APPLICATIONS OF ISING AND OTHER DISCRETE SPIN MODELS

by

Yan Naung Oak

A senior thesis submitted to the faculty of

Middlebury College

in partial fulfillment of the requirements for the degree of

Bachelor of Arts

Department of Physics

Middlebury College

May 2009

MIDDLEBURY COLLEGE

DEPARTMENT APPROVAL

of a senior thesis submitted by

Yan Naung Oak

Date	Jeffery Dunham, Advisor
Date	, Comittee Member
Date	,

ABSTRACT

APPLICATIONS OF ISING AND OTHER DISCRETE SPIN MODELS

Yan Naung Oak

Department of Physics

Bachelor of Arts

The Ising model from statistical mechanics has found applications not only in physics and also in diverse fields ranging from biology to the social sciences. Analytical methods to solving the one-dimensional model and approximation techniques such as mean field and renormalization group theory are discussed. Monte Carlo simulations of the Ising model are also carried out. The literature concerning three related models from the social sciences: Schelling's segregation model, Sznajd's model opinion dynamics and Durlauf and Blume's social interactions model are reviewed.

ACKNOWLEDGMENTS

I would like to thank Professor Dunham for inspiring me throughout my Middlebury career. This thesis would not have been possible if it weren't for his pateince in explaining difficult concepts from a wide range of disciplines. I would also like to thank the rest of the physics faculty and my fellow physics majors for four wonderful years of experiences and many beneficial lessons. I would also like to thank Professors Peter Matthews and David Colander from the economics department for introducing me to the incredible world of complexity in the social sciences. Last, but not least, I would like to thank my family and friends for their love and support for me over the years.

Contents

List of Figures

Chapter 1

Introduction

1.1 The Ising model: a brief history

The layman's most familiar conception of magnetism comes from the common ferromagnet. Usually a magnetized bar of iron or some transition metal such as Cobalt or Nickel, it is ubiquitous in everyday life, from refrigerator magnets to compass needles. We learn in elementary science of how a bar of iron can become magnetized by applying an external magnetic field, and how it can be demagnetized through heating. We also learn the difference between permanent and temporary magnets, and how the former are also known as ferromagnets. The term is derived from the element iron (chemical symbol Fe), for which this phenomenon is most readily observabed.

Within a piece of iron there are magnetic "domains," which are clusters of atomic spins that have aligned themselves parallel to each other. Unlike the spins in an ideal paramagnet, for which the alignment of the spins is influenced only by the presence of an external magnetic field, the spins in a ferromagnet exert an influence on their neighbors. As we will see later, this type of 'cooperative' phenomenon in physical and social systems is the primary motivation for this thesis.

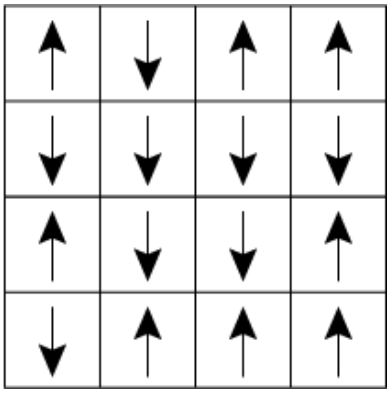


Figure 1.1 The 2D Ising square lattice. The spins are seen to been arranged as either up or down in this simple model of ferromagnetism.

Due to this cooperative behavior between neighboring spins, ferromagnets have the distinctive property of being able to undergo spontaneous autonomous magnetization. This is when there is a net magnetization present in the material even in the absence of an external magnetic field. Another noteworthy property of ferromagnets is that they undergo a phase transition when the temperature exceeds a critical point called the "Curie temperature." At this transition the ferromagnet loses its spontaneous magnetization and becomes a paramagnet, a material that does not display net magnetization without an external field. We will be taking a more detailed look at phase transitions in Section 2.2.

Motivated by these properties of ferromagnets, Wilhelm Lenz, and later on, his student Ernst Ising, proposed a simple model that sought to explain the phenomena of spontaneous magnetization and phase transitions [10]. Their model assumed that the ferromagnet can be represented as a regular lattice of spins, which can point in either of two directions. Figure 1.1 shows an example of such a lattice, consisting of 16 spins, in two dimensions.

The state ν of a particular physical system represented by the model is the arrangement of the spins. So, a lattice consisting of N spins would have a total of 2^N

possible states. In this model expressed the energy E of the system at a particular state ν would be expressed as

$$E(\nu) = -\sum_{i=1}^{N} H\mu s_i - J\sum_{\langle ij \rangle} s_i s_j, \tag{1.1}$$

where the first term represents the energy due to the external magnetic field H summed over each of the spins denoted by s_i, which can take a value of either 1 or -1. In the second term, the subscript $\langle ij \rangle$ denotes summation over all neighboring pairs of spins, multiplied by the coupling constant J.

In his doctoral dissertation in 1925, Ising published an analytic solution to the one-dimensional version of his model. Contrary to his expectation, his model failed to predict the phase transition observed in ferromagnets. It did, however capture the tendency for magnetization to occur at low temperatures. Details of the one-dimensional model will be discussed in Section 2.3.

In higher dimensions, however, it could be shown that the Ising model does display a phase transition. A breakthrough in the development of the Ising model came in a paper published in 1944 by Lars Onsager [20], in which he derived the exact analytic solution to the two-dimensional Ising model with no external magnetic field. In the solution, he showed that the two-dimensional Ising model did exhibit a phase transition at a certain critical temperature. The mathematics involved in Onsager's solution is, however, extremely complicated and outside the scope of this thesis. Onsager's solution prompted a search for an analytical solution to the Ising model in even higher dimensions, especially in the three-dimensional world that real ferromagnetic lattices inhabit. In 2000, a computational biologist working at Sandia National Laboratories, Sorin Istrail, published a paper in which he showed that the solution of the three-dimensional Ising model was computationally intractable, in the sense that the problem is NP-complete [15].

This lack of an analytic solution means that most applications of the Ising model are solved by using either approximations or simulations. A variety of approximation methods have been applied to the Ising model. The methods that will be examined in this thesis include the mean-field approximation and renormalization group theory [11, Ch. 5]. The former approach replaces the interactions of the many different neighbors on each spin with the average effect these interactions collectively have on the spin. This effectively reduces the many-body problem involved in solving the Ising model to a one-body problem, making it much easier to tackle analytically. The mean-field method correctly predicts the phase transition that occurs in the Ising model in two or more dimensions, but also wrongly predicts that the one-dimensional model also goes through a phase transition, which we know from Ising's original solution is not the case. The mean-field approach also predicts the critical temperature at which the phase transition occurs, increasing in accuracy as we increase the number of dimensions in the model. Renormalization group theory, developed in 1971 by Kenneth Wilson, uses the Ising model's tendency to exhibit self similarity at different scales. As it would be more beneficial to discuss the details of both these approximation methods with recourse to the appropriate mathematical procedures, I will defer a more thorough discussion of the methods to Chapter 2.

Another approach to solving the Ising model is through the use of computer simulations that implement Monte Carlo methods. The Metropolis-Hastings algorithm is a standard procedure used for carrying out such simulations, and was first used by Metropolis et al. in 1953 [17]. The procedure involves first storing the state of the Ising lattice in a computer and then randomly sampling spins and flipping them, changing the state of the lattice. After this, the before and after states are compared to see which has a higher probability of occurring, and the state with the higher probability is then stored as the outcome of the iteration. After many iterations, the

lattice settles into a high probability state. Monte Carlo simulations will be used as the primary tool to analyse various versions of Ising and similar models in Chapter 3.

Since that the Ising model can account for phase transitions, it became widely accepted in the physics community towards the 1930's and 40's as a useful tool for describing a wide variety of cooperative phenomena [19]. Physical systems such as binary alloys and transitions from liquid to gaseous states can all be modelled in ways that make them mathematically equivalent to the Ising model. Throughout the years, this generality has made the Ising model a workhorse in the field of statistical mechanics. Tens of thousands of papers have been published on the topic. In the last twenty years or so, the Ising model has been applied even to fields outside of physics. A review of these interdisciplinary applications is the primary motivation for this thesis.

1.2 Interdisciplinary applications

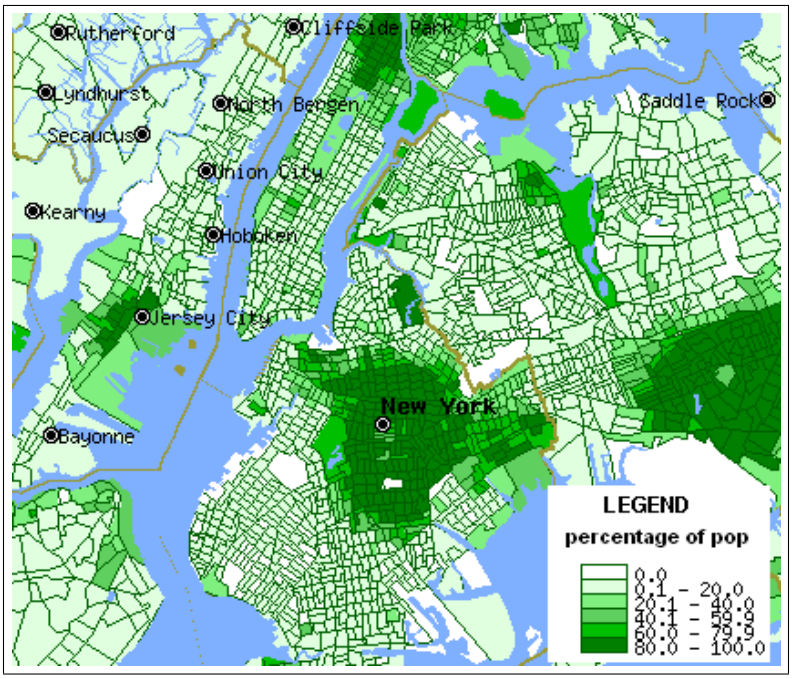
To date, the Ising model and others that were directly inspired by it have been applied to fields as varied as urban segregation, linguistics [24], political science [5,27,29], neural networks [4, p. 145], and many others. The thread of similarity running through all of these applications is that these phenomena can all be modeled using simple dynamics of interactions between neighbors in a population. The arrangements of the basic components of the model, be they individuals, nodes in a neural network, or atomic spins, can take various forms. These models can be run on different structural arrangements, from lattices to complex networks, the latter arrangement being especially relevant in social science applications. This thesis will consider some examples of Ising type models both on complex networks and on lattices.

We will look at two particular models in significant detail, Thomas Schelling's model of urban segregation, and Katarzyna Sznajd-Weron's model of opinion dynamics, which are introduced below.

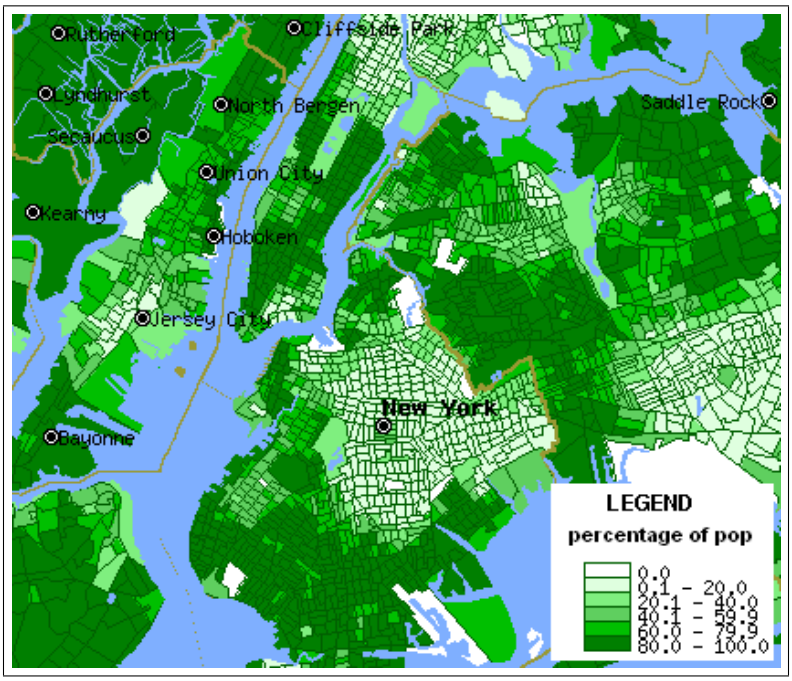
1.2.1 Schelling's segregation model

In 1971, Thomas Schelling, an economist and Nobel laureate, published a model of segregation, which was later made famous in his book 'Micromotives and Macrobehavior' [21, Ch. 4]. The model consisted of a two-dimensional grid on which resided two different populations, made up of individuals who inhabited each point on the grid. Each individual prefers to live in a cell where her neighbors belong to the same group as she. On a square lattice, let's say each person's neighbors consisted of the eight inhabitants of the cells to the North, North-East, East, etc., of her current cell. We now define a critical threshold, T_c, of neighbors belonging to the same group, below which the individual will be 'unhappy' and will be prompted to move to another random location, in hopes of finding more preferable neighbors. This simple model of preference for neighbors and random migration turns out to be a pretty accurate model for the formation of urban neighborhoods segregated along racial and other lines. For instance, the outcomes in a typical simulation of Schelling's model resemble the racial clusters that exist in a city like New York, as shown in Fig. 1.2.

Schelling's model clearly has a similarity to the two-dimensional Ising model on a square lattice. In fact, with some additional assumptions, these two models can be modeled in the same way [25]. There are, however, some differences worth noting. In the original Schelling model, in addition to cells in the lattice occupied by individuals from two groups, there are also empty spaces, which either group can move in to occupy. Also, Schelling's model had no analog to the external magnetic field, H, that we saw in Eq. 1.1. In Chapter 4, we will explore Schelling's model using the



(a) Black neighborhoods.



(b) White neighborhoods.

Figure 1.2 Distribution of black and white neighborhoods in New York City. Real world patterns of urban racial segregation, such as this resemble clusters that result from simulations of Schelling's model [3].

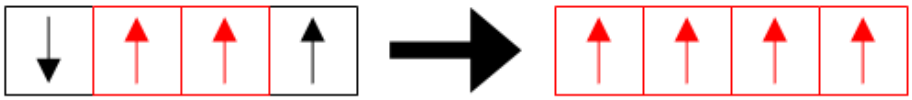


Figure 1.3 One of the rules in the 1D Sznajd model that models the spread of opinions.

assumptions he made in his original model, and also in the modified version which makes it equivalent to the Ising model.

1.2.2 Sznajd's opinion dynamics model

In 2000, Katarzyna Sznajd-Weron and Jósef Sznajd published a sociophysics model that was inspired by the Ising model [28]. Like the Ising and Schelling models, the model consists of a collection of individuals whose 'opinions' or 'spin' can take either of two values. Also, as in the previous models, the outcome of each individual's opinion in this model is influenced by its neighbors. In their paper, they ran Monte Carlo simulations for a one-dimensional version of their model, which has come to be known as the Sznajd model. In the last decade, various extensions have been made to the model and it has been applied to various fields.

To illustrate how the Sznajd model works, consider the following dynamical rule from the one-dimensional version of the model, as shown in Fig. 1.3. Notice that when the pair of spins in the center of the group of 4 spins were aligned in the same direction, the next iteration causes this alignment to spread to the two nearby spins, causing all 4 spins to point in the same direction. This rule models the spread of opinions through a population starting from a initial group of poeple who are in agreement with each other. We will discuss the rules and the results for the one-dimensional and two dimensional versions of the Sznajd model in Section 4.3.

1.2.3 Durlauf and Blume's social interactions model

Steven Durlauf, from Cornell University, and Lawrance Blume, from the Univerisity of Wisconsin, are both economists and also researchers at the Santa Fe Institute. They are among a growing number of economists and social scientists who are developing models that are directly influenced by discrete spin models from statistical mechanics such as the Ising model. Schelling's model was a forerunner of this kind of research in the social sciences, but was not intended to have sound theoretical underpinnings and was instead designed to prove a simple point that micromotives could lead to interesting macrobehavior. Sznajd's model and other models that have been recently developed by physicists also tend to ignore some the established theoretical foundations of the social sciences, even though these physicists' models offer useful insights into social phenomena. One example of such theoretical foundations would be the assumption, at least in economics, that agents act rationally to maximize utility, and are not merely following simple behavioral rules.

Durlauf and Blume present an example of an interactions-based model in a paper that summarises their decade long work in the field [7]. They use a mean field theory approach to solve their model, but do not use any computational techniques. Their model is discussed in more detail in Section 4.4.

1.3 Overview of thesis

We will look at some analytical and computational solutions to the Ising model and also review some of the literature on the social science applications of Ising and other discrete spins models. Chapter 2 looks at the analytic and approximate solutions to the Ising model. It covers some basic statistical mechanics concepts, and derives the solution to the one-dimensional Ising model. After that, it covers the mean field

approximation and the renormalization group theory for the one-dimesional and two-dimensional Ising models. Chapter 3 covers the computational methods that are used to solve the Ising model. The Metropolis algorithm is introduced and a proof of the convergence of the algorithm is given, after which results from the two-dimensional and three-dimensional simulations of the Ising model are presented. Chapter 4 looks at three different models from the social sciences that are influenced by or are similar to the Ising model. As discussed in Section 1.2, they are Schelling's segregation model, Sznajd's opinion dynamics model, and Durlauf and Blume's social interactions model respectively. Chapter 5 concludes.

Chapter 2

Analytic and Approximate Solutions to the Ising Model

2.1 Introduction to some statistical mechanics

In order to understand the mathematics behind the Ising model, we first have to introduce some of the concepts from statistical mechanics that form the theoretical basis of the model. The idea behind statistical mechanics is that the macroscopic properties of physical systems can be studied through the statistical analysis of the microscopic interactions that take place among the presumed microscopic constituents of those systems.

While classical thermodynamics can be used to analyse the macroscopic behavior of physical systems, it is not an approach based on intearactions between individual atoms and molecules. Statistical mechanics, on the other hand, does delve into the realm of these microstates, and seeks to explain macroscopic thermodynamic properties as emerging from the microinteractions when a system is at equilibrium. Statistical mechanics deals specifically with systems involving a large number of constituent

particles. For example, in systems that are typically examined by thermodynamics, the number of particles present is of the order of ~10^{23}. From the famous three body problem in mechanics, we know that it is impossible to make predictions of the long term behavior for a system consisting of many particles. However, there are statistical regularities that emerge from the interactions of large numbers of particles, and in many cases these systems can be analyzed using just a handful of macroscopic variables, such as temperature and pressure.

A system of interacting particles can exist in various "microstates." Each microstate describes the specific configuration of each particle in the system. The fundamental assumption of statistical mechanics is that, in an isolated system in thermal equilibrium, all accessible microstates are equally accessible [22, p. 57]. The macrostate defined, for example, by a temperature of $T = 300$ K, thus becomes a probabilistic measure dependent on the distribution of microstates that allow this macrostate to occur. An equilibrium temperature, for instance, is simply the macrostate which is the most likely to occur, given all the possible microstates of the system plus a heat bath. One additional concept that needs to be defined before the relationship betweem microstates and macrostates can be made more clear is that of multiplicity. The multiplicity of a system, Ω, is the number of microscopic states that are accessible given a specific macrostate that the system is in. Therefore, the equilibrium macrostate, is that which has the highest multiplicity and hence the highest likelihood for occurring.

From this notion of microstates and multiplicity, we derive the concept of entropy, which is defined as

$$S = k \ln \Omega, \tag{2.1}$$

so that where the entropy, S, of a system, is directly proportional to the log of the

multiplicity Ω. The constant k, is Boltzmann's constant, with $k = R/N_A$, the ratio of the gas constant R and Avogadro's constant N_A. As such,

$$k = \frac{R}{N_A} \tag{2.2}$$

$$= \frac{8.314472\text{JK}^{-1}\text{mol}^{-1}}{6.02214179 \times 10^{23}\text{mol}^{-1}} \tag{2.3}$$

$$= 1.3806504 \times 10^{-23}\text{JK}^{-1} \tag{2.4}$$

$$= 8.617343 \times 10^{-5}\text{eVK}^{-1}. \tag{2.5}$$

From this definition of entropy, we can restate the earlier observation about multiplicity. Since the equilibirum macrostate is one which has the highest multiplicity, the entropy of the system will also be at its maximum for a system in equilibrium. This definition of entropy as a logarithmic function of multiplicity leads to the statistical formulation of the second law of thermodynamics,

$$S(N, V, E) > S(N, V, E, \text{internal constraint}), \tag{2.6}$$

which means that the entropy of a system of a particular number of particles, N, volume , V, and internal energy, E, will always increase as we remove internal constraints [11, p. 61]. An example of an internal constraint would be a partitioning of the system into two sub-systems, each of which would only allow a subset of the possible microstates if the system were not partitioned.

2.1.1 Boltzmann factor

Before introducing the Boltzmann factor, we have to discuss the properties of a thermodynamic "system" that is in thermal contact with a much larger "reservior," as shown in Fig. 2.1. The internal energy of the reservoir is denoted U_R, while the internal energy of the system is denoted E.

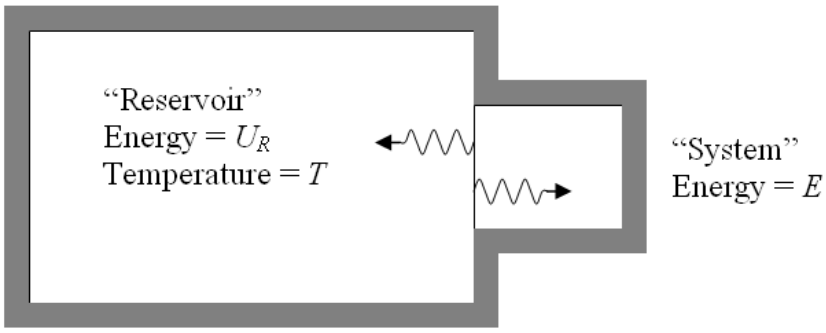


Figure 2.1 A "system" is in thermal contact with a much larger "reservoir" at a temperature T [22, p. 221].

The entropy, S_R, volume, V_R, internal energy, U_R, and number of particles, N_R, of the reservoir, together with the temperature, T, and pressure, P, of both the system and the reservoir are related to each other through an important identity from classical thermodynamics that will not be rederived here. The identity shows the relation between four partial derivatives,

$$dU_R = TdS_R - PdV_R + \mu dN_R, \tag{2.7}$$

and is known as the thermodynamic identity [22, p. 117]. It is essentially a statement of the conservation of energy. The constant μ is the chemical potential. If we are to consider a reservoir with fixed number of particles and volume and therefore hold V_R and N_R constant, we obtain the relationship,

$$dS_R = \frac{1}{T}dU_R. \tag{2.8}$$

If the system is changing from microstate ν_1 to state ν_2, Eq. 2.8 can be rewritten as

$$S_R(\nu_2) - S_R(\nu_1) = \frac{1}{T}[U_R(\nu_2) - U_R(\nu_1)] = -\frac{1}{T}[E(\nu_2) - E(\nu_1)]. \tag{2.9}$$

since any change in the internal energy of the reservoir, U_R, must be matched by a corresponding change in the opposite direction in the internal energy of the system, E, in order to conserve the total energy of the system and the reservoir.

Now, we know that the probability of a system microstate, ν, is directly proportional to the multiplicity, Ω, of reservior states associated with that microstate. Hence, the ratio of probabilities of two macrostates can be written as

$$\frac{\mathcal{P}(\nu_2)}{\mathcal{P}(\nu_1)} = \frac{\Omega_R(\nu_2)}{\Omega_R(\nu_1)} \tag{2.10}$$

$$= \frac{e^{S_R(\nu_2)/k}}{e^{S_R(\nu_1)/k}} \qquad \text{(substituting from Eq. 2.1)} \tag{2.11}$$

$$= \frac{e^{-E(\nu_2)/kT}}{e^{-E(\nu_1)/kT}} \qquad \text{(substituting from Eq. 2.9),} \tag{2.12}$$

$$\tag{2.13}$$

and therefore,

$$\frac{\mathcal{P}(\nu_2)}{e^{-E(\nu_2)/kT}} = \frac{\mathcal{P}(\nu_1)}{e^{-E(\nu_1)/kT}} \tag{2.14}$$

From Eq. 2.14, we can see that the ratio of the probability of a particular system microstate, $\mathcal{P}(\nu)$, to the term, $e^{-E(\nu)/kT}$, is constant regardless of the microstate ν. This denominator term is called the Boltzmann factor. It provides a relative way to measure the probability of a certain state given only its energy and the temperature. Also, from this constant ratio obtained in Eq. 2.14, we can derive the following relation for the probability of a microstate ν with a normalization factor of $1/Z$ as

$$\mathcal{P}(\nu) = \frac{1}{Z} e^{-E(\nu)/kT}. \tag{2.15}$$

Given the probability of a microstate ν, we can define entropy, S, of a system in a different way from that derived from the multiplicity as we saw in Eq. 2.1. Without derivation, we will state this second definition of entropy as

$$S = -k \sum_{\nu} \mathcal{P}(\nu) \ln \mathcal{P}(\nu). \tag{2.16}$$

2.1.2 Partition function

The term Z from Eq. 2.15 is called the partition function. In order to calculate the value for Z, we utilize the fact the the sum of probabilities of the possible microstates, ν, of the system must be equal to 1. Thus,

$$\sum_{\nu} \mathcal{P}(\nu) = 1 \tag{2.17}$$

$$\sum_{\nu} \frac{1}{Z} e^{-E(\nu)/kT} = 1. \tag{2.18}$$

Multiplying both sides of Eq. 2.18 by the partition function, Z, we can see that it is equal to the sum of all the Boltzmann factors for the possible microstates of a system. It is common to use the letter β, which is defined as $\beta \equiv 1/kT$, to simplify the notation. Therefore, we can rewrite the expression for the partition function as

$$Z = \sum_{\nu} e^{-E(\nu)/kT} \tag{2.19}$$

$$= \sum_{\nu} e^{-\beta E(\nu)}. \tag{2.20}$$

We will now derive an expression for the average energy of all the particles in a system, $\overline{E}$. This is simply the probability weighted sum of the energies of all the possible microstates of the system:

$$\overline{E} = \sum_{\nu} E(\nu)\mathcal{P}(\nu) \tag{2.21}$$

$$= \frac{1}{Z} \sum_{\nu} E(\nu) e^{-\beta E(\nu)}. \tag{2.22}$$

In order to express $\overline{E}$ in more concise terms, consider the following partial derivatives of Z and $\ln Z$. Using the expression for Z from Eq. 2.20, we can derive

$$\frac{\partial Z}{\partial \beta} = -\sum_{\nu} E(\nu) e^{-\beta E(\nu)}, \tag{2.23}$$

and also,

$$\frac{\partial}{\partial\beta}\ln Z = \frac{1}{Z}\frac{\partial Z}{\partial\beta}. \tag{2.24}$$

$\overline{E}$ can now be expressed as

$$\overline{E} = -\frac{1}{Z}\frac{\partial Z}{\partial\beta} = -\frac{\partial}{\partial\beta}\ln Z. \tag{2.25}$$

2.1.3 Free energies

Being discussing the Ising model, we have to define additional terms and concepts that will prove useful in the later discussion. In thermodynamics, there is a concept known as free energy, which is the amount of energy that can be recovered from a system that is convertible to work. We define two distinct types of free energies. The first is the Helmholtz free energy, which is the energy that can be recovered from a system which is at constant temperature and volume. In thermodynamics, it is defined as

$$F \equiv E - TS, \tag{2.26}$$

where the Helmholtz free energy, F, is equal to the internal energy of the system, E, minus the heat that must flow to or from the system to a surrounding heat bath, which is the product of the temperature, T, and the entropy, S. From this thermodynamic definition of the Helmholtz free energy, we can derive a statistical mechanical expression for F. We begin with the definition of entropy given in Eq.

2.16 and derive an an expression equivalent to Eq. 2.26, as [23, p. 108]:

$$S = -k\sum_{\nu}\mathcal{P}(\nu)\ln\mathcal{P}(\nu) \tag{2.27}$$

$$= -k\sum_{\nu}\frac{1}{Z}e^{-\beta E(\nu)}\ln\left(\frac{1}{Z}e^{-\beta E(\nu)}\right) \tag{2.28}$$

$$= -k\sum_{\nu}\frac{1}{Z}e^{-\beta E(\nu)}\left(-\beta E(\nu) - \ln Z\right) \tag{2.29}$$

$$= k\beta E + k\ln Z\sum_{\nu}\frac{1}{Z}e^{-\beta E(\nu)} \tag{2.30}$$

$$= \frac{E}{T} + k\ln Z. \tag{2.31}$$

From Eq. 2.31, we can simply solve for $E-TS$, after which we arrive at the statistical mechanical definition of Helmholtz free energy:

$$F = -kT\ln Z. \tag{2.32}$$

The other concept of free energy that we have to define is the Gibbs free energy. It is related to Helmholtz free energy but instead of holding temperature and volume to be constant, we hold temperature and pressure to be constant [12, p. 75]. From the definition of Gibbs free energy,

$$G \equiv E - TS + PV = F + PV, \tag{2.33}$$

we see that the Helmholtz free energy, F, and Gibbs free energy, G, differ only by the addition of the product PV.

