Licentiate Thesis

Examining Hausdorff dimension and Scaling behaviour with Worm algorithm

Simon Rydell

Theoretical Particle Physics, Department of Theoretical Physics,
School of Engineering Sciences
Royal Institute of Technology, SE-106 91 Stockholm, Sweden

Stockholm, Sweden 2018

Typeset in LaTeX

Akademisk avhandling för avläggande av teknologie licentiatexamen (TeknL) inom ämnesområdet teoretisk fysik.

Scientific thesis for the degree of Licentiate of Engineering (Lic Eng) in the subject area of Theoretical physics.

TRITA-FYS-2010:05

Printed in Sweden by Universitetsservice US AB, Stockholm May 2018

Abstract

Preface

Contents

Part I

Introduction and background material

Chapter 1

Introduction

Chapter 2

Background

2.1 Ising model

2.1.1 Loop expansion

The Ising model energy

$$E = -J \sum_{\langle ij \rangle} S_i S_j \tag{2.1}$$

Let $K = \beta J$ where $\beta = 1/k_{\mathrm{B}}T$.

$$\beta E = -K \sum_{\langle ij \rangle} S_i S_j \tag{2.2}$$

The partition function Z.

$$Z = \sum_{\text{all states}} e^{-\beta E} = \sum_{\text{all states}} e^{K \sum_{\langle ij \rangle} S_i S_j} = \sum_{\text{all states}} \Pi_{\langle ij \rangle} e^{K S_i S_j} \tag{2.3}$$

Since $S_i S_j = \pm 1$ in the Ising model, Euler identities can be used to expand the exponential in equation (2.3).

$$\begin{aligned} e^{K S_i S_j} &= \frac{e^K + e^{-K}}{2} + S_i S_j \frac{e^K - e^{-K}}{2} \\ &= \cosh(K) + S_i S_j \sinh(K) \\ &= \{T = \tanh(K)\} \\ &= (1 + T S_i S_j) \cosh(K) \end{aligned}$$

For N spins there are 2N bonds, therefore the partition function is

$$\begin{aligned} Z &= \sum_{\text{all states}} \Pi_{\langle ij \rangle} (1 + T S_i S_j) \cosh(K) \\ &= \cosh^{2N}(K) \cdot 2^N \left(2^{-N} \sum_{\text{all states}} \Pi_{\langle ij \rangle} (1 + T S_i S_j) \right) \\ &= \cosh^{2N}(K) \cdot 2^N Z' \end{aligned}$$

And

$$\begin{aligned} Z' &= 2^{-N} \sum_{\text{all states}} \Pi_{\langle ij \rangle} (1 + T S_i S_j) \\ &= 2^{-N} \sum_{S_1 = \pm 1} \sum_{S_2 = \pm 1} \dots \sum_{S_N = \pm 1} \left(1 + T \sum_{l=1} S_i S_j + T^2 \sum_{l=2} (S_i S_j)(S_{i'} S_{j'}) + \dots \right) \end{aligned}$$

Where the sums $\sum_{l=L}$ should be interpreted as the sum over all sets where the link length is L. Link length is the coupling between S-terms as can be seen in Figure (2.1).

Since $\sum_{S_i = \pm 1} S_i = 0$, only terms with an even number of S_i are contributing to Z'. Call these terms closed, indicating that they represent a closed loop. The sum over all contributing terms gives a factor of 2^N, canceling the 2^{-N}.

Rewrite Z' in terms of loop lengths.

$$Z' = \sum_L g(L) T^L \tag{2.4}$$

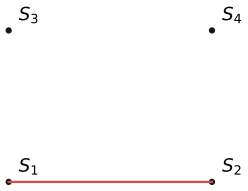
Where $g(L)$ is the number of loops with length L. Finally write the expression for the partition function.

$$Z = 2^N \cosh^{2N}(K) \sum_L g(L) T^L \tag{2.5}$$

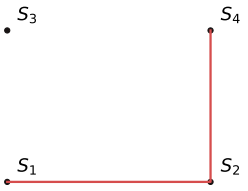
2.1.2 Energy calculation

$$E = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = -\frac{J}{Z} \frac{\partial Z}{\partial K} \tag{2.6}$$

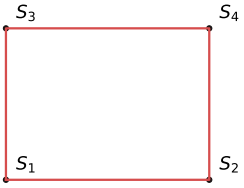
Therefore



(a) $(S_1S_2),\ L=1$



(b) $(S_1S_2)(S_2S_3),\ L=2$



(c) $(S_1S_2)(S_2S_3)(S_3S_4)(S_4S_1),\ L=4$

Figure 2.1: Link structure of an Ising lattice where (a) and (b) are open while (c) is closed.

$$\begin{aligned}\frac{\partial Z}{\partial K} &= 2^N 2N\cosh^{2N-1}(K)\cdot\frac{\partial\cosh(K)}{\partial K}Z' + 2^N\cosh^{2N}(K)\cdot\frac{\partial Z'}{\partial K}\\ &= 2^N\cosh^{2N}(K)\left(2N\tanh(K)Z' + \frac{\partial Z'}{\partial K}\right)\\ &= 2^N\cosh^{2N}(K)\tanh(K)\left(2NZ' + \frac{1}{\tanh(K)}\frac{\partial Z'}{\partial K}\right)\end{aligned}$$

Examine $\tanh^{-1}(K)\frac{\partial Z'}{\partial K}$.

