Master Thesis

Worm simulation of Hausdorff dimension of critical loop fluctuations

Simon Rydell

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

Stockholm, Sweden 2018

Typeset in LaTeX

För avläggande av teknologie examen inom ämnesområdet teoretisk fysik.

Scientific thesis for the master degree of Engineering in the subject area of Theoretical physics.

TRITA-SCI-GRU 2018:405

Printed in Sweden by Universitetsservice US AB, Stockholm November 2018

Abstract

This thesis examines the Hausdorff dimension of cluster configurations at the critical temperature generated by the Worm algorithm. The Hausdorff dimension of the largest cluster configuration on a $2D$ Ising lattice is compared to the fractal dimensions of self avoiding and random walks, showing similarities to the self avoiding walk. These similarities suggests a new typ of self-avoiding walk where 'twisted loops' are allowed. The box counting measure [1] gives a Hausdorff dimension $D_H = 1.3519(5)$ and the scaling method [2] $D_H = 1.386(2)$ compared to the theoretical geometric dimension $D_H^G = 1.375$ [3]. The critical temperature of $3D$ XY lattice in the Villain approximation is determined by calculating the winding number as $T_c = 0.3331(1)$. The Hausdorff dimension is calculated for the largest cluster of the $3D$ XY model to be $D_H = 1.7747(8)$ and is compared to results in the literature [4][2].

Keywords: Hausdorff dimension, worm algorithm, Ising model, XY model, Villain model, winding number, critical phenomena, finite size scaling

Sammanfattning

Den här uppsatsen behandlar Hausdorffdimensionen av klusterkonfigurationer vid den kritiska temperaturen som är genererade med hjälp av wormalgoritmen. Hausdorffdimensionen hos klusterkonfigurationer på ett $2D$ Isinggitter jämförs med slumpvandringar och självundvikande banor och visar likheter till de slälvundvikande banorna. Dessa klusterkonfigurationer föreslås vara en ny typ av självundvikande banor där 'tvistade loopar' tillåts. Lådräkningsmåttet [1] ger Hausdorffdimensionen $D_H = 1.3519(5)$ och skalningsmetoden [2] $D_H = 1.386(2)$ jämfört med den teoretiska geometriska dimensionen $D_H^G = 1.375$ [3]. Den kritiska temperaturen på ett $3D$ XYgitter i Villainapproximationen bestäms med hjälp av vindningstalet och ger resultatet $T_c = 0.3331(1)$. Hausdorffdimensionen hos det största spinnkluster i $3D$ XYmodellen bestäms på samma sätt till $D_H = 1.7747(8)$ och jämförs med resultat från litteraturen [4] [2].

Nyckelord: Hausdorffdimension, wormalgoritm, Isingmodell, XYmodel, Villainmodellen, vindningstal, kritiska fenomen, skalningsteori

Preface

This thesis is the result of 8 months of work between April and November 2018 for the degree of Master of Science in Engineering Physics. The work was written under the supervision of Prof. Mats Wallin at the Theoretical Physics department at KTH Royal Institute of Technology, Sweden. All large scale simulations in this thesis were run on the Octopus cluster belonging to the physics department at KTH.

Acknowledgements

I would like to thank my supervisor Prof. Mats Wallin for his advice and guidance during this project. It has been a great learning experience working on this thesis and I feel fortunate to have had the opportunity to do so.

Lastly, I want to thank my fellow thesis writers, in no particular order, Robert Vedin, Simon Sandell, Julia Hannukainen, Gunnar Bollmark, and Kristoffer Aronsen for the exchange of ideas and discussions which, in my opinion, has enhanced the thesis.

Contents

Chapter 1

Introduction

This chapter first introduces the notion of fractal dimension in terms of a scaling property, as well as a graph filling measure. The models and algorithms used in this thesis are thereafter introduced.

1.1 Fractal dimension

Most people agrees that the dimension of a point is zero, and that of a smooth line is one, but what about a set of ordered points? One definition would be to have the dimension linked to the number of degrees of freedom needed to parametrize the object [1]. A point would describe itself, and a line could be described as the curve distance from some point on the line.

This leads to the idea of fractals. Take for example the Koch curve, seen in Fig. (1.1). 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 when n grows to infinity, so does the length. Now each two points on the curve has at least one bump between them. However each bump has $\sim\ n$ bumps on it, making the length between the points infinite. This cannot be described by the curve distance between points, implying that it can not have a dimension of 1. Although, it is not two-dimensional either since it clearly has 'holes' between the lines. Consequently the dimension should be somewhere between 1 and 2.

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

$$d = \frac{\ln m}{\ln r} \tag{1.1}$$

Then 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 $\ln 4 / \ln 3 \approx 1.26$.

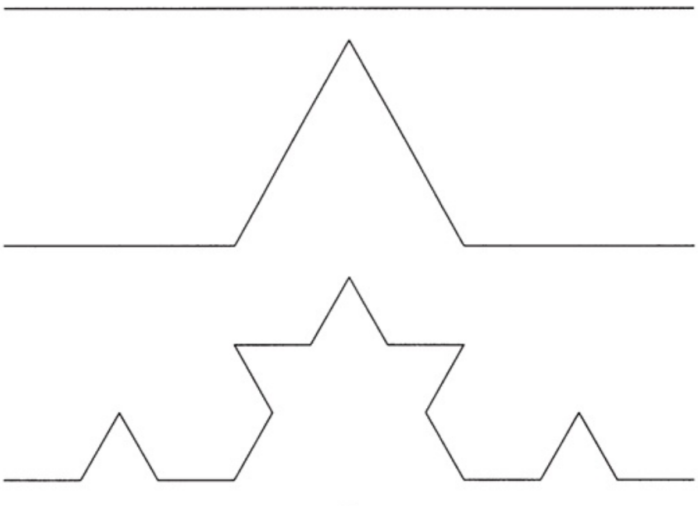


Figure 1.1: The Koch curve during three iterations [1].

The box counting method used in this thesis ressembles the method used in Eq. (1.1). But the scaling factor is not known, so the idea is to try to cover the fractal with a set of boxes of side length ϵ. In the same way, the fractal dimension is defined as

$$d = \lim_{\epsilon \to 0} \frac{\ln N(\epsilon)}{\ln 1/\epsilon} \tag{1.2}$$

where $N(\epsilon)$ is the number of boxes with side length ϵ needed to cover the fractal [1]. Hence, this is limited by the dimension of each box. This also says something about how a curve that fully covers the plane will have a dimension of 2, and analogously a curve which covers the whole room will have a dimension of 3.

