Universiteit Utrecht

Faculteit Bètawetenschappen

A comparison of cluster algorithms for the bond-diluted Ising model

Master Thesis

A.H. Kole

Theoretical Physics & Computing Science

Supervisors:

Prof. Dr. G.T. Barkema
Department of Information and Computing Sciences
Dr. L. Fritz
Institute for Theoretical Physics

July, 2021

Abstract

Monte Carlo cluster algorithms are popular for their efficiency in studying the Ising model near its critical temperature. We might expect that this efficiency extends to the bond-diluted Ising model. We show, however, that this is not always the case by comparing how the correlation times τ_w and τ_{sw} of the Wolff and Swendsen-Wang cluster algorithms scale as a function of the system size L when applied to the two-dimensional bond-diluted Ising model. We demonstrate that the Wolff algorithm suffers from a much longer correlation time than in the pure Ising model, caused by isolated (groups of) spins which are infrequently visited by the algorithm. These cause the correlation time to scale as $\tau_w \sim L^{z_w}$ with a dynamical exponent $z_w = \gamma/\nu \approx 1.75$ independent of the bond concentration p for $0.5 < p < 1$. Moreover, we show that the Swendsen-Wang algorithm does not suffer from the same problem. Consequently, it has a much shorter correlation time, shorter than in the pure Ising model even. Numerically at $p = 0.6$, we find that its dynamical exponent is $z_{\mathrm{sw}} = 0.09(4)$. Lastly, we tested a novel way of determining the dynamical exponent for the Metropolis algorithm and confirmed that it worked properly. With this method we determined that $z_m = 3.337(3)$ at $p = 0.6$.

Acknowledgement

First I would like to thank my supervisors, Gerard Barkema and Lars Fritz, for our weekly talks to discuss the progress of my thesis. They kept me motivated and stimulated me to be at my best and to make the project my own. Their feedback helped me to improve and to know when I was on the right track. I would also like to thank Gerard for suggesting what became the topic of my thesis. Finding a topic that had enough depth on both the Theoretical Physics and Computing Science side was not easy so this was very helpful. On that note, let me also thank Joost de Graaf, Matthijs Vákár, Marjan van den Akker and all others who helped me with my search for a suitable topic. Furthermore, I would be remiss not to mention the members of the Friday group meeting who gave me a place to talk about what was going on in my thesis. Lastly, I would like to thank my friends and family who provided me with much-needed distraction and support whenever I needed it throughout this year of working at home. In particular, I want to thank my parents who have always been supportive through all of my studies as well as Marit who helped me when I was struggling to manage my time and to find a good way to give structure to my days.

Contents

1 Introduction

The Ising model is one of the most popular models in statistical physics because its simplicity makes it easy to study while it is complex enough that a lot can be learnt from it [1]. Since its inception, numerous variants of the Ising model have been proposed to study different phenomena. An important class of such variants are the Ising models with impurities. These are used to investigate how the presence of impurities, which occur frequently in nature, affect the properties of a system. Common ways to model impurities in the Ising model is by randomly removing spins (site-dilution [2–4]), bonds (bond-dilution [5–9]) or alternatively by randomly modifying the strength of the interactions in some other way [6, 10]. In this thesis we will focus on the variant with bond-dilution.

The introduction of bond-dilution to the model changes its properties significantly. For example, it has been shown that the critical temperature that separates the ferromagnetic and paramagnetic phases of the Ising model changes depending on the extent of the bond-dilution [8]. This even introduces a new type of phase transition because the critical temperature drops to zero at a certain bond concentration creating two phases (zero and non-zero critical temperature) separated by what is referred to as the percolation threshold [11]. In addition, it appears that the presence of impurities also alters the university class of the model [2].

A common approach to study the Ising model is the use of Monte Carlo methods. The choice of the algorithm should not change any of the equilibrium properties, as all algorithms sample the same (Boltzmann) distribution. However, the dynamics of different algorithms can vary wildly leading to big differences in the efficiency for studying a certain model. In the pure Ising model cluster algorithms such as the Wolff and Swendsen-Wang algorithms have proven themselves to be much more effective at criticality than single spin-flip algorithms like Metropolis [12]. This difference is expected to be even more pronounced in the bond-diluted Ising model since it has been recently shown that single spin-flip algorithms suffer from a diverging correlation time when the percolation threshold is approached [5]. The dynamics of cluster algorithms for the bond-diluted Ising model remains poorly studied and so it is still unclear whether they actually are more effective. Some studies have proposed that the efficiency of these cluster algorithms carry over to the bond-diluted Ising model and that correlation times actually decrease when site- or bond-dilution is introduced [4, 13]. We will present a quantitative analysis of the dynamics of the Wolff and Swendsen-Wang algorithms to show that this is in fact not the case for the Wolff algorithm. We will demonstrate that the Wolff algorithm suffers from much longer correlation times than in the pure model, caused by isolated (groups of) spins, a fact which was previously hinted at by Ballesteros et al [14].

This thesis is organised as follows. Sections 2 through 6 are dedicated to the theory that will help us to understand our results. Here we will introduce the principles behind Monte Carlo algorithms, the Ising model both in its pure and bond-diluted form, the algorithms that were studied, how to characterise their efficiency and what observables and techniques were used in our analysis. Section 7 is about the results from our simulations and what we can learn from them. Finally, we summarise our main results in section 8 and discuss potential avenues for further research.

2 Monte Carlo methods in Statistical Physics

Statistical physics is a branch of thermodynamics that studies systems that can, as the name already suggests, be accurately described with statistics. Within statistical physics you have multiple ensembles which represent different types of systems. The ensemble that we are interested in for this thesis is the canonical ensemble which describes a system in contact with a heat bath or thermal reservoir at temperature T [15]. The system can exchange energy with this thermal reservoir and through this exchange our system is also kept at the temperature T [15]. No other exchanges are allowed so the number of particles N in our system and the volume V are fixed as well as the temperature T of the reservoir [15]. It can be shown that the possible states for such a system follow a probability distribution called the Boltzmann distribution P and that this distribution is proportional to the so-called Boltzmann factor $P \sim e^{-\beta E}$ where E is the energy of a state and $\beta = \frac{1}{k_B T}$ with k_B the Boltzmann constant and T the temperature [15]. We have plotted the Boltzmann factor in figure 2.1. An important thing to note in this plot is that the Boltzmann factor rapidly decays so

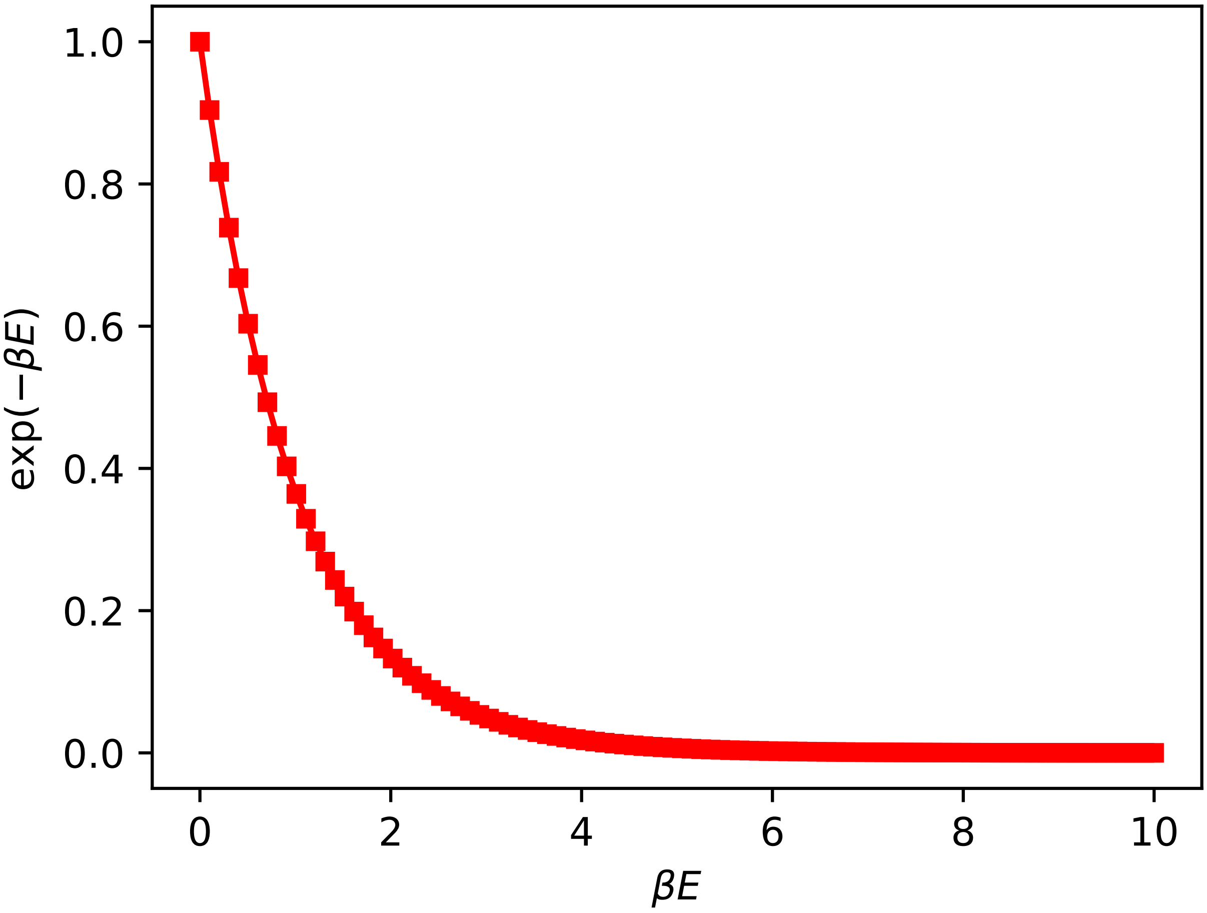


Figure 2.1: Plot of the Boltzmann factor $e^{-\beta E}$, where $\beta = \frac{1}{k_B T}$ and E the energy of a state. Note how it rapidly decays such that the probability of finding a state is very small if $\beta E \gg 1$.

that the probability of finding the system in a particular state quickly becomes very small if $\beta E \gg 1$.

When studying these systems, we are usually interested in calculating properties that could also be measured in experiments. For example, if the system is a gas of particles, we might be interested in its pressure under certain conditions. In statistical physics these properties can usually be calculated as ensemble averages. Say that we are interested in some quantity Q then its ensemble average $\langle Q \rangle$ is the average of the value that the quantity has in every possible state weighted with the probability of that state, so we can write it as [12],

$$\langle Q \rangle = \frac{\sum_\mu Q_\mu e^{-\beta E_\mu}}{\sum_\mu e^{-\beta E_\mu}} \tag{2.1}$$

where the sum is over all states μ, Q_μ is the value of Q in state μ and E_μ is the energy of state μ.

While it might seem straightforward to compute this average, in reality the total number of possible states is often so large that computing this sum explicitly becomes intractable [12]. So instead we might try to sum over a subset of M states to get an estimate Q_M of the ensemble average $\langle Q \rangle$. This leaves the question of how to choose the subset to sum over. Say that we want to sample the states at random from the state space according to some probability distribution p_μ where p_μ is the probability of choosing state μ. With this choice our subset of states is $\{\mu_1, \mu_2, \ldots, \mu_M\}$ and our estimator Q_M becomes [12],

$$Q_M = \frac{\sum_{i=1}^{M} Q_{\mu_i} p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{j=1}^{M} p_{\mu_j}^{-1} e^{-\beta E_{\mu_j}}}. \tag{2.2}$$

This still leaves us with the choice of the probability distribution p_μ. The easiest choice would be to sample the states uniformly at random so that all probabilities are equal and p_μ = constant [12]. With this choice the estimator becomes [12],

$$Q_M = \frac{\sum_{i=1}^{M} Q_{\mu_i} e^{-\beta E_{\mu_i}}}{\sum_{j=1}^{M} e^{-\beta E_{\mu_j}}} \tag{2.3}$$

where we have cancelled all probabilities since they are constant and equal. For many cases, however, this turns out to be a poor choice because as we already saw the Boltzmann factor will be very small for many of the states so that there is a good chance that we will mostly sample states with a small Boltzmann factor that do not really contribute to the ensemble average making our estimator Q_M a bad representation of $\langle Q \rangle$ [12]. Equilibrium Monte Carlo methods were proposed as a solution to this problem. In its essence, a Monte Carlo algorithm is a recipe for sampling the states in such a way that the probability of sampling a state is exactly proportional to the Boltzmann factor, i.e. $p_\mu \sim e^{-\beta E_\mu}$ [12]. We call this importance sampling. If we now plug $p_\mu \sim e^{-\beta E_\mu}$ into equation 2.2 we find that our estimator becomes a simple expression [12],

$$Q_M = \frac{1}{M} \sum_{i=1}^{M} Q_{\mu_i}. \tag{2.4}$$

So we can compute our estimator by simply computing the arithmetic mean of our subset of states and this will in general give us a much better estimate of the ensemble average [12].

Now that we know that a Monte Carlo algorithm tries to sample the states of a system according to the Boltzmann distribution, we can answer the question of how this sampling could work. The Monte Carlo algorithms we consider in this thesis are based on a Markov process. This means that they work by generating a new sample from an existing sample [12]. So say that we have a state μ our Monte Carlo algorithm will be a recipe for producing a new state ν. We then obtain our subset of states by repeating this recipe until enough states have been generated. This generation is usually a stochastic process where we have a collection of transition probabilities $P(\mu \to \nu)$ dictating the probability that we will generate state ν if we are in state μ [12]. These probabilities of course need to be normalised so,

$$\sum_\nu P(\mu \to \nu) = 1. \tag{2.5}$$

The key ingredient for constructing a correct Monte Carlo algorithm that generates states in such a way that the probability of finding a state is exactly proportional to its Boltzmann factor is to choose these transition probabilities so that they satisfy certain properties. These properties will then ensure that we will sample the states according to the correct Boltzmann distribution. The first condition is called ergodicity and it is the requirement that we should always be able to reach any state of the system from any other state if we run the algorithm for long enough [12]. Note that this does not require that we are not allowed to set some transition probabilities to zero, but they should be chosen so that there is always a path of possible transitions between any pair of states [12]. The second condition is that we require our Markov process to satisfy detailed balance. This means that we require that in equilibrium for any pair of states μ and ν the rate of transitions from μ to ν is equal to the rate of transitions from ν to μ [12]. We can write this down as [12],

$$p_\mu P(\mu \to \nu) = p_\nu P(\nu \to \mu) \tag{2.6}$$

where p_μ is the probability of being in state μ in our simulation. If we rewrite this a bit and remember that we want that $p_\mu \sim e^{-\beta E_\mu}$ we find that,

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{p_\nu}{p_\mu} = e^{-\beta(E_\nu - E_\mu)}. \tag{2.7}$$

In this form detailed balance gives us a concrete condition that our choice of transition probabilities has to satisfy. To summarise, if we can show that a Monte Carlo algorithm satisfies the conditions of ergodicity and detailed balance we are ensured that it is correct and that it will generate states according to the correct Boltzmann distribution [12]. There are still a lot of ways in how we can satisfy these two conditions, but discussing this is beyond the scope of this thesis. We will, however, give some examples of correct Monte Carlo algorithms later on.

Two final remarks are in order. First of all, we mentioned that we want to satisfy detailed balance in equilibrium, but how do we know if our Markov process is in equilibrium? Also, how do we start the Markov process, i.e. what should be our initial state? These are not always easy questions to answer, but luckily it can be shown that satisfying these two conditions also ensures that no matter what state we start our simulation in our Markov process will always tend towards equilibrium if we run the simulation for a sufficiently long time [12]. Consequently, our sample probabilities p_μ will always tend to the Boltzmann distribution [12]. The process of running the simulation for a sufficiently long time to reach equilibrium is referred to as thermalisation [12]. Secondly, the conditions of ergodicity and detailed balance are sufficient conditions for a correct Monte Carlo algorithm but they are not necessary [12]. Therefore, it is also possible to construct an algorithm which is correct but that does not satisfy these conditions. However, we will only consider algorithms which do satisfy these conditions in this thesis.

3 The Ising model

Before giving some examples of Monte Carlo algorithms it is useful to introduce the model that we studied in our simulations. We will start with the pure Ising model and then explain how to introduce bond-dilution to get the bond-diluted Ising model.

3.1 The pure Ising model

Of all statistical physics models the pure Ising model is certainly one of the most well studied [1]. In its essence it is a model for a physical system consisting of particles arranged on a regular lattice where each particle can be in two states [1]. It is usually introduced as being a model for atoms in a crystal where each atom acts like a little magnet and has a spin which can be either up or down, and this is also the interpretation that we will use in this thesis, but it could be equally well-used to describe any other lattice system where each lattice point can be in two states such as a mixture of two kinds of molecules or a mixture of particles and holes (empty spaces) [1]. Furthermore, it is the pure Ising model in the sense that we are considering the model without any modifications or extensions as it was originally proposed. We are free to choose the lattice so we will restrict ourselves to one of the simpler cases, a two-dimensional square lattice. With this choice we can visualise the model as in figure 3.1.

