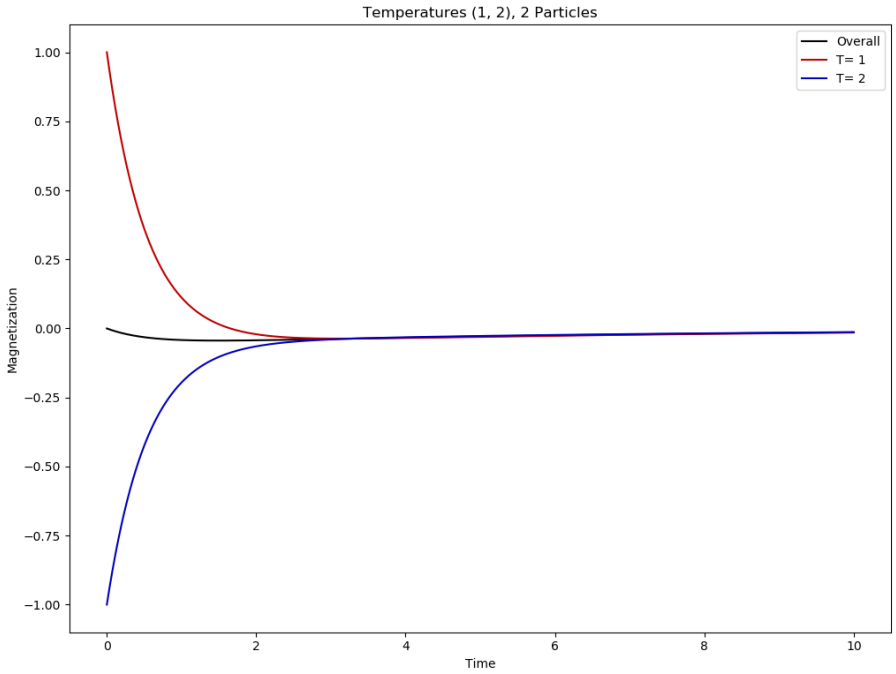

Figure 17: The sublattice and system magnetizations for a system with periodic boundaries, size $N = 2$, and temperatures (1, 2) plotted in red and blue respectively. In this case we have an identical plot to what an alternating temperature pattern would produce.

Now we grow the blocks by a single spin each, to $N/X = 2$. Note how the overall magnetization in Figure 18 is completely flat.

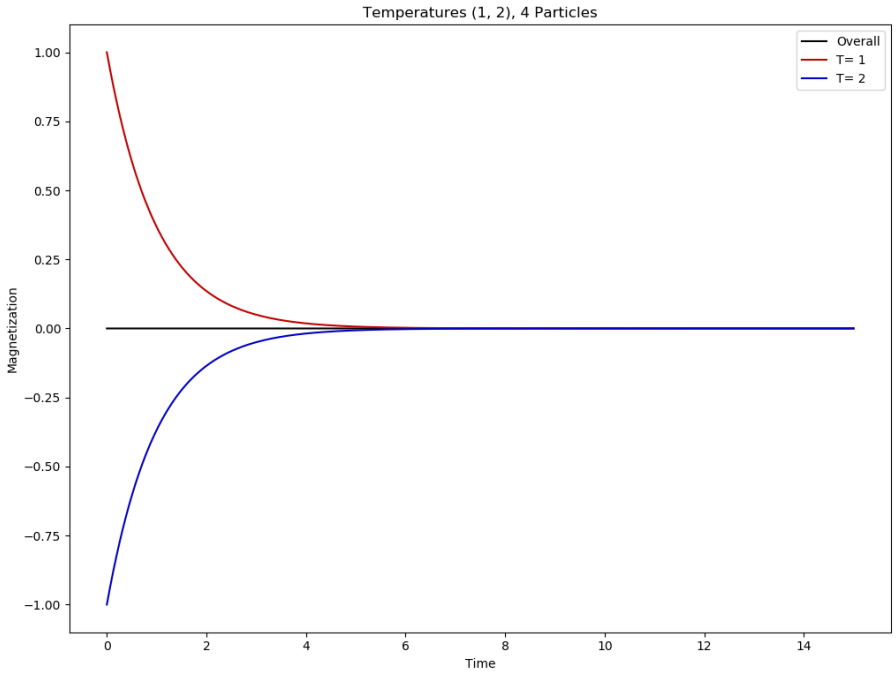

Figure 18: The sublattice and system magnetizations for a system with periodic boundaries, size $N = 4$, and temperatures (1, 2). The overall magnetization is exactly zero at all times, indicating that the two sublattice magnetizations mirror one another.

Next, observe the same system setup with $N/X = 3$. The overall magnetization shown in Figure 19 now moves in the opposite direction from zero before relaxing to the non-equilibrium steady state.

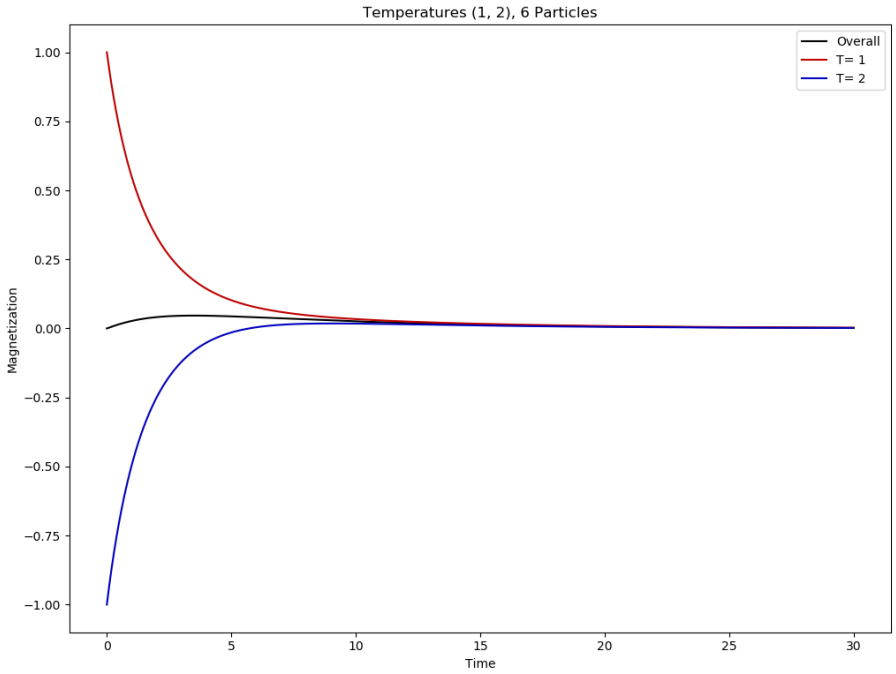

Figure 19: The sublattice and system magnetizations for a system with periodic boundaries, size $N = 6$, and temperatures $(1, 2)$. Note how the total system magnetization now goes positive before relaxing to zero.

Just adding a few spins to our system has caused it to act in a very novel manner. This appears to be the only point of transition, as approaching the thermodynamic limit simply magnifies the present behaviors, shown by Figure 20 below using a large system of $N/X = 5000$.

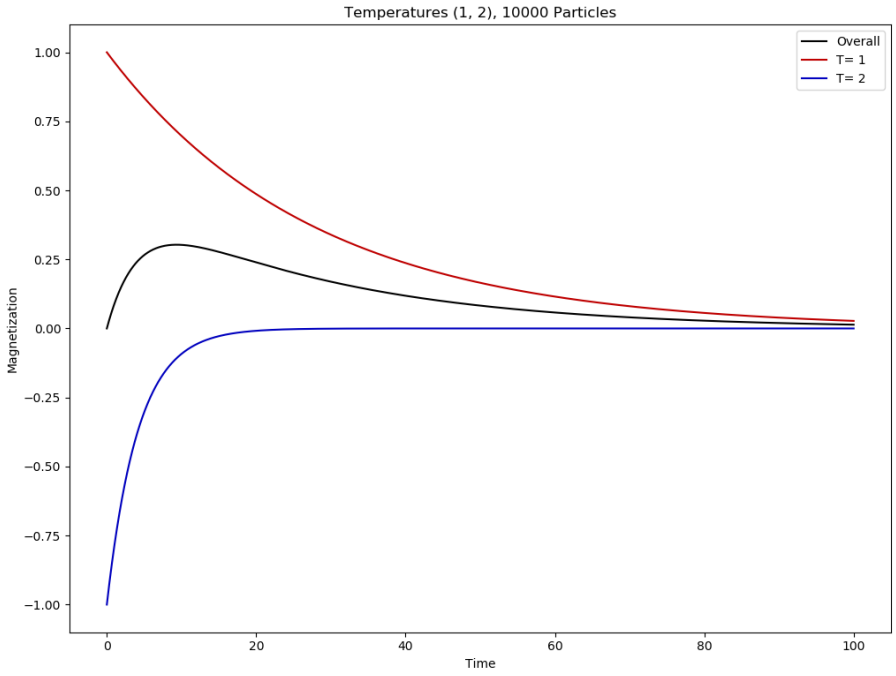

Figure 20: The sublattice and system magnetizations for a system with periodic boundaries, size $N = 10,000$ and temperatures $(1,\ 2)$. The overall magnetization's movement is now significant, and the timescale of the dynamics of the system reaching a steady state is much longer.

This is very interesting dynamical behavior which we can explore by playing with the expectation value equations. It is important to note that because it is time-dependent relaxational behavior rather than steady state behavior of the magnetization, we know it is strongly influenced by our method of setting initial conditions.

For simplicity, in the following analysis we will assume $X = 2$, periodic boundary conditions, and initial states of opposite spins for each sublattice. First we start with the general single particle expectation value equation.

$$\frac{\partial}{\partial t}q_i(t) = -q_i + \frac{\gamma_i}{2}(q_{i-1} + q_{i+1}) \tag{20}$$

Consider what happens with the $N = 2$ case where our system is $\{\sigma_1, \sigma_2\}$. We consider the equation for σ_1, noting that switching the subscripts will produce the equation for the other particle.

$$\frac{\partial}{\partial t}q_1(t) = -q_1 + \gamma_1 q_2 \tag{21}$$

The $-q_1$ is driving the relaxation to the steady state magnetization, but the $\gamma_1 q_2$ is also accelerating this process because the initial magnetizations of the two particles

have opposite signs. This comes about from each particle only neighboring particles with initial spin positions opposite of its own. Recall that the exact analytical solutions independent of initial conditions for this specific system can be found in Racz and Zia.

Now we look at the $N = 4$ case, where Equation 22 below describes the changes in σ_1's expectation value.

$$\frac{\partial}{\partial t} q_1(t) = -q_1 + \frac{\gamma_1}{2}(q_4 + q_2) \tag{22}$$

By symmetry, each particle's equation has a sum of one particle in its sublattice and one particle from the other sublattice; in this case σ_2 is in the same block while σ_4 is in the other. We call such particles edge particles for its block, and systems with $N/X = 2$ consist entirely of edge particles. For all such systems with an even number of sublattices (i.e. $X \bmod 2 \equiv 0$) we have an initial condition of each particle's neighbors canceling one another out. This happening throughout the lattice means that all the spins evolve with the same exponential decay $C\mathrm{e}^{-t}$ where C is the initial spin value, resulting in the completely flat overall magnetization.

On the other hand, the thermodynamic limit sees $N/X \to \infty$ which means that essentially every particle will have two neighbors of the same sublattice. So we can think of the ferromagnetic interactions between them as resisting the relaxation. This intuition can be confirmed by looking at sublattice magnetizations which are found by summing over each particle in the sublattice x.

$$\sum_x \frac{\partial}{\partial t} q_i(t) = \sum_x -q_i + \sum_x \frac{\gamma_i}{2}(q_{i-1} + q_{i+1}) \tag{23}$$

In the thermodynamic limit we neglect the effect of the edge particles for an approximation of the sublattice magnetization. In doing so we allow ourselves to say $q_{i-1} = q_i = q_{i+1}$. This approximation results in Equation 24 below.

$$\sum_x \frac{\partial}{\partial t} q_i(t) = \sum_x -q_i + \sum_x \gamma_i q_i \tag{24}$$

We then find an expression for the sublattice magnetization m_x. Recall that particles in the same sublattice x have the same temperature bath with T_x and therefore the same γ_x.

$$\begin{aligned}
\sum_x \frac{\partial}{\partial t} q_i(t) &= \sum_x -q_i + \sum_x \gamma_i q_i \\
\frac{\partial}{\partial t} m_x(t) &= -m_x + \gamma_x m_x \\
\frac{\partial}{\partial t} m_x(t) &= (\gamma_x - 1) m_x \\
m_x(t) &= C\ \mathrm{e}^{(\gamma_x - 1)t}
\end{aligned} \tag{25}$$

This approximate behavior is supported by numerical simulations and curve fitting to the sublattice magnetizations. In Figure 21 below we have a system of size $N = 200$, with lime green lines showing the curve fitting on each sublattice. Note how the best fit for one sublattice appears to start slightly below its true initial state, but otherwise the exponential decay fit is close enough that it obscures the original sublattice magnetization lines.

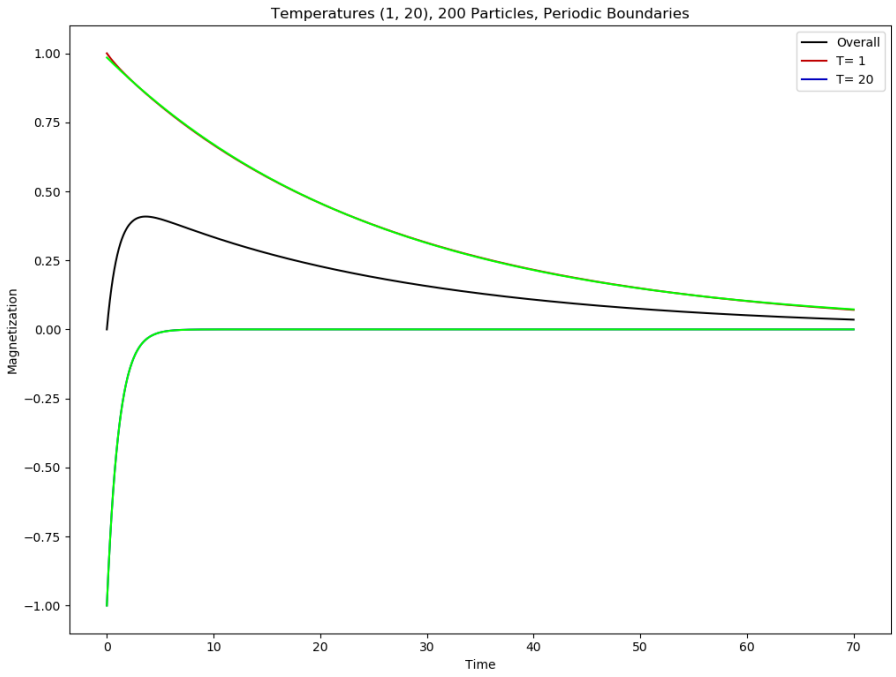

Figure 21: The sublattice and system magnetizations for a system with periodic boundaries, size $N = 200$ and temperatures $(1, 20)$. The lime green lines are the numerical lines of best fit for each sublattice magnetization, taking the form of a simple exponential decay.

However, our single exponential fit runs into more trouble with smaller systems. For example we can see the same plot but for $N = 20$ in Figure 22 below, and note that the curve fit is slightly worse.

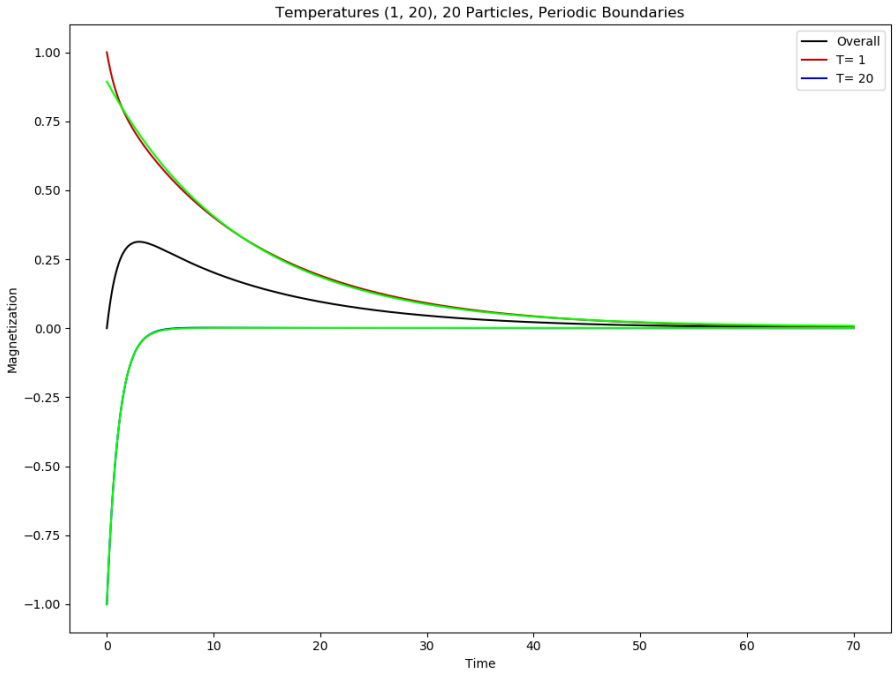

Figure 22: The sublattice and system magnetizations for a system with periodic boundaries, size $N = 20$ and temperatures $(1, 20)$. The lime green lines are the numerical lines of best fit for each sublattice magnetization, taking the form of a simple exponential decay. Note how this equation is unable to fully capture the time-dependent dynamics of the sublattices.

Importantly, this sublattice behavior is maintained with the introduction of additional sublattices. This can be seen in the following Figures 23 - 25 where $X = 3$. Recall again that an odd number of sublattices means the initial overall magnetization is nonzero due to sublattice homogeneity at $t = 0$ and ensuring that each spin begins with spin ± 1.

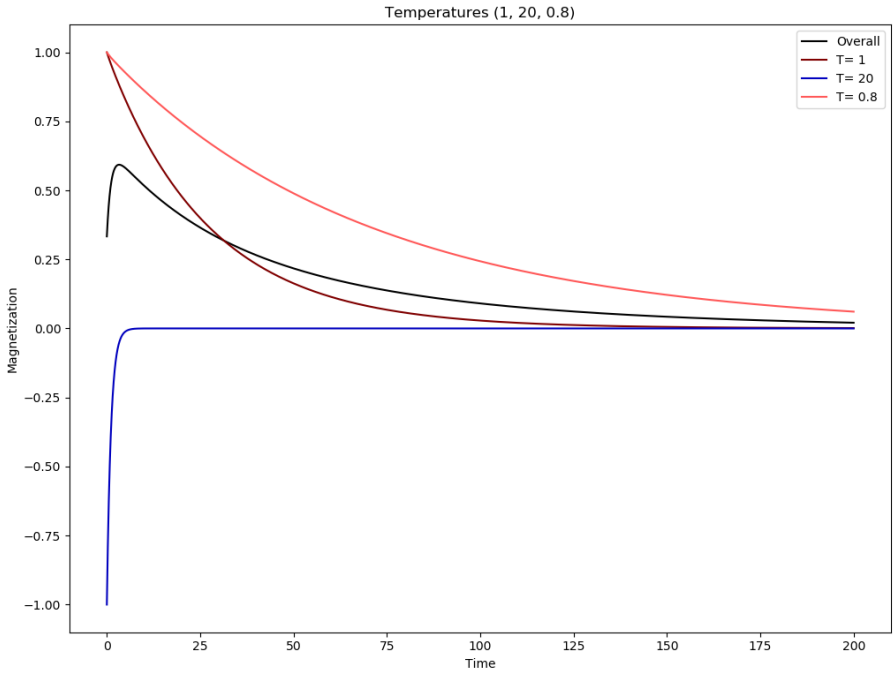

Figure 23: The sublattice magnetizations for a system with $N = 450$ and temperatures (1, 20, 0.8). The overall magnetization starts at $\frac{1}{3}$ and is plotted in black.