$$\begin{aligned}\tanh^{-1}(K)\frac{\partial Z'}{\partial K} &= \tanh^{-1}(K)\frac{\tanh(K)}{\partial K}\sum_L g(L)L\tanh^{L-1}(K)\\ &= \frac{1}{\tanh^2(K)\cosh^2(K)}\sum_L g(L)L\tanh^L(K)\\ &= \frac{Z'}{\sinh^2(K)}\frac{\sum_L g(L)L\tanh^L(K)}{\sum_L g(L)\tanh^L(K)}\\ &= \frac{Z'}{\sinh^2(K)}\langle L\rangle\end{aligned}$$

And finally

$$E = -J\tanh(K)\left(2N + \frac{\langle L\rangle}{\sinh^2(K)}\right) \tag{2.7}$$

where

$$\langle L\rangle = \frac{\sum_L g(L)L\tanh^L(K)}{\sum_L g(L)\tanh^L(K)} \tag{2.8}$$

2.1.3 Heat capacity

$$C = \frac{\partial E}{\partial T} = -\beta^2\frac{\partial E}{\partial \beta} = -K\beta\frac{\partial E}{\partial K} \tag{2.9}$$

Let $A = 2N + \frac{1}{\sinh^2(K)}\langle L\rangle$. Then

$$\frac{E}{J} = -\tanh(K)A \tag{2.10}$$

and

$$\frac{1}{J}\frac{\partial E}{\partial K} = -\frac{\partial \tanh(K)}{\partial K}A - \tanh(K)\frac{\partial A}{\partial K} \tag{2.11}$$

where

$$\begin{aligned}\frac{\partial A}{\partial K} &= \langle L\rangle\frac{\partial \sinh^{-2}}{\partial K} + \tanh^{-1}(K)\sinh^{-2}(K)\left(\langle L^2\rangle - \langle L\rangle^2\right)\frac{\partial \tanh(K)}{\partial K}\\ &= \frac{1}{\sinh^2(K)\tanh(K)}\left(-2\langle L\rangle + \frac{\langle L^2\rangle - \langle L\rangle^2}{\cosh^2(K)}\right)\end{aligned}$$

and finally

$$C = \frac{K^2}{\sinh^2(K)}\left(\frac{\langle L^2\rangle - \langle L\rangle^2}{\cosh^2(K)} - E\tanh(K) - 2\langle L\rangle\right) \tag{2.12}$$

2.2 n-vector model

The n-vector model describes a classical system of n-dimensional classical spins s_i of unit length interacting on a lattice. It is a generalization of the Ising model where each spin can have a continuous set of values. The Hamiltonian is then

$$H = -J \sum_{\langle ij \rangle} s_i \cdot s_j \tag{2.13}$$

where J is the bond strength and $\langle ij \rangle$ refers to a nearest neighbour interaction.

2.3 XY model

A special case of the n-vector model is the XY model when $n = 2$. Here the spins are two dimensional rotors as $s_i = (\cos\theta_i,\ \sin\theta_i)$. This yields the Hamiltonian

$$H = -J \sum_{\langle ij \rangle} \cos(\theta_i - \theta_j) \tag{2.14}$$

The partition function is therefore

$$Z = \prod_i \int \frac{\mathrm{d}\theta_i}{2\pi} e^{-\beta H} = \prod_i \int \frac{\mathrm{d}\theta_i}{2\pi} e^{K \sum_{\langle ij \rangle} \cos(\theta_i - \theta_j)} \tag{2.15}$$

where $K = J\beta$.

2.3.1 Loop expansion

Since equation (2.15) is invariant under the transformation $\theta_i - \theta_j \to \theta_i - \theta_j + 2\pi n$, $n \in \mathbb{Z}$, it can be expanded using the identity

$$e^{\alpha \cos\beta} = \sum_{\gamma=-\infty}^{\infty} I_\gamma(\alpha) e^{i\gamma\beta} \tag{2.16}$$

where $I_\gamma(\alpha)$ is the modified Bessel function. Using that $e^{\sum_i x_i} = \prod_i e^{x_i}$

$$Z = \prod_i \int \frac{\mathrm{d}\theta_i}{2\pi} \sum_{J_{\langle ij \rangle}=-\infty}^{\infty} \prod_{b=\langle ij \rangle} I_{J_{\langle ij \rangle}}(K) e^{iJ_{\langle ij \rangle}(\theta_i - \theta_j)} \tag{2.17}$$

$$= \prod_i \sum_{J_b} \left(\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i - \theta_j)} \right) \left(\prod_b I_{J_b} \right) \tag{2.18}$$

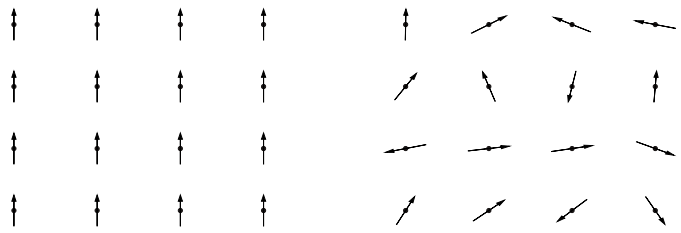
Where in the last step, $\prod_{\langle ij \rangle} e^{iJ_{\langle ij \rangle}(\theta_i - \theta_j)} = e^{iN_i(\theta_i - \theta_j)}$. N_i is therefore the sum of J for the nearest neighbours of site i. Noting that

$$\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i-\theta_j)} = C\delta_{N_i 0} \tag{2.19}$$

leads to the conclusion that the sum of incoming and outgoing flux J into a site i must be zero, in other words, the configurations are divergence free. This in turn means that a configuration of the system must contain closed loops of flux J.