As can be seen in the figure, our particles, which we will refer to as spins since we are using the magnet interpretation of the Ising model, are arranged on the lattice points of a two-dimensional square lattice. Each particle can be in either one of two states which we refer to as up and down respectively. A single state or configuration of the Ising model can then simply be obtained by assigning the choice up or down to each of the spins in the lattice. An example state is given in figure 3.2. In principle, a lattice should be infinite. However, to be able to run Monte Carlo simulations we need finite size systems so we define the system size L to be the number of spins on one side of the square as is illustrated in figure 3.3. With this definition our lattice will contain a total of N spins where $N = L^2$ for a two-dimensional square lattice.

The model would not be very interesting if the spins did not interact with each other. However, if we add too many interactions the model can quickly become intractable. Luckily, a lot of interesting things can already be learned if we use the approximation that spins only interact with their nearest neighbours. With this approximation, the Hamiltonian for the system can be written as [12],

$$H = -J \sum_{\langle ij \rangle} s_i s_j \tag{3.1}$$

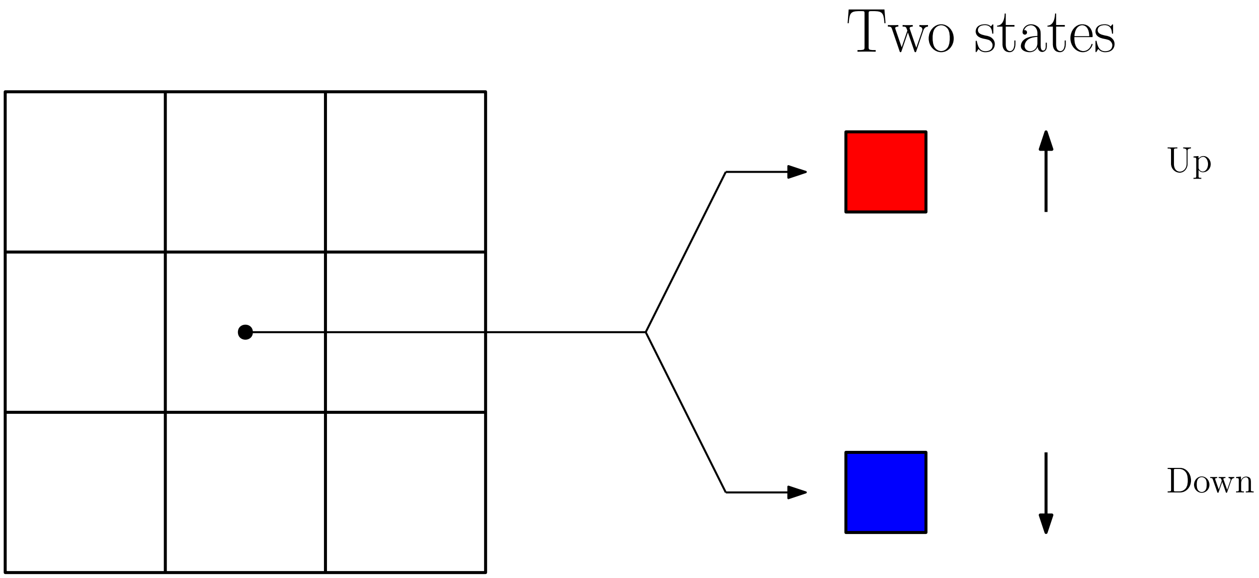


Figure 3.1: Overview of our conventions for the pure two-dimensional Ising model on a square lattice. Each square represents one of the spins in the lattice which can be in one of two states as is illustrated on the right. The same conventions will be used in all the Ising model illustrations.

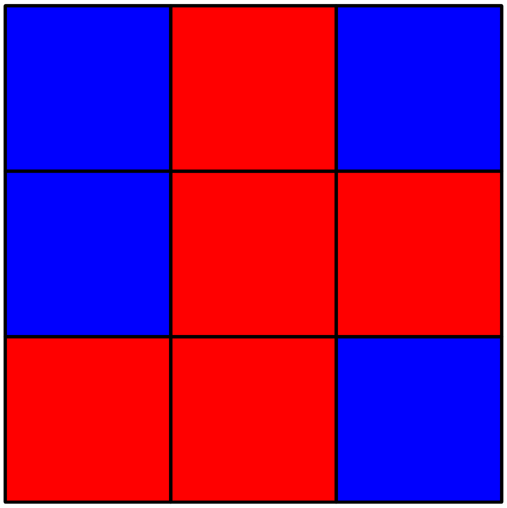


Figure 3.2: An example of what a state would look like with the conventions from figure 3.1.

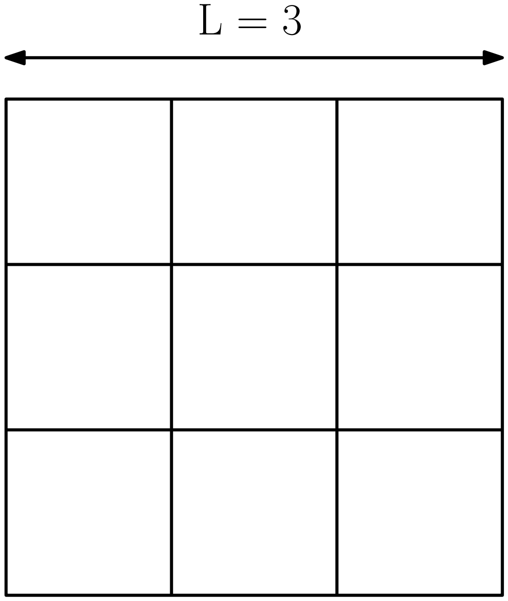


Figure 3.3: An illustration to explain how the size of the lattice L is defined in our model.

where the sum is over all nearest neighbour pairs in the lattice, s_i represents the spin of site i which can take on two values ($s_i = \pm 1$) and J is the coupling constant which is a measure of the strength of the interaction. Intuitively, each spin interacts with each of its four neighbours and if two neighbours are in the same state (i.e. the spins are parallel) this gives a contribution of $-J$ to the total energy and if two spins are in opposite states (i.e. they are anti-parallel) this gives a contribution $+J$ to the total energy. The Hamiltonian then is the sum of all these contributions. This is depicted in figure 3.4. Sometimes people also include an external field term in the Hamiltonian, but we will ignore that here and only consider the zero field case [12]. Besides the energy another interesting property of a state is the sum over all the spin values s_i. Since each spin represents a magnet this quantity is referred to as the total magnetisation M of the lattice and it is given by [12],

$$M = \sum_i s_i. \tag{3.2}$$

See figure 3.5 for a diagrammatic explanation of the magnetisation.

The aforementioned discussion hopefully convinces the reader that the Ising model is indeed simple enough to be easy to study. Yet, despite its simplicity the model has enough complexity to exhibit some interesting properties. One of the most noteworthy is that it actually has a phase transition as a function of the temperature T [12]. As we vary the temperature the model will transition from a ferromagnetic phase, where on average the majority of spins are pointing in the same direction (i.e. the average total magnetisation M is non-zero), to a paramagnetic phase where the orientation of the spins is practically random such that the average total magnetisation M is zero [12]. This can be nicely visualised by plotting the average total magnetisation per spin M/N as is done in figure 3.6.

The critical temperature T_c at which this phase transition occurs can be determined numerically in many ways, for example by using Monte Carlo methods. One of the most impressive accomplishments in the history of the Ising model, however, is the exact analytical solution of the two-dimensional Ising model by Lars Onsager in 1942 [1]. This solution also gives us the critical temperature which turns out to be $T_c = 2J/\log\left(1+\sqrt{2}\right) \approx 2.269J$ [12]. The presence of a phase transition means that the model can also be used to study the behaviour of a system at criticality (i.e. in the vicinity of the critical temperature) and as we will see later this is also interesting for studying the dynamics of Monte Carlo algorithms.

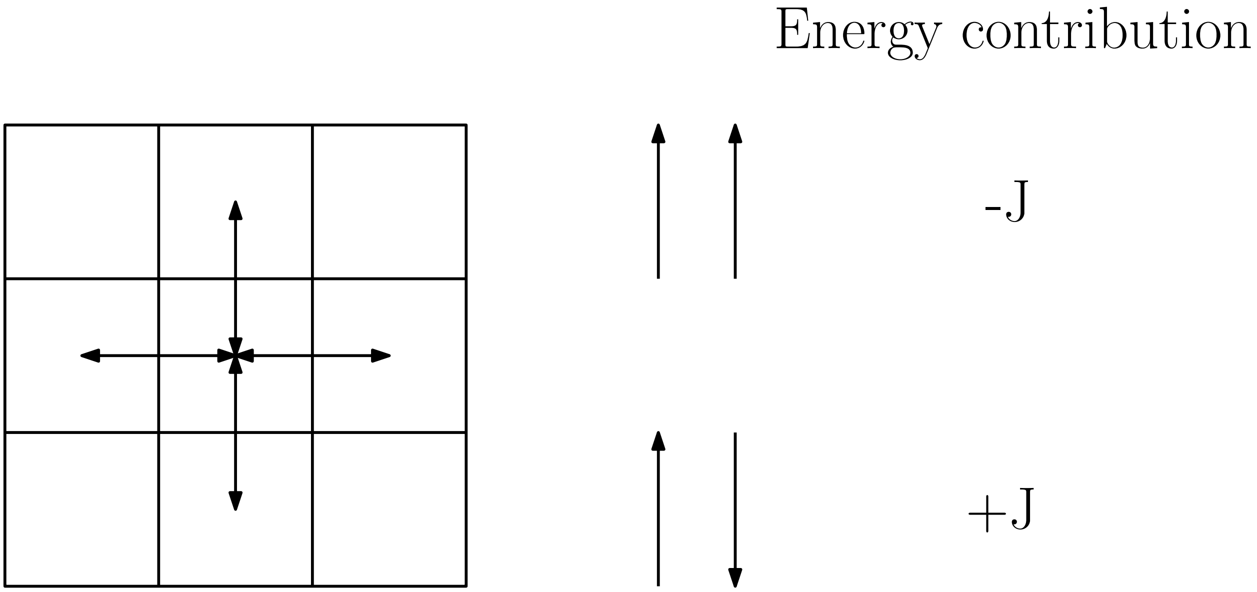


Figure 3.4: An illustration to explain the interactions that contribute to the energy of a state for the pure two-dimensional Ising model. Each spin (square) interacts with each of its four nearest neighbours. The contribution of such an interaction to the total energy is given on the right. Here J is the coupling constant that also occurs in the Hamiltonian.

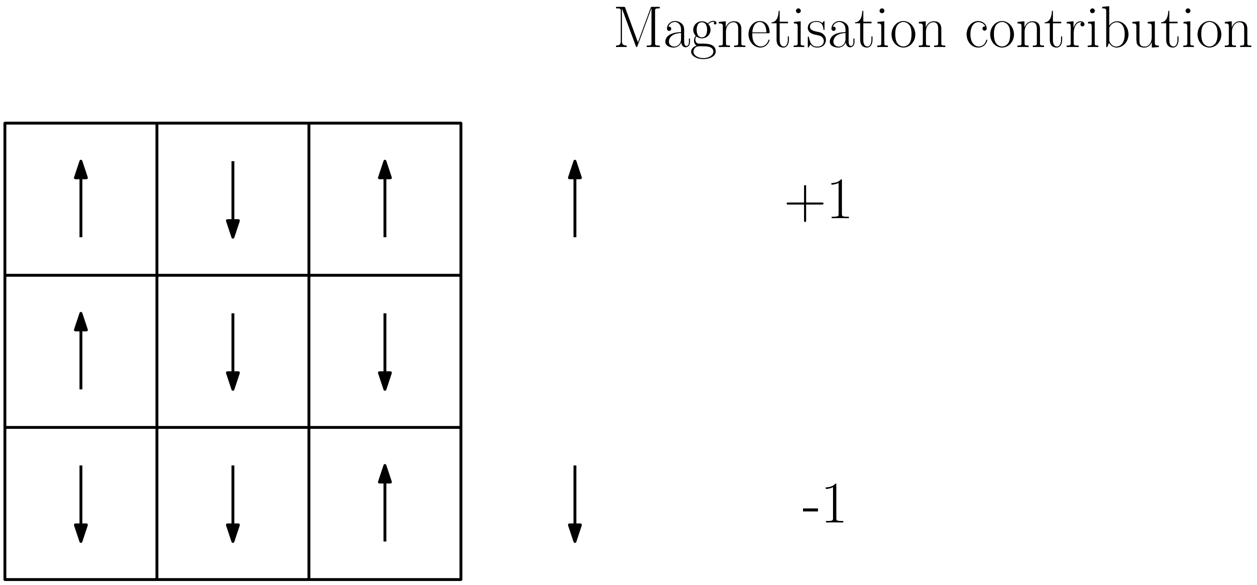


Figure 3.5: An illustration to explain the magnetisation of a state for the pure two-dimensional Ising model. Each spin is assigned a contribution to the total magnetisation depending on the state of the particular spin as indicated on the right. The total magnetisation is then simply the sum of all these contributions.

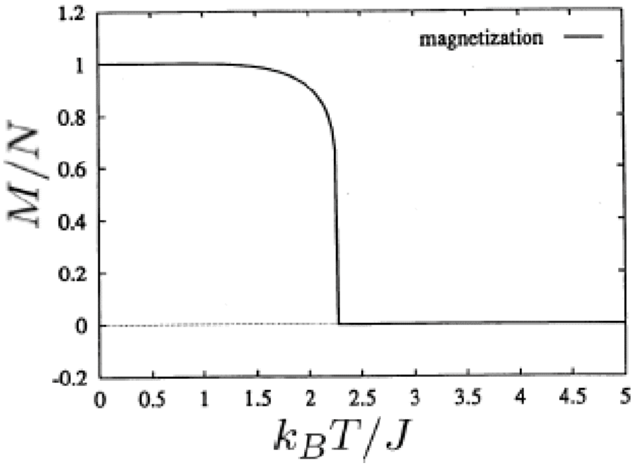


Figure 3.6: A plot of the average magnetisation M per spin for the pure two-dimensional Ising model on a square lattice as a function of the temperature T. Note that M is non-zero below a certain temperature and zero above this temperature, indicating a phase transition from a ferromagnetic to a paramagnetic phase with the magnetisation as the order parameter. The critical temperature at which this phase transition occurs is analytically known to be $T_c = 2J/\log\left(1+\sqrt{2}\right) \approx 2.269J$ where J is the coupling constant from the Hamiltonian [12]. Image taken from: `https://www.semanticscholar.org/paper/MATH-505-Project-%3A-Ising-model-%E2%80%93-Phase-transition-Wu-Li/de75b1ed085ada344a8fe3e28307bbe52ac73b83` (accessed 02-07-2021).

3.2 The bond-diluted Ising model

Now that we are familiar with the pure Ising model we can introduce the model that was studied in this thesis, the bond-diluted Ising model. Introducing bond-dilution to the model is actually pretty straightforward. If we consider the presence of the interaction between two nearest neighbours as a bond in the lattice, then we can simply go from the pure Ising model to the bond-diluted Ising model by randomly removing a fraction $1-p$ of the bonds from the lattice [5]. Here we have defined p to be the bond concentration (i.e. the fraction of bonds that is still present in the model). This definition of p also gives us a simple way of defining the Hamiltonian. The process of removing bonds is illustrated in figure 3.7.

We will now turn our attention to defining the Hamiltonian. The choice of the letter p for the bond concentration is no coincidence. It turns out that p can be interpreted as the probability that there is a bond between two nearest neighbours [5]. Consequently, if we choose $p = 1$ we retrieve the pure Ising model while $p = 0$ corresponds to a collection of non-interacting spins. The interactions in the bond-diluted Ising model are exactly the same as in the pure Ising model. The only difference is that a spin only interacts with one of its nearest neighbours if there is a bond between them [5]. With this in mind, we can write down the Hamiltonian as [5],

$$\mathcal{H} = -J\sum_{\langle ij\rangle} c_{ij}(p)s_is_j \tag{3.3}$$

where the sum runs over all pairs of nearest-neighbour sites, $s_i = \pm 1$ is the spin on site i and $c_{ij}(p)$ is a constant that follows a Bernoulli distribution with probability p, i.e. it has value 1 with probability p and value 0 with probability $1-p$. We will refer to an instantiation of the c_{ij}'s for all nearest-neighbour pairs as a randomness configuration of the model. The bond-dilution is frozen in for a particular configuration, in other words the values of the c_{ij}'s are fixed for a specific configuration. We can also compute the magnetisation of a state for the bond-diluted Ising model and it is defined in the exact same way as for the pure Ising model. All through the thesis, energy is measured in units of J.

