Trinity College
The University of Dublin

Theoretical Physics

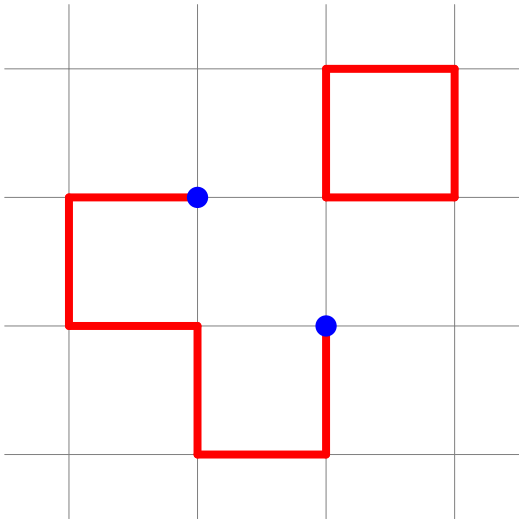
Final Year Project

# Binder Cumulants for the Ising Model using Worm algorithms

*Author:*
Ruairí Brett

*Supervisor:*
Prof. John Bulava

March 2015



**Abstract**

Prokof'ev & Svistunov's Worm algorithm is an alternative simulation strategy for classical statistical systems. It is based upon updating closed path configurations produced by high-temperature expansions through the motion of end points of a disconnected path. We begin by presenting an analysis of this algorithm and the reformulation of the 2D Ising Model using such high-temperature expansions/duality transformations. What then follows is a reproduction of relevant results from published literature before the main focus of this project. Here, we extend the simulation scheme to what we call a *Two Worm* algorithm, in order to obtain direct Monte Carlo estimators that are not available in the standard worm representation. An ergodic update scheme that satisfies detailed balance is subsequently introduced and our new algorithm is applied using C++ code. At this stage, we use such newly retrievable observables to calculate fourth-order Binder Cumulants for the 2D square lattice Ising model and examine the effects of finite size scaling on the algorithm. Finally, we present a comparison of our algorithm to the original worm method using magnetic susceptibility as our common observable between the two update schemes.

# Declaration

I declare that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as cited, and has not been submitted, either in part or whole, for a degree at this or any other university.

---

Ruairí Brett

March 2015

# Acknowledgements

I would like to thank Prof. John Bulava for all of his help throughout the course of this project and the Botany Bay football pitch for granting me an extension via a broken leg. I would also like to thank Kevin & Eamonn for proof-reading and Áine for correcting my codeine addled ramblings.

# Contents

# List of Figures

# Chapter 1

# Motivation & Background Material

## 1.1 Motivation

The Metropolis-Hastings algorithm [1] is usually the most widely used approach to Monte Carlo simulations as it is universal, fast and easy to program. However, near phase transition points (critical points), its advantages are virtually cancelled out. The belief is that any scheme based on local Metropolis-type updates will be inefficient at the transition point as autocorrelation times will scale as $L^{\xi}$ where $L$ is the linear size of the system and $\xi$ is the dynamical critical exponent, often close to 2 [2]. In 1998, Prokof'ev and Svistunov introduced their quantum "Worm" algorithm as a novel quantum Monte Carlo scheme that allows for efficient calculations of Green's functions and the study of large disordered systems [3]. Then, in 2001, the approach was presented for use in simulations of classical spin systems in two and three dimensions [4]. The power of this new method is that it is a local, Metropolis-type scheme that seems to have dynamical critical exponents close to zero (i.e., with efficiency comparable to the best cluster methods today). Much work in the Condensed Matter and High Energy Theory communities has been done with the worm algorithm in applying the method to different physical systems and investigating it as a viable alternative to current Monte Carlo simulations methods [5–8].

In this project we will first familiarise ourselves with the theory behind the (classical) worm algorithm, in particular focusing on the exact reformulation of the 2D Ising model using high-temperature expansions/duality transformations. We will then present some results of simple single worm simulations written in C++ and compare the results to published data [9]. This consideration will lead us to the main focus of the project; the extension of Prokof'ev & Svistunov's algorithm to what we refer to as a *Two Worm* algorithm in order to extend the range of spin observables for which we can retrieve direct Monte Carlo estimators not available in the single worm representation. Our primary motivation here is to calculate fourth-order *Binder Cumulants* (eq. (3.5)), a common tool used in determining critical points, for the 2D Ising

model. As these Binder Cumulants depends on higher order spin observables than are directly retrievable from the original (single) worm algorithm, we present our extended formulation from which direct estimators of such observables may be taken (eg. eq. (3.6)). We will present this further reformulation of the algorithm and observables along with a detailed comparison of the method to the original (single) worm algorithm.

It will be useful to consider how autocorrelation times are affected by the new method and compare the effects of finite size scaling in our simulations. We will calculate dynamical critical exponents, $\xi$, for each of our primary observables with the hope that they will be close to zero, indicating a complete absence of critical slowing down.

## 1.2 Ising Model

The Ising Model, developed by German physicist Ernst Ising, is a mathematical model of ferromagnetism in statistical mechanics. As one of the simplest models in statistical mechanics to display phase transitions, the Ising Model has been applied not just across various fields in physics but to research in financial markets and to neural networks [10]. For the purposes of our project, we will study the two dimensional zero field square lattice Ising model.

In his PhD thesis (1924) Ising was able to solve the one dimensional linear model which displays no phase transition. That is, for positive temperature, $\beta$, the correlations between spins $\langle \sigma_i \sigma_j \rangle$ decay exponentially with the distance $|i-j|$

$$\langle \sigma_i \sigma_j \rangle \leq C(\beta) e^{-f(\beta)|i-j|},$$

where $C(\beta), f(\beta)$ are positive for $\beta > 0$, meaning that the system is disordered. In 1936, Rudolf Peierls proved that in two or higher dimensions the model undergoes a phase transition between an ordered and disordered phase [11]. The 2D square lattice Ising model with no magnetic field was then solved analytically by Onsager [12] (1944).

### 1.2.1 Definition

Given a set of lattice sites $\Lambda$, each with a set of neighbouring sites, we denote for each site $k \in \Lambda$ the discrete variable $\sigma_k$ such that $\sigma_k \in \{+1, -1\}$, representing the spin at $k$. A **lattice configuration**, $\sigma = (\sigma_k)_{k \in \Lambda}$, is an assignment of a specific spin value at each lattice site and we will use $\{\sigma\}$ to denote the set of all possible spin configurations.

In the zero field model (no external magnetic field) the energy of a configuration $\sigma$ is given by the Hamiltonian

$$E(\sigma) \equiv H(\sigma) = -\sum_{\langle i,j \rangle} J_{ij} \sigma_i \sigma_j, \tag{1.1}$$

where $\langle i, j \rangle$ indicates that $i$ and $j$ are nearest neighbours and $J_{ij}$ is the interaction between any two adjacent sites $i$ and $j$. In our case we will assume that each set of nearest neighbours $\langle i, j \rangle$ has the same interaction strength ($J_{ij} = J, \forall \langle i, j \rangle \in \Lambda$). The configuration probability is then given by the usual Boltzmann distribution with inverse temperature $\beta \geq 0$

$$P_\beta(\sigma) = \frac{e^{-\beta E(\sigma)}}{Z_B}, \tag{1.2}$$

where the normalisation constant

$$Z_B = \sum_{\{\sigma\}} e^{-\beta E(\sigma)}$$

is the usual Boltzmann partition function.

### 1.2.2 Observables

We calculate observables in the usual way using this configuration probability for some quantity $A(\sigma)$ that is a function of the spins using

$$\langle A(\sigma) \rangle = \frac{1}{Z_B} \sum_{\{\sigma\}} A(\sigma) e^{-\beta E(\sigma)}. \tag{1.3}$$

This leads to simple calculations to derive expressions for interesting physical quantities associated with the model such as the average **magnetisation**

$$\langle M \rangle = \langle \bar{\sigma} \rangle. \tag{1.4}$$

However, since we are looking at the zero field Ising Model, the average magnetisation is zero, meaning

$$\langle \bar{\sigma} \rangle = \langle M \rangle = 0. \tag{1.5}$$

The **magnetic susceptibility** is then

$$\begin{aligned} \langle \chi \rangle &= \langle M^2 \rangle - (\underbrace{\langle M \rangle}_{=0})^2, \\ &= \langle \bar{\sigma}^2 \rangle. \end{aligned}$$

## 1.3 Monte Carlo Methods

Monte Carlo methods refers to a broad class of computational algorithms that employ random sampling to obtain numerical results. Such methods vary in implementation, but tend to follow a similar pattern:

1. Define a domain of possible input values
2. Generate inputs randomly from a probability distribution over that domain
3. Perform a deterministic computation using these inputs
4. Aggregate the results

### 1.3.1 Markov Chain Monte Carlo

A **Markov Chain** is a stochastic process that possesses the Markov property, that is, the value of the random variable at the current state of the chain depends only on the value assumed at the immediately previous step. **Markov Chain Monte Carlo** (MCMC) methods are then a class of algorithms for sampling from a given probability distribution (Monte Carlo) with the intention of then constructing a Markov Chain that has the desired distribution as its equilibrium distribution. After a number of time steps, the state of the chain can then be used as a sample of the desired distribution.

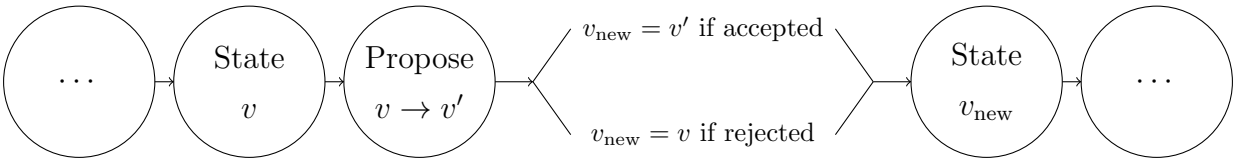



Figure 1.1: One 'link' of a Markov Chain

MCMC methods are used throughout Statistical Physics to evolve systems in time through their configuration space states. The Ising Model is usually simulated numerically using Markov Chain Monte Carlo methods with the most commonly used Monte Carlo algorithm for the Ising model is the *Metropolis-Hastings algorithm* [1].

### 1.3.2 Detailed Balance

For a probability distribution of states $\boldsymbol{\pi}$, we require that the Markov chain we construct with our algorithm must be reversible so that $\boldsymbol{\pi}$ is a steady-state distribution. We first must note that for an ergodic chain $\{X_n\}$ with transition probability $p^X$ and stationary distribution $\boldsymbol{\pi}$,

the **time-reversed** process $\{Y_n\}$ will also satisfy the Markov property. This process is then defined as

$$p_{ij}^Y = P\big(Y_{n+1} = j \mid Y_n = i\big) = \left(\frac{\pi_j}{\pi_i}\right) p_{ij}^X. \tag{1.6}$$

and $\{Y_n\}$ is then a Markov chain. We will then call an ergodic chain $\{X_n\}$ **reversible** if its transition matrix is equal to the transition matrix of the time-reversed chain. That is

$$p_{ij}^Y = p_{ij}^X, \qquad \forall\ i, j. \tag{1.7}$$

These two equations can then be expressed as

$$\pi_i p_{ij} = \pi_j p_{ji}, \tag{1.8}$$

which is the condition for a Markov chain to satisfy **detailed balance**. As checking detailed balance is often the simplest way to verify that a particular distribution is stationary, we will need to make sure that the Markov chain created by our algorithm satisfies the condition.

## 1.4 Statistics

### 1.4.1 Autocovariance and Autocorrelation

In order to get an idea of the effectiveness of any Markov process we can look at the **Autocorrelation** of our measured observables. Autocorrelation is a measure of the correlation or relationship between members of a series of observations taken over time and the same values taken at a fixed time interval later. Simply, we can think of the autocorrelation as a measure of how long it takes for an observable produced by the algorithm to become independent of the previous measurements.

We start by defining the **autocovariance** of a Markov chain $\{x_i\}, i = 1, \ldots, N$ with average value $\bar{x}$ as

$$\begin{aligned} R(t) &= \langle (x_i - \bar{x})(x_{i+t} - \bar{x}) \rangle, \\ &= \langle x_i x_{i+t} \rangle - \bar{x}^2, \\ &\approx \sum_{i=1}^{N-t} \frac{x_i x_{i+t}}{(N-t)}. \end{aligned} \tag{1.9}$$

Noting that $R(0) = \sigma^2$ ($\sigma^2 \equiv \langle (x - \bar{x})^2 \rangle$) is the variance, it is now more convenient to work with the **autocorrelation** which we define as

$$\rho(t) = \frac{R(t)}{R(0)}, \tag{1.10}$$

so that $\rho(0) = 1$ and $-1 \leq \rho(t) \leq 1$. In practice it is useful to define the **integrated autocorrelation**:

$$\tau_{\text{int}}(\tau) = \frac{1}{2} + \sum_{t=0}^{\tau} \rho(t), \tag{1.11}$$

which will converge to the **integrated autocorrelation time**, $\tau_{\text{int}} = \lim_{\tau \to \infty} \tau_{\text{int}}(\tau)$, an explicit measure of the autocorrelation which we can use for quick comparison between data sets.