2.3.2 Winding number

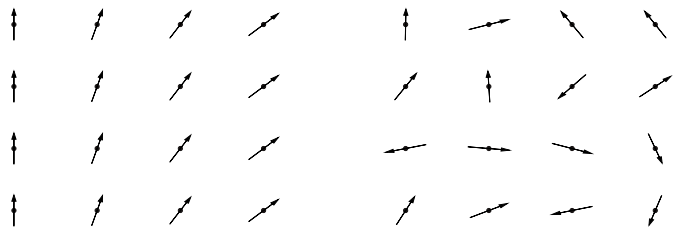
In the ground state all the spins are aligned, while at higher energy states, the spins are pointed in random directions as can be seen in figure (2.2).



(a) Ground state (b) Higher energy state

Figure 2.2: Energy states for XY model

Therefore, making a constant phase shift $\Phi_\mu = \frac{A}{L}$ of $\theta_i - \theta_j$ in the μ direction would change the energy drastically for the ground state while, on a statistical average, not change the higher states energy at all (see figure (2.3)).



(a) Ground state (b) Higher energy state

Figure 2.3: A phase shift for $\mu = x$

The free energy change ΔF for such a shift is

$$\Delta F = L^d \cdot \frac{1}{2}\rho_s \left(\frac{A}{L}\right)^2 \Rightarrow \rho_s = \lim_{A\to 0} L^{2-d}\frac{\partial^2 \Delta F}{\partial A^2} \tag{2.20}$$

where d is the dimension and ρ_s is the superfluid density which is zero for a high energy state. The free energy is

$$F = -T\ln(Z) \Rightarrow F'' = T\left(\left(\frac{Z'}{Z}\right) - \left(\frac{Z''}{Z}\right)^2\right) \tag{2.21}$$

where $F' = \partial F/\partial A$. Examining Z from (2.18) with the added shift

$$Z = \prod_i \int \frac{\mathrm{d}\theta_i}{2\pi} \sum_{J_{\langle ij\rangle}=-\infty}^{\infty} \prod_{b=\langle ij\rangle} I_{J_{\langle ij\rangle}}(K) e^{iJ_{\langle ij\rangle}(\theta_i-\theta_j+\Phi_\mu)} \tag{2.22}$$

$$= \prod_i \sum_{J_b} \left(\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i-\theta_j)}\right)\left(\prod_b I_{J_b}\right) \cdot e^{iA\frac{1}{L}\sum_i J_{i,i+\mu}} \tag{2.23}$$

$$\tag{2.24}$$

where in the last step

$$\prod_i \left(\prod_{\langle ij\rangle} e^{iJ_{\langle ij\rangle}\Phi_\mu}\right) = \tag{2.25}$$

$$\{\Phi_\mu \neq 0 \text{ only for neighbours in the } \mu \text{ direction}\} = \tag{2.26}$$

$$\prod_i \left(e^{iJ_{i,i+\mu}\Phi_\mu}\right) = \tag{2.27}$$

$$e^{iA\frac{1}{L}\sum_i J_{i,i+\mu}} \tag{2.28}$$

Introduce the winding number in the μ direction as

$$W_\mu = \frac{1}{L}\sum_i J_{i,i+\mu} \tag{2.29}$$

Intuitively, this describes the net flux in the μ-direction. Given a loop within the bounds of the lattice, the winding number is always zero. This is since an equal amount of flux in $+\mu$ as in $-\mu$ is needed to form a loop. However, this is not the case for a percolating cluster going, for example, from $-\mu$ to $+\mu$ connecting with periodic boundary conditions. For such a 'winding' cluster, the winding number will be $+1$. An example can be seen in figure (2.4).

Using the definition (2.29) for the winding number in the partition function in equation (2.24) yields

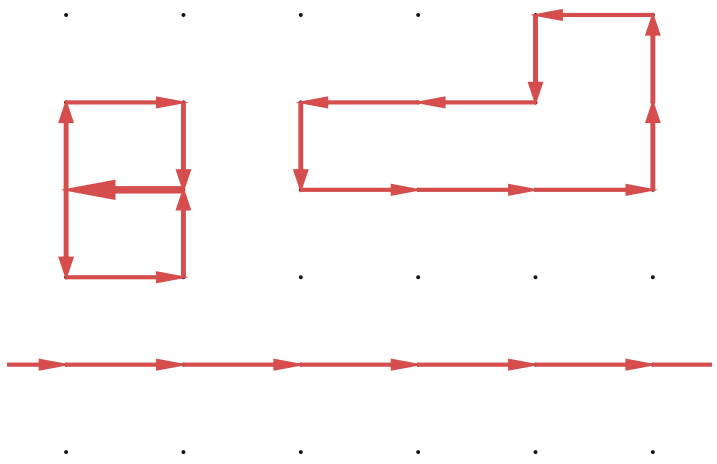


Figure 2.4: Three flux clusters on a square lattice. One percolating cluster with $W_x = +1$. The size of arrow corresponds to the number of flux quanta between two sites.

$$Z = \sum_{J_b} \left(\prod_b I_{J_b} \right) \prod_i \left(\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i - \theta_j)} \right) \cdot e^{iAW_\mu} \tag{2.30}$$

$$= \sum_{J_b,\ N_i=0} Z_0 \cdot e^{iAW_\mu} \tag{2.31}$$

$$= \sum_{W_\mu} Z_0 \cdot e^{iAW_\mu} \tag{2.32}$$

Using this result in equation (2.21)

$$\frac{\partial^2 F}{\partial A^2} = T \left(\left(\frac{\sum_{W_\mu} (iW_\mu) Z_0 e^{iAW_\mu}}{\sum_{W_\mu} Z_0 e^{iAW_\mu}} \right)^2 - \frac{\sum_{W_\mu} (-W_\mu^2) Z_0 e^{iAW_\mu}}{\sum_{W_\mu} Z_0 e^{iAW_\mu}} \right) \tag{2.33}$$

$$= T \left(-\langle W_\mu \rangle^2 + \langle W_\mu^2 \rangle \right) \tag{2.34}$$

$$= T \langle W_\mu^2 \rangle \tag{2.35}$$

Where $\langle W_\mu \rangle = 0$ since there is an equal chance of percolating from $-\mu$ to μ as the other way around.