In condensed matter physics, a lot of simulations operates on a lattice of spins, representing for example atoms in a grid. It is often the case that the models are too large or too complicated to be solved by hand and therefore computational simulations are needed. In 2001 an algorithm called the Worm algorithm was proposed [5]. Instead of operating on individual spins, it generates spin configurations

by 'connecting' lattice sites together. This algorithm then generates a set of objects which have fractal like properties [3].

The simplest such lattice is called the Ising lattice, where each spin can have one of two values. When the Worm algorithm is applied, the resulting fractal in a two dimensional Ising lattice show similar dimension to a self avoiding walk. It is proposed to be a new such type of walk where 'twisted loops' are allowed where the walk is allowed to cross itself, but still has elements of self-avoidance.

The XY model adds some complexity by allowing each spin to rotate around some axis. This complexity is shown in the Worm algorithm by adding a direction and weight to the connection. Hence, instead of connecting site a and site b, it connects site a *to* site b with the weight w. There has been some contesting results of the fractal dimension of spin configurations on a $3D$ XY lattice [4][2] and this thesis attempts to add to that discussion.

Chapter 2

Background

In this chapter the different models used in this thesis are described. The loop expansion, giving the context of the Worm algorithm is examined for each model. The method of determining the critical temperature of an XY lattice using the winding number is discussed. Finally the definition of the Hausdorff dimension is shown and different ways of determining it is discussed.

2.1 Ising model

The Ising model consists of discrete atomic spins S that can be found in two states represented by the values $\{-1, 1\}$. This can be applied to a lattice where S_i is the spin of lattice site i.

The energy of an Ising spin configuration is given by the Hamiltonian

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

where J is the bond strength in the lattice and $\langle ij \rangle$ refers to a pariwise interaction between nearest neighbours only.

2.1.1 Loop expansion

Let $K = \beta J$ where $\beta = 1/k_{\mathrm{B}}T$. Then from Eq. (2.1)

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

The partition function Z can therefore be written as

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

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

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

In a $2D$ lattice with N spins and periodic boundary conditions there are $2N$ bonds between sites. Therefore the partition function is

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

where

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

where the sums $\sum_{l=L}$ should be interpreted as the sum over all sets where the link length is L. The link length is the number of coupling terms $S_i S_j$ as can be seen in Fig. (2.1).

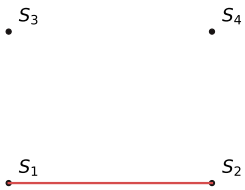
Since $\sum_{S_i = \pm 1} S_i = 0$, only terms with an even number of S_i are contributing to Z'. Call these terms closed, indicating that they represent a closed loop. The

sum over all contributing terms gives a factor of 2^N. Rewriting Z' in terms of loop lengths gives

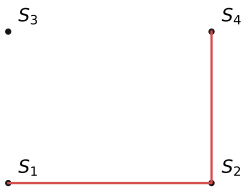
$$Z' = \sum_L g(L) T^L \tag{2.4}$$

where $g(L)$ is the number of loops with length L. Finally, the partition function can be written as

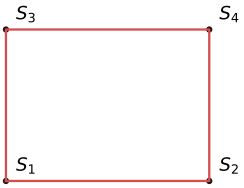
$$Z = 2^N \cosh^{2N}(K) \sum_L g(L) T^L \tag{2.5}$$



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



(b) $(S_1S_2)(S_2S_4),\ L = 2$



(c) $(S_1S_2)(S_2S_4)(S_4S_3)(S_3S_1),\ L = 4$

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

2.1.2 Correlation Function

By the fluctuation-dissipation theorem the susceptibility can be written as

$$\chi = \frac{\beta}{N} \sum_{ij} G_{ij} \tag{2.6}$$

where $G_{ij} = \langle S_i S_j \rangle - \langle S_i \rangle^2$ is the connected correlation function between two lattice points i and j [6].

In the high-temperature expansion the term $\langle S_i \rangle^2$ goes to zero. Thus, for a $2D$ Ising model

$$G_{ij} = \frac{1}{Z} \sum_{\text{all states}} S_i S_j e^{-\beta E} \tag{2.7}$$

$$= \frac{1}{Z} \cosh^{2N}(K) \sum_{\text{all states}} S_i S_j \Pi_{\langle ij \rangle} (1 + \tanh(K) S_i S_j) \tag{2.8}$$

2.2 XY model

The XY model describes a classical system of 2-dimensional classical spins s_i of unit length interacting on a lattice. The Hamiltonian is

$$H = -J \sum_{\langle ij \rangle} s_i \cdot s_j \tag{2.9}$$

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

The spins can be described as $s_i = (\cos\theta_i,\ \sin\theta_i)$ yielding the Hamiltonian

$$H = -J \sum_{\langle ij \rangle} \cos(\theta_i - \theta_j) \tag{2.10}$$

The partition function is therefore

$$Z = \prod_i \int_0^{2\pi} \frac{\mathrm{d}\theta_i}{2\pi} e^{-\beta H} = \prod_i \int_0^{2\pi} \frac{\mathrm{d}\theta_i}{2\pi} e^{K \sum_{\langle ij \rangle} \cos(\theta_i - \theta_j)} \tag{2.11}$$

where $K = J\beta$.

2.2.1 Loop expansion

Since Eq. (2.11) 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

$$e^{\alpha \cos\beta} = \sum_{\gamma=-\infty}^{\infty} I_\gamma(\alpha) e^{i\gamma\beta} \tag{2.12}$$

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

$$Z = \prod_i \int \frac{\mathrm{d}\theta_i}{2\pi} \sum_{J_{\langle ij\rangle}=-\infty}^{\infty} \prod_{\langle ij\rangle} I_{J_{\langle ij\rangle}}(K) e^{iJ_{\langle ij\rangle}(\theta_i-\theta_j)} \tag{2.13}$$

$$= \prod_i \sum_{J_b} \left(\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i-\theta_j)} \right) \left(\prod_b I_{J_b} \right) \tag{2.14}$$

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 integer variables J located on the links of the lattice. Since they are conjugate to the phases θ_i, the J variables are currents, and are henceforth called link currents. Noting that

$$\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i-\theta_j)} = C\delta_{N_i 0} \tag{2.15}$$

leads to the conclusion that the sum of incoming and outgoing link currents 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 only closed loops of link currents.