2.1.4 Heat capacity

Another important thermodynamic property we have to consider is the heat capacity, C. It is a measure of the change in internal energy with respect to temperature. We are especially interested in the heat capacity at a constant volume, C_V, which is

defined as [12, p. 46]:

$$C_V = \left(\frac{\partial E}{\partial T}\right)_V. \tag{2.34}$$

To find the heat capacity associated with the mean energy $\overline{E}$ of the system, we can substitute the value of $\overline{E}$ from Eq. 2.25, giving

$$C_V = \frac{\partial \overline{E}}{\partial T} = \frac{d\beta}{dT}\frac{\partial \overline{E}}{\partial \beta}, \tag{2.35}$$

$$= \frac{d}{dT}\left[\frac{1}{kT}\right]\frac{\partial}{\partial \beta}\left[-\frac{1}{Z}\frac{\partial Z}{\partial \beta}\right], \tag{2.36}$$

$$= \frac{1}{kT^2}\left[\frac{1}{Z}\frac{\partial^2 Z}{\partial \beta^2} - \frac{1}{Z^2}\left(\frac{\partial Z}{\partial \beta}\right)^2\right]. \tag{2.37}$$

From Eqs. 2.20 and 2.22, we get the following expression:

$$\overline{E^2} = \frac{1}{Z}\sum_{\nu} E^2(\nu)e^{-\beta E(\nu)} = \frac{1}{Z}\frac{\partial^2 Z}{\partial \beta^2}, \tag{2.38}$$

which means we can rewrite C_V as

$$C_V = \frac{1}{kT^2}\left[\overline{E^2} - \overline{E}^2\right]. \tag{2.39}$$

The term in the square brakets, the variance of the internal energy, can now be used to calculate the heat capacity of the system.

2.1.5 Magnetization

The magnetization, $\overline{M}$, is defined as the sum of the individual spins, s_i, in the system multiplied by the magnetic moment μ, expressed as [12, p. 229]

$$\overline{M} = \mu\sum_{i=1}^{N} s_i. \tag{2.40}$$

We can also define a measure of mean magnetization per spin, as $m \equiv \overline{M}/N$, which is simply $\overline{M}$ divided by the number of spins, N.

For the purpose of studying the Ising model, we need to derive an expression for $\overline{M}$ that is based on the partition function Z. To do so, we have to first define the total energy, E_T, of a system in the presence of a magnetic field as

$$E_T = E_0 - MB, \tag{2.41}$$

where E_0 is the energy of the interaction of the spins with each other. The second term is the product of the mean magnetization, $\overline{M}$, and the magnetic field, B. E_T can be defined for each microstate ν. This means we can rewrite the partition function, Z, by summing over all the microstates, ν, as

$$Z = \sum_{\nu} e^{-\beta(E_0(\nu) - \overline{M}(\nu)B)}, \tag{2.42}$$

which give a partial derivative with respect to B as

$$\frac{\partial Z}{\partial B} = \sum_{\nu} \beta M(\nu) e^{-\beta(E_0(\nu) - M(\nu)B)}. \tag{2.43}$$

We can now express the mean magnetization, $\overline{M}$, as

$$\overline{M} = \frac{1}{Z} \sum_{\nu} M(\nu) e^{-\beta(E_0(\nu) - M(\nu)B)} \tag{2.44}$$

$$= \frac{1}{\beta Z} \frac{\partial Z}{\partial B} = kT \frac{\partial \ln Z}{\partial B}. \tag{2.45}$$

Since the Helmholtz free energy is given by $F = -kT \ln Z$, we get

$$\overline{M} = \frac{\partial F}{\partial B}. \tag{2.46}$$

2.1.6 Magnetic susceptibility

The magnetic susceptibility, χ, shows the relation between the mean magnetization, $\overline{M}$, and the magnetic field, B; for a constant temperature, T, it is defined by

$$\chi \equiv \left(\frac{\partial \overline{M}}{\partial B} \right)_T. \tag{2.47}$$

We can also derive an expression for χ that does not rely on partial derivatives but instead resembles the form we derived for C_V in Eq. 2.39. To do so, we start with the expression for $\frac{\partial Z}{\partial B}$ given in Eq. 2.43. From there, we rederive χ by writing $\overline{M}$ from Eq. 2.46 as a weighted mean of all the $M(\nu)$ values for all the possible microstates, as

$$\chi = \frac{\partial \overline{M}}{\partial B} \tag{2.48}$$

$$= \frac{\partial}{\partial B}\left(\frac{1}{Z}\sum_{\nu} M(\nu)e^{-\beta(E_0(\nu)-M(\nu)B)}\right) \tag{2.49}$$

$$= \frac{1}{Z^2}\left[Z\sum_{\nu}\beta M^2(\nu)e^{-\beta(E_0(\nu)-M(\nu)B)} - \frac{\partial Z}{\partial B}\sum_{\nu} M(\nu)e^{-\beta(E_0(\nu)-M(\nu)B)}\right] \tag{2.50}$$

$$= \beta\left[\frac{1}{Z}\sum_{\nu} M^2(\nu)e^{-\beta(E_0(\nu)-M(\nu)B)} - \frac{1}{Z^2}\left(\sum_{\nu} M(\nu)e^{-\beta(E_0(\nu)-M(\nu)B)}\right)^2\right] \tag{2.51}$$

$$= \frac{1}{kT}\left[\overline{M^2} - \overline{M}^2\right], \tag{2.52}$$

where we have substituted the value for $\overline{M}$ from Eq. 2.44.

2.2 Phase transitions

As mentioned in Chapter 1, the Ising model was originally developed to explain phase transitions in ferromagnets. Magnets are not the only physical systems which undergo phase transitions; another familiar example is the transition between phases of matter, such as boiling or freezing. Superconductivity and superfluidity in some materials are also examples of phase transitions.

There are two broad categories of phase transitions, namely first and second-order transitions. First-order transitions involve latent heat, which means that the material either absorbs or emits a certain about of energy without a change in temperature during the transition. When we heat solid ice at a constant pressure and observe it

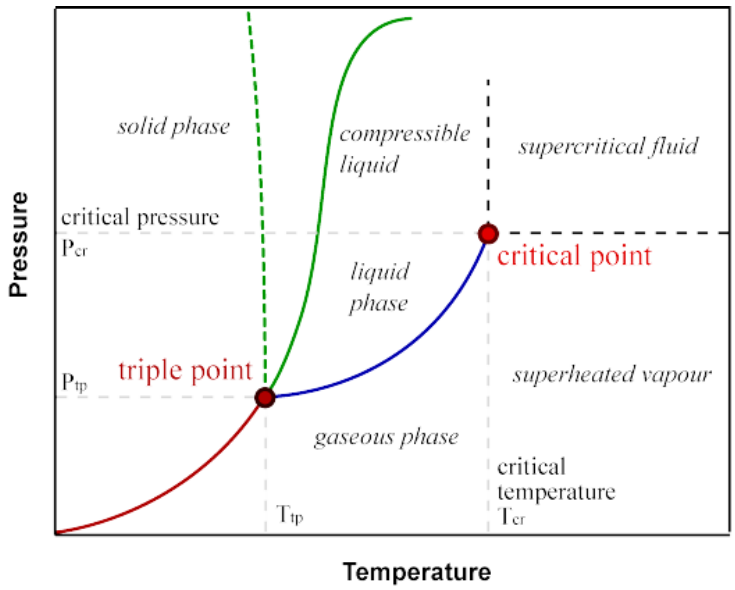


Figure 2.2 Phase diagram for H_2O, showing the different phases for different temperature and pressure values [1].

melt and eventually boil, the changes from solid to the liquid and gaseous states are first-order phase transitions. Second-order phase transitions, also known as continuous phase transitions, do not involve latent heat. The ferromagnetic phase transition associated with the Ising model is an example of a second-order phase transition. Another example is that of water at the critical point, which occurs around a temperature of 647K and a pressure of 22.1MPa [22, p. 168]. Figure 2.2 shows the phase transitions that occur for H_2O across different temperatures and pressures.

In a first order phase transition, the two phases exist in the system simultaneously. For instance when ice melts at 0°C, we can observe the solid and liquid phases existing together. However, for the second order phase transition, there is a more complex transition from one phase to another. There are local fluctuations that take place across the system at every length scale. For instance, in the Ising model, the phase transition is marked by the existence of magnetic domains of all sizes cropping up across the different length scales of the system, from the microscopic to the

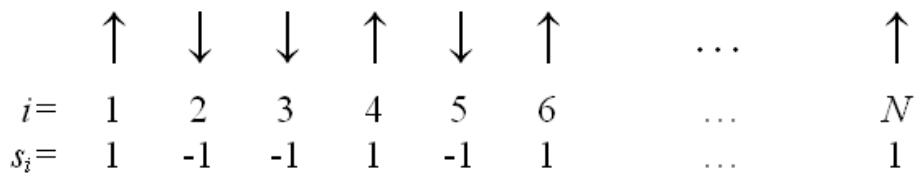


Figure 2.3 A representation of the spins in the one-dimensional Ising model.

macroscopic.

2.3 Ising model in one dimension

We are now ready to tackle the analytic solution of the Ising model in one dimension. Consider an Ising chain in one dimension, in the absence of an external magnetic field. From Eq. 1.1, the energy, $E(\nu)$, of a particular microstate, ν, of the Ising chain can be written as

$$E(\nu) = -J \sum_{\langle ij \rangle} s_i s_j \tag{2.53}$$

$$= -J(s_1 s_2 + s_2 s_3 + s_3 s_4 + ... s_{N-1} s_N). \tag{2.54}$$

Figure 2.3 clarifies the expression in Eq. 2.54. In this formulation, it is a chain and not a ring, which means there is no interaction betweeen s_1 and s_N.

The partition function, Z, is therefore

$$Z = \sum_{\nu} e^{\beta\left(J \sum_{\langle ij \rangle} s_i s_j\right)} \tag{2.55}$$

$$= \sum_{\nu} e^{\beta J s_1 s_2} e^{\beta J s_2 s_3} ... e^{\beta J s_{N-1} s_N}. \tag{2.56}$$

The summation over all possible states, ν, in Eq. 2.56 can be expanded into a

series of nested sums. Since each s_i can take the values ± 1, we can expand it as

$$Z = \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \sum_{s_3=\pm 1} ... \sum_{s_N=\pm 1} e^{\beta J s_1 s_2} e^{\beta J s_2 s_3} ... e^{\beta J s_{N-1} s_N} \tag{2.57}$$

$$= \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} e^{\beta J s_1 s_2} \sum_{s_3=\pm 1} e^{\beta J s_2 s_3} ... \sum_{s_N=\pm 1} e^{\beta J s_{N-1} s_N}. \tag{2.58}$$

For each pair of spins, the product, $s_i s_{i+1}$, can only take the values ± 1, $\forall i \in \{1, 2, 3, ...N-1\}$. This means that the sum,

$$\sum_{s_{i+1}=\pm 1} e^{\beta J s_i s_{i+1}} = e^{-\beta J} + e^{\beta J} \tag{2.59}$$

$$= 2 \cosh(\beta J). \tag{2.60}$$

We now use this to simplify the partition function, Z,

$$Z = \sum_{s_1=\pm 1} [2 \cosh(\beta J)]^{N-1} \tag{2.61}$$

$$= 2^N [\cosh(\beta J)]^{N-1} \tag{2.62}$$

$$\approx [2 \cosh(\beta J)]^N \text{ , for large } N. \tag{2.63}$$

From the partition function, we can derive the average energy of the system using Eq. 2.25:

$$\overline{E} = -\frac{\partial}{\partial \beta} \ln Z \tag{2.64}$$

$$= -NJ \tanh(\beta J). \tag{2.65}$$

The average energy, $\overline{E}$, is shown in Eq. 2.65 as a function of $\beta \equiv 1/kT$. This is the analytic solution to the one-dimensional Ising model. The significance of this solution is that unlike in the case of real world ferromagnets, whose behavior Ising intended to duplicate with his model, the one-dimensional Ising model does not show a phase transition as the temperature changes. The phase transition in a ferromagnet

is a spontaneous change in the net magnetization of the system as the temperature changes. In order to see why this does not occur in the one-dimensional Ising model, consider the plot in Fig. 2.4 of the average energy per particle $\overline{E}/NJ$, as a function of the temperature T. The function is smooth, has no discontinuities, and is differentiable at every point. We see that $\overline{E}/NJ$ approaches 1 as $T \to 0$, and $\overline{E}/NJ$ slowly approaches 0 as $T \to \infty$.

Despite the lack of a phase transition in the one-dimensional Ising model, it does show spontaneous magnetization occuring at $T \to 0$. As we can see from Fig. 2.65, $\overline{E}/NJ$ approaches 1 at low temperatures, which means that the spins are all aligned in the same direction.

The quantity $\overline{E}/N$ is not quite a measure of the magnetization of the system; however, it is directly correlated with it. Consider an Ising chain in which every spin s_i is parallel to every other, such that each product $s_i s_{i+1} = 1$. From Eq. 2.54, we can see that the absolute value of the energy, $|E|$, is at its maximum. This is also the case if the Ising chain consisted of neighboring spins that were antiparallel to each other, such that each product $s_i s_{i+1} = -1$. This situation will also yield the maximum value for $|E|$. In the former case, net magnetization is at its maximum, and in the latter, it is 0. Therefore, we have established that the maximum magnetization corresponds with maximum energy. If, however, exactly half of the neighboring spins were parallel and the other half were antiparallel, we would get a situation in which $|E| = 0$ according to Eq. 2.54. If the spins are randomly arranged, it is likely that there will be a mix of parallel and antiparallel spins, and therefore a low value for $|E|$. At the same time, probabalistically, the expected magnetization for a random arrangement of spins is 0. Therefore, a low average energy corresponds with low net magnetization.

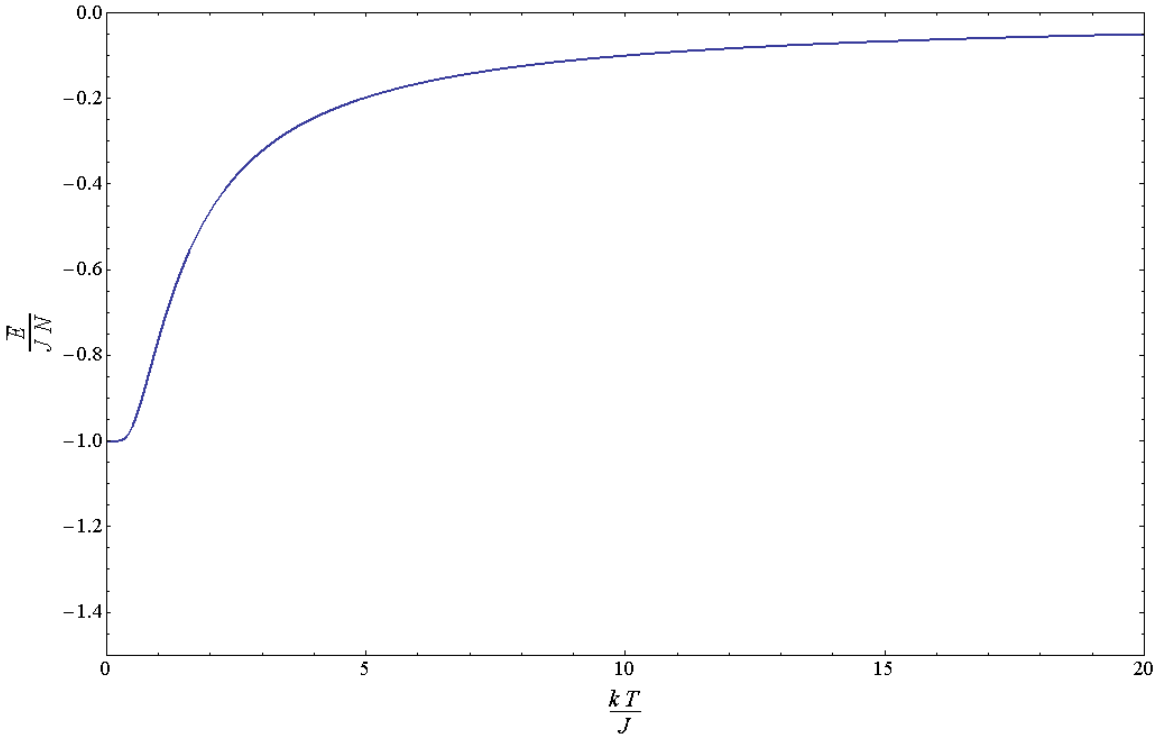


Figure 2.4 Average energy vs. temperature in the one-dimensional Ising model. The smooth curve indicates that there is no phase transition.

2.4 Ising model in two dimensions

The one-dimensional Ising model is easy to solve analytically; however, the analytical solution to the two-dimensional model, first discovered by Onsager in 1944, is much harder to solve. We will merely cite the solution here [11, p. 123]. The partition function for the two-dimensional Ising model with no external magnetic field is

$$Z = \left[2\,\cosh(\beta J)e^{I}\right]^{N}, \tag{2.66}$$

where

$$I = (2\pi)^{-1}\int_{0}^{\pi} d\phi\,\ln\,\left\{\frac{1}{2}\left[1+(1-\kappa^{2}\,\sin\,^{2}\phi)^{1/2}\right]\right\}, \tag{2.67}$$

with

$$\kappa \equiv \frac{2\,\sinh\,(2\beta J)}{\cosh\,^{2}(2\beta J)}. \tag{2.68}$$

From Onsager's solution, we can derive the critical temperature, T_c, at which the phase transition in the two-dimensional Ising model takes place. T_c is derived from the relation

$$\sinh\frac{2J}{kT_c} = 1, \tag{2.69}$$

so,

$$\frac{kT_c}{J} = \frac{2}{\ln(1+\sqrt{2})} \approx 2.269. \tag{2.70}$$

In Chapter 3, we will run Monte-Carlo simulations of the two-dimensional Ising model. As such, it would be beneficial to know what the analytic solutions are, in order to check whether the simulation results are correct. Thus, we will now look at the analytic results for the energy, E, heat capacity, C, and magnetization, M, that are derived from Onsager's solution.

The analytical solution for E [12, p. 247] is given by the expression

$$E = -2NJ\tanh 2\beta J - NJ\frac{\sinh^2\beta J - 1}{\sinh 2\beta J \cosh 2\beta J}\left[\frac{2}{\pi}K_1(\kappa) - 1\right], \tag{2.71}$$

where

$$K_1(\kappa) \equiv \int_0^{\pi/2} \frac{d\phi}{\sqrt{1-\kappa^2\sin^2\phi}}. \tag{2.72}$$

$K_1(\kappa)$ is known as the complete elliptic integral of the first kind.

The heat capacity, C [12, p. 248], derived using Eq. 2.34 is

$$C = Nk\frac{4}{\pi}(\beta J\coth 2\beta J)^2\left[K_1(\kappa) - E_1(\kappa)\right. \tag{2.73}$$

$$\left. -(1-\tanh^2 2\beta J)(\frac{\pi}{2} + (2\tanh^2 2\beta J - 1)K_1(\kappa))\right], \tag{2.74}$$

where

$$E_1(\kappa) \equiv \int_0^{\pi/2} d\phi\sqrt{1-\kappa^2\sin^2\phi}. \tag{2.75}$$

$E_1(\kappa)$ is known as the complete elliptic integral of the second kind.

The mean magnetization per spin, $m = M/N$, for Onsager's solution was was published by C. N. Yang in 1952 [12, p. 249] and is given by

$$m(T) = \begin{cases} (1 - [\sinh 2\beta J]^{-4})^{1/8} & T < T_c, \\ 0 & T > T_c. \end{cases} \tag{2.76}$$

The plots of E, C and $|m|$ with respect to temperature are shown in Figs. 2.5 [12, p. 249], 2.6 [12, p. 249], and 2.7 respectively. Figure 2.5 shows E is a smooth function of T; however, it has a point of inflection at the critical temperature $T_c \approx 2.269k/J$. Figures 2.6 and 2.7 however, have discontinuities at T_c. The magnetization, for instance, drops suddenly from $|m| \approx 1.0$ to $|m| = 0.0$. We will try to reproduce these results computationally in Chapter 3.

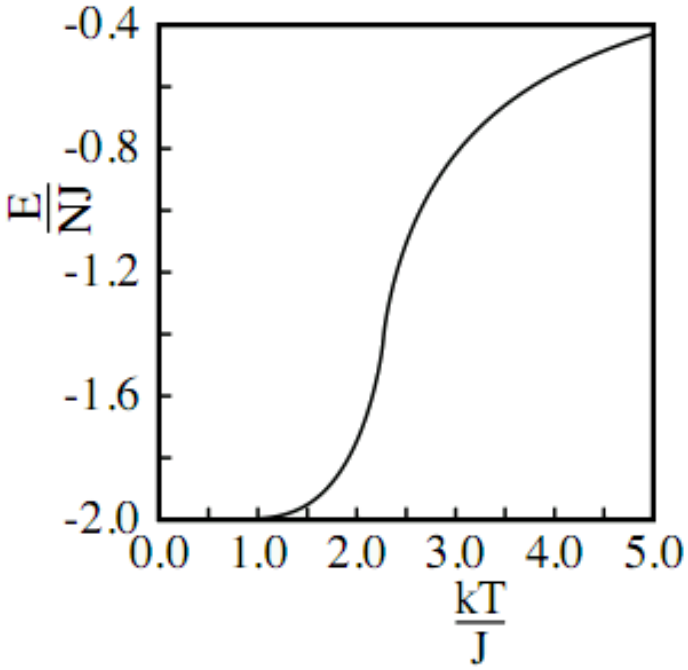


Figure 2.5 Energy as a function of dimensionless temperature in the 2D Ising model.

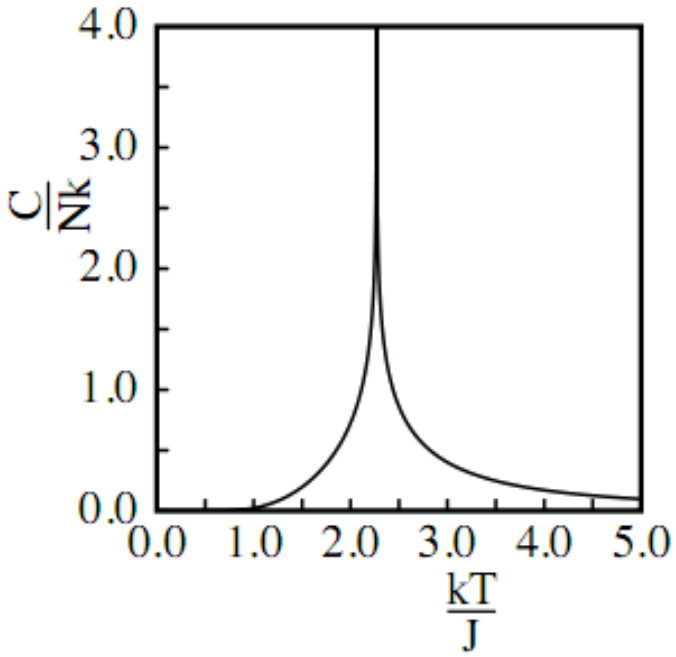


Figure 2.6 Heat capacity as a function of dimensionless temperature in the 2D Ising model.

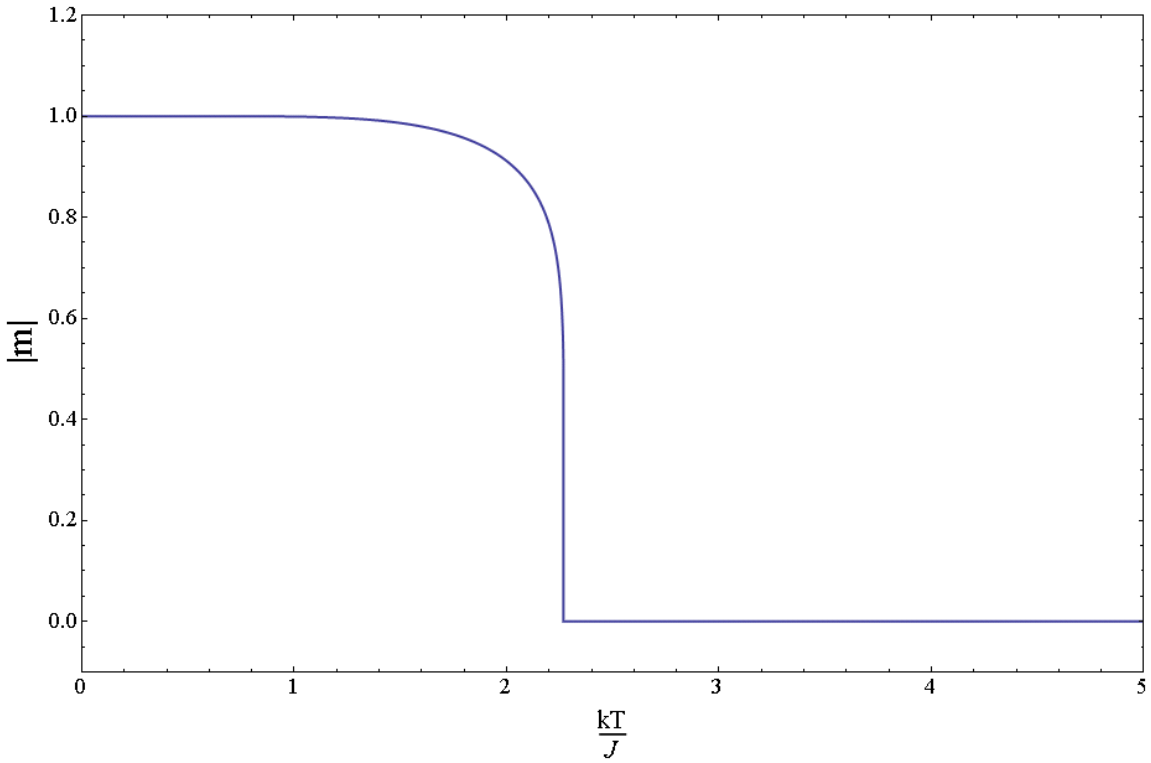


Figure 2.7 The absolute value of mean magnetization per spin as a function of dimensionless temperature in the 2D Ising model.

Due to the difficulties concerning the exact analytic solution of the Ising model, most analytic approaches to solving the model in higher dimensions usually take the form of approximate solutions. We will explore two such methods in this chapter, mean field theory and renormalization group theory.

2.5 Mean field theory

The approach used by mean field theory is to simplify the neighborhood interactions of the spins by focusing on just a single spin in the Ising lattice. In mean field theory, we assume that the other spins in the lattice are not free to change their directions independently but instead cooperatively determine the mean field at the single spin in question. If we denote the single spin of interest as s_i, and also assume that there is no external magnetic field, the expression for the energy of the single spin can be written as

$$E(s_i) = -J \sum_{j \in \text{ neighbors}} s_i s_j \tag{2.77}$$

$$= -Jn\overline{s}s_i, \tag{2.78}$$

where $\overline{s}$ is the average alignment of the neighbors and n is the number of neighboring spins. The average alignment is simply the sum of the values of the spins of the neighbors divided by the number of neighbors. For example, if the spin s_i is in a two-dimensional square lattice with 4 neighbors, and 3 of them point upwards while one points downwards, the value for $\overline{s} = (1 + 1 + 1 - 1)/4 = 1/2$. The number of neighbors, n, depends on the dimensionality and the structure of the Ising lattice. For example, a spin in a one-dimensional Ising chain will have a total of two neighbors, one to the left and one to right of itself. In two dimensions, a spin will have 4 neighbors if it resides on a square lattice and 3 if it resides on a triangular lattice. In

three dimensions, there can be many more variations. Figure 2.8 shows three different three-dimensional lattices: the simple cubic lattice, the body-centered cubic lattice, and the face centered cubic lattice. As can be seen from the figure, each spin has 6, 8 and 12 neighbors, respectively, in the three lattices that are shown.