It must also be noted that we can use the following approximations to calculate the error for the autocorrelation function:

$$(\delta\rho(t))^2 \approx \frac{1}{N} \sum_{k=1}^{\infty} \left[ \rho(k+t) + \rho(k-t) - 2\rho(k)\rho(t) \right]^2, \tag{1.12}$$

and for the integrated autocorrelation time:

$$(\delta\tau_{\text{int}}(\tau))^2 \approx \frac{2(2\tau_{\text{int}}+1)}{N}\tau_{\text{int}}^2(\tau). \tag{1.13}$$

### 1.4.2 Jackknife Resampling

As is common in these types of Monte Carlo simulations, we will need some method of estimating the precision of our sample statistics using **resampling** techniques. Using **jackknife resampling** we will be able to estimate the uncertainties in our primary observables. The jackknife estimate of a parameter is found by systematically leaving out each measurement from the dataset and calculating the estimate from the remaining observations. The overall jackknife estimator is then found by averaging each of these subsample estimators.

Given a set of $N$ measurements, $\{\theta_i\,;\, i = 1, \ldots, N\}$, each subsample estimate, $\theta_{J_i}$, is given by

$$\theta_{J_i} = \frac{1}{N-1}\sum_{\substack{k=1\\k\neq i}}^{N}\theta_k, \tag{1.14}$$

and the overall jackknife estimator is then given by [13]

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_{J_i}. \tag{1.15}$$

An estimate of the variance of an estimator can then easily be calculated from these jackknife estimates using [14]

$$\text{Var}(\theta) = \frac{N-1}{N}\sum_{i=1}^{N}\left(\theta_{J_i} - \bar{\theta}\right)^2. \tag{1.16}$$

#### Binning

To reduce the effects of minor observation errors and to save computational time when calculating jackknife estimates, we will group our observations in small intervals, or bins, which will be replaced in our calculations by a representation of that interval. This process is known as **data binning**.

# Chapter 2

# Single Worm Algorithm

## 2.1 Overview

As seen in most Statistical Mechanics or Numerical Methods courses, the traditional way to simulate the Ising Model using MCMC simulations is with a spin flip Metropolis scheme. This amounts to sweeping through the lattice flipping individual spins based on Metropolis acceptance probabilities dependant on the energy of the system. While both easy to represent in computer code and efficient away from transition points, the method struggles for practical simulations. The advantages of this local method are virtually cancelled out at phase transition points as the autocorrelation time, $\tau_{\text{int}}$, will scale as $L^{\xi}$ where $L$ is the system linear dimension (lattice size) and $\xi \approx 2$ is the dynamical critical exponent. At this critical point then is both where numerical simulations begin to break down while being the physically most interesting domain. The advent of cluster algorithms introduced by Swendsen and Wang [15] (1987) and further developments [16–19] provided a huge improvement over single spin flip updates at the critical point yet, both the classical and quantum methods rely on global updates and are essentially non-local schemes.[1]

In 2001, Prokof'ev and Svistunov proposed a **Worm Algorithm** for classical statistical models [4] that eliminates this problem of slowing at the critical point yet remains a local scheme. Using high temperature expansions we move to a new configuration space of closed paths[2] (figure 2.1), whose evolution in time we simulate through the motion of the end points of a disconnected path. As an alternative to cluster methods, the Worm algorithm is based on radically different principles and as such, has another range of potential applications. For example, the closed path representation is suitable for the study of superfluid models by having direct Monte Carlo estima-

[1] Updates are local if they address configuration elements and change them without knowing about other elements outside of the updated part

[2] In the case of the Ising model this is equivalent to a duality transformation between high- and low-temperature states

tors for superfluid stiffness [20], which are not available through the standard site representation.[3]

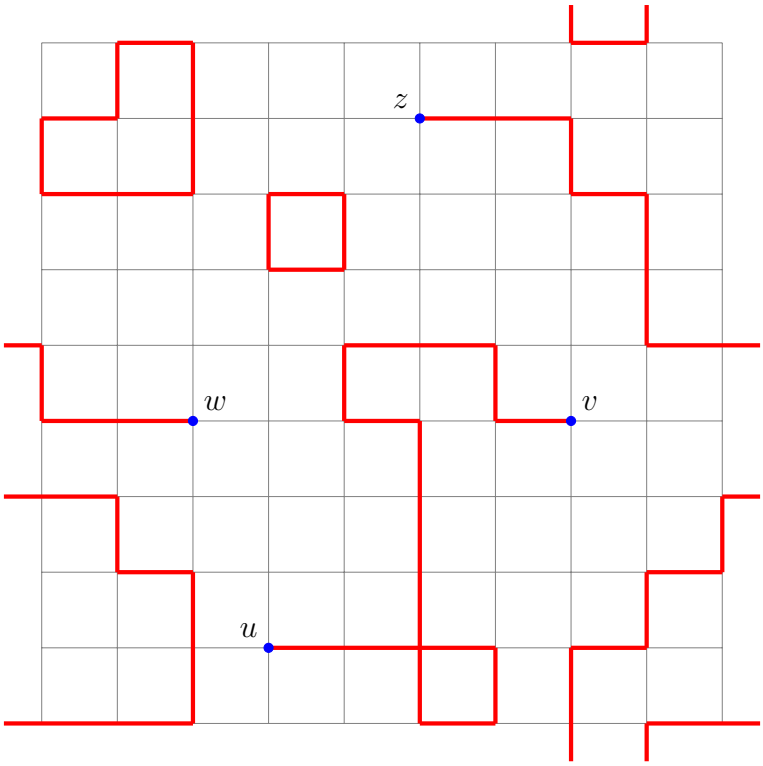



Figure 2.1: Typical lattice configuration in the two worm reformulation (See later)

The real power of the Worm algorithm is then that we are not just discussing a new Monte Carlo method, but an exact reformulation where physical observables have completely different estimators, variances and autocorrelations which all deserve study. The algorithm has seen great success in applications such as the modelling of weakly interacting 2D and 3D Bose gases or the direct simulation of ultra-cold atoms in a 3D optical lattice. At the same time it has also seen increasing popularity as a possible strategy for use in lattice field theory [6, 7].

[3]We will see later how this leads us to consider the possibility of constructing a 2 worm algorithm for direct computation of higher order spin observables

## 2.2 Reformulating the Ising Model

### 2.2.1 Closed Loop Constraints

In a rather confusing notation similar to that seen in [9], we define

$$\begin{aligned} Z(u,v) &= \sum_{\{\sigma\}} \sigma_u \sigma_v \, e^{\beta \sum\limits_{\langle x,y\rangle} \sigma_x \sigma_y}, \\ &= \sum_{\{\sigma\}} \prod_{\langle x,y\rangle} \sigma_u \sigma_v \, e^{\beta \sigma_x \sigma_y}, \end{aligned} \tag{2.1}$$

where the outer sum is over all Ising spin configurations $\{\sigma_i = \pm 1\}$, and in the exponent we sum over nearest neighbour links $\langle x, y\rangle$.
Then, using the identity:

$$e^{\beta \sigma_x \sigma_y} = \cosh\beta \sum_{k=0,1} \left(\sigma_x \sigma_y \tanh\beta\right)^k, \tag{2.2}$$

we get

$$Z(u,v) = \sum_{\{\sigma\}} \sigma_u \sigma_v \prod_{\langle x,y\rangle} \left[ \cosh\beta \sum_{k=0,1} \left(\sigma_x \sigma_y \tanh\beta\right)^k \right],$$

letting $l = \langle x, y\rangle$ denote the link between sites $x$ and $y$ then gives us (dividing by $2^{N_x}$ for convenience, where $N_x$ is the total number of sites)

$$\begin{aligned} Z(u,v) &= \frac{1}{2^{N_x}} \sum_{\{\sigma\}} \sigma_u \sigma_v \prod_{\langle x,y\rangle} \left[ \cosh\beta \sum_{k=0,1} \left(\sigma_x \sigma_y \tanh\beta\right)^k \right], \\ &= \frac{1}{2^{N_x}} \left(\cosh\beta\right)^{N_l} \sum_{\{\sigma,k\}} \sigma_u \sigma_v \prod_{l} \left[ \tanh\beta \prod_{x \in \partial l} \sigma_x \right]^{k_l}, \end{aligned} \tag{2.3}$$

where $N_l$ is the total number of **links** or **dimers**, $k_l = 0, 1$ is the bond/link variable ($0$ = link off, $1$ = link on) and $\partial l = \{x, y\}$ is the boundary set of nearest neighbours $x$ and $y$.

Now, if we sum over the spins, $\{\sigma\}$, we arrive at the (global) constraint

$$\frac{1}{2^{N_x}} \sum_{\{\sigma\}} \sigma_u \sigma_v \prod_{l} \left( \prod_{x \in \partial l} \sigma_x \right)^{k_l} = \Theta(k; u, v) \in \{0, 1\}, \tag{2.4}$$

where $\Theta$ factorises into local constraints with

$$\theta(k;y) = \begin{cases} 1 & \text{if } \sum_{l,\partial l \ni y} k_l = \text{even}, \\ 0 & \text{else}, \end{cases} \tag{2.5}$$

and its complement

$$\bar{\theta}(k;y) = 1 - \theta(k;y). \tag{2.6}$$

These then combine to give

$$\Theta(k;u,v) = \prod_{y \notin \{u,v\}} \theta(k;y) \times \begin{cases} \theta(k;u) & \text{if} \quad u = v, \\ \bar{\theta}(k;u)\bar{\theta}(k;v) & \text{else.} \end{cases} \tag{2.7}$$

This constraint then, in words, means that if

- $\mathbf{u} = \mathbf{v}$ : The number of 'on' dimers/links at any site must be <u>even</u> (all closed loops).
- $\mathbf{u} \neq \mathbf{v}$ : $u$ and $v$ (the end points of the worm/disconnected path) must be surrounded by an <u>odd</u> number of 'on' dimers/links with all other sites surrounded by an <u>even</u> number

The algorithm's namesake then presumably derives from the fact that the constraint $\Theta(k;u,v) \neq 0$ requires a line of active links/dimers ($k_l = 1$) connecting $u$ and $v$. However, this series of connecting links is not unique, so the worm is 'fuzzy' so to speak with only the head and tail $(u,v)$ fixed.

### 2.2.2 Duality Transformations

On a more technical point, it is important to note that the steps taken above in the high-temperature/strong coupling expansion [21] in $\tanh\beta$ (eq. 2.2) coincide with those made in a duality transformation [22].[4] In the case of the 2D square lattice Ising model, at high temperatures most spins are independent and we can consider how the energy changes as small subsets of sites interact. Drawing lines between the interactions resembles vertices analogous to our links/dimers. At very low energies, almost all sites are aligned, and we can consider how the energy changes as small islands of opposite spins appear. These resemble cell clusters. Thus in the Ising model lattice duality interchanges high- and low-temperature states. Indeed, the square lattice is **self-dual** meaning that it is its own dual.