The introduction of bond-dilution to the Ising model makes it possible to study the effect of impurities

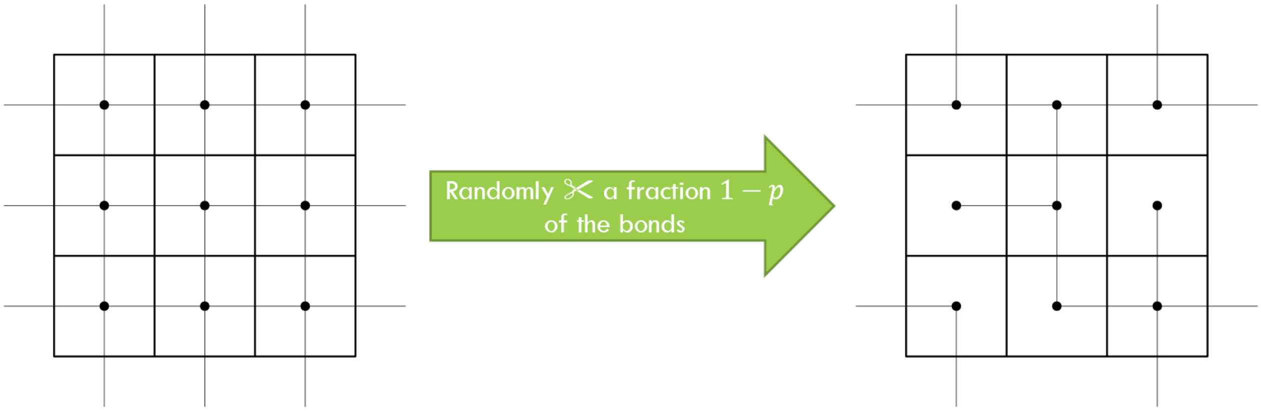


Figure 3.7: An illustration to explain how to go from the pure Ising model to the bond-diluted Ising model. If we consider the interaction between two nearest neighbours as a bond in the lattice, then all we have to do to get the bond-diluted Ising model is to simply remove a random fraction $1-p$ of the bonds from the lattice where p is the bond concentration, i.e. the fraction of bonds that remain in the lattice.

on the properties of a system. One such effect is that bond-dilution seems to change the critical temperature T_c of the ferromagnetic to paramagnetic phase transition. It has been shown that T_c gets smaller when lowering the bond concentration p and eventually drops to zero for $p_c = 0.5$ which is referred to as the percolation threshold [5]. This temperature dependence of T_c is plotted in figure 3.8. Interestingly enough, this introduces a second phase transition to the model, one where the critical temperature T_c of the other phase transition is the order parameter, i.e. it is zero below a certain value of p and non-zero above it. As was mentioned in the introduction, the bond-diluted Ising model also has other new properties but this is beyond the scope of this thesis.

4 Three algorithms

Now that the model has been introduced, let us give some examples of Monte Carlo algorithms for the bond-diluted Ising model that were used in our simulations. We will start with the Metropolis algorithm, which is one of the simplest algorithms that exists for the Ising model. It is a single spin flip algorithm, meaning that at each step we only consider a single spin to be flipped. A single step proceeds as follows [12],

1. choose a spin at random from the lattice,
2. calculate the change in energy ΔE for flipping the spin,
3. flip the spin with probability $e^{-\beta \Delta E}$ if $\Delta E > 0$ or with probability 1 otherwise.

These steps are repeated until enough samples have been generated. Another term frequently used in conjunction with the Metropolis algorithm is a sweep which is equal to N Metropolis flips where N is the number of spins in the lattice (L^2 in two dimensions) [12]. Note the occurrence of a Boltzmann factor-like term in the probability for flipping the spin. It is straightforward to show that by choosing the probability in this way we satisfy detailed balance and indeed generate states such that the probability of finding a state is proportional to its Boltzmann factor (just use that the transition probability from the high energy to the low energy state is 1 while it is $e^{-\beta \Delta E}$ in the opposite direction) [12]. Since the probability of flipping a spin is always non-zero it is hopefully also clear that we can reach any state from any state just by flipping enough spins which ensures we also satisfy ergodicity [12].

The simplicity of the Metropolis algorithm makes it easy to implement and implementations are usually also very fast (in terms of CPU time used per sweep). However, it is not without its difficulties. Near the critical temperature of the Ising model we often find large domains of parallel spins in the lattice [12]. It is difficult for Metropolis to flip a spin in the bulk of such a domain because this would increase the energy

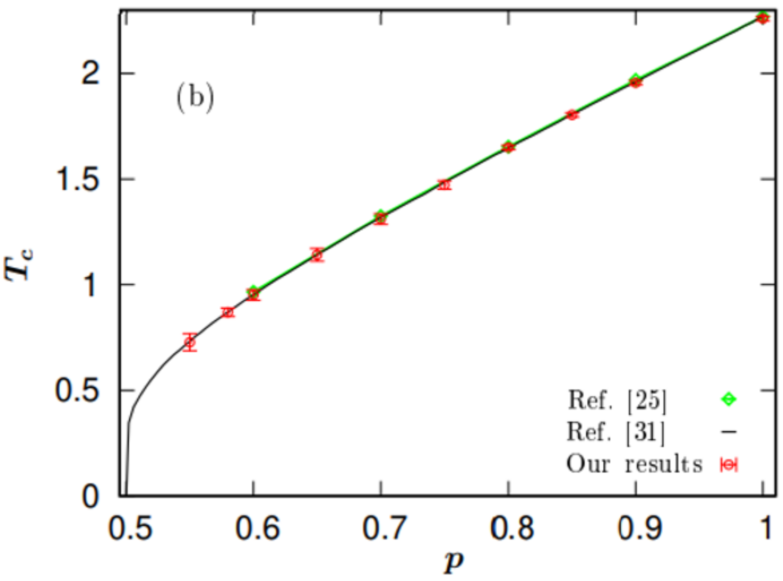


Figure 3.8: A plot of the critical temperature T_c of the ferromagnetic to paramagnetic phase transition of the bond-diluted Ising model as a function of the bond concentration p. We see that the introduction of bond-dilution lowers the critical temperature until it becomes zero at $p_c = 0.5$ which we call the percolation threshold [5]. Note that this in fact introduces a second phase transition to the model, one where the critical temperature T_c of the other phase transition is the order parameter, i.e. it is zero below a certain value of p and non-zero above it. Image taken from [5].

by a large amount (i.e. ΔE is relatively large and positive such that the probability for the flip is small). This means that the primary way for the Metropolis algorithm to change these domains is by flipping spins at the boundaries which is a slow process. Therefore, it takes a long time to reach a significantly different state. This problem is illustrated in figure 4.1. To get around this problem algorithms were proposed that flip entire clusters of spins at once which we will refer to as cluster algorithms.

The first example of these is the Wolff algorithm. The basic idea behind this algorithm is to grow a cluster of spins and flip all the spins in this cluster simultaneously with probability 1. To grow a cluster we perform the following steps [12],

1. choose a spin at random from the lattice,
2. consider each of its neighbours. If the spins are aligned, add the neightbour to the cluster with probability $1 - e^{-2\beta J}$ with $\beta = \frac{1}{k_B T}$ and J the coupling constant from the Hamiltonian,
3. for each of the neighbours added in step 2 also consider all their neighbours to be added to the cluster and repeat this until no more neighbours exist that have not yet been considered.

It can be shown that by growing the cluster in this way we satisfy both ergodicity and detailed balance [12]. It is important to note here that in the bond-diluted Ising model two spins are only considered to be neighbours if there is a bond between them. A single cluster move of the Wolff algorithm is visualised in figure 4.2.

The second cluster algorithm under consideration is the Swendsen-Wang algorithm. Similar to the Wolff algorithm, clusters of spins are grown according to the aforementioned procedure. It differs, however, in the fact that we do not just grow a single cluster, but cover the entire lattice with clusters and flip each of these with probability $\frac{1}{2}$ in a single step [12]. Since clusters are grown in the same way as in the Wolff algorithm,

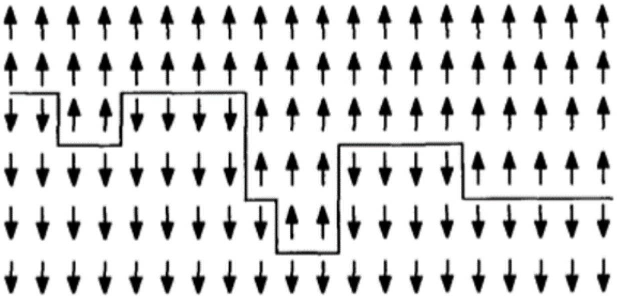


Figure 4.1: Illustration of the difficulties that the Metropolis algorithm faces near the critical point. Around the critical temperature the lattice often consists of relatively large domains of parallel spins. The primary way for the Metropolis algorithm to change these domains is by flipping spins at the boundaries which is a slow process. Therefore, it takes a long time to reach a significantly different state. Image taken from: `https://www.chegg.com/homework-help/questions-and-answers/page-mentioned-approximation-way-evaluate-2d-ising-model-even-though-exact-solution-gives--q45099410` (accessed 02-07-2021).

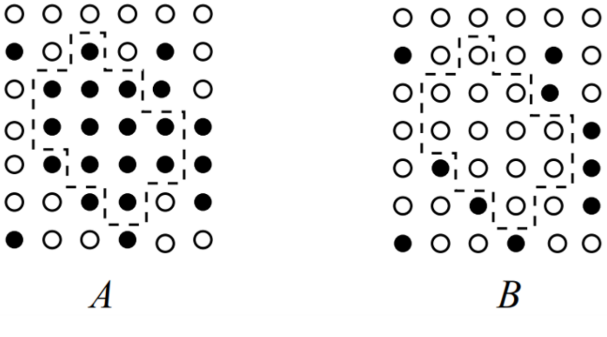


Figure 4.2: Visualisation of a single cluster move in the Wolff algorithm. We go from state A to state B in a single step. Note that the Wolff cluster that has been grown is indicated by the spins enclosed in the dashed line and that all spins in the cluster are flipped in this cluster move. Image taken from [16].

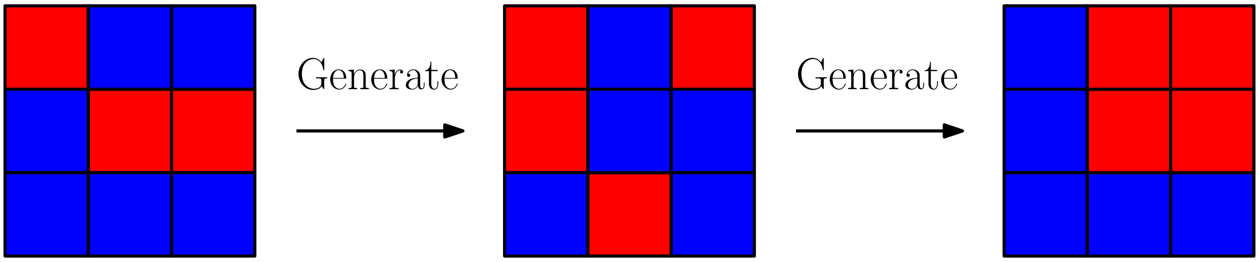


Figure 5.1: Illustration of how a Markov Chain Monte Carlo algorithm produces a chain of states.

showing that the Swendsen-Wang algorithm satisfies ergodicity and detailed balance proceeds analogously [12].

5 Characterising efficiency

Before we can compare the efficiency of the algorithms mentioned in the previous section, we first need to clarify what we mean when we are talking about the efficiency of a Monte Carlo algorithm. As we mentioned earlier, our Monte Carlo algorithms work via a Markov process that generates a new state from an existing state. When we repeat this generation step we end up with a chain of states as is illustrated in figure 5.1. Because of this, these algorithms are also known as Markov Chain Monte Carlo algorithms [12]. Ultimately, we are interested in generating a set of samples that are uncorrelated [12]. However, it is hopefully intuitive that if you generate a state from another state then these two states will probably be correlated to some degree. Therefore, we could ask ourselves the following question, if we start in state μ how many times do we need to repeat the generating step to end up in a state ν that is uncorrelated to μ? If we measure our time in Monte Carlo steps, then the number of steps that we need for this defines a timescale which is called the correlation time τ [12]. This correlation time is exactly how we will characterise the efficiency of our algorithms. More efficient algorithms have shorter correlation times.

Now that we have established that we can characterise the efficiency of our algorithms with their correlation times, we will discuss how we can quantify it. First, we need some way of quantifying correlations. A common way to do this for two random variables X and Y is by calculating their covariance which is defined by,

$$\mathrm{cov}(X,Y) = \langle (X - \langle X \rangle)(Y - \langle Y \rangle) \rangle . \tag{5.1}$$

When computing $(X - \langle X \rangle)(Y - \langle Y \rangle)$ we have two possibilities, (i) X and Y are on the same side of their average (both larger or both smaller) which gives us a positive number, (ii) X and Y are on opposite sides of their average which gives us a negative number. The intuition behind the covariance formula is that when X and Y are positively correlated, we expect that that possibility (i) will occur most frequently so that the covariance will yield a positive number, while if X and Y are negatively correlated possibility (ii) will occur most frequently and the covariance yields a negative number. If, however, X and Y are not correlated then possibility (i) and (ii) will occur the same number of times and the covariance will yield zero. So we see that the covariance is indeed a good measure of correlation between two random variables.

The random variables that we are interested in for computing the correlation time are the states of the system separated by some time difference Δt in our simulation. But since the state itself is not something which has a numerical value, we will instead use one of the quantities that we can compute for a state, i.e. the magnetisation for an Ising model state. So our random variables would be the magnetisation at two different time steps, $M(t)$ and $M(t+\Delta t)$. This allows us to define a useful quantity called the time-displaced autocorrelation of the magnetisation $\chi(\Delta t)$ as [12],

$$\chi(\Delta t) = \int dt\, [m(t) - \langle m \rangle]\, [m(t+\Delta t) - \langle m \rangle] \tag{5.2}$$

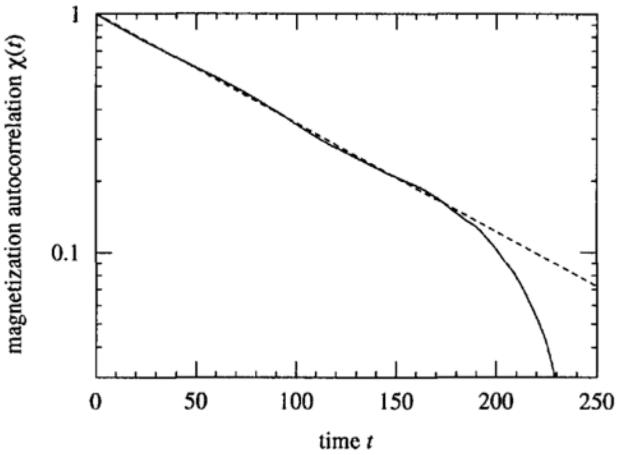


Figure 5.2: Plot of the magnetic autocorrelation $\chi(t)$ versus time t for the Metropolis algorithm and the pure two-dimensional Ising model on a square lattice with time measured in Metropolis sweeps. Note that for small t the autocorrelation seems to decay exponentially, which allows us to define a correlation time τ by assuming $\chi(t) \sim e^{-t/\tau}$. τ could be determined by fitting a line of that form through the data. For this example this would give $\tau = 95(5)$. Image taken from [12].

where $m = M/N$ is the magnetisation per site. Notice how this has the same structure as the covariance as we intended. An example of what this function could like for a simulation of the pure two-dimensional Ising model with the Metropolis algorithm is given in figure 5.2. Notice how it starts out as a straight line in a plot with a logarithmic y-axis. This suggests that the correlations decay exponentially at first. If we know propose that $\chi(t) \sim e^{-t/\tau}$ then this gives us a way of quantitatively defining the correlation time τ [12]. It also proposes a way of calculating τ, by fitting a line of this form through the autocorrelation data. For example, for figure 5.2 this would give $\tau = 95(5)$ with time measured in Metropolis sweeps [12] for this specific data set. Unfortunately, the story is not always this simple and sometimes the decay will be more complicated. Still, this gives us a reasonable way of quantifying τ and as we will now show we actually do not need to measure τ explicitly to compare the efficiency of algorithms.

While we do characterise the efficiency of our algorithms through the correlation time τ, we will not use it directly to compare their efficiency. The reason behind this, is that the correlation time is not a constant for a specific algorithm but also depends on many other properties of our system. For example, it is temperature dependent as is illustrated in figure 5.3 for the correlation time of the Metropolis algorithm. Notice the sharp peak around what appears to be the critical temperature T_c of the pure Ising model. It turns out that correlation times are often peaked around critical points which is commonly referred to as critical slowing down [12]. Moreover, the correlation time often depends on the system size L [12]. The nice thing about the scaling of the correlation time with system size is that it has an explicit form near the critical temperature. In that regime $\tau \sim L^z$ where the exponent z is called the critical dynamical exponent [12]. This dynamical exponent is, in fact, a constant for a specific algorithm and for this reason we will use z as the measure of efficiency that we will compare our algorithms with [12]. A larger z means that the correlation time increases more rapidly with system size, so if we have two algorithms with exponents z_1 and z_2 and $z_1 > z_2$ then the correlation times for algorithm 1 will always become larger than those for algorithm 2 if we make the system large enough. This should convince the reader that z is a good measure to compare algorithms, the smaller z the more efficient the algorithm. As an example, let us give the dynamical exponents z_m, z_w and z_{sw} for the Metropolis, Wolff and Swendsen-Wang algorithms respectively as they have been determined for the two-dimensional pure Ising model, $z_m = 2.167(1)$, $z_w = 0.25(1)$ and $z_{\text{sw}} = 0.25(1)$ [12]. This confirms that the cluster algorithms are indeed more efficient than Metropolis near the critical temperature as was intended.