2.2.2 Winding number

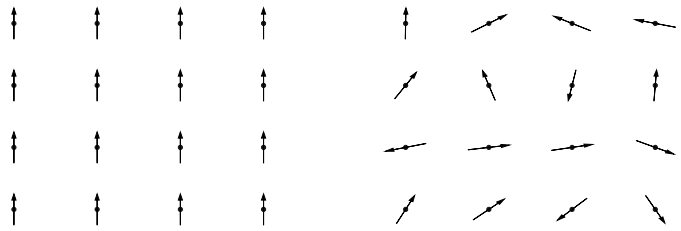
In the ground state all the spins are aligned, while at higher energy states, the spins points in random directions as can be seen in Fig. (2.2).

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 Fig. (2.3)).

The free energy change ΔF for such a shift is

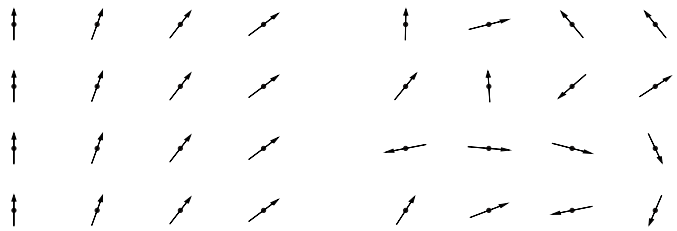
$$\Delta F = L^d \cdot \frac{1}{2}\rho_s \left(\frac{A}{L}\right)^2 \Rightarrow \rho_s = \lim_{A\to 0} L^{2-d} \frac{\partial^2 \Delta F}{\partial A^2} \tag{2.16}$$

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



(a) Ground state (b) Higher energy state

Figure 2.2: Energy states for the XY model.



(a) Ground state (b) Higher energy state

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

$$F = -T\ln(Z) \Rightarrow F'' = T\left(\left(\frac{Z'}{Z}\right) - \left(\frac{Z''}{Z}\right)^2\right) \tag{2.17}$$

where $F' = \partial F/\partial A$. Examining Z from Eq. (2.14) with the added shift yields

$$Z = \prod_i \int \frac{\mathrm{d}\theta_i}{2\pi} \sum_{J_{\langle ij\rangle}=-\infty}^{\infty} \prod_{\langle ij\rangle} I_{J_{\langle ij\rangle}}(K) e^{iJ_{\langle ij\rangle}(\theta_i - \theta_j + \Phi_\mu)} \tag{2.18}$$

$$= \prod_i \sum_{J_b} \left(\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i - \theta_j)}\right) \left(\prod_b I_{J_b}\right) \cdot e^{iA\frac{1}{L}\sum_i J_{i,i+\mu}} \tag{2.19}$$

$$\tag{2.20}$$

where in the last step

$$\prod_i \left(\prod_{\langle ij \rangle} e^{iJ_{\langle ij \rangle} \Phi_\mu} \right) = e^{iA \frac{1}{L} \sum_i J_{i,i+\mu}} \tag{2.21}$$

Introduce the winding number in the μ direction as

$$W_\mu = \frac{1}{L} \sum_i J_{i,i+\mu} \tag{2.22}$$

Geometrically this describes the net number of times the loop crosses the system 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 Fig. (2.4).

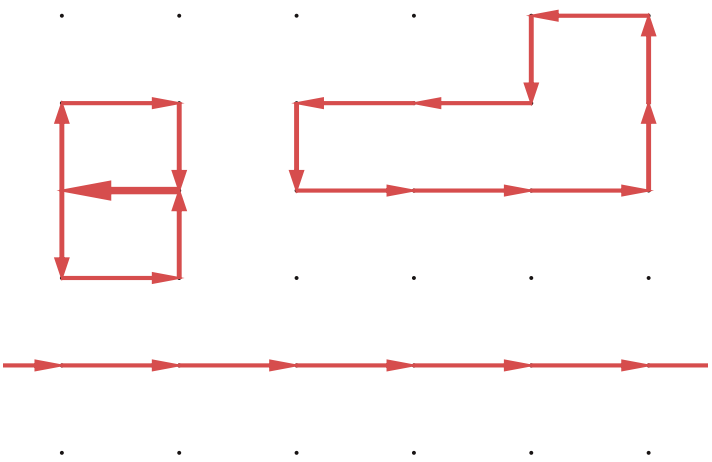


Figure 2.4: Three flux clusters on a square lattice. The straight link current percolates the system with $W_x = +1,\ W_y = 0$ and the other two loops do not percolate and therefore has the winding numbers $W_x = 0,\ W_y = 0$.

Using the definition (2.22) for the winding number in the partition function in Eq. (2.20) yields

$$Z = \sum_{J_b} \left(\prod_b I_{J_b} \right) \prod_i \left(\int \frac{\mathrm{d}\theta_i}{2\pi} e^{iN_i(\theta_i - \theta_j)} \right) \cdot e^{iAW_\mu} \tag{2.23}$$

$$= \sum_{J_b,\ N_i=0} Z_0 \cdot e^{iAW_\mu} \tag{2.24}$$

$$= \sum_{W_\mu} Z_0 \cdot e^{iAW_\mu} \tag{2.25}$$

Using this result in Eq. (2.17) gives

$$\frac{\partial^2 F}{\partial A^2} = T \left(\left(\frac{\sum_{W_\mu} (iW_\mu) Z_0 e^{iAW_\mu}}{\sum_{W_\mu} Z_0 e^{iAW_\mu}} \right)^2 - \frac{\sum_{W_\mu} (-W_\mu^2) Z_0 e^{iAW_\mu}}{\sum_{W_\mu} Z_0 e^{iAW_\mu}} \right) \tag{2.26}$$

$$= T \left(-\langle W_\mu \rangle^2 + \langle W_\mu^2 \rangle \right) \tag{2.27}$$

$$= T \langle W_\mu^2 \rangle \tag{2.28}$$

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

$$\rho_s = L^{2-d} T \langle W_\mu^2 \rangle \tag{2.29}$$

This can be used to determine the critical temperature by varying the temperature and system size L as the superfluid density is expected to drop sharply to zero at T_c.

2.2.3 Villain approximation

In the Villain model the Hamiltonian has the form [7]

$$H = \sum_{ij} V_{ij}(\theta_i - \theta_j) \tag{2.30}$$

By taking

$$V_{ij}(\theta_i - \theta_j) = -J \cos(\theta_i - \theta_j) \tag{2.31}$$

the XY model in Eq. (2.10) is recovered.

The Villain approximation consists of replacing the interaction with a periodic Gaussian function [7]. Through a series of transformations the energy in this approximation can be written as [8]

$$E = \frac{1}{2}\sum_i J_i^2 \tag{2.32}$$

where J_i^2 is the flux from site i.