[4] Updated version available online: [23]

### 2.2.3 Partition Functions

Now, by generalising the approach seen in [4] we consider a new partition function (for one worm):

$$\mathcal{Z}_1 = \sum_{u,v} \rho^{-1}(u,v) Z(u,v), \tag{2.8}$$

$$= \sum_{\substack{u,v \\ \{k_l\}}} \frac{\Theta(k;u,v)}{\rho(u,v)} e^{-\mu \sum_l k_l}, \tag{2.9}$$

where the coupling $\beta$ has been replaced with the dimer chemical potential

$$\begin{aligned} \tanh\beta &= e^{-\mu}, \\ \Rightarrow \prod_l (\tanh\beta)^{k_l} &\to e^{-\mu \sum_l k_l}. \end{aligned} \tag{2.10}$$

The pairs of sites $u, v$ are now the 'phase space' of our system and $0 < \rho(x) < \infty$ is a weighting function that we demand possesses the lattice periodicity. We also normalise the weighting factor at the origin as constant positive factors in $\rho$ will be irrelevant

$$\rho(0) = 1.$$

We can now calculate observables $A(k;u,v)$, which may now also depend on $k_l$, in the usual way

$$\langle A(k;u,v) \rangle_1 = \frac{1}{\mathcal{Z}_1} \sum_{\substack{u,v \\ \{k_l\}}} A(k;u,v) \frac{\Theta(k;u,v)}{\rho(u,v)} e^{-\mu \sum_l k_l}, \tag{2.11}$$

where $\langle \cdots \rangle_1$ denotes the thermodynamic average with respect to (2.9). In practice however, for deriving the form of particular observables, we will use a less explicit form of the partition function

$$\mathcal{Z}_1 = \sum_{u,v} \sum_{\{\sigma\}} \frac{\sigma_u \sigma_v}{\rho(u,v)} e^{-\beta E(\sigma)}, \tag{2.12}$$

where we have just taken the usual Boltzmann partition function and introduced a sum over all configurations of the end points of the worm. Spin observables with respect to this form of the partition function are then easily derived using

$$\langle A(\sigma) \rangle_1 = \frac{1}{\mathcal{Z}_1} \sum_{u,v} \sum_{\{\sigma\}} A(\sigma) \frac{\sigma_u \sigma_v}{\rho(u,v)} e^{-\beta E(\sigma)}, \tag{2.13}$$

instead of 2.11.

### 2.2.4 Observables

As a test of the single worm algorithm, we will use existing results from [9] for the susceptibility of the Ising model in 2D as a benchmark against which to test. This gives us the opportunity to introduce the manner in which we will be able to relate spin observables to easily calculated observables in our new configuration space. In [9] the two-point function is used to relate average nearest neighbour correlations to the energy and susceptibility whereas we will more explicitly relate powers of average spin in the original Ising model to observables that we can retrieve from the reformulated model. Note also that from here we will set the weighting factor $\rho \equiv 1$ for all $u$ and $v$ for simplicity and for more accurate confirmation of results.

We recall that for the $d$-dimensional square lattice Ising model the average spin is given by

$$\bar{\sigma} = \frac{1}{L^d} \sum_i \sigma_i, \tag{2.14}$$

which is used to calculate the susceptibility

$$\langle \chi \rangle = \langle \bar{\sigma}^2 \rangle. \tag{2.15}$$

By considering Kronecker delta functions as our worm observables, equivalent to an average of the number of times worm end points $u$ and $v$ coincide, we can relate an easily retrievable observable to the susceptibility.
We start with

$$\langle \delta_{uv} \rangle_1 = \frac{1}{\mathcal{Z}_1} \sum_{u,v} \sum_{\{\sigma\}} \delta_{uv} \sigma_u \sigma_v e^{-\beta E(\sigma)}, \tag{2.16}$$

sum over $u$ and $v$

$$\begin{aligned} \langle \delta_{uv} \rangle_1 &= \frac{1}{\mathcal{Z}_1} \sum_{v} \sum_{\{\sigma\}} \sigma_v \sigma_v e^{-\beta E(\sigma)}, \\ &= \frac{L^d}{\mathcal{Z}_1} \sum_{\{\sigma\}} e^{-\beta E(\sigma)}, \end{aligned}$$

and then rewrite $\mathcal{Z}_1$ using the expression for average spin (2.14) as

$$\begin{aligned}\langle \delta_{uv} \rangle_1 &= \frac{L^d \sum\limits_{\{\sigma\}} e^{-\beta E(\sigma)}}{\sum\limits_{uv} \sum\limits_{\{\sigma\}} \sigma_u \sigma_v e^{-\beta E(\sigma)}}, \\ &= \frac{L^d \sum\limits_{\{\sigma\}} e^{-\beta E(\sigma)}}{L^{2d} \sum\limits_{\{\sigma\}} (\bar{\sigma})^2 e^{-\beta E(\sigma)}}, \\ &= \frac{1}{L^d \langle \bar{\sigma}^2 \rangle},\end{aligned}$$

so in 2D:

$$\langle \chi \rangle = \frac{1}{L^2 \langle \delta_{uv} \rangle_1}. \tag{2.17}$$

This and similar Kronecker delta observables will be the most useful and easy to estimate quantity for calculating observables both with the single worm algorithm, and later when we extend to two worms. In general, the most physically interesting observables tend to be of the derived type; nonlinear functions, typically ratios of primary observables (see eq. (3.5)) and details of these are found throughout existing literature. In our case, we will use the susceptibility as a benchmark for both the one and two worm case and will then look at $4^{\text{th}}$ order Binder Cumulants as an example of a secondary observable in the two worm method.

## 2.3 Designing an Algorithm

With the algorithm from [4] we sample the statistical ensemble (eq. 2.9) produced using high-temperature expansions. The method exploits the fact that we can base an ergodic Monte Carlo algorithm for the ensemble entirely on two elementary alternating steps that together make a single compound update:

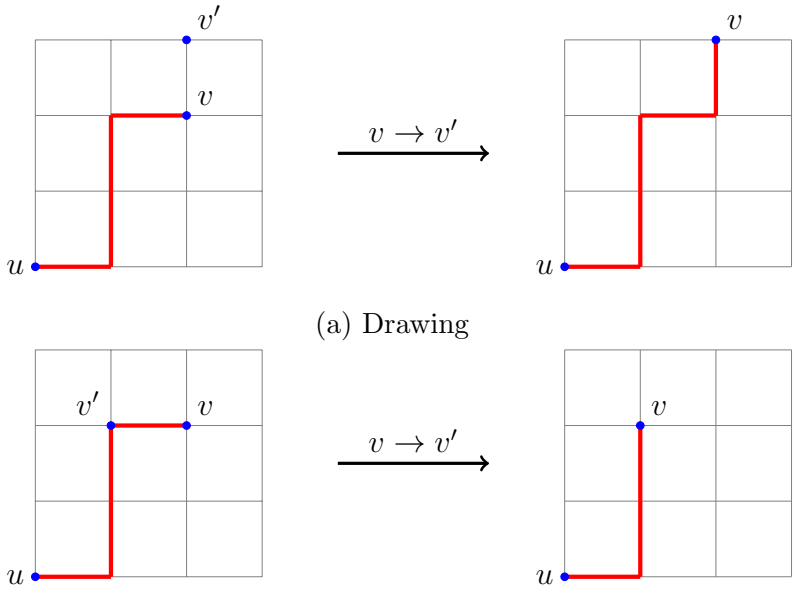
- **Move:** (Figure 2.2) Pick one of the $v$'s nearest neighbours with equal probability and denote it $v'$ with connecting link $l$. The proposed move is then $v \to v'$ with simultaneous adjustment $k_l \to 1 - k_l$ and is accepted with the Metropolis probability:

$$p_{\text{acc}} = \min\left(1, \frac{\rho(u,v)}{\rho(u,v')} e^{\mu(2k_l - 1)}\right), \tag{2.18}$$

  otherwise the original configuration is maintained. Since the update and system are translationally invariant, we only need to move $v$ and can keep $u$ fixed.

- **Kick:** (Figure 2.3) If the system is in a configuration with $u = v$, we 'kick' the coincident pair to another randomly chosen lattice site with unchanged $\{k_l\}$ (no links are changed) with probability $0 < p_{\text{kick}} < 1$. For the dominant case where $u \neq v$, we do nothing in this step.

We will call each move-kick pair of updates a *micro-step*, each of which will require O(1) operations, regardless of the lattice size. However, in practice when we run simulations, we will group $L^2$ of these micro-steps into a single iteration or sweep of the algorithm re-introducing the lattice size into CPU time.




(a) Drawing

(b) Erasing

Figure 2.2: Successful **Move** updates

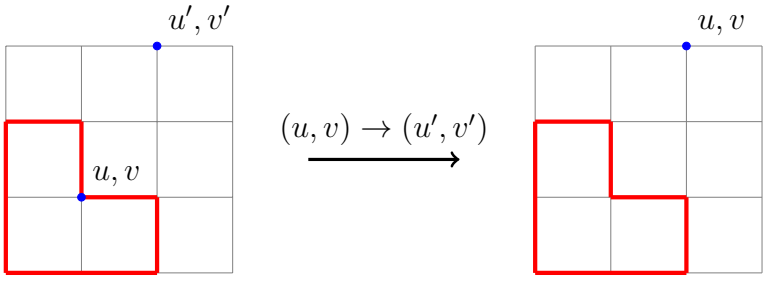



Figure 2.3: Successful **Kick** update

Any two configurations may be connected with non-zero probability by the worm 'dismantling' all active links of the first configuration and then 're-building' the second one over successive updates. The kick step allows for the worm to move from one connected component to the next.

As demonstrated in [8] and as we will show later, a correct and still efficient algorithm can be based purely on worm move updates ($p_{\text{kick}} = 0$). While autocorrelation times do increase without the kick update, the increase is minor. Later, when we consider the system with two worms, this fact will be useful in allowing us to run simulations without needing an ergodic kick update.

### 2.3.1 Simulation notes

For the purposes of our simulations, as stated above, we will set $\rho(x) \equiv 1$ for all $x$ as to match the chosen literature and will initially set $p_{\text{kick}} = 0.5$, although we will see later how this will have little impact on our numerical results. For a 2D square lattice Ising model, we will group together $L^2$ micro-steps as defined above into a single *iteration*, during which we will accumulate our binary observables $\langle \delta_{uv} \rangle_1$, mostly implicit zeros. As such, we 'always measure' the system and will never give away any information. Such an iteration will have a computational complexity comparable to a sweep in standard algorithms, with the advantage of having $L^2$ measurements per sweep as opposed to just one per sweep.

For each of our simulations we have performed $10^6$ iterations/sweeps, allowing the system to reach equilibrium for the first 10% of these and taking statistics afterwards. Each simulation was run with a 2-dimensional square lattice of side $L$ with periodic (toroidal) boundary conditions.

## 2.4 Numerical Results & Observations

### 2.4.1 Magnetic Susceptibility

In figures 2.4 and 2.5, we see how as we increase lattice size, plots of the susceptibility, $\chi$, approach the analytic singularity solution at the critical temperature. The steepness of the peak at $\beta_c = \frac{1}{2}\ln(1+\sqrt{2}) \approx 0.44$ displays the phase transition point as expected, fully agreeing with traditional spin flip simulations. Of note also is the asymptotic levelling off (consistent with spin flip methods) of $\chi$ for all lattice sizes for $\beta > \beta_c$ as we enter the broken phase where metropolis updates are nearly all rejected due to the low temperature (eq. (2.18)).

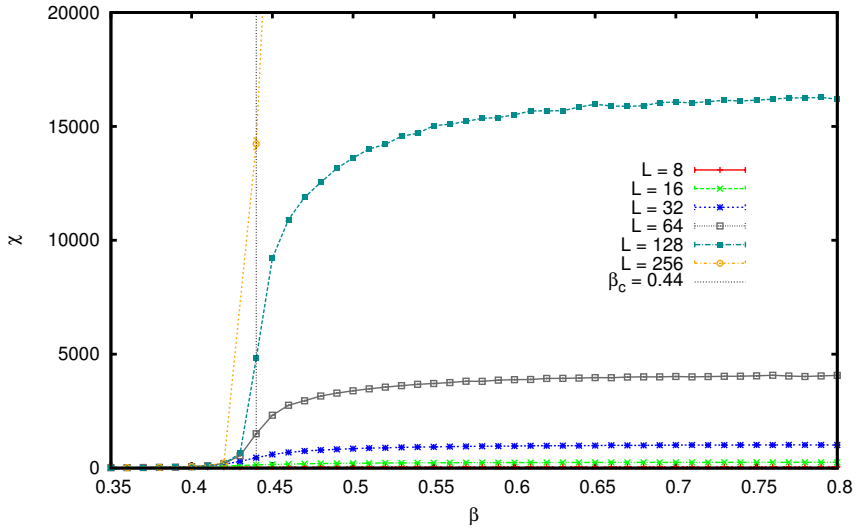



Figure 2.4: Susceptibility ($\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$