The superfluid density can finally be determined as

$$\rho_s = L^{2-d} T \langle W_\mu^2 \rangle \tag{2.36}$$

2.3.3 Villain approximation

The scaling behaviour of the heat capacity is

$$C = \frac{e}{t} = \tilde{a} t^{-\alpha} + \tilde{b} \tag{2.37}$$

where $t = |T - T_c|$, $\alpha = -0.01$, and $\tilde{a},\ \tilde{b}$ are some constants.
Therefore, the energy per site, e is

$$e = a t^{1-\alpha} + b \tag{2.38}$$

And the total energy

$$E = L^d (a t^{1-\alpha} + b) \propto L^3, \text{ at } T = T_c \tag{2.39}$$

where d is the dimension of the sample. This scaling behaviour for the total energy can be seen in Figure (6.7) for $d = 3$.

2.4 Hausdorff dimension

2.5 Box dimension

Chapter 3

Method

3.1 Monte Carlo Simulations

3.2 Ergodicity

An ergodic system is one where its time average coincides with its ensemble average.

Intuitively, ergodicity is the assumption that a Markov chain starting from some state S_a with a non zero Boltzmann weight can reach any other state S_b withing a finite number of updates [1].

This is necessary to assume, since otherwise there could be a non zero contribution to the partition function not being sampled by the Markov chain.

3.3 Detailed Balance

Generally, a Markov process can be described through the Master equation

$$\frac{\mathrm{d}P_a}{\mathrm{d}t} = \sum_{a \neq b} (P_b W_{ba} - P_a W_{ab}) \tag{3.1}$$

where P_a is the probability to find the system in the state a, and W_{ab} is the transition rate from the state a to b. This equation describes the equilibrating of P_a into $P_a^{eq} \propto e^{-\beta E_a}$. The resulting equation is called detailed balance

$$W_{ba} e^{\beta E_b} = W_{ab} e^{\beta E_a} \tag{3.2}$$

This describes an equal rate of flow into the state a as out of it.

3.4 Worm Algorithm

3.5 Graph Dividing Algorithm

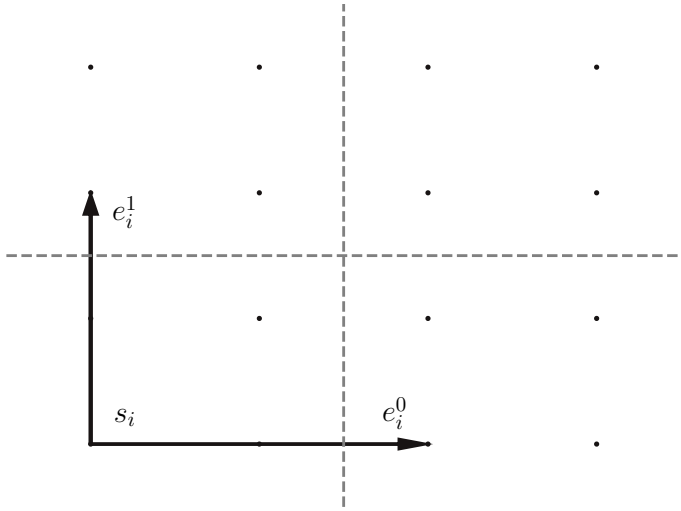


Figure 3.1: One step in the graph dividing algorithm where $l_i = 4$. e_i^0 and e_i^1 are drawn from site s_i. Summed permutations of $\{e_i^0, e_i^1\}$ give the starts for the next boxes. The next iteration of boxes are shown via the dividing dotted lines.

In order to calculate the box dimension the lattice need to be divided into boxes of decreasing size. A step by step instruction of a graph dividing algorithm is provided below, and an implementation in pseudocode is available in the Appendix at Section 8.1.

For brevity some abbreviations are introduced.

d = dimension	l_i = side length of the current box
l_0 = side length of the smallest box allowed	e_i^j = vector of length $l_i/2$ in the j'th direction
perm(v) = All permutations of v	s_i = starting site of the current box

1. If $l_i \geq l_0$, go to 2, else stop.
2. Save all sites in the current box, starting for s_i going l_i in d directions.
3. Find all starting points for new boxes.

(i) Form the matrix $E = (e_i^0, e_i^1, \ldots, e_i^d)^T$

(ii) For all vectors v_k in perm$(0, 0, \ldots, 0)$, perm$(1, 0, \ldots, 0)$, $\ldots$, perm$(1, 1, \ldots, 1)$, create the new start s_k as

$$s_k = v_k E$$

4. For each start s_k:

(i) $s_i = s_k$, $l_i = l_i/2$

(ii) Go to 1.

This algorithm can be written to perform in linear time, as can be seen in Figure (3.2).