We can order the temperatures in such a way that the overall magnetization switches signs as in Figure 24 below. This dynamic behavior is something which the alternating temperature structure does not produce. The key here is that the second sublattice relaxes much more slowly than the others so that it dominates the overall magnetization, and this is not possible without the large blocks which insulate particles from the effects of spins in the opposite direction.

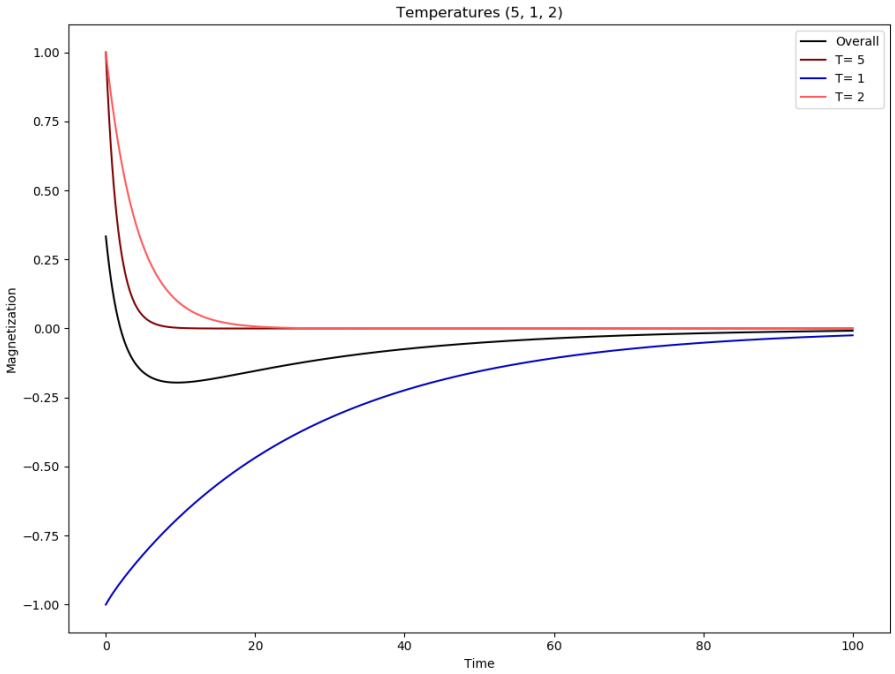

Figure 24: The sublattice magnetizations for a system with $N = 450$ and temperatures (5, 1, 2). The overall magnetization starts at $\frac{1}{3}$.

This insulating effect of the larger blocks also means that we can reorder the sublattice temperatures to get more complicated effects on the overall magnetization. Now the two sublattices which start with positive spins have vastly different relaxation rates, which paired with the negative spins relaxing sometime in-between creates two local extrema in the overall magnetization.

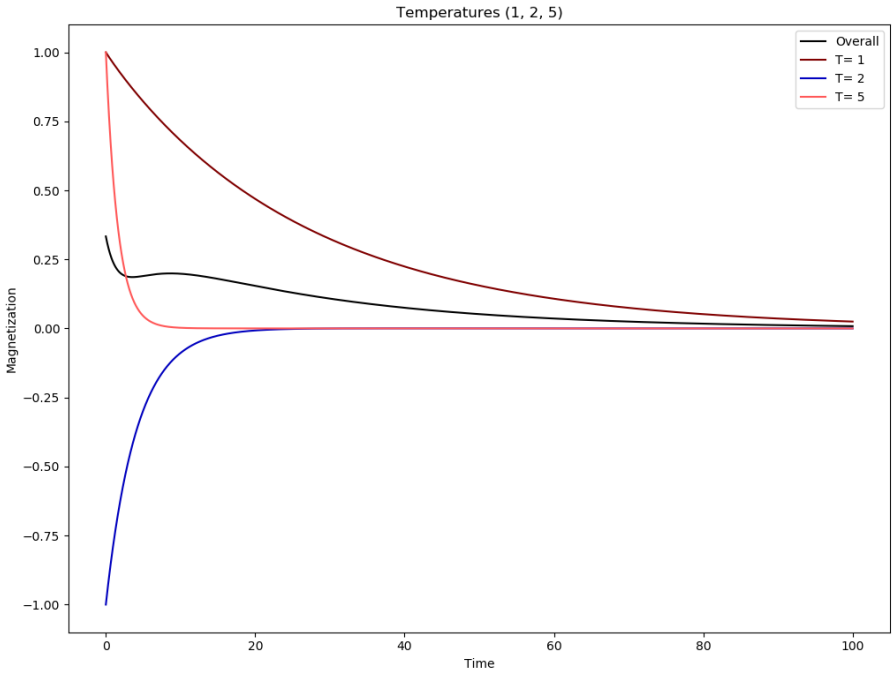

Figure 25: The sublattice magnetizations for a system with $N = 450$ and temperatures (1, 2, 5). The overall magnetization starts at $\frac{1}{3}$ and exhibits interesting dynamics with local magnetization extrema.

These local extrema are a predictable phenomenon in the thermodynamic limit, and we see in Figure 26 below that they are clearly impacted by the block size N/X. See the plot of the overall magnetizations for the system above where the block sizes are adjusted from 150 to 100, 50, 20, and 10 particles.

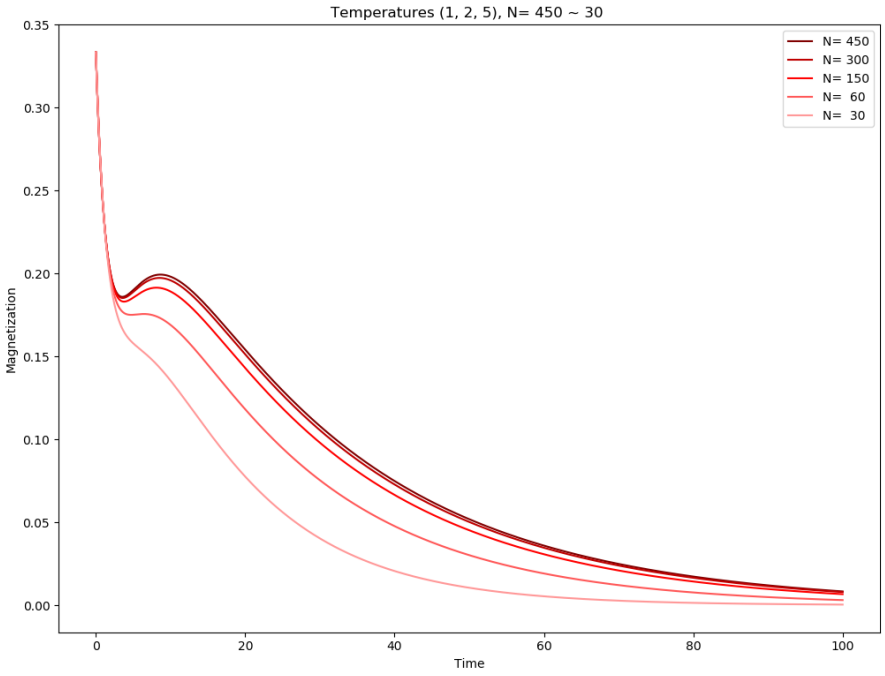

Figure 26: The overall magnetizations for systems with temperatures (1, 2, 5) are plotted for system sizes of 450, 300, 150, 60, and 30 particles. Note how the 30 particle system does not have local extrema and the novel dynamic behavior grows as the system grows.

In this manner the multi-temperature generalization of temperature blocks allows for fascinating new dynamics. Fortunately, they can be qualitatively understood and predicted via the conception of each temperature block's magnetization relaxing with a basic exponential decay, the rate of which can be approximated by working with the thermodynamic limit. This does mean that we no longer see a sublattice magnetization flipping signs as in the alternating temperature case, again with the caveat that very small systems can display his behavior to a much more limited degree.

Finally, we should note the effect of hard boundaries on the temperature block system. Given our analysis of the system with periodic boundaries and the impact of hard boundaries on the alternating temperature structure, it is natural to think that the boundary type is only important for smaller systems. This is in fact what we find, with the hard boundaries acting to reinforce the ferromagnetic interactions within sublattices.

For smaller systems, the effects can be notable. Recall the two block system in Figure 27 we previously saw in Figure 22 to note how the simple exponential decay fit is imprecise for smaller systems.

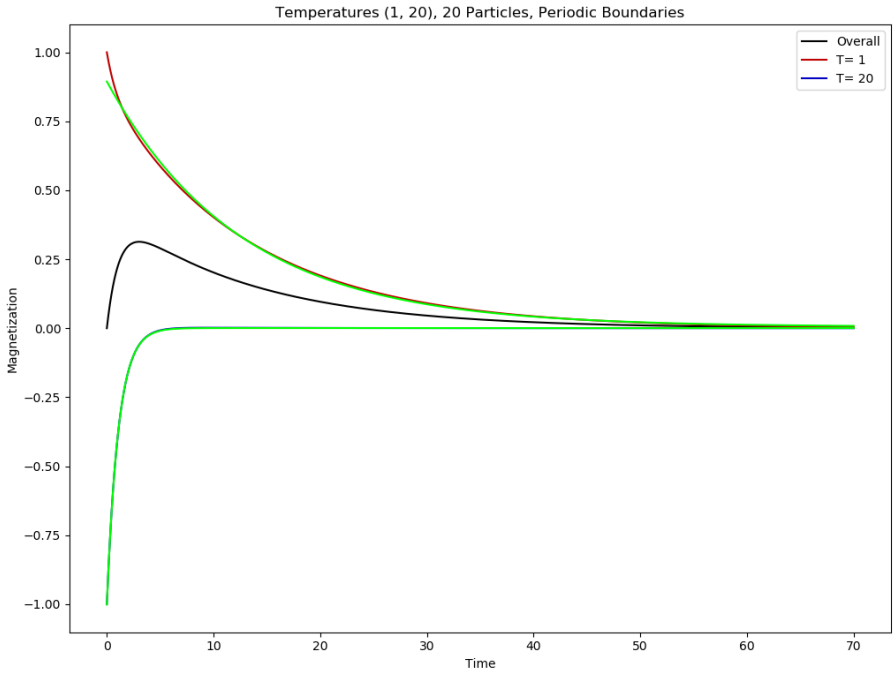

Figure 27: The sublattice and system magnetizations for a system with periodic boundaries, size $N = 20$ and temperatures $(1, 20)$. The lime green lines are the numerical lines of best fit for each sublattice magnetization, taking the form of a simple exponential decay.

Now look at the magnetizations of the nearly identical system but with hard boundaries in Figure 28 below. They may not be very different by inspection, but their deviation from the lime green line of best fit using a single exponential decay shows their minor differences. There are half as many boundaries between sublattices, and the deviation from the exponential powers in the thermodynamic limit is also roughly halved. This is the extent of the hard boundary effect with 20 particles, and the effects are minor in comparison to the small system size effects which are necessarily present for the boundary effect to be noticeable.

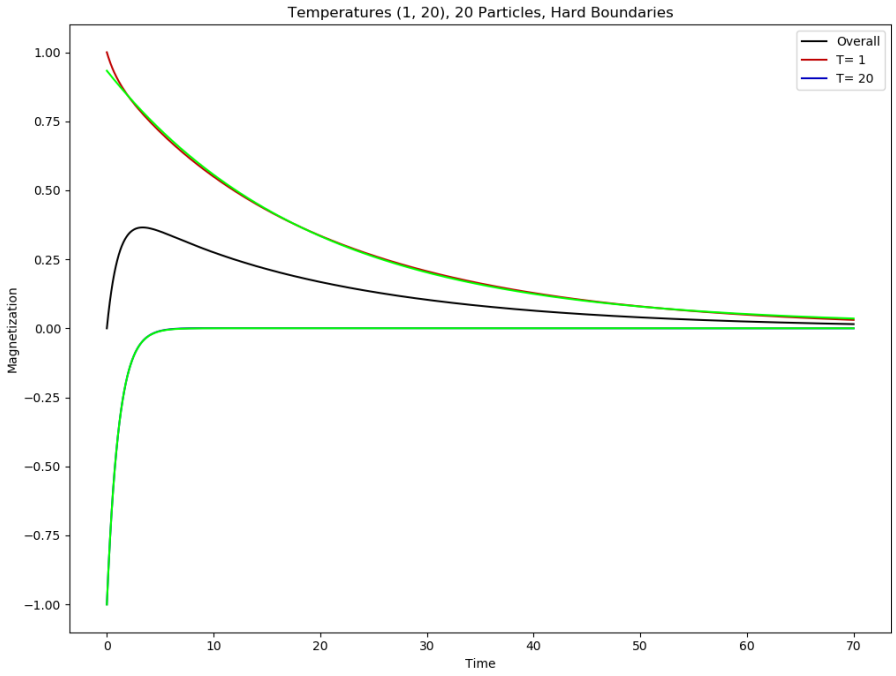

Figure 28: The sublattice and system magnetizations for a system with hard boundaries, size $N = 20$ and temperatures (1, 20). The lime green lines are the numerical lines of best fit for each sublattice magnetization, taking the form of a simple exponential decay.

2.4. Physical Analogies

We have now looked at the two basic temperature bath structures of interest for our one-dimensional kinetic Ising model. In doing so, we have consistently found that relaxation rates can be modeled with exponential functions. This was something which was shown analytically for the two-temperature and alternating temperature case by Racz and Zia, and which we showed above in the case of the block structure in the thermodynamic limit. Even moving beyond these specific limiting cases we find that numerical curve fitting is capable of very closely matching the resulting collective magnetizations. Sums of exponentials have a wide range of physical applications [18], and we will touch on a couple here that are particularly interesting to think about.

Through this pattern one might recognize a surprising analogy to the damped harmonic oscillator. This is a simple system which is introduced to students in undergraduate physics. We have a mass attached to a spring displaced from equilibrium, and a linear damping force bringing the block to rest after some dynamic movement. Solutions to this model take the form of x below, where m is the block mass, k is the spring constant, and c is the damping factor.

$$x = e^{\lambda t}$$
$$\lambda = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m} \tag{26}$$

The solutions to the damped harmonic oscillator are split into three categories based on the sign of $c^2 - 4mk$. If it is positive we have over-damping, if it is negative we have under-damping, and if it is exactly zero we have critical damping. The analogy to the kinetic Ising model would come from the over-damped oscillator, as we do not have oscillatory behavior in the magnetization as it relaxes to zero. In particular, the temperature block structure could be thought of as a collection of multiple over-damped oscillators, insofar the form of the mathematical descriptions is considered.

Exponential decay is also used in a simple model of heat transfer, as two bodies in contact approach a shared equilibrium temperature. Despite some shared characteristics, the actual physics at play is very different from our multi-temperature kinetic Ising model. In our model the temperature baths exchange energy but are held at a constant temperature, with the exponential decay being in the spin magnetizations. The similarity between the mathematical equations contrasted with the distinct physical systems at hand makes this an interesting comparison.

Another potential connection is the stretch exponential relaxation, which arises in systems like molecular and electronic glasses [19]. The stretch exponential takes the form $e^{-(\lambda t)^\beta}$ where β is some stretch factor, and can arise from a continuous sum of exponential decays [20]. The connection here comes from how the overall magnetization is a sum of the sublattice magnetizations, but can often be numerically approximated by a sum of exponential decays which has fewer terms. An example of this is how a system with three temperature blocks has sublattice magnetizations which are each exponential decays, but the overall magnetization can often be given as a sum of two exponentials which each have different relaxation coefficients. This behavior is something potentially worth exploring in future work, as further simplifications to modeling the dynamics of our system could be a source of significant insight.

2.5. Constant Magnetic Fields

Throughout our analysis thus far, we have consistently seen magnetizations which relax to a steady state of zero. We have therefore focused much of our analysis on the time-dependent relaxational behavior of the magnetizations. However, this is because we have only dealt with systems without a magnetic field.

The introduction of this magnetic field adds some additional complexity to our master equation, shown below. The change comes in the form of a multiplicative factor on the transition rate using $\beta_i = \tanh(\frac{H}{T_i})$ where H is the magnetic field's strength.

$$w_i(\sigma_i) = \frac{1}{2}(1 - \frac{1}{2}\gamma_i \sigma_i(\sigma_{i-1} + \sigma_{i+1})) \cdot (1 - \beta_i \sigma_i) \tag{27}$$

The preceding sections have used $H = 0 \rightarrow \beta = 0$ to simplify the derivation for the single particle expectation values. In Equation 28 we find that the expectation value equations in full generality have two additional terms and include the same time two particle spin correlations $r_{i,j}(t) \equiv < \sigma_i(t)\sigma_j(t) >$.

$$\frac{\partial}{\partial t}q_i(t) = -q_i + \frac{\gamma_i}{2}(q_{i-1} + q_{i+1}) + \beta_i - \frac{\beta_i\gamma_i}{2}(r_{i-1,i} + r_{i,i+1}) \tag{28}$$

Unfortunately we do not have analytical expressions for these correlation terms in full generality. Glauber was able to find expressions for them due to the uniform temperatures throughout the system. Racz and Zia were able to find expressions for them using the large amount of symmetry within the alternating two-temperature structure with periodic boundaries. These approaches give an idea of how one might approach a multi-temperature kinetic Ising model specific approximation in full generality. However, the temperature-dependent effects of the magnetic field mean that a fully general treatment is beyond the scope of this thesis.

Instead, we make a mean field approximation on the system. By averaging over the individual effects we lose some information, but are able to have a convenient substitution where $r_{i,j}(t) \equiv < \sigma_i(t)\sigma_j(t) > \approx < \sigma_i >< \sigma_j > \equiv q_i\, q_j$. Thus for our numerical solutions in Python we implement the following equation:

$$\frac{\partial}{\partial t}q_i(t) = -q_i + \frac{\gamma_i}{2}(q_{i-1} + q_{i+1}) + \beta_i - \frac{\beta_i\,\gamma_i}{2}(q_{i-1}\,q_i + q_i\,q_{i+1}) \tag{29}$$

One important observation to make is that the steady state magnetization is no longer zero in general. This is because steady state magnetizations most naturally come about from $\frac{\partial}{\partial t}q_i(t) = 0$ for all $1 \leq i \leq N$, and we can see that $\frac{\partial}{\partial t}q_i(t) = \beta_i$ if all particles are at zero means that this will no longer be a steady state in general.

For a constant field the steady state magnetization value of sublattices and the entire system can be solved numerically on their own by solving the system of equations given by the particle expectation values and setting their derivatives to zero. This is implemented in a Mathematica file which can be found in Appendix B. A time-dependent field such as a sine wave will not have a single steady state magnetization value, so cannot be approached in this manner.