2.2.4 Finite Size Scaling

The number of particles in a typical physical sample is close in orders of magnitude to Avogadro's number $N_A \sim 10^{23}$, much larger than that of a normal simulated system. Finite size scaling is a way to relate the results from simulations on smaller lattices to that of larger, physical systems. When scaling the system by some factor l, the singular part of the free-energy density is assumed to scale as [9]

$$f_s(t, L^{-1}) = l^{-d} f_s(l^{y_t} t, lL^{-1}) \tag{2.33}$$

where $t = (T - T_c)/T_c$, and L is the linear system size, for a system without an external magnetic field. By choosing $l = L$ the free energy only depends on t, and the scaling behaviour of any thermodynamic quantity, describable by the free energy, can be determined. The scaling behaviour of the heat capacity is [9]

$$C = \frac{e}{t} = \tilde{a}t^{-\alpha} + \tilde{b} \tag{2.34}$$

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

$$e = at^{1-\alpha} + b \tag{2.35}$$

and the total energy

$$E = L^d(at^{1-\alpha} + b) \propto L^d, \text{ at } T = T_c \tag{2.36}$$

where d is the dimension of the sample. This scaling behaviour for the total energy can be seen in Fig. (4.11) for $d = 3$.

2.3 Hausdorff dimension

This section introduces the Hausdorff measure which can be used to rigorously define the Hausdorff dimension. Let X be a metric space, α be some positive real number, then the α-Hausdorff measure of a subset $A \subset X$ is defined as

$$\mathcal{H}_\alpha^\delta(A) = \inf\left\{\sum_{B\in\mathcal{B}} (\mathrm{diam}(B))^\alpha\right\} \tag{2.37}$$

where $\mathcal{B}$ is a cover of A of closed balls with diameter no larger than δ [10]. Taking the limit where $\delta \to 0$, the number of possible covers decreases. Since the limit is bounded from below, the limit exists [11] and

$$\lim_{\delta\to 0} \mathcal{H}_\alpha^\delta(A) = \mathcal{H}_\alpha(A) \in [0,\infty) \tag{2.38}$$

Examining Eq. (2.38) gives

$$\lim_{\delta\to 0} \mathcal{H}_\alpha^\delta(A) = \lim_{\delta\to 0} \inf\left\{\sum_{B\in\mathcal{B}} (\mathrm{diam}(B))^\alpha\right\} \tag{2.39}$$

$$\leq \lim_{\delta\to 0} \inf\left\{\sum_{B\in\mathcal{B}} \delta^\alpha\right\} \tag{2.40}$$

$$= \lim_{\delta\to 0} \inf\left\{N_\delta^A \delta^\alpha\right\} \tag{2.41}$$

where N_δ^A is the number of balls with diameter δ that can cover A. Though not a proof, one can intuitively say that if for some $\alpha > 0$ the limit is finite, then $\mathcal{H}_{\alpha'} = 0$ for each $\alpha' > \alpha$. Therefore the number

$$\dim_H(A) = \inf\{\alpha > 0 : \mathcal{H}_\alpha(A) = 0\} \tag{2.42}$$

exists and is called the Hausdorff dimension of A [10].

2.4 Box dimension

Given some subset A of a metric space X, let $N(\epsilon)$ be the number of boxes with side length ϵ needed to cover A. Then the box dimension d of A is defined as [1]

$$d = \lim_{\epsilon\to 0} \frac{\ln N(\epsilon)}{\ln 1/\epsilon} \tag{2.43}$$

if the limit exists. For intuition it helps to look at some examples. If A is a smooth line of length l, then the number of boxes needed to cover A scales as

$$N_l(\epsilon) \propto \frac{L}{\epsilon} \tag{2.44}$$

while for some two dimensional region with area Λ the boxes needed is

$$N_\Lambda(\epsilon) \propto \frac{\Lambda}{\epsilon^2} \tag{2.45}$$

such that for a d-dimensional subset A, the boxes needed will scale as

$$N(\epsilon) \propto \frac{1}{\epsilon^d} \Rightarrow d = \frac{\ln N(\epsilon)}{\ln 1/\epsilon} \tag{2.46}$$

Note that the box dimension and the Hausdorff dimension coincides for fractals that satisfy the open set condition [12]. The open set condition [13] says that for a sequence of contractions $c_1, c_2, ..., c_m$ there exists a nonempty open set V such that

$$\cup_{i=1}^m c_i(V) \subset V, \text{ and } c_i(V) \cap c_j(V) = \emptyset \text{ for } i \neq j \tag{2.47}$$

where a contraction is a function f such that

$$|f(x) - f(y)| < |x - y| \tag{2.48}$$

Intuitively, this means that the images $c_i(V)$ do not overlap.

2.4.1 Scaling Dimension

Another way of determining the Hausdorff dimension is to consider a scaling system [14]. The Hausdorff dimension for the largest cluster in a configuration is then defined as

$$N = L^{D_H} \tag{2.49}$$

where N is the number of links in the largest cluster, L is the linear system length and D_H is the Hausdorff dimension. This relation provides a convenient way of calculating the dimension of loop configurations in simulations.

Chapter 3

Method

The numberical simulations used in this thesis are presented in this chapter, starting with the Monte Carlo simulations, laying the background for the Worm algorithm. The Hoshen Kopelman algorithm, used for determining the location of the clusters in the graph is discussed, and an algorithm used for dividing the graph into boxes used in the box counting method is outlined. Finally, error estimation and code implementation is discussed.

3.1 Monte Carlo Simulations

A Monte Carlo simulation is a stochastic algorithm to get statistical results for a model system with many degrees of freedom. In Markov Chain Monte Carlo the idea is to use Markov Chains to propose the next step in the algorithm, i.e. the next proposed step is only dependent on the previous one. Using a well-chosen stationary probability distribution, together with ergodicity, the desired distribution can be sampled very efficiently.

3.2 Ergodicity

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

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 within a finite number of updates [15].

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.

Together with ergodicity, detailed balance ensures that the algorithm samples a desired target distribution, here the Boltzmann statistics [16].

3.4 Metropolis Algorithm

It is often very useful to factorize the transition rate according to

$$W_{ab} = T_{ab} A_{ab} \tag{3.3}$$

where T_{ab} is a trial proposition probability and A_{ab} is an acceptance probability. Letting $T_{ab} = T_{ba},\ \forall a,\ b$ and choosing one of the acceptance probabilities $A_{ab},\ A_{ba}$ to be equal to 1, together with detailed balance, the Metropolis algorithm is realized [16]. The acceptance probability is therefore

$$A_{ab} = \min\left\{1, \frac{P_b}{P_a}\right\} = \min\{1, e^{-\beta(E_b - E_a)}\} \tag{3.4}$$

3.5 Worm Algorithm

The Worm algorithm operates on graph configurations instead of individual spins. This way, the algorithm can stay local and can avoid, to some extent, the so called critical slowdown that happens near transition points [5].