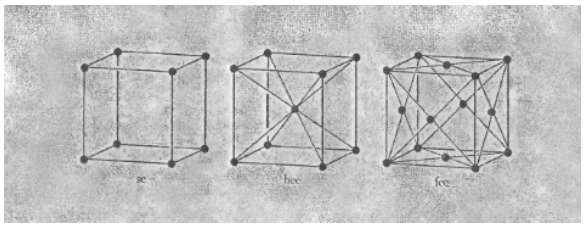


Figure 2.8 The leftmost configuration is the simple cubic lattice, with $n = 6$. In the middle is the body-centered lattice, with $n = 8$. The rightmost configuration is the face-centered lattice, with $n = 12$ [16].

The spin s_i can take the value of either -1 or 1. These are the only two possible states. Therefore, using the expression for energy obtained in Eq. 2.78, we can write down the partition function as

$$Z(s_i) = e^{-\beta J n \overline{s}} + e^{\beta J n \overline{s}} = 2\,\cosh(\beta J n \overline{s}), \tag{2.79}$$

where $\overline{s}$ not only gives us the average alignment, but also the net magnetization of the Ising lattice, since we assume, that $\overline{s}$ is representative of the spins in the entire lattice. Also, since the mean field approach assumes that the average value of the spins is the same across the lattice, we can also assume that the average alignment of the neighbors, $\overline{s}$, is equal to the average spin of s_i. Therefore, $\overline{s} = \overline{s_i}$. In order to calculate $\overline{s_i}$, we simply take a probability weighted sum of the possible values of s_i,

which are ± 1, as

$$\overline{s_i} = \sum_{s_i \pm 1} s_i \mathcal{P}(s_i) \tag{2.80}$$

$$= \sum_{s_i \pm 1} s_i \left(\frac{1}{Z_i} e^{-\beta E(s_i \pm 1)} \right) \tag{2.81}$$

$$= \frac{1}{Z_i} \left[(1) e^{\beta J n \overline{s}} + (-1) e^{-\beta J n \overline{s}} \right] \tag{2.82}$$

$$= \frac{2 \sinh(\beta J n \overline{s})}{2 \cosh(\beta J n \overline{s})} \tag{2.83}$$

$$= \tanh(\beta J n \overline{s}). \tag{2.84}$$

Now, we equate $\overline{s}$ with $\overline{s_i}$,

$$\overline{s} = \overline{s_i} = \tanh(\beta J n \overline{s}), \tag{2.85}$$

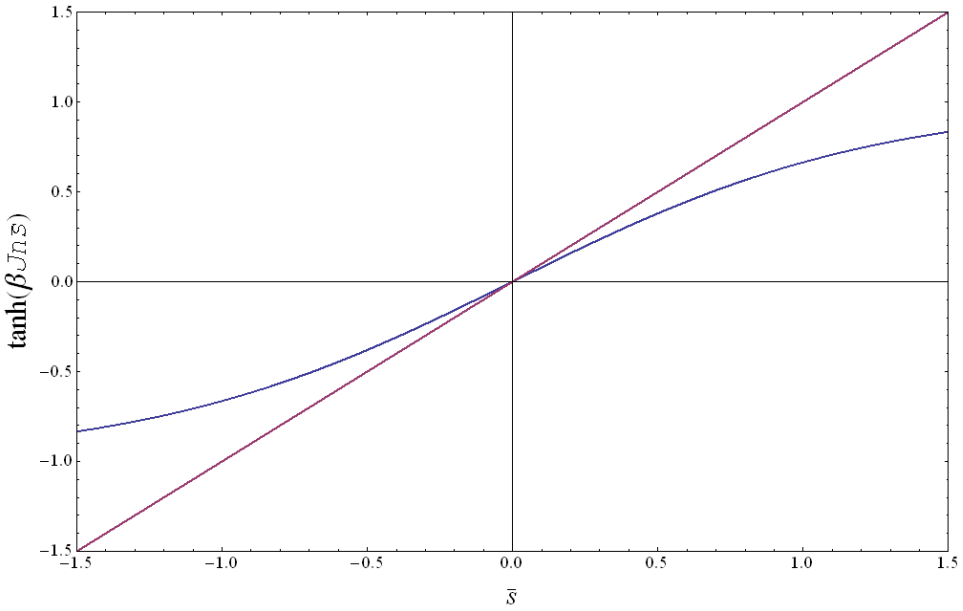
yielding a formula for $\overline{s}$ that is a function of $\overline{s}$ itself. The magnetization in the Ising lattice can therefore be determined by solving Eq. 2.85 for $\overline{s}$. This cannot be done analytically, but we can find $\overline{s}$ graphically if we plot a graph of the functions $y = \tanh(\beta J n \overline{s})$ and $y = \overline{s}$ and see where they intersect to solve for $\overline{s}$. Figure 2.9 shows the plots of these functions for two values of $\beta J n$. The first one, Fig. 2.9(a), shows the plot for $\beta J n < 1$, with $J = 1$, $n = 4$, and $\beta = 0.5$, *i.e.*, $\beta J n = 2$. The second one, Fig. 2.9(b), shows the plot for $\beta J n > 1$, with $J = 1$, $n = 4$, and $\beta = 0.2$, *i.e.*, $\beta J n = 0.8$. This means the two plots are solutions for a two-dimensional square lattice with $n = 4$, and they have the same coupling constant $J = 1$. The only difference is the temperature, which appears through $\beta = 1/kT$. A change in temperature can cause spontaneous magnetization to occur, as shown by the non-zero value of the net magnetization $\overline{s}$. In Fig. 2.9(b), we observe that the non-trivial solutions have opposite signs, but the same magnitude, which is expected from the reflection symmetry of the hyperbolic tangent function. These two non-trivial solutions, which are equal in magnitude but have oppisite signs, occur due to

the isotropy of the model. The average spin, $\overline{s}$, can have either a positive or negative solution because the spins in the Ising model can become arranged in parallel in either up or down directions, with each one being equally likely.

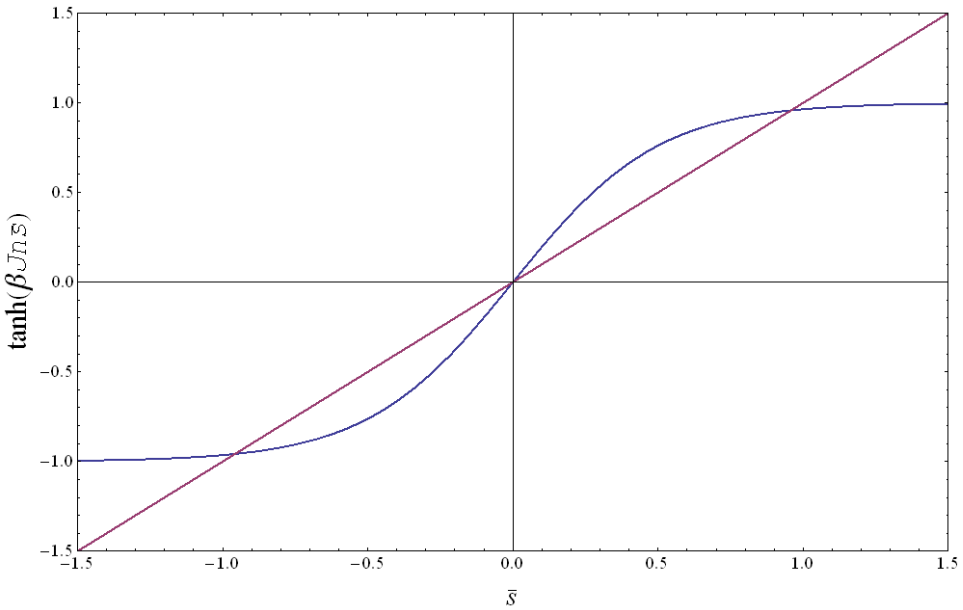
From the two graphical solutions shown in Fig. 2.9, we can see that as the value for β changes, there is a sudden transition from the three solution scenario with non-zero magnetization to a one solution scenario with zero magnetization. This represents the phase transition that is seen in ferromagnets. However, the mean-field approach is an approximation technique and has its shortcomings. For instance, notice that the method predicts that the phase transition will take place regardless of the number of nearest neighbors assumed for each spin, n, as long as the value of β changes to compensate. We know, however, that in the exact solution to the one-dimensional Ising model, no phase transition takes place. The approximation does, however, become more accurate as we increase the number of dimensions. In Chapter 3, we will test the predictions of the mean field approach for various dimensions and lattice arrangements with results from Monte Carlo simulations of the Ising model.

2.6 Renormalization group theory

The idea behind renormalization group (RG) theory is that the a system retains some properties across different length scales. If a system can be described by the same set of equations at a different length scale, but with changes in the coefficients, all we need to know is the relationship between the coefficients on one length scale and those on the other. This is called the recursion relation. It gives us a powerful approximation method for solving problems in statistical mechanics. Renormalization group theory was first developed by Kenneth Wilson in 1971 to study second-order phase transitions in statistical mechanics, such as the ferromagnetic transition in the



(a) $\beta Jn < 1$.



(b) $\beta Jn > 1$.

Figure 2.9 Graphical solution of Eq. 2.85 with two different values for $\beta Jn = 2$. There is only one trivial solution for $\beta Jn < 1$, but there are 3 solutions, one trivial and two nontrivial, for $\beta Jn > 1$.

Ising model. Wilson was awarded the Nobel prize in Physics in 1982 for his work on RG theory.

We will use renormalization group theory to study the one-dimensional and two-dimensional versions of the Ising model. We will demonstrate that the RG solution to the two-dimensional Ising model predicts a phase transition and provides a good approximate answer for the critical temperature T_c, which is close to the exact solution derived by Onsager.

2.6.1 Renormalization group for the one-dimensional Ising model

In the Ising model, the renormalization group works by summing over subgroups of spins in order to continually remove degrees of freedom. For the one-dimensional model, we go from a partition function that takes into account every spin from $s_i = 1, 2, 3, ...N$ to a partition function that skips every other spin, reducing the system to one with only $N/2$ spins, such that $s_i' = 1, 3, 5, ...\frac{N}{2} - 1$. Just as we did for the exact solution to the one-dimensional Ising model, we consider an Ising chain instead of an Ising ring.

For simplicity, we combine the coupling constant, J, and the reciprocal of the temperature $\beta = 1/kT$ into a single variable

$$K \equiv \beta J = \frac{J}{kT} \tag{2.86}$$

Thus, the partition function, Z, for the one-dimensional Ising model can be written

$$Z(K) = \sum_{\nu} e^{K(s_1 s_2 + s_2 s_3 + s_3 s_4 + s_4 s_5 + ...)}. \tag{2.87}$$

We can group the spins together and rewrite this as

$$Z(K) = \sum_{\nu} e^{K(s_1 s_2 + s_2 s_3)} e^{K(s_3 s_4 + s_4 s_5)} ... \tag{2.88}$$

$$= \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \sum_{s_3=\pm 1} ... \sum_{s_N=\pm 1} e^{K(s_1 s_2 + s_2 s_3)} e^{K(s_3 s_4 + s_4 s_5)} ... \tag{2.89}$$

$$= \sum_{s_1, s_3, s_5, ...} \sum_{s_2=\pm 1} e^{K(s_1 s_2 + s_2 s_3)} \sum_{s_4=\pm 1} e^{K(s_3 s_4 + s_4 s_5)} ..., \tag{2.90}$$

Performing the sums over the even numbered spins $s_2, s_4, ...$, we get

$$Z(K) = \sum_{s_1, s_3, s_5, ...} \left[e^{K(s_1+s_3)} + e^{-K(s_1+s_3)} \right] \left[e^{K(s_3+s_5)} + e^{-K(s_3+s_5)} \right] ... \tag{2.91}$$

We now begin our renormalization approach by assuming that there is some recursion relation which allows us to derive K' from K in the following manner:

$$e^{K(s_1+s_3)} + e^{-K(s_1+s_3)} = f(K) e^{K' s_1 s_3}. \tag{2.92}$$

Substituting the relation of Eq. 2.92 into the partition function in Eq. 2.91, we notice that we arrive at a new partition function for an Ising chain of $N/2$ spins. It is a product of the old partition function and a coefficient $f(K)$

$$Z(K, N) = \sum_{s_1, s_3, s_5, ...} f(K) e^{K' s_1 s_3} f(K) e^{K' s_3 s_5} ... \tag{2.93}$$

$$= [f(K)]^{N/2} Z(K', N/2). \tag{2.94}$$

Now, if we can derive a function relating K to K' and vice versa, we can continually rescale the Ising model by halving the number of spins each time. To do so, we must first solve for $f(K)$. If we substitute all the possible values for the spins s_1 and s_3 into Eq. 2.92, we get

$$e^{2K} + e^{-2K} = f(K) e^{K'}, \qquad \text{for } s_1 = s_3 = 1 \text{ or } s_1 = s_3 = -1, \tag{2.95}$$

and

$$2 = f(K) e^{-K'}, \qquad \text{for } s_1 = -1 \text{ and } s_3 = 1 \text{ or } s_1 = 1 \text{ and } s_3 = -1. \tag{2.96}$$

This means $f(K) = 2e^{K'}$, which, when substituted back into Eq. 2.95, yields

$$e^{2K} + e^{-2K} = 2e^{2K'} \tag{2.97}$$

$$e^{2K'} = \cosh(2K) \tag{2.98}$$

$$K' = \frac{1}{2}\ln\left[\cosh(2K)\right]. \tag{2.99}$$

Alternatively, we can solve for K to get

$$K = \frac{1}{2}\cosh^{-1}(e^{2K'}). \tag{2.100}$$

Using these relations between K and K', we now compute some iterations for K, starting with $K = 0.01$, as shown in Table 2.1. Notice that $K' < K$, as shown in Fig. 2.10. Also, we can see that K' diverges as we take the limit $K \to \infty$ and converges to 0 as $K \to 0$. This means that the iterative procedure for K' will eventually settle at either of two fixed points, $K = 0$ or $K = \infty$. This is indicative that the Ising model in one-dimension does not have a non-trivial critical point at which a phase transition occurs. The only critical point is at $T = 0$.

2.6.2 Renormalization group for the two-dimensional Ising model

When applying the renormalization group approach to the two-dimensional Ising model, we perform a similar procedure of eliminating half of the spins in each iteration of the renormalization process. We start with the setup in Fig. 2.11, where we eliminate every other spin in the rectangular grid. As a result, we end up with a grid that is rotated at a 45° angle, in which the nearest neighbors are connected by the dotted lines.

Just as in the one-dimensional model, we start by writing the partition function, Z, with the spins grouped together, four at a time, according to Fig. 2.11. We look

Table 2.1 Iterative application of Eq. 2.100 to calculate K values.

Iterations	K'	K
1	0.010000	0.100334
2	0.100334	0.327447
3	0.327447	0.636247
4	0.636247	0.972710
5	0.972710	1.316710
6	1.316710	1.662637
7	1.662637	2.009049
8	2.009049	2.355582
9	2.355582	2.702146
10	2.702146	3.048717

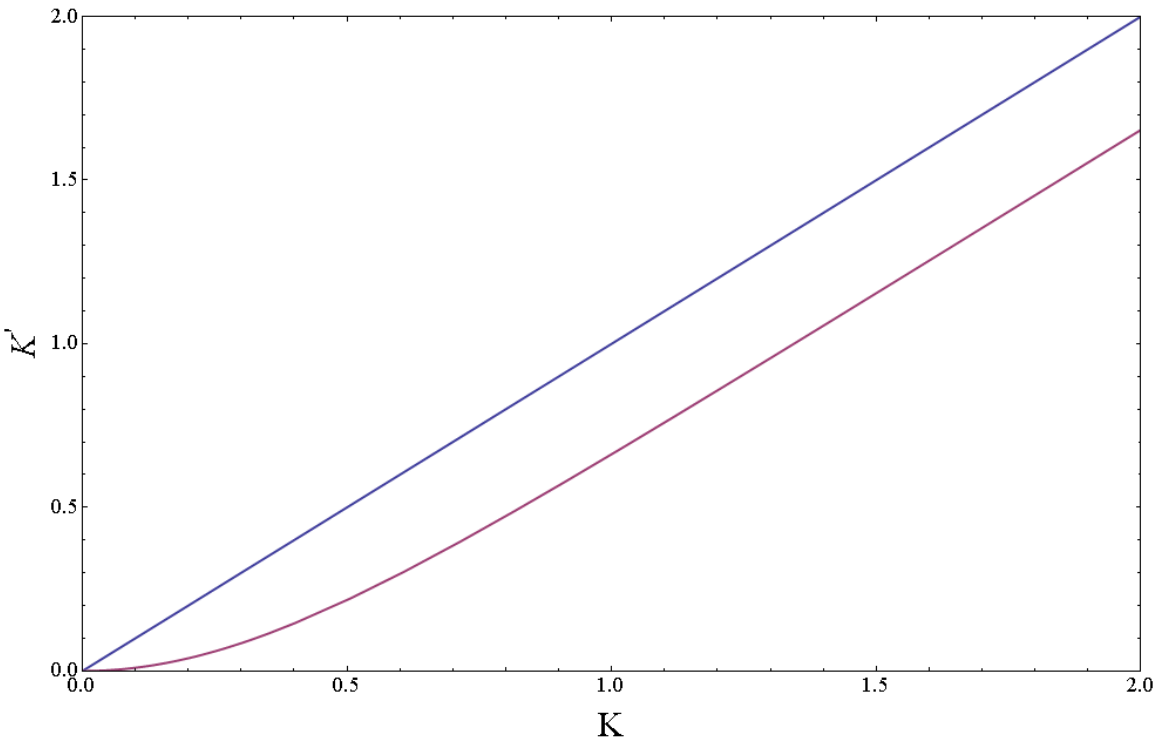


Figure 2.10 Plot of K' as a function of K for the one-dimensional Ising renormalization group. Notice $K' < K$, and K' diverges as we take the limit $K \to \infty$ and converges to 0 as $K \to 0$.

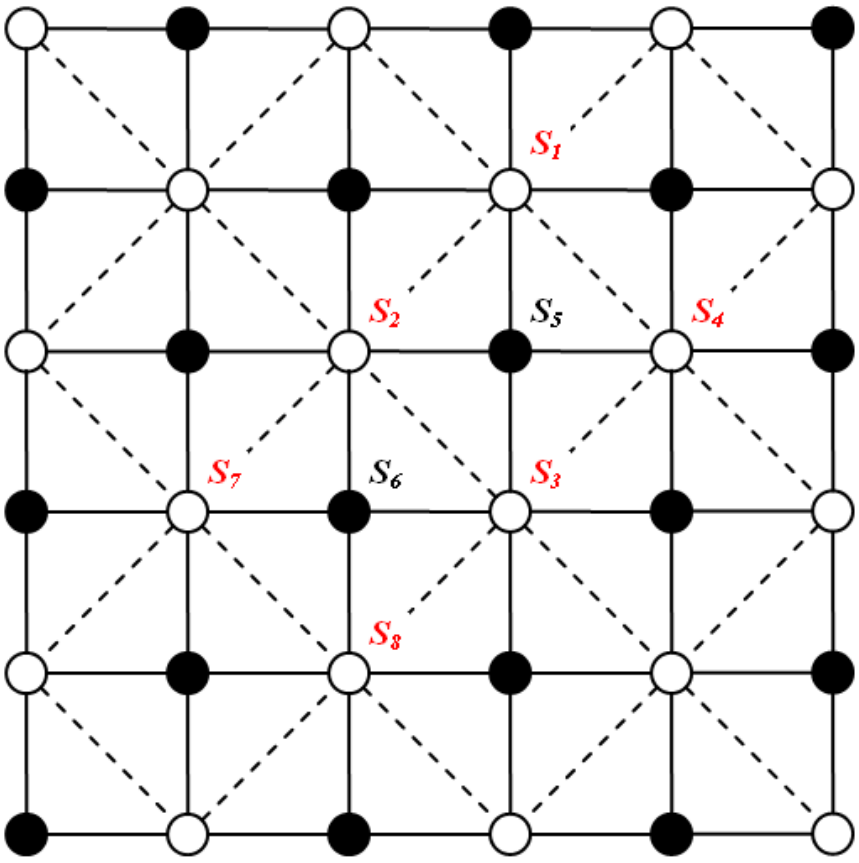


Figure 2.11 Renormalization group applied to the two-dimensional Ising model, every other spin is eliminated, leaving a sparser grid represented by the white circles connected by the dotted lines.

at the interaction between a particular spin and its nearest neighbors. For example, s_5 has neighbors s_1, s_2, s_3 and s_4. Thus, we have

$$Z(K) = \sum_{\nu} e^{K(s_1s_2+s_2s_3+s_3s_4+s_4s_5+...)} \tag{2.101}$$

$$= \sum_{\nu} e^{Ks_5(s_1+s_2+s_3+s_4)} e^{Ks_6(s_2+s_3+s_7+s_8)} ... \tag{2.102}$$

$$= \sum_{s_1=\pm1} \sum_{s_2=\pm1} \sum_{s_3=\pm1} ... \sum_{s_N=\pm1} e^{Ks_5(s_1+s_2+s_3+s_4)} e^{Ks_6(s_2+s_3+s_7+s_8)} ... \tag{2.103}$$

$$= \sum_{\text{remaining } s_i'\text{s}} \sum_{s_5=\pm1} e^{Ks_5(s_1+s_2+s_3+s_4)} \sum_{s_6=\pm1} e^{Ks_6(s_2+s_3+s_7+s_8)} .... \tag{2.104}$$

Summing over the spins s_5, s_6, etc, we get

$$Z(K) = \sum_{\text{remaining } s_i'\text{s}} \left[e^{K(s_1+s_2+s_3+s_4)} + e^{-K(s_1+s_2+s_3+s_4)}\right] \left[e^{K(s_2+s_3+s_7+s_8)} + e^{-K(s_2+s_3+s_7+s_8)}\right] ... \tag{2.105}$$

Just like in the one-dimensional model, we assume that the Boltzmann factors can be converted across length scales in the following way

$$e^{K(s_1+s_2+s_3+s_4)} + e^{-K(s_1+s_2+s_3+s_4)} = f(K) e^{K'(s_1s_2+s_2s_3+s_3s_4+s_4s_1)}. \tag{2.106}$$

Since each spin s_i can be $+1$ or -1, there are $2^4 = 8$ possible arrangements for the spins s_1 through s_4 in Eq. 2.106. Since the model does not change when we interchange the directions of the all spins, we can cut down the number of possibilities by half to 4. Substituting in these spin values will give us four simultaneous equations. However, we need to solve for only two unknowns, $f(K)$ and K'. Therefore, we need to increase the number of degrees of freedom for there to be the four equations and four unknowns.

First, we look at the possible arrangements of the spins. As shown in Fig. 2.12, the spins are either

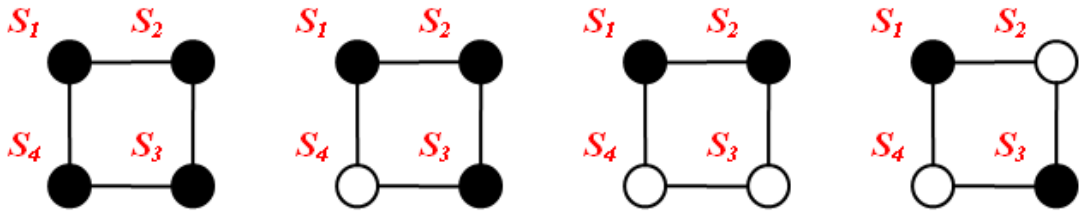


Figure 2.12 Possible arrangement of spins for two-dimensional Ising renormalization. Circles with the same color represent spins aligned in the same direction.

- all in the same direction ($s_1 = s_2 = s_3 = s_4 = \pm 1$),
- have three in one direction and one in the opposite direction ($s_1 = s_2 = s_3 = -s_4 = \pm 1$),
- have two pairs of spins that are adjacent to one another in same direction ($s_1 = s_2 = -s_3 = -s_4 = \pm 1$), or
- have two pairs of spins that are opposite each other across the diagonal in the same direction ($s_1 = -s_2 = s_3 = -s_4 = \pm 1$).

Now, we have to consider the problem of only having two unknowns. To fix this, we separate the original K' in three different variables: K_1 K_2 and K_3. K_1 deals with interactions between the nearest-neighbors. It is important when we are deriving the recursion relation to make sure that each nearest neighbor pair taken into account exactly once. If we look at Fig. 2.11, we see that when summing over the nearest neighbors for the reduced squares $s_1 s_2 s_3 s_4$ and $s_2 s_3 s_8 s_7$, the pair $s_3 s_3$ is counted twice. Therefore, we will need to multiply K_1 by 1/2. K_2 deals with the interaction between the next-nearest-neighbors. For example in Fig. 2.11, s_1 and s_3 would interact with coefficient K_2. K_3 deals with the interaction involving all the spins around a square, such as that formed by s_1, s_2, s_3, and s_4 in Fig. 2.11. This changes the recursion

relation for the Boltzmann factors from Eq. 2.106 to

$$e^{K(s_1+s_2+s_3+s_4)} + e^{-K(s_1+s_2+s_3+s_4)} = f(K)\left[e^{\frac{K_1}{2}(s_1s_2+s_2s_3+s_3s_4+s_4s_1)+K_2(s_1s_3+s_2s_4)+K_3(s_1s_2s_3s_4)}\right]. \tag{2.107}$$

Substituting the possible spin arrangements in Fig. 2.12 into Eq. 2.107, we have

$$e^{4K} + e^{-4K} = f(K)e^{2K_1+2K_2+K_3}, \tag{2.108}$$

$$e^{2K} + e^{-2K} = f(K)e^{-K_3}, \tag{2.109}$$

$$2 = f(K)e^{-2K_2+K_3}, \tag{2.110}$$

and

$$2 = f(K)e^{-2K_1+2K_2+K_3}. \tag{2.111}$$

Now we solve this system of equations. First, we equate the right hand sides of Eqs. 2.110 and 2.111 to get

$$f(K)e^{-2K_2+K_3} = f(K)e^{-2K_1+2K_2+K_3} \tag{2.112}$$

$$\tag{2.113}$$

which implies that

$$-2K_2 = -2K_1 + 2K_2 \tag{2.114}$$

$$K_1 = 2K_2. \tag{2.115}$$

Next we solve for $f(K)$ in terms of K_3 using Eq. 2.109, as

$$e^{2K} + e^{-2K} = f(K)e^{-K_3} \tag{2.116}$$

$$2\cosh(2K) = f(K)e^{-K_3} \tag{2.117}$$

$$f(K) = 2e^{K_3}\cosh(2K). \tag{2.118}$$

Substituting Eq. 2.118 into Eq. 2.110, we get

$$2 = 2\cosh(2K)e^{K_3}e^{-2K_2+K_3} \tag{2.119}$$

$$\frac{1}{\cosh(2K)} = e^{-2K_2+2K_3}. \tag{2.120}$$

Substituting Eq. 2.118 into Eq. 2.108, we get

$$e^{4K} + e^{-4K} = 2\cosh(2K)e^{K_3}e^{2K_1+2K_2+K_3} \tag{2.121}$$

$$\cosh(4K) = \cosh(2K)e^{6K_2+2K_3}, \text{(substituting } K_1 = 2K_2) \tag{2.122}$$

$$\frac{1}{\cosh(2K)} = \frac{1}{\cosh(4K)}e^{6K_2+2K_3} \tag{2.123}$$

$$e^{-2K_2+2K_3} = \frac{1}{\cosh(4K)}e^{6K_2+2K_3}, \text{(substituting from Eq. 2.120)} \tag{2.124}$$

$$-2K_2 + 2K_3 = \ln\left(\frac{1}{\cosh(4K)}\right) + 6K_2 + 2K_3 \tag{2.125}$$

$$-8K_2 = \ln\left(\frac{1}{\cosh(4K)}\right) \tag{2.126}$$

$$K_2 = \frac{1}{8}\ln\left(\cosh(4K)\right). \tag{2.127}$$

Thus, by Eq. 2.115

$$K_1 = \frac{1}{4}\ln\left(\cosh(4K)\right). \tag{2.128}$$

Now, we multiply Eqs. 2.110 and 2.111 to get

$$[f(K)]^2 e^{2K_1-2K_3} = 4 \tag{2.129}$$

$$f(K) = 2\left(e^{2K_1-2K_3}\right)^{1/2} \tag{2.130}$$

$$f(K) = 2e^{K_1-K_3} \tag{2.131}$$

$$2e^{K_3}\cosh(2K) = 2e^{-K_1+K_3}, \text{(substituting from Eq. 2.118)} \tag{2.132}$$

$$\cosh(2K) = e^{K_1-2K_3} \tag{2.133}$$

$$\ln\left(\cosh(2K)\right) = K_1 - 2K_3 \tag{2.134}$$

$$K_3 = \frac{1}{2}\left[\frac{1}{4}\ln\left(\cosh(4K)\right) - \ln\left(\cosh(2K)\right)\right] \tag{2.135}$$

$$K_3 = \frac{1}{8}\ln\left(\cosh(4K)\right) - \frac{1}{2}\ln\left(\cosh(2K)\right). \tag{2.136}$$

Finally, we solve for $f(K)$ by substituting K_3 from Eq. 2.136 into Eq. 2.118, to get

$$f(K) = 2e^{\left[\frac{1}{8}\ln(\cosh(4K)) - \frac{1}{2}\ln(\cosh(2K))\right]}\cosh(2K) \tag{2.137}$$

$$f(K) = 2\left[\cosh(2K)\right]^{1/2}\left[\cosh(4K)\right]^{1/8}. \tag{2.138}$$

We have solved for all the components of Eq. 2.107. Just as we did in the one-dimensional model in Eq. 2.94, it is now possible to write a recursion relation for the partition function that accounts for half the number of spins, as

$$Z(K,N) = [f(K)]^{N/2} \tag{2.139}$$

$$\times \sum_{\text{remaining } s_i's} e^{\frac{K_1}{2}(s_1s_2+s_2s_3+s_3s_4+s_4s_1)+K_2(s_1s_3+s_2s_4)+K_3(s_1s_2s_3s_4)} \tag{2.140}$$

$$\times\, e^{\frac{K_1}{2}(s_2s_3+s_3s_8+s_7s_8+s_7s_2)+K_2(s_2s_8+s_7s_3)+K_3(s_2s_7s_8s_3)}\ldots \tag{2.141}$$

$$\sum_{N \text{ spins}} e^{\sum_{\langle ij\rangle} s_is_j} = [f(K)]^{N/2} \sum_{N/2 \text{ spins}} e^{\left[K_1\sum_{\langle ij\rangle} s_is_j + K_2\sum_{\langle\langle ij\rangle\rangle} s_is_j + K_3\sum_{\langle ijkl\rangle} s_is_js_ks_l\right]}, \tag{2.142}$$

where $\langle ij\rangle$ represents sums over the nearest neighbors, $\langle\langle ij\rangle\rangle$ represents sums over the next-nearest neighbors, and $\langle ijkl\rangle$ represents sums over the spins around a square.