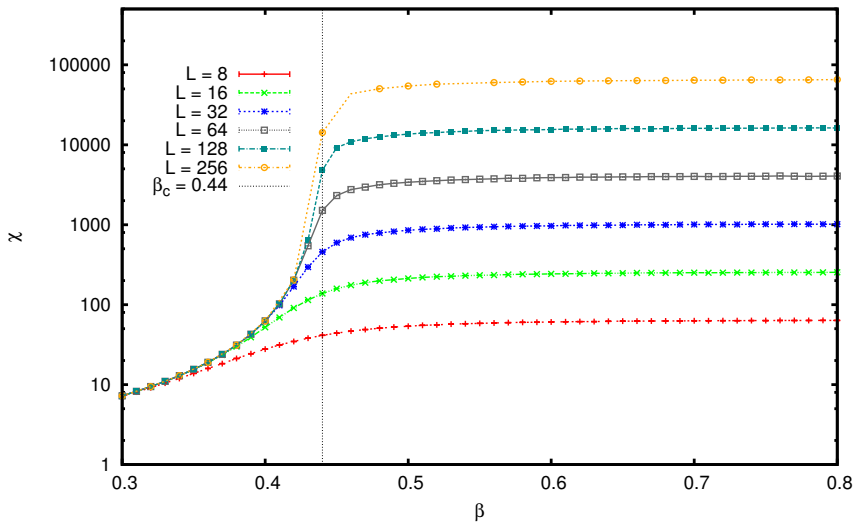



Figure 2.5: Susceptibility ($\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$ (Logscale)

We can then compare our values for $L^{\frac{7}{4}}/\chi$ with those found in [9], as confirmation that the

code is indeed doing what we want it to do and that our method is sound (table 2.1). When we plot $L^{\frac{7}{4}}/\chi$ for increasing $\beta$, we again see how, at the critical temperature, there is a clear phase transition that approaches the typical analytic discontinuity with increasing lattice size $L$.

| L | $L^{\frac{7}{4}}/\chi$ | $\tau_{\text{int},\chi}$ |
|---|---|---|
| 8 | 0.9202(24) | 0.703(23) |
| 16 | 0.9270(29) | 0.738(25) |
| 32 | 0.9420(35) | 0.777(28) |
| 64 | 0.9589(41) | 0.799(31) |
| 128 | 1.0114(50) | 1.019(42) |
| 256 | 1.1509(64) | 1.172(59) |

Table 2.1: $L^{\frac{7}{4}}/\chi$ with autocorrelation times - Single Worm

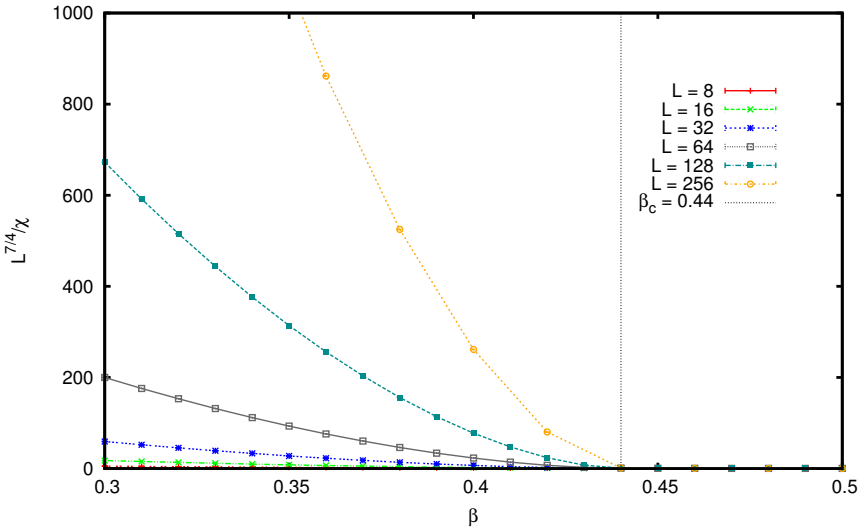



Figure 2.6: Susceptibility ($L^{\frac{7}{4}}/\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$

### 2.4.2 Autocorrelations

Now that we've been able to reproduce typical magnetic susceptibility values and plots as a concrete test of the method, we can now look at what is arguably the most interesting attribute of the single worm algorithm. If we look at how the (integrated) autocorrelation times, $\tau_{\text{int}}$, scale with increasing lattice size at critical temperature $\beta_c$ (figure 2.8), we see that $\tau_{\text{int}}$ scales as $L^{\xi}$ with dynamical critical exponent $\xi = 0.00146 \pm 0.00009$. What this indicates is that autocorrelations for the single worm algorithm essentially do not scale as lattice size is increased, obviously this is a vast improvement over the widely accepted value of $\xi = 2.125$ for usual spin flip methods [24]. We see what is fundamentally a complete absence of critical slowing down for the observable $\chi$.

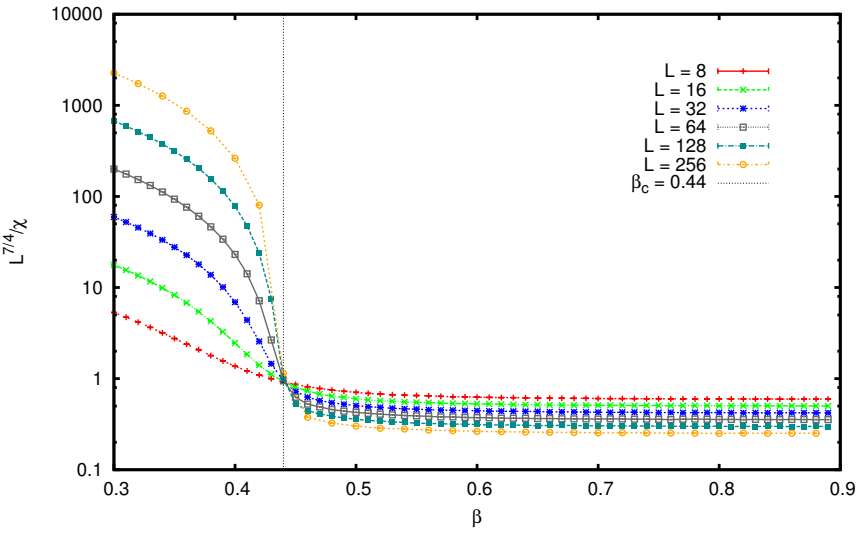



Figure 2.7: Susceptibility ($L^{\frac{7}{4}}/\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$ (Logscale)

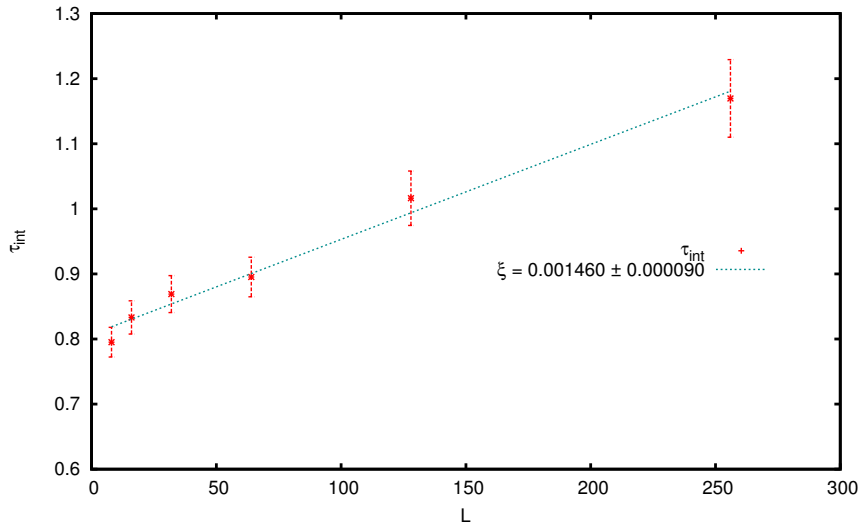



Figure 2.8: Integrated autocorrelation times for $L = \{16, \ldots, 256\}$

### 2.4.3 Conclusions & next steps

What we have shown so far is a reproduction of typical susceptibility results for Prokof'ev and Svistunov's Worm algorithm for the 2D Ising model with no biasing. We have also demonstrated the complete absence of critical slowing down for autocorrelations of the magnetic susceptibility. While there are many physically interesting systems and observables we could then move onto simulating, most of these examples have already been well documented by others [4, 5, 8, 9]. Instead, we are now motivated to investigate further the effect that *finite size scaling* has on the method by considering intersections of Binder cumulants [25], a common technique in these types of simulations. However, as these quantities depend on higher powers of average spin than are directly retrievable from the current formulation, we must now further expand the algorithm to allow the direct calculation of such estimators.

# Chapter 3

# Extending the Algorithm

## 3.1 Motivation

While the Worm algorithm can at first seem like just an efficient and somewhat novel simulation tool, the real power of the method lies in the reformulated configuration space in which it operates. It provides direct Monte Carlo estimators for many quantities [20] that are not available in the standard site representation. We saw with the standard single worm algorithm how through simply keeping track of coincidences of our two worm points we can quite efficiently measure magnetic susceptibility. We then however encounter a limitation of the model: when we try to derive worm observables for higher order spin observables (such as $\langle \bar{\sigma}^4 \rangle$) we immediately see that such quantities are simply not retrievable from the single worm reformulation.

This limitation leads to the obvious question of how can the method be adapted in order to calculate such higher order spin observables? In what we believe to be an original extension to the algorithm, we will attempt to further reformulate the Ising model as to directly retrieve these observables. The hope is that, by *adding a second worm* to the system, we will be able to calculate $4^{\text{th}}$-order Binder Cumulants as an alternate method to determine the critical point while greatly reducing finite size effects.

## 3.2 Adding a second worm to the 2D Ising Model

As an extension to the worm algorithm, we should be able to add a second "worm" to the system by adding two more lattice sites $w, z$ to the set of points we keep track of with each update:

$$\{u, v\} \to \{u, v, w, z\}.$$

To study this new system we will need to devise a new partition function that includes contributions from the second worm.
As usual, we will use the standard Boltzmann partition function,

$$Z_B = \sum_{\{\sigma\}} e^{-\beta E(\sigma)}, \tag{3.1}$$

when calculating canonical thermodynamic observables (Energy, Magnetic Susceptibility, etc.). We had the partition function for a single worm system,

$$\mathcal{Z}_1 = \sum_{u,v} \sum_{\{\sigma\}} \sigma_u \sigma_v \frac{e^{-\beta E(\sigma)}}{\rho\,(u,v)}, \tag{3.2}$$

from which we will be able to extrapolate a partition function for the two worm system.
The natural extension of our partition function to a two worm system is then

$$\mathcal{Z}_2 = \sum_{\substack{u,v \\ w,z}} \sum_{\{\sigma\}} \sigma_u \sigma_v \sigma_w \sigma_z \frac{e^{-\beta E(\sigma)}}{\rho\,(u,v,w,z)}, \tag{3.3}$$

where we now keep track of four worm ends instead of two. This new partition function will allow us to calculate observables with respect to the two worm system, denoted $\langle \cdots \rangle_2$.
Such observables will then be given by

$$\langle A \rangle_2 = \frac{1}{\mathcal{Z}_2} \sum_{\substack{u,v \\ w,z}} \sum_{\{\sigma\}} \sigma_u \sigma_v \sigma_w \sigma_z A \frac{e^{-\beta E(\sigma)}}{\rho\,(u,v,w,z)}. \tag{3.4}$$

## 3.3 Binder Cumulant & Two Worm Observables

A frequently used method to determine critical points of phase transitions in various physical systems is to use the intersection points of *Binder Cumulants* [25]. The most important advantage of the Binder Cumulant method is that finite size effects are much reduced. The 4th order Binder Cumulant $U_L$ is defined as

$$U_L = 1 - \frac{\langle \bar{\sigma}^4 \rangle}{3\langle \bar{\sigma}^2 \rangle^2}, \tag{3.5}$$

where $\bar{\sigma}$ is the average spin given by $\bar{\sigma} = \frac{1}{L^2} \sum_i \sigma_i$ for the 2-dimensional square lattice Ising model.

Typically, $U_L$ behaves as follows:

- In the symmetric phase ($\beta < \beta_c$): $U_L = 0 + \mathcal{O}\left(\frac{1}{V}\right)$ as $L \to \infty$
- In the broken phase ($\beta > \beta_c$): $U_L = \frac{2}{3} + \mathcal{O}\left(\frac{1}{V}\right)$ as $L \to \infty$

- At the critical point ($\beta_c$): $U_L \to U_L^*$ where $0 < U_L^* < \frac{2}{3}$

Locating the critical point using $U_L$ is then very easy. Using various lattice volumes, $L^d$, $U_L$'s must be calculated as functions of $\beta$. Then, the intersection points where $U_L(\beta)$ curves cross will give the critical point $\beta_c$. Usually, it is helpful to find the crossings using ascending *pairs* of volumes $\left(\frac{L_1}{L_2}, \frac{L_2}{L_3}, \dots \text{ where } L_1 < L_2 < L_3 < \cdots\right)$ (figures 3.12, 3.13).