In the thermodynamic limit of the blocks of temperatures, we can actually find an analytic solution for the sublattice (and thus the overall) magnetizations. By using the mean field approximation and letting each particle have the same spin as its neighbor, we find the following expression for the sublattice equilibrium magnetization $\overline{m_x}$:

$$\begin{aligned}\overline{m_x} = \frac{1}{2}\coth(\frac{H}{T_x})\Bigg(1 + \coth(\frac{2}{T_x})\Bigg(\\ -1 + \sqrt{1 + \tanh^2(\frac{2}{T_x}) + \tanh(\frac{2}{T_x})\big(-2 + 4\tanh^2(\frac{H}{T_x})\big)}\Bigg)\Bigg)\end{aligned} \tag{30}$$

This can be evaluated for a given temperature T_x and constant field strength H. Averaging over these sublattice magnetizations will naturally produce the steady state overall magnetization. An interesting result of the magnetic field is that different sublattices relax to different spin values based on their temperatures. This can be seen in Figure 29 below.

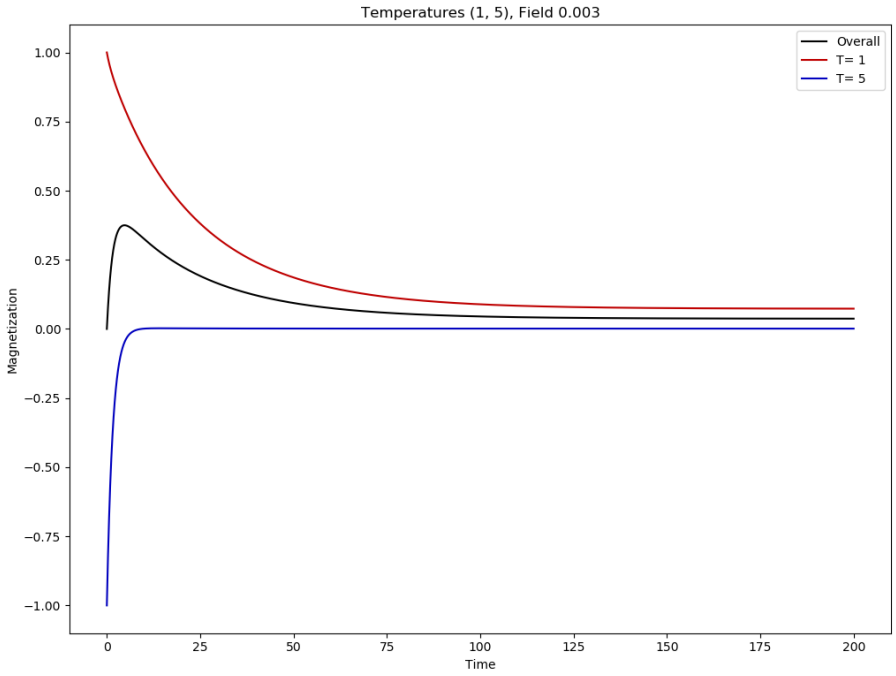

Figure 29: The sublattice and system magnetizations for a system of 100 particles with periodic boundaries, blocks of temperatures (1, 5), and a magnetic field with $H = 0.003$ in arbitrary units. We again see the higher temperature block relaxing more quickly, but to a different value than the other sublattice.

In Figure 30 below we find that the expression we computed for the thermodynamic limit is the limiting behavior in our numerical simulations. The requisite system size to visibly reach this limiting behavior is similar to what we saw for the temperature blocks without a magnetic field to reach steady states similar to their thermodynamic limits.

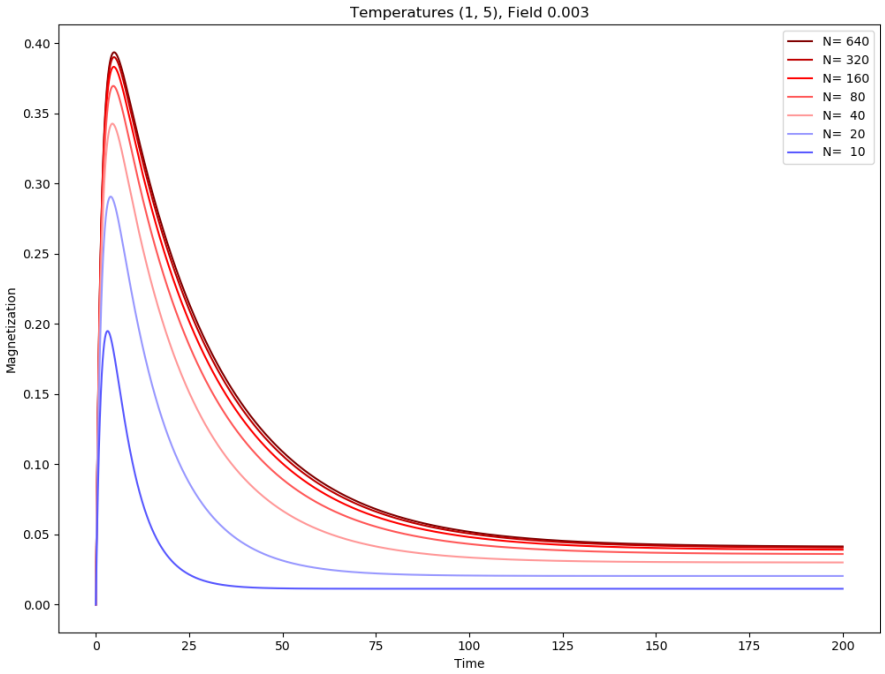

Figure 30: The overall magnetizations of temperature blocks of temperatures (1, 5), a magnetic field of strength $H = 0.003$ and varying system size. Here we show systems of 10, 20, 40, 80, 160, 320, and 640 particles. The larger systems have larger magnetizations both in the time-dependent dynamics and the steady state, and we can see the systems approach the thermodynamic limit.

On the other end of the system size spectrum, we once again see a reversal of the short term dynamics when we go between the block and alternating sublattice structures. As demonstrated in Figure 31 below, there are very large differences between systems block sizes N/X of 1, 2, and 3. However, the magnetic field does alter the steady state magnetizations the systems reach. In the specific example below, the alternating system's magnetization goes negative before settling at roughly 0.01. By contrast, the thermodynamic limit of the steady state magnetization is about ten times that, at 0.1.

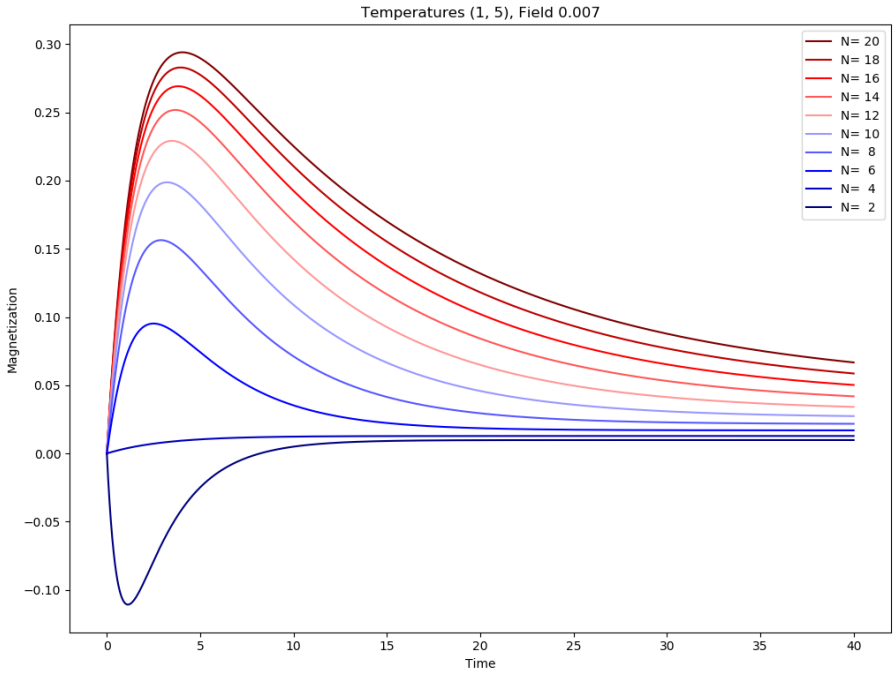

Figure 31: The overall magnetizations of temperature blocks of temperatures (1, 5), a magnetic field of strength $H = 0.007$ and varying system size. Here we show systems of $2i$ particles for $1 \leq i \leq 10$. Note how the $N/X = 1$ system is qualitatively very different, and the steady state magnetization rising as the system size increases.

We find that varying the field magnitude can move the system between two limiting cases. First, the zero field limit which we have discussed at length. Second, the maximal field strength limit where the magnetization shoots up to a steady state of $+1$. This is demonstrated in Figure 32 below using positive fields of magnitudes $H = 3^i$ for $-7 \leq i \leq 2$.

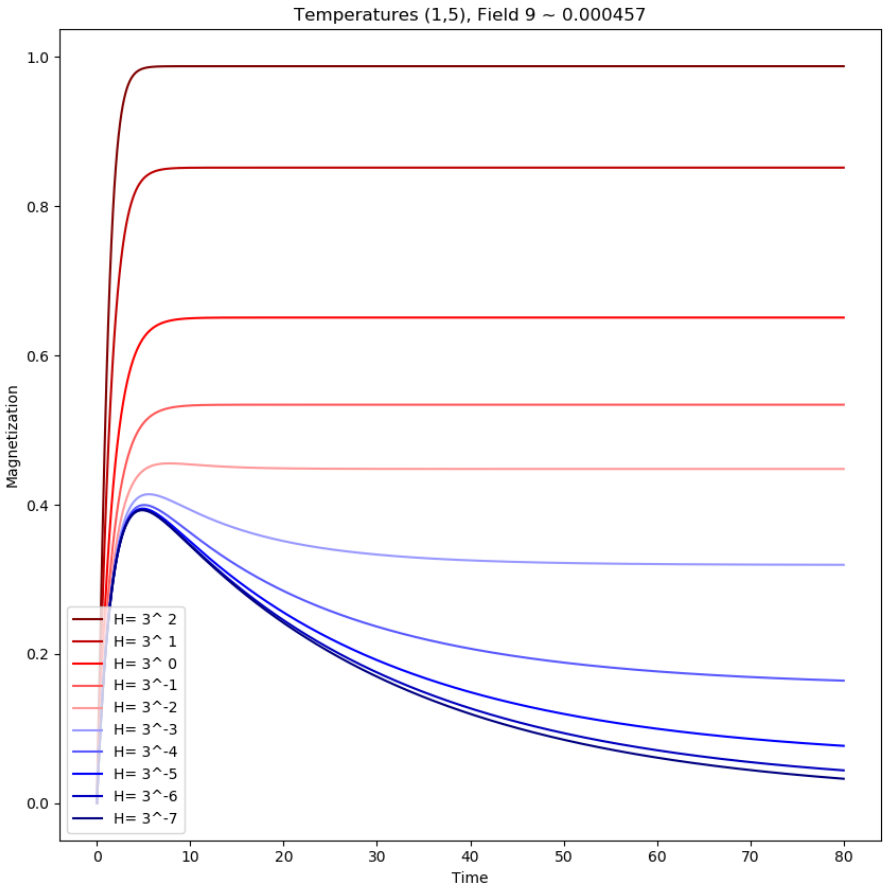

Figure 32: The overall magnetizations of periodic block systems in the thermodynamic limit with temperatures (1, 5). The field strengths are given by $H = 3^i$ for integers $-7 \leq i \leq 2$, with the strongest fields producing a steady state magnetization of nearly $+1$ and gradually dropping to the zero field limit we have previously explored.

This logic can be extended to the cases with negative field magnitudes, as shown below in Figure 33. Notably, the initial movement is towards the positive magnetization even with a negative field because of the time-dependent dynamics of the initial conditions paired with the two block temperatures. The higher temperature blocks relax more quickly, so if the temperatures were swapped the initial movement would be negative. Note that with a sufficiently large field strength this initial tendency can be overpowered completely.

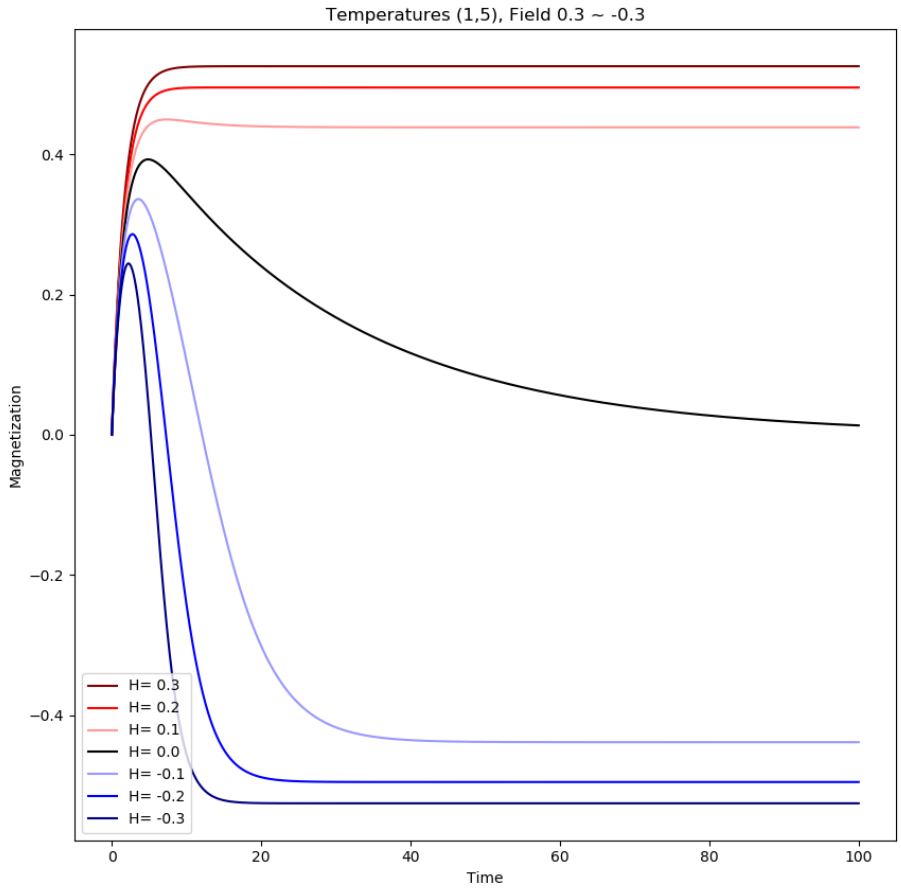

Figure 33: The overall magnetizations of periodic block systems in the thermodynamic limit with temperatures (1, 5). The field strengths are given by $H = i/10$ for integers $-3 \leq i \leq 3$. Note that the steady states of the negative fields are simply the opposite signs of their positive counterparts.

Another interesting quantity to play with is the temperature while keeping its ratio with the magnetic field strength constant. Recall that the magnetic field is included in the equation via $\beta = \tanh(\frac{H}{T_i})$, which is the root of the sublattice temperature dependent steady states we saw above. This means that by keeping the ratio the same we maintain the field's absolute influence on the system while adjusting the temperature dependent spin-spin interactions strength.

An example of this is shown in Figure 34 below, where the two sublattice temperatures and the magnetic field strength are all linearly dependent on a single parameter. We find that the systems generally share a similar magnetization curve, although it does depend on the magnitude of the steady state magnetization versus the peak of the time-dependent dynamics. By keeping the temperature and field ratio

constant we are effectively keeping our field factor β constant, so the larger steady states are a trait of the lower temperature systems rather than the systems with a stronger field parameter.

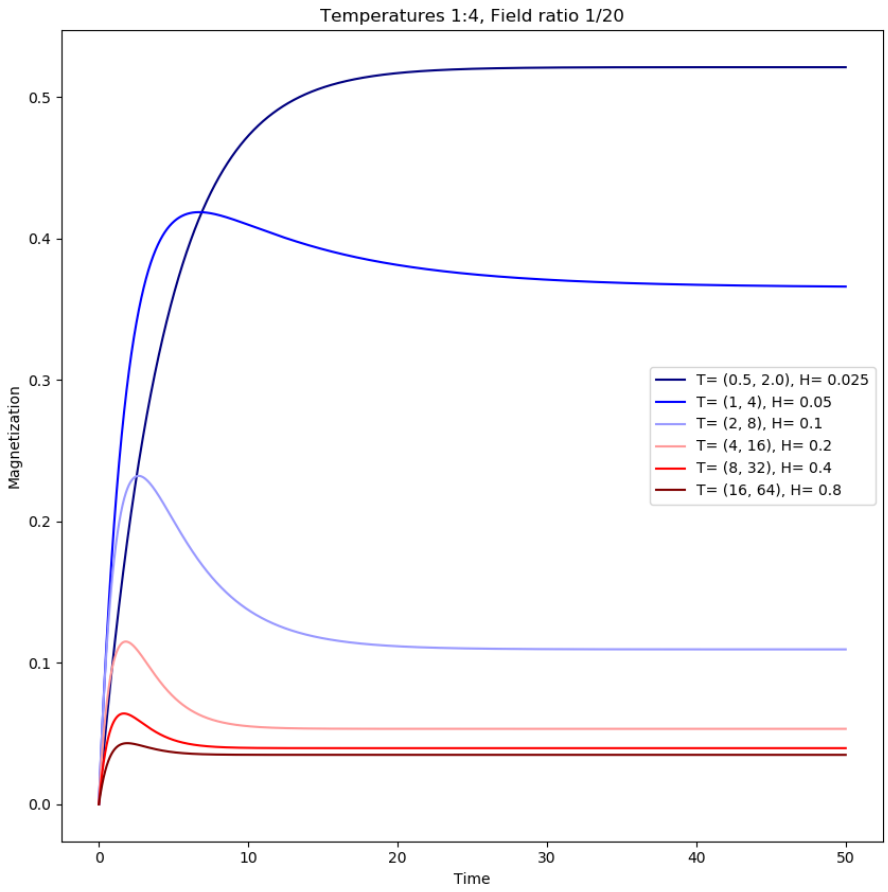

Figure 34: The overall magnetizations of block systems where the initially negative spin sublattice temperature is four times that of the initially positive, and the positive magnetic field strength is kept at $\frac{1}{20}$ of the value of the lower temperature. The units are arbitrary but the relations are constant while the temperatures are adjusted. We see that the overall magnetizations generally have a similar shape scaled by their steady state value, until the case where the steady state is a larger value than the time-dependent dynamics reach. This larger steady state is produced by lowering each value, i.e. when the magnetic field itself is weaker.

Much like the no field limit, these behavior trends can be directly translated to the systems with more sublattices. In Figures 35, 36, and 37 below we have three examples of systems with three temperature blocks near the thermodynamic limit. The steady

state values can once again be computed by averaging the sublattice steady states, and the time-dependent dynamics are a result of the different relaxation times.

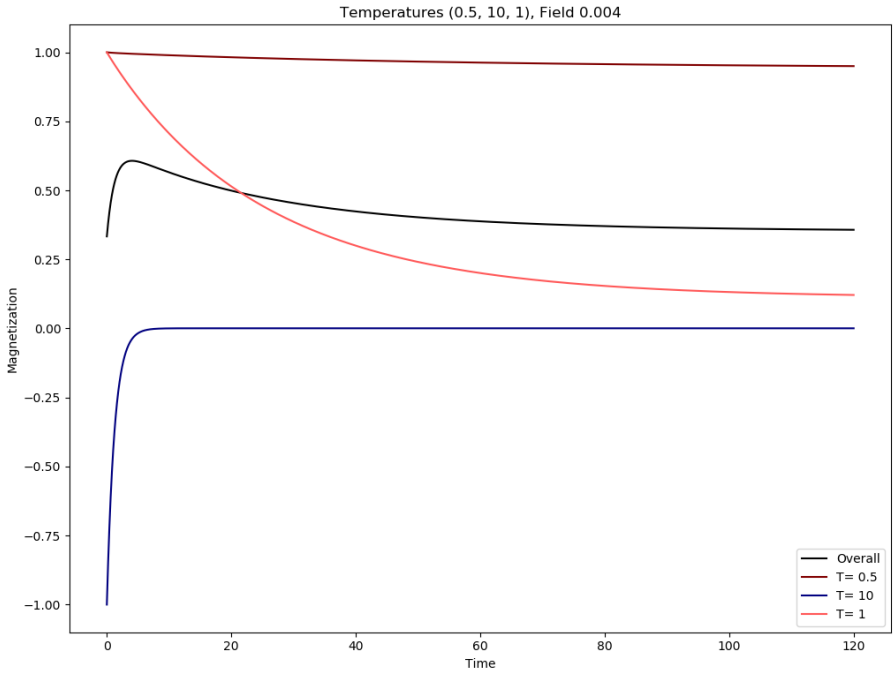

Figure 35: The dark red, blue, and light red lines represent the magnetizations of sublattices of temperatures $(\frac{1}{2}, 10, 0.1)$ respectively. The positive field of magnitude $\frac{1}{250}$ produces the positive steady state magnetizations.