The algorithm still uses the Metropolis acceptance rates and therefore it fulfills detailed balance (see Sec. 3.3).

The algorithm is divided into three parts. First a site and one of its neighbours are chosen at random on the lattice. The site and its neighbour n is then connected with some acceptance probability. If the move was accepted, the 'worm' is now at n and chooses a neighbour to n at random, creating a path onto the lattice. The algorithm finishes whenever the worm is back at its starting site, forming a loop. Now thermodynamic quantaties can be sampled and used for averages when sufficient data has been collected.

Intuitively, the 'worm' part can be seen as a 'magic marker' that can connect or remove the connection between two sites. This way the marker draws patterns onto a lattice that corresponds to a configuration, typically of the partition function. This paper is only concerned with whenever the marker reaches the same lattice site that it started on, forming loops, or clusters.

3.6 Hoshen-Kopelman Algorithm

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 [17]. It is a variant of the union-find algorithm and is most easily described through the associated functions, *find* and *union*. The union-find algorithm groups related sites together under a *label*. 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), essentially setting i and j under the same label.

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. This is done by following the steps outlined bellow.

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 [17] 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.

3.7 Graph Dividing Algorithm

In order to calculate the box dimension the lattice is divided into boxes of decreasing size (see Sec. 2.4). For this a new algorithm was written exploiting that all graphs in this thesis had a linear system size of 2^n, where n is some integer. The algorithm could easily be generalized to x^n or using other configurations by a well-chosen stopping length l_0. What follows is a brief overview of the algorithm and for intuition, one step in the recursion can be seen in Fig. (3.1).

For brevity some abbreviations are first 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

The graph dividing algorithm is defined by the following steps:

1. If $l_i \geq l_0$, go to 2, else stop.
2. Save all sites in the current box, starting from 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 Fig. (3.2).

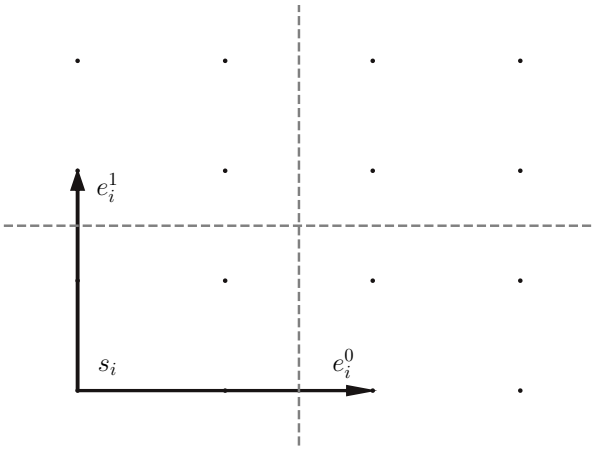


Figure 3.1: One step in the graph dividing algorithm where $l_i = 4$. e_i^0 and e_i^1 are drawn from site s_i. Summed permutations of $\{e_i^0, e_i^1\}$ give the starts for the next boxes. The next iteration of boxes are shown via the dividing dotted lines.

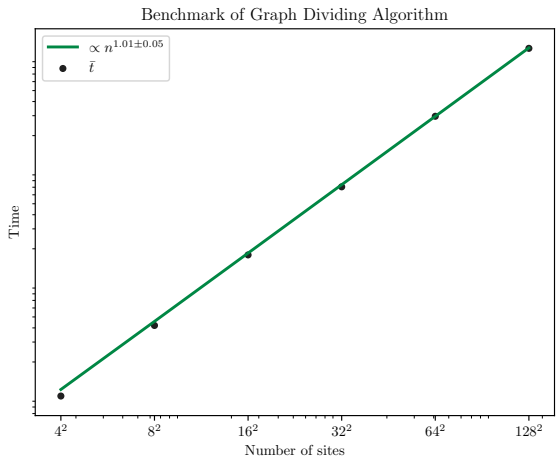


Figure 3.2: Loglog plot of a benchmark of the graph dividing algorithm. The y-axis show the time taken to perform one full graph divide normalized with the smallest time value.

3.8 Error Estimation

In this thesis a number of error estimation techniques were used to control the statistical uncertainty of the results. In this section these techniques will be reviewed.

3.8.1 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

$$\bar{A} = \frac{1}{N}\sum_{i=1}^{N} A_i \tag{3.5}$$

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.

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

where the fact that A_i are random samples from the distribution of A was used in Eq. (3.8).

The standard deviation of this estimate can be calculated as

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

or

$$\sigma_{\bar{A}} = \frac{\sigma_A}{\sqrt{N}} \tag{3.15}$$

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

3.8.2 Bootstrap Error Analysis

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.

3.9 Optimization

All simulation code were implemented by the author. Hence, a large part of this project was spent optimizing the simulations. To fascilitate this process, the Google C++ framework *benchmark* was used for measuring and comparing running times of different algorithms. Only the C++ code was optimized in this way 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.

3.9.1 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. However, it is much slower than a single array, or a squashed graph, most likely due to cache misses[18]. 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

$$n = x + yL + zL^2 + ... \tag{3.16}$$

where L is the linear system size.

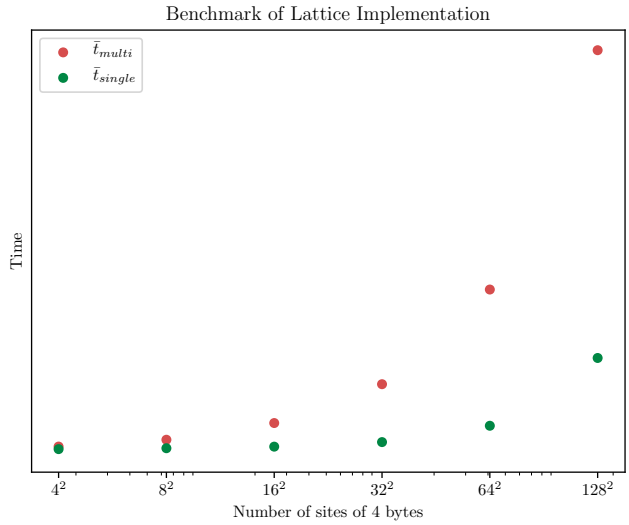


Figure 3.3: Plot of a sweep benchmark of two different lattice implementations. The red dots labeled $\bar{t}_{multi}$ is an array of arrays, while the green dots labeled $\bar{t}_{single}$ is a single array. The y-axis show the time taken to perform one full graph divide normalized against the maximum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

3.9.2 Saving Warmed Up Graphs

In order to produce valid results from a simulation, a physical system must be sampled. Since there is no reliable way to guess a physical state of a system at the start of a simulation, each system must be 'warmed up' to produce valid results. One way of doing this is to iteratively update the system until measurements of the quantity of intereset is sufficiently stable. This is then assumed to be a physical state. A representation of the graph is stored on disk to be used as a starting point for following simulations. This greatly helped with performance as it decreased the simulation times.