In Eq. 2.142, we no longer need to multiply K_1 by $1/2$ because we are consciously summing over all the nearest neighbors only once.

We can approximate the partition function $Z(N, K)$ in terms of $Z(N/2, K)$ with varying degrees of accuracy by taking into account the sums over these different types of neighboring spins. Now, we will attempt such an approximation by taking into account K_1 and K_2, *i.e.*, the nearest neighbors and next-nearest neighbor sums, respectively. To do so, we first write the approximation

$$K_1 \sum_{\langle ij \rangle} s_i s_j + K_2 \sum_{\langle\langle ij \rangle\rangle} s_i s_j \approx K'(K_1, K_2) \sum_{\langle ij \rangle} s_i s_j. \tag{2.143}$$

Using this approximation, the recursion relation for the partition function from Eq. 2.142 becomes

$$Z(K, N) = [f(K)]^{N/2} Z\left[K'(K_1, K_2), N/2\right]. \tag{2.144}$$

Now, we need to derive an approximate relationship between K' and K_1 and K_2. To do so, consider the spins arranged in a two-dimensional square lattice as in Fig. 2.13. Each dotted line segment connecting two white circles represents a complete nearest-neighbor bond, each dotted line segment that is connected to only one white circle represents half a nearest-neighbor bond. Thus, we can see that for a lattice of $N/2 = 4$ spins, such as the four white circles, there are $4 + (1/2)8 = 8 = N$ nearest-neighbor bonds. In addition, each line segment connecting a white circle and a black circle represents half a next-nearest-neighbor bond. We can see that there are $N = 8$ next-nearest-neighbor bonds. Therefore, if we assume that all of these 4 spins were aligned in the +1 direction,

$$K_1 \sum_{\langle ij \rangle} s_i s_j + K_2 \sum_{\langle\langle ij \rangle\rangle} s_i s_j = NK_1 + NK_2 \tag{2.145}$$

$$K'(K_1, K_2) \sum_{\langle ij \rangle} s_i s_j = NK' \approx NK_1 + NK_2. \tag{2.146}$$

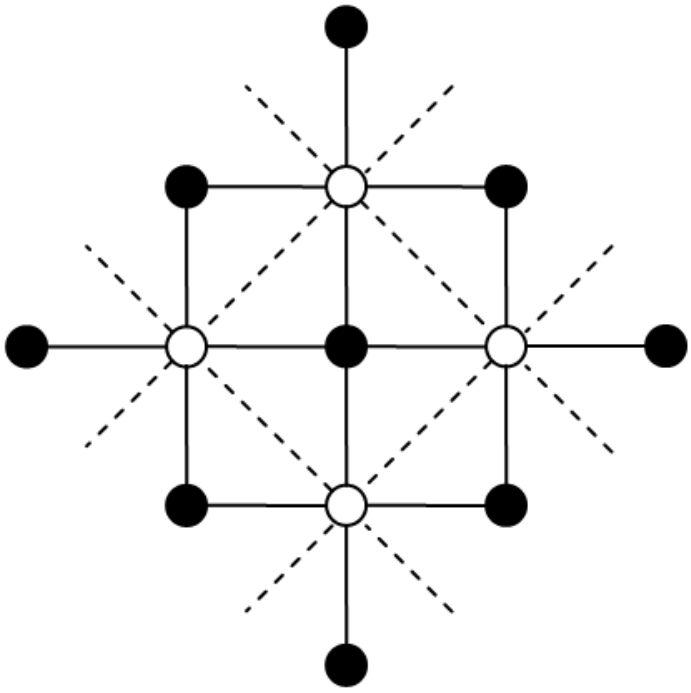


Figure 2.13 Arrangement of bonds in a two-dimensional square lattice. There are 4 spins (white circles) connected together by 8 nearest-neighbor and 8 next-nearest-neighbor bonds.

Substituting K_1 and K_2 from Eqs. 2.128 and 2.127, we can write K' as

$$K' = \frac{3}{8}\ln\left[\cosh(4K)\right] \tag{2.147}$$

Thus, we have finally derive the recursion relation for K' in the two-dimensional Ising model. Inversely, we can derive K as

$$K = \frac{1}{4}\cosh^{-1}\left(e^{8K'/3}\right). \tag{2.148}$$

The interesting fact about this solution for the renormalization group approximation of the two-dimensional Ising model is that, unlike the one-dimensional model which yielded only two trivial fixed points as $T \to 0$ and $T \to \infty$, the two-dimensional version has a non-trivial fixed point. This is made clear in Table 2.2 and Fig. 2.14, which shows the iterative application of Eq. 2.148 and a plot of K' as function of K

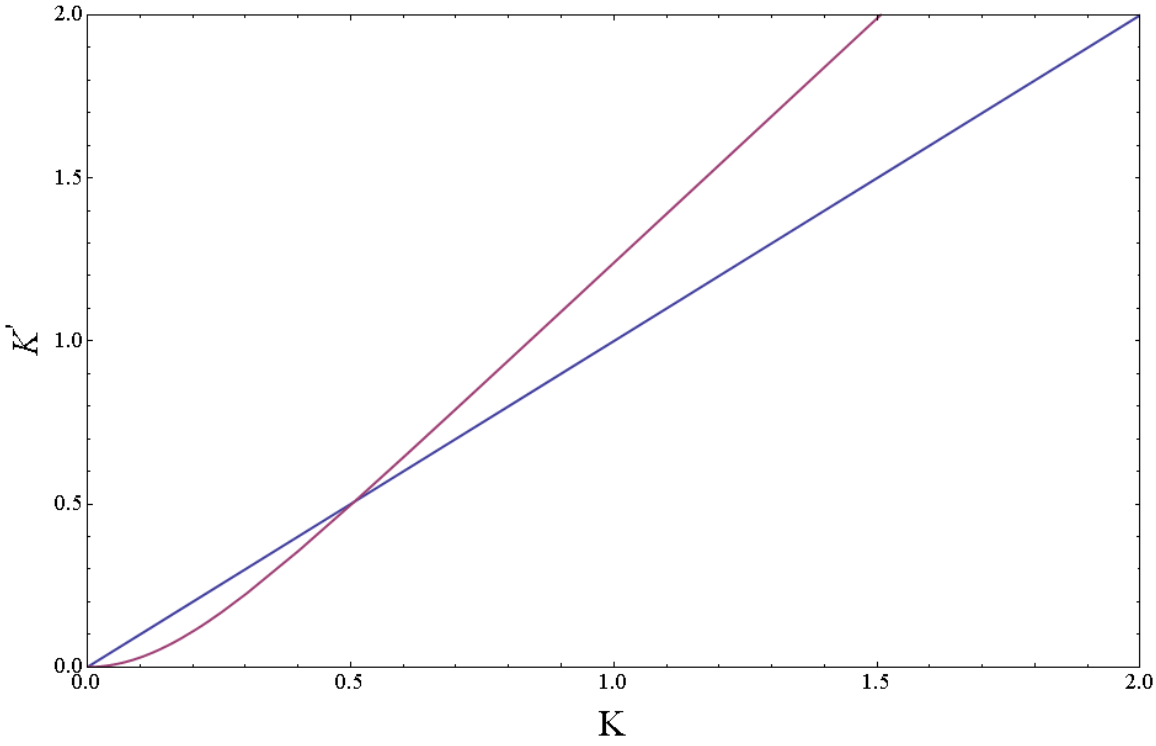


Figure 2.14 Plot of K' as a function of K for two-dimensional Ising renormalization group. Notice the non-trivial fixed point at $K = 0.50698$.

respectively. It can be seen that the non-trivial fixed point is $K_c = 0.50698$. This is the critical value of K for which the phase transition occurs.

Since $K \equiv \frac{J}{kT}$, we can calculate the value for the critical temperature predicted by the renormalization group approach by simply taking the inverse of K. Thus, $T_c = 1.972\ k/J$, which is close to the exact Onsager solution of $T_c = 2.269\ k/J$.

Table 2.2 Iterative application of Eq. 2.148 to calculate K values.

Iterations	K'	K
1	0.500000	0.502160
2	0.502160	0.503652
3	0.503652	0.504683
4	0.504683	0.505394
5	0.505394	0.505886
6	0.505886	0.506225
7	0.506225	0.506459
8	0.506459	0.506621
9	0.506621	0.506732
10	0.506732	0.506809
11	0.506809	0.506863
12	0.506863	0.506899
13	0.506899	0.506925
14	0.506925	0.506942
15	0.506942	0.506954
16	0.506954	0.506962
17	0.506962	0.506968
18	0.506968	0.506972
19	0.506972	0.506975
20	0.506975	0.506977

Chapter 3

Computational Solutions to the Ising Model

3.1 Monte Carlo methods and the Metropolis-Hastings algorithm

One of the principal methods for studying the Ising model and many other models in statistical mechanics is Monte Carlo simulation. The main use of Monte Carlo methods is for problems in which the state space of the model under consideration is too large to allow practical computation of the probabilities of all the different possible states. For example, in the Ising model with N spins, there are 2^N possible microstates, so even for a small two-dimensional square lattice of 20×20 spins, the possible number of microstates is $2^{20^2} \approx 10^{120}$. Therefore, we use the Monte-Carlo approach, which takes a random sampling of as many states as possible, compute the Boltzmann factors for these randomly sampled states, and from them compute the desired thermodynamic variables, such as the magnetization and energy.

However, another glance at the number of microstates in the Ising model shows

that a purely random Monte Carlo sampling of the possible microstates will not be adequate. Consider the example above of 10^{120} for the 20×20 square lattice. Any reasonably sized random sampling that samples each microstate with equal probability will be a minute sample of the possible state space. Therefore, we need a smarter way of sampling the microstates that are the most important for our study.

In 1953, Nicholas Metropolis et al. published a paper in which they outlined a Monte Carlo method using a sampling procedure that favored low energy states [17]. For any given lattice, there is a much larger number of possible microstates at higher energies than is at lower energies, so an algorithm that favors lower energy states systematically reduces the number of microstate that will be sampled in the Monte Carlo simulation. Metropolis's original method was specifically designed for systems that obey Boltzmann statistics. In 1970, W. K. Hastings generalized the algorithm [14]. This method is now called the Metropolis-Hastings algorithm and has been applied to various problems across the natural and social sciences.

We will be concerned primarily with the application of the Metropolis algorithm to Ising-type models that can be described by Boltzmann statistics. For the Ising model, the main routine of the algorithm is as

1. Start with any random microstate, *i.e.*, a randomly assigned arrangement of spins on a lattice.

2. Pick a spin at random on the lattice, flip it, and compute the energy difference ΔE caused by this flip.

3. If $\Delta E \leq 0$, keep this flip.

4. If $\Delta E > 0$, keep this flip with probability given by $e^{-\Delta E/kT}$, otherwise undo the flip.

5. Start again by picking another spin at random.

By iterating this procedure many times, so that each spin on the grid will, on average, have been flipped multiple times, the system will settle to a microstate of low energy. It can readily be proven that the algorithm will converge to a fixed point W [6], where W is a probability distribution of equilibrium microstates. Before we procede with the proof, we have to define a few new concepts and terms.

First is the concept of strong ergodicity. It is simply the condition that any microstate of the system is accessible by the Metropolis algorithm from any other microstate. In other words, given any two microstates ν and μ, the probability $\mathcal{P}(\mu \leftarrow \nu) > 0$. We can see that the Metropolis algorithm satisfies strong ergodicity by considering the following argument: any two given microstates of a lattice will be different by a finite number of spins pointing in a different direction in each microstate. Since the Metropolis algorithm has a non-zero probability of sampling each spin and flipping it in the opposite direction, we can see that there is a non-zero probability that the algorithm will sample all the spins that account for the difference between the two microstates and in the process get from one microstate to the other.

Next, we define a probability distribution $Q(\nu)$ that is a function of the microstate ν of the Ising lattice, where

$$\sum_{\nu} Q(\nu) = 1. \tag{3.1}$$

Then we define an evolution probability $\mathcal{P}(\mu \leftarrow \nu)$ to transform the microstate a to the microstate b. These kinds of transformations satisfy the condition

$$Q'(\mu) = \sum_{\mu} \mathcal{P}(\mu \leftarrow \nu) Q(\nu), \tag{3.2}$$

In shorthand notation, we will write $\mathcal{P}Q$ to represent the evolution probability $\mathcal{P}$ acting on distribution Q. The desired fixed point W is also a probability distribution,

except $\mathcal{P}W = W$, which means that once the system reaches this distribution, it stays there permanently.

Now we are ready to present the proof for the convergence of the Metropolis algorithm. It is taken from a paper by Bhanot [6].

We will show that starting from any microstate, we will always reach a fixed point W by following the Metropolis algorithm. First, we define a distance, $d(Q_1, Q_2)$, between two points in the space of probability distributions, Q_1 and Q_2, as

$$d(Q_1, Q_2) = \sum_a |Q_1(\nu) - Q_2(\nu)|. \tag{3.3}$$

The metric d has to satisfy the following conditions before it can be regarded as a distance:

$$d(Q_1, Q_1) = 0 \tag{3.4}$$

$$d(Q_1, Q_2) > 0, \text{ for } Q_1 \neq Q_2 \qquad \text{(positivity)} \tag{3.5}$$

$$d(Q_1, Q_2) = d(Q_1, Q_2) \qquad \text{(symmetry)} \tag{3.6}$$

$$d(Q_1, Q_2) + d(Q_2, Q_3) \geq d(Q_2, Q_3) \qquad \text{(transitivity)} \tag{3.7}$$

The first three conditions are obviously satistfied by d. We now show that d also satisfies the fourth condition of transitivity using the triangle inequality. We have

$$d(Q_1, Q_3) = \sum_a |Q_1(\nu) - Q_2(\nu) + Q_2(\nu) - Q_3(\nu)| \tag{3.8}$$

$$\leq \sum_a |Q_1(\nu) - Q_2(\nu)| + \sum_a |Q_2(\nu) - Q_3(\nu)| \tag{3.9}$$

$$= d(Q_1, Q_2) + d(Q_2, Q_3). \tag{3.10}$$

The proof follows from the satisfaction of two theorems, one of which shows that the Metropolis algorithm acting on two distributions always decreases the distance between them, and another which shows that the distance between any two distributions can be made arbitrarily small by repeated application of the Metropolis algorithm.

Theorem 1: If $\mathcal{P}$ satisfies the strong ergodicity condition, $\mathcal{P} > 0$, then it defines a contraction mapping with respect to the metric d. That is

$$d(PQ_1, PQ_2) \leq \alpha d(Q_1, Q_2) \tag{3.11}$$

for all Q_1, Q_2, and with $\alpha < 1$.

Proof of theorem 1: If $Q_1 = Q_2$, the theorem is true according to the condition defined in Eq. 3.4. If $Q_1 \neq Q_2$, define

$$\Delta Q(\nu) \equiv Q_1(\nu) - Q_2(\nu). \tag{3.12}$$

We have

$$d(PQ_1, PQ_2) = \sum_{\nu} |\mathcal{P}Q_1(\nu) - \mathcal{P}Q_2(\nu)| \tag{3.13}$$

$$= \sum_{\nu} \left| \sum_{\mu} [\mathcal{P}(\nu \leftarrow \mu)Q_1(\mu) - \mathcal{P}(\nu \leftarrow \mu)Q_2(\mu)] \right| \tag{3.14}$$

$$= \sum_{\nu} \left| \sum_{\mu} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) \right|. \tag{3.15}$$

We break up the sum $\sum_b$ into two parts:

$$\sum_{\mu} = \sum_{\mu^+} + \sum_{\mu^-} \tag{3.16}$$

where b^+ and b^- are regions where ΔQ is positive or negative, respectively. Then

$$d(PQ_1, PQ_2) = \sum_{\nu} \left| \sum_{\mu^+} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) + \sum_{\mu^-} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) \right| \tag{3.17}$$

$$= \sum_{\nu} \left(\sum_{\mu^+} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) - \sum_{\mu^-} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) \right) \tag{3.18}$$

$$- 2 \sum_{\nu} \min_{\pm} \left| \sum_{\mu^+, \mu^-} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) \right|. \tag{3.19}$$

using the identity $||x| - |y|| = |x| + |y| - 2[\min(|x|, |y|)]$)

$$d(PQ_1, PQ_2) = \sum_{\nu} \sum_{b} \mathcal{P}(\nu \leftarrow \mu)|\Delta Q(\mu)| - 2\sum_{\nu} \min_{\pm} \left| \sum_{\mu^+,\mu^-} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) \right| \tag{3.20}$$

$$= \sum_{\mu} |\Delta Q(\mu)| - 2\sum_{\nu} \min_{\pm} \left| \sum_{\mu^+,\mu^-} \mathcal{P}(\nu \leftarrow \mu)\Delta Q(\mu) \right| \tag{3.21}$$

$$\leq \sum_{\mu} |\Delta Q(\mu)| - 2\sum_{\nu} \min_{\pm} \left(\mathcal{P}^{\nu}_{\min} \left| \sum_{\mu^+,\mu^-} \Delta Q(\mu) \right| \right) \tag{3.22}$$

where $\mathcal{P}^{a}_{\min} = \min[\mathcal{P}(\nu \leftarrow \mu)] > 0$.

Now, using Eqs. 3.1 and 3.12, we get

$$\sum_{\mu^+} \Delta Q(\mu) + \sum_{\mu^-} \Delta Q(\mu) = \sum_{\mu} Q_1(\mu) - \sum_{\mu} Q_2(\mu) = 0. \tag{3.23}$$

Hence

$$\left| \sum_{\mu^+} \Delta Q(\mu) \right| = \left| \sum_{\mu^-} \Delta Q(\mu) \right| = \frac{1}{2} \sum_{\mu} |\Delta Q(\mu)|. \tag{3.24}$$

Noting that $\sum_b |\Delta Q(\mu)| = d(Q_1, Q_2)$, and combining Eqs. 3.22 to 3.24, we get

$$d(\mathcal{P}Q_1, \mathcal{P}Q_2) \leq \left(1 - \sum_{\nu} \mathcal{P}^{\nu}_{\min} \right) d(Q_1, Q_2) \tag{3.25}$$

which proves the theorem with

$$0 < \alpha \leq 1 - \mathcal{P}^{\nu}_{\min}. \tag{3.26}$$

Now we proceed to the next stage of the proof which shows that repeated applications of $\mathcal{P}$ to Q will lead to convergence, *i.e.*, $\mathcal{P}^{\infty}Q = W$.

Theorem 2: The sequence, $(Q, \mathcal{P}Q, \mathcal{P}^2Q, ..., \mathcal{P}^nQ, ..., \mathcal{P}^{n'}Q, ...)$ is such that for any arbitrarily small ε, there exists an N such that for all n,$n' > N$

$$0 \leq d(\mathcal{P}^nQ, \mathcal{P}^{n'}Q) < \varepsilon. \tag{3.27}$$

Proof of theorem 2: If Q is preserved by $\mathcal{P}$, the proof is obvious. Otherwise, by repeated application of the transitivity condition defined in Eq. 3.7,

$$d(\mathcal{P}^n Q, \mathcal{P}^{n'} Q) \leq d(\mathcal{P}^N Q, \mathcal{P}^n Q) + d(\mathcal{P}^N Q, \mathcal{P}^{n'} Q) \tag{3.28}$$

$$\leq \sum_{k=0}^{n-N-1} d(\mathcal{P}^{N+k} Q, \mathcal{P}^{N+k+1} Q) + \sum_{k=0}^{n'-N-1} d(\mathcal{P}^{N+k'} Q, \mathcal{P}^{N+k'+1} Q) \tag{3.29}$$

$$\leq \sum_{k=0}^{n-N-1} \alpha^{N+k} d(Q, \mathcal{P}Q) + \sum_{k=0}^{n'-N-1} \alpha^{N+k'} d(Q, \mathcal{P}Q) \tag{3.30}$$

$$= \alpha^N d(Q, \mathcal{P}Q) \left(\sum_{k=0}^{n-N-1} \alpha^k + \sum_{k=0}^{n'-N-1} \alpha^{k'} \right) \tag{3.31}$$

$$= \alpha^N d(Q, \mathcal{P}Q) \left(\frac{2 - \alpha^{n-N} - \alpha^{n'-N}}{1-\alpha} \right). \tag{3.32}$$

Hence

$$d(\mathcal{P}^n Q, \mathcal{P}^{n'} Q) \leq \frac{2\alpha^N}{1-\alpha} d(Q, \mathcal{P}Q). \tag{3.33}$$

Thus, with

$$N \geq \ln \left(\frac{\varepsilon(1-\alpha)}{2d(Q, \mathcal{P}Q)} \right) (\ln \alpha)^{-1} \tag{3.34}$$

we have

$$d(\mathcal{P}^n Q, \mathcal{P}^{n'} Q) < \varepsilon \text{ for all } \varepsilon > 0, \tag{3.35}$$

which proves theorem 2, thus completing the proof of the convergence of the Metropolis algorithm.

In the rest of this chapter, we will look at simulation results for the Ising model in two and three dimensions, and also look at an alternative way of applying renormalization, which uses the output generated from the Metropolis algorithm.

A brief note on the methodology of the Monte Carlo simulations presented in this chapter is worth mentioning. The simulations are all started from a lattice with

randomly assigned spins. Each step of a Monte Carlo simulation picks a spin at random from the lattice with equal probability. In order for the simulation to better approximate the true randomness of this uniform distribution, code for a special random number generator with a period of 2^{144} was used [8]. This ensures that we will not be sampling the same sequence of spins repeatedly when running simulations with large N for many iterations.

The number of iterations us organized into sweeps, with one sweep consisting of N iterations for a lattice of N spins. This means that in one sweep, the expected number of times any particular spin has been flipped is 1. Each time a macro-parameter such as temperature or size of the external field is changed, the simulation is allowed to run for $10,000$ sweeps prior to collecting any data to allow for the simulation to reach equilibrium. After this, data collection to measure macro-parameters such as magnetization and energy is conducted at the end of each sweep, after which the measures are averaged across the sweeps.

3.2 Simulation results for the two-dimensional Ising model

The simulations for the two-dimensional Ising model presented here assume the following:

1. A square lattice.

2. Only nearest-neighbor interactions are considered with the coupling constant set to $J = 1$.

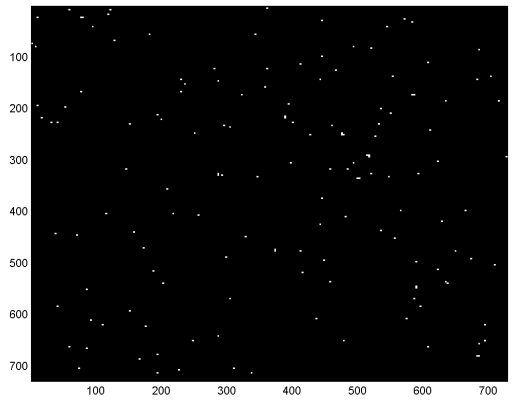
3. Unless otherwise stated, the external magnetic field, $H = 0$.

4. Toroidal boundary conditions in order to eliminate surface effects.

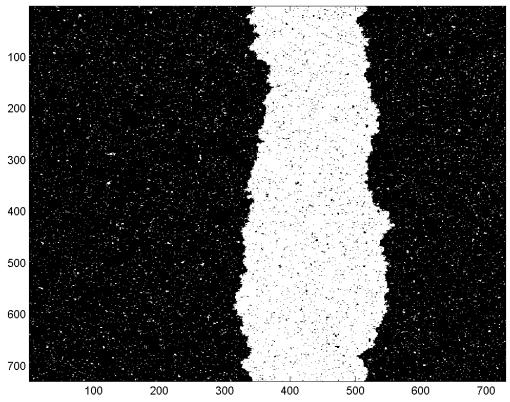
From Onsager's solution to the two-dimensional Ising model in the absence of an external magnetic field, we know that the model exhibits a phase transition at the critical temperature of $T_c = 2.269J/k$. This is one of the three fixed points for the model. The other fixed points occur at $T = 0$ and $T = \infty$. Below the critical temperature, the spins are aligned in the same direction across the lattice, above the critical temperature, the spins are aligned at random and are not correlated across the lattice. An alternative to the alignment of spins in parallel at $T < T_C$ is a metastable state, in which the spins form two or three large domains across the lattice in which the spins are aligned in parallel within each domain but the spins across domains are not parallel. Figure 3.1 shows snapshots of each of the four scenarios described, two for $T < T_c$ (one fully parallel, one metastable), one for $T \approx T_c$, and one for $T > T_c$. All of the snapshots were taken on simulations run on a lattice of 729×729 spins after $10,000$ Monte Carlo sweeps, with each sweep representing 729^2 iterations of the Metropolis algorithm.

One thing to note about the Ising model at the phase transition is the property of scale invariance. As can be seen in Fig. 3.1(c), the spins are arranged in domains that are of various length scales. In fact, if we zoom into the snapshot, we will find that the distribution of the relative length scales of the domains will be unchanged. Here, the Ising ferromagnet is in a critical state.

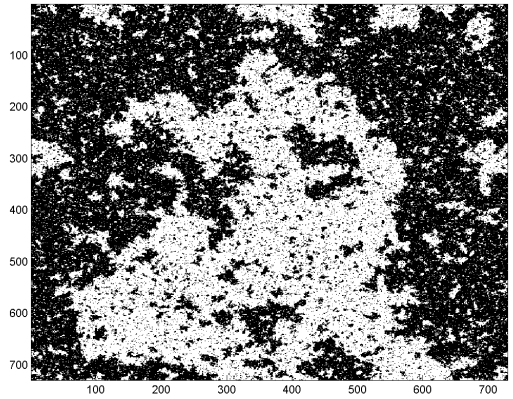
The first macro-property that we will measure in the simulations will be the magnetization per spin $m \equiv M/N$. As predicted by the Onsager solution, the plot of $|m|$ as a function of temperature should look like the plot on Fig. 2.7. The plot shown on Fig. 3.2 shows the results from the simulation run with a lattice of 100×100 spins, with data taken for $20,000$ sweeps, each of 100^2 iterations, for each value of the temperature. The temperature is varied from $kT/J = 1.5$ to 3.5 in increments of



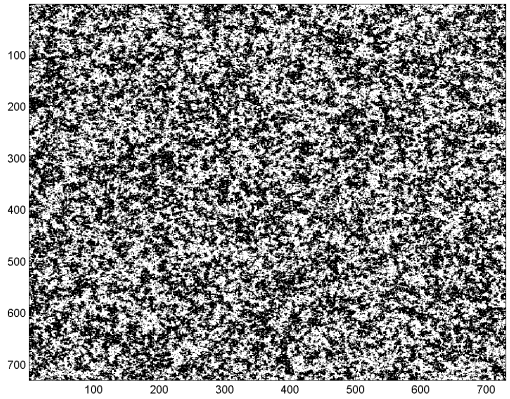
(a) $T < T_c$, almost all the spins are aligned in parallel.