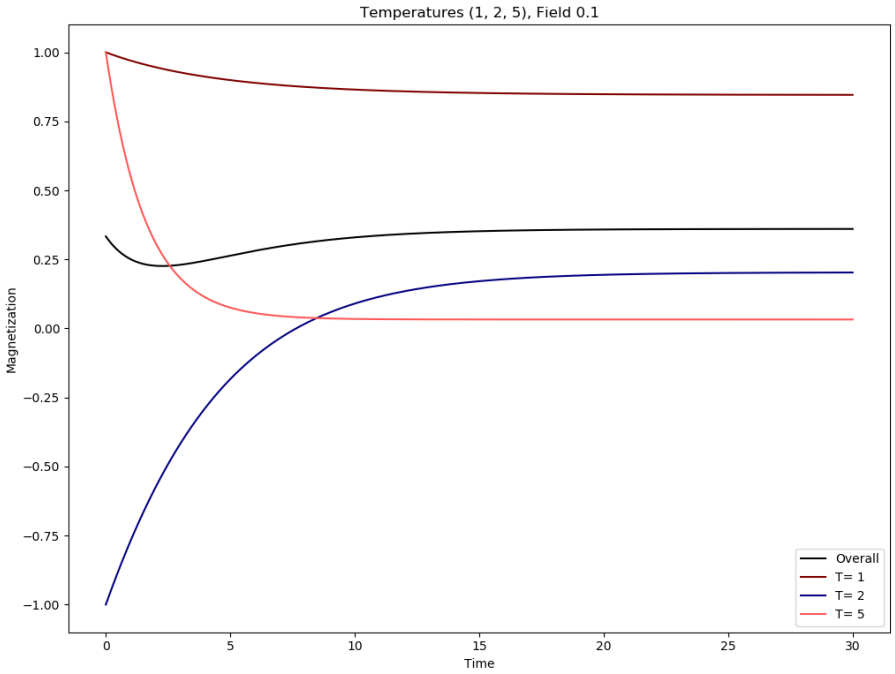

Figure 36: The dark red, blue, and light red lines represent the magnetizations of sublattices of temperatures (1, 2, 5) respectively. The positive field of magnitude 0.1 produces the positive steady state magnetizations.

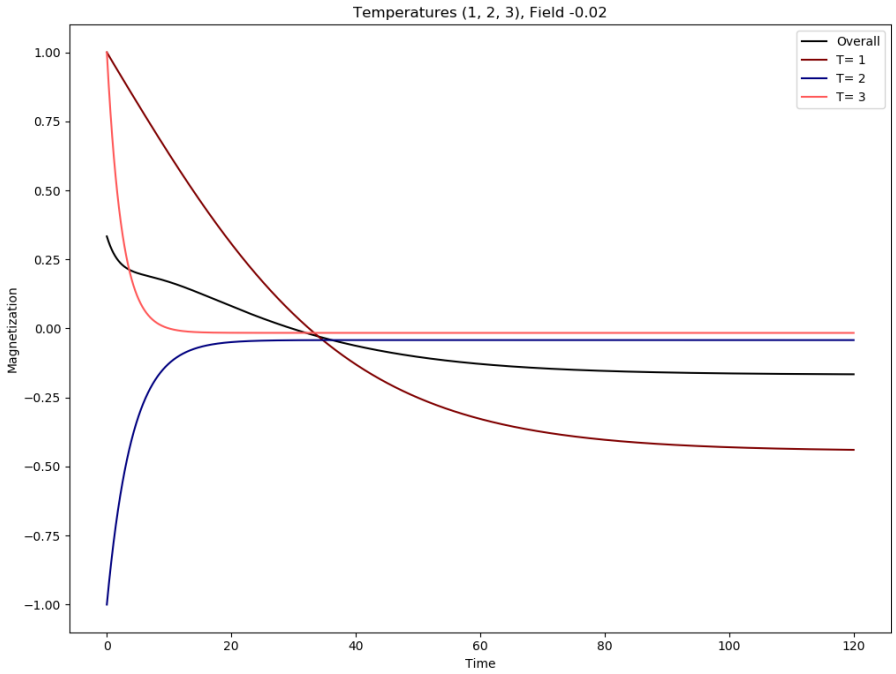

Figure 37: The dark red, blue, and light red lines represent the magnetizations of sublattices of temperatures (1, 2, 3) respectively. The negative field of magnitude 0.02 produces the negative steady state magnetizations.

Similarly, going beyond two temperatures in the alternating sublattice structure does not produce any unfamiliar behaviors. While we do not have a formal proof, it anecdotally appears that each sublattice's magnetization can still be modeled as a sum of two exponentials plus a constant. It is important to note that these steady state offsets are dependent on the global temperature pattern rather than just the temperature bath associated with the given sublattice. This is because the alternating temperature pattern has a lot of interactions between sublattices, as opposed to the thermodynamic limit of the block structure where these are negligible.

We can see the effects of swapping temperatures around in Figure 38 below. While having two copies of the overall and each of four sublattice magnetizations makes it a little cumbersome, note how the steady states do not all line up. We again see that higher temperatures are less influenced by the magnetic field and neighboring spins, due to their smaller β and γ values in the expectation value equations.

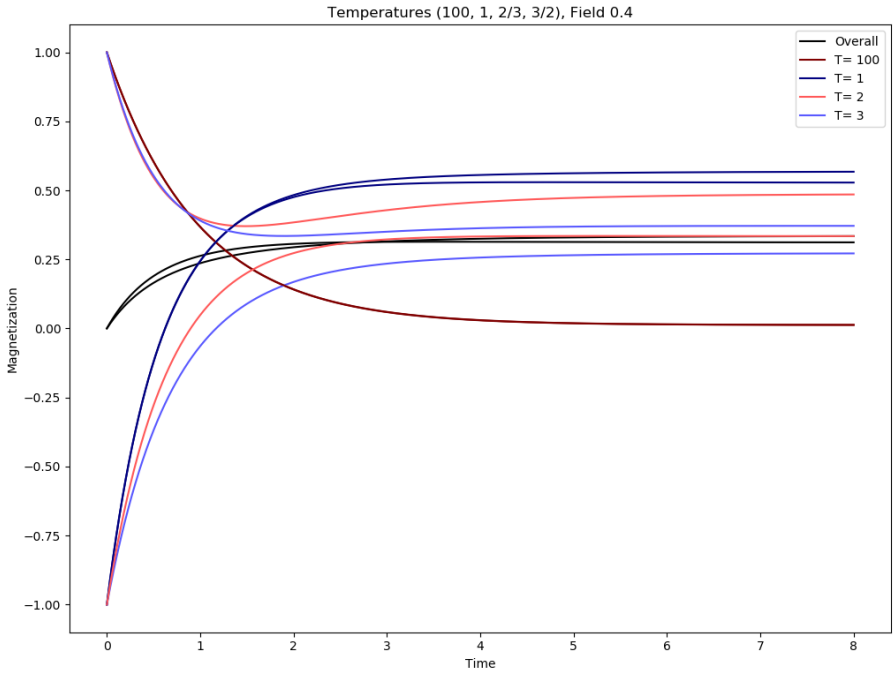

Figure 38: A plot with two sets of magnetizations, for temperatures (100, 1, 3, 2) and (100, 1, 2, 3) with an alternating temperature structure. Note how the sublattices generally reach different steady states, with the exception of the dark red lines which are essentially identical. In fact the overall steady state magnetizations differ slightly as well, though this is hard to see due to the many overlapping curves. This is distinct from the temperature block structure where steady state values are independent of neighboring sublattice temperatures in the thermodynamic limit.

What does the constant magnetic field mean for our magnetization equations? We still find the overall magnetization to be a sum of exponential decays, but now with constants added to specify the non-zero steady states. The form is otherwise unchanged.

This does however add a twist to some possible physical analogies. One case where this can be understood quite clearly is the blocks of temperatures in particular acting as independent damped harmonic oscillators. We are quite familiar with the effect of constant forces on harmonic motion. In the standard example of a spring on a mass hanging down from a spring, the force of gravity merely changes the position of the equilibrium based on the spring constant and the mass hanging from it. In a similar manner, we can see the kinetic Ising model's magnetic field as effectively shifting the position of equilibrium based on the temperature of the sublattice. As we move away from the thermodynamic limit of the blocks, the sublattices lose their independence, but are easily understood on a qualitative level and fully studying the particular system of equations does allow for exact predictions of this steady state.

As mentioned in the beginning of this section, the general expectation value

equations with the magnetic field are derived using the mean field approximation. Therefore, we cannot be certain that the analysis above is a complete one, and should be supplemented with further work. In particular, comparisons between Monte Carlo simulations and the expectation value work would serve to check the accuracy of the mean field approximation. A full cross-examination would require averaging the results of an enormous number of Monte Carlo simulations, which should probably be preceded by revisiting our existing code attached in Appendix C. It is functional but given the computational power in question we would want to increase the efficiency, which might even call for using a language other than Python.

3. Cayley Trees

Having analyzed a variety of configurations for the kinetic Ising model in one dimension, we would like to look at the model's behaviors when applied to different graphs. However, this introduces some complications which we will quickly outline below.

We still maintain the same flip rate, shown in a more general form below for a particle with n neighbors indexed by j. At a given time step we choose a random particle σ_i, and flip it with the probability w_i. However, the process of turning this transition rate into expectation value equations depends on the graph and how particles are connected as neighbors.

$$w_i(\sigma_i) = \frac{1}{2}(1 - \frac{1}{n}\gamma_i \sigma_i \sum_{j=1}^{n} \sigma_j) \cdot (1 - \beta_i \sigma_i) \tag{31}$$

Thankfully Cayley Trees are a relatively straightforward graph, providing many symmetries to simplify the problem [21]. Two examples are shown below in Figures 39 and 40. We categorize them based on their number of generations g (not including the single generation 0 particle located in the center of the image) and the number of neighbors which is given by the coordination number z. Particles in the final generation are called leaves and only have one neighbor unlike all other generations.

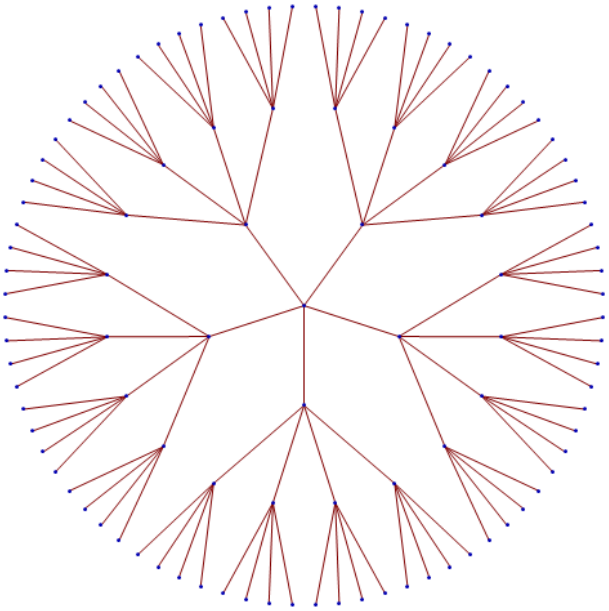

Figure 39: A Cayley Tree with three generations and a coordination number $z = 5$.

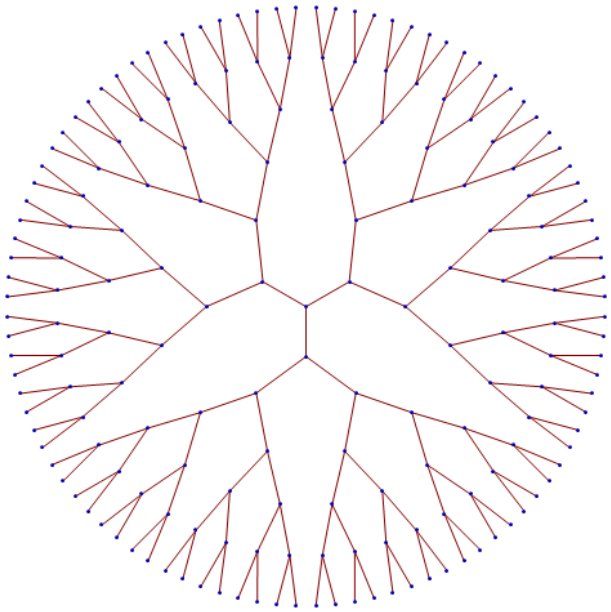

Figure 40: A Cayley Tree with six generations and a coordination number $z = 3$.

Despite all this symmetry, the expectation value equations are still challenging to express in closed form. Even in the single temperature case without a field, where there is thermal equilibrium and all γ values are the same, the equation must be approximated and has coefficients which are non-trivially dependent on the coordination number z [22]. Below are these approximated expectation value equations to third order, where each Cayley Tree generation is taken to be homogeneous so that all particles within a given generation have identical expectaiton values. The i in q_i is the generation number, $\gamma \equiv \tanh(\frac{1}{T})$ is the inverse temperature for the whole system, and γ' is an inverse temperature where the temperature T is replaced by $T/3$.

$$\begin{aligned}
\frac{\partial q_0}{\partial t} &= -q_0 + \frac{3}{4}(\gamma' + \gamma)q_1 + \frac{1}{4}(\gamma' - 3\gamma)q_1^3 \\
\frac{\partial q_i}{\partial t} &= -q_i + \frac{1}{4}(\gamma' + \gamma)(q_{i-1} + 2q_{i+1}) + \frac{1}{4}(\gamma' - 3\gamma)q_{i-1}q_{i+1}^2 \\
\frac{\partial q_g}{\partial t} &= -q_g + \gamma q_{g-1}
\end{aligned} \tag{32}$$

This produces a system of equations far more complicated than what we had for the one-dimensional systems, and may even have additional terms arise with the multi-temperature generalization. This is all without a magnetic field, which in the one-dimensional systems doubled the number of terms and required approximations for a closed form equation.

Numerical work of this kind may be interesting to pursue in the future. However, we sidestep this issue by doing Monte Carlo simulations of these systems and averaging over a large amount of trials, using the code shown in Appendix D. This approach allows us to avoid any approximations in exchange for requiring significantly more computational time to get expectation values. Therefore the following analysis is more qualitative and less thorough than the numerical expectation value analysis we conducted for one-dimensional systems.

3.1. Monte Carlo Simulations

For Cayley Trees we treat each generation as a sublattice, with uniform initial states and temperature baths. We then alternate temperatures each generation, so that moving from the central particle down the generations looks like moving along a one-dimensional lattice with the alternating configuration.

First let us see what a single evolution of a two-temperature Cayley Tree looks like. In Figure 41 we see a discontinuous and erratic overall magnetization, as expected due to the stochastic nature of the system. We also see that nearby spins are highly correlated in this particular run, with very few neighbor pairs having opposite spins.

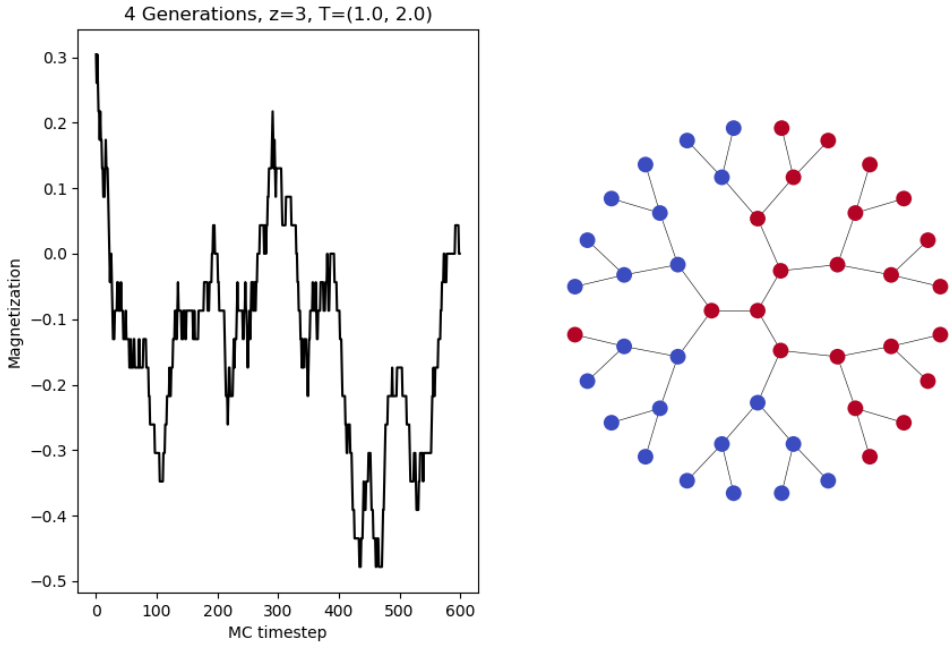

Figure 41: Left: The overall magnetization of Cayley Tree with 4 generations, a coordination number $z = 3$, and temperatures (1, 2) alternating by generation. Right: The final state of the particle spins after 600 steps, with red and blue representing up and down respectively. Note how the up and down spins are grouped together due to the neighbor interactions.

Now we observe the averaged behavior of this system in Figure 42 below, which approximates the expectation values by combining the results of 3500 simulations. We see familiar exponential decay towards zero magnetization due to the lack of a field, with each sublattice of the same temperature and initial orientation acting similarly. The lone exception is the final generation, the leaves plotted in green, which decay notably more quickly and see the average magnetization flip signs in the time-dependent dynamics.