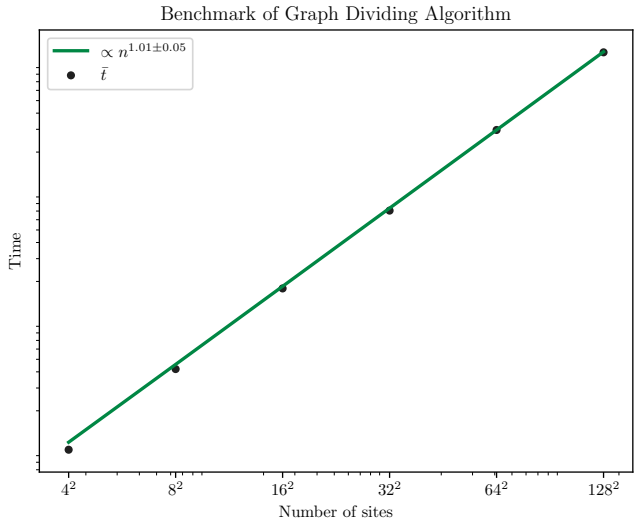


Figure 3.2: Loglog plot of a benchmark of the graph dividing algorithm. The y-axis show the time taken to perform one full graph divide normalized against the smallest time value.

3.6 Error Estimation

In this thesis a number of error estimation techniques were used to know how much data was needed for each measurement. In this section the techniques will be described intuitively.

3.6.1 Monte Carlo error estimation

Given a Monte Carlo simulation where polling of some quantity A has been done N times an estimation of the expectation value of A is

$$\bar{A} = \frac{1}{N}\sum_{i=1}^{N} A_i \tag{3.3}$$

where each sampling was labeled A_i. To show that this is an unbiased estimator the expectation value of the difference between the estimation and the real value $\langle A \rangle$ is used.

$$\begin{aligned}
\langle \bar{A} - \langle A \rangle \rangle &= \langle \bar{A} \rangle - \langle A \rangle && (3.4)\\
&= \left\langle \frac{1}{N}\sum_{i=1}^{N} A_i \right\rangle - \langle A \rangle && (3.5)\\
&= \frac{1}{N}\sum_{i=1}^{N} \langle A_i \rangle - \langle A \rangle && (3.6)\\
&= \frac{1}{N}\sum_{i=1}^{N} \langle A \rangle - \langle A \rangle && (3.7)\\
&= \frac{1}{N} N \langle A \rangle - \langle A \rangle = 0 && (3.8)
\end{aligned}$$

where the fact that A_i is a random sampling from the distribution of A was used in (3.7).

The standard deviation of this estimate can be calculated through the variance.

$$\begin{aligned}
\sigma_{\bar{A}}^2 &= V\left(\bar{A} - \langle A \rangle\right) && (3.9)\\
&= V\left(\bar{A}\right) - V\left(\langle A \rangle\right) && (3.10)\\
&= \{\langle A \rangle \text{ is a constant} \Rightarrow V(\langle A \rangle) = 0\} && (3.11)\\
&= V\left(\frac{1}{N}\sum_{i=1}^{N} A_i\right) && (3.12)\\
&= \{\text{Monte Carlo simulations give independent samples}\} && (3.13)\\
&= \frac{1}{N^2}\sum_{i=1}^{N} V(A_i) && (3.14)\\
&= \{V(A_i) = \sigma_A^2\} && (3.15)\\
&= \frac{1}{N^2} N \sigma_A^2 = \frac{\sigma_A^2}{N} && (3.16)
\end{aligned}$$

or

$$\sigma_{\bar{A}} = \frac{\sigma_A}{\sqrt{N}} \tag{3.17}$$

So the standard error in this estimation decreases as $N^{-1/2}$.

3.6.2 Bootstrap

Bootstrap is a resampling method to examine a probability distribution. In this thesis it was used to estimate the error propagation of parameters in curve fitting.

Given a set $\boldsymbol{x}$ of N measurements from an unknown distribution $\hat{\phi}$, some statistical calculation of interest can be done as $\theta = s(\boldsymbol{x})$. A resampling $\boldsymbol{x}_0$ of $\boldsymbol{x}$ comprised of N random measurements from $\boldsymbol{x}$ (where one measurement can be included several times), can then be used to calculate $\theta_0^* = s(\boldsymbol{x}_0)$. Repeating this N_B times gives an estimate $\theta^* = (\theta_0^*, \theta_1^*, ..., \theta_{N_B}^*)$ of the distribution $\hat{\theta}$. Assuming N_B is large then, by the central limit theorem, $\hat{\theta}$ is a normal distribution with some standard deviation σ_θ that can be used as an error estimation for θ.

3.7 Optimization

A large part of this project was spent optimizing the simulations. To fascilitate this process, the Google project *benchmark* was used for measuring and comparing running times of different algorithms. Only the C++ code was optimized, since the prototypes written in Python were not concerned with computational speed. What follows is a summation of the biggest optimizations that were achieved during this project.

3.7.1 Lattice implementation - Squashing the Graph

During most of the calculations a sweep through the entire lattice was necessary. The computational time taken by such a sweep is heavily affected by the implementation of the lattice data structure.

The first implementation used an array of arrays to represent the lattice. This has the advantage of a similar interface to mathematical matrices with rows and columns. This is however much slower than a single array, or a squashed graph, most likely due to cache misses[2]. Figure (3.3) shows a comparison between the time it takes to do one full sweep in the two implementations.

Furthermore, this simplifies the generalization to lattices of different dimensions since the need to allocate a set number of arrays in each array disappears. It also makes it easier to reuse the same algorithm for lattices of different dimensions.

The mapping used between the array index and the Euclidean space is

$$n = x + yL + zL^2 + ... \tag{3.18}$$

where L is the linear system size.