(b) $T < T_c$, this snapshot shows a metastable state.



(c) $T \approx T_c$, undergoing a phase transition.



(d) $T > T_c$, the alignment of the spins is random.

Figure 3.1 Snapshots of each of the simulations of the 2D Ising model with 729×729 spins, two for $T < T_c$ (one fully parallel, one metastable), one for $T \approx T_c$, and one for $T > T_c$.

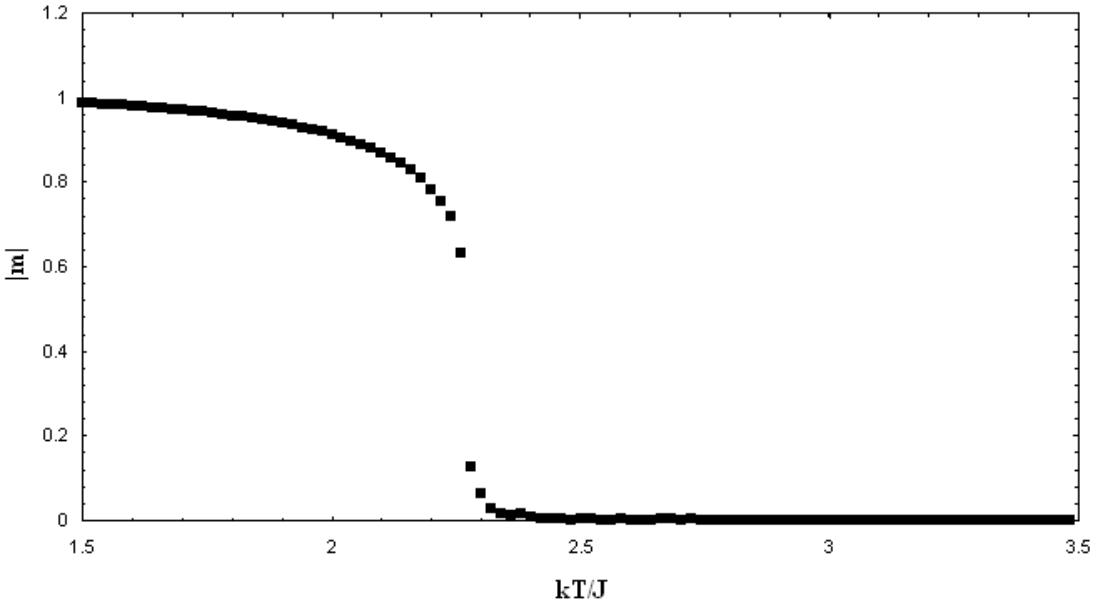


Figure 3.2 Plot of $|m|$ with respect to dimensionless temperature for a simulation of the two-dimensional Ising model. Note the drop off in $|m|$ at critical temperature.

0.02, which gives 100 data points. These specifications are the same that were used to generate Figs. 3.3, 3.4, and 3.5, which show energy, specific heat and magnetic susceptibility respectively.

As can be seen, the simulation results match the analytical results, with a sharp transition at the critical temperature of $T_c \approx 2.27$. Below the critical temperature, the magnetization per spin is approximately 1, and above critical temperature, it is approximately 0.

The second macro-property that we will determine by simulation is energy. This is given by the formula

$$E = -J \sum_{\langle ij \rangle} s_i s_j. \tag{3.36}$$

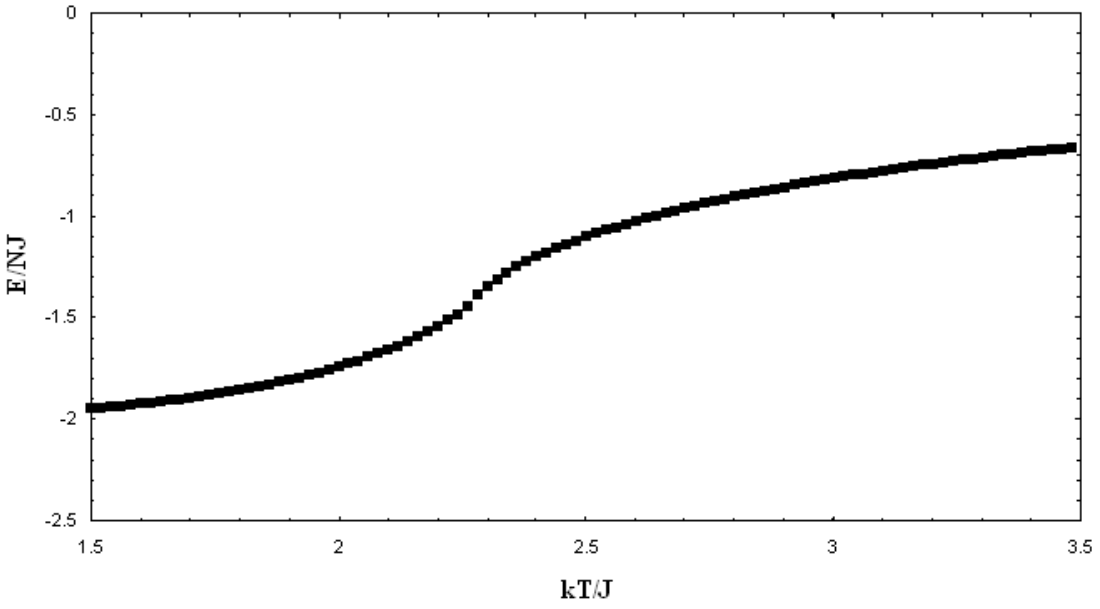


Figure 3.3 Plot of E with respect to dimensionless temperature for a simulation of the two-dimensional Ising model.

The Onsager solution predicts that the plot of E as a function of temperature looks like the plot of Fig. 2.5. The plot shown in Fig. 3.3 shows the results from the simulation run. We can see the function for simulation values of E has a point of inflection at $T = T_c$, just as in the Onsager solution.

Next is the specific heat, or heat capacity per spin, C/N, which we derived in the previous chapter in Eq. 2.39 as $C = \frac{1}{kT^2}\left[\overline{E^2} - \overline{E}^2\right]$. The plot of the result from the Onsager solution is shown in Fig 2.6. The plot shown on Fig. 3.4 shows the results from the simulation run. The function peaks at the critical temperature.

Finally, we have the magnetic susceptibility, χ which we derived in Eq. 2.52 as $\chi = \frac{1}{kT}\left[\overline{M^2} - \overline{M}^2\right]$. The plot shown in Fig. 3.4 shows the results from the simulation run.

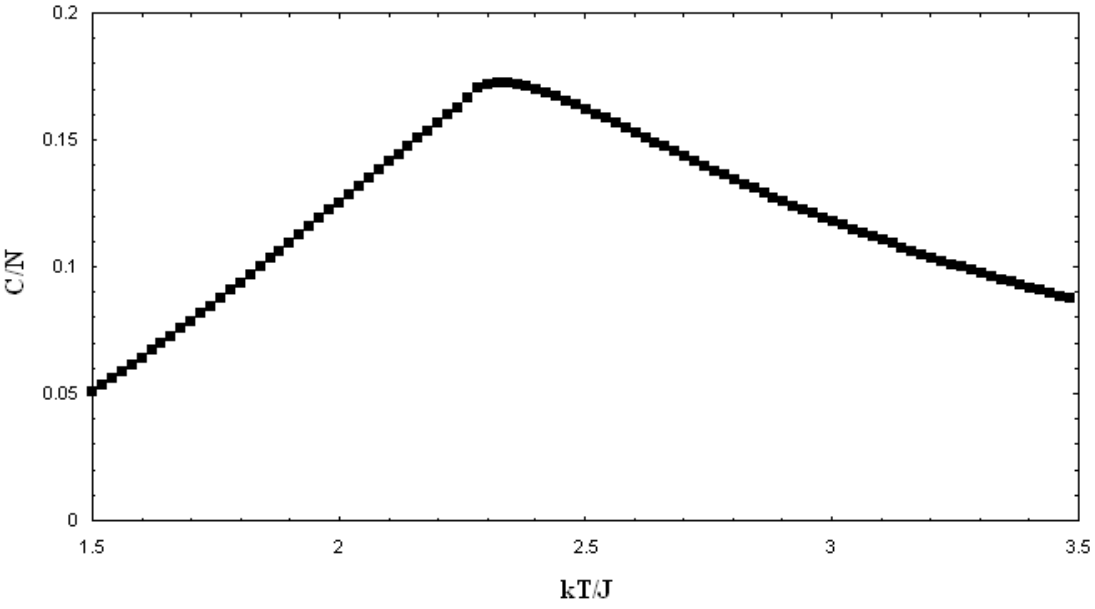


Figure 3.4 Plot of C/N with respect to dimensionless temperature for a simulation of the two-dimensional Ising model.

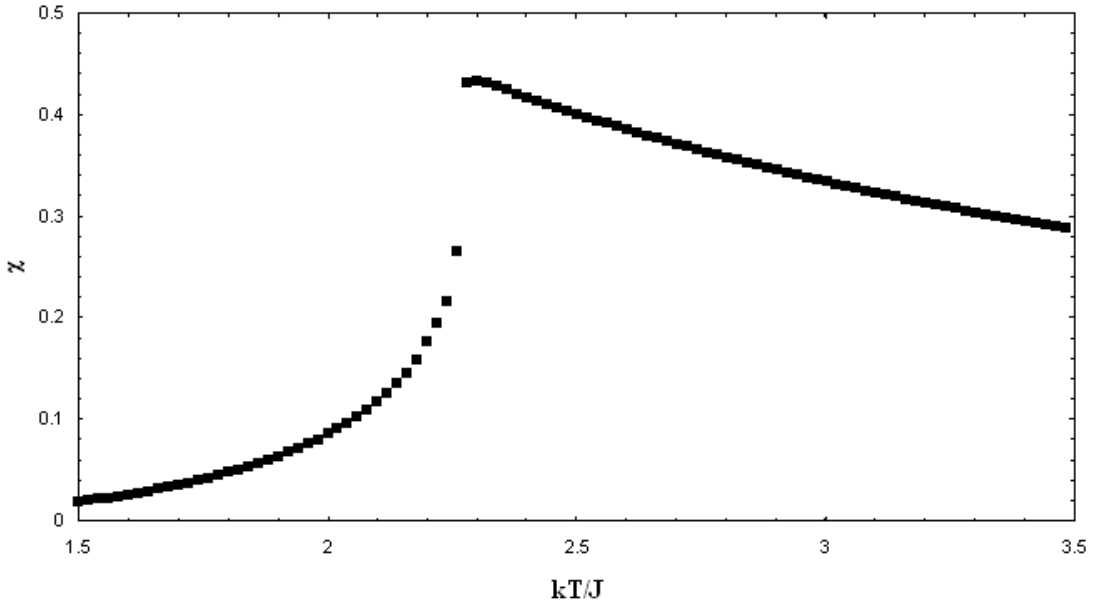


Figure 3.5 Plot of χ with respect to dimensionless temperature for a simulation of the two-dimensional Ising model.

3.3 Simulation results for the three-dimensional Ising model

For the simulations of the three-dimensional Ising model, we make the same set of assumptions as in the two-dimensional simulations given at the start of Section 3.2, except the first one. Instead of a square lattice, we run the simulations on the analog of the square lattice for three-dimensions, which is the simple cubic lattice, in which each spin has 6 nearest-neighbors. The leftmost lattice in Fig. 2.8 in Section 2.5 shows the structure of the simple cubic lattice. The three-dimensional Ising model does not have an analytic solution, therefore we primarily rely on simulations and approximation methods to study it. In this section, just as we did for the two-dimensional model, we will measure how three macro-properties, magnetization per spin, $|m|$, energy, E, and susceptibility, χ.

The simulations results which follow were run with a lattice of $50 \times 50 \times 50$ spins, with data taken up to $20,000$ sweeps for each value of the temperature. The temperature is varied from $kT/J = 2.5$ to 5.5 with increments of 0.02, which gives 150 data points. These specifications are the same those used to generate Figs. 3.6, 3.7, and 3.4, measuring magnetization per spin, energy, and magnetic susceptibility, respectively.

In Fig. 3.6, we see that the magnetization per spin drops off from 1 towards 0 at the critical temperature, which, from the simulation we can determine to be $T_c \approx 4.5J/k$. The plots seen in Figs. 3.7 and 3.8 corroborate this measurement of the critical temperature. The plot of E as a function of T shows a point of inflection at the critical temperature. The plot of χ as a function of T peaks at the critical temperature.

Another thing to note in this set of results is the presence of some outlier points

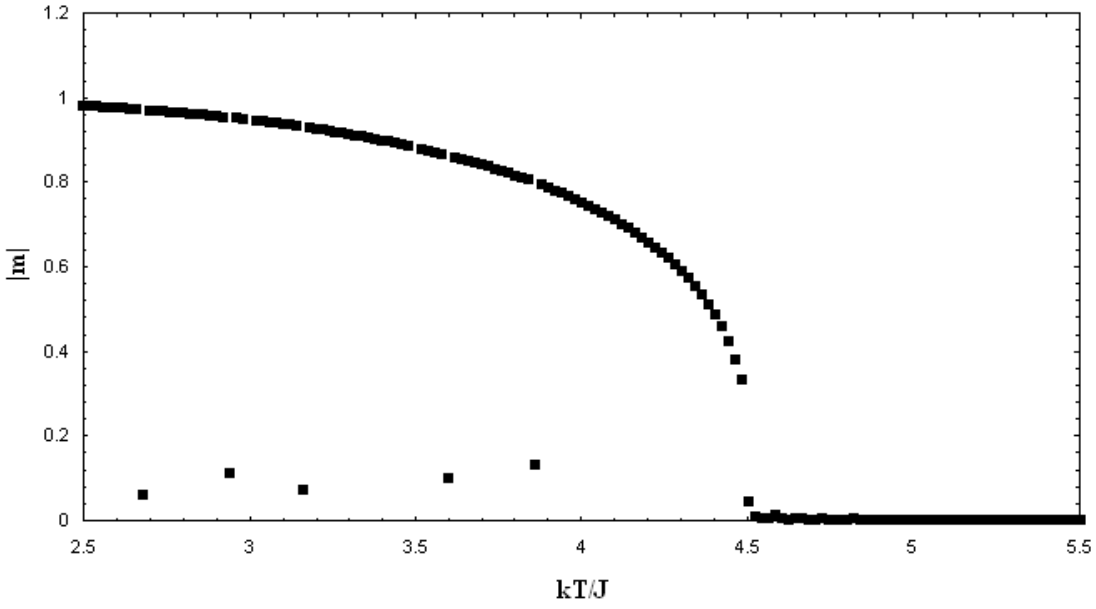


Figure 3.6 Plot of $|m|$ with respect to dimensionless temperature for a simulation of the three-dimensional Ising model. Note the drop-off in $|m|$ at critical temperature.

at $T < T_c$ which do not fit into the curves in all three of the figures. These are caused by the system settling into a metastable state. We know of this because the magnetization per spin is near zero, which means approximately half the spins are pointing up while the rest are pointing down.

3.4 Monte Carlo renormalization

In this section, we will look at a renormalization method that uses results obtained from Monte Carlo simulations applying the Metropolis algorithm. Specifically, the method we will look at is called the block-spin transformation. Like the renormalization methods we looked at in Section 2.6, it also relies on the fact that the Ising

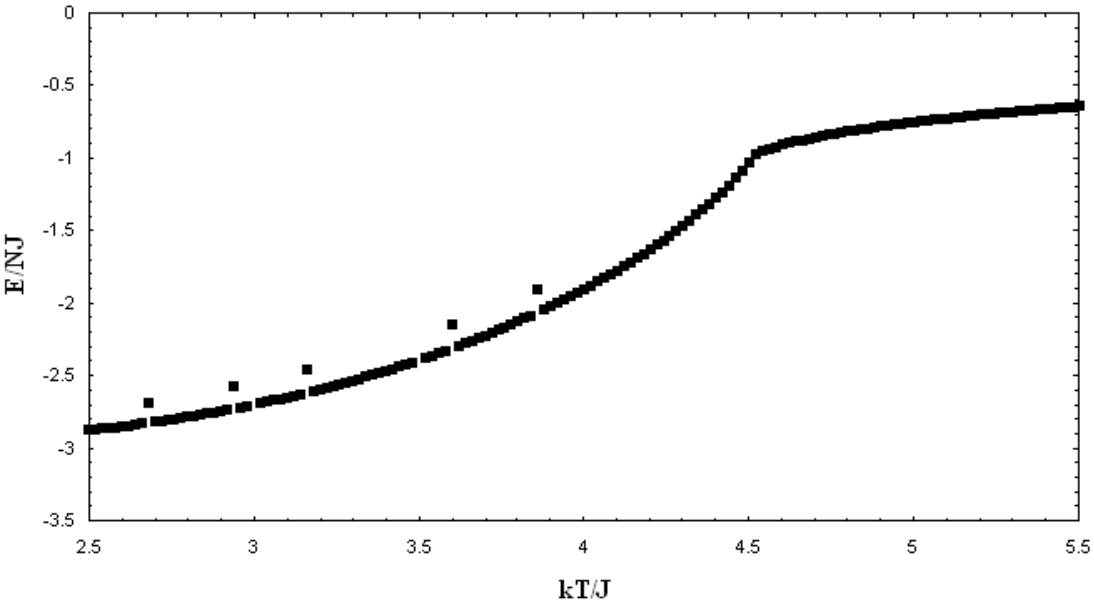


Figure 3.7 Plot of E with respect to dimensionless temperature for a simulation of the three-dimensional Ising model.

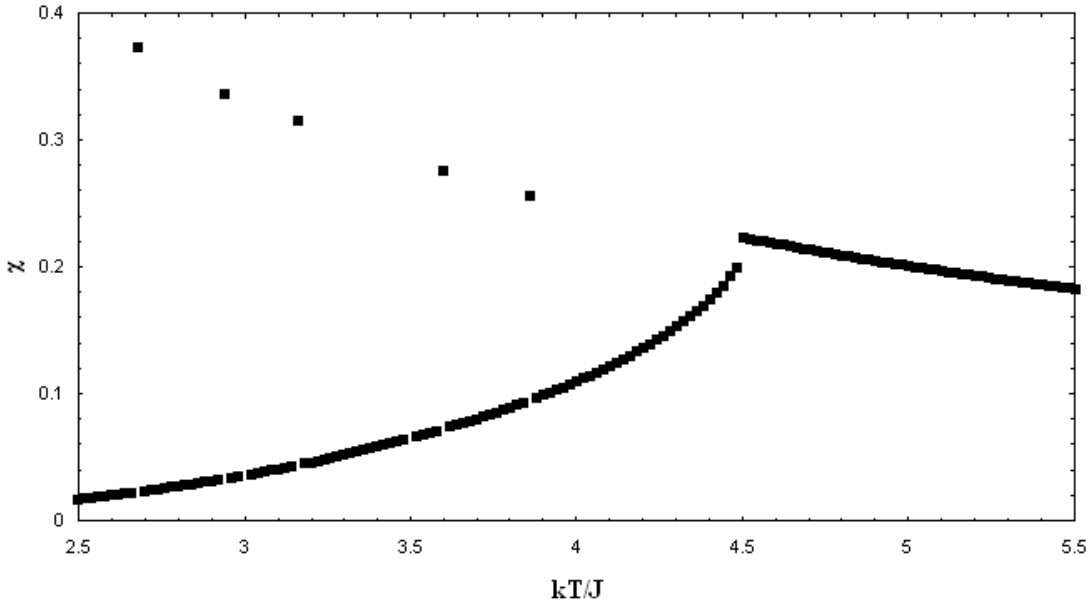


Figure 3.8 Plot of χ with respect to dimensionless temperature for a simulation of the three-dimensional Ising model.

model retains certain properties across different length scales. Also, like other renormalization methods, it allows us to determine the fixed points in the Ising model. We will specifically look at the two-dimensional version of the Ising model.

The procedure involves iteratively breaking the lattice into small groups of identical numbers of spins. For our purposes, we break a 729×729 lattice into groups of 3×3. After breaking the initial lattice into groups of spins, the spins in each of the groups are flipped so that they are all parallel. This can be done using a simple majority rule, in our case, if 5 or more spins are in a particular direction, the rest of the spins are flipped to the same direction. For our choice of groups of $3 \times 3 = 9$ spins each, the majority rule gives us a unique answer because the number of spins is odd. If we were to choose groups with an even number of spins, a procedure has to be followed to resolve scenarios in which half of the spins are in one direction and half are in another. Simply picking a direction at random to which all the spins in the group are to be aligned is one way of resolving such conflicts. After flipping the spins, the groups are combined to form a new lattice, but now with each spin in the new lattice represented by the spin of a particular group. Figure 3.9 illustrates the process. Now, we can apply this procedure repeatedly until we are left with a lattice of $N = 1$.

The interesting results, however, appear after a few iterations of the procedure. Since the two-dimensional Ising model has three fixed points, at temperature $T = 0$, T_c and ∞, respectively, we expect our renormalization procedure to distinguish between those three fixed points. When applying the block-spin transform to a Monte Carlo simulation result of the two-dimensional Ising lattice that is at $T < T_c$, a few iterations of the procedure will yield a lattice with spins aligned in the same direction. Figure 3.10 shows such a procedure with four iterations starting from a 729×729 lattice.

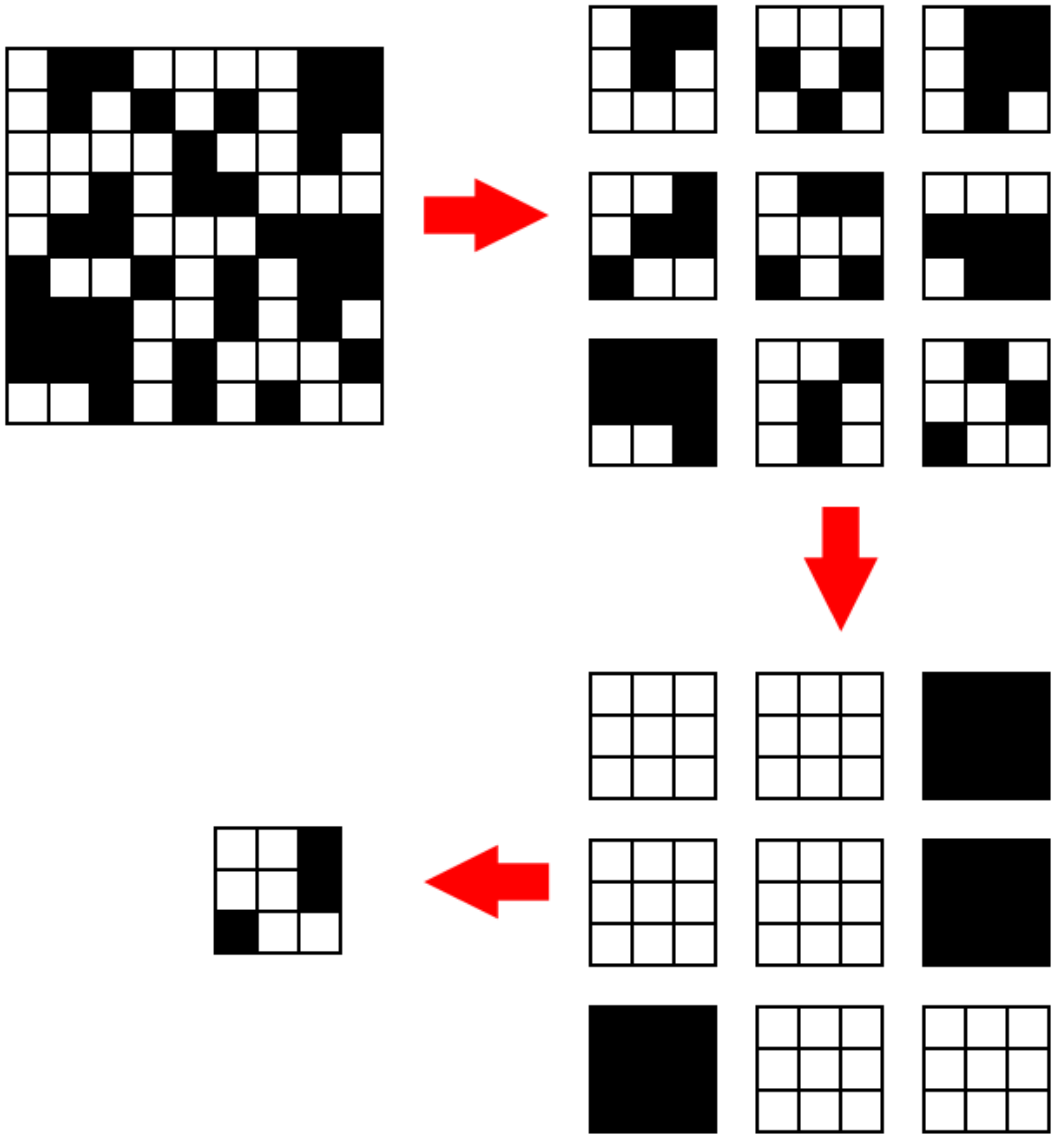
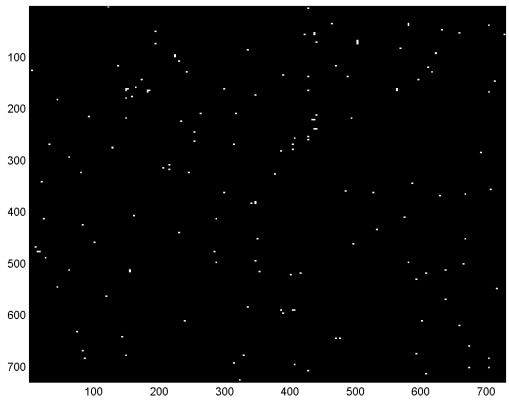
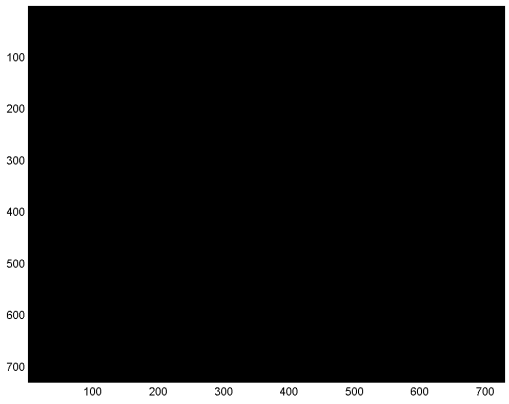


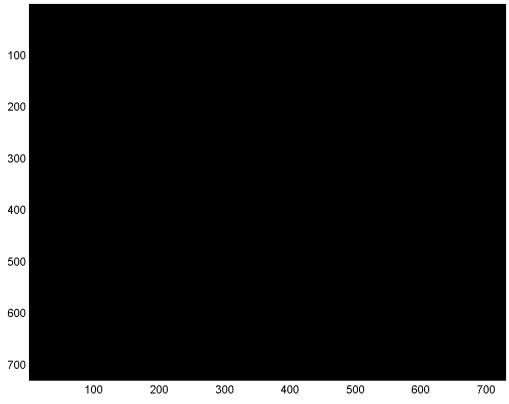
Figure 3.9 Illustration of the block-spin transform renormalization procedure.



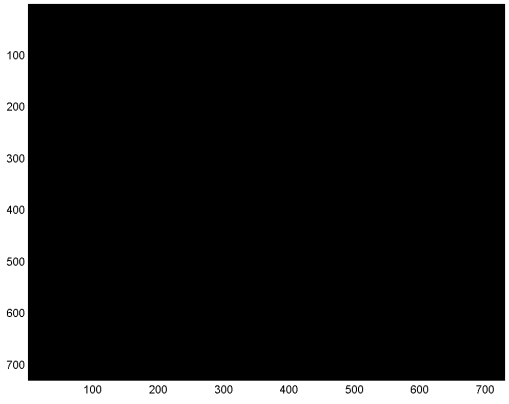
(a) 729×729.



(b) 729×729.



(c) 81×81.