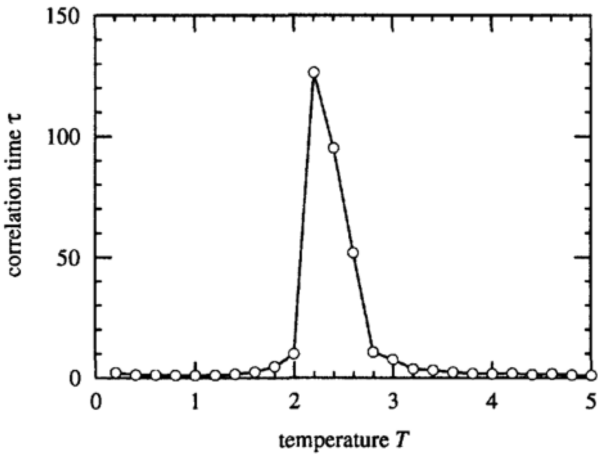


Figure 5.3: Plot of the correlation time τ of the Metropolis algorithm for the pure two-dimensional Ising model on a square lattice as a function of temperature T. Note the sharp peak around a certain temperature. This is no coincidence as the peak is located near the critical temperature T_c of the ferromagnetic to paramagnetic phase transition of the model. It is often the case that correlation times of Monte Carlo algorithms peak around critical points [12]. Image taken from [12].

Before we end this section we will take a short detour and return to the issue of thermalisation. If we study how the probability of finding a state ν in our Markov chain develops during our simulation, we will denote the probability at a certain time t by $w_\nu(t)$, then we see that the probability one time step later is related to the probabilities at this time step via [12],

$$w_\nu(t+1) = \sum_\mu P(\mu \to \nu) w_\mu(t). \tag{5.3}$$

In this notation $t = 0$ corresponds to the start of our simulation (so when we start the thermalisation process). If we collect all the probabilities in a vector $\mathbf{w}(t)$ and all the transition probabilities in a matrix $\mathbf{P}$, then we can concisely write this relation for all probabilities as a matrix vector equation [12],

$$\mathbf{w}(t+1) = \mathbf{P} \cdot \mathbf{w}(t). \tag{5.4}$$

Now if we write $\mathbf{w}(0)$ as a linear combination of the right eigenvectors $\mathbf{v}_i$ of $\mathbf{P}$ then we can use equation 5.4 to write $\mathbf{w}(t)$ as [12],

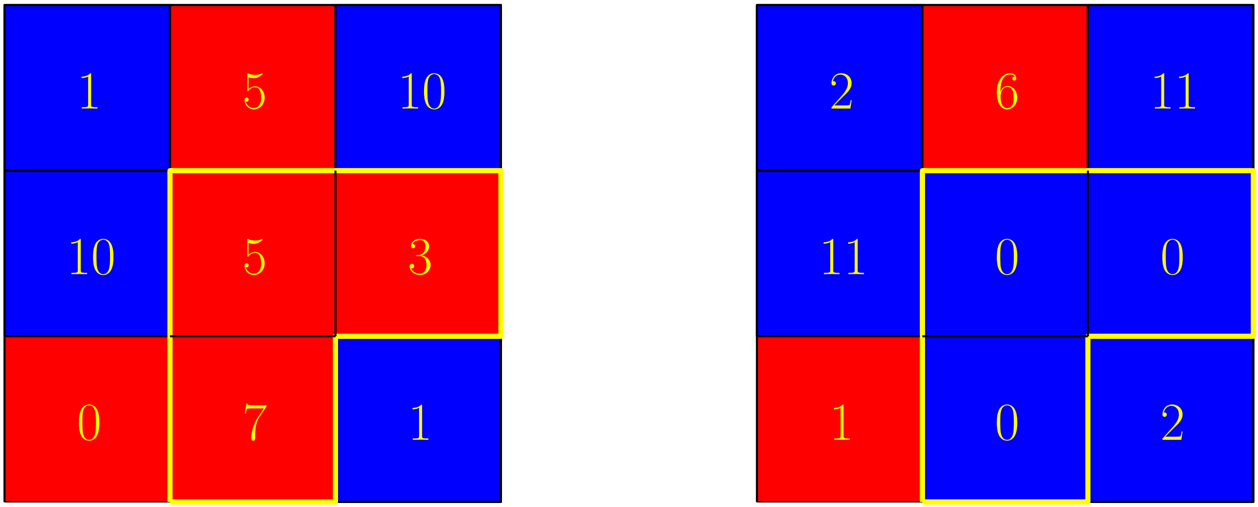
$$\mathbf{w}(t) = \mathbf{P}^t \cdot \sum_i a_i \mathbf{v}_i = \sum_i a_i \lambda_i^t \mathbf{v}_i = a_0 \lambda_0^t \mathbf{v}_0 + \sum_{i \neq 0} a_i e^{-t/\tau_i} \mathbf{v}_i \tag{5.5}$$

where a_i are coefficients that depend on the configuration of our system at $t = 0$, λ_i is the eigenvalue corresponding to eigenvector $\mathbf{v}_i$ (which we have ordered from largest eigenvalue to smallest eigenvalue, so λ_0 is the largest eigenvalue) and we defined a set of quantities τ_i for all $i \neq 0$ as [12],

$$\tau_i = -\frac{1}{\log \lambda_i}. \tag{5.6}$$

The reason that we use the symbol τ_i for these quantities is no coincidence. They are directly related to the correlation time τ that we defined earlier in this section [12]. From this form it becomes explicit that $\mathbf{w}(t)$ will approach $\mathbf{v}_0$, the eigenvector corresponding to the largest eigenvalue λ_0, for large times t (if t becomes large enough λ_0^t will be much bigger than all other λ_i^t). It turns out that $\mathbf{v}_0$ corresponds to the Boltzmann distribution if our transition probabilities satisfy ergodicity and detailed balance and this confirms that $\mathbf{w}(t)$ indeed tends to the Boltzmann distribution for large times [12].

We can learn two more things from this. Firstly, we see that there is in fact not a single correlation time



(a) State A

(b) State B

Figure 6.1: Illustration to explain how the age labels are updated after a single Wolff cluster move. This cluster moves takes us from state A to state B. The numbers in each square represent the age label of the corresponding spin for that state. The Wolff cluster that was grown in this cluster move is indicated by the yellow line which encloses the cluster of spins. Note that the age labels of the spins inside the cluster are set to 0 after this move while all other age labels are incremented by 1.

but actually as many as there are states in the system [12]. However, for sufficiently long simulations the largest of these correlation times, τ_1, will significantly dominate the dynamics so that we can still compare different algorithms with a reasonable degree of accuracy by assuming that the system only has a single correlation time τ [12]. Moreover, the correlation time also pops up when discussing the dynamics of thermalisation. Therefore, the timescale on which our system converges to its equilibrium during thermalisation is also dictated by the correlation time.

6 Observables and Methods

6.1 Observables

During our simulations we keep track of several quantities. This includes the energy E and magnetisation M of a state, which follow directly from the definition of the model and require no further explanation. In addition, we measured a self-defined quantity which we will refer to as the spin age and which we will define below.

To extract more information about the dynamics of the Wolff algorithm from our simulations we label each site in the lattice with a spin age a_i, which we define to be the time since site i was last visited (i.e. was part of a Wolff cluster) measured in the number of Wolff cluster moves. In other words, when a site is visited, its age is set to 0 and each subsequent Wolff cluster move where the site is not visited the age is incremented by 1, this is illustrated in figure 6.1. Once the system is thermalised, both with respect to its configuration of spins and the distribution of ages, we then count how often a certain age occurs at various steps in the simulation, to produce a histogram showing the distribution of ages in equilibrium. To be specific, at certain steps in the simulation (between moves) we measure for each age a how many spins in the lattice are labelled with that age at that step and we call this number the age frequency $f_L(a)$.

6.2 Finite size scaling

Ultimately, we want to determine the dynamical exponent for each algorithm to compare their efficiency for the bond-diluted Ising model. A straightforward way to do this would be to calculate the correlation time τ

from our data (i.e. by calculating autocorrelation functions) for different system sizes L and then fit a line $\sim L^z$ through these points to determine z. However, it is not always easy to accurately determine τ because the decay of the correlation might not be exponential for the entire range of your measurements. This would introduce additional errors because one would need to estimate where the exponential decay stops and which data is used to determine τ. Moreover, as we will see later the decay might not be exponential at all, but rather something more complicated such as stretched exponential. Therefore, we used a different technique to determine τ called finite size scaling which we will explain below.

We will explain the method by studying how a quantity called the equilibrium mean-square displacement of the magnetisation evolves with the single spin flip dynamics of the Metropolis algorithm. Our discussion here follows the discussion on the same topic from the paper by Walter and Barkema [16]. We will define the equilibrium mean-square displacement of the magnetisation $h(t)$ as

$$h(t) = \left\langle [M(t) - M(0)]^2 \right\rangle \tag{6.1}$$

where $M(t)$ is the magnetisation at time t during our simulation with t measured in Metropolis sweeps. When we start at $t = 0$ how does $h(t)$ evolve when t increases. At first, the spin flips that Metropolis performs will be uncorrelated. Moreover, the number of flips performed on our two-dimensional square lattice scales as $L^2 t$. We therefore expect that for small times ($t < 1$) $h(t)$ will behave diffusively, i.e.

$$h(t) \sim L^2 t. \tag{6.2}$$

On the other hand, for large times ($t > \tau$) we expect that $M(t)$ and $M(0)$ are no longer correlated such that $\langle M(t) \cdot M(0) \rangle \approx 0$. So to see how $h(t)$ behaves at these large times we can expand the parentheses,

$$\begin{aligned} h(t) &= \left\langle [M(t) - M(0)]^2 \right\rangle \\ &= \left\langle M(t)^2 + M(0)^2 - 2M(t) \cdot M(0) \right\rangle \\ &\approx 2 \left\langle M^2 \right\rangle . \end{aligned}$$

Now if we recall the definition of the magnetic susceptibility

$$\chi = \beta / N \left(\left\langle M^2 \right\rangle - \langle M \rangle^2 \right) \tag{6.3}$$

and use that $\langle M \rangle \approx 0$ at $T \geq T_c$ combined with the fact that $\chi \sim L^{\gamma/\nu}$ at the critical temperature and that $N = L^2$ in two dimensions, we find that for large times $h(t) \sim L^{2+\gamma/\nu}$. So far, we have learned that $h(t)$ behaves as $\sim L^2 t$ for $t < 1$ and as $\sim L^{2+\gamma/\nu}$ for $t > \tau$. Furthermore, we know that $\tau \sim L^z$ at the critical point so $h(t)$ has to grow from $h(t \approx 1) \sim L^2$ to $h(t \approx \tau \sim L^z) \sim L^{2+\gamma/\nu}$. If we now assume that $h(t)$ exhibits power law behaviour we can conclude that at intermediate times $h(t) \sim t^{\gamma/(\nu z)}$. Putting it all together we can write down the following form for $h(t)$,

$$h(t) \sim L^{2+\gamma/\nu} \mathcal{F}(t/L^z) \tag{6.4}$$

where $\mathcal{F}(x)$ is a scaling function that satisfies $\mathcal{F}(x) = \text{constant}$ for $x \gg 1$ and $\mathcal{F}(x) \sim x^{\gamma/(\nu z)}$ at intermediate times. Having derived this form for $h(t)$ we can now easily explain how finite size scaling works. Say that instead of plotting $h(t)$ vs t we were to plot $h(t)/L^{2+\gamma/\nu}$ vs t/L^z for different system sizes L we see that we are in fact plotting $\mathcal{F}(x)$ as a function of x for all values of L. Consequently, these curves should all fall on top of each other. The key is that this only happens if we use the correct values of γ/ν and z, so if we tune these values until the curves collapse we have a way of determining the values of these critical exponents.

The previous discussion was for the specific example of the equilibrium mean-square displacement of the magnetisation for the Metropolis algorithm. However, the approach is the same for different quantities and algorithms, we rewrite the quantity in such a way that all the L dependence becomes explicit. This way an L-dependent scaling of the axes allows us to collapse curves for different system sizes. In our simulations, this scaling will usually involve L^z so that we can determine the dynamical exponent z by tuning its value until the curves fall on top of each other. This approach has the advantage that we do not need to know anything about how the correlations decay, we just need to find the right scaling. Therefore, this is often easier to do while also giving more accurate results.

Lastly, we want to mention that the aforementioned discussion also suggests an alternative way of determining z that only requires us to simulate at a single system size L. We mentioned that for intermediate times $h(t) \sim t^{\gamma/(\nu z)}$. So assuming that we have some other way of calculating γ/ν we can get an estimate of z by fitting a line of this form through the $h(t)$ data of a single simulation.

7 Results and Discussion

7.1 The Wolff algorithm

We will first discuss the behaviour of the Wolff algorithm applied to the bond-diluted Ising model. To study the behaviour we ran simulations with the Wolff algorithm for various system sizes with $p = 0.6$ at $(\beta J)^{-1} = 0.940$ where $\beta = \frac{1}{k_B T}$ and J the coupling constant. We chose this value for p because the effects of bond-dilution become more pronounced when the bond fraction p is significantly below 1. The temperature was chosen to be in the vicinity of the critical temperature as determined with the Binder cumulant. The value we found is also in good agreement with the critical temperature found in other papers, see for example [5]. Unless otherwise mentioned we used $100,000$ different realizations of the bond dilution in each simulation. For all the results we measure time in cluster moves of the algorithm used, because we found this the most intuitive timescale for understanding the results. However, when evaluating the performance of an algorithm, we prefer to measure time such that it scales with required CPU time. Since the CPU time per single Wolff cluster move can vary significantly, we require also a second timescale for the Wolff algorithm. A good candidate is to measure time such that $t = 1$ corresponds with the situation where on average as many spins are flipped as there are in the lattice. It can be shown that the relation between this new time t and our previous time, which we will denote by t_{steps} for Wolff, is given by $t = t_{\text{steps}} \frac{\langle n \rangle}{L^2}$ where $\langle n \rangle$ is the average cluster size of a Wolff cluster [12]. It can also be shown that $\langle n \rangle$ scales as $L^{\gamma/\nu}$ at the critical temperature so that we can use $L^{\gamma/\nu-2}$ as a conversion factor when required [12]. By construction the same number of spins, all the spins in the lattice, are visited by the Swendsen-Wang algorithm in each cluster move, so t_{steps} already scales with CPU time for Swendsen-Wang and no additional timescale is required and we will just use t to denote the time measured in Swendsen-Wang cluster moves.

Figure 7.1 shows the evolution of the energy of the system towards its thermal equilibrium value as a function of Wolff cluster moves. For $L = 40$ we ran for 400 cluster moves per configuration, for $L = 100$ we ran for 300 cluster moves and in between we tuned the number of cluster moves to roughly keep the CPU time used per simulation constant. At $t_{\text{steps}} = 0$ the system starts in the configuration with all spins pointing up ($s_i = 1$ for all i). Notice how the curve seems to transition from a fast decay for small t_{steps} to a slower decay at large t_{steps}. When the vertical and horizontal axes are scaled with L^2 the right tails of the curves, the regions of slower decay, collapse. Since these right tails are the limiting factor in convergence of the energy to its equilibrium this suggests that the correlation time $\tau_{\text{steps},w}$ scales as L^2 such that τ_w scales as L^{z_w} with $z_w = \gamma/\nu$. Numerically, it is reported that γ/ν is independent of p for $p \geq 0.6$, and actually indistinguishable from $\gamma/\nu = 1.75$ as in the regular Ising model [8]. Note that, while the equilibrium exponents are numerically indistinguishable, the dynamic exponent is very different: in the regular 2D Ising model the dynamic exponent is reported as $z_w = 0.25(1)$ [12].