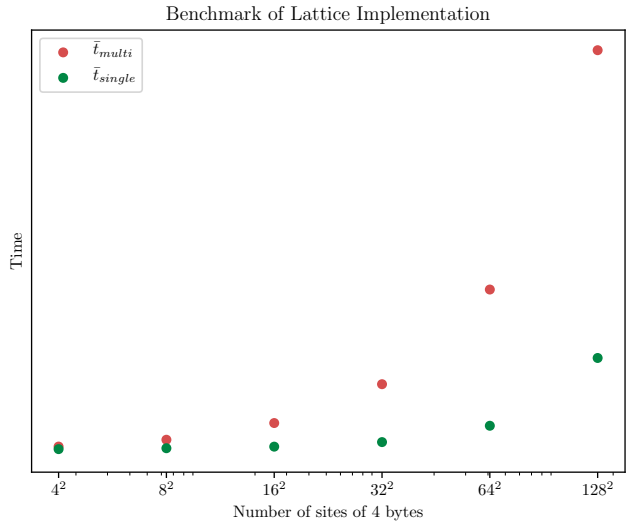


Figure 3.3: Plot of a sweep benchmark of two different lattice implementations. The red dots labeled $\bar{t}_{multi}$ is an array of arrays, while the green dots labeled $\bar{t}_{single}$ is a single array. The y-axis show the time taken to perform one full graph divide normalized against the maximum of the smallest time value for the two implementations.

3.8 Testing

Testing is an essential part of writing a simulation. The correctness of the code ensures that the physical model is aptly described. To facilitate the development, a number of coding practices were applied, such as unit testing (Section 3.8.1) and regression testing (Section 3.8.2). The tests for the prototypes were written in the Python standard library utility *unittest*, and those for the main simulation used the C++ framework *Catch2*.

3.8.1 Unit Testing

Unit testing refers to the practice of isolating a 'unit' of code and testing its correctness. A unit can be any small piece of code with an expected behaviour.

This was done by manually calculating the expected output of some code, given some input, and ensuring that the piece of code produced an equivalent result.

The parts of the simulation using a pseudorandom number generator were tested by randomly selecting a set of seeds on which the tests were run upon. The correctness is then assumed from this subset of possible inputs.

When multiple unit tests are tested together, it is called integration testing. This assumes that each unit is correct by itself, and it was used to more closely ressemble the actual simulation.

3.8.2 Regression Testing

Rerunning the relevant tests in a continuous manner after each update is called regression testing. This ensures that, however many unintended consequences were introduced during the update, the expected behaviour of the program is still intact.

This was done by writing a 'hook' such that, whenever a piece of code were recompiled, the relevant tests were subsequently compiled and run. With sufficient tests in place, the correctness of the code after the update was assumed.

Chapter 4

Graph Division and Hausdorff Dimension

4.1 Graph Labeling

4.1.1 Hoshen Kopelman

To find the clusters a modified version of the Hoshen Kopelman algorithm was used. A raster scan is used to label disjoint sets into groups with some canonical label[3]. It is a variant on the union-find algorithm and is most easily described through the associated functions. Intuitively, applying the find function on a site i returns the canonical, often implemented as the smallest, label in the cluster that i belongs to. Union uses find to ensure that two sites i and j are connected by setting the canonical label of i to that of j (or vice versa).

An example implementation would be to have a 2D graph without periodic boundary conditions of zeros and ones, where a site is occupied if it has a one associated with it, and unoccupied otherwise. A disjoint set here is a number of occupied sites neighbouring each other with unoccupied sites surrounding them. For simplicity the scan can start in the lower left corner, moving right and up, while search for neighbours left and down, ensuring that if a neighbouring site is occupied, it has been labeled before.

Start by setting each site to a unique label, putting all sites in individual clusters. Go through the lattice until an occupied site i is found. Search the neighbours below and to the left. If none of these neighbours are occupied, label i have a unique label and move to the next site. If i has one occupied neighbour it must have been labeled before, so i inherits the neighbours label. Finally if both neighbours are occupied, site i must be connecting a cluster and a union is performed on the neighbours to join their labels. A final pass through the lattice using the find function ensures that all sites have their canonical label.

In this paper an occupied site corresponds to a site with connections to the neighbouring sites (in the 2D example above, each site could have four such connections). In the original paper by Hoshen and Kopelman the labels for the sites who did not originally carry the canonical label, were set to a negative integer, symbolizing that they were aliases. A positive value was used at the canonical label, showing the number of sites in that cluster. This was not used in this project since the number of links in a cluster is not necessarily equal to the number of sites.

4.2 Fractals

Everyone agrees that the dimension of a point is zero, and that of a smooth line is one, but what about a set of points? A definition could be to say that the dimension is the minimum number of coordinates needed to describe every point in the set. Effectively, a point would describe itself, and a curve could be parametrized to the distance of some point on the same curve.

The situation is more complex when examining fractals. Take for example the Koch curve, it starts out as a line segment of length L_0, and successively adds a 'bump', making the total length $L_1 = 4/3 \cdot L_0$. Iterating n times gives a line length of $L_n = (4/3)^n \cdot L_0$, and so the final fractal length is infinite.

Any two point on the final curve has a distance of infinity between them, so parametrization is impossible. But the area is still finite, so the dimension should intuitively be somewhere between one and two.

A useful concept here is the similarity dimension, defined by the scaling of each iteration. If m is the number of similar elements after an iteration and r is the scaling factor, the dimension is defined by $m = r^d$, or equivalently

$$d = \frac{\ln m}{\ln r} \tag{4.1}$$

So for the Koch curve, each segment is divided into fourths with each having one third the length from the previous iteration, giving it a dimension of $\ln 4 / \ln 3 \approx 1.26$.