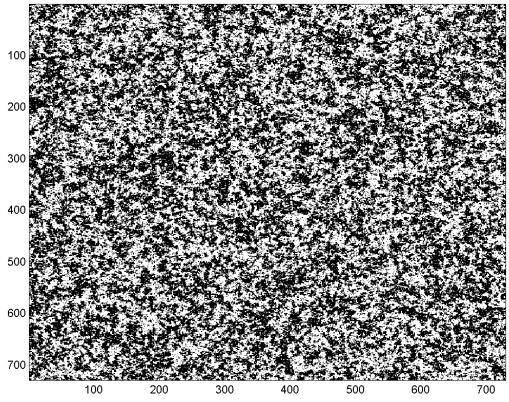


(d) 27×27.

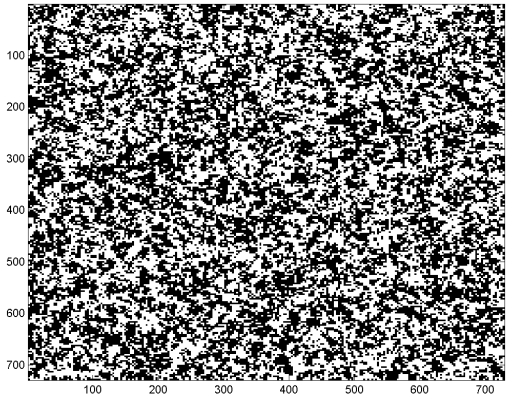
Figure 3.10 Snapshots of four iterations of the block spin transformation procedure for the two-dimensional Ising model with $T < T_c$.

When applying the block spin transform to a Monte Carlo simulation result of two-dimensional Ising lattice that is at $T > T_c$, a few iterations of the procedure will yield a lattice with spins aligned in random directions. Figure 3.11 shows such a procedure with four iterations starting from a 729×729 lattice.

As for the case when $T \approx T_c$, the block-spin transformation will tend to keep the structure of the original lattice over several iterations of the procedure. Figure 3.12 shows such a procedure with four iterations starting from a 729×729 lattice. This effect is due to the fact that at the critical temperature, the two-dimensional Ising model displays scale invariance, as discussed in Section 3.2.

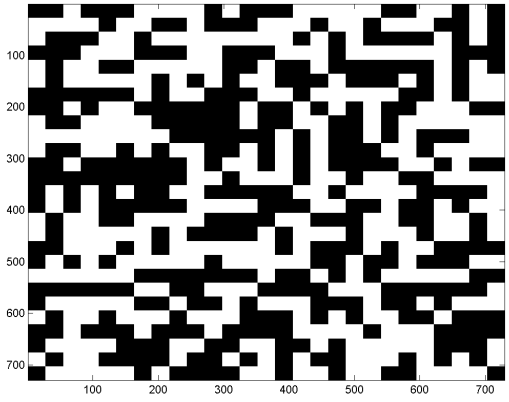


(a) 729×729.



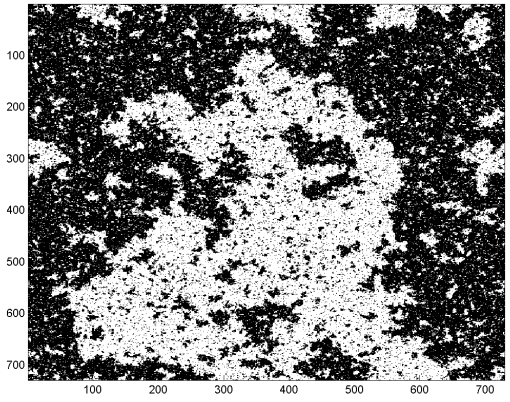
(b) 243×243.

(c) 81×81.

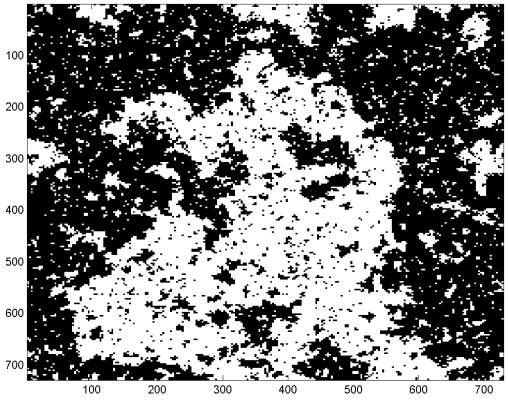


(d) 27×27.

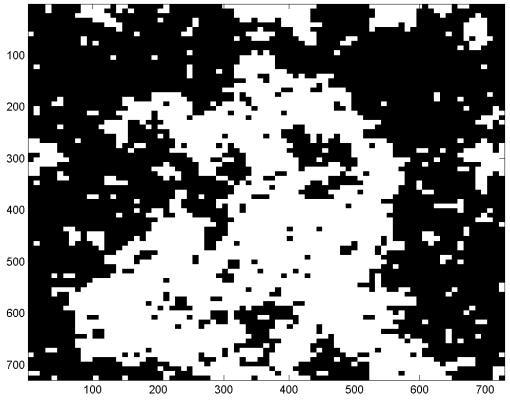
Figure 3.11 Snapshots of four iterations of the block-spin transformation procedure for the two-dimensional Ising model with $T > T_c$.



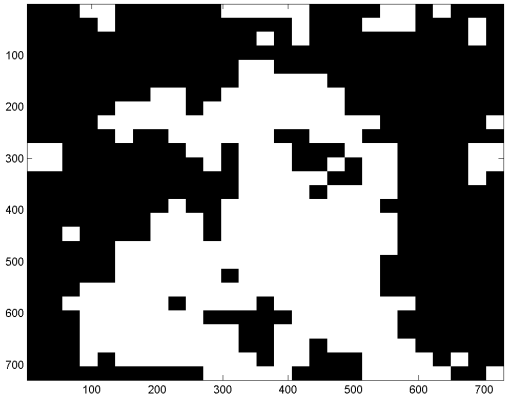
(a) 729×729.



(b) 243×243.



(c) 81×81.



(d) 27×27.

Figure 3.12 Snapshots of four iterations of the block-spin transformation procedure for the two-dimensional Ising model with $T \approx T_c$.

Chapter 4

A Review of Social Science Applications

4.1 Some concepts from statistical mechanics

While the previous chapters focused on the Ising model as it applied to the physical world of ferromagnets, this chapter gets to the interdisciplinary applications of models that are influenced by or are directly analogous to the Ising model. In order to explore these social science applications, however, we have to look at some additional concepts from statistical mechanics that we have not covered in the previous chapters. We will look briefly at a modified version of the Ising model called the Potts model, and two additional Monte Carlo methods, the heat-bath and Kawasaki algorithms.

4.1.1 Potts model

In the Ising model, spins can assume either of two values, $+1$ or -1. This is generalized in the Potts model such that a spin s_i can assume any number of integer values [18, p. 120]. For a q-state Potts model, we have $s_i = 1, 2, ...q$. The Hamiltonian for the

Potts model can be written as

$$E(\nu) = -J \sum_{\langle ij \rangle} \delta_{s_i s_j}, \tag{4.1}$$

which looks similar to the Hamiltonian of the Ising model shown in Eq. 2.54, except that it is suming the function $\delta_{s_i s_j}$ instead of the product of the spins $s_i s_j$. $\delta_{s_i s_j}$ is the Kronecker delta, where $\delta_{s_i s_j} = 1$ if $s_i = s_j$ and $\delta_{s_i s_j} = 0$ if $s_i \neq s_j$. As in the Ising model, J is the coupling constant and $\langle ij \rangle$ indicates the nearest-neighbor pairs.

For the Potts model with just $q = 2$ states, the Hamiltonian can be written in a way such that makes it equivalent to the Ising model. We can rewrite Eq. 4.1 as

$$E(\nu) = -\frac{1}{2} J \sum_{\langle ij \rangle} 2 \left(\delta_{s_i s_j} - \frac{1}{2} \right) - \sum_{\langle ij \rangle} \frac{1}{2} J. \tag{4.2}$$

We can see that if $s_i = s_j$, the expression $2\left(\delta_{s_i s_j} - \frac{1}{2}\right)$ is equal to $+1$, and is equal to -1 if $s_i \neq s_j$, just as we have in the Ising model. This means that the Hamiltonian of the Ising model, $E_{\text{Ising}}(\nu)$, can be related to the Hamiltonian of the 2-state Potts model, $E_{\text{Potts}}(\nu)$, by the expression

$$E_{\text{Ising}}(\nu) = E_{\text{Potts}}(\nu) + \sum_{\langle ij \rangle} \frac{1}{2} J, \tag{4.3}$$

where the second term on the right hand side is a constant independent of the microstate ν. Note also that for the two Hamiltonians to be equal, the values of the coupling constants also have to satisfy the relation $J_{\text{Ising}} = 2J_{\text{Potts}}$.

The Potts model's allowance for the spins to assume any number of values comes in handy for social applications, because often it is difficult to simplify a social situation into a system where each of the constituent elements can only assume two possible values.

Like the Ising model, the Potts model can be solved by using Monte Carlo methods such as the Metropolis algorithm. However, due to the fact that the Potts model

allows for many states for each spin, the Monte Carlo simulations will take much longer to reach equilibrium as they randomly sift through the much larger state space. In addition to the Metropolis algorithm, there are other Monte Carlo methods that are used to solve Ising-type models in statistical mechanics, some of which are are superior to the Metropolis algorithm in that they arrive at the equilibrium state much more quickly. We will look at one of these algorithms, called the heat-bath algorithm, in the next section.

4.1.2 Heat-bath algorithm

As described in Section 3.1, the Metropolis algorithm works by selecting a spin at random from the lattice, and calculating the probability of flipping the direction according to

$$\mathcal{P}(s_i \rightarrow -s_i) = \begin{cases} e^{-\Delta E/kT} & \text{if flipping the spin results in } \Delta E > 0, \\ 1 & \text{if flipping the spin results in } \Delta E \leq 0. \end{cases} \tag{4.4}$$

If we use the Metropolis algorithm to solve the Potts model instead of the Ising model, instead of selecting a spin and flipping its sign, we randomly select another integer from $\{1, 2, ..q\}$, and calculate the probabilities based on the changes in energy brought about by this new spin value.

In the heat-bath algorithm, we follow the following procedure [18, p. 122].

1. Start with any random microstate, *i.e.*, a randomly assigned arrangement of spins on a lattice.

2. Pick a spin at random, and choose a new value, n, for this spin out of possible values $\{1, 2, ...q\}$, with a probability

$$\mathcal{P}(s_i = n) = \frac{e^{-E(s_i=n)/kT}}{\sum_{m=1}^{q} e^{-E(s_i=m)/kT}}. \tag{4.5}$$

3. Start again by picking another spin from the lattice at random.

For the Ising model, the heat-bath algorithm does not make a huge difference in terms of the time taken to reach equilibrium, but it does work faster for the Potts model as the number of states, q, increases. To see why this is the case, consider the Metropolis algorithm for the q-state Potts model. It starts by first picking a spin value out of $\{1, 2, ...q\}$ with equal probability, and then decides whether to change the spin to this newly selected value depending on whether $\Delta E > 0$ or otherwise, as explained in Eq. 4.4. In a Potts model with many possible states for each spin, *i.e.* $q \gg 2$, we can see from the Hamiltonian in Eq. 4.1 that the energy decreases only when we go from a situation in which $s_i \neq s_j$ to one in which $s_i = s_j$. Since this scenario is relatively rare, most of the time, the Metropolis algortihm will cause the Potts model to be stuck in the same point in phase space, especially when the temperature is near zero, causing $e^{-\Delta E/kT} \rightarrow 0$.

In the heat-bath algorithm, however, the new values for the spins are picked according to the probabilities given by Eq. 4.5. This ensures that spin values leading to lower energies are are more likely to be picked.

For the purposes of this thesis, we will not be concerned with Potts models for which have $q > 3$. This means that it does not matter much whether we use the heat-bath algorithm or the Metropolis algorithm in terms of time to reach equilibrium. However, it is important to be aware that there are other Monte Carlo algorithms that can be used to solve these models.

4.1.3 Kawasaki algorithm

The Ising model can be has many different applications even within the domain of physics. One of them is to model lattice gases. A lattice gas is a model of a gas in which particles are moving around in space. A lattice of spins representing a volume

in which an empty site has a spin value of -1 and a site occupied by a particle has a spin value of 1 makes this model equivalent to the ferromagnetic Ising model. The total number of particles has to stay the same in a lattice gas model, whereas the total number of spins that are either $+1$ or -1 in the Ising model is constantly changing. Therfore, in the Ising model, the parameter M, or magnetization, is not conserved. Lattice gas models belong to a class of models called conserved-order-parameter models; the name is self-explanatory. We will see later that conserved-order-parameter models also have important applications in social science.

One Monte Carlo algorithm that can be used to to solve a conserved-order-parameter model is called the Kawasaki algorithm [18, p. 138]. It is a slight modification to the Metropolis algorithm. Instead of picking a spin at random from the lattice and flipping it, we pick a pair of adjacent spins and interchange their values. After doing so, we simply calculate the energy difference caused by this interchange, and decide whether to keep this change based on probability calculations similar to those used in the Metropolis algorithm described in Eq. 4.4. If we call the state of the lattice prior to interchanging the pair of spins ν, and the resulting state after the interchange μ, the probability of changing from state ν to state μ, is

$$\mathcal{P}(\mu \leftarrow \nu) = \begin{cases} e^{-\Delta E/kT} & \text{if interchanging the spins results in } \Delta E > 0, \\ 1 & \text{if interchanging the spins results in } \Delta E \leq 0, \end{cases} \tag{4.6}$$

just like in the Metropolis algorithm.

We now go on to look at various models in social science that are similar to or are directly inspired by the Ising model.

4.2 Schelling's segregation model

In Section 1.2.1, we gave a brief overview of Schelling's segregation model. We now return to take a more thorough look. Thomas Schelling originally implemented a Monte Carlo algorithm by hand to solve his model, literally moving dimes and pennies around a checkerboard. To quote from his decription of the model from his book 'Micromotives and Macrobehavior' [21, p. 147],

> "Get a roll of pennies, a roll of dimes, a ruled sheet of paper divided into one-inch squares, preferably at least the size of a checkerboard and find some device for selecting squares at random. We place dimes and pennies on some of the squares, and suppose them to represent the members of two homogenous groups - men and women, blacks and whites, French-speaking and English-speaking, officers and enlisted men, students and faculty, surfers and swimmers, the well dressed and poorly dressed, or any other dichotomy that is exhaustive and recognizable. We can spread them at random or put them in contrived patterns. We can use equal numbers of dimes and pennies or let one be the minority. And we can stipulate various rules for individual decision."

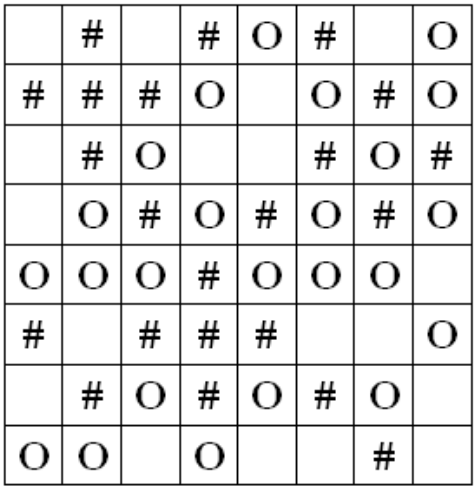
After emphasizing the generality of the model, he goes on to describe an algorithm to solve it. Each individual is at the center of a 3×3 neighborhood surrounded by 8 neighbors. She desires to live near neighbors that are of the same type. We now choose a certain fraction for the threshold of same-type neighbors that she wants in order to be satisfied, say 1/3. Next, we pick individuals at random, and if they do not have 1/3 or more same-type neighbors, we randomly place them in one of the free sites on the checkerboard where they will have 1/3 or more same-type neighbors. The process continues until everyone is in a satisfactory location.

We can see intuitively that this process correlates with real world social behavior in which people will move out of undesirable neighborhoods, and decide, with some degree of randomness, to move into a new neighborhood. The desirability of the neighborhood is in turn determined by the neighbors themselves. Even in highly

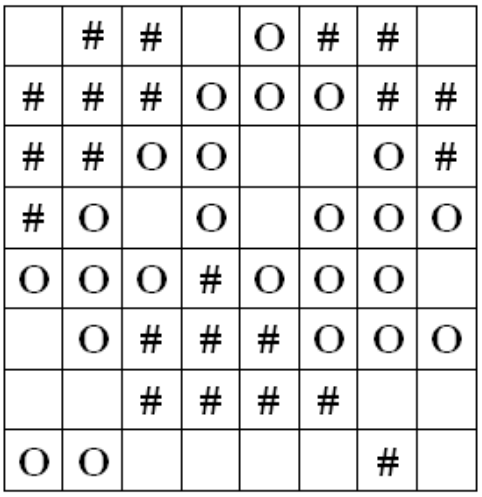
tolerant and cosmopolitan cities and towns, people will still prefer to live near other people with similar tastes and interests.

Schelling describes how the aforementioned algorithm leads to clustering of the two types of individuals into groups consisting of same-type neighbors. To show a result from his original manual simulation, Fig. 4.1(a) shows the starting configuration. It consists of two types of individuals, denoted by # and O. There are 22 #'s and 24 O's, arranged on a 8×8 grid, which leaves $(8 \times 8) - (24 + 22) = 18$ blank spots on the grid. The threshold of satisfaction is reached if 1/3 of their neighbors are of the same type. Note that unlike the Ising simulations carried out in Chapter 3, there are no toroidal boundary conditions. Figures 4.1(b) and 4.1(c) show two different outcomes of the simulation, if we had the configuration in Fig 4.1(a) as the starting point. As can be seen from the figures, the #'s and O's are noticeably segregrated. One thing to note is that the segregation does not lead to tightly clustered groups, but instead leaves a lot of blank spaces within each cluster. This results from Schelling's original algorithm, in which the individuals stop moving once they satisfy their minimal requirements to be happy and do not seek to maximize the number of same-type neighbors. In Fig 4.1(d), the rules were slightly modified so that instead of moving until 1/3 of the neighbors were of the same type, the individuals were moving until at least 3 of the nieghbors were of the same type, regardless of the fraction of their total number of neighbors. This algorithm produces a more dense clustering. All the figures in Fig. 4.1 were adapted from the book "Micromotives and Macrobehavior" [21, p. 147].

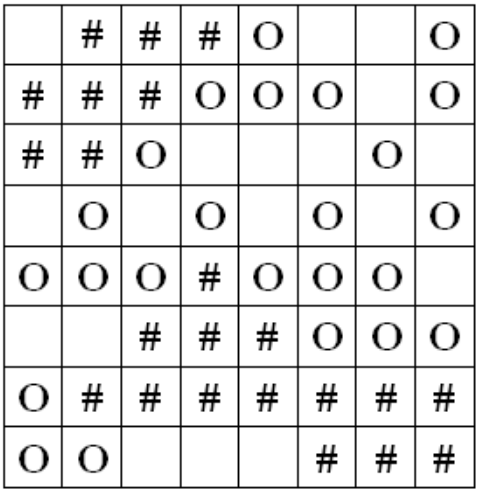
The importance of Schelling's model to the social sciences can be captured by the name of his book, "Micromotives and Macrobehavior". Just as in the Ising ferromagnet, macro-patterns such as spontaneous magnetization and segregation of neighborhoods emerge from the interactions of many different individuals responding to local conditions. These non-obvious emergent outcomes due to micro-level



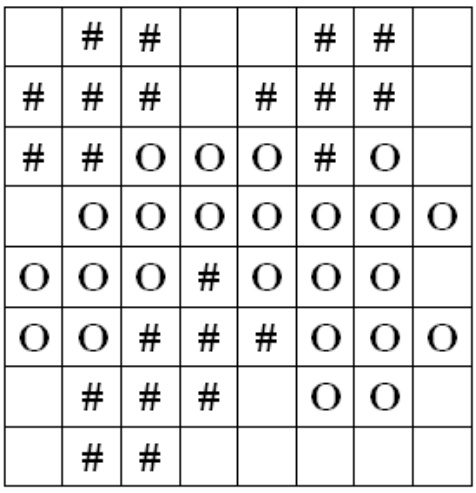
(a) Original configuration.



(b) Outcome 1, move until 1/3 of neighbors are same-type.



(c) Outcome 2, move until 1/3 of neighbors are same-type.



(d) Outcome 3, move until 3 neighbors are same-type.

Figure 4.1 Simulation results from the original Schelling model [21, pp. 149-151].

cooperative phenomena play an important role in the social sciences. For instance, modern economics is based around Adam Smith's idea of the "Invisible hand" of the free market, in which society-wide benefits can emerge from individuals' decisions to selfishly pursue their own interest.

4.2.1 Schelling's model reformulated as the Ising model

We now look at the similarities between the Ising and Schelling models, and look at how a version of the Ising model can be made into a more generalized case of Schelling's model. This kind of comparison has been described by Stauffer and Solomon [25].

The description of the Ising model starts off with a Hamiltonian. However, there is no real analog for energy in the social world. Instead, we have to invent a fictional parameter to serve as an analog, let's call it E. For a site i on a lattice, there is an associated E_i, which depends on the value of the spin at site i and the value of the neighboring spins. We also define two additional terms: the number of neighbors who are of a different type as i, $n_d(i)$, and the number of neighbors who are of the same type as i, $n_s(i)$. Thus, we have

$$E_i = \begin{cases} +1 & \text{if } n_d(i) > n_s(i), \\ -1 & \text{if } n_d(i) \leq n_s(i). \end{cases} \tag{4.7}$$

In other words, the individuals are satisfied at $E_i = -1$ and dissatisfied at $E_i = 1$. This conception of satisfaction deviates slightly from Schelling's model, in which satisfaction was determined by whether or not the number of same-type neighbors exceeded a certain variable threshold. In our case, we set the threshold at a constant equal to 1/2. Also, we have to note that unlike in the standard Ising model, which takes into account only the four nearest neighbors (North, South, East, and West),

Schelling's model also takes into account the next-nearest-neighbors (North-East, South-East, North-West, and South-West).

In Schelling's model, an individual would choose to move to a new spot only if it increased her satisfaction. There is no chance that she would randomly move to a new spot that offered the same or a lower level of satisfaction. Keep this in mind as we describe the next feature of the model.

In the Ising model, an important parameter is the temperature, T. If we look at the Metropolis algorithm for solving the Ising model, we see that the probability of choosing a certain value, either ± 1, for the spin s_i, $\mathcal{P}(s_i = \pm 1)$ is given by the expression in Eq. 4.4. Analogously, we can define a probability of an individual moving from site i to j as

$$\mathcal{P}(j \leftarrow i) = \begin{cases} e^{-\Delta E/T} & \text{if moving from } i \text{ to } j \text{ results in } \Delta E > 0, \\ 1 & \text{if moving from } i \text{ to } j \text{ results in } \Delta E \leq 0. \end{cases} \tag{4.8}$$

where

$$\Delta E = E_j - E_i, \tag{4.9}$$

or the change in E resulting from the move from site i to j. In the expression in the top line of Eq. 4.8, there is a variable T which is taken directly from the temperature in the Ising model. However, notice that if $T = 0$, $e^{-\Delta E/T} = 0$, which means that $\mathcal{P}(j \leftarrow i) = 0$ if $\Delta E > 0$ and $\mathcal{P}(j \leftarrow i) = 1$ if $\Delta E \leq 0$, which means that nobody moves to a place which decreases satisfaction, as described earlier. In this case, *i.e.* when $T = 0$, the Ising model is identical to the Schelling model. For $T > 0$, we can think of it as a generalization of Schelling's model in which due to extraneous circumstances, some people have to move to undesirable locations, despite of their dissatisfaction in doing so. We can think of T in this model as tolerance. Since $e^{-\Delta E/T}$ increases monotonically with T, we can say that a higher tolerance increases

the probability of moving into a undesirable location.

This Ising equivalent form of Schelling's model has to take into account two additional features before being totally compatible. Firstly, Schelling's model consists of three different possibilites at each lattice point, it can either be occupied by an individual from one of the two groups, or it can be empty. This means that the Ising model, which consists of only two possible states, cannot be used; instead we must use a 3-state Potts model, as described in Section 4.1.1. Also, note that Schelling's model is a conserved-order-parameter model, in which the number of individuals from each group and the number of empty sites remain constant. This means that we need to use an algorithm such as the Kawasaki algorithm described in Section 4.1.3 to solve this conserved-order-parameter Schelling model.

We now see that Schelling's original model is a special case of an Ising-type model. Specifically, it is a 3-state conserved-order-parameter Potts model on a square lattice with nearest-neighbor and next-nearest-neighbor interactions, at a temperature of $T = 0$. One can easily think of other social applications of the more generalized Ising-type model, if we relax any of those constraints. To list a few of these applications:

- One case that we have already discussed is to have a variable temperature or "tolerance".

- Stauffer and Solomon [25] discuss how we do not have to assume a conserved-order-parameter model if we allow for the individuals to migrate. If we think of the lattice as a town or city, we can imagine that any time an individual from group A moves out of a location, she actually leaves town and is replaced by someone moving in from out of town who is from group B. This justification is somewhat plausible, since obviously there is migration to and from cities in the real world.

- Instead of just two groups of individuals, we can extend it to any integer number of groups with a q-state Potts model. This is definitely applicable in the real world as there is a variety of 'social groups'.

- Instead of just considering the neighborhood of 8 nearest and next-nearest neighbors, we can extend it to any kind of neighborhood we want. We can also change the square lattice to something more socially realistic like a complex network, which we will discuss in Section 4.5.

4.3 Sznajd's opinion dynamics model

We have briefly touched upon Sznajd's opinion dynamics model in Section 1.2.2. Now, we will look at it in more detail and present some of the results of the model in one and two dimensions. Like the Ising model, the Sznajd model is also a discrete spin model in which the spins located on a lattice can take either of two values, $+1$ or -1. The two values can represent two different types of opinions, for instance, a 'yes' or a 'no' vote in an election. The model is driven by rules that determine the iterative behavior of the spins depending on the values of their neighboring spins.

4.3.1 The Sznajd model in one dimension

In Fig. 1.3 in Section 1.2.2, we introduced one of the two rules that drive the behavior of the one-dimensional Sznajd model. In a one-dimensional chain of spins $(s_i; i = 1, 2, 3, ..., N)$, the following rules are used to iteratively update the direction of the spins:

1. If $s_i s_{i+1} = 1$, then s_{-1} and s_{i+2} take the direction of the pair $i, i + 1$.

2. If $s_i s_{i+1} = -1$, then s_{-1} takes the direction of s_{i+1}, while and s_{i+2} takes that of s_i.

The first rule seems intuitive in a social application, a group of people having the same opinion causes this opinion to spread to the other people close to the group. The second rule seems less intuitively appealing, since it requires that a disagreement among individuals within a group causes this disagreement to propagate to the neighbors. The model is solved computationally by follows a simple Monte-Carlo algorithm, which picks a pair of neighboring spins in a chain, $s_i s_{i+1}$, and following one of the rules described above depending on whether $s_i s_{i+1} = 1$ or $s_i s_{i+1} = -1$. This process is then repeated with another pair of spins picked at random.