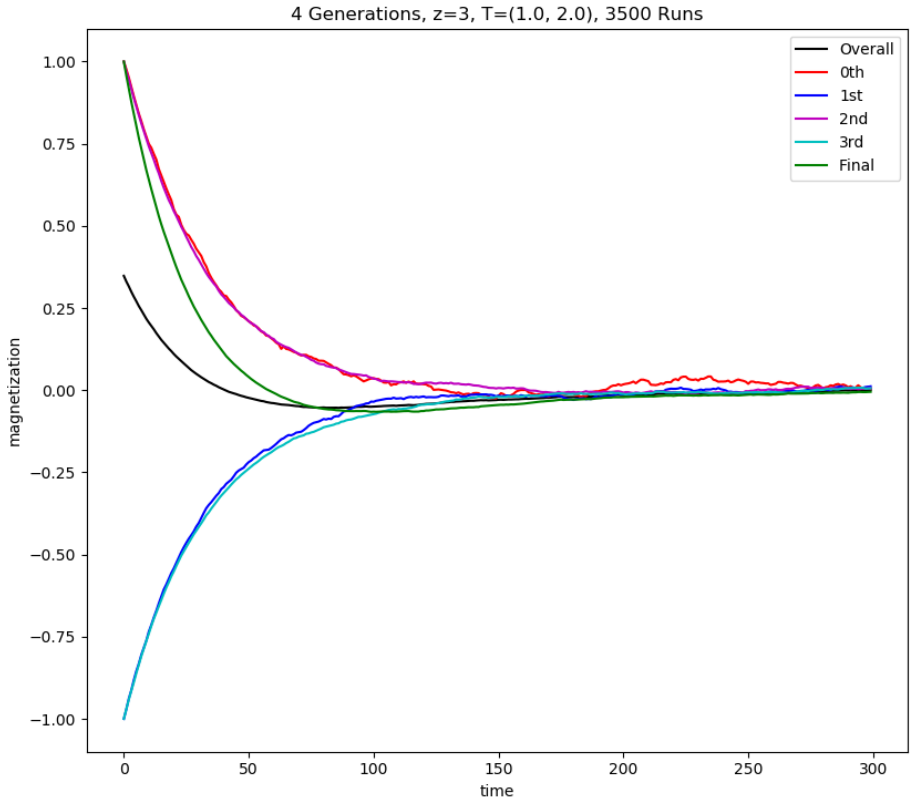

Figure 42: The overall and generation magnetizations for the same system above, averaged over 3500 runs. Note how the earlier generations have less smooth lines because there are fewer particles in them to average over. The green line which represents the leaves' average magnetization is clearly the odd one out, relaxing more quickly than the red or purple lines.

This behavior is present regardless of the number of generations, as seen in Figure 43 below, where the two additional generations follow the relaxational path of the previous ones with the final generation once again being an exception. This is presumably due to the structure of their equations being different, with the single neighbor making it easier for the leaf particle to flip between up and down regardless of the rest of the system. The large amount of symmetry in the system is likely responsible for this behavior.

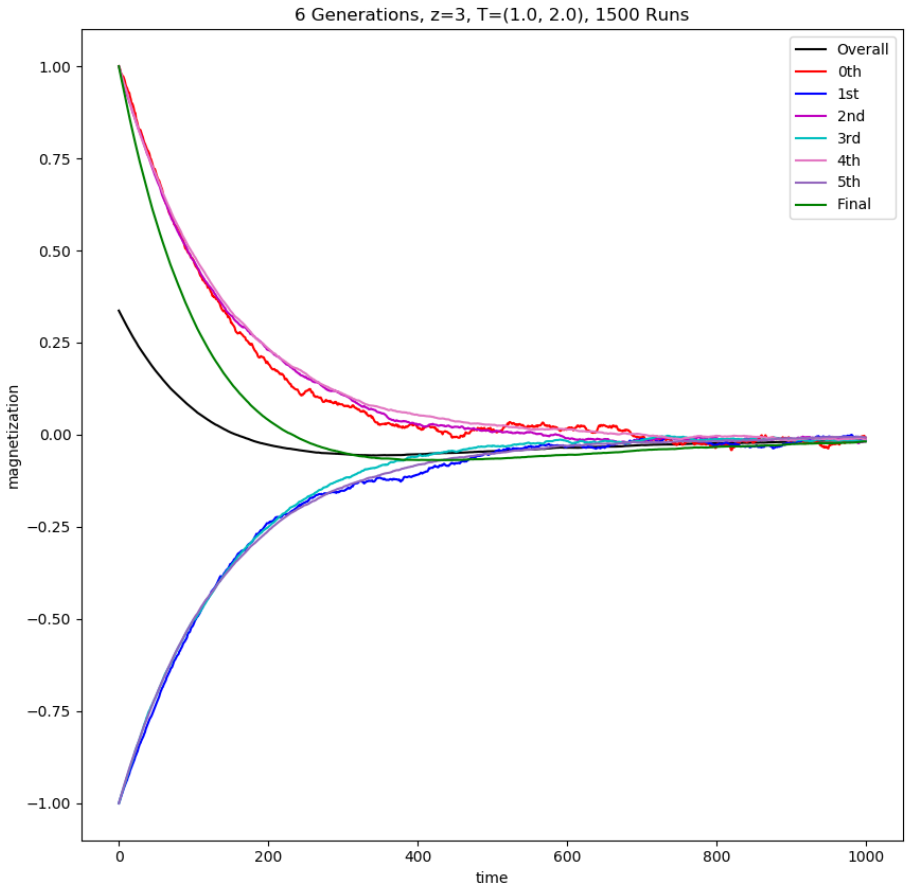

Figure 43: The overall and generation magnetizations for a Cayley Tree with 6 generations, coordination number $z = 3$, and temperatures (1,2) averaged over 1500 runs. Note the final leaf generation's unique behavior.

What happens when we adjust the coordination number z? An educated guess is that we might observe a slower decay to zero with more neighbors, but otherwise the same dynamics. This is in fact what we find, as demonstrated below by Figure 44. Note how the larger z makes the system's overall initial magnetization higher because there are now far more particles in each subsequent generation (in the example below each generation is 5 times the size of the previous one as opposed to the doubling which occurs for $z = 3$).

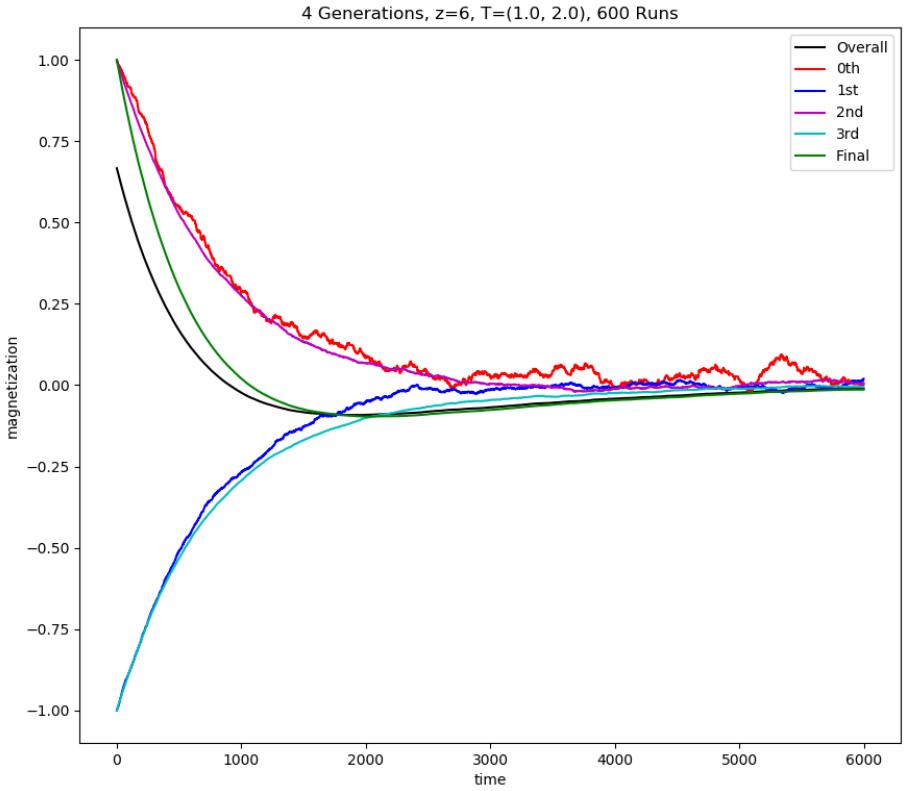

Figure 44: The overall and generation magnetizations for a Cayley Tree with 4 generations, coordination number $z = 6$, and temperatures (1,2) averaged over 600 runs. Note the significantly longer time scale, as well as the larger initial overall magnetization.

Next, we demonstrate the effects of changing the temperatures. This also behaves according to expectations, as larger temperature differences mean larger time-dependent deviations. This can be seen in Figure 45 below which plots the overall magnetization of systems with different temperatures in different colors. The time it takes to reach the steady state visibly varies, and the precise mechanisms behind this are an open question to be tackled with more analytical work as opposed to our Monte Carlo simulations.

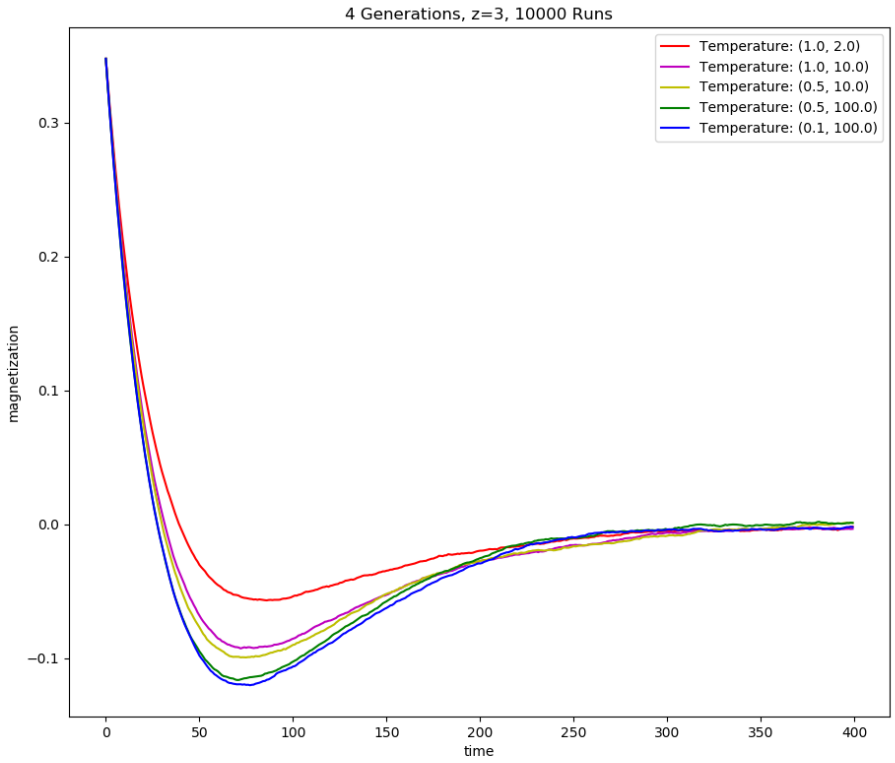

Figure 45: The overall magnetization for a system with four generations and $z = 3$ for a variety of temperature pairings, obtained by averaging over 10,000 simulations.

In the multi-temperature extension, we lose the symmetry of the two-temperature systems and see a greater variety of relaxation rates as demonstrated in Figure 46 below. Without the field, the steady state is still zero.

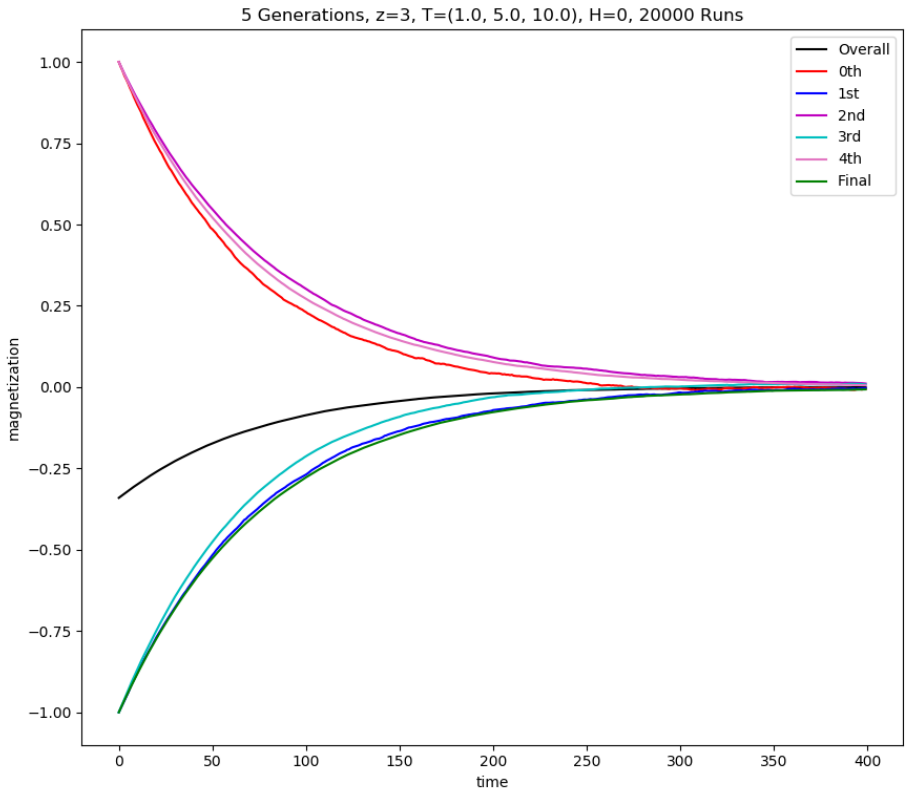

Figure 46: The overall and generation magnetizations of a Cayley Tree with five generations, $z = 3$, and temperatures (1,5,10) averaged over 20,000 simulations. Note how the generations act differently because none of them share both initial conditions and temperature.

Finally, the inclusion of a magnetic field changes the steady state magnetization of the generations as can be seen below. Notice again how the final leaf generation behaves uniquely in Figure 47 for a system with two temperatures.

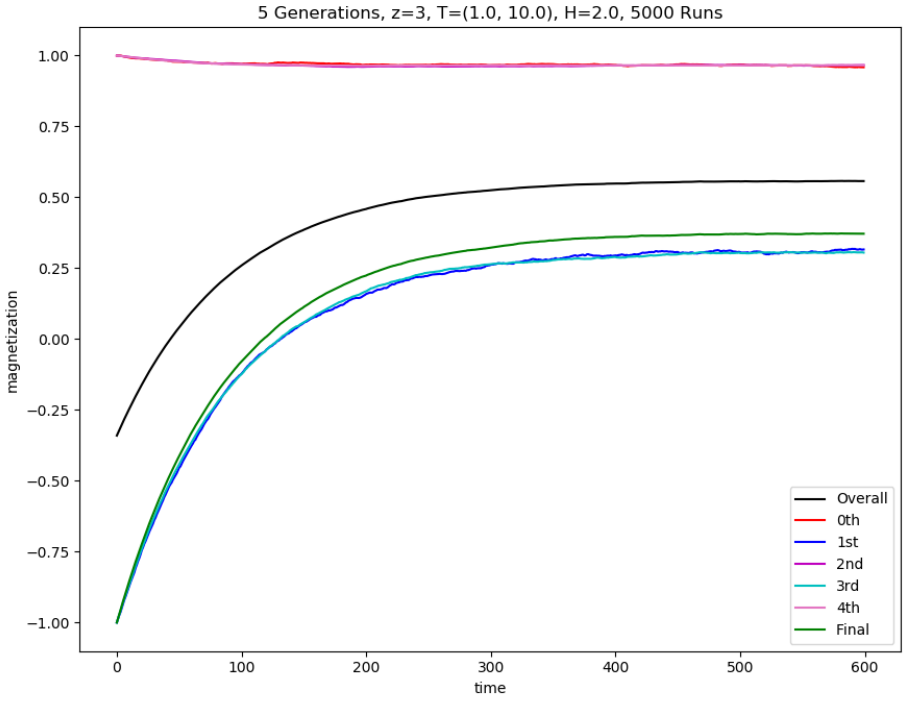

Figure 47: The overall and generation magnetizations for five generations, $z = 3$, temperatures (1,10), and field strength $H = 2$ averaged over 5,000 simulations. Again the final generation shown in green behaves differently.

Again, this anomalous behavior does not depend on factors such as the generations or coordination number. We can see the same behavior in Figure 48 for a system with one fewer generations, meaning the anomalous leaf generation once again stands out in green.

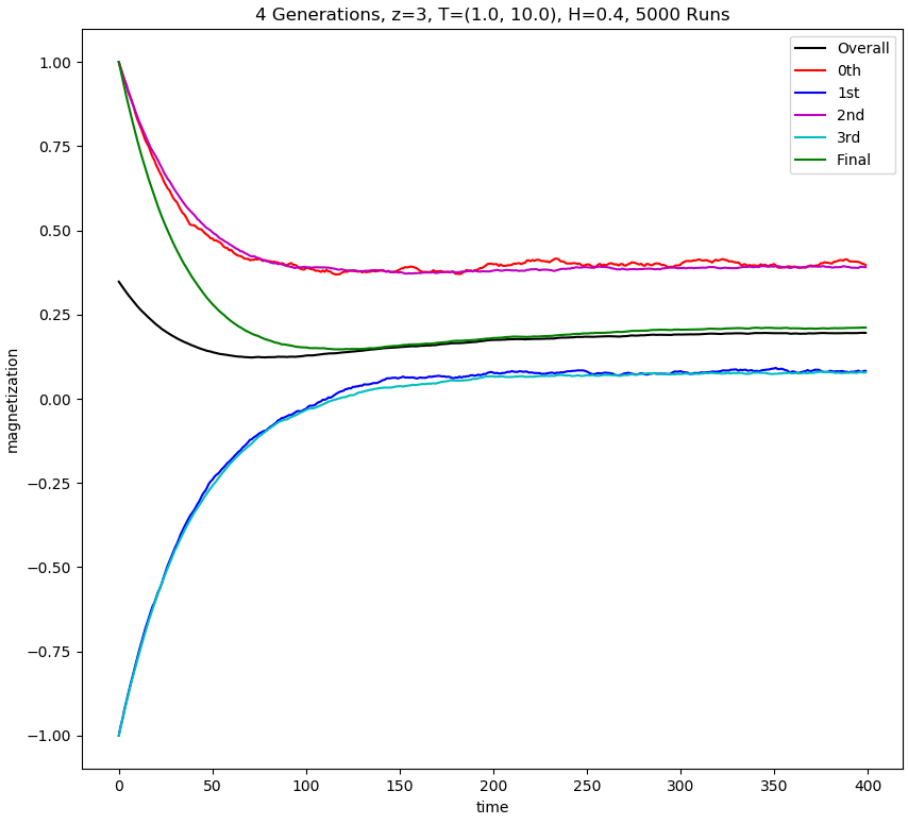

Figure 48: The overall and generation magnetizations for four generations, $z = 3$, temperatures (1,10), and field strength $H = 0.4$ averaged over 5,000 simulations.

Generalizing to more temperatures causes a loss in symmetry, and we have more variety in the steady states as seen in Figure 49. One pair of same temperature generations do end up with similar values (shown in pink and blue below), but the other two do not. This is is likely due to the 0^{th} and leaf generations having different equation structures. The new difference for the 0^{th} generation is likely because the number of temperatures is now three rather than two, so it only neighbors particles of the second temperature (three $T = 1$ particles in the example below) vs. the 3^{rd} generation which neighbors particles of two different temperatures (one $T = 10$ and two $T = 1$ particles in the example below). More analytical work needs to be done to better understand the exact mechanisms at play and confirm this hypothesis.