Chapter 5

Connection between Hausdorff Dimension and Scaling Behaviour

Chapter 6

Results

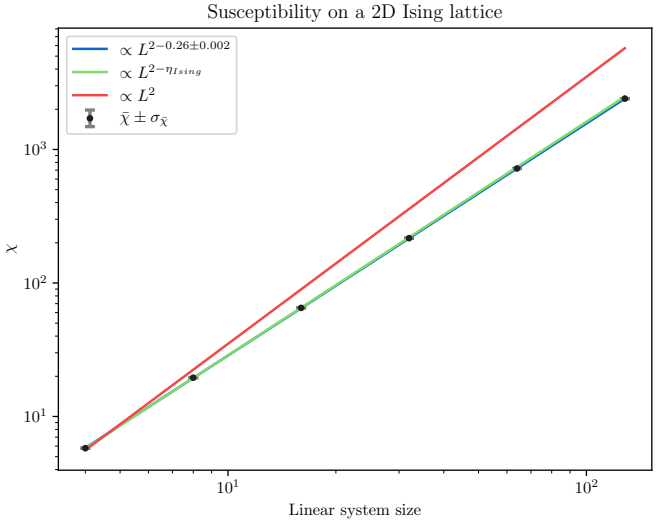


Figure 6.1: Scaling of susceptibility at T_c on an Ising lattice of varying sizes. The measured critical exponent $\eta = 0.26 \pm 0.002$ is compared to the theoretical value $\eta_{Ising} = 0.25$. The red line labeled $\propto L^2$ illustrates data where susceptibility scales with system size.

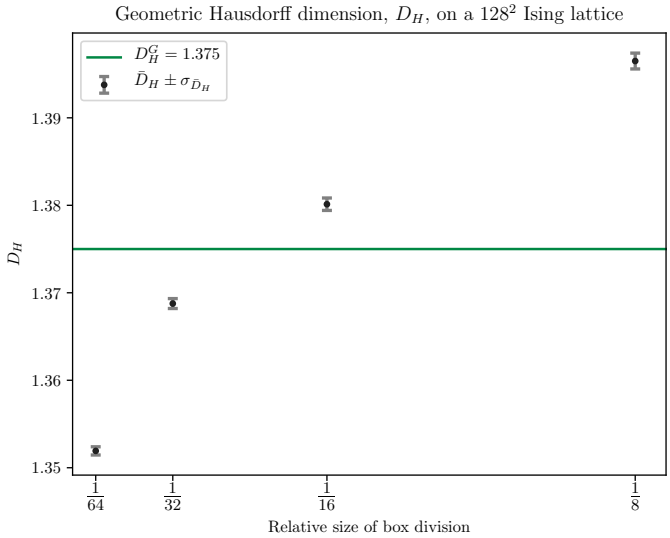


Figure 6.2: Hausdorff dimension of the maximum loop length on a 128^2 Ising lattice at T_c using the box dimension. The x axis shows the size of one box relative to the side length of the lattice. The green line indicates the theoretical dimension of the geometric Ising cluster, $D_H^G = 1.375$ [4]. Comparing to the smallest box size as $D_H = 1.35193 \pm 5 \cdot 10^{-4}$.

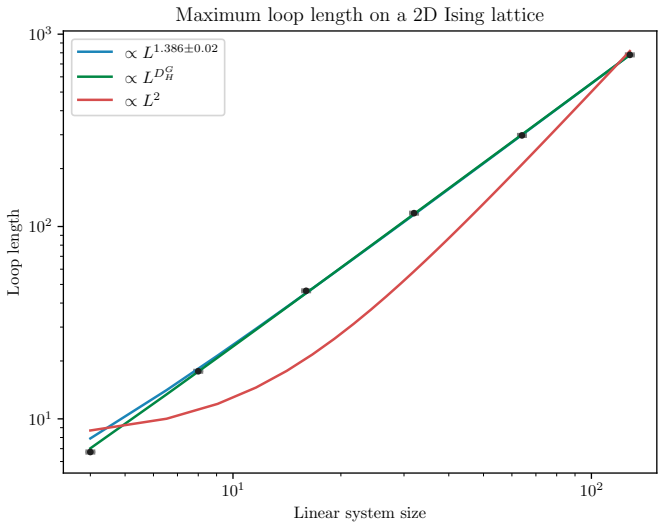


Figure 6.3: Log-log plot of the maximum loop length at T_c on an Ising lattice of varying sizes. The measured scaling factor is 1.386 ± 0.02 compared to the theoretical Hausdorff dimension, $D_H^G = 1.375$ [4]. The red line labeled $\propto L^2$ illustrates data where maximum loop length scales with system size.

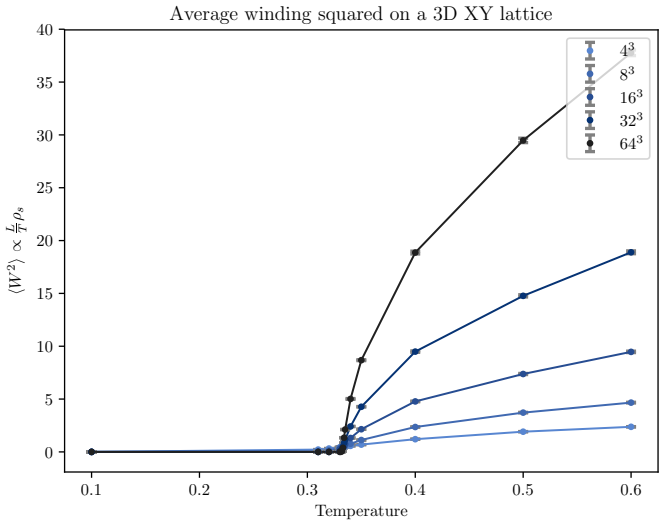


Figure 6.4: Average winding number squared, $\langle W^2 \rangle \propto \rho_s$, plotted on a 3D XY lattice of varying sizes. The overall structure of the average winding number squared as a function of the temperature is shown.