So in order to calculate these Binder Cumulants for our two worm Ising Model, we will need to relate $\frac{\langle\bar{\sigma}^4\rangle}{\langle\bar{\sigma}^2\rangle^2}$ to observables that we can directly retrieve from the reformulated model. Much like in the single worm case where the susceptibility is given by

$$\langle\chi\rangle = \frac{1}{L^2\langle\delta_{uv}\rangle_1},$$

we can see how similar observables can be used to give us the desired relation for $U_L$.

In order to maintain statistical consistency, we want all of our observables in the Binder Cumulant to be retrievable from a single simulation of the system. To do this, we will consider a combination of observables that suit our purposes. As such, we will again consider worm end coincidences in the same way as with the single worm system.
We begin by explicitly writing the expectation value of each worm becoming a closed loop at the same time, $\langle\delta_{uv}\delta_{wz}\rangle_2$, and then re-expressing it in terms of average spin $\bar{\sigma}$. We start with

$$\langle\delta_{uv}\delta_{wz}\rangle_2 = \frac{1}{\mathcal{Z}_2}\sum_{\substack{u,v\\w,z}}\sum_{\{\sigma\}}\delta_{uv}\delta_{wz}\sigma_u\sigma_v\sigma_w\sigma_z e^{-\beta E(\sigma)},$$

sum over $u$ and $v$

$$\begin{aligned}\langle\delta_{uv}\delta_{wz}\rangle_2 &= \frac{1}{\mathcal{Z}_2}\sum_{\substack{v,w\\z}}\sum_{\{\sigma\}}\delta_{wz}\sigma_v\sigma_v\sigma_w\sigma_z e^{-\beta E(\sigma)},\\ &= \frac{L^2}{\mathcal{Z}_2}\sum_{w,z}\sum_{\{\sigma\}}\delta_{wz}\sigma_w\sigma_z e^{-\beta E(\sigma)},\end{aligned}$$

and then over $w$ and $z$

$$\begin{aligned}&= \frac{L^2}{\mathcal{Z}_2}\sum_{z}\sum_{\{\sigma\}}\sigma_z\sigma_z e^{-\beta E(\sigma)},\\ &= \frac{L^4}{\mathcal{Z}_2}\sum_{\{\sigma\}}e^{-\beta E(\sigma)},\end{aligned}$$

which can now be written using the expression for the average spin in 2D, $\bar{\sigma} = \frac{1}{L^2}\sum_i \sigma_i$, as

$$
\begin{aligned}
&= \frac{L^4 \sum\limits_{\{\sigma\}} e^{-\beta E(\sigma)}}{\sum\limits_{\substack{u,v\\w,z}} \sum\limits_{\{\sigma\}} \sigma_u \sigma_v \sigma_w \sigma_z e^{-\beta E(\sigma)}}, \\
&= \frac{L^4 \sum\limits_{\{\sigma\}} e^{-\beta E(\sigma)}}{(L^2)^4 \sum\limits_{\{\sigma\}} (\bar{\sigma})^4 e^{-\beta E(\sigma)}}, \\
&= \frac{1}{L^4 \langle \bar{\sigma}^4 \rangle},
\end{aligned}
$$

giving

$$
\langle \bar{\sigma}^4 \rangle = \frac{1}{L^4 \langle \delta_{uv} \delta_{wz} \rangle_2}. \tag{3.6}
$$

Now that we have an expression for $\langle \bar{\sigma}^4 \rangle_2$, we still need to calculate $\langle \bar{\sigma}^2 \rangle_2$. We can do this in much the same way, calculating now the expectation value for **any two** of the four endpoints coinciding, $\langle \delta_{uv} \rangle_2$.

$$
\langle \delta_{uv} \rangle_2 = \frac{1}{\mathcal{Z}_2} \sum_{\substack{u,v\\w,z}} \sum_{\{\sigma\}} \delta_{uv} \sigma_u \sigma_v \sigma_w \sigma_z e^{-\beta E(\sigma)},
$$

again summing over $u$ and $v$

$$
\begin{aligned}
\langle \delta_{uv} \rangle_2 &= \frac{1}{\mathcal{Z}_2} \sum_{\substack{v,w\\z}} \sum_{\{\sigma\}} \sigma_v \sigma_v \sigma_w \sigma_z e^{-\beta E(\sigma)}, \\
&= \frac{L^2}{\mathcal{Z}_2} \sum_{w,z} \sum_{\{\sigma\}} \sigma_w \sigma_z e^{-\beta E(\sigma)},
\end{aligned}
$$

and now rewriting in terms of average spin

$$
\begin{aligned}
&= \frac{L^2 \sum_{w,z} \sum_{\{\sigma\}} \sigma_w \sigma_z e^{-\beta E(\sigma)}}{\sum_{\substack{u,v \\ w,z}} \sum_{\{\sigma\}} \sigma_u \sigma_v \sigma_w \sigma_z e^{-\beta E(\sigma)}}, \\
&= \frac{L^2 \left(L^2\right)^2 \sum_{\{\sigma\}} \left(\bar{\sigma}\right)^2 e^{-\beta E(\sigma)}}{\left(L^2\right)^4 \sum_{\{\sigma\}} \left(\bar{\sigma}\right)^4 e^{-\beta E(\sigma)}}, \\
&= \frac{\sum_{\{\sigma\}} \left(\bar{\sigma}\right)^2 e^{-\beta E(\sigma)}}{L^2 \sum_{\{\sigma\}} \left(\bar{\sigma}\right)^4 e^{-\beta E(\sigma)}}, \\
&= \frac{\langle \bar{\sigma}^2 \rangle}{L^2 \langle \bar{\sigma}^4 \rangle},
\end{aligned}
$$

giving us

$$\langle \bar{\sigma}^2 \rangle = L^2 \langle \delta_{uv} \rangle_2 \langle \bar{\sigma}^4 \rangle. \tag{3.7}$$

Finally, combining (3.6) and (3.7), we have

$$\frac{\langle \bar{\sigma}^4 \rangle}{\langle \bar{\sigma}^2 \rangle^2} = \frac{\langle \delta_{uv} \delta_{wz} \rangle_2}{\left(\langle \delta_{uv} \rangle_2\right)^2}, \tag{3.8}$$

so that the Binder Cumulant is retrievable from our two worm system using

$$U_L = 1 - \frac{\langle \delta_{uv} \delta_{wz} \rangle_2}{3\left(\langle \delta_{uv} \rangle_2\right)^2}. \tag{3.9}$$

### 3.3.1 Susceptibility with two worms

As a test of our new two worm algorithm, we will again use the magnetic susceptibility of the system for comparison both to our own single worm simulations and to existing literature [9]. We had, for the zero field Ising model, the susceptibility

$$\langle \chi \rangle \equiv \langle \bar{\sigma}^2 \rangle,$$

which we can express in terms of the two worm observables derived above. Subbing (3.6) into (3.7), we get

$$
\begin{aligned}
\langle \bar{\sigma}^2 \rangle &= L^2 \langle \delta_{uv} \rangle_2 \left( \frac{1}{L^4 \langle \delta_{uv} \delta_{wz} \rangle_2} \right), \\
&= \frac{\langle \delta_{uv} \rangle_2}{L^2 \langle \delta_{uv} \delta_{wz} \rangle_2},
\end{aligned}
$$

giving us an expression for $\chi$ in terms of two worm observables

$$\langle \chi \rangle = \frac{\langle \delta_{uv} \rangle_2}{L^2 \langle \delta_{uv} \delta_{wz} \rangle_2}. \tag{3.10}$$

## 3.4 Designing an Algorithm

In the single worm model, the algorithm itself has been previously defined both by its creators [4] and in even more explicit detail by many others [9] so that we can assume most of the usual conditions that such a MCMC method must hold (detailed balance, ergodicity, etc.). As we are now in uncharted territory however, when designing our numerical strategies we have the added complication of ensuring that our methods satisfy detailed balance, etc. while also wanting to reproduce results from the single worm method with sufficient accuracy.

### 3.4.1 Move Update

First we will consider the **Move** update for two worms as this is the main part of the algorithm. As mentioned when we discussed the single worm algorithm, the move update alone is enough to simulate the system to similar levels of accuracy with the absence of a kick update only slightly increasing autocorrelation times. Here, as we are now dealing with four distinct worm end points, the system is no longer translationally invariant and we must now move each of the worm ends. The order in which we move the ends will not matter when ensuring ergodicity and detailed balance so long as each end point is moved on average the same number of times. However, we will see how the autocorrelation times for moving a random worm end will be slightly lower than if we move the ends sequentially (figure 3.8).

Our two worm move update is then much the same as the single worm update:

- **Move:** Pick one of the four worm ends, $\{u, v, w, z\}$, at random (say $z$ for example) as the current 'active end'. Then, with equal probability, propose one of $z$'s nearest neighbours as the destination site denoting it $z'$. The proposed move is then $z \to z'$ with simultaneous adjustment of the link/dimer between them $k_l \to 1 - k_l$ and is accepted with the same metropolis probability:

  $$p_{\text{acc}} = \min\left(1, \frac{\rho(u, v, w, z)}{\rho(u, v, w, z')} e^{\mu(2k_l - 1)}\right),$$

  otherwise the previous configuration is maintained.

Our move step is then essentially the same as for the single worm model and as such we know it must both be ergodic and satisfy detailed balance for all lattice configurations. Again, we know

that the update will require O(1) operations, regardless of the lattice size. We will again group $L^2$ updates into a single iteration/sweep of the algorithm.

Again, any two configurations will be connected with non-zero probability in the same way as with the single worm by the four end points 'dismantling' all active links of the first configuration and then 're-building' the second one over successive updates.

### Detailed Balance

As a quick sanity check, we can consider quick examples of worm configurations separated by a single move update and check that detailed balance is satisfied. Consider the following **drawing/erasing** move updates:

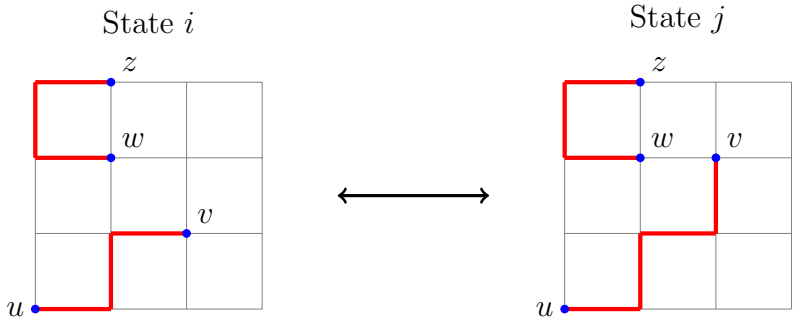



Figure 3.1: Detailed Balance: **Move** Update for two worms

The transition between these states must then satisfy the detailed balance equation (1.8):

$$\pi_i p_{ij} = \pi_j p_{ji},$$

where $p_{ij} \equiv p_{i \leftarrow j}$ is the probability of going from state $j$ to $i$. Since we have a stationary distribution $\pi_i = \pi_j$, we compute $p_{ij}$ and $p_{ji}$:

$$p_{ij} = \underbrace{\frac{1}{4}}_{\substack{\text{pick end} \\ \text{to move}}} \times \underbrace{\frac{1}{4}}_{\substack{\text{pick} \\ \text{destination}}} \times p_{\text{acc}},$$

$$p_{ji} = \underbrace{\frac{1}{4}}_{\substack{\text{pick end} \\ \text{to move}}} \times \underbrace{\frac{1}{4}}_{\substack{\text{pick} \\ \text{destination}}} \times p_{\text{acc}},$$

$$\Rightarrow p_{ij} = p_{ji}.$$

While these are only elementary examples for two configurations connected by a single move update, we can easily convince ourselves that by combining a chain of these drawing and erasing updates we will maintain detailed balance between any two configurations separated by an arbitrary number of updates.

### 3.4.2 Kick Update

Unlike the move update, constructing an ergodic kick update is not as simple as extending the single worm update to include two more potential 'active sites'. We run into problems in satisfying detailed balance with more than two worm ends. By considering various transitions between lattice configurations we can quickly see how simply picking a coincident pair of worm ends with equal probability (if there is more than one) and performing the single worm kick update on this chosen pair will **not** satisfy detailed balance.

#### Failing Detailed Balance

If we consider detailed balance following the single worm kick update probabilities for the following example, we will see how detailed balance is not satisfied and that we must consider an alternative update scheme.