We will now show that the long tail in the convergence of the energy is caused by a few spins which survive for a very long time (i.e. they are not touched by the algorithm for many steps). Consequently, the correlation stored in these spins, however small it might be, will also survive for a long time. To demonstrate this we computed a histogram of the distribution of the spin ages throughout a simulation with the Wolff algorithm in the manner described in the Model and Methods section. For these simulations we used 10,000 realizations of the bond-dilution. To initialise the system we first thermalise with 50 Swendsen-Wang moves, starting from a state with all spins pointing up. We also first run the simulation for $5N$ Wolff cluster moves to make

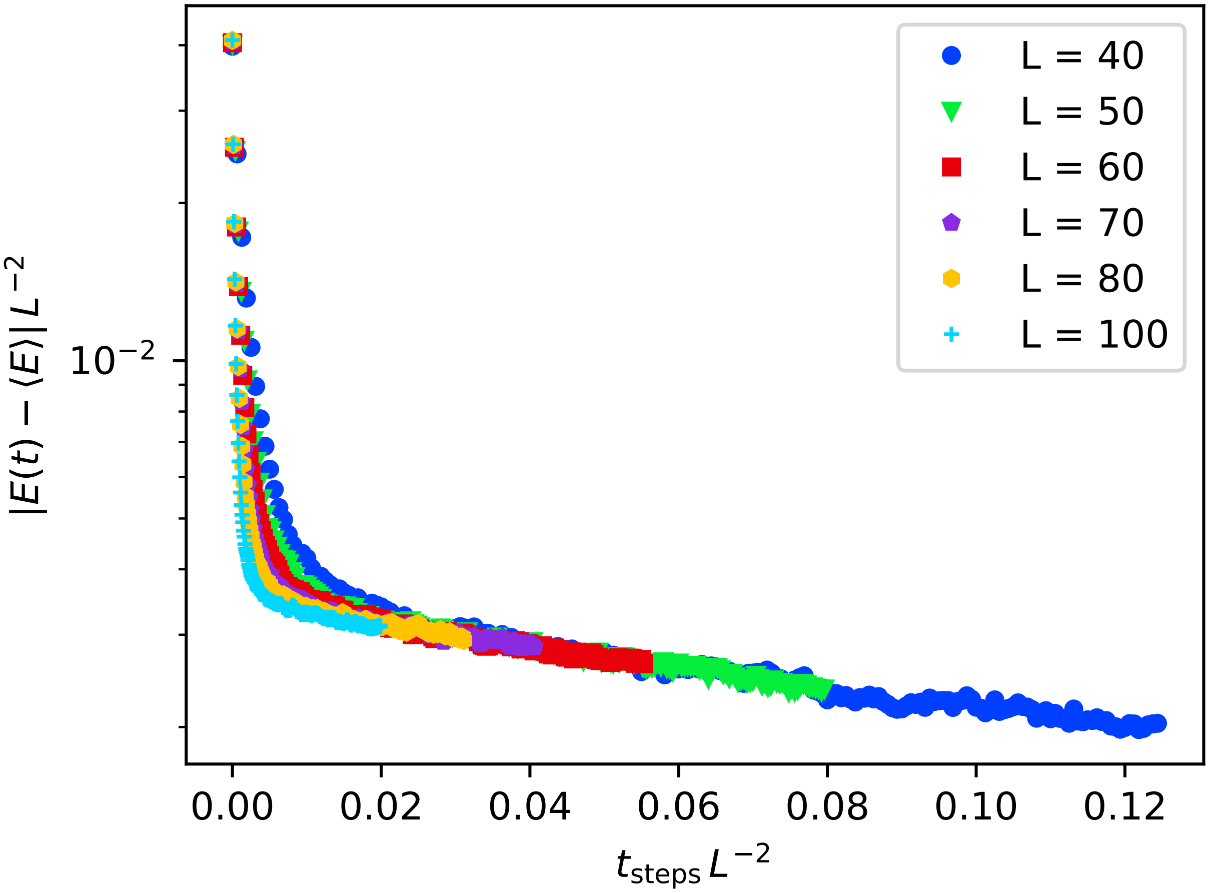


Figure 7.1: Convergence of the energy $E(t)$ to the thermal equilibrium $\langle E \rangle$ during thermalisation with the Wolff algorithm for different system sizes L with $p = 0.6$ at $(\beta J)^{-1} = 0.940$ where $\beta = \frac{1}{k_B T}$ and J the coupling constant. For $t_{\text{steps}} = 0$ the system starts in a state with all spins pointing up. Both the vertical and horizontal axes were scaled with L^2. Note the collapse of the right tails of the curves, suggesting that the correlation time $\tau_{\text{steps},w} \sim L^2$.

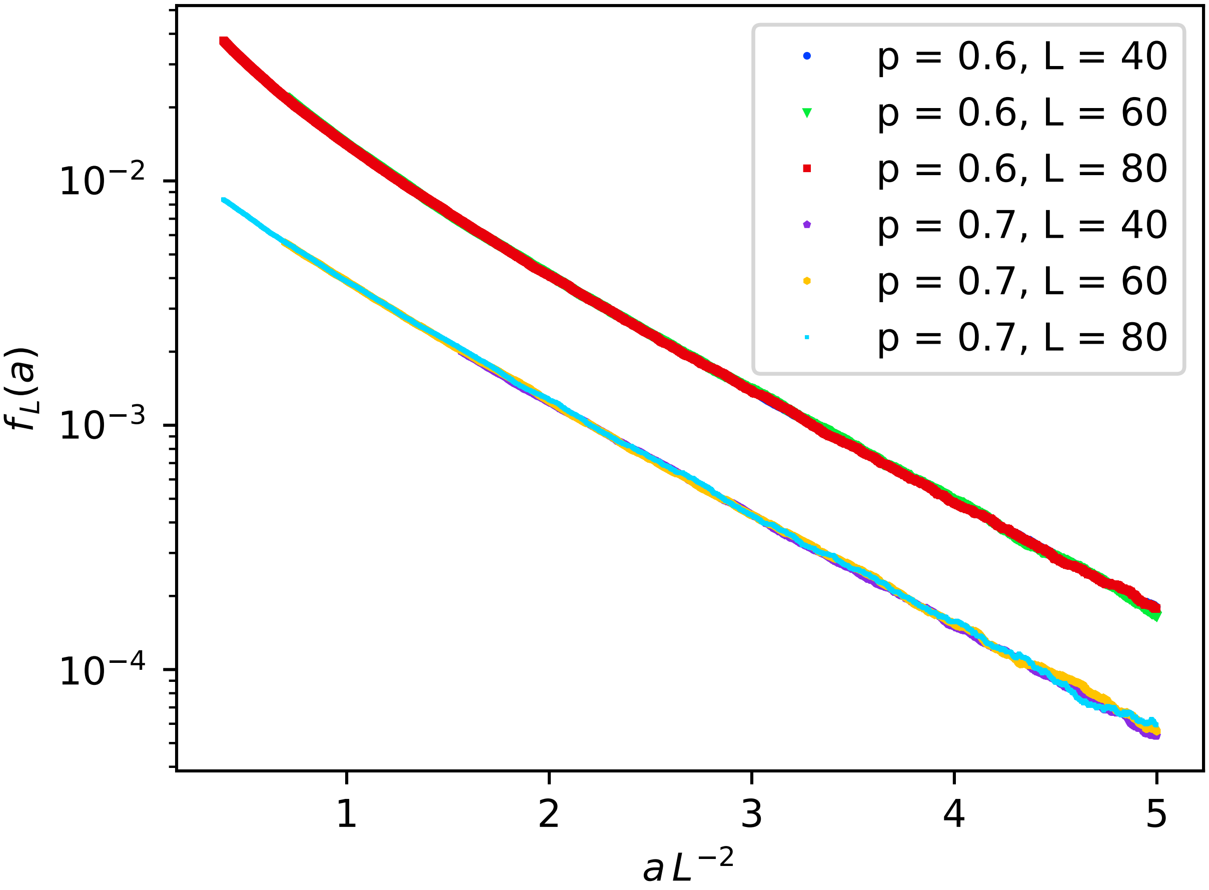


Figure 7.2: Distribution of spin ages a during a simulation with the Wolff algorithm at equilibrium with $p = 0.6$ at $(\beta J)^{-1} = 0.940$ and with $p = 0.7$ at $(\beta J)^{-1} = 1.310$ where $\beta = \frac{1}{k_B T}$ and J the coupling constant. The spin age is defined as the time since the site was last visited, measured in Wolff cluster moves. The horizontal axis was scaled with L^2. The collapse of the curves again suggests that the correlation time $\tau_{\text{steps},w}$ of the Wolff algorithm scales as L^2, in agreement with figure 7.1.

sure that spins can actually reach all the ages that we report in the histogram. Finally, we measure the age for an additional 1000 consecutive Wolff steps. We did the simulations for both $p = 0.6$ at $(\beta J)^{-1} = 0.940$ as before and for $p = 0.7$ at $(\beta J)^{-1} = 1.310$, which we found to be in the vicinity of the critical temperature at that bond fraction p, again in agreement with the critical temperature found in other papers [5]. The results are shown in figure 7.2.

The figure clearly shows that some spins survive for a very long time. Also note the strikingly good collapse of the curves when we scale the horizontal axis with L^2, both for $p = 0.6$ and $p = 0.7$. This supports our earlier finding that τ_w scales as L^{z_w} with $z_w = \gamma/\nu \approx 1.75$. It also seems to suggest that the scaling of τ_w is independent of the bond concentration p for $p < 1$. We believe that these long surviving spins are actually isolated spins in the lattice (i.e. spins which have all their bonds removed or small groups of spins which have their bonds to the rest of the lattice removed, this is illustrated in figure 7.3). Such spins would only be flipped by the Wolff algorithm if they are chosen as the seed spin. And since each spin is equally likely to be picked and there are L^2 spins, this would explain the L^2 limiting factor in the correlation time $\tau_{\text{steps},w}$.

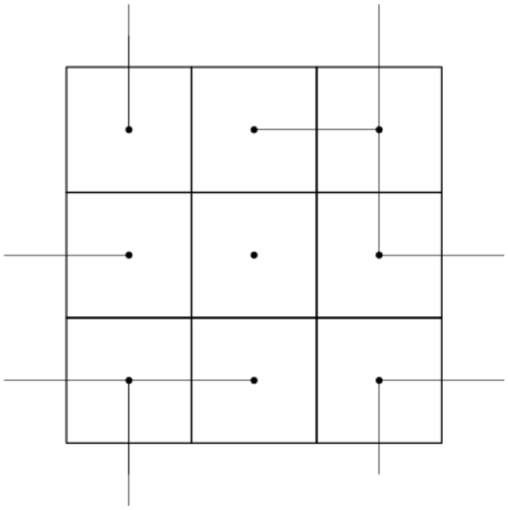


Figure 7.3: Illustration of an isolated spin which might be created in the bond-diluted Ising model if all bonds for a particular spin are removed. The isolated spin in question is the square in the center of the lattice.

7.2 The Swendsen-Wang algorithm

Now we will turn our attention to the Swendsen-Wang algorithm. By construction, it visits every spin in the lattice each step, so it should not suffer from the problems encountered with the Wolff algorithm, originating from long surviving spins. Similar to the Wolff algorithm we ran simulations for various system sizes L at $p = 0.6$ and $(\beta J)^{-1} = 0.940$. Figure 7.4 shows the analogue of figure 7.1 but then for Swendsen-Wang. In addition, it contains an inset figure that shows the same data but plotted in a different way. At $L = 30$ we ran for 300 Swendsen-Wang steps per configuration while at $L = 100$ we ran for 100 steps; in between we tuned the steps to keep the CPU time used roughly constant. In the main part of the figure we can see that the energy quickly converges to its thermal equilibrium value and the slowly decaying tail from figure 7.1 is absent. Moreover, when scaling the vertical axis with L^2 and the horizontal axis with $L^{z_{\text{sw}}}$ with $z_{\text{sw}} = 0.09(4)$ the curves collapse suggesting that the correlation time τ_{sw} for Swendsen-Wang at $p = 0.6$ scales as $L^{z_{\text{sw}}}$. Note that the dynamical exponent z_{sw} is significantly smaller at $p = 0.6$ than for the regular 2D Ising model ($p = 1$) where $z_{\text{sw}} = 0.25(1)$ [12]. This is the opposite of the super slowing down observed for the Metropolis algorithm [5]. Finally, in the inset figure the data for $h(t)$ versus time t is plotted. Here $h(t) = -\log\left(c\,|E(t) - \langle E\rangle|\right)$ with $c = |E(0) - \langle E\rangle|^{-1}$. The blue curve is a straight line with slope 0.87. Since the data seems to be parallel to this blue curve instead of a curve with slope 1, the convergence of the energy seems to be stretched exponential.

We already determined a value for the dynamical exponent z_{sw} from figure 7.4. However, this plot shows data from simulations out-of-equilibrium so to confirm the scaling of the correlation time we also determined it from equilibrium simulations. For this we computed the evolution of the mean-square displacement of the energy $\langle [E(t) - E(0)]^2\rangle$ from the same data as was used for figure 7.4. To obtain equilibrium data we discarded all data before the system was thermalised. For $L = 30$ this meant all data before $t = 50$ and for all other system sizes all data before $t = 20$ (i.e. these times became the new $t = 0$ for determining $\langle [E(t) - E(0)]^2\rangle$). The results are shown in figure 7.5. After scaling the vertical axis with the numerically determined limit values of the curves, we can collapse the curves using a horizontal scaling of $L^{z_{\text{sw}}}$ with $z_{\text{sw}} = 0.09(4)$. This confirms our earlier numerical estimate of the dynamical critical exponent for the Swendsen-Wang algorithm at $p = 0.6$.

The stretched exponential decay during thermalisation was unexpected, so to investigate whether this also occurs in the pure Ising model we repeated the simulations that were used to produce figure 7.4 for $p = 1$ and $(\beta J)^{-1} = 2.269$ with otherwise exact same conditions. At $p = 1$ there is only one possible randomness

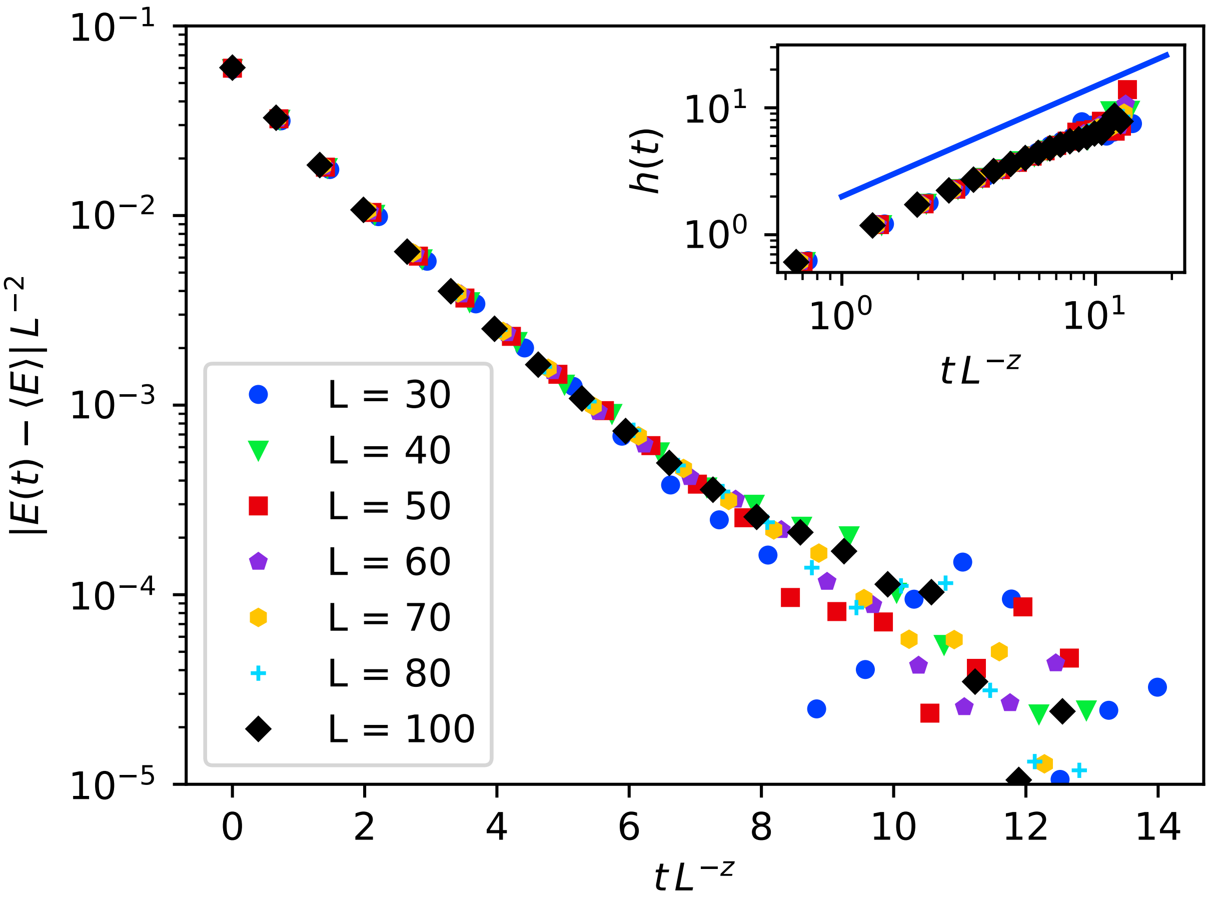


Figure 7.4: Convergence of the energy $E(t)$ to the thermal equilibrium $\langle E \rangle$ during thermalisation with the Swendsen-Wang algorithm for different system sizes L with $p = 0.6$ at $(\beta J)^{-1} = 0.940$ where $\beta = \frac{1}{k_B T}$ and J the coupling constant. For $t = 0$ the system starts in a state with all spins pointing up. The vertical axis was scaled with L^2 and the horizontal axis with $L^{z_{\mathrm{sw}}}$ with $z_{\mathrm{sw}} = 0.09(4)$. Note that this plot is equivalent to figure 7.1 but for the Swendsen-Wang algorithm. The collapse of the curves suggests that the correlation time for the Swendsen-Wang algorithm scales as $L^{z_{\mathrm{sw}}}$ with $z_{\mathrm{sw}} = 0.09(4)$. Also note the absence of a slowly decaying tail, demonstrating that the Swendsen-Wang algorithm does not suffer from the same problems that plague the Wolff algorithm (see figure 7.1). The inset figure in the top-right shows the same data but plotted differently. Here $h(t) = -\log\left(c\,|E(t) - \langle E \rangle|\right)$ with $c = |E(0) - \langle E \rangle|^{-1}$. The blue curve is a straight line with slope 0.87. Since the data seems to be parallel to this blue curve instead of a curve with slope 1, the convergence of the energy seems to be stretched exponential.