In the one-dimensional model, beginning from a random configuration of the spins, Sznajd observed that the two rules described above cause the model to end up in one of three different steady states:

1. All the spins are in aligned in the $+1$ direction.

2. All the spins are in aligned in the -1 direction.

3. The spins are divided equally between $+1$ and -1 directions, provided the total number of spins, N, is even. In this case, the spins are arranged in an antiferromagnetic pattern: $-1, 1, -1, 1, -1, ...$

If instead of having the initial conditions be a random configuration of spins, we choose a certain fraction, c_B, to point in specified direction, while the rest of the spins are made to point in the other direction. We can either make these $c_B \times N$ spins that are pointing in the same direction dispersed randomly throughout the chain of spins, or have them in a consolidated in a cluster. Figure 4.2 shows how the choice of c_B affects the probability of reaching each of the three steady states. These results

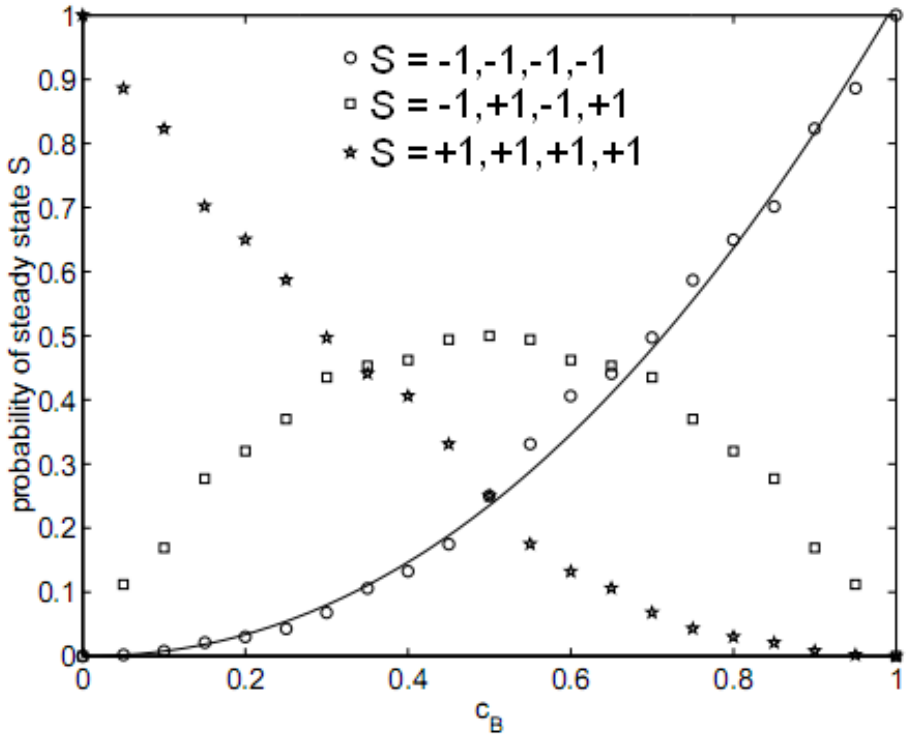


Figure 4.2 Dependence of the one-dimensional Sznajd model's results on the initial conditions. The circles represent the steady state with all spins aligned in the -1 direction. The stars represent the steady state with all spins aligned in the $+1$ direction. The squares represent the steady state with half the spins in the $+1$ and the other half in the -1 direction, arranged in an alternating (or antiferromagnetic) pattern. The fitted curve has the function $\mathcal{P}(S) = 1.02c_B^{2.12}$.

shown in Fig. 4.2 are taken from Sznajd's original paper [28], in which she ran multiple Monte Carlo simulations with different values of c_B. It is also observed in the simulations that it does not make a difference whether the spins pointing in the same direction are clustered or spread randomly throughout the chain.

The curve fitted through the points in Fig. 4.2 has the function $\mathcal{P}(S) = 1.02c_B^{2.12}$. Thus, we can see that in order to have the probability of achieving the steady state with all spins in the -1 direction $\mathcal{P}(S) > 0.5$, we have to have $c_B > 0.7$.

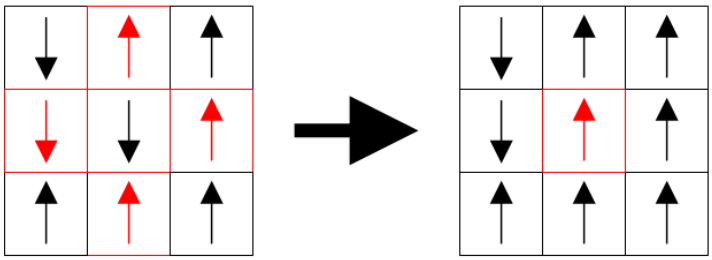
4.3.2 The Sznajd model in two dimensions

To extend the Sznajd model to two dimensions, we modify the rules slightly. In a paper by Stauffer et al. [26], the authors present a number of different rules that follow the logic of the original one-dimensional model but apply to a two-dimensional square grid. One of the most frequently used rules for the two-dimensional model is shown in Fig. 4.3(b). If there is a 2×2 block of spins that are aligned in the same direction, the spins that form the nearest neighbors of these 2×2 block will also align in the same direction. The intuition behind the rule and the Sznajd model in general is that when a group of people have a particular uniform opinion, their neighbors tend to follow this opinion as well.

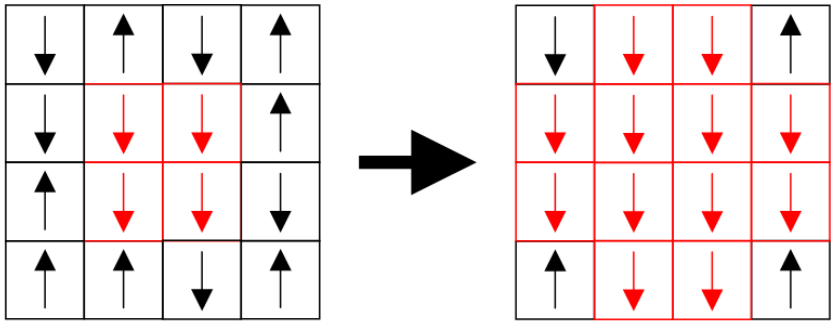
The main difference between the Sznajd model on one hand and Ising and Schelling's models on the other is that in the latter two, the main unit of concern is each cell in the lattice and its surrounding cells, whereas in the Sznajd model, the main unit of concern are pairs of cells and their neighbors, instead of individual cells and their neighbors. This change from individuals to pairs give the Sznajd model a different dynamics for the flow of information from that of the Ising model. We explain more about this below.

In the Ising model, a particular spin's direction is determined by the direction of the neighboring spins. This means that the information that brings about the change in the population's opinions flows inwards, towards a particular spin from its neighbors. Figure 4.3(a) shows this dynamic in the two-dimensional Ising model. In the Sznajd model, it is the outcome neighbors' spins which are influenced by direction of the spins of the original pair or 2×2 block. This means that the information which brings about changes in the population's opinions flows outwards. Figure 4.3(b) shows this dynamic in the two-dimensional Sznajd model with the rule described earlier.

Just as in the one-dimensional version of the model, the outcomes in the two-



(a) Change occurring in the 2D Ising model.



(b) Change occurring in the 2D Sznajd model.

Figure 4.3 Comparison of information flows in the Ising and Sznajd models. The Ising model shows information flowing inwards from the neighbors while the Sznajd model shows information flowing outwards to the neighbors.

dimensional Sznajd model are also influenced by the initial conditions. In the paper by Stauffer et al. [26] which introduced the two-dimensional Sznajd model [26], the authors ran multiple simulations of the model using various values for c_B, which is the initial fraction of spins that are pointing in a certain direction, just as we described in the one-dimensional model. Figure 4.4 shows the results from the simulations. It shows the total number of runs out of 1000 sample runs for which the lattice settled into a steady state with all of the spins pointing in the direction specified by c_B. Three different lattice sizes were used, one with 53×53 spins, one with 101×101 spins, and finally one with 301×301 spins. As can be seen in Fig. 4.4, all three lattice sizes showed a phase transition occuring at $c_B = 0.5$. For $c_B < 0.5$, none of the runs end up with the spins aligning in the direction specified by c_B, whereas for $c_B > 0.5$, all of the runs end up with the spins aligning in the direction used to c_B. Also, a smaller lattice size makes for a sharper phase transition.

The Sznajd model has been applied to a variety of phenomena including politics, marketing, and finance [27]. In most of these applications, the solution to the Sznajd model is arrived at through Monte-Carlo simulations. One thing to note with the applications of the Sznajd model is that many of them were not carried out on a two-dimensional grid but rather on complex networks. These networks consists of nodes (spins) which are connected through edges to other nodes (spins). Instead of a regular relationship between spins and their neighbors as would be found on a grid, the networks allow for complex relations between neighbors that resemble real world social networks. For the Sznajd model, which is primarily designed to model social phenomena, the complex network versions of the model make it a better approximation of reality. In Section 4.5, we discuss some basic ideas behind complex networks.

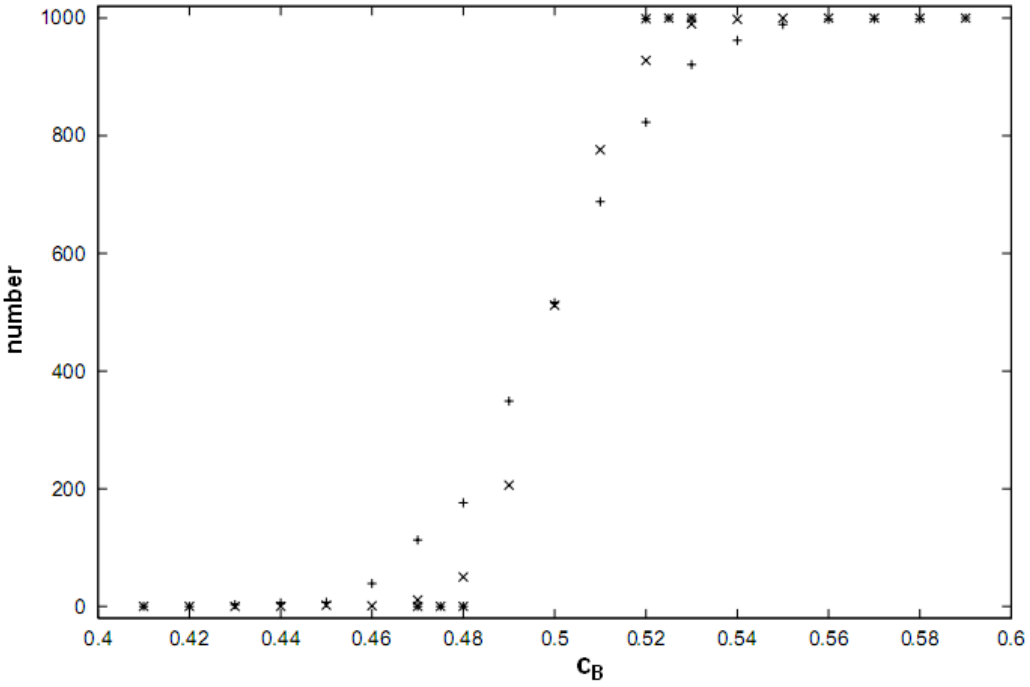


Figure 4.4 Dependence of the two-dimensional Sznajd model's results on the initial conditions. The points marked '$\times$', '$+$', and '$*$' represent the represent the results from the runs with a 53×53 lattice, a 101×101 lattice, and a 301×301 lattice respectively.

4.4 Durlauf and Blume's social interactions model

We gave some background to Durlauf and Blume's social interactions model in Section 1.2.3 and will now proceed to explore it in more detail. The derivation below is adapted from a paper by Brock and Durlauf [9], which was cited in turn by Durlauf and Blume [7]. We begin the description of the model by first assuming a population of I agents, each with an individual utility function, V, defined as

$$V(\omega_i) = h\omega_i + J\omega_i\overline{m}_i^e + \varepsilon(\omega_i), \tag{4.10}$$

where V is a function of $\omega_i \in \{-1, 1\}$, which represents a binary choice for individual i. The function is a sum of three parts. The first term, $h\omega_i$, is the private utility associated with choice ω_i, which means that the utility derived is independent of other people's choices. The second term is the social utility, and is a product of a coefficient J, the choice ω_i and $\overline{m}_i^e$, which is defined as

$$\overline{m}_i^e = \frac{1}{I-1}\sum_{j\neq i} m_{i,j}^e, \tag{4.11}$$

where $m_{i,j}^e$ is the subjective expected value from the perspective of individual i of individual j's choice. This means that the second term in Eq. 4.10, the social utility, depends on the what individual i thinks the choices of the other individuals will be.

Finally, the last term in Eq. 4.10, $\varepsilon(\omega_i)$, is a random utility term, independently and identically distributed across the population of individuals. We assume that the difference between the random utility values $\varepsilon(-1)$ and $\varepsilon(1)$ have the probablity distribution

$$\mathcal{P}(\varepsilon(-1) - \varepsilon(1) \leq x) = \frac{1}{1 - e^{-\beta x}}. \tag{4.12}$$

Equation 4.12 introduces the term β, which determines the extent to which the random component of the utility makes a difference to the choice ω_i. As $\beta \to 0$,

$\mathcal{P}(\varepsilon(-1) - \varepsilon(1) \leq x) \to 1/2$, which means the difference between the two choices for ω_i is equally likely to be less or greater than any given x. As $\beta \to \infty$, $\mathcal{P}(\varepsilon(-1) - \varepsilon(1) \leq x) \to 1$, which eliminates the randomness in the utility function.

Now, the individuals choose a value for ω_i such that they maximize $V(\omega_i)$. Since $V(\omega_i)$ is deterministic, except for the term $\varepsilon(\omega_i)$, we can derive a probability distribution for ω_i that is based on the assumed distribution for $\varepsilon(\omega_i)$ in Eq. 4.12. We will not go over this step in the derivation, but will merely quote the answer from reference [9] for the probability distrbution, $\mathcal{P}(\omega_i)$, over the choice ω_i, which states

$$\mathcal{P}(\omega_i) = \frac{e^{\beta(h\omega_i + J\omega_i \overline{m}_i^e)}}{\sum_{\omega_i} e^{\beta(h\omega_i + J\omega_i \overline{m}_i^e)}}. \tag{4.13}$$

From this probability function in Eq. 4.13, we can derive the expected value, $E(\omega_i)$, for the choice ω_i. To do so, we simply multiply the possible ω_i values with their respective probabilities, to give

$$E(\omega_i) = 1 \cdot \frac{e^{\beta(h + J\overline{m}_i^e)}}{e^{\beta(h + J\overline{m}_i^e)} + e^{-\beta(h + J\overline{m}_i^e)}} - 1 \cdot \frac{e^{-\beta(h + J\overline{m}_i^e)}}{e^{\beta(h + J\overline{m}_i^e)} + e^{-\beta(h + J\overline{m}_i^e)}} \tag{4.14}$$

$$= \tanh\left(\beta(h + J\overline{m}_i^e)\right). \tag{4.15}$$

The solution shown in Eq. 4.15 bears a striking resemblance to the solution to the mean-field approximation of the Ising model derived in Eq. 2.5.

We go on to make the assumption, like in the mean-field approximation of the Ising model, that $E(\omega_i) = \overline{m}_i^e = m^*$, which means that the expected value of individual i's choice is equal to the mean value of the choices of all the other individuals. This assumption reduces Eq. 4.15 to

$$m^* = \tanh\left(\beta(h + Jm^*)\right), \tag{4.16}$$

which is very similar to the answer in the mean field approximation of the Ising model. Thus, the solution to 4.16 can be observed in the plots in Fig. 2.9 from Section 2.5.

We can therefore conclude that the model has three different solutions if $\beta J > 1$ and $h = 0$, while it has a unique solution if $\beta J < 1$ and $h = 0$.

Durlauf and Blume's work on social interaction models show that models from statistical mechanics need not be inconsistent with theoretical assumptions in the social sciences. For instance, we have shown above that the assumption of rational utility maximizing individuals can be applied to a discrete spin model borrowed from statistical mechanics.

4.5 Complex networks

In 1967, social psychologist Stanley Milgram conducted a famous experiment to test the "small world phenomenon," the idea that everybody in the world is separated from each other through a chain of only a few people. For instance persons A and F might live on different continents but A knows B, who knows C, who knows D, etc., until we link the chain up to person F. Milgram's experimental setup was to ask random people from various places in the United States to send a letter to a person in Boston. In doing so, they were required to mail the letter to someone that they knew personally, who in turn might be more likely to know the final recipient personally. The results of the experiment showed that the average length of the chains of letters was 5.2 links [30].

Sociologists have always been interested in the properties of social networks. Harrison White, a theoretical physicist turned sociologist, pioneered research in social networks. His student, Mark Granovetter, furthered White's research in a highly influential 1973 paper called "The Strength of Weak Ties," in which he studied the process of how people got jobs and found that acquaintances were much more likely to be helpful for job seekers than close friends and family. He explained that weak ties

that connect together local social clusters are "indispensable to individuals' opportunities and to their integration into communities" [13]. In their early research on social networks, these sociologists were anticipating the major breakthroughs in the study of networks that occurred in the 1990s. In the past decade, the study of networks has yielded insights into the complex structure of networks in the real world, not only in the social realm but in a variety of fields ranging from biology to the structure of the world wide web.

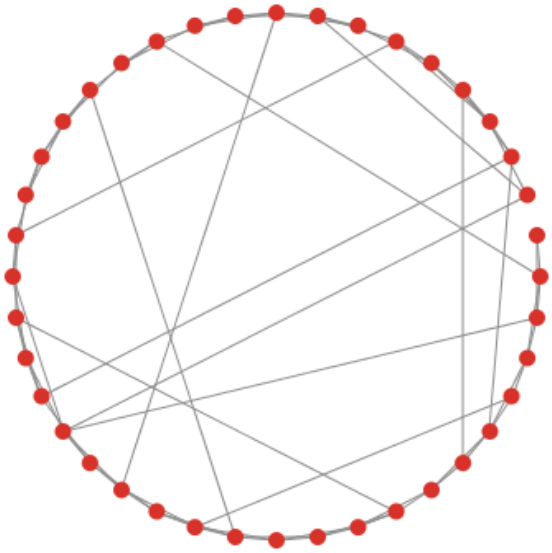
All the models described in this thesis can potentially be extended to a complex network. Changing from a two or three-dimensional lattice to a complex network does not change any of the inner workings of the models, but may require modifications to the rules. For instance, if the Sznajd model is applied to a complex network, we will have to come up with new behavioral rules. For example, we first select a chain of four spins that are linked together, and if the pair in the center is aligned in the same direction, the other two spins at the ends will also align.

4.5.1 Small world networks

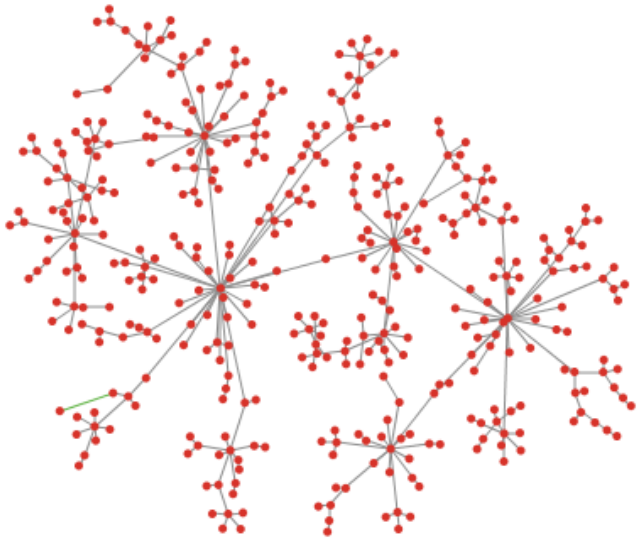
The first of these breakthroughs was in 1998, when Steven Strogatz and Duncan Watts published their empirical study of three completely different networks: the network of Hollywood actors, the electrical power grid of the western United States, and the neural network of the nematode worm *C. elegans* [31]. They discovered that all three of these networks display the same feature that Milgram found in his experiment: they were all "small worlds". Each of the nodes on the networks were connected to any other by only a few links. Figure 4.5(a) shows an example of a small world network with $n = 40$ nodes.

4.5.2 Scale-free networks

Another breakthrough in the study of complex networks came in 1999. In a paper entitled "Diameter of the World-Wide Web", Albert-László Barabási, Réka Albert and Hawoong Jeong published their findings on the network structure of the world wide web [2]. They discovered that the degree distribution of nodes (web pages) followed a power law. The degree distribution of a network is the probability distribution of nodes according to the number of connections they have to other nodes. For example, on the internet, a select number of very popular websites have incredibly high degrees while the overwhelming majority of other sites have degrees that are several orders of magnitude below that of the most popular sites. These kinds of networks are called scale free networks because of their distinctive degree distributions. Many real world networks have since been discovered to be scale free. Figure 4.5(b) shows an example of a scale free network with $n = 400$ nodes.



(a) An example of a small world network with $n = 40$ nodes.



(b) An example of a scale-free network with $n = 400$ nodes.

Figure 4.5 Examples of small world and scale-free networks.

Chapter 5

Conclusion

More than 80 years after Ernst Ising first published his model of the ferromagnet, the Ising model is still an incredibly fertile field of research. It has moved beyond the physics community to find applications in various other disciplines, including the social sciences. In this thesis, we explored the Ising model through various approaches. We first looked at the original one-dimensional model developed by Ising in his 1925 dissertation and then derived an analytical solution. We then looked at the various approximation methods used to solve the model in higher dimensions, namely mean field theory and renormalization group theory. We then moved on to the computational approach and used the Metropolis algorithm to solve the Ising model in two and three dimensions. The computational solutions agreed with the analytic results of the two-dimensional model derived by Onsager.

Next, we looked at the applications of the Ising and discrete spin models in the social sciences. We specifically explored three different models: Schelling's segregation, Sznajd's opinion dynamics, and Durlauf and Blume's social interaction model. These three models give us a rough overview of how discrete spin models have been used in the social sciences by researchers coming from different backgrounds. The applica-

tion of discrete spin models in the social sciences is still in its infancy and is currently a very active field of research. It would be more beneficial for both physicists and social scientists, however, to be more open-minded and knowledgable of each other's disciplines. To compare the Sznajd model, which was developed by physicists, and Durlauf and Blume's model, which was developed by economists, we can see that the two models have very different theoretical underpinnings even though they share similarities in the end results, such as the presence of a phase transition. On the other hand, it might be the case that the models taken from physics serve as a generalized form of models in the social sciences, as we saw in the case of the Ising and Schelling models.

The work presented in this thesis provides many directions for future research, since the Ising model itself is highly customizable and generalizable. We can look into different analytic and approximate techniques, focusing on different kinds of lattice structures, for instance. In terms of computational work, there are various other algorithms, such as the heat-bath algorithm, that can be implemented and compared with the Metropolis algorithm. A thorough study of the computational complexity of these algorithms would also be a fruitful direction for research. As for the interdisciplinary applications, there are seemingly limitless directions for new research. One particular area that unfortunately gets only a brief treatment in this thesis is the application of discrete spin models to complex networks. This topic can be treated both analytically and computationally. In addition, the application of models from statistical mechanics to the social sciences raises a number of different methodological questions. A concise treatment of these methodological issues is also an interesting direction for future research.

Bibliography

[1] Wikipedia article on "phase transition". http://en.wikipedia.org/wiki/File:Phase-diag2.svg, 2009.

[2] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. The diameter of the world wide web. *Nature*, 401:130, 1999.

[3] Elizabeth Anderson and Jeffery Jones. Understanding segregation maps. http://www.umich.edu/~lawrace/understandingmaps.htm, 2002.

[4] Y. Bar-Yam. *Dynamics of Complex Systems*. Westview Press, July 1997.

[5] A. T. Bernardes, D. Stauffer, and J. Kertsz. Election results and the Sznajd model on Barabasi network. *The European Physical Journal B - Condensed Matter and Complex Systems*, 25:123 – 127, 01 2002.

[6] Gyan Bhanot. The metropolis algorithm. *Rep. Prog. Phys.*, 51:429–457, 1988.

[7] L.E. Blume and S.N. Durlauf. The interactions-based approach to socioeconomic behavior. Working papers 1, Wisconsin Madison - Social Systems, 2000.

[8] Paul Bourke. Uniform random number generator. online, 1990.

[9] William A. Brock and Steven N. Durlauf. Discrete choice with social interactions. *The Review of Economic Studies*, 68(2):235–260, 2001.

[10] Stephen G. Brush. History of the Lenz-Ising model. *Rev. Mod. Phys.*, 39(4):883–893, Oct 1967.

[11] David Chandler. *Introduction to Modern Statistical Mechanics*. Oxford Unversity Press, 1987.

[12] Harvey Gould and Jan Tobochnik. *Thermal and Statistical Physics*. Princeton University Press, 2010. Forthcoming, http://stp.clarku.edu/notes/.

[13] Mark S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[14] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[15] Sorin Istrail. Statistical mechanics, three-dimensionality and NP-completeness: I. universality of intracatability for the partition function of the ising model across non-planar surfaces (extended abstract). In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 87–96, New York, NY, USA, 2000. ACM.

[16] Charles Kittel. *An Introduction to Solid State Physics*. Wiley, seventh edition, 1995.

[17] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[18] M. Newman and G. Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1999.

[19] Martin Niss. History of the Lenz-Ising model 19201950: From ferromagnetic to cooperative phenomena. *Archive for History of Exact Sciences*, 59(3):267–318, March 2005.

[20] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65(3-4):117–149, Feb 1944.

[21] Thomas Schelling. *Micromotives and Macrobehavior*. Norton and Company, 1978.

[22] Daniel Schroeder. *An Introduction to Thermal Physics*. Addison Wesley Longman, 2000.

[23] J. P. Sethna. *Statistical Mechanics: Entropy, Order Parameters, and Complexity*. Oxford Unversity Press, 2006.

[24] D. Stauffer. Social applications of two-dimensional Ising models. *American Journal of Physics*, 76(4):470–473, 2008.

[25] D. Stauffer and S. Solomon. Ising, Schelling and self-organising segregation. *The European Physical Journal B - Condensed Matter and Complex Systems*, 57(4):473–479, 2007.

[26] D. Stauffer, A. O. Sousa, and S. M. de Oliveira. Generalization to Square Lattice of Sznajd Sociophysics Model. *International Journal of Modern Physics C*, 11:1239–1245, 2000.

[27] K. Sznajd-Weron. Sznajd model and its applications. *ACTA PHYSICA POLONICA B*, 36:2005, 2005.

[28] K. Sznajd-Weron and J. Sznajd. Opinion evolution in closed community. *International Journal of Modern Physics C*, 11:1157, 2000.

[29] K. Sznajd-Weron and J. Sznajd. Who is left, who is right? *Physica A: Statistical Mechanics and its Applications*, 351(2-4):593 – 604, 2005.