In this thesis, the graph was saved to a text file, that could then be parsed with Python to plot the data, or C++ to simulate the system further. The drawback of this solution is the need for a self-written parser. Since the simulation and plotting were done in two different programming languages, the better solution would be a combined interface such as a database.

3.10 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 (Sec. 3.10.1) and regression testing (Section 3.10.2). 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*.

3.10.1 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. Combining multiple units into one test is called integration testing. This assumes that each unit is correct by itself, and it was used to more closely ressemble the actual simulation.

3.10.2 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 was verified after any modification was done.

Chapter 4

Results

As a first step, the susceptibility of an Ising lattice was examined as can be seen in Fig. (4.1). This is done by measuring the average number of steps taken from each loop in the Worm algorithm. This corresponds to sampling the correlation function, that is related to the susceptibility as $\chi \propto \sum G_{ij}$, where G_{ij} is the correlation function between site i and j.

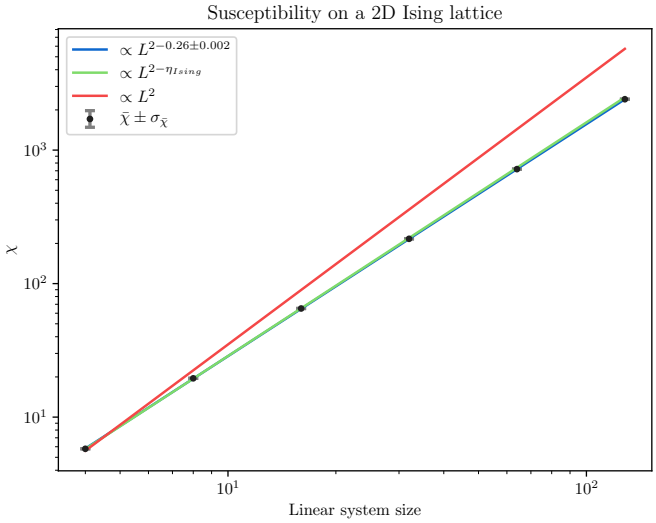


Figure 4.1: Scaling of susceptibility at T_c on an Ising lattice of varying sizes. The measured critical exponent $\eta = 0.262(2)$ is compared to the theoretical value $\eta_{Ising} = 0.25$ [9].

Sampling this correlation function is done for a sequence of different system sizes, $L = 2^n = 4,\ 8,\ 16,\ \ldots,\ 128$, at the critical temperature $T_c = 2/\ln(1+\sqrt{2}) \approx 2.27$. According to the known exact solution [9], the expected scaling exponent for the Ising susceptibility is $\eta_{Ising} = 0.25$, and is computed here to be $\eta = 0.262(2)$.

Figure (4.2) displays the box dimension of the largest cluster on a 128^3 Ising cluster. The largest cluster is found by first labeling all clusters with the Hoshen Kopelman algorithm, and then isolating the one with the largest number of links. The graph is then divided into a set of boxes with decreasing sidelength, as showed in the x-axis of Fig. (4.2). The box dimension is then calculated as

$$d = \lim_{\epsilon \to 0} \frac{\ln N(\epsilon)}{\ln 1/\epsilon} \tag{4.1}$$

where $N(\epsilon)$ is the number of boxes needed to cover the cluster. The green line shows the theoretical value for these clusters as $D_H = 1.375$ [3], and is computed here to be $D_H = 1.3519(5)$. This illustrates that the achieved accuracy of the box dimension is not very good and a better measure is needed. This is discussed next.

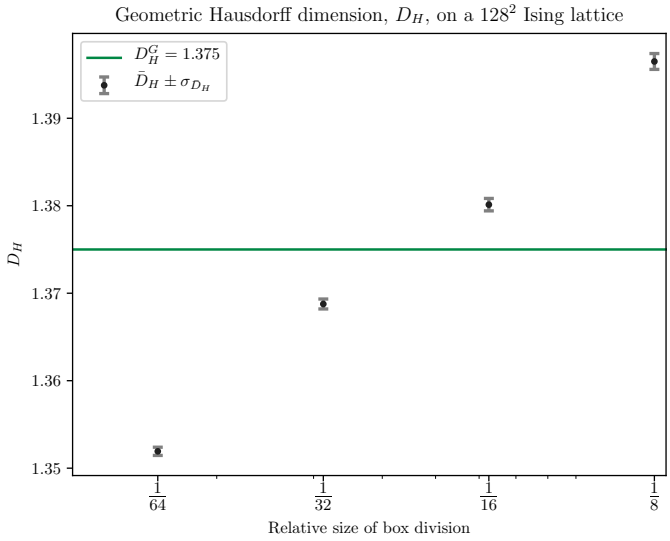


Figure 4.2: Hausdorff dimension of the maximum loop length on a 128^2 Ising lattice at T_c using the box dimension. The x axis shows the size of one box relative to the side length of the lattice. The green line indicates the theoretical dimension of the geometric Ising cluster, $D_H^G = 1.375$ [3]. Comparing to the smallest box size as $D_H = 1.3519(5)$.

A similar approach is used in Fig. (4.3), where the scaling dimension of the largest cluster is examined. The largest cluster in terms of number of links is determined over a sequence of system sizes at T_c. This is then fitted to produce a scaling exponent of $D_H = 1.386(2)$ compared to the theoretical value $D_H^G = 1.375$ [3]. This agrees much better with the theoretical value than the box dimension estimate.