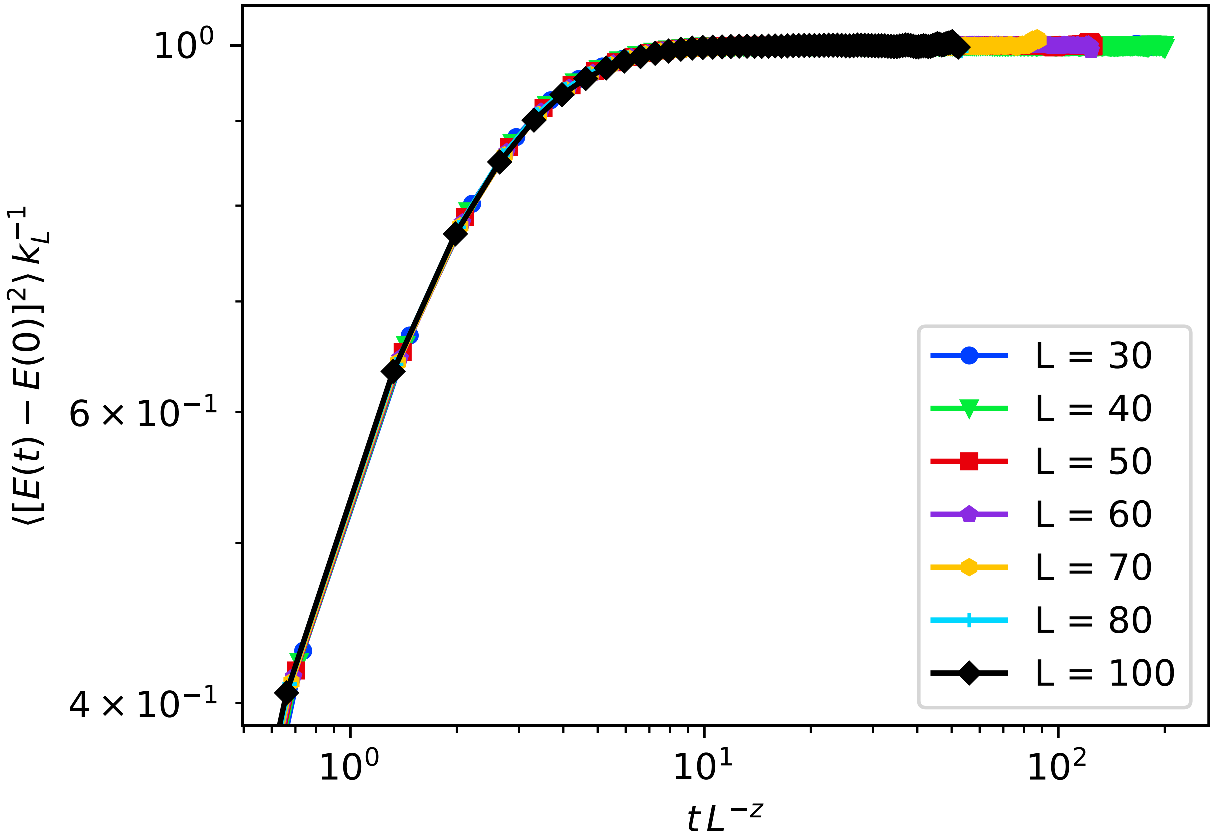


Figure 7.5: Mean-square displacement of the energy $\langle[E(t)-E(0)]^2\rangle$ in thermal equilibrium as a function of Swendsen-Wang moves t for different system sizes L with $p=0.6$ at $(\beta J)^{-1}=0.940$ where $\beta=\frac{1}{k_B T}$ and J the coupling constant. The vertical axis was scaled with the numerically determined limit value of the curves, while the horizontal axis was scaled with $L^{z_{\text{sw}}}$ with $z_{\text{sw}}=0.09(4)$. The collapse of the curves confirms the numerical value of the dynamical critical exponent for the Swendsen-Wang algorithm.

configuration (as all bonds are still there and there is no randomness), but we still averaged over $100,000$ independent simulations for better statistics. The results can be seen in figure 7.6. From the inset figure we can see that the decay is still stretched exponential for $p = 1$ but this time the slope of the curve is 0.81 so it is slightly different. We also see a nice collapse of the curves when using $z_{\text{sw}} = 0.25(1)$ which is consistent with the dynamical exponent from the literature as expected [12].

7.3 The Metropolis algorithm

For reference, we also studied the dynamics of the Metropolis algorithm. Since this has been studied in other papers, this allowed us to confirm that our methods are correct and to test a new way of determining the dynamical exponent for the Metropolis algorithm. We ran simulations with the Metropolis algorithm for various system sizes L at $p = 0.6$ and $(\beta J)^{-1} = 0.940$. Each simulation was first thermalised with 50 Swendsen-Wang steps. After thermalisation we ran each simulation for an additional 300 Metropolis sweeps. Since sweeps are directly proportional to CPU time we will denote time measured in sweeps simply with t and no other timescale is needed. Figure 7.7 shows the equilibrium mean-square displacement of the absolute magnetisation $\langle[|M(t)|-|M(0)|]^2\rangle$ as a function of sweeps t. Notice that the simulations were not long enough to reach the long timescale where the mean-square displacement saturates to a limit value, which made it impossible to use finite size scaling for determining z. This is an unfortunate side effect of the super slowing down observed for Metropolis that causes the correlation time to become very long [5].

To get around this problem we used the alternative technique for determining z that was mentioned at the end of section 6. Recall that at intermediate times the mean-square displacement behaves as $\sim t^{\gamma/(\nu z)}$ (technically this was for the magnetisation M not the absolute magnetisation $|M|$, but this should still hold in our case because the Ising model has an up-down symmetry so that the sign of the total magnetisation does not matter). So if we could fit a curve through our data at intermediate times this allows us to determine z. We decided to use the $L = 80$ data for the fit. Since it is not immediately clear from the data where the intermediate timescale begins we fit a curve through the data that transitions from t^α to t^β behaviour, namely a function $f(t)$ of the form $f(t) = (1-s(t))c_1t^\alpha + s(t)(c_2t^\beta + c_3)$ where $s(t) = 0.5 + 0.5\tanh((t-a)/b)$ is a switching function. This avoids the problem of having to guess where the $t^{\gamma/(\nu z)}$ behaviour begins, we can just fit through all the data and β gives us the exponent we are interested in. The result of the fit can be seen in figure 7.8. Combining the result of the fit with our assumption that γ/ν is approximately numerically independent of p and equal to its value in the pure Ising model, i.e. $\gamma/\nu \approx 1.75$ [8], we find that $z_m = 3.337(3)$ at $p = 0.6$. This is in good agreement with other results which confirms that this is also a good way of determining the dynamical exponent for the Metropolis algorithm [5]. This method might actually be preferable since simulations only need to be long enough to reach the intermediate timescale and simulations at a single system size L are already sufficient.

8 Conclusion and Outlook

We have shown how the correlation times τ_w and τ_{sw} of the Wolff and Swendsen-Wang cluster algorithms scale as a function of the system size L when applied to the two-dimensional bond-diluted Ising model. We demonstrated that the Wolff algorithm suffers from a much longer correlation time than in the pure Ising model, caused by isolated (groups of) spins which are infrequently visited by the algorithm. These cause the correlation time to scale as L^{z_w} where $z_w = \gamma/\nu \approx 1.75$ independent of the bond concentration p for $0.5 < p < 1$. Moreover, we have shown that the Swendsen-Wang algorithm does not suffer from the same problem, by construction. It has a much shorter correlation time, even shorter than in the pure Ising model. Numerically, we have found that its correlation time scales as $L^{z_{\text{sw}}}$ with $z_{\text{sw}} = 0.09(4)$ at $p = 0.6$. Lastly, we tested a novel way of determining the dynamical exponent for the Metropolis algorithm and confirmed that it worked properly. With this method we determined that $z_m = 3.337(3)$ at $p = 0.6$.

We expect that the Wolff algorithm will suffer from the same problems in the three-dimensional bond-diluted Ising model, albeit to a lesser degree as more bonds will have to be removed to create isolated spins. In addition, we think the same will hold for the site-diluted and weakly diluted (i.e. where you weaken instead of removing the bonds) Ising models. This could be something to explore in the future. Finally,

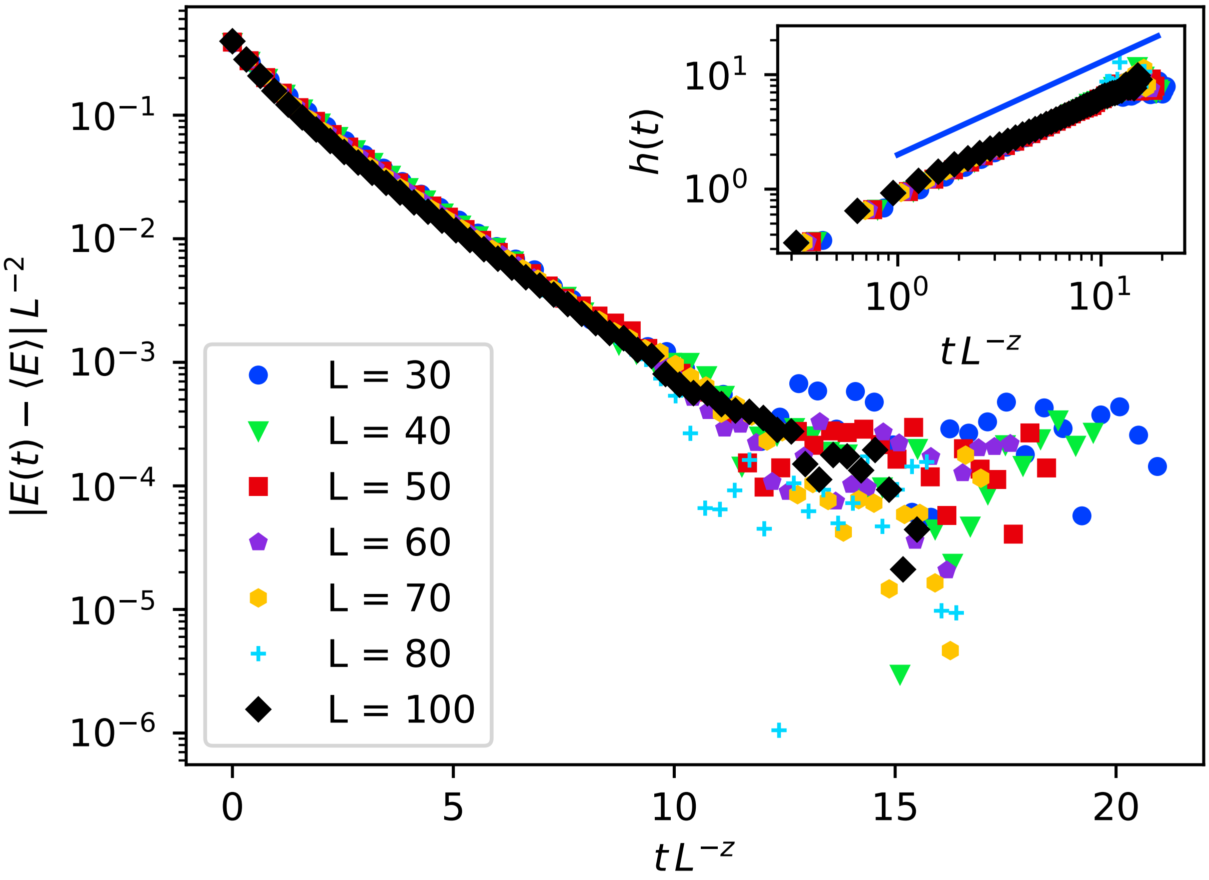


Figure 7.6: Convergence of the energy $E(t)$ to the thermal equilibrium $\langle E \rangle$ during thermalisation with the Swendsen-Wang algorithm for different system sizes L with $p = 1$ at $(\beta J)^{-1} = 2.269$ where $\beta = \frac{1}{k_B T}$ and J the coupling constant. For $t = 0$ the system starts in a state with all spins pointing up. The vertical axis was scaled with L^2 and the horizontal axis with $L^{z_{\mathrm{sw}}}$ with $z_{\mathrm{sw}} = 0.25(1)$. Note that this plot is equivalent to figure 7.4 but for $p = 1$. The collapse of the curves suggests that the correlation time for the Swendsen-Wang algorithm scales as $L^{z_{\mathrm{sw}}}$ with $z_{\mathrm{sw}} = 0.25(1)$ as was expected. The inset figure in the top-right shows the same data but plotted differently. Here $h(t) = -\log\left(c\,|E(t) - \langle E \rangle|\right)$ with $c = |E(0) - \langle E \rangle|^{-1}$. The blue curve is a straight line with slope 0.81. Since the data seems to be parallel to this blue curve instead of a curve with slope 1, the convergence of the energy seems to be stretched exponential.

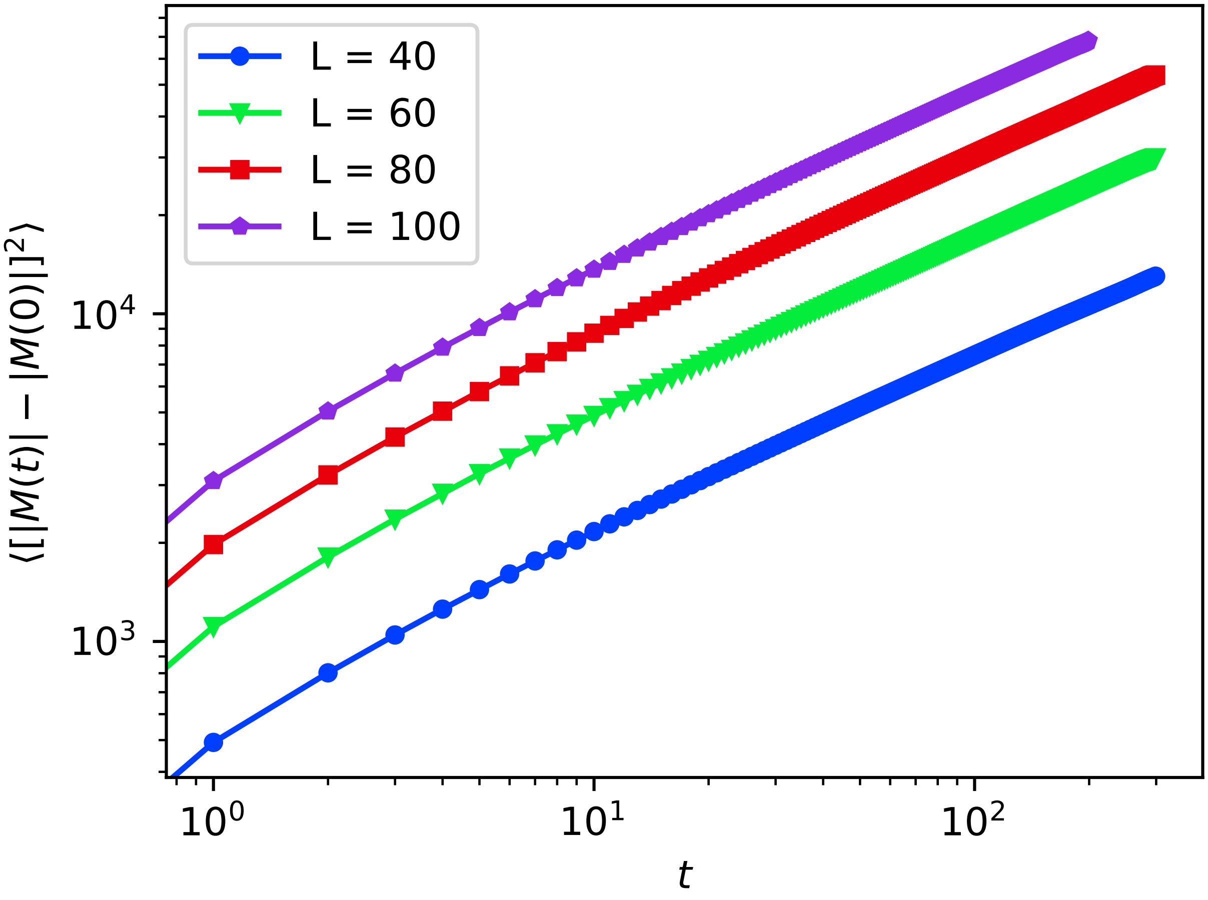


Figure 7.7: Mean-square displacement of the absolute magnetisation $\langle[|M(t)| - |M(0)|]^2\rangle$ in thermal equilibrium as a function of Metropolis sweeps t for different system sizes L with $p = 0.6$ at $(\beta J)^{-1} = 0.940$ where $\beta = \frac{1}{k_B T}$ and J the coupling constant.