Take the kick update $(v, w) \leftrightarrow (v', w')$:

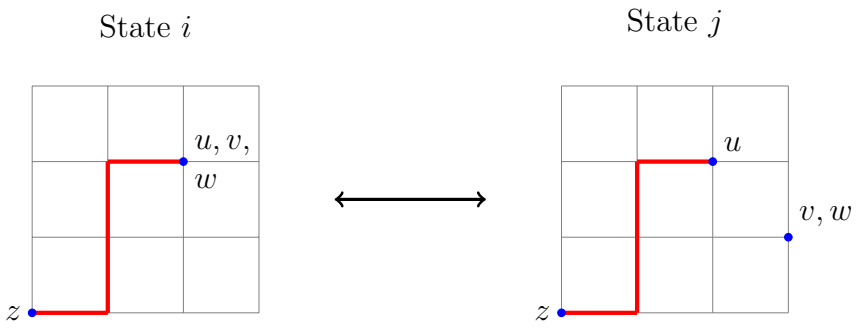



Figure 3.2: Detailed Balance: **Kick** Update for two worms

The update probabilities are:

$$p_{ij} = \underbrace{\frac{1}{1}}_{\substack{\text{Pick pair} \\ \text{to move}}} \times \underbrace{\frac{1}{L^2}}_{\substack{\text{Pick} \\ \text{destination}}} \times p_{\text{kick}},$$

$$p_{ji} = \underbrace{\frac{1}{3}}_{\substack{\text{Pick pair} \\ \text{to move}}} \times \underbrace{\frac{1}{L^2}}_{\substack{\text{Pick} \\ \text{destination}}} \times p_{\text{kick}},$$

$$\Rightarrow p_{ij} \neq p_{ji}.$$

Detailed balance then is not satisfied for this version of the kick update. It is clear that in order to have any chance of designing an update that satisfies detailed balance, we will need to factor in the arrangement of coincident pairs both in the initial state along with the destination state.

**Kick Update Attempts**

In our attempts to construct an ergodic kick update for the two worm algorithm that satisfies detailed balance for all possible transitions, we have investigated many iterations of a similar update scheme. For each such update, we were careful to maintain the following properties:

- Between each measurement of any observables, we perform a compound update consisting of a Metropolis **move** followed by a **kick**.
- Each sub-update must preserve the equilibrium distribution. The easiest way to check this is to show that the sub-update satisfies detailed balance with that distribution.
- Each of these sub-updates need **not** be ergodic on their own. In fact, **none** of our sub-updates will be ergodic on their own. This is common, for example, in the standard Ising spin flip algorithm with spin flip updates flipping one particular spin over and over again is certainly not ergodic.
- Furthermore, the compound updates need not satisfy detailed balance as each sub-update will maintain the equilibrium distribution.

The order in which these sub-updates are applied doesn't matter, but it is certainly far simpler to ensure detailed balance is satisfied if the kick update doesn't have any dependence on the move update that it follows.

The crux of the potential kick update is in configuring the acceptance probabilities for transitions between certain configuration as to always satisfy detailed balance. While the time constraints imposed by this project have prevented us from fully constructing and testing a consistent kick method, we are confident that such an update can be formed by considering a kick probability, $p_{\text{kick}}(i \leftarrow j)$, that contains explicit dependence on the nature of configurations $i$ and $j$.

### 3.4.3 Simulation Notes

Much the same as in our considerations of the single worm algorithm, we will set $\rho(x) \equiv 1$ for all $x$ for comparative purposes. As we do not now have a consistent kick update step, we will run the two worm simulations with $L^2$ single move updates making a single *iteration*, during which we will accumulate our (no longer binary) observables $\langle \delta_{uv} \rangle_2$ and $\langle \delta_{uv}\delta_{wz} \rangle_2$, again mostly implicit zeros. For each simulation we have performed $10^6$ iterations/sweeps, allowing the system to reach equilibrium for the first 10% of these and taking statistics afterwards. Each simulation was run on a 2-dimensional square lattice of side $L$ with periodic (toroidal) boundary conditions.

## 3.5 Numerical Results & Observations

### 3.5.1 Magnetic Susceptibility

Using our newly derived spin observables in terms of easily retrievable worm observables (eqs. (3.6) and (3.7)) we were able to compute susceptibilities and produce the same plots (figs. 3.3 to 3.6) of $\chi$ and $L^{\frac{7}{4}}/\chi$ against $\beta$ for different lattice sizes as we have for the single worm algorithm. At a glance the plots are largely the same for each algorithm although with the two worm algorithm, once we enter the broken phase ($\beta > \beta_c$) $\chi$ oscillates erratically around the single worm values for larger lattice sizes. We can attribute this behaviour again to the difficulty in having any of our metropolis updates accepted for low temperatures (high $\beta$).

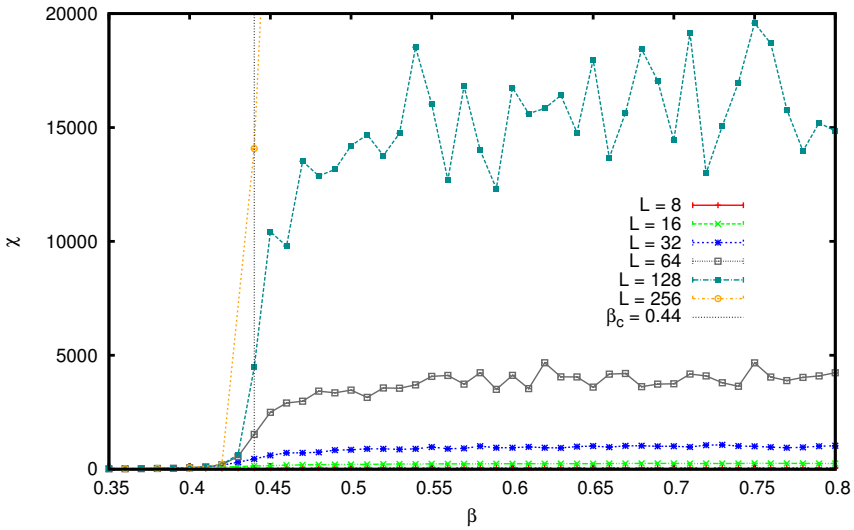



Figure 3.3: Susceptibility ($\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$

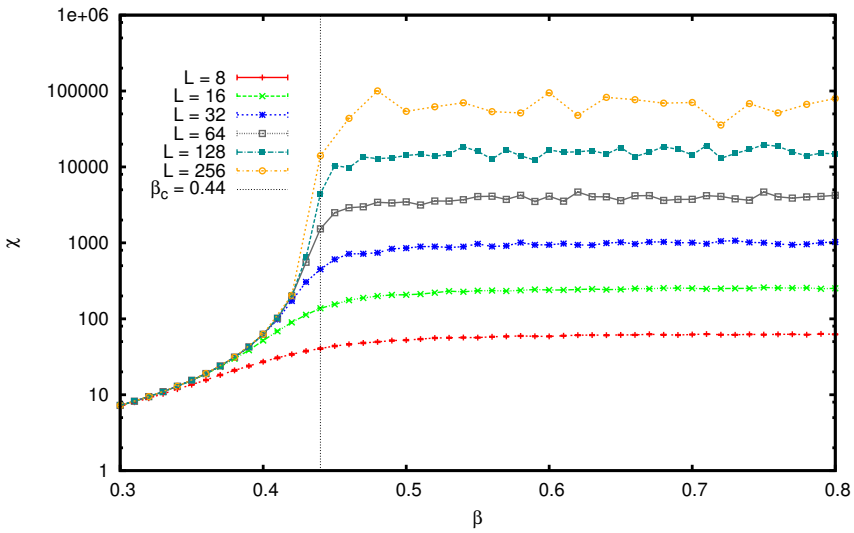



Figure 3.4: Susceptibility ($\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$ (Logscale)

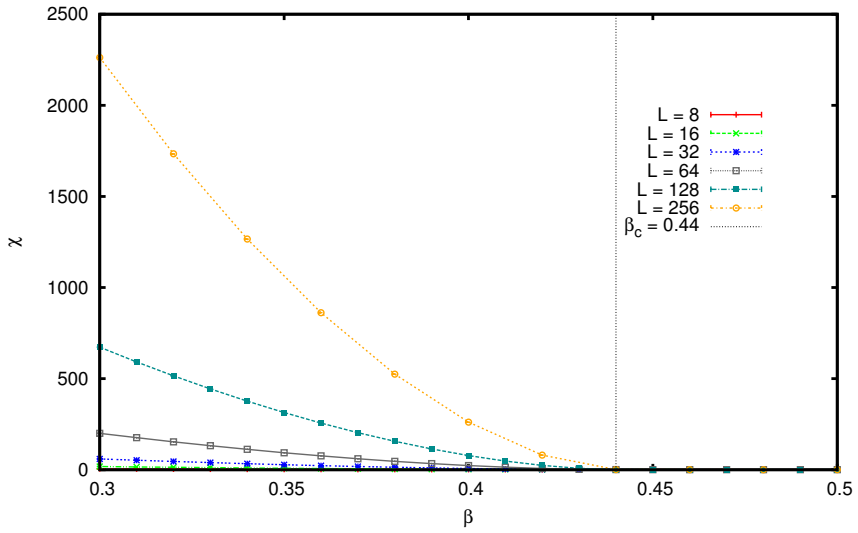



Figure 3.5: Susceptibility ($L^{\frac{7}{4}}/\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$

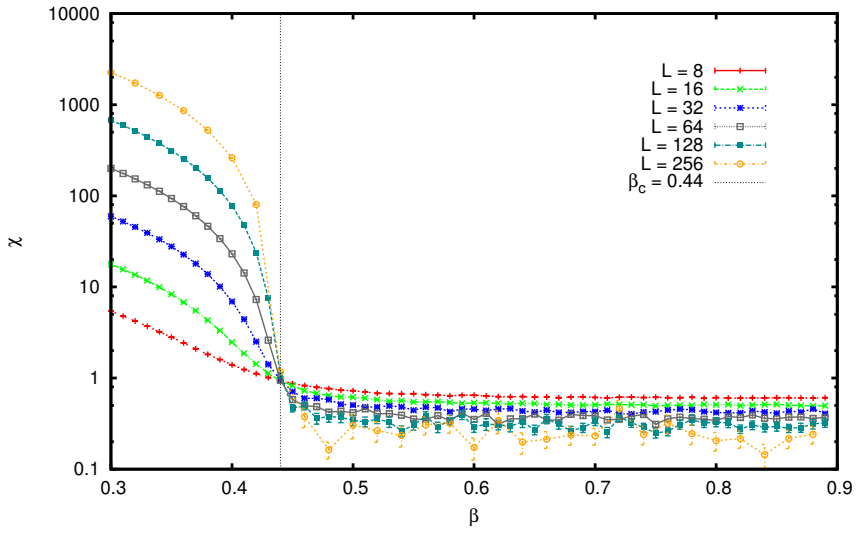



Figure 3.6: Susceptibility ($L^{\frac{7}{4}}/\chi$) against $\beta$ for $L = \{16, \ldots, 256\}$ (Logscale)

In fig. 3.7 we have plotted the susceptibility, $\chi$, against $\beta$ comparing the original single worm algorithm to our expanded two worm algorithm. For small lattices, the two methods give largely the same results and as $L$ is increased the susceptibility from two worm method starts to oscillate wildly about the single worm solution above $\beta_c$. Again, as this is well into the broken phase this shouldn't be any cause of concern for the validity of the algorithm. Since the duality transformation to our worm configuration space is centred around a high-temperature (low $\beta$) expansion, it should come as no surprise that, away from this region of low $\beta$ and the critical point, the method becomes less reliable. Reassuringly, at the critical point, $\beta_c$, we recover the same results using each version of the algorithm (table 3.1) consistent with published data [9].

| L | $L^{\frac{7}{4}}/\chi$ | $\tau_{\text{int,two}}$ | $\tau_{\text{int,four}}$ |
|---|---|---|---|
| 8 | 0.9378(07) | 0.772(21) | 0.603(11) |
| 16 | 0.929(12) | 0.797(22) | 0.573(10) |
| 32 | 0.956(21) | 0.793(22) | 0.544(09) |
| 64 | 0.946(35) | 0.829(25) | 0.500(07) |
| 128 | 1.089(64) | 0.856(27) | 0.508(07) |
| 256 | 1.164(103) | 0.977(37) | 0.500(07) |

Table 3.1: $L^{\frac{7}{4}}/\chi$ with autocorrelation times for two and four coincidences - Two Worms

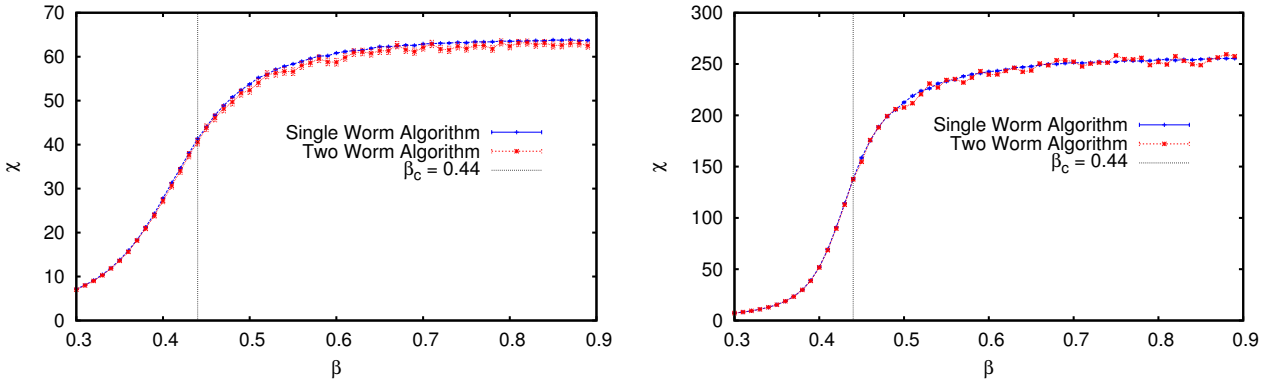
### 3.5.2 Autocorrelations

We again use the integrated autocorrelation time (eq. (1.11)) for measured observables as a measure of the efficiency of our algorithm. All standard errors again calculated using eq. (1.13). In the development of the move update for the two worm algorithm, rather than randomly choosing the worm end to be updated, we originally updated the worm ends sequentially. When comparing integrated autocorrelation times for our two primary observables (eqs. (3.6) and (3.7)) with a sequential update versus a randomly sequenced move update, $\tau_{\text{int}}$ for the random update is slightly lower for both of our observables (fig. 3.8). Therefore, we have employed a random update scheme for all of our two worm simulations.