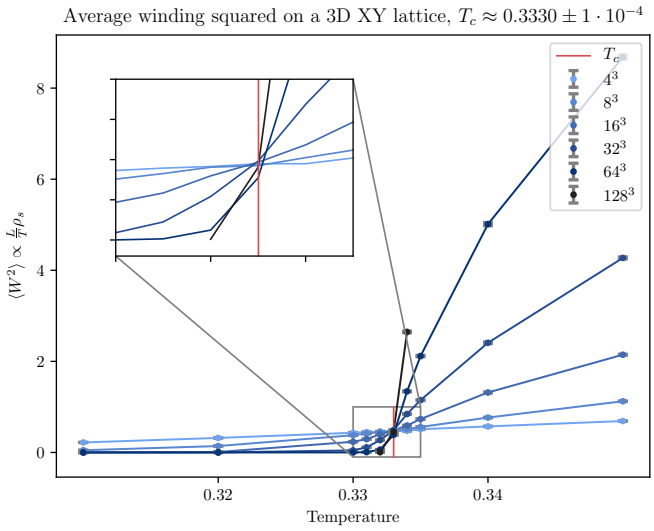


Figure 6.5: Average winding number squared, $\langle W^2 \rangle \propto \rho_s$, plotted on a 3D XY lattice of varying sizes. Due to the Villain approximation the transition is flipped such that $\rho_s \neq 0$ for $T > T_c$, and T_c is translated from ≈ 2.2 to ≈ 0.333 indicated by the intersection. By taking the weighted average of the intersections the critical temperature can be estimated to $T_c = 0.3330 \pm 1 \cdot 10^{-4}$.

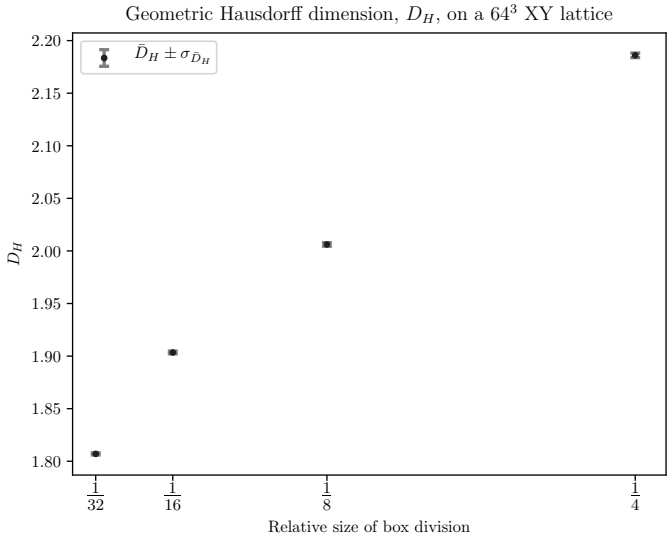


Figure 6.6: Hausdorff dimension of the maximum loop length on a 64^3 XY lattice at T_c using the box dimension. The x axis shows the size of one box relative to the side length of the lattice. Assuming that the smallest box size gives the best approximation, the result is $D_H = 1.807 \pm 5 \cdot 10^{-6}$.

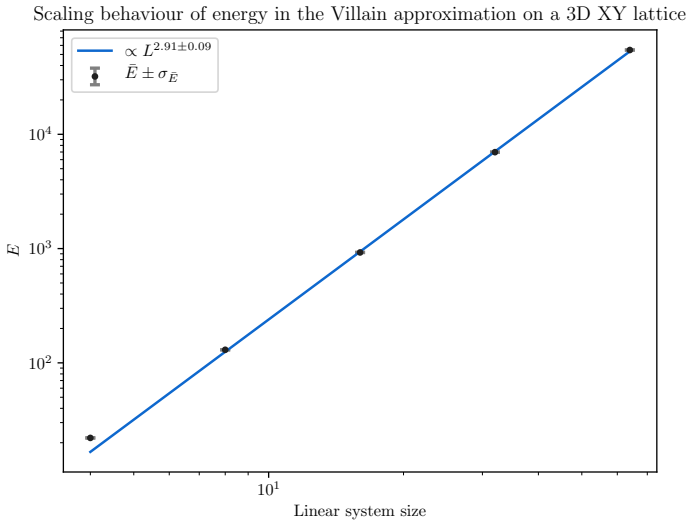


Figure 6.7: Log-log plot of the energy at T_c on an 3D XY lattice of varying sizes. The measured scaling factor is 2.91 ± 0.09. The energy in the Villain approximation is proportional to the sum of the squares of flux flowing through the lattice.

Chapter 7

Summary and conclusions

Chapter 8

Appendix

8.1 Pseudo Code for Box Division Algorithm

Bibliography

[1] H. C. Andersen and D. Chandler, *Robert W. Zwanzig: Formulated nonequilibrium statistical mechanics*, Proceedings of the National Academy of Sciences **111**, 11572 (2014).

[2] P. J. Hanlon *et al.*, *The Combinatorics of Cache Misses during Matrix Multiplication*, Journal of Computer and System Sciences **63**, 80 (2001).

[3] J. Hoshen and R. Kopelman, *Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm*, Physical Review B **14**, 3438 (1976).

[4] B. Duplantier, *Conformally Invariant Fractals and Potential Theory*, Physical Review Letters **84**, 1363 (2000).