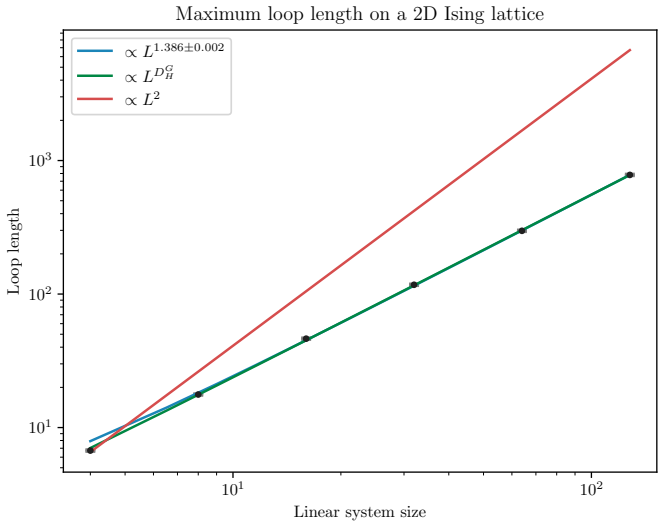


Figure 4.3: Log-log plot of the maximum loop length at T_c on an Ising lattice of varying sizes. The measured scaling factor is 1.386(2) compared to the theoretical Hausdorff dimension, $D_H^G = 1.375$ [3]. Good agreement is obtained.

Figure (4.4) displays a comparison between the two methods showed in Fig. (4.2-4.3) together with the dimensions of random and self avoiding walks. The dimensions computed with the scaling and the box counting method show that the shape of the Ising cluster is similar to that of the self avoiding walk. This can be seen further by examining Figure (4.5), where an isolated largest cluster on a 128^2 Ising lattice is shown.

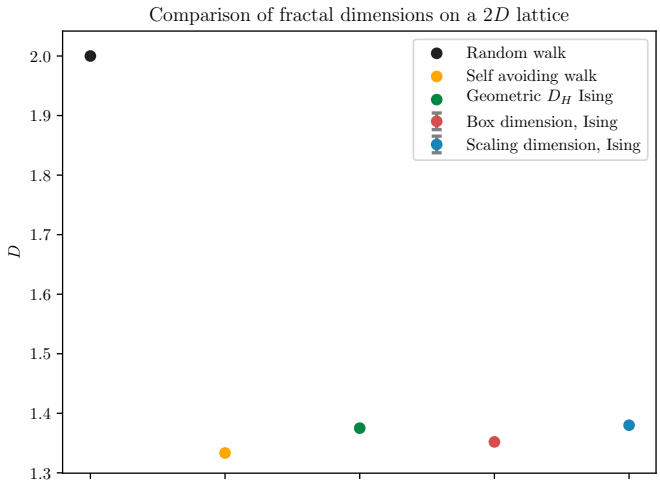


Figure 4.4: Comparison between different algorithms and ways of calculating the fractal dimension on a $2D$ lattice. The random walk has a dimension of 2, the self avoiding walk has $4/3$ [19]. The geometric Hausdorff dimension has an exact value of 1.375 [3]. This is then estimated using the worm algorithm and calculated using both the box counting method and the scaling dimension method explained in Sec. 2.4.1.

The difference between the self avoiding walk and the Ising cluster is that the latter allows the path to cross itself, creating a structure ressembling 'twisted loops' as can be seen in Fig. (4.5). This might indicate a new type of self avoiding loop, where self crossing is allowed.

Figure (4.6) displays a purely self avoiding walk. The plot show some similarities to Fig. (4.5) in that they both expand away from itself. The self avoiding walk never crosses itself and therefore does not fill the space in its path, resulting in a lower fractal dimension. This can be seen as the coarseness is lower in Fig. (4.6) compared to Fig. (4.5).

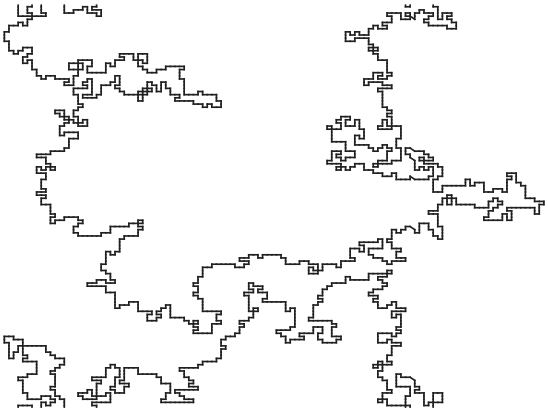


Figure 4.5: Isolated largest cluster on a 128^2 Ising lattice with periodic boundary conditions.

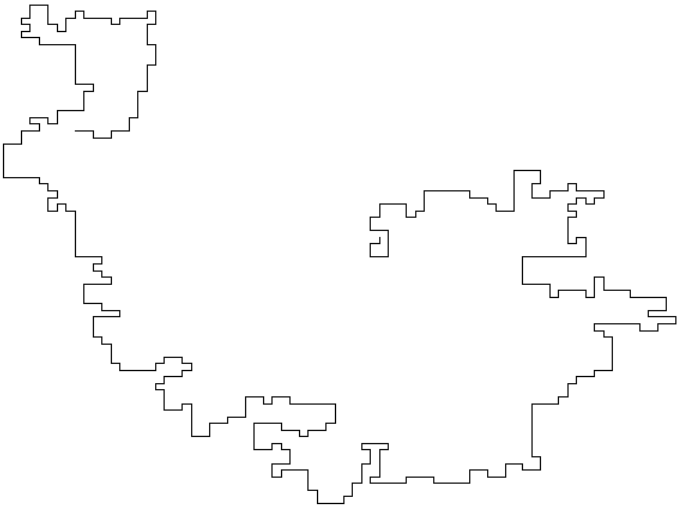


Figure 4.6: Self avoiding walk.

Next, the XY model was examined. To simplify the algorithm, the Villain approximation [7] was used. This gives a new expression for the energy as

$$E = \frac{1}{2}\sum_i J_i^2 \tag{4.2}$$

where J_i is the flux associated with site i. However, in this approximation, the critical temperature is shifted due to the fact that the acceptance probability is changed by the new expression of the energy. One way of determining the new critical temperature is to measure the superfluid density, as it should fall to zero as the temperature is shifted from above, to below T_c. This is done by measuring the winding number, as it relates to the superfluid density as

$$\langle W^2 \rangle \propto \frac{L}{T}\rho_s \tag{4.3}$$

for a $3D$ XY lattice. Fig. (4.7) show the overall structure of the average winding number squared, plotted for a range of system sizes and temperatures.

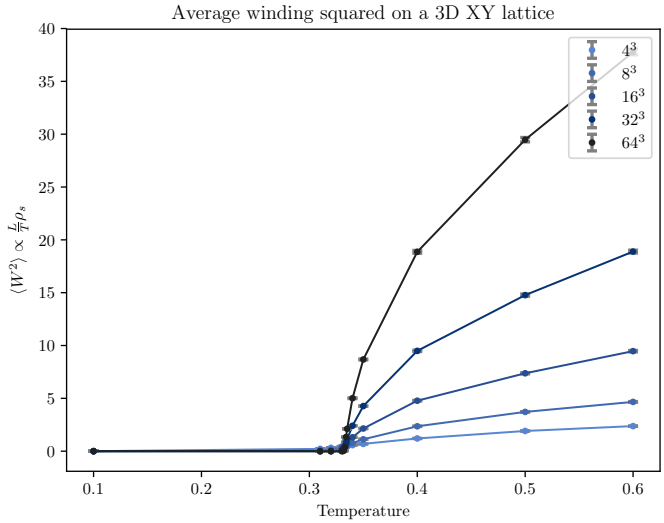


Figure 4.7: Average winding number squared, $\langle W^2 \rangle \propto \rho_s$, plotted on a 3D XY lattice of varying sizes. The overall structure of the average winding number squared as a function of the temperature is shown.

The intersection between all system sizes of average winding number squared is then determined in Fig. (4.8). The inset, together with the red line shows $T_c = 0.3331(1)$ taken as the weighted average of all intersections with the system sizes as weights. Note here that the temperature in the Villain approximation goes as one over the real temperature [7], giving a non-zero superfluid density for high 'temperature'.

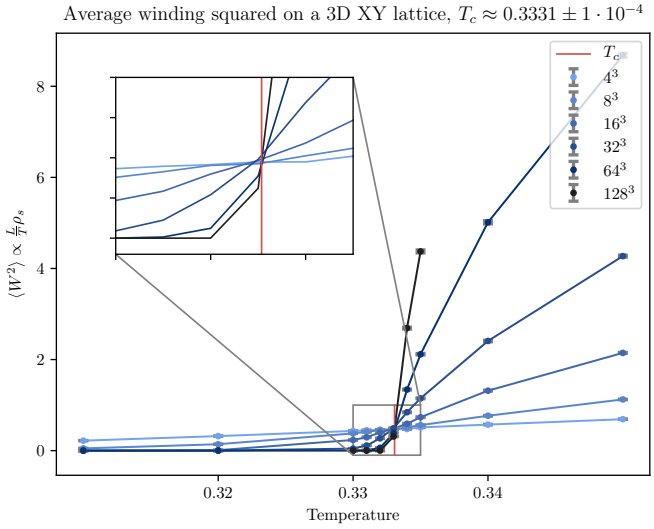


Figure 4.8: Average winding number squared, $\langle W^2 \rangle \propto \rho_s$, plotted on a 3D XY lattice of varying sizes. Due to the duality translation the temperature scale is inverted such that $\rho_s \neq 0$ for $T > T_c$, and T_c is translated from ≈ 2.2 [20] to ≈ 0.333 indicated by the intersection. By taking the weighted average of the intersections the critical temperature can be estimated to $T_c = 0.3331(1)$.

When the critical temperature has been computed, the box counting method can be applied to the XY system. Figure (4.9) displays the Hausdorff dimension as a function of the relative size of the box. The smallest box indicates the most exact value, $D_H = 1.7747(8)$, and is compared to data from Prokof'ev and Svistunov, $D_H^{PS} = 1.765(2)$ [4], and Hove, Mo and Sudbo, $D_H^{HMS} = 2.287(2)$ [2]. Good agreement with Prokof'ev and Svistunov, while Hove et al. deviates. This discrepancy is presumably caused by the different quantity studied by Hove et al., namely vortex loops and phase representations.

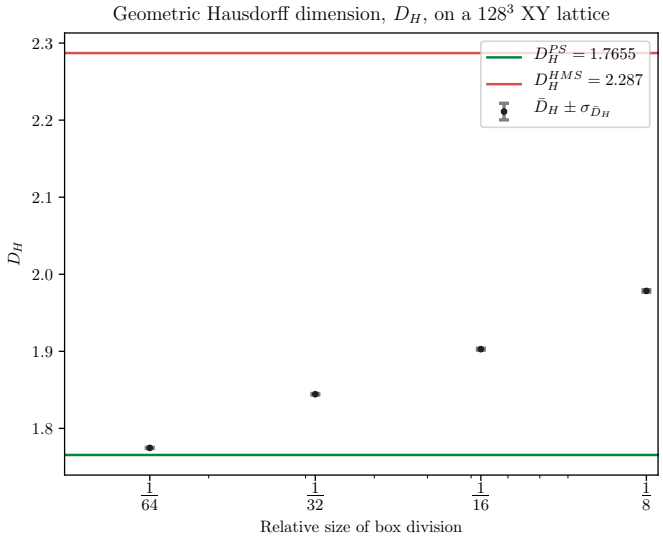


Figure 4.9: Hausdorff dimension of the maximum loop length on a 128^3 XY lattice at T_c using the box dimension. The x axis shows the size of one box relative to the side length of the lattice. Assuming that the smallest box size gives the best approximation, the result is $D_H = 1.7747(8)$. This is compared to data from Prokof'evs and Svistunovs paper $D_H^{PS} = 1.765(2)$ [4] and Hove, Mo and Sudbo's paper $D_H^{HMS} = 2.287(2)$ [2].

A comparison without the extra data from the larger box sizes is shown in Fig. (4.10). The measured dimension from the box counting method closely ressembles that of Prokof'ev and Svistunov.

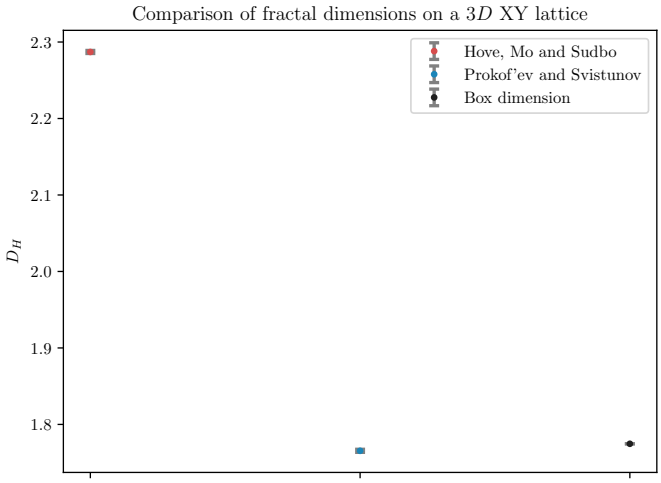


Figure 4.10: Comparison between results from different papers of the Hausdorff dimension