#### Comparing the algorithms

Figure 3.9 shows $\tau_{\text{int}}$ for two worm ends coinciding, $\langle \delta_{uv} \rangle_2$, at the critical temperature for lattices of size $L = 16$ & 256. Immediately, we see that autocorrelation times have improved for the two worm method over the single worm algorithm and as the lattice size, $L$, is increased, the improvement over what is an already very efficient algorithm increases. Looking at table 3.1, we see how the autocorrelation time for finding all four worm ends coinciding ($\tau_{\text{int,four}}$) converges to the "perfect" value of 0.5 for large $L$. While we cannot compare this with the single worm algorithm, it is in itself an astounding result.

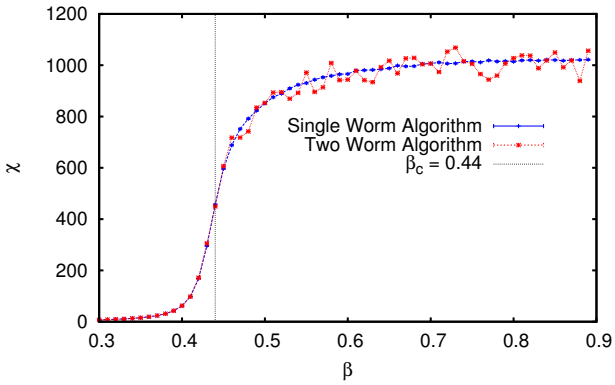
Furthermore, in fig. 3.10 we have plotted the maximum/asymptotic value for $\tau_{\text{int}}$ as a function of $L$ for both $\langle \delta_{uv} \rangle_2$ and $\langle \delta_{uv}\delta_{wz} \rangle_2$. This plot shows what is in many ways the biggest success of the single worm algorithm, further improved by our new two worm method. At $\beta_c$, we see that $\tau_{\text{int,two}}$ scales at $L^{\xi}$, much like with the single worm method but with dynamical critical exponent $\xi_{\text{two}} = 0.000780 \pm 0.000055$, an order of magnitude closer to zero than for the single worm algorithm. Again, we can take this to mean that $\tau_{\text{int}}$ simply does not scale as lattice size is increased. Similarly for $\tau_{\text{int,four}}$ we have $\xi_{\text{four}} = -0.000332 \pm 0.000157$, closer again to zero scaling. Again, we see a complete absence of critical slowing down for primary observables.



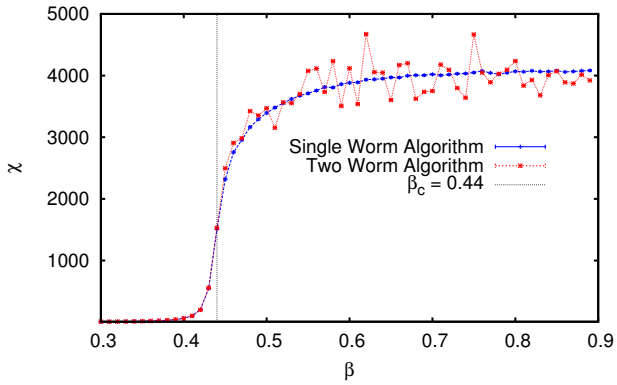

(a) $L = 8$



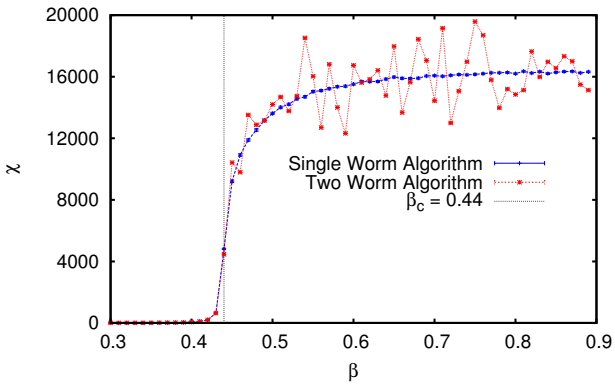

(b) $L = 16$



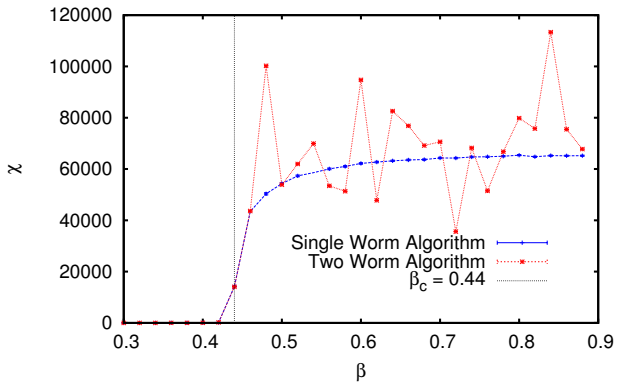

(c) $L = 32$




(d) $L = 64$




(e) $L = 128$




(f) $L = 256$

Figure 3.7: Comparison of $\chi$ for single worm vs two worm algorithms with various lattice sizes

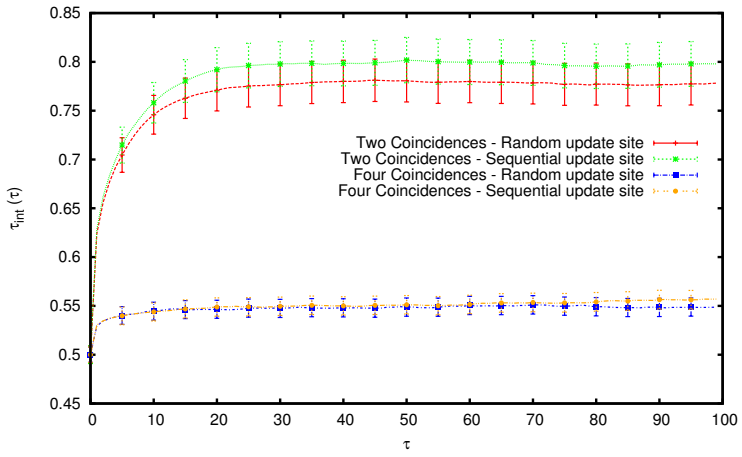



Figure 3.8: Integrated autocorrelation for sequential vs random site update selection at $\beta_c$

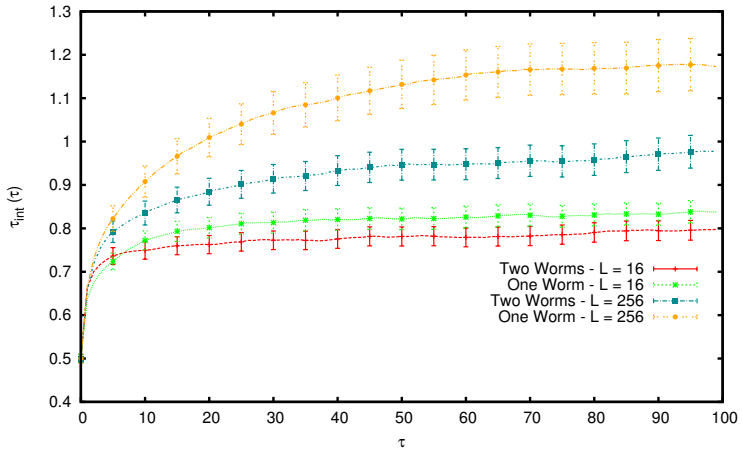



Figure 3.9: Integrated autocorrelation for two worm ends coinciding at $\beta_c$ for $L = 16, 256$ - One worm vs Two worm

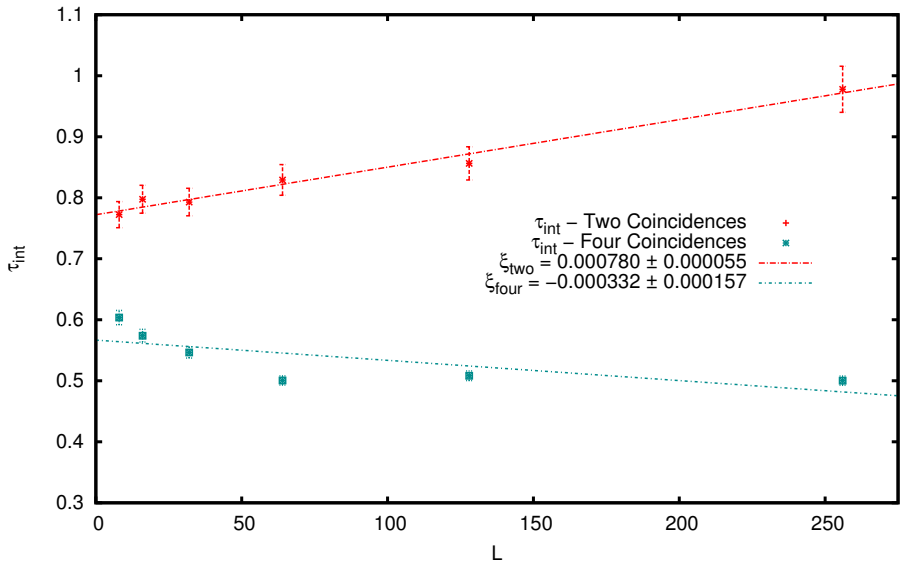



Figure 3.10: Integrated autocorrelation times for $L = \{16, \ldots, 256\}$ at $\beta_c$

### 3.5.3 Binder Cumulants

Finally, we have reached the ultimate goal set out in this project, calculating Binder Cumulants (eq. (3.5)) for the 2D Ising Model using worm algorithm methods. Using eq. (3.9) to relate the fourth-order Binder Cumulant, $U_L$, to our primary observables, $\langle \delta_{uv} \rangle_2$ and $\langle \delta_{uv}\delta_{wz} \rangle_2$, we have been able to calculate $U_L$ for various lattice sizes in an attempt to see how finite size effects manifest in the two worm algorithm. In fig. 3.11 we have plotted $U_L$ against $\beta$ for various lattice sizes which show, much like in our plots of $\chi$, how as $L$ increases, the slope of $U_L$ increases and approaches a sharp peak at the critical temperature/phase transition point $\beta_c$. As expected, in the broken phase ($\beta > \beta_c$) $U_L$ levels off around the theoretical value of $\frac{2}{3}$, behaving exactly how we expect.

| L | $U_L$ |
|---|---|
| 8 | 0.6080(29) |
| 16 | 0.6085(52) |
| 32 | 0.5979(89) |
| 64 | 0.608(17) |
| 128 | 0.562(26) |
| 256 | 0.564(47) |

Table 3.2: Fourth-Order Binder Cumulants ($U_L$) at the critical point $\beta_C$

The intersection points of Binder Cumulants for ascending volume pairs are often used to determine the critical point, $\beta_c$, often giving more accurate results than the maximum locations of magnetic susceptibility as finite size effects are greatly reduced. Figure 3.12 is a zoomed in view of the intersections of $U_L$ for different lattice sizes from which we can take estimates of the intersection points for ascending volume pairs $\left(\frac{L_1}{L_2}, \frac{L_2}{L_3}, \dots \text{ where } L_1 < L_2 < L_3 < \cdots\right)$. Then in fig. 3.13 we look at the temperature at which these intersections occur and plot them against $1/L_{\min}$ where $L_{\min}$ is the smaller lattice size of the coincident pair. Within our statistical errors there is no clear systematics for finite $L$, however the largest pair, $\frac{128}{256}$, gives a value for the critical temperature of $\beta = 0.440182(12)$, close to the known analytic value of $\beta_c = \frac{\ln(1+\sqrt{2})}{2} \approx 0.440686$. By running our simulations for more values of $\beta$ around the critical point, this number could potentially be improved upon as we have here only values for $U_L$ for $\beta = \{\dots, 0.43, 0.44, 0.45, \dots\}$.