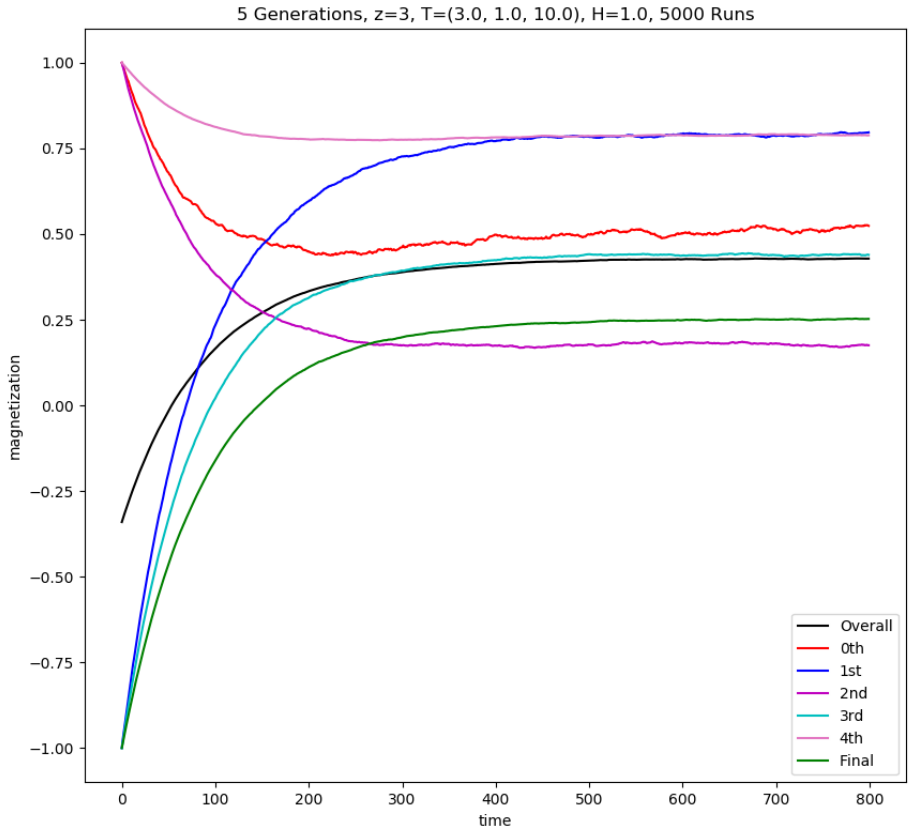

Figure 49: The overall and generation magnetizations for five generations, $z = 3$, temperatures (3, 1, 10), and field strength $H = 1$ averaged over 5,000 simulations. Note how the first and fourth generations have the same steady states, but the others do not pair up as one might have naively expected.

The breaking symmetry coming from the initial and final generations can be seen in Figure 50 below as well, this time with an additional generation and a negative field. The influence of the negative field is as we expect based on what we observed with the one-dimensional systems.

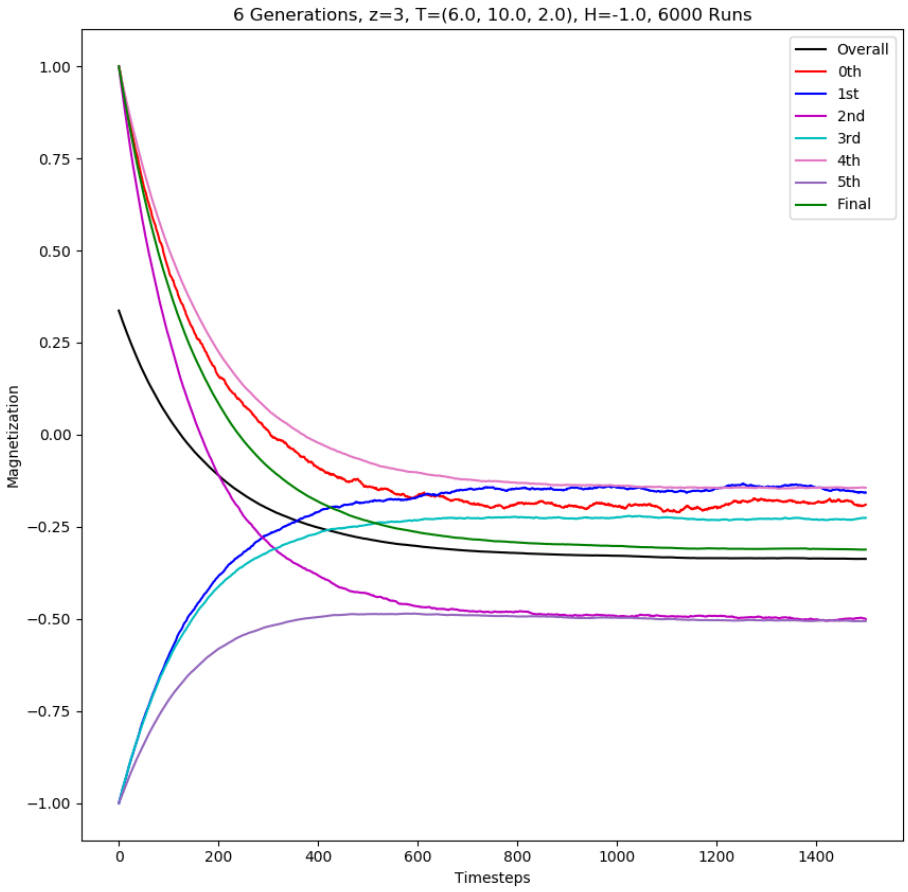

Figure 50: The overall and generation magnetizations for six generations, $z = 3$, temperatures (3, 1, 10), and field strength $H = -1$ averaged over 6,000 simulations. Note how the initial, third, and final generations have different steady state values.

In addition to the obvious next step of more analytic work, an interesting idea for further work is trying different temperature patterns on the Cayley Tree. Perhaps something like the equivalent of the blocks of temperatures for the Cayley Tree where neighboring generations are grouped together would allow for them to be treated somewhat independently as in the one-dimensional case. This would likely increase the number of time steps necessary to reach the magnetization steady state, further increasing the computational power necessary for each simulation. The different graph structure may make it harder to treat the blocks independently, requiring a large number of generations to observe the limiting behavior, and therefore even further computational time.

It should also be noted that analytic work does not exclusively mean deriving a wider range or higher orders of the time-dependent expectation value equations. It may

be valuable to simply try computing the steady state magnetizations of each generation in a field assuming homogeneity, taking a common approach in non-equilibrium literature of putting aside messy dynamics and focusing on studying tractable non-equilibrium steady states. It may be that the assumption of homogeneity will not accurately reflect the actual dynamics on Cayley Trees due to particle correlations not present in the one-dimensional system, but this is not presently known. Comparing even rudimentary analytical work to the results of Monte Carlo simulations can serve to reinforce our understanding going forward.

4. Conclusions

In this thesis we have seen a wide range of behaviors exhibited by the multi-temperature kinetic Ising model. We have focused on the dynamics of the spin magnetizations as well as the non-equilibrium steady states. In doing so we have found several patterns, developing an intuition for how the magnetizations evolve in time. Greater temperature differences lead to more extreme dynamics while the steady states are determined by the field strength as well as the temperatures and their configurations on the graphs. Boundary conditions and sublattice sizes and their effects were explored, as we found a smooth transition from the one-dimensional thermodynamic limits of blocks to the alternating temperatures. For the temperature blocks we see that each sublattice can be treated as fairly independent of the others, simplifying the analysis of the problem at hand. Many of these behaviors translated to the model on Cayley Trees, where we found that the final leaf generation acted in a unique manner in the two-temperature systems, likely due to the graph structure. The multi-temperature systems lost some symmetry in the relaxational dynamics, but still followed familiar exponential decays. Finally, the addition of a magnetic field was explored, and we saw the initial generation join the leaf generation in deviating from the remaining middle generations due to the structure of Cayley Tree graphs.

We also pointed towards some physical analogies for the one-dimensional system, which arise from the mathematical structure of the expectation value equations. These analogies can both enhance our understanding of the systems at hand and prompt novel angles to approach analyses of them. They may also serve to better engage an audience in the rather abstract matter of spins in a lattice flipping between two orientations.

Next steps along similar lines of this thesis could involve exploring a wider range of graphs and temperature patterns. We saw that an understanding of the one-dimensional system aided comprehension of the model on Cayley Trees, and analysis of a system on higher dimensional lattices may be aided by the work presented in this thesis. As analytic work becomes more complicated, the Monte Carlo approach shown for the Cayley Trees is a helpful first step, but explicit steady state magnetizations may be a worthwhile endeavor as well. An example of this is how explicit analytical results may better explain the mechanism behind the leaf generation's behavior. The Ising model in higher dimensions shows phase transitions absent in the one-dimensional lattice, and exploring the potential implications of this in the multi-temperature kinetic Ising model is an interesting natural next step from the work in this thesis. An additional area to explore is time-dependent magnetic fields. This additional complication has been shown to introduce glassy behavior in very simple kinetic Ising models, and its full implications in the variety of setups we have explored is unknown. This could potentially strengthen the analogies to molecular glasses and stretched exponential decays, furthering cross-model comparisons.

For the larger goal of better understanding of non-equilibrium systems in general, it may also be interesting to explore things like two-spin correlations. This can be easily

done in the Monte Carlo approach as well, but do require much more computational power, particularly as the systems become larger and more intricately connected. Framing the problem in terms of energy and energy fluxes will likely help make future work more accessible for others in the field who are interested in looking for insights across a wide variety of graphs and systems.

This is just an example of future work which is connected to the work in this thesis in a straightforward manner. As the search for a unifying framework continues, it is important to approach the problem with an open mind, as we do not know the source of the next breakthrough. It is unlikely to come from this thesis, but the hope is that this addition to the great body of existing research may someday come in handy for physicists seeking a more comprehensive understanding of the complicated behaviors of non-equilibrium systems.

5. Bibliography

[1] Borchers, Nicholas, Michel Pleimling, and R. K. P. Zia. "Nonequilibrium statistical mechanics of a two-temperature Ising ring with conserved dynamics." *Physical Review E* 90.6 (2014): 062113.

[2] Baxter, Rodney J.*Exactly solved models in statistical mechanics.* Elsevier, 2016.

[3] Lavrentovich, M. O., and R. K. P. Zia. "Energy flux near the junction of two Ising chains at different temperatures." *EPL (Europhysics Letters)* 91.5 (2010): 50003.

[4] Mobilia, M., Beate Schmittmann, and R. K. P. Zia. "Exact dynamics of a reaction-diffusion model with spatially alternating rates." *Physical Review E* 71.5 (2005): 056129.

[5] Bauer, Michel, and Françoise Cornu. "Thermal contact through a two-temperature kinetic Ising chain." *Journal of Physics A: Mathematical and Theoretical* 51.19 (2018): 195002.

[6] Cornu, F., and M. Bauer. "Thermal contact through a diathermal wall: a solvable toy model." *Journal of Statistical Mechanics: Theory and Experiment* 2013.10 (2013): P10009.

[7] Lavrentovich, Maxim O. "Steady-state properties of coupled hot and cold Ising chains." *Journal of Physics A: Mathematical and Theoretical* 45.8 (2012): 085002.

[8] Lavrentovich, Maxim O. "Steady-state properties of coupled hot and cold Ising chains." *Journal of Physics A: Mathematical and Theoretical* 45.8 (2012): 085002.

[9] Baek, Seung Ki, and Fabio Marchesoni. "Nonequilibrium steady state of the kinetic Glauber-Ising model under an alternating magnetic field." *Physical Review E* 89.2 (2014): 022136.

[10] Ising, Ernst. "Beitrag zur theorie des ferromagnetismus." *Zeitschrift für Physik* 31.1 (1925): 253-258.

[11] Blundell, Stephen J., and Katherine M. Blundell. *Concepts in thermal physics.* OUP Oxford, 2009.

[12] Glauber, Roy J. "Time-dependent statistics of the Ising model." *Journal of mathematical physics* 4.2 (1963): 294-307.

[13] Rácz, Z., and R. K. P. Zia. "Two-temperature kinetic Ising model in one dimension: steady-state correlations in terms of energy and energy flux." *Physical Review E* 49.1 (1994): 139.

[14] Krapivsky, Pavel L., Sidney Redner, and Eli Ben-Naim. *A kinetic view of statistical physics.* Cambridge University Press, 2010.

[15] Diekmann, Odo, and Johan Andre Peter Heesterbeek. *Mathematical epidemiology of infectious diseases: model building, analysis and interpretation.* Vol. 5. John Wiley & Sons, 2000.

[16] Murray, James D. "Mathematical biology: I. An introduction. Interdisciplinary applied mathematics." *Mathematical Biology, Springer* (2002).

[17] Liggett, Thomas M. *Stochastic interacting systems: contact, voter and exclusion processes.* Vol. 324. springer science & Business Media, 2013.

[18] Gardner, Donald G., et al. "Method for the analysis of multicomponent exponential decay curves." *The journal of chemical physics* 31.4 (1959): 978-986.

[19] Phillips, J. C. "Stretched exponential relaxation in molecular and electronic glasses." *Reports on Progress in Physics* 59.9 (1996): 1133.

[20] Johnston, D. C. "Stretched exponential relaxation arising from a continuous sum of exponential decays." *Physical Review B* 74.18 (2006): 184430.

[21] Ostilli, Massimo. "Cayley Trees and Bethe Lattices: A concise analysis for mathematicians and physicists." *Physica A: Statistical Mechanics and its Applications* 391.12 (2012): 3417-3423.

[22] Mélin, R., et al. "Glassy behaviour in the ferromagnetic Ising model on a Cayley tree." *Journal of Physics A: Mathematical and General* 29.18 (1996): 5773.

Acknowledgments

Thank you to Professors Irina Mazilu, Dan Mazilu, and Tom McClain for being my thesis committee and putting up with me in this process. I am also grateful to Professor Laurentiu Stoleriu for being a great help with the Monte Carlo code.