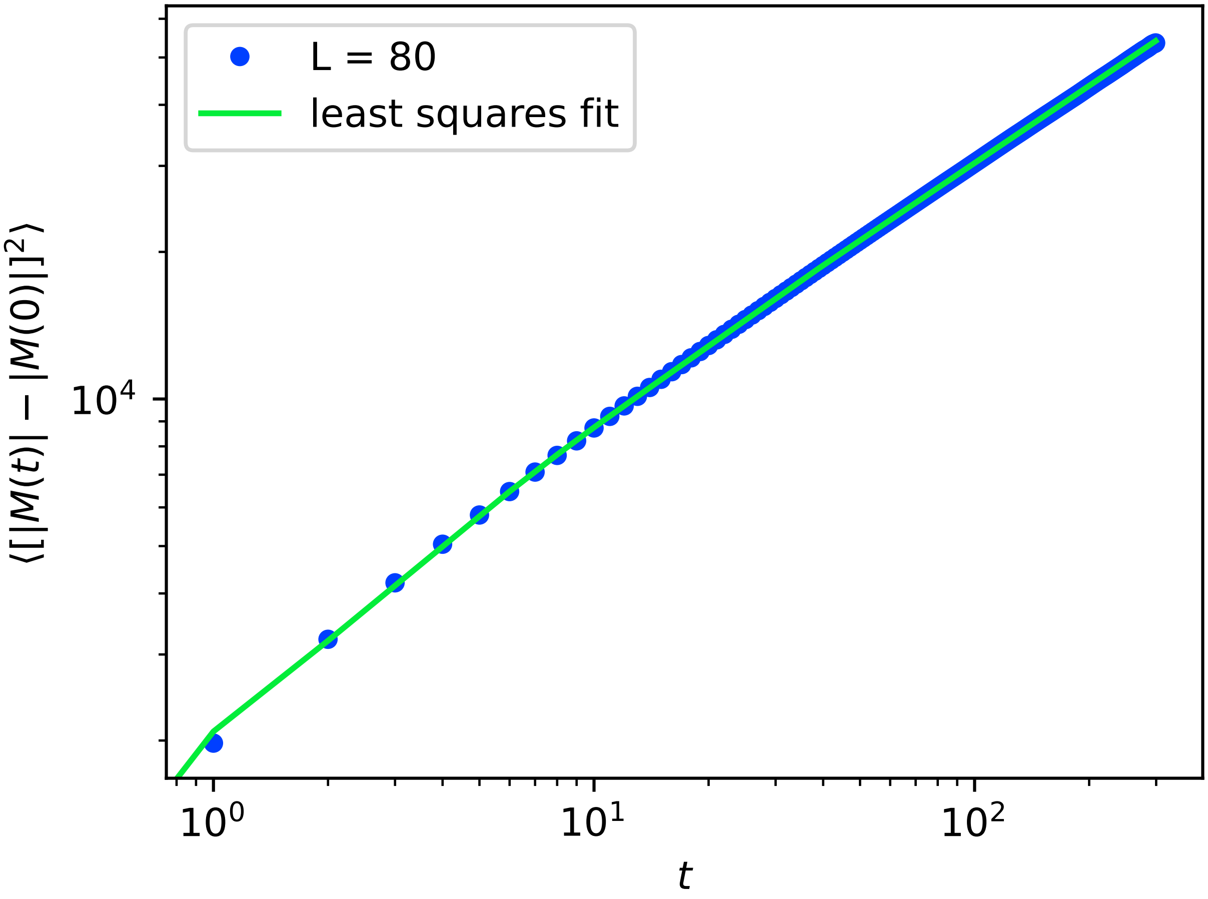


Figure 7.8: Mean-square displacement of the absolute magnetisation $\langle [|M(t)| - |M(0)|]^2 \rangle$ in thermal equilibrium as a function of Metropolis sweeps t for $L = 80$ with $p = 0.6$ at $(\beta J)^{-1} = 0.940$ where $\beta = \frac{1}{k_B T}$ and J the coupling constant. The curve through the points is a least squares fit to the data of the form $f(t) = (1 - s(t))c_1 t^\alpha + s(t)(c_2 t^\beta + c_3)$ where $s(t) = 0.5 + 0.5 \tanh((t - a)/b)$ is a switching function. $f(t)$ models a curve that transitions from t^α to t^β behaviour. Ultimately, we were interested in the exponent β since z could be determined from that via $\beta = \gamma/(\nu z)$. From the fit we find that $z_m = 3.337(3)$.

we stated that z_w is independent of the bond concentration for $0.5 < p < 1$. We believe that this is the case because the limiting factor in the correlation time is the presence of isolated spins and there is always a possibility of isolated spins for $0.5 < p < 1$. However, as we start approaching $p = 1$ the number of isolated spins in a single randomness configuration will drop significantly until at some point the average expected number of isolated spins drops below 1. Technically, we will still encounter isolated spins in this regime if we make our systems large enough or sample enough randomness configurations and so the correlation problem will still exist to some degree. However, the situation where some configurations have an isolated spin while others have none is fundamentally different from the situation where each configuration has some finite concentration of isolated spins. It might be interesting to investigate how z_w behaves in this regime.

A Determining the critical temperature T_c

We wanted to study the dynamics of our Monte Carlo algorithms near the critical point of the ferromagnetic to paramagnetic phase transition in the bond-diluted Ising model. This requires us to know what the corresponding critical temperature is. For this we could consult the literature, but for completeness we also want to show how we can determine the critical temperature ourselves. We will use the Binder cumulant method here, where the Binder cumulant $U(T, L)$ is defined as [17],

$$U(T,L) = 1 - \frac{\langle M^4 \rangle}{3 \langle M^2 \rangle^2} \tag{A.1}$$

where M is the total magnetisation of the Ising model. The intuition behind the Binder cumulant method is that in the ferromagnetic phase the distribution of the magnetisation M tends to two Gaussians centered around the spontaneous magnetisation values (which have ± symmetry), such that $U(T, L)$ tends to a non-zero value [17]. Meanwhile, in the paramagnetic phase the distribution of M becomes a single Gaussian centered around zero such that $U(T, L)$ also tends towards zero [17]. At the critical point there is a crossover between these two regimes and it turns out that the curves for different system sizes L all pass through the same point [17]. Therefore, if we plot $U(T, L)$ as a function of the temperature $(\beta J)^{-1}$ for different system sizes L and locate the intersection point then this gives us an estimate of the critical temperature.

We ran multiple simulations with the Swendsen-Wang algorithm for a range of temperatures and system sizes L at $p = 0.6$. For each temperature we used $25,000$ randomness configurations and for each configuration we ran for 100 Swendsen-Wang moves. The first 20 moves were discarded to allow the system to come to equilibrium. The computed Binder cumulants $U(T, L)$ as a function of the temperature $(\beta J)^{-1}$ can be seen in figure A.1. Notice that the curves indeed pass through the same point at a certain temperature. To be able to locate this intersection point better we plotted the same data again but zoomed in on the region around the intersection point in figure A.2. From the location of the intersection point we find for the critical temperature at $p = 0.6$ that $(\beta_c J)^{-1} = 0.955(5)$. This confirms that the simulations that we ran at $(\beta J)^{-1} = 0.940$ were in the vicinity of the critical point as intended. Our estimate is also in good agreement with other results [5].

We repeated these simulations with the same exact conditions but for $p = 0.7$. The results can be seen in figure A.3. By locating the intersection point we find that $(\beta_c J)^{-1} = 1.320(10)$ at $p = 0.7$. This again confirms that our simulations at $(\beta J)^{-1} = 1.310$ were near the critical point. This result is also consistent with other papers [5].

B Determining the critical exponents γ/ν

We mentioned that it has been shown that γ/ν is numerically approximately independent of the bond concentration p and equal to its value in the pure Ising model, i.e. $\gamma/\nu \approx 1.75$ [8]. Here we will give an example of how we could numerically determine γ/ν ourselves using a finite size scaling technique.

It is known that near the critical point the correlation length ξ and the magnetic susceptibility per spin $\chi = \frac{\beta}{N} \left(\left\langle |M|^2 \right\rangle - \langle |M| \rangle^2 \right)$ behave as [12],

$$\xi \sim |t|^{-\nu} \tag{B.1}$$

$$\chi \sim |t|^{-\gamma} \tag{B.2}$$

where t is the reduced temperature $t = ((\beta J)^{-1} - (\beta_c J)^{-1})/(\beta_c J)^{-1}$. Also notice that we used the absolute magnetisation $|M|$ in our definition of the susceptibility. We do this because in the absence of an external field the Ising model has an up-down symmetry so that the sign of the total magnetisation does not matter. During our simulations our system might move between equivalent states that differ only in the sign of the total magnetisation. If we would not take the absolute value then the presence of both positive and negative

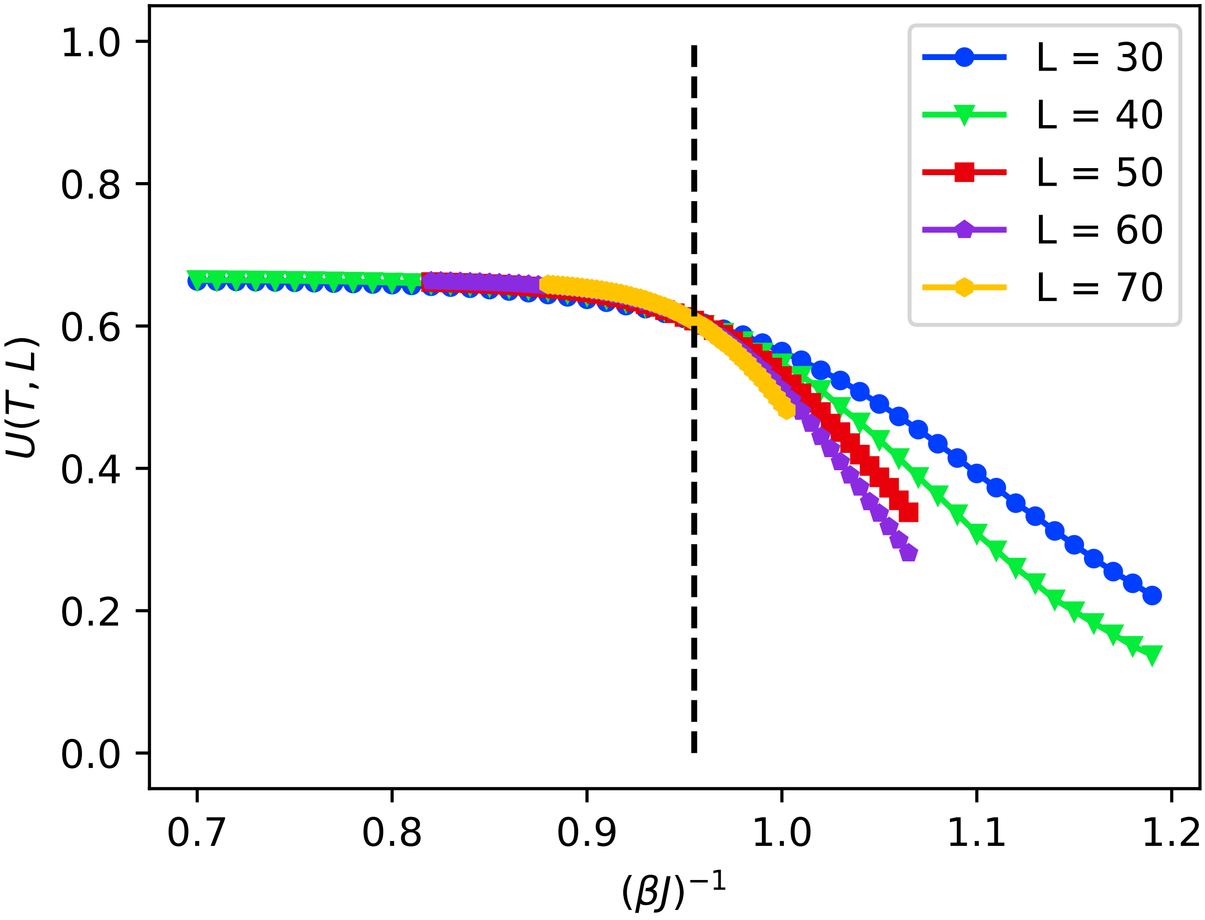


Figure A.1: The Binder cumulant $U(T, L)$ as a function of the temperature $(\beta J)^{-1}$ for various system sizes L at $p = 0.6$. Notice how all the curves go through the same point at a certain temperature, as indicated by the vertical dashed line. This intersection point gives us an estimate of the critical temperature of the ferromagnetic to paramagnetic phase transition. From this intersection point we find for the critical temperature that $(\beta_c J)^{-1} = 0.955(5)$.

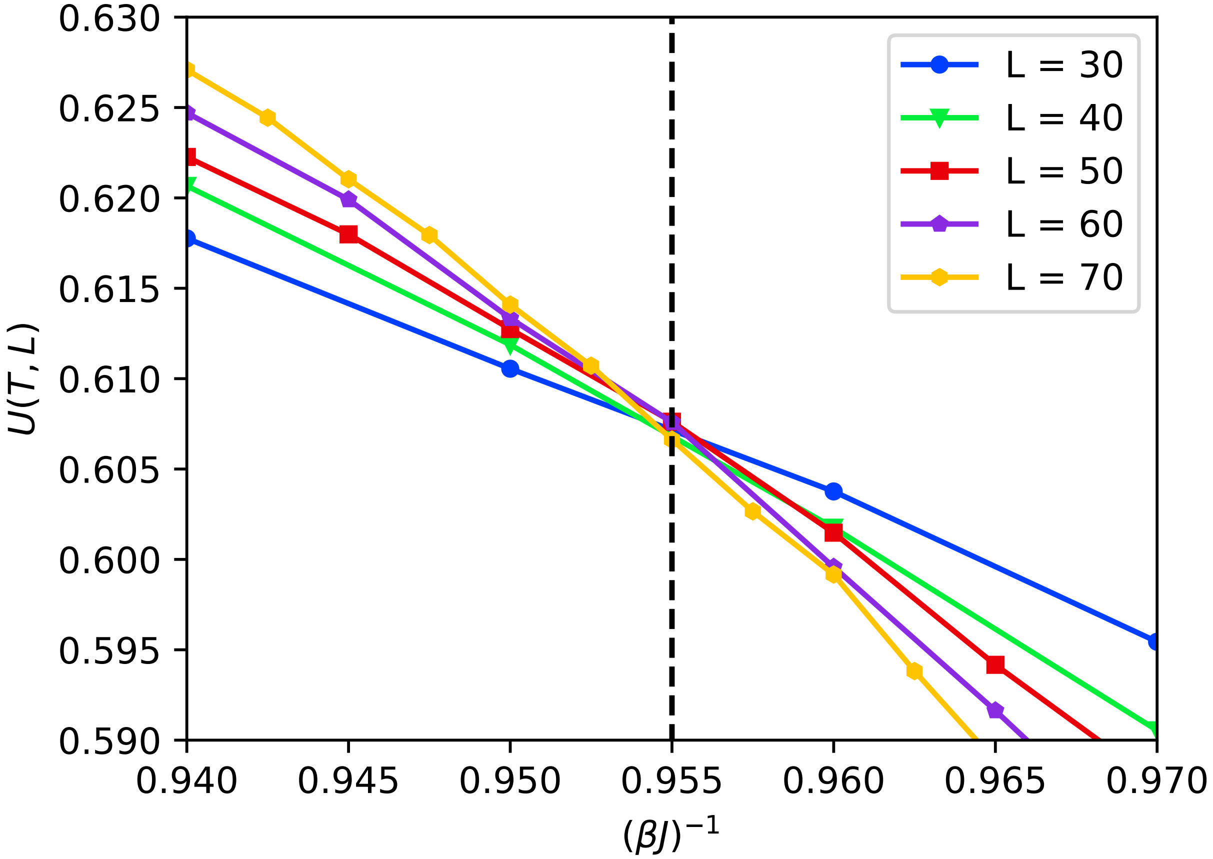


Figure A.2: The Binder cumulant $U(T,L)$ as a function of the temperature $(\beta J)^{-1}$ for various system sizes L at $p = 0.6$. This is the same plot as figure A.1 but zoomed in on the region surrounding the intersection point. From this plot we can more easily see that the intersection point is located at $(\beta_c J)^{-1} = 0.955(5)$ which is also indicated by the vertical dashed line.