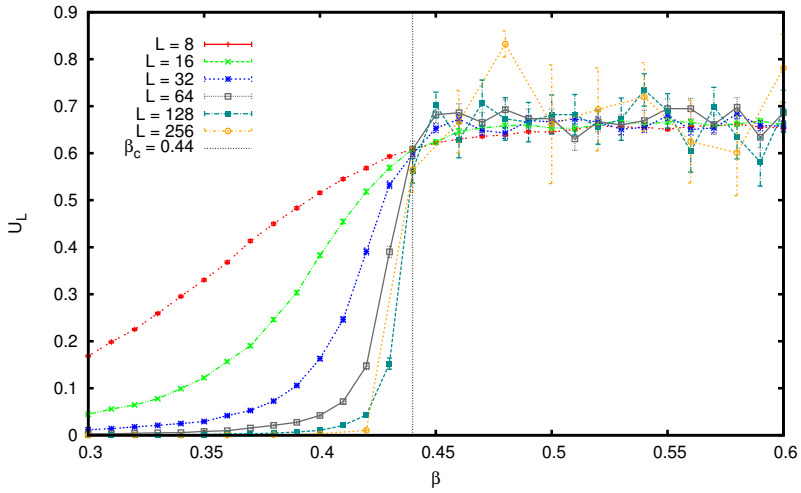



Figure 3.11: Binder Cumulants against $\beta$ for $L = \{16, \ldots, 256\}$

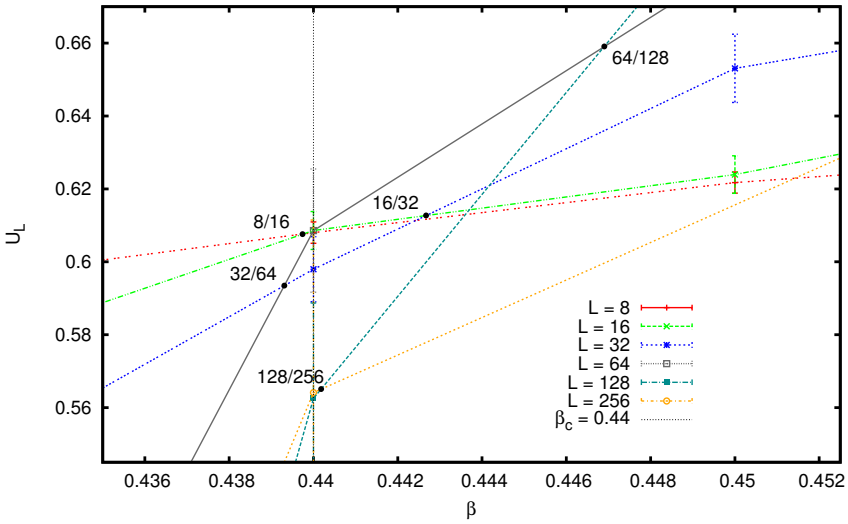



Figure 3.12: Binder Cumulant intersections for $L = \{16, \ldots, 256\}$

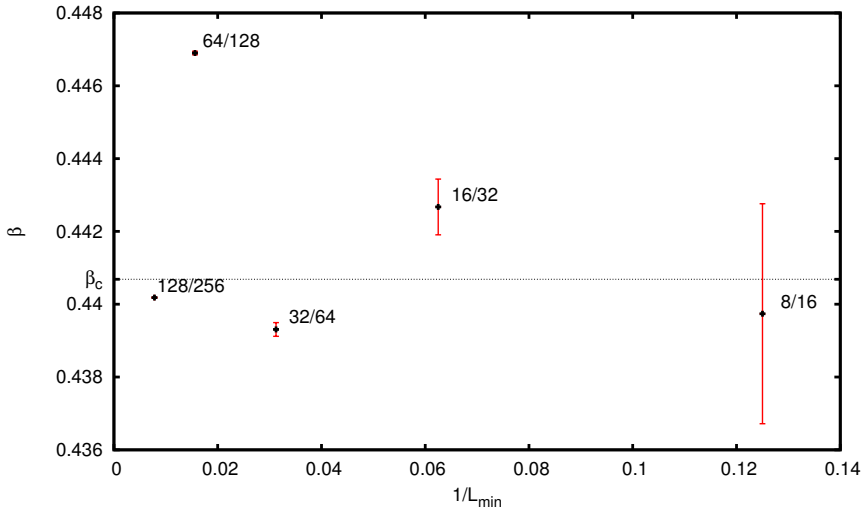



Figure 3.13: Binder Cumulant intersections for ascending volume pairs

# Chapter 4

# Conclusions

Our efforts during this project have all been with a single goal in mind; to expand upon Prokof'ev and Svistunov's worm algorithm with the aim of calculating higher order average spin observables so that we can examine finite size scaling using intersections of Binder Cumulants (eq. (3.5)). To achieve this we have first investigated the original *Single Worm* algorithm, including a detailed discussion of the duality transformation to the worm configuration space. We have detailed the development of an ergodic Monte Carlo Markov Chain (MCMC) algorithm update scheme and included numerical results for simulations of the 2D square lattice Ising model reproducing known results from published literature [4, 9].

The bulk of our work however has been in extending the original single worm method to what we call a *Two Worm* algorithm. The primary aim being to allow for the direct calculation of higher powers of average spin estimators with the intention of calculating Binder Cumulants using the method. As a relatively non-trivial extension to a new but well-studied algorithm, included is detailed discussion of the development of an update scheme that both is ergodic and satisfies detailed balance for transitions between all available configurations. We have derived expressions for the desired spin observables in terms of primary observables that we may directly measure in this two worm configuration space. Of note is the fact that while spin observables retrievable from both worm formulations have different expressions in terms of primary observables for the two methods, the results are the same.

In developing the two worm algorithm, it has been important and very useful to consider (integrated) autocorrelation times for our measured observables as a measure of the effectiveness of our scheme. We have used this metric in the development of our update strategy and as a comparison tool for the two worm algorithm against the original single worm algorithm. Much like the single worm method, our two worm algorithm shows what is essentially a complete absence of critical slowing for each of the primary observables we have considered. For spin observables

eqs. (3.6) and (3.7) we obtain dynamical critical exponents of $\xi_{\text{two}} = 0.000780 \pm 0.000055$ and $\xi_{\text{four}} = -0.000332 \pm 0.000157$ respectively, indicating an efficiency comparable to the best cluster models with no size scaling (scaling as $L^{\xi}$).

Finally, we have been able to calculate the fourth-order Binder Cumulant, $U_L$, for various lattice size, $L$, on the 2D square lattice Ising model. Such Binder Cumulant methods are traditionally used to avoid finite size effects when determining critical points/temperatures. Using our two worm algorithm, we have obtained the predicted theoretical limits for $U_L$ for both high- and low-$\beta$ and at the critical temperature $\beta_c$. By considering the intersection points of ascending volume pairs of Binder Cumulant curves we have also retrieved accurate estimates of the (known) critical temperature.

The next step (other than running simulations for higher $L$, $\beta$ resolution, etc.) to get this extension to a publishable state is then, as seen in [8] with the single worm method, to simulate the three dimensional Ising model for which there remains no analytic solution along with a more in depth analysis of autocorrelation scaling behaviour. Of interest is the performance of the two worm scheme in simulating the two-point function for the 3D Ising model, a calculation for which the single worm algorithm has been noted to be more efficient than the best cluster methods.

# Bibliography


[1] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. pp. 97–109, 1970.

[2] H. Yahata and M. Suzuki, "Critical slowing down in the kinetic ising model," *Journal of the Physical Society of Japan*, vol. 27, no. 6, pp. 1421–1438, 1969.

[3] N. Prokof'ev, B. Svistunov, and I. Tupitsyn, "Worm algorithm in quantum monte carlo simulations," *Physics Letters A*, vol. 238, no. 45, pp. 253 – 257, 1998.

[4] N. Prokof'ev and B. Svistunov, "Worm Algorithms for Classical Statistical Models," *Physical Review Letters*, vol. 87, p. 160601, Oct. 2001.

[5] U. Wolff, "Simulating the All-Order Strong Coupling Expansion III: O(N) sigma/loop models," *Nucl.Phys.*, vol. B824, pp. 254–272, 2010.

[6] U. Wolff, "Simulating the All-Order Strong Coupling Expansion IV: CP(N-1) as a loop model," *Nucl.Phys.*, vol. B832, pp. 520–537, 2010.

[7] T. Korzec and U. Wolff, "Simulating the All-Order Strong Coupling Expansion V: Ising Gauge Theory," *Nucl.Phys.*, vol. B871, pp. 145–163, 2013.

[8] Y. Deng, T. M. Garoni, and A. D. Sokal, "Dynamic Critical Behavior of the Worm Algorithm for the Ising Model," *Physical Review Letters*, vol. 99, p. 110601, Sept. 2007.

[9] U. Wolff, "Simulating the All-Order Strong Coupling Expansion I: Ising Model Demo," *Nucl.Phys.*, vol. B810, pp. 491–502, 2009.

[10] E. Schneidman, M. J. Berry, R. Segev, and W. Bialek, "Weak pairwise correlations imply strongly correlated network states in a neural population," *Nature*, vol. 440, no. 7087, pp. 1007–1012, 2006.

[11] R. Peierls, "On ising's model of ferromagnetism," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 32, pp. 477–481, 10 1936.

[12] L. Onsager, "Crystal statistics. i. a two-dimensional model with an order-disorder transition," *Phys. Rev.*, vol. 65, pp. 117–149, Feb 1944.

[13] B. Efron and C. Stein, "The jackknife estimate of variance," *The Annals of Statistics*, vol. 9, no. 3, pp. pp. 586–596, 1981.

[14] B. Efron, *The Jackknife, the Bootstrap, and Other Resampling Plans*. CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1982.

[15] R. H. Swendsen and J.-S. Wang, "Nonuniversal critical dynamics in monte carlo simulations," *Physical review letters*, vol. 58, no. 2, p. 86, 1987.

[16] U. Wolff, "Collective Monte Carlo updating for spin systems," *Physical Review Letters*, vol. 62, pp. 361–364, Jan. 1989.

[17] J. Machta, Y. S. Choi, A. Lucke, T. Schweizer, and L. V. Chayes, "Invaded cluster algorithm for equilibrium critical points," *Phys. Rev. Lett.*, vol. 75, pp. 2792–2795, Oct 1995.

[18] H. G. Evertz, G. Lana, and M. Marcu, "Cluster algorithm for vertex models," *Phys. Rev. Lett.*, vol. 70, pp. 875–879, Feb 1993.

[19] A. Barbu and S.-C. Zhu, "Generalizing swendsen–wang for image analysis," *Journal of Computational and Graphical Statistics*, vol. 16, no. 4, 2007.

[20] E. L. Pollock and D. M. Ceperley, "Path-integral computation of superfluid densities," *Phys. Rev. B*, vol. 36, pp. 8343–8352, Dec 1987.

[21] G. Parisi, *Statistical field theory*. Frontiers in Physics, Redwood City, CA: Addison-Wesley, 1988.

[22] F. J. Wegner, "Duality in generalized ising models and phase transitions without local order parameters," *Journal of Mathematical Physics*, vol. 12, no. 10, pp. 2259–2272, 1971.

[23] F. J. Wegner, "Duality in generalized Ising models." `http://arxiv.org/abs/1411.5815`, 2014. Accessed: 5 Jan 2015.

[24] J. K. Williams, "Monte carlo estimate of the dynamical critical exponent of the 2d kinetic ising model," *Journal of Physics A: Mathematical and General*, vol. 18, no. 1, p. 49, 1985.

[25] K. Binder, "Finite size scaling analysis of ising model block distribution functions," *Zeitschrift für Physik B Condensed Matter*, vol. 43, no. 2, pp. 119–140, 1981.