Appendix A. Python Module for Expectation Values

```
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 25 22:41:53 2019
Updated 05/13/2021

@author: shogi

One D lattice, periodic and hard boudaries
Alternating and block temperature baths
Glauber dynamics with magnetic field
ODE solver
"""

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
import math
from scipy.optimize import curve_fit

##set colors as RGB tuples in [0,1]
c00='k' #black
c01=(0,1,0) #light green
c0=(0,0,0.5) #darkblue
c1=(0,0,0.75)
c2=(0,0,1)
c3=(0.35,0.35,1)
c4=(0.6,0.6,1) #lightblue
c5=(1,0.6,0.6) #lightred
c6=(1,0.35,0.35)
c7=(1,0,0)
c8=(0.75,0,0)
c9=(0.5,0,0) #darkred

#####define functions
def f1(y0, t, N, g0, B0): #Hard bounaries, removed the incorrect 1/2s
    derivs = np.zeros(N) #create empty array of differential equations
    derivs[0] = -y0[0]+g0[0]*(y0[1])+B0[0]\
    -B0[0]*g0[0]*(y0[0]*y0[1]) #set first endpoint
    derivs[N-1] = -y0[N-1]+g0[N-1]*(y0[N-2])+B0[N-1]\
    -B0[N-1]*g0[N-1]*(y0[N-1]*y0[N-2]) #second endpoint
```

```
    for i in range(1,N-1): #generate all central equations
        derivs[i] = -y0[i]+g0[i]*(y0[i-1]+y0[i+1])/2+B0[i]\
        -B0[i]*g0[i]*(y0[i]*y0[i+1]+y0[i]*y0[i-1])/2
    return derivs

def f0(y0, t, N, g0, B0): #periodic boundary conditions
    derivs = np.zeros(N) #create empty array of differential equations
    derivs[0] = -y0[0]+g0[0]*(y0[1]+y0[N-1])/2+B0[0]\
        -B0[0]*g0[0]*(y0[0]*y0[1]+y0[0]*y0[N-1])/2 #set first endpoint
    derivs[N-1] = -y0[N-1]+g0[N-1]*(y0[N-2]+y0[0])/2+B0[N-1]\
        -B0[N-1]*g0[N-1]*(y0[N-1]*y0[N-2]+y0[N-1]*y0[0])/2#second
            ↪ endpoint
    for i in range(1,N-1): #generate all central equations
        derivs[i] = -y0[i]+g0[i]*(y0[i-1]+y0[i+1])/2+B0[i]\
            -B0[i]*g0[i]*(y0[i]*y0[i+1]+y0[i]*y0[i-1])/2
    return derivs

def ODO(x, a, b, c,d,e):
    return a * np.exp(b * x) + c* np.exp(d*x)+e

def ODOPaper(x,i,j,k):
    Tau = math.sqrt(g0[0] * g0[1])
    return i * np.exp(j *(1-Tau)* x) + k* np.exp(j* (1+Tau)*x)

def ODOSimp(x, a, b, c):
    return a * np.exp(b * x) +c

def MatGen(g0,bounds):
    size= g0.size
    A=np.zeros((size,size))
    for i in range(size):
        for j in range(size):
            if i==j:
                A[i,j]=-1
            elif abs(i-j)==1:
                A[i,j]= np.sqrt(g0[i]*g0[j])/2
    if bounds ==0:
        A[0,size-1]=np.sqrt(g0[0]*g0[size-1])/2
        A[size-1,0]=np.sqrt(g0[0]*g0[size-1])/2
    return A
```

```
def AltMag(T,Mult,timestop,ystart,bounds,field,const):
    """
    T is of the form (T1,T2,,,)
    Mult is number of times temperature pattern repeats
    timestop defines plot range
    ystart = 0 for alternate, 1 for linear spread, 2 for constant 0
    bounds = 0 for periodic, 1 for hard
    field = 0 for off, 1 for constant, 2 for constant by temp
    const = constant for field strength, this value is run through a
       ↪ tanh
    """
    Cyc=len(T)
    N=Cyc*Mult
    #gamma values based on temprature bath
    T0=np.zeros(N)
    g0=np.zeros(N)
    for i in range(Mult):
        for j in range(Cyc):
            T0[i*Cyc+j]=T[j]
            g0[i*Cyc+j]= np.tanh(2/T[j])

    #time array
    timesteps= 10000
    tinc= timestop/ timesteps #this number is the number of steps
    t= np.arange(0,timestop,tinc)
    x_data =t

    #set initial spin conditions
    y0=np.zeros(N)
    ##alternate initial between 1 and -1. Note odd # temperatures will
       ↪ be have nonzero initial system magnetization
    if ystart==0:
        for i in range(Mult):
            for j in range(Cyc):
                y0[i*Cyc+j]= (-1)**(j)
    ##Linearly spread initial state between 1 and -1, T1 being 1.
    if ystart==1:
        for i in range(Mult):
            for j in range(Cyc):
                y0[i*Cyc+j]=1-(2*j/(Cyc-1))
    #constant initial
    if ystart==2:
```

```
    for i in range(N):
        y0[i]=1

##Magnetic field
if field==0:
    b0=np.zeros(N)
if field==1:
    b0=np.zeros(N)+ const
if field==2: #field depending on temperature bath
    b0=np.zeros(N)
    for i in range(Mult):
        for j in range(Cyc):
            b0[i*Cyc+j]= const*(-1)**(j)
B0=np.zeros(N)
B0=np.tanh(b0/T0)
#boundary conditons+ solve system
if bounds==1:
    psoln = odeint(f1, y0, t, args=(N,g0,B0))
if bounds==0:
    psoln = odeint(f0, y0, t, args=(N,g0,B0))
magnetization= np.sum(psoln,axis=1)/N
y_data = magnetization

####Matrix work
#matrix = MatGen(g0,bounds)
#eigVal, eigVec =np.linalg.eig(matrix)
#print("eigenvalues:")
#print(eigVal)
#print("eigenvectors:")
#print(eigVec)

#plot particles of interest
fig2 = plt.figure(2, figsize=(12,9))
ax2 = fig2.add_subplot(111)
#ax1 = fig2.add_subplot(111)
#ax1.plot(t, magnetization, color=c00)
ax2.set_xlabel('Time')
ax2.set_ylabel('Magnetization')
#ax2.set_title("Temperatures {}, Field {}".format(T,const))
ax2.set_title("Temperatures␣{}".format(T))
#ax2.set_title("Temperatures (100, 1, 2/3, 3/2), Field {}".format(
    ↪ const))
```

```
    p1=0
    p2=1
    #p3=2 #Mult
    #p4=3
    #p5=4 #N-1
    ax2.plot(t, magnetization, color=c00, label='Overall')
    ax2.plot(t, psoln[:,p1], color=c8, label='T=␣{}'.format(T[0%len(T)])
       ↪ )
    ax2.plot(t, psoln[:,p2], color=c1, label='T=␣{}'.format(T[1%len(T)])
       ↪ )
    #ax2.plot(t, psoln[:,p3], color=c6, label='T= {}'.format(T[2%len(T)
       ↪ ]))
    #ax2.plot(t, psoln[:,p4], color=c3, label='T= {}'.format(T[3%len(T)
       ↪ ]))
    #ax2.plot(t, psoln[:,p5], color=c5, label='T= {}'.format(T[4%len(T)
       ↪ ]))

    #ax2.plot(t, psoln[:,p5], color=c8)
    # single particle fit
    #popt_2, pcov_2 = curve_fit(ODO, x_data, psoln[:,p1], p0
       ↪ =(1,-2,-1,-1,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_2), color=c01)
    #print(np.round(popt_2,4))
    #popt_3, pcov_3 = curve_fit(ODO, x_data, psoln[:,p2], p0
       ↪ =(1,-2,-1,-1,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_3), color=c01)
    #print(np.round(popt_3,4))
    #popt_4, pcov_4 = curve_fit(ODO, x_data, psoln[:,p3], p0
       ↪ =(1,-1,-1,-2,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_4), color=c01)
    #print(np.round(popt_4,4))
    #popt_5, pcov_5 = curve_fit(ODO, x_data, psoln[:,p4], p0
       ↪ =(1,-1,-1,-2,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_5), color=c01)
    #print(np.round(popt_5,4))
    #fig2.savefig('Alt{}_Sublat.png'.format((T,field,const)))

    ax2.legend()

    #curve fit+plot
# ydata = y_data
# fig=plt.figure(figsize=(12,12))
```

```
# plt.plot(x_data, ydata, color=c00, label='data')
# popt, pcov = curve_fit(ODO, x_data, ydata, p0=(-1,-2,1,-1,0), maxfev
    ↪ =10000)
# #popt2, pcov2 = curve_fit(ODOPaper, xdata, ydata, p0=(-1,-1,1),
    ↪ maxfev=10000)
# fitCon=np.round(popt,4)
# #print(fitCon)
# #fitCon2=np.round(popt2,4)
# #plt.plot(x_data, ODO(x_data, *popt), color=c01)
# #plt.plot(xdata, ODOPaper(xdata, *popt2), 'y-')
# plt.xlabel('Time')
# plt.ylabel('Magnetization')
# plt.title((T,field,const), loc='left')
# plt.title(fitCon,loc='right')
    #plt.show()
    #fig.savefig('Alt{}.png'.format((T,field,const)))
    #return popt


def BloMag2(T,N,timestop,ystart,bounds,field,const):
    """
    T is of the form (T1,T2)
    N is the total system size
    timestop defines plot range
    ystart = 1 for linear spread, 2 for constant 0
    bounds = 0 for periodic, 1 for hard
    field = 0 for off, 1 for constant, 2 for opposite fields by block
    const = constant for field strength, this value is run through a
       ↪ tanh
    """
    #Cutoff value
    Cutoff=N/2
    #gamma values based on temprature bath
    T0=np.zeros(N)
    g0=np.zeros(N)
    for i in range(N):
        if i<Cutoff:
            T0[i]=T[0]
        elif i>=Cutoff:
            T0[i]=T[1]
        g0[i]= np.tanh(2/T0[i])
    #time array
```

```
timesteps= 10000
tinc= timestop/ timesteps #this number is the number of steps
t= np.arange(0,timestop,tinc)
x_data =t

#set initial spin conditions
y0=np.zeros(N)
##alternate initial between 1 and -1. Note odd # temperatures will
   ↪ be weird
#if ystart==0:
 # for i in range(Mult):
  # for j in range(Cyc):
   # y0[i*Cyc+j]= (-1)**(j)
##Linearly spread initial state between 1 and -1
for i in range(N):
    if i < Cutoff:
        y0[i]= 1
    if i >= Cutoff:
        y0[i] = -1
#constant initial
if ystart==2:
    for i in range(N):
        y0[i]=1

##Magnetic field
if field==0:
    b0=np.zeros(N)
if field==1:
    b0=np.zeros(N)+ const
if field==2: #field depending on temperature bath
    b0=np.zeros(N)
    for i in range(N):
        if i < Cutoff:
            b0[i]= const
        if i >= Cutoff:
            b0[i] = -const
B0=np.zeros(N)
B0=np.tanh(b0/T0)
#boundary conditons+ solve system
if bounds==1:
    psoln = odeint(f1, y0, t, args=(N,g0,B0))
if bounds==0:
```

```
    psoln = odeint(f0, y0, t, args=(N,g0,B0))
magnetization= np.sum(psoln,axis=1)/N
y_data = magnetization

####Matrix work
#matrix = MatGen(g0,bounds)
#eigVal, eigVec =np.linalg.eig(matrix)
#print("eigenvalues:")
#print(eigVal)
#print("eigenvectors:")
#print(eigVec)

#plot results
fig2 = plt.figure(2, figsize=(12,9))
ax2 = fig2.add_subplot(111)
ax2.set_xlabel('Time')
ax2.set_ylabel('Magnetization')
ax2.set_title("Temperatures␣{},␣{}␣Particles".format(T,N))
#ax2.set_title("Temperatures {}, Field {}".format(T,const))
#ax2.plot(t, magnetization, color=c3, label="N= 10")
ax2.plot(t, magnetization, color=c00, label="Overall")

#particles of interest
#p0=0
#p1=1
#p2=2
#p3=3
#ax2.plot(t, psoln[:,p0], color=c9)
#ax2.plot(t, psoln[:,p1], color=c6)
#ax2.plot(t, psoln[:,p2], color=c3)
#ax2.plot(t, psoln[:,p3], color=c0)

#sublattices
magnetization_1= np.sum(psoln[:,:int(Cutoff)],axis=1)/(N/2)
magnetization_2= np.sum(psoln[:,int(Cutoff):],axis=1)/(N/2)
ax2.plot(t, magnetization_1, color=c8, label='T=␣{}'.format(T[0]))
ax2.plot(t, magnetization_2, color=c1, label='T=␣{}'.format(T[1]))


#popt_2, pcov_2 = curve_fit(ODOSimp, x_data, magnetization_1, p0
   ↪ =(1,-1,0), maxfev=100000)
#ax2.plot(x_data, ODOSimp(x_data, *popt_2), color=c01)
```

```
    #print(np.round(popt_2,4))
    #print(np.round(popt_2[1]- (np.tanh(2/T[0])-1),4))
    #popt_3, pcov_3 = curve_fit(ODOSimp, x_data, magnetization_2, p0
        ↪ =(-1,-1,0), maxfev=100000)
    #ax2.plot(x_data, ODOSimp(x_data, *popt_3), color=c01)
    #print(np.round(popt_3,4))
    #print(np.round(popt_3[1]- (np.tanh(2/T[1])-1),4))
    ax2.legend()


# #curve fit+plot
# xdata = x_data
# y = y_data
# ydata = y
# fig=plt.figure(figsize=(12,12))
# plt.plot(xdata, ydata, color=c00, label='data')
# popt, pcov = curve_fit(ODO, xdata, ydata, p0=(-1,-2,1,-1,0), maxfev
    ↪ =10000)
# #popt2, pcov2 = curve_fit(ODOPaper, xdata, ydata, p0=(-1,-1,1),
    ↪ maxfev=10000)
# #print(T,N)
# fitCon=np.round(popt,4)
# #fitCon2=np.round(popt2,4)
# plt.plot(xdata, ODO(xdata, *popt), color=c01)
# #plt.plot(xdata, ODOPaper(xdata, *popt2), 'y-')
# plt.xlabel('x')
# plt.ylabel('y')
# plt.title((T,field,const), loc='left')
# plt.title(fitCon,loc='right')
# #plt.show()
# fig.savefig('Blo2{}.png'.format((T,field,const)))

    #return magnetization



def BloMag3(T,N,timestop,ystart,bounds,field,const):
    """
    T is of the form (T1,T2,T3)
    N is the total system size
    timestop defines plot range
    ystart = 1 for linear spread, 2 for constant, 0 alternating
```

```
    bounds = 0 for periodic, 1 for hard
    field = 0 for off, 1 for constant, 2 for fields by block
    const = constant for field strength, this value is run through a
       ↪ tanh
    """
    #Cutoff value
    Cutoff1=N/3
    Cutoff2=2*N/3
    #gamma values based on temprature bath
    T0=np.zeros(N)
    g0=np.zeros(N)
    for i in range(N):
        if i<Cutoff1:
            T0[i]=T[0]
        elif i<Cutoff2:
            T0[i]=T[1]
        else:
            T0[i]=T[2]
        g0[i]= np.tanh(2/T0[i])
    #time array
    timesteps= 10000
    tinc= timestop/ timesteps #this number is the number of steps
    t= np.arange(0,timestop,tinc)
    x_data =t


    #set initial spin conditions
    y0=np.zeros(N)
    ##alternate initial between 1 and -1. Note odd # temperatures will
       ↪ be weird
    if ystart==0:
        for i in range(N):
            if i < Cutoff1:
                y0[i]= 1
            elif i < Cutoff2:
                y0[i] = -1
            else:
                y0[i] = 1
    ##Linearly spread initial state between 1 and -1
    if ystart==1:
        for i in range(N):
            if i < Cutoff1:
```

```
            y0[i]= 1
        elif i < Cutoff2:
            y0[i] = 0
        else:
            y0[i] = -1
#constant initial
if ystart==2:
    for i in range(N):
        y0[i]=0

##Magnetic field
if field==0:
    b0=np.zeros(N)
if field==1:
    b0=np.zeros(N)+ const
if field==2: #field depending on temperature bath
    b0=np.zeros(N)
    for i in range(N):
        if i < Cutoff1:
            b0[i]= const
        elif i < Cutoff2:
            b0[i] = 0
        else:
            b0[i] = -const
B0=np.zeros(N)
B0=np.tanh(b0/T0)
#boundary conditons+ solve system
if bounds==1:
    psoln = odeint(f1, y0, t, args=(N,g0,B0))
if bounds==0:
    psoln = odeint(f0, y0, t, args=(N,g0,B0))
magnetization= np.sum(psoln,axis=1)/N
y_data = magnetization

####Matrix work
#matrix = MatGen(g0,bounds)
#eigVal, eigVec =np.linalg.eig(matrix)
#print("eigenvalues:")
#print(eigVal)
#print("eigenvectors:")
#print(eigVec)
```

```
#plot just results
#fig = plt.figure(1, figsize=(10,10))
#ax1 = fig.add_subplot(111)
#ax1.plot(t, magnetization, color=c00)
#ax1.set_xlabel('time')
#ax1.set_ylabel('magnetization')
##particles of interest
#p1=0
#ax2 = fig.add_subplot(212)
#ax2.plot(t, psoln[:,p1], color=c0)

#particle(s) of interest
fig2 = plt.figure(2, figsize=(12,9))
ax2 = fig2.add_subplot(111)
#ax1 = fig2.add_subplot(111)
#ax1.plot(t, magnetization, color=c00)
ax2.set_xlabel('Time')
ax2.set_ylabel('Magnetization')
#ax2.set_title("Temperatures {}, Field {}".format(T,const))
ax2.set_title("Temperatures␣{}".format(T))
ax2.plot(t, magnetization, color=c00, label='Overall')

#p1=0 #round(N*1/6)
#p2=1 #round(N*3/6)
#p3=3 #round(N*5/6)
#p4=3
#p5=4
#ax2.plot(t, psoln[:,p4], color=c3)
#ax2.plot(t, psoln[:,p5], color=c8)

#sublattices
magnetization_1= np.sum(psoln[:,:int(Cutoff1)],axis=1)/(N/3)
magnetization_2= np.sum(psoln[:,int(Cutoff1):int(Cutoff2)],axis=1)/(
    ↪ N/3)
magnetization_3= np.sum(psoln[:,int(Cutoff2):],axis=1)/(N/3)
ax2.plot(t, magnetization_1, color=c9, label='T=␣{}'.format(T[0]))
ax2.plot(t, magnetization_2, color=c1, label='T=␣{}'.format(T[1]))
ax2.plot(t, magnetization_3, color=c6, label='T=␣{}'.format(T[2]))

# sublattice fit
```

```
    #popt_2, pcov_2 = curve_fit(ODO, x_data, magnetization_1, p0
       ↪ =(1,-2,-1,-1,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_2), color=c01)
    #print(np.round(popt_2,4))
    #popt_3, pcov_3 = curve_fit(ODO, x_data, magnetization_2, p0
       ↪ =(1,-1,-1,-5,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_3), color=c01)
    #print(np.round(popt_3,4))
    #popt_4, pcov_4 = curve_fit(ODO, x_data, magnetization_3, p0
       ↪ =(1,-2,-1,-1,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_4), color=c01)
    #print(np.round(popt_4,4))

    #particle fits
    #popt_5, pcov_5 = curve_fit(ODO, x_data, psoln[:,p4], p0
       ↪ =(1,-1,-1,-2,0), maxfev=100000)
    #ax2.plot(x_data, ODO(x_data, *popt_5), color=c01)
    #print(np.round(popt_5,4))
    #fig2.savefig('Blo3{}_Sublat.png'.format((T,field,const)))

    ax2.legend()

# #curve fit+plot
# xdata = x_data
# y = y_data
# ydata = y
# fig=plt.figure(figsize=(12,12))
# plt.plot(xdata, ydata, color=c00, label='data')
# popt, pcov = curve_fit(ODO, xdata, ydata, p0=(1,-2,-1,-1,0.3),
   ↪ maxfev=100000)
# #popt99, pcov99 = curve_fit(ODOPaper, xdata, ydata, p0=(-1,-1,1),
   ↪ maxfev=10000)
# #print(T,N)
# fitCon=np.round(popt,4)
# #fitCon2=np.round(popt99,4)
# plt.plot(xdata, ODO(xdata, *popt), color=c01)
# #plt.plot(xdata, ODOPaper(xdata, *popt2), 'y-')
# plt.xlabel('x')
# plt.ylabel('y')
# plt.title((T,field,const), loc='left')
# plt.title(fitCon,loc='right')
    #plt.show()
```

```
    #fig.savefig('Blo3{}.png'.format((T,field,const)))
    #return popt
```

Appendix B. Mathematica Derivation of One-Dimensional Steady States

1D Blocks

Large N limit where edges don't matter
With mean field!!!! $<\sigma\ \sigma> = <\sigma><\sigma>$ is an approximation

Clear[B, σ]

GRules = $\{\beta \to$ Tanh $\left[\frac{B}{T}\right], \gamma \to$ Tanh $\left[\frac{2}{T}\right]\}$

$\{\beta \to \text{Tanh}\left[\frac{B}{T}\right], \gamma \to \text{Tanh}\left[\frac{2}{T}\right]\}$

DerivB = $-\sigma + \frac{\gamma}{2}(\sigma + \sigma) + \beta - \beta\frac{\gamma}{2}(\sigma\sigma + \sigma\sigma)$

$\beta - \sigma + \gamma\sigma - \beta\gamma\sigma^2$

{TrashB, EqiB} = Solve[DerivB == 0, σ]/.GRules//Simplify;

EqiB

$$\left\{\sigma \to \frac{1}{2}\text{Coth}\left[\frac{B}{T}\right]\left(1 + \text{Coth}\left[\frac{2}{T}\right]\left(-1 + \sqrt{1 + \text{Tanh}\left[\frac{2}{T}\right]^2 + \text{Tanh}\left[\frac{2}{T}\right]\left(-2 + 4\text{Tanh}\left[\frac{B}{T}\right]^2\right)}\right)\right)\right\}$$

{Block1} = EqiB/.{$B \to -0.007, T \to 1$}//N

{Block2} = EqiB/.{$B \to -0.007, T \to 5$}//N

$\{\sigma \to -0.187963\}$

$\{\sigma \to -0.00225787\}$

((σ/.Block1) + (σ/.Block2))/2

-0.0951103

1D Alt

Clear[$\sigma 1, \sigma 2$]

$\mathbf{GRules = \left\{\beta \to \text{Tanh}\left[\frac{B}{T}\right], \gamma \to \text{Tanh}\left[\frac{2}{T}\right]\right\}}$

$\left\{\beta \to \text{Tanh}\left[\frac{B}{T}\right], \gamma \to \text{Tanh}\left[\frac{2}{T}\right]\right\}$

$\mathbf{Temps = \{10, 2, 5\}}$

$\mathbf{\sigma Alt = \{\sigma 1, \sigma 2, \sigma 3\}}$

$\{10, 2, 5\}$

$\{\sigma 1, \sigma 2, \sigma 3\}$

Copy +Paste +Edit this:

$\mathbf{0 == -\sigma Alt[[1]] + \frac{\gamma}{2}(\sigma Alt[[3]] + \sigma Alt[[2]]) + \beta -}$

$\mathbf{\beta\frac{\gamma}{2}(\sigma Alt[[1]] * \sigma Alt[[3]] + \sigma Alt[[1]] * \sigma Alt[[2]])/.GRules/.T \to Temps[[1]];}$

Solve:

$\mathbf{B = 1;}$

$\mathbf{\{Trash1, Trash2, Trash3, Trash4, EqiAlt\} =}$

$\mathbf{NSolve[}$

$\mathbf{\{0 == -\sigma Alt[[1]] + \frac{\gamma}{2}(\sigma Alt[[3]] + \sigma Alt[[2]]) + \beta -}$

$\mathbf{\beta\frac{\gamma}{2}(\sigma Alt[[1]] * \sigma Alt[[3]] + \sigma Alt[[1]] * \sigma Alt[[2]])/.GRules/.T \to Temps[[1]],}$

$\mathbf{0 == -\sigma Alt[[2]] + \frac{\gamma}{2}(\sigma Alt[[1]] + \sigma Alt[[3]]) + \beta -}$

$\mathbf{\beta\frac{\gamma}{2}(\sigma Alt[[2]] * \sigma Alt[[1]] + \sigma Alt[[2]] * \sigma Alt[[3]])/.GRules/.T \to Temps[[2]],}$

$\mathbf{0 == -\sigma Alt[[3]] + \frac{\gamma}{2}(\sigma Alt[[2]] + \sigma Alt[[1]]) + \beta -}$

$\mathbf{\beta\frac{\gamma}{2}(\sigma Alt[[3]] * \sigma Alt[[2]] + \sigma Alt[[3]] * \sigma Alt[[1]])/.GRules/.T \to Temps[[3]]\},}$

$\mathbf{\{\sigma 1, \sigma 2, \sigma 3\}]}$

$\{\{\sigma 1 \to -13.103, \sigma 2 \to -65.5793, \sigma 3 \to 7.56309\}, \{\sigma 1 \to -27.3474, \sigma 2 \to 2.26206, \sigma 3 \to -76.912\}, \{\sigma 1 \to 48.8296, \sigma 2 \to -73.3147, \sigma 3 \to -54.3842\}, \{\sigma 1 \to -1.4215, \sigma 2 \to -10.0329, \sigma 3 \to -3.46828\}, \{\sigma 1 \to 0.191276, \sigma 2 \to 0.607355, \sigma 3 \to$