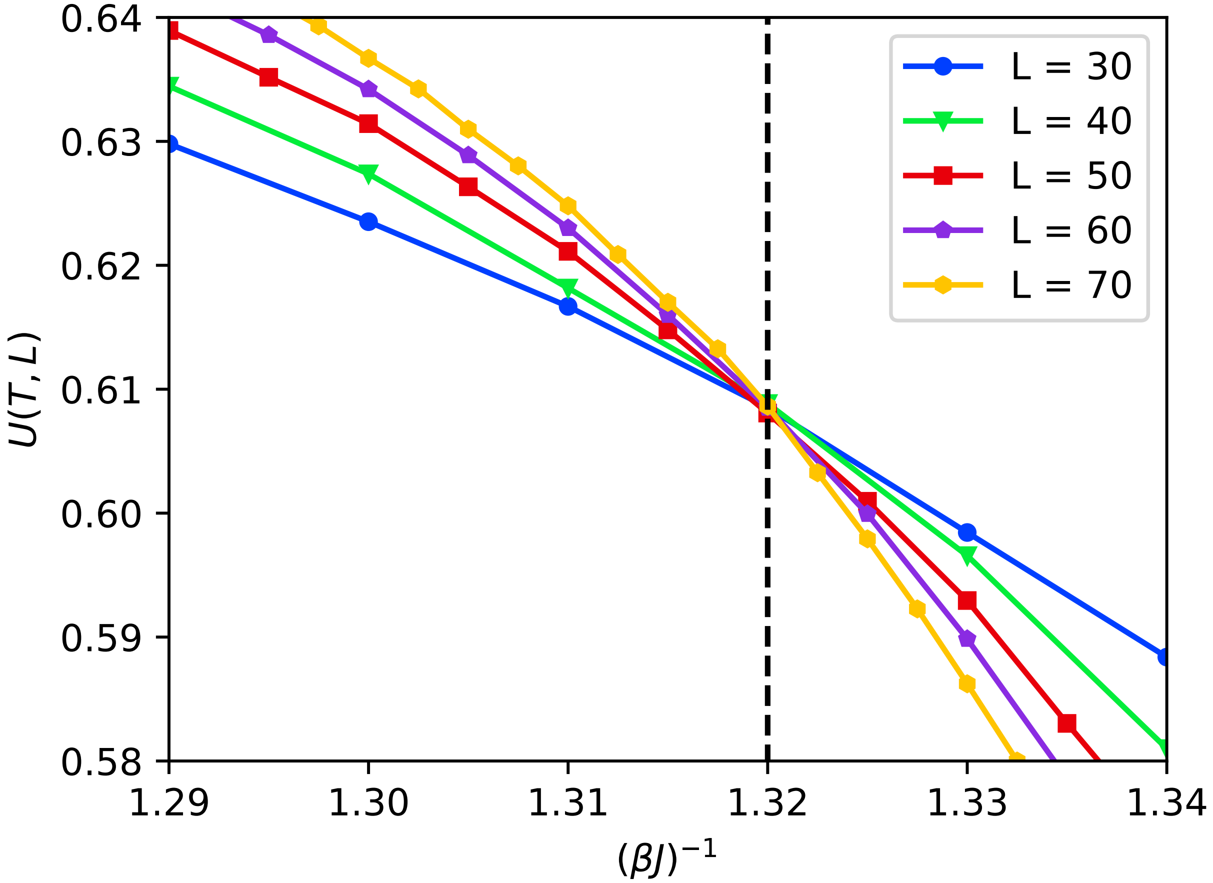


Figure A.3: The Binder cumulant $U(T, L)$ as a function of the temperature $(\beta J)^{-1}$ for various system sizes L at $p = 0.7$. Notice how all the curves go through the same point at a certain temperature, as indicated by the vertical dashed line. This intersection point gives us an estimate of the critical temperature of the ferromagnetic to paramagnetic phase transition. From this intersection point we find for the critical temperature that $(\beta_c J)^{-1} = 1.320(10)$.

(equivalent) values of M will give an incorrect estimate of the variance in the magnetisation and therefore an incorrect susceptibility χ. If we combine these two relations we find that [12],

$$\chi \sim \xi^{\gamma/\nu}. \tag{B.3}$$

This already suggests that the magnetic susceptibility will depend on our system size L. Now analogous to the discussion on the equilibrium mean-square displacement of the magnetisation $h(t)$ from section 6 we can rewrite χ to make its L dependence explicit. We will skip over most of the details here, the interested reader is referred to section 8.3.2 in [12], but we can show that we can write the magnetic susceptibility $\chi_L(t)$ as a function of the reduced temperature t in the form [12],

$$\chi_L(t) = L^{\gamma/\nu}\tilde{\chi}(L^{1/\nu}t) \tag{B.4}$$

where $\tilde{\chi}(x)$ is a scaling function independent of the system size L that satisfies [12],

$$\tilde{\chi}(x) \to \text{constant as } x \to 0. \tag{B.5}$$

Therefore, if we would plot $\chi_L(t)L^{-\gamma/\nu}$ as a function of $L^{1/\nu}t$ then we would be plotting the same scaling function $\tilde{\chi}(x)$ as a function of x for all system sizes L and all the curves should fall on top of each other, provided that we use the correct value for $(\beta_c J)^{-1}$, $\alpha = \gamma/\nu$ and ν [12].

We used the same data as was used in appendix A to compute $\chi_L(t)$ for a range of temperatures and system sizes L for both $p = 0.6$ and $p = 0.7$. In figure B.1 we can see the results for $p = 0.6$. We get an excellent collapse of the curves if we use $(\beta_c J)^{-1} = 0.955$, $\alpha = 1.79(3)$ and $\nu = 1.20(10)$. This suggests that $\gamma/\nu = \alpha = 1.79(3)$ at $p = 0.6$.

Figure B.2 shows the same plot but for $p = 0.7$. Again we can obtain a very good collapse when using $(\beta_c J)^{-1} = 1.320$, $\alpha = 1.77(3)$ and $\nu = 1.15(15)$. Therefore, we conclude that $\gamma/\nu = \alpha = 1.77(3)$ at $p = 0.7$. Our results for $p = 0.6$ and $p = 0.7$ support our earlier statement that γ/ν is numerically approximately independent of the bond concentration p and close to its value in the pure Ising model, i.e. $\gamma/\nu \approx 1.75$ [8].

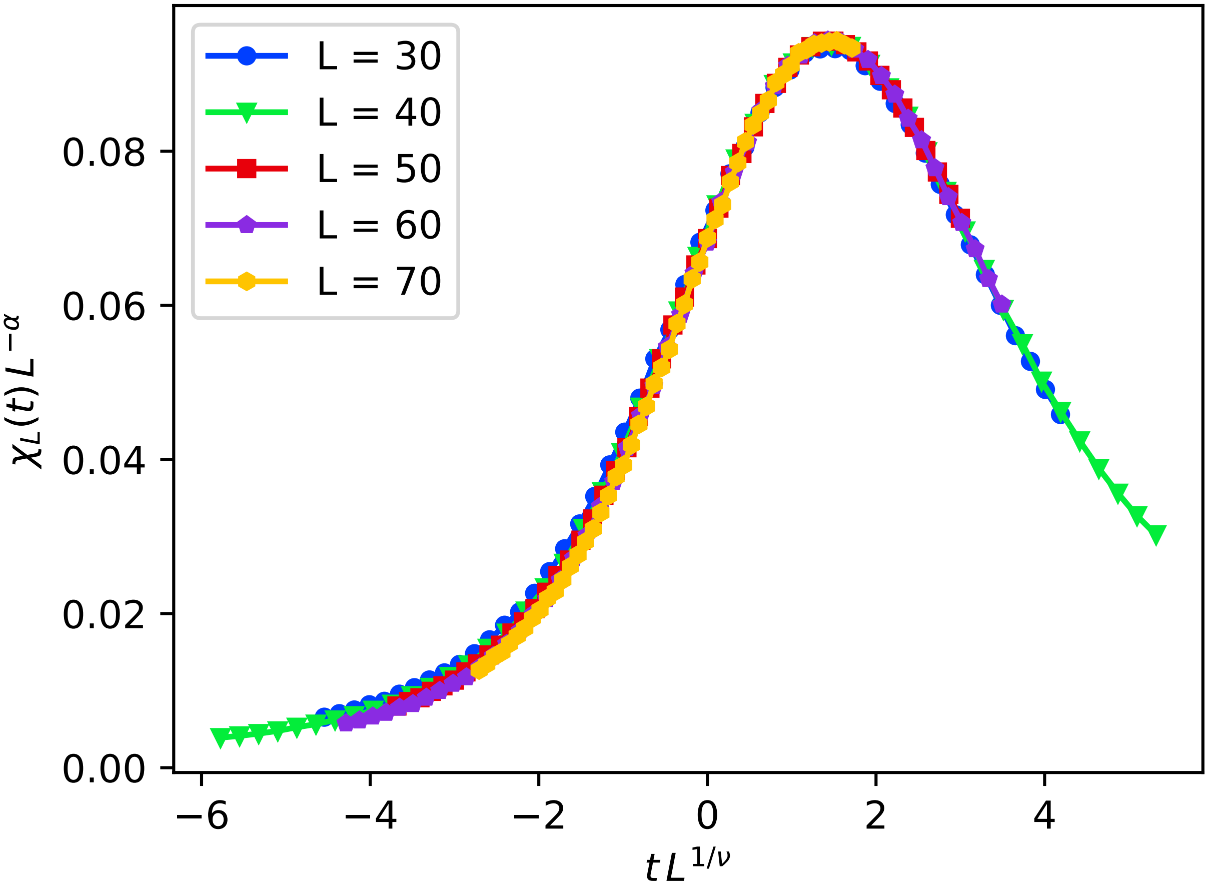


Figure B.1: The magnetic susceptibility $\chi_L(t)$ as a function of the reduced temperature t where $t = ((\beta J)^{-1} - (\beta_c J)^{-1})/(\beta_c J)^{-1}$ for various system sizes L at $p = 0.6$. For $(\beta_c J)^{-1}$ we used the value determined in figure A.1, i.e. $(\beta_c J)^{-1} = 0.955$. Notice the strikingly good collapse when we scale the vertical axis with $L^{-\alpha}$ and the horizontal axis with $L^{1/\nu}$ where $\alpha = 1.79(3)$ and $\nu = 1.20(10)$. From this we learn that the critical exponents $\gamma/\nu = \alpha = 1.79(3)$ at $p = 0.6$. The collapse of the curves also supports our estimate of $(\beta_c J)^{-1}$.

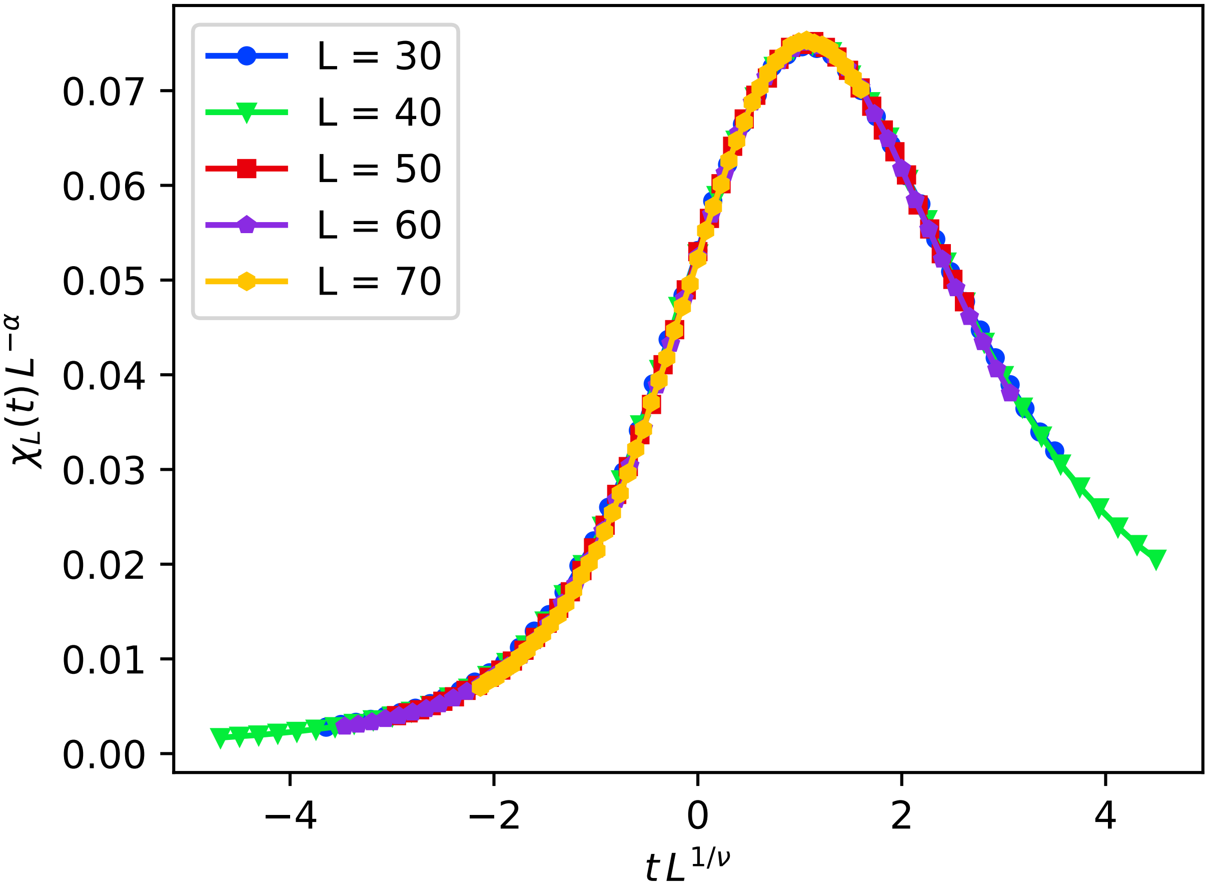


Figure B.2: The magnetic susceptibility $\chi_L(t)$ as a function of the reduced temperature t where $t = ((\beta J)^{-1} - (\beta_c J)^{-1})/(\beta_c J)^{-1}$ for various system sizes L at $p = 0.7$. For $(\beta_c J)^{-1}$ we used the value determined in figure A.3, i.e. $(\beta_c J)^{-1} = 1.320$. Notice the strikingly good collapse when we scale the vertical axis with $L^{-\alpha}$ and the horizontal axis with $L^{1/\nu}$ where $\alpha = 1.77(3)$ and $\nu = 1.15(15)$. From this we learn that the critical exponents $\gamma/\nu = \alpha = 1.77(3)$ at $p = 0.7$. The collapse of the curves also supports our estimate of $(\beta_c J)^{-1}$.

References

[1] Stephen G Brush. History of the lenz-ising model. *Reviews of modern physics*, 39(4):883, 1967.

[2] Martin Hasenbusch, Francesco Parisen Toldin, Andrea Pelissetto, and Ettore Vicari. The universality class of 3d site-diluted and bond-diluted ising systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(02):P02016, 2007.

[3] HG Ballesteros, LA Fernández, Víctor Martín-Mayor, A Munoz Sudupe, G Parisi, and JJ Ruiz-Lorenzo. Ising exponents in the two-dimensional site-diluted ising model. *Journal of Physics A: Mathematical and General*, 30(24):8379, 1997.

[4] Dmytro Ivaneyko, J Ilnytskyi, Bertrand Berche, and Yu Holovatch. Criticality of the random-site ising model: Metropolis, swendsen-wang and wolff monte carlo algorithms. *arXiv preprint cond-mat/0501291*, 2005.

[5] Wei Zhong, Gerard T. Barkema, and Debabrata Panja. Super slowing down in the bond-diluted ising model. *Physical Review E*, 102(2), Aug 2020. ISSN 2470-0053. doi: 10.1103/physreve.102.022132. URL http://dx.doi.org/10.1103/PhysRevE.102.022132.

[6] Martin Hasenbusch, Francesco Parisen Toldin, Andrea Pelissetto, and Ettore Vicari. Universal dependence on disorder of two-dimensional randomly diluted and random-bond $\pm j$ ising models. *Physical Review E*, 78(1), Jul 2008. ISSN 1550-2376. doi: 10.1103/physreve.78.011110. URL http://dx.doi.org/10.1103/PhysRevE.78.011110.

[7] D. Zobin. Critical behavior of the bond-dilute two-dimensional ising model. *Phys. Rev. B*, 18:2387–2390, Sep 1978. doi: 10.1103/PhysRevB.18.2387. URL https://link.aps.org/doi/10.1103/PhysRevB.18.2387.

[8] Ioannis A. Hadjiagapiou. Monte carlo analysis of the critical properties of the two-dimensional randomly bond-diluted ising model via wang–landau algorithm. *Physica A: Statistical Mechanics and its Applications*, 390(7):1279–1288, 2011. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2010.12.022. URL https://www.sciencedirect.com/science/article/pii/S037843711001054X.

[9] Pierre-Emmanuel Berche, Christophe Chatelain, Bertrand Berche, and Wolfhard Janke. Bond dilution in the 3d ising model: a monte carlo study. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(3):463–474, 2004.

[10] WF Wolff and J Zittartz. On the phase transition of the random bond ising model. *Zeitschrift für Physik B Condensed Matter*, 52(2):117–126, 1983.

[11] S Jain. Anomalously slow relaxation in the diluted ising model below the percolation threshold. *Physica A: Statistical Mechanics and its Applications*, 218(3-4):279–290, 1995.

[12] M. E. J. Newman and G. T. Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford University Press, Oxford, 1999.

[13] M Hennecke and U Heyken. Critical dynamics of cluster algorithms in the dilute ising model. *Journal of statistical physics*, 72(3):829–844, 1993.

[14] HG Ballesteros, LA Fernández, Víctor Martín-Mayor, A Munoz Sudupe, G Parisi, and JJ Ruiz-Lorenzo. Critical exponents of the three-dimensional diluted ising model. *Physical review B*, 58(5):2740, 1998.

[15] S.J. Blundell and K.M. Blundell. *Concepts in Thermal Physics*. Oxford University Press, Oxford, 2015.

[16] J.-C. Walter and G.T. Barkema. An introduction to monte carlo methods. *Physica A: Statistical Mechanics and its Applications*, 418:78 – 87, 2015. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2014.06.014. URL http://www.sciencedirect.com/science/article/pii/S0378437114004798. Proceedings of the 13th International Summer School on Fundamental Problems in Statistical Physics.

[17] K. Binder. Critical properties from monte carlo coarse graining and renormalization. *Phys. Rev. Lett.*, 47:693–696, Aug 1981. doi: 10.1103/PhysRevLett.47.693. URL `https://link.aps.org/doi/10.1103/PhysRevLett.47.693`.