[30] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[31] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

Appendix A

Source code for 2D Ising model

filename: ising2d.c

```
001 /*
002  *  ising2d.c
003  *
004  *  Created by Yan Naung Oak on 26/02/09.
005  *  For Physics Thesis PHYS705 (Spring 2009)
006  * Professor Jeff Dunham
007  *
008  */
009
010 #include <stdio.h>
011 #include <stdlib.h>
012 #include <string.h>
013 #include <math.h>
014 #include <time.h>
015 #include "randomlib.h"
016 #include "randomlib.c"
017
018
019 #ifdef _APPLE__
020 #include <GLUT/glut.h>
021 #else
022 #include <windows.h>
023 #include <GL/glut.h>
024 #endif
025
026 #define GRID_SIZE 729
027 #define WIDTH  729
028 #define HEIGHT  729
029 #define RMINUS   1.0
030 #define GMINUS   1.0
031 #define BMINUS   0.0
032 #define RPLUS   0.0
033 #define GPLUS   0.0
034 #define BPLUS  1.0
035
036 #define NUMSNAPSHOTS 5
037
```

```
038
039 FILE *ifp, *ofp1, *ofp2;
040
041 double coupling = 1.0;
042 double externalField = 0.0;
043 long sweeps = 200000;
044 double temp = 0.1;
045 double tempMin = 1.87;
046 double tempMax = 2.67;
047 double tempStep = 0.4;
048
049 long updatePlot;
050 int pickX, pickY, spins, deltaE1;
051 float eMinus, ePlus, deltaE, deltaE2;
052 double beta, magnetization, energy, susceptibility, hCapacity, msq, esq;
053
054 /* Temporary variables for calculating statistics */
055 double n,m,e,z;
056 double magnetization_temp, energy_temp, msq_temp, esq_temp;
057 double magnetization_display, energy_display, msq_display, esq_display, susceptibility_display,
hCapacity_display;
058 double bracket_m, bracket_e;
059
060 int half_rand_max;
061 int seedTime1 = 0;
062 int seedTime2 = 0;
063 int sweepCounter = 0;
064 int snapCounter = 0;
065
066
067 /* Declare Grid */
068
069 int grid[GRID_SIZE][GRID_SIZE];
070 int initgrid[GRID_SIZE][GRID_SIZE];
071 int snapshots[GRID_SIZE][GRID_SIZE][NUMSNAPSHOTS+1];
072
073 // for periodic boundaries
074 int prev[GRID_SIZE];
075 int next[GRID_SIZE];
076
077 // for precalculating exponents
078 float exponents[18];
079
080
081 void isingLoop(int start, int end);
082 float calcMag();
083 void startLoop();
084 void exportData();
085 void resetGrid();
086
087 /*-----------------------------------------------*/
088 /*   Monte Carlo Metropolis Algorithm        */
089 /*-----------------------------------------------*/
090
091 void isingLoop(int start, int end)
092 {
093   int i;
094
095   for (i=start; i<end; i++) {
096     pickX = RandomInt(0,GRID_SIZE-1);//rand() % GRID_SIZE;
097     pickY = RandomInt(0,GRID_SIZE-1);//rand() % GRID_SIZE;
098     spins = grid[prev[pickX]][pickY] + grid[next[pickX]][pickY] + grid[pickX][prev[pickY]]
+ grid[pickX][next[pickY]];
099
100         if (grid[pickX][pickY] == 1) {
```

```
101       deltaE = 2*coupling*(float)spins + 2*externalField;
102         } else {
103            deltaE = (-2)*coupling*(float)spins -
2*externalField;
104         }
105
106         if (deltaE <= 0) {
107           grid[pickX][pickY] *= -1;
108         } else {
109       if(RandomDouble((double)0.0,(double)1.0) <
(double)exp(-(float)beta*deltaE))
110             grid[pickX][pickY] *= -1;
111         }
112
113   }
114 }
115
116 /*-----------------------------------------------*/
117 /*   Calculate Stats               */
118 /*-----------------------------------------------*/
119
120 float calcStats()
121 {
122   int i,j;
123   double boltzmann, bracket_e_temp, bracket_m_temp;
124   magnetization_temp = 0.0;
125   energy_temp = 0.0;
126   msq_temp = 0.0;
127   esq_temp = 0.0;
128   m = e = z = boltzmann = bracket_m_temp = bracket_e_temp = 0.0;
129
130   for (i=0; i<GRID_SIZE; i++) {
131     for (j=0; j<GRID_SIZE; j++) {
132       m = (double)grid[i][j];
133       e = coupling * (double)grid[i][j] *  (double)
(grid[next[i]][j] + grid[i][next[j]] + grid[prev[i]][j] + grid[i][prev[j]]);
134
135
136       magnetization_temp += m;
137       energy_temp -= e/2;
138       msq_temp += m*m;
139       esq_temp += e*e/4;
140     }
141   }
142
143
144   magnetization += magnetization_temp;
145   energy += energy_temp;
146   msq += msq_temp;
147   esq += esq_temp;
148   susceptibility = (beta)*((msq/(n*(sweepCounter - sweeps/2))) -
(magnetization/(n*(sweepCounter - sweeps/2)))*(magnetization/(n*(sweepCounter -
sweeps/2))));
149   hCapacity = (beta*beta)*((esq/(n*(sweepCounter - sweeps/2))) -
(energy/(n*(sweepCounter - sweeps/2)))*(energy/(n*(sweepCounter - sweeps/2))));
150 }
151
152
153 /*-----------------------------------------------*/
154 /*   Control Loop                  */
155 /*-----------------------------------------------*/
156
157 void startLoop()
158 {
159   int i,j;
```

```
160   n = (double)(GRID_SIZE*GRID_SIZE);
161   beta=1/(double)temp;
162   magnetization_display = energy_display = msq_display =
esq_display = hCapacity_display = susceptibility_display = 0.0;
163   magnetization = energy = susceptibility = hCapacity = msq = esq = 0.0;
164   bracket_m = bracket_e = 0.0;
165
166   while (sweepCounter < sweeps + 1){
167     isingLoop(1,(GRID_SIZE*GRID_SIZE));
168     sweepCounter++;
169     if (sweepCounter >= sweeps/2)
170     calcStats();
171
172     if (((sweepCounter - sweeps/2) % updatePlot == 0) && (sweepCounter > sweeps/2)){
173       magnetization_display = magnetization/(n*(sweepCounter - sweeps/2));
174       energy_display = energy/(n*(sweepCounter - sweeps/2));
175       susceptibility_display = susceptibility;////(sweepCounter - sweeps/2);
176       hCapacity_display = hCapacity;////(sweepCounter - sweeps/2);
177       snapCounter++;
178       printf("xField = %f, Temp = %f, Sweep %d, M = %f, E = %f,
M^2 = %f, E^2 = %f, Chi = %f, C = %f\n",
179         externalField,temp,sweepCounter,magnetization_display,energy_
display,msq_display,esq_display,susceptibility_display,
hCapacity_display);
180       fprintf(ofp2," %f, %f, %d, %f, %f, %f, %f\n",
181         externalField,temp,sweepCounter,magnetization_display,energy_
display,susceptibility_display,hCapacity_display);
182       for (i=0; i<GRID_SIZE; i++) {
183         fprintf(ofp1, "\n");
184         for (j=0; j<GRID_SIZE; j++) {
185           snapshots[i][j][snapCounter] = grid[i][j];
186           fprintf(ofp1, "%d, ",grid[i][j]);
187         }
188       }
189       //fprintf(ofp1, "\n");
190     }
191   }
192
193
194 }
195
196 /*-----------------------------------------------*/
197 /*   Reset Grid                    */
198 /*-----------------------------------------------*/
199
200 void resetGrid()
201 {
202   int i,j;
203   for (i=0; i<GRID_SIZE; i++)
204   for (j=0; j<GRID_SIZE; j++)
205     grid[i][j] = initgrid[i][j];
206 }
207
208
209
210 /*-----------------------------------------------*/
211 /*   Display                    */
212 /*-----------------------------------------------*/
213
214 void display()
215 {
216     int i,j;
217   int hRatio, wRatio;
218   hRatio = HEIGHT/GRID_SIZE;
219   wRatio = WIDTH/GRID_SIZE;
```

```
220
221   glClear(GL_COLOR_BUFFER_BIT);
222
223   for (i=0; i<GRID_SIZE; i++)
224     for (j=0; j<GRID_SIZE; j++){
225       if (grid[i][j]== -1)
226         glColor3f(RMINUS, GMINUS, BMINUS);
227       else if (grid[i][j]== 1)
228         glColor3f(RPLUS, GPLUS, BPLUS);
229       glRecti(i*wRatio,j*hRatio,(i+1)*wRatio,(j+1)*hRatio);
230
231     }
232   glFlush();
233   glutSwapBuffers();
234 }
235
236
237 /*-----------------------------------------------*/
238 /*   Export Data                 */
239 /*-----------------------------------------------*/
240
241 void exportData()
242 {
243
244 }
245
246
247 /*-----------------------------------------------*/
248 /*   Reshape                   */
249 /*-----------------------------------------------*/
250
251 void myReshape(int w, int h)
252 {
253     glViewport(0, 0, w, h);
254
255   glMatrixMode(GL_PROJECTION);
256     glLoadIdentity();
257   if (w <= h)
258     gluOrtho2D(0.0, 0.0, (GLfloat) WIDTH, (GLfloat) HEIGHT* (GLfloat) h / (GLfloat) w);
259     else
260     gluOrtho2D(0.0, 0.0, (GLfloat) WIDTH * (GLfloat) w / (GLfloat) h, (GLfloat) HEIGHT);
261   glMatrixMode(GL_MODELVIEW);
262
263   display();
264 }
265
266 /*-----------------------------------------------*/
267 /*   Initialize Variables              */
268 /*-----------------------------------------------*/
269
270 void myinit()
271 {
272   glClearColor (0.0, 0.0, 0.0, 0.0);
273     gluOrtho2D(0.0, (GLfloat) WIDTH, 0.0, (GLfloat) HEIGHT);
274
275   int i,j;
276   half_rand_max = RAND_MAX/2;
277   updatePlot = sweeps / (2*NUMSNAPSHOTS);
278   magnetization = energy = susceptibility = hCapacity = msq = esq = 0.0;
279
280   seedTime1 = rand() % 30000;
281   seedTime2 = rand() % 30000;
282   RandomInitialise(seedTime1,seedTime2);
283
284
```

```
285   for (i=0; i<GRID_SIZE; i++)
286     for (j=0; j<GRID_SIZE; j++){
287       if (RandomDouble((double)0.0,(double)1.0) < 0.5) {
288         grid[i][j] = -1;
289         initgrid[i][j] = -1;
290         snapshots[i][j][0] = -1;
291       } else {
292         grid[i][j] = 1;
293         initgrid[i][j]= 1;
294         snapshots[i][j][0] = 1;
295       }
296     }
297
298   for (i=1; i<GRID_SIZE-1; i++) {
299     next[i]=i+1;
300     prev[i]=i-1;
301   }
302
303   next[GRID_SIZE-1]=0;
304   next[0]=1;
305   prev[GRID_SIZE-1]=GRID_SIZE-2;
306   prev[0]=GRID_SIZE-1;
307 }
308
309
310
311 /*-----------------------------------------------*/
312 /*   Keyboard, Mouse and Menu Handlers        */
313 /*-----------------------------------------------*/
314
315 // Keyboard Listener
316 void keyboard(unsigned char key,int x, int y)
317 {
318    switch (key)
319    {
320    case 27:
321      exit(0);
322      break;
323    case 's':
324      fprintf(ofp2,"xField, Temp, Sweep, M , E, Chi, C\n");
325      startLoop();
326      display();
327      break;
328    case 'd':
329      fprintf(ofp2,"xField, Temp, Sweep, M , E, Chi, C\n");
330      for (temp=tempMin; temp<=tempMax; temp+=tempStep) {
331       startLoop();
332       display();
333       resetGrid();
334       exportData();
335       snapCounter=0;
336       sweepCounter=0;
337      }
338      break;
339    }
340 }
341
342
343
344
345 /*-----------------------------------------------*/
346 /*   Main                       */
347 /*-----------------------------------------------*/
348
349 main(int argc, char *argv[])
```

```
350 {
351   ofp1 = fopen("Ising2Doutput02.csv", "w");
352   ofp2 = fopen("Ising2Dsummary02.csv", "w");
353
354   srand(time(NULL));
355
356   time_t rawtime;
357   struct tm *timeinfo;
358   char mytime[19];
359
360   time ( &rawtime );
361   timeinfo = localtime ( &rawtime );
362   printf ( "Current local time and date: %d \n",
(*timeinfo).tm_year + 1900 );
363
364     glutInit(&argc, argv);
365     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB );
366     glutInitWindowSize(WIDTH, HEIGHT);
367     glutCreateWindow("Ising 2D");
368     myinit();
369     glutReshapeFunc(myReshape);
370     glutDisplayFunc(display);
371   glutKeyboardFunc (keyboard);
372
373     glutMainLoop();
374
375   fclose(ofp1);
376   fclose(ofp2);
377   return 0;
378 }
379
380


381
```

Appendix B

Source code for 3D Ising model

filename: ising3d.c

```
001 /*
002  *  ising2d.c
003  *
004  *  Created by Yan Naung Oak on 26/02/09.
005  *  For Physics Thesis PHYS705 (Spring 2009)
006  * Professor Jeff Dunham
007  *
008  */
009
010 #include <stdio.h>
011 #include <stdlib.h>
012 #include <string.h>
013 #include <math.h>
014 #include <time.h>
015 #include "randomlib.h"
016 #include "randomlib.c"
017
018
019 #ifdef _APPLE__
020 #include <GLUT/glut.h>
021 #else
022 #include <windows.h>
023 #include <GL/glut.h>
024 #endif
025
026 #define GRID_SIZE 50
027 #define WIDTH  400
028 #define HEIGHT  400
029 #define RMINUS   0.0
030 #define GMINUS   0.0
031 #define BMINUS   0.0
032 #define RPLUS   0.0
033 #define GPLUS   0.0
034 #define BPLUS  1.0
035 #define SC    1
036 #define BCC    2
037 #define FCC    3
```

```
038
039 #define NUMSNAPSHOTS 5
040
041
042 FILE *ifp, *ofp1, *ofp2;
043
044 double coupling = 1.0;
045 double externalField = 0.0;
046 long sweeps = 200000;
047 double temp = 4.5;
048 double tempMin = 2.50;
049 double tempMax = 3.0;
050 double tempStep = 0.02;
051
052 long updatePlot;
053 int pickX, pickY, pickZ, spins, deltaE1;
054 float eMinus, ePlus, deltaE, deltaE2;
055 double beta, magnetization, energy, susceptibility, hCapacity, msq, esq;
056
057 /* Temporary variables for calculating statistics */
058 double n,m,e,z;
059 double magnetization_temp, energy_temp, msq_temp,
esq_temp;
060 double magnetization_display, energy_display, msq_
display, esq_display, susceptibility_display, hCapacity_display;
061 double bracket_m, bracket_e;
062
063 int half_rand_max;
064 int seedTime1 = 0;
065 int seedTime2 = 0;
066 int sweepCounter = 0;
067 int snapCounter = 0;
068
069
070 /* Declare Grid */
071 int lattice = SC;
072 int grid[2*GRID_SIZE][2*GRID_SIZE][2*GRID_SIZE];
073 int initgrid[2*GRID_SIZE][2*GRID_SIZE][2*GRID_SIZE];
074
075 // for periodic boundaries
076 int prev[2*GRID_SIZE];
077 int next[2*GRID_SIZE];
078
079 // for display
080 static GLdouble viewer[]= {(float)WIDTH, (float)WIDTH,
(float)WIDTH}; /* initial viewer location */
081 static GLfloat theta[] = {0.0,5.0,5.0};
082 static GLint axis = 2;
083 int rotating = 0;
084 float radius = 0;
085
086 void isingLoop(int start, int end);
087 float calcMag();
088 void startLoop();
089 void exportData();
090 void resetGrid();
091 void renderSphere(float x, float y, float z,
float radius,int subdivisions,GLUquadricObj *quadric);
092 void renderSphere_convenient(float x, float y,
float z, float radius,int subdivisions);
093
094
095 /*----------------------------------------------*/
096 /*   Monte Carlo Metropolis Algorithm        */
097 /*----------------------------------------------*/
```

```
098
099 void isingLoop(int start, int end)
100 {
101   int i;
102
103   for (i=start; i<end; i++) {
104     pickX = RandomInt(0,GRID_SIZE-1);
105     pickY = RandomInt(0,GRID_SIZE-1);
106     pickZ = RandomInt(0,GRID_SIZE-1);
107     pickX *= 2;
108     pickY *= 2;
109     pickZ *= 2;
110
111     // for Simple Cubic Lattice (SC)
112     spins = grid[next[pickX]][pickY][pickZ] + grid[prev[pickX]][pickY][pickZ]
113         + grid[pickX][next[pickY]][pickZ] + grid[pickX][prev[pickY]][pickZ]
114         + grid[pickX][pickY][next[pickZ]] + grid[pickX][pickY][prev[pickZ]];
115
116         if (grid[pickX][pickY][pickZ] == 1) {
117       deltaE = 2*coupling*(float)spins + 2*externalField;
118         } else {
119            deltaE = (-2)*coupling*(float)spins - 2*externalField;
120         }
121
122         if (deltaE <= 0) {
123           grid[pickX][pickY][pickZ] *= -1;
124         } else {
125       if(RandomDouble((double)0.0,(double)1.0) <
126 (double)exp(-(float)beta*deltaE))
127             grid[pickX][pickY][pickZ] *= -1;
128         }
129
130   }
131 }
132
133 /*-----------------------------------------------*/
134 /*   Calculate Stats                             */
135 /*-----------------------------------------------*/
136
137 float calcStats()
138 {
139   int i,j,k;
140   double boltzmann, bracket_e_temp, bracket_m_temp;
141   magnetization_temp = 0.0;
142   energy_temp = 0.0;
143   msq_temp = 0.0;
144   esq_temp = 0.0;
145   m = e = z = boltzmann = bracket_m_temp = bracket_e_temp = 0.0;
146
147   for (i=0; i<2*GRID_SIZE; i++)
148   for (j=0; j<2*GRID_SIZE; j++)
149   for (k=0; k<2*GRID_SIZE; k++) {
150     if (grid[i][j][k] != 0){
151
152       m = (double)grid[i][j][k];
153       e = coupling * (double)grid[i][j][k] *
154       (double)(grid[next[i]][j][k] + grid[prev[i]][j][k] + grid[i][next[j]][k]
155 + grid[i][prev[j]][k] + grid[i][j][next[k]] + grid[i][j][prev[k]]);
156
157       magnetization_temp += m;
158       energy_temp -= e/2;
159       msq_temp += m*m;
160       esq_temp += e*e/4;
161     }
162   }
```

```
161
162
163   magnetization += magnetization_temp;
164   energy += energy_temp;
165   msq += msq_temp;
166   esq += esq_temp;
167   susceptibility = (beta)*((msq/(n*(sweepCounter - sweeps/2))) -
(magnetization/(n*(sweepCounter - sweeps/2)))*(magnetization/(n*(sweepCounter - sweeps/2))));
168   hCapacity = (beta*beta)*((esq/(n*(sweepCounter - sweeps/2))) -
(energy/(n*(sweepCounter - sweeps/2)))*(energy/(n*(sweepCounter - sweeps/2))));
169 }
170
171
172 /*-----------------------------------------------*/
173 /*   Control Loop                  */
174 /*-----------------------------------------------*/
175
176 void startLoop()
177 {
178   int i,j,k;
179   n = (double)(GRID_SIZE*GRID_SIZE*GRID_SIZE);
180   beta=1/(double)temp;
181   magnetization_display = energy_display = msq_display =
esq_display = hCapacity_display = susceptibility_display = 0.0;
182   magnetization = energy = susceptibility = hCapacity = msq = esq = 0.0;
183   bracket_m = bracket_e = 0.0;
184
185   while (sweepCounter < sweeps + 1){
186     isingLoop(1,(int)n/8);
187     sweepCounter++;
188     if (sweepCounter >= sweeps/2)
189     calcStats();
190
191     if (((sweepCounter - sweeps/2) % updatePlot == 0) && (sweepCounter > sweeps/2)){
192       magnetization_display = magnetization/(n*(sweepCounter - sweeps/2));
193       energy_display = energy/(n*(sweepCounter - sweeps/2));
194       susceptibility_display = susceptibility;////(sweepCounter - sweeps/2);
195       hCapacity_display = hCapacity;////(sweepCounter - sweeps/2);
196       snapCounter++;
197       printf("xField = %f, Temp = %f, Sweep %d, M = %f, E = %f,
M^2 = %f, E^2 = %f, Chi = %f, C = %f\n",
198         externalField,temp,sweepCounter,magnetization_display,energy_
display,msq_display,esq_display,susceptibility_display,
hCapacity_display);
199       fprintf(ofp2," %f, %f, %d, %f, %f, %f, %f\n",
200         externalField,temp,sweepCounter,magnetization_display,energy_
display,susceptibility_display,hCapacity_display);
201       for (i=0; i<2*GRID_SIZE; i++) {
202         fprintf(ofp1, "\n");
203         for (j=0; j<2*GRID_SIZE; j++)
204         for (k=0; k<2*GRID_SIZE; k++){
205           //snapshots[i][j][snapCounter] = grid[i][j];
206           fprintf(ofp1, "%d, ",grid[i][j][k]);
207         }
208       }
209     }
210   }
211
212 }
213
214 /*-----------------------------------------------*/
215 /*   Reset Grid                    */
216 /*-----------------------------------------------*/
217
218 void resetGrid()
```

```
219 {
220   int i,j,k;
221   for (i=0; i<2*GRID_SIZE; i++)
222   for (j=0; j<2*GRID_SIZE; j++)
223   for (k=0; k<2*GRID_SIZE; k++)
224     grid[i][j][k] = initgrid[i][j][k];
225 }
226
227
228
229 /*-----------------------------------------------*/
230 /*   Display                     */
231 /*-----------------------------------------------*/
232
233 void display()
234 {
235     int i,j,k;
236   int hRatio, wRatio;
237   hRatio = HEIGHT/GRID_SIZE;
238   wRatio = WIDTH/GRID_SIZE;
239
240   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
241   glLoadIdentity();
242   gluLookAt(viewer[0],viewer[1],viewer[2], 0.0, 0.0, 0.0, viewer[0]-1.5, viewer[1]-1.5,
viewer[2]+1.0);
243
244   /* rotate cube */
245
246   glRotatef(theta[0], 1.0, 0.0, 0.0);
247   glRotatef(theta[1], 0.0, 1.0, 0.0);
248   glRotatef(theta[2], 0.0, 0.0, 1.0);
249
250
251   for (i=0; i<2*GRID_SIZE; i++)
252   for (j=0; j<2*GRID_SIZE; j++)
253   for (k=0; k<2*GRID_SIZE; k++)
254   if (grid[i][j][k] == 1){
255     glColor3f(RPLUS, GPLUS, BPLUS);
256     renderSphere_convenient((float)(i*2)*radius-((float)WIDTH/2.0),
257     (float)(j*2)*radius-((float)WIDTH/2.0),(float)(k*2)*radius-((float)WIDTH/2.0),radius,8);
258     //glRecti(i*wRatio,j*hRatio,(i+1)*wRatio,(j+1)*hRatio);
259   }
260   glFlush();
261   glutSwapBuffers();
262 }
263
264 // the functions renderSphere() and renderSphere_convenient() are obtained from:
265 // http:// lifeofaprogrammergeek.blogspot.com/2008/08/opengl-example-rendering-cylinders.html
266 void renderSphere(float x, float y, float z,
float radius,int subdivisions,GLUquadricObj *quadric)
267 {
268   glPushMatrix();
269   glTranslatef( x,y,z );
270   gluSphere(quadric, radius, subdivisions,subdivisions);
271   glPopMatrix();
272 }
273
274 void renderSphere_convenient(float x, float y,
float z, float radius,int subdivisions)
275 {
276   //the same quadric can be re-used for drawing many spheres
277   GLUquadricObj *quadric=gluNewQuadric();
278   gluQuadricNormals(quadric, GLU_SMOOTH);
279   renderSphere(x,y,z,radius,subdivisions,quadric);
280   gluDeleteQuadric(quadric);
```

```
281 }
282
283 /*-----------------------------------------------*/
284 /*   Export Data                 */
285 /*-----------------------------------------------*/
286
287 void exportData()
288 {
289
290 }
291
292
293 /*-----------------------------------------------*/
294 /*   Reshape                   */
295 /*-----------------------------------------------*/
296
297 void myReshape(int w, int h)
298 {
299   glViewport(0, 0, w, h);
300
301   /* Use Orthogonal view */
302
303   glMatrixMode(GL_PROJECTION);
304   glLoadIdentity();
305
306
glOrtho(-2.0*(float)WIDTH,2.0*(float)WIDTH,-2.0*(float)WIDTH,2.0*(float)WIDTH,2.0*(float)WIDTH,-2.0*(float)WIDTH);
307   /*
308   if(w<=h) glFrustum(-(float)WIDTH, (float)WIDTH, -(float)HEIGHT,
309     (float)HEIGHT , 50, 3.0*(float)WIDTH);
310   else glFrustum(-(float)WIDTH, (float)WIDTH, -(float)HEIGHT,
311     (float)HEIGHT , 50, 3.0*(float)WIDTH);
312   */
313   glMatrixMode(GL_MODELVIEW);
314
315   display();
316 }
317
318 /*-----------------------------------------------*/
319 /*   Initialize Variables            */
320 /*-----------------------------------------------*/
321
322 void myinit()
323 {
324   glClearColor (0.0, 0.0, 0.0, 0.0);
325
326   int i,j,k;
327   half_rand_max = RAND_MAX/2;
328   updatePlot = sweeps / (2*NUMSNAPSHOTS);
329   magnetization = energy = susceptibility = hCapacity = msq = esq = 0.0;
330
331   radius = (float)WIDTH/(2.5*GRID_SIZE);
332
333   seedTime1 = 12345;//rand() % 30000;//12345
334   seedTime2 = 23456;//rand() % 30000;//23456
335   RandomInitialise(seedTime1,seedTime2);
336
337   for (i=0; i<2*GRID_SIZE; i++)
338   for (j=0; j<2*GRID_SIZE; j++)
339   for (k=0; k<2*GRID_SIZE; k++)
340   grid[i][j][k] = 0;
341
342   for (i=0; i<2*GRID_SIZE; i+=2)
343   for (j=0; j<2*GRID_SIZE; j+=2)
344   for (k=0; k<2*GRID_SIZE; k+=2) {
```

```
345     if (RandomDouble((double)0.0,(double)1.0) < 0.5) {
346       grid[i][j][k] = -1;
347       initgrid[i][j][k] = -1;
348       //snapshots[i][j][0] = -1;
349     } else {
350       grid[i][j][k] = 1;
351       initgrid[i][j][k]= 1;
352       //snapshots[i][j][0] = 1;
353     }
354   }
355
356   for (i=2; i<2*GRID_SIZE-2; i++) {
357     next[i]=i+2;
358     prev[i]=i-2;
359   }
360
361   next[2*GRID_SIZE-1]=1;
362   next[2*GRID_SIZE-2]=0;
363   next[0]=2;
364   next[1]=3;
365   prev[2*GRID_SIZE-1]=2*GRID_SIZE-3;
366   prev[2*GRID_SIZE-2]=2*GRID_SIZE-4;
367   prev[0]=2*GRID_SIZE-2;
368   prev[1]=2*GRID_SIZE-1;
369 }
370
371
372
373 /*-----------------------------------------------*/
374 /*   Keyboard, Mouse and Menu Handlers        */
375 /*-----------------------------------------------*/
376
377 // Keyboard Listener
378 void keyboard(unsigned char key,int x, int y)
379 {
380    switch (key)
381    {
382    case 27:
383      exit(0);
384      break;
385    case 's':
386      fprintf(ofp2,"xField, Temp, Sweep, M , E, Chi, C\n");
387      startLoop();
388      display();
389      resetGrid();
390      snapCounter=0;
391      sweepCounter=0;
392      temp+=tempStep;
393      break;
394    case 'd':
395      fprintf(ofp2,"xField, Temp, Sweep, M , E, Chi, C\n");
396      for (temp=tempMin; temp<=tempMax; temp+=tempStep) {
397       startLoop();
398       display();
399       resetGrid();
400       snapCounter=0;
401       sweepCounter=0;
402      }
403      break;
404    case 'r':
405      theta[0] = 0.0; theta[1] = 5.0; theta[2] = 5.0;
406      display();
407      break;
408    }
409 }
```

```
410
411 // Mouse Handler
412 void mouse(int button, int state, int x, int y)
413 {
414   rotating = 1;
415   if(button==GLUT_LEFT_BUTTON && state == GLUT_DOWN) axis = 0;
416   if(button==GLUT_MIDDLE_BUTTON && state == GLUT_DOWN) axis = 1;
417   if(button==GLUT_RIGHT_BUTTON && state == GLUT_DOWN) axis = 2;
418   theta[axis] += 2.0;
419   if( theta[axis] > 360.0 ) theta[axis] -= 360.0;
420   display();
421 }
422
423
424 /*------------------------------------------------*/
425 /*   Main                        */
426 /*------------------------------------------------*/
427
428 main(int argc, char *argv[])
429 {
430   ofp1 = fopen("Ising3Doutput2530.csv", "w");
431   ofp2 = fopen("Ising3Dsummary2530.csv", "w");
432
433   srand(time(NULL));
434
435   time_t rawtime;
436   struct tm *timeinfo;
437   char mytime[19];
438
439   time ( &rawtime );
440   timeinfo = localtime ( &rawtime );
441   printf ( "Current local time and date: %d \n",
(*timeinfo).tm_year + 1900 );
442
443     glutInit(&argc, argv);
444     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB );
445     glutInitWindowSize(WIDTH, HEIGHT);
446     glutCreateWindow("Ising 3D");
447     myinit();
448     glutReshapeFunc(myReshape);
449     glutDisplayFunc(display);
450   glutKeyboardFunc(keyboard);
451   glutMouseFunc(mouse);
452     glutMainLoop();
453
454   fclose(ofp1);
455   fclose(ofp2);
456   return 0;
457 }
458
459


460
```

Appendix C

Source code for Monte Carlo renormalization

filename: Renormalize.m

```
001 % Reads csv file and applies renormalization group by renormalizing one
002 % 3-by-3 cell at a time.
003
004 gridSize = 729;
005 Array = csvread('Ising2Doutput04-01-09.csv');
006 CurrentSnapshot = zeros(gridSize);
007 dimension = 2;
008 avgSize = 3;
009 colormap(gray);
010
011 renormSteps = (log(gridSize)/log(3))-1;
012 iterations = size(Array,1)/gridSize;
013 pic = zeros(gridSize);
014
015 for i=1:iterations
016     CurrentSnapshot = Array((i-1)*gridSize+1:i*gridSize,1:gridSize);
017     for j=renormSteps:-1:1
018         for k=1:avgSize:size(CurrentSnapshot,1)
019             for l=1:avgSize:size(CurrentSnapshot,2)
020                 if sum(sum(CurrentSnapshot(k:k+avgSize-1,l:l+avgSize-1))) > 0
021                     CurrentSnapshot(k:k+avgSize-1,l:l+avgSize-1) = ones(avgSize);
022                 else
023                     CurrentSnapshot(k:k+avgSize-1,l:l+avgSize-1) = -1*ones(avgSize);
024                 end
025             end
026         end
027         pic = imagesc(CurrentSnapshot);
028         if (i<10)
029             saveas(pic,strcat('Renorm2D00', int2str(i),'step',int2str(renormSteps-j)),'png');
030         elseif (i>9 && i<100)
031             saveas(pic,strcat('Renorm2D0', int2str(i),'step',int2str(renormSteps-j)),'png');
```

```
032         else
033             saveas(pic,strcat('Renorm2D', int2str(i),'step',int2str(renormSteps-j)),'png');
034         end
035         avgSize = avgSize*3;
036     end
037     avgSize = 3;

038 endd
```