0.338945}}

Average sublattices:

$$(\sigma\text{Alt}[[1]] + \sigma\text{Alt}[[2]] + \sigma\text{Alt}[[3]])/3/.\text{EqiAlt}$$

0.379192

2 temp

$$\text{Temps} = \{1, 5\}$$

$$\sigma\text{Alt} = \{\sigma 1, \sigma 2\};$$

$\{1, 5\}$

$$B = 0.007;$$

$$\{\text{Trash1}, \text{EqiAlt}\} =$$

$$\text{NSolve}[$$

$$\{0 == -\sigma\text{Alt}[[1]] + \tfrac{\gamma}{2}(\sigma\text{Alt}[[2]] + \sigma\text{Alt}[[2]]) + \beta -$$

$$\beta\tfrac{\gamma}{2}(\sigma\text{Alt}[[1]] * \sigma\text{Alt}[[2]] + \sigma\text{Alt}[[1]] * \sigma\text{Alt}[[2]])/.\text{GRules}/.T \to \text{Temps}[[1]],$$

$$0 == -\sigma\text{Alt}[[2]] + \tfrac{\gamma}{2}(\sigma\text{Alt}[[1]] + \sigma\text{Alt}[[1]]) + \beta -$$

$$\beta\tfrac{\gamma}{2}(\sigma\text{Alt}[[2]] * \sigma\text{Alt}[[1]] + \sigma\text{Alt}[[2]] * \sigma\text{Alt}[[1]])/.\text{GRules}/.T \to \text{Temps}[[2]]\},$$

$$\{\sigma 1, \sigma 2\}]$$

$\{\{\sigma 1 \to -204.714, \sigma 2 \to -87.2842\}, \{\sigma 1 \to 0.0131745, \sigma 2 \to 0.00640558\}\}$

Average sublattices

$$(\sigma\text{Alt}[[1]] + \sigma\text{Alt}[[2]])/2/.\text{EqiAlt}$$

0.00979003

Appendix C. Python Monte Carlo on One-Dimensional Lattices

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 25 08:05:28 2020
@author: LaurentiuStoleriu

Last Modified 4/23/2021
Modified by Sho Gibbs for for the one dimensional chain
"""
import random
import numpy as np

### Averaging in console

#plots = []
#for i in range(500):
# runfile('C:/Users/shogi/Desktop/Honors Thesis/Cayley1D.py', wdir='C
    ↪ :/Users/shogi/Desktop/Honors Thesis')
# plots.append(M)
#
#plotting = []
#for i in range(len(plots[0])):
# avg = 0
# for j in range(len(plots)):
# avg += plots[j][i]
# avg = avg / NTot
# plotting.append(avg/len(plots))
#
##plot results
#fig2 = plt.figure(2, figsize=(10,10))
#ax2 = fig2.add_subplot(111)
#ax2.set_xlabel('time')
#ax2.set_ylabel('magnetization')
#ax2.set_title("test")
##ax2.set_title("Temperatures {}, Field {}".format(T,const))
#ax2.plot(list(range(len(plotting))), plotting, color='k')

NGen = 10 # number of generations (> 1), 'first' gen = 0
NNeigh = 2 # Do not change for 1D!!!!

k = 1.0 # Boltzmann constant
```

```
T = (1.0, 5.0) # Temperature
J = 1.0 # Coupling constant

Duration = 1000

NTot = 1 + NNeigh * NGen

system = np.zeros(NTot, dtype=int) # the system
neighs = np.full((NTot, NNeigh), -1, dtype=int) # neghbors indexes (
    ↪ fill with -1)
neighCount = np.zeros(NTot, dtype=int) # counting number of neighbors
sys_temp = np.ones(NTot, dtype=float) # spins temperatures

M = np.zeros(Duration, dtype=float)
Corr = np.zeros((Duration,NTot,NNeigh),dtype=float)

### START Generating
# [-1,1][random.randrange(2)] it's just a shortcut to random -1 and +1
# neighbors strategy:
# for generation 0 we're defining forward neighbors (neigh[0, X]) and
    ↪ backward neighbors (neigh[X, 0] = 0)
# for intermediate generations same as generation 0
# for leafs (last generation) no need to bother about neighbors (
    ↪ missing ones are == -1)

# generation 0
system[0] = [-1,1][random.randrange(2)]
for n in range(NNeigh):
    neighs[0, n] = n+1
    neighs[n+1, 0] = 0
edge_count = NNeigh

# NEXT GENERATION
for i in range(1,NGen):
    NTotUntilNow = 1 + NNeigh * (i-1)
    NThisGen = NNeigh * (NNeigh-1)**(i-1)
    NNextGen = NTotUntilNow + NThisGen
    for j in range( NThisGen ):
        #system[NTotUntilNow + j] = [-1,1][random.randrange(2)] #set
            ↪ initial spin randomly?
        for n in range(NNeigh-1):
            neighs[NTotUntilNow + j, n+1] = NNextGen + j*(NNeigh-1) + n
```

```
            neighs[NNextGen + j*(NNeigh-1) + n , 0] = NTotUntilNow + j
        edge_count = edge_count + (NNeigh-1)

# last generation
NTotUntilNow = NTot - NNeigh
NThisGen = NNeigh * (NNeigh-1)**(NGen-1)
for j in range( NThisGen ):
    system[NTotUntilNow + j] = [-1,1][random.randrange(2)]

#getting neighbor count
for particle in range(NTot):
    for n in range(NNeigh):
        if neighs[particle, n] >= 0:
            neighCount[particle] += 1

# assign temerature to spins
#alternating temps
for i in range(NTot):
    sys_temp[i] = T[((i+1)//2)%len(T)]
    system[i] = (-1)**((i+1)//2)
##block temps
#for i in range(NTot):
# sys_temp[i] = T[i%len(T)]
# system[i] = (-1)**i
### DONE Generating



### DISCOVERY (time stepping)
overall_time_step_count = 0

for t in range(Duration):
    overall_time_step_count = overall_time_step_count + 1
    particle = random.randrange(NTot)
    beta = 1.0 / k / sys_temp[particle]
    neigh_sum = 0
    for n in range(1, NNeigh):
        if neighs[particle, n] >= 0:
            neigh_sum = neigh_sum + system[neighs[particle, n]]

    w = 0.5 * (1.0 - system[particle] * np.tanh(beta*J*neigh_sum) /
       ↪ neighCount[particle]) #double check
```

```
    if ( w > random.random()):
        system[particle] = (-1) * system[particle]

    if ((t % (Duration//10)) == 0):
        print(f"MC step {t} of {Duration}")

    M[t] = np.sum(system)

    for i in range(NTot):
        for j in range(NNeigh):
            if(neighs[i, j] > i): #forward links only
                Corr[t,i,j] = system[i]*system[neighs[i,j]]
### DONE Discovering

### Export Data? (M and Corr)




### LOWER DECKS (plotting)
#import networkx as nx
#import matplotlib.pyplot as plt
#
#fig, [ax1, ax2] = plt.subplots(nrows=1, ncols=2)
#
#if True:
# ax1.plot(range(Duration), M/NTot, color = "indianred")
# #ax1.set_title("Magnetization")
# ax1.set_xlabel("MC timestep")
# ax1.set_ylabel("Magnetization")


##if False:
# G = nx.Graph()
# G.add_nodes_from(range(NTot))
#
# edges = np.empty(edge_count, dtype=object)
# count = 0
# for i in range(NTot):
# for j in range(NNeigh):
# if(neighs[i, j] > i): #forward links only
```

```
# edges[count] = (i, neighs[i, j])
# count = count + 1
#
# G.add_edges_from(edges)
#
# radius = 1
# pos = {} #nice positions on concentric circles
# pos[0] = (0, 0)
# count = 1
# for i in range(1,NGen+1):
# NThisGen = NNeigh * (NNeigh-1)**(i-1)
# for j in range( NThisGen ):
# pos[count] = (radius*i*np.cos(j*2.0*np.pi/NThisGen + np.pi/NThisGen)
   ↪ , radius*i*np.sin(j*2.0*np.pi/NThisGen + np.pi/NThisGen))
# count = count + 1
#
# nx.draw(G, pos, ax=ax2, with_labels=False, font_weight='bold',
   ↪ node_color=system, node_size=100, linewidths=0, width =0.4, cmap
   ↪ =plt.cm.coolwarm)
# plt.axis("equal")
# plt.show()
```

Appendix D. Python Monte Carlo on Cayley Trees

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 25 08:05:28 2020
@author: LaurentiuStoleriu
Modified by Sho Gibbs May 2021

Assigns initial spins alternating by generation or randomly
Assigns temperatures alternating by generation or randomly
change the two above settings by commenting/ uncommenting lines in
    ↪ generation

Average multiple simulations using commented text in beginning (lines
    ↪ 20-91)
Need to modify the runfile/ working directory paths in line 23
Want to comment out this file's plotting when doing averages.
"""
import random
import numpy as np

### Averaging in console (max 6 gens)
#
#from datetime import datetime
#startTime = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
#plots = []
#for run in range(1,501):
# runfile('C:/Users/shogi/Desktop/Honors Thesis/Cayley.py', wdir='C:/
    ↪ Users/shogi/Desktop/Honors Thesis')
# plots.append([M,Msub])
# if ((run % 100) == 0):
# print("Run", run)
#plotting = []
#leafplotting = []
##restplotting = []
#genplotting0 = []
#genplotting1 = []
#genplotting2 = []
#genplotting3 = []
#genplotting4 = []
#genplotting5 = []
#for i in range(len(plots[0][0])):
```

```
# avg = 0
# leafavg = 0
## restavg = 0
# genavg0 = 0
# genavg1 = 0
# genavg2 = 0
# genavg3 = 0
# genavg4 = 0
# genavg5 = 0
# for j in range(len(plots)):
# avg += plots[j][0][i]
# leafavg += plots[j][1][i][-1]
## restavg += np.sum(plots[j][1][i][:-1])
# genavg0 += np.sum(plots[j][1][i][0])
# genavg1 += np.sum(plots[j][1][i][1])
# genavg2 += np.sum(plots[j][1][i][2])
# genavg3 += np.sum(plots[j][1][i][3])
# genavg4 += np.sum(plots[j][1][i][4])
# genavg5 += np.sum(plots[j][1][i][5])
# avg = avg / NTot
# leafavg = leafavg / (NTot - genCutOffs[-1])
## restavg = restavg / (genCutOffs[-1])
# genavg0 = genavg0 / (genCutOffs[1] - genCutOffs[0])
# genavg1 = genavg1 / (genCutOffs[2] - genCutOffs[1])
# genavg2 = genavg2 / (genCutOffs[3] - genCutOffs[2])
# genavg3 = genavg3 / (genCutOffs[4] - genCutOffs[3])
# genavg4 = genavg4 / (genCutOffs[5] - genCutOffs[4])
# genavg5 = genavg5 / (genCutOffs[6] - genCutOffs[5])
# plotting.append(avg/len(plots))
# leafplotting.append(leafavg/len(plots))
## restplotting.append(restavg/len(plots))
# genplotting0.append(genavg0/len(plots))
# genplotting1.append(genavg1/len(plots))
# genplotting2.append(genavg2/len(plots))
# genplotting3.append(genavg3/len(plots))
# genplotting4.append(genavg4/len(plots))
# genplotting5.append(genavg5/len(plots))
#fig2 = plt.figure(2, figsize=(10,10))
#ax2 = fig2.add_subplot(111)
#ax2.set_xlabel('Timesteps')
#ax2.set_ylabel('Magnetization')
```

```
#ax2.set_title("{} Generations, z={}, T={}, H={}, {} Runs".format(NGen
    ↪ , NNeigh, T, H, run))
#ax2.plot(list(range(len(plotting))), plotting, color='k', label='
    ↪ Overall')
##ax2.plot(list(range(len(restplotting))), restplotting, color='y',
    ↪ label='interior')
#ax2.plot(list(range(len(genplotting0))), genplotting0, color='r',
    ↪ label='0th')
#ax2.plot(list(range(len(genplotting1))), genplotting1, color='b',
    ↪ label='1st')
#ax2.plot(list(range(len(genplotting2))), genplotting2, color='m',
    ↪ label='2nd')
#ax2.plot(list(range(len(genplotting3))), genplotting3, color='c',
    ↪ label='3rd')
#ax2.plot(list(range(len(genplotting4))), genplotting4, color='tab:
    ↪ pink', label='4th')
#ax2.plot(list(range(len(genplotting5))), genplotting5, color='tab:
    ↪ purple', label='5th')
#ax2.plot(list(range(len(leafplotting))), leafplotting, color='g',
    ↪ label='Leaf')
#ax2.legend()
#print(startTime)
#print(datetime.now().strftime("%d/%m/%Y %H:%M:%S"))

NGen = 4 # number of generations (> 1), first gen = 0
NNeigh = 3 # number of neighbors (>2 )

k = 1.0 # Boltzmann constant
T = (6.0, 10.0, 2.0) # Temperatures
J = 1.0 # Coupling constant
H = -1.0 # constant field strength

Duration = 500

NTot = 1 + NNeigh * (1 - (NNeigh - 1) ** NGen) // (1 - (NNeigh - 1))

system = np.zeros(NTot, dtype=int) # the system
neighs = np.full((NTot, NNeigh), -1, dtype=int) # neghbors indexes (
    ↪ fill with -1)
sys_temp = np.ones(NTot, dtype=float) # spins temperatures
neighCount = np.zeros(NTot, dtype=int) # counting number of neighbors
genCutOffs = [] # what particles are the first of a new gen
```

```
### START Generating
# [-1,1][random.randrange(2)] it's just a shortcut to random -1 and +1
# neighbors strategy:
# for generation 0 we're defining forward neighbors (neigh[0, X]) and
    ↪ backward neighbors (neigh[X, 0] = 0)
# for intermediate generations same as generation 0
# for leafs (last generation) no need to bother about neighbors (
    ↪ missing ones are == -1)

# generation 0
#system[0] = [-1,1][random.randrange(2)] #random initial
system[0] = 1 #inital by alteranting
#sys_temp[0] = T[random.randrange(len(T))] # random temp
sys_temp[0] = T[0] #temp by gen
for n in range(NNeigh):
    neighs[0, n] = n+1
    neighs[n+1, 0] = 0
edge_count = NNeigh
genCutOffs.append(0)
genCutOffs.append(1)

# NEXT GENERATION
for i in range(1,NGen):
    NTotUntilNow = 1 + NNeigh * (1 - (NNeigh - 1) ** (i-1)) // (1 - (
        ↪ NNeigh - 1))
    NThisGen = NNeigh * (NNeigh-1)**(i-1)
    NNextGen = NTotUntilNow + NThisGen
    for j in range( NThisGen ):
        #system[NTotUntilNow + j] = [-1,1][random.randrange(2)] #random
            ↪  inital
        system[NTotUntilNow + j] = (-1)**i #alternating initial
        #sys_temp[NTotUntilNow + j] = T[random.randrange(len(T))] #
            ↪ random temps
        sys_temp[NTotUntilNow + j] = T[(i)%len(T)] # alternating temps

        for n in range(NNeigh-1):
            neighs[NTotUntilNow + j, n+1] = NNextGen + j*(NNeigh-1) + n
            neighs[NNextGen + j*(NNeigh-1) + n , 0] = NTotUntilNow + j
        edge_count = edge_count + (NNeigh-1)
```

```
    genCutOffs.append(NNextGen)

# last generation
NTotUntilNow = 1 + NNeigh * (1 - (NNeigh - 1) ** (NGen-1)) // (1 - (
    ↪ NNeigh - 1))
NThisGen = NNeigh * (NNeigh-1)**(NGen-1)
for j in range( NThisGen ):
    #system[NTotUntilNow + j] = [-1,1][random.randrange(2)] #random
        ↪ inital
    system[NTotUntilNow + j] = (-1)**NGen #alternating initial
    #sys_temp[NTotUntilNow + j] = T[random.randrange(len(T))] #random
        ↪ temps
    sys_temp[NTotUntilNow + j] = T[(NGen)%len(T)] # alternating temps


#getting neighbor count
for particle in range(NTot):
    for n in range(NNeigh):
        if neighs[particle, n] >= 0:
            neighCount[particle] += 1

##flip all spins initially
#for particle in range(NTot):
# system[particle] = - system[particle]
### DONE Generating




### DISCOVERY (time stepping)
overall_time_step_count = 0

#overall magnetization
M = np.zeros(Duration, dtype=int)
#correlations
Corr = np.zeros((Duration,genCutOffs[-1],NNeigh),dtype=int)
#sublattices
Msub = np.zeros((Duration,len(genCutOffs)), dtype=int)

for t in range(Duration):
    overall_time_step_count = overall_time_step_count + 1
    particle = random.randrange(NTot)
```

```
    beta = 1.0 / k / sys_temp[particle]
    neigh_sum = system[neighs[particle, 0]]
    field_eff = H / sys_temp[particle]

    M[t] = np.sum(system)

    for subGen in range(len(genCutOffs)-1):
        Msub[t, subGen] = np.sum(system[genCutOffs[subGen]:genCutOffs[
            ↪ subGen+1]])
    Msub[t, len(genCutOffs)-1] = np.sum(system[genCutOffs[-1]:])

    for n in range(NNeigh):
        if neighs[particle, n] >= 0:
            neigh_sum = neigh_sum + system[neighs[particle, n]]

    w = 0.5 * (1.0 - system[particle] * np.tanh(beta*J*neigh_sum)/
        ↪ neighCount[particle]) * (1 - np.tanh(field_eff)*system[
        ↪ particle]) #

    if ( w > random.random()):
        system[particle] = (-1) * system[particle]

# if ((t % 1000) == 0):
# print(f"MC step {t} of {Duration}")



# for i in range(genCutOffs[-1]):
# for j in range(NNeigh):
# if(neighs[i, j] > i): #forward links only
# Corr[t,i,j] = system[i]*system[neighs[i,j]]
### DONE Discovering

### Export Data? (M, Msub, and Corr)




### LOWER DECKS (plotting)
import networkx as nx
import matplotlib.pyplot as plt
```

```
#fig, [ax1, ax2] = plt.subplots(nrows=1, ncols=2)
#
#if True:
# ax1.plot(range(Duration), M/NTot, color = "k", label='Overall')
# #ax1.set_title("Magnetization")
# ax1.set_xlabel("MC timestep")
# ax1.set_ylabel("Magnetization")
# ax1.set_title("{} Generations, z={}, T={}".format(NGen, NNeigh, T))
# ax1.legend()

##if False:
# G = nx.Graph()
# G.add_nodes_from(range(NTot))
#
# edges = np.empty(edge_count, dtype=object)
# count = 0
# for i in range(NTot):
# for j in range(NNeigh):
# if(neighs[i, j] > i): #forward links only
# edges[count] = (i, neighs[i, j])
# count = count + 1
#
# G.add_edges_from(edges)
#
# radius = 1
# pos = {} #nice positions on concentric circles
# pos[0] = (0, 0)
# count = 1
# for i in range(1,NGen+1):
# NThisGen = NNeigh * (NNeigh-1)**(i-1)
# for j in range( NThisGen ):
# pos[count] = (radius*i*np.cos(j*2.0*np.pi/NThisGen + np.pi/NThisGen)
   ↪ , radius*i*np.sin(j*2.0*np.pi/NThisGen + np.pi/NThisGen))
# count = count + 1
#
# nx.draw(G, pos, ax=ax2, with_labels=False, font_weight='bold',
   ↪ node_color=system, node_size=100, linewidths=0, width =0.4, cmap
   ↪ =plt.cm.coolwarm)
# plt.axis("equal")
# plt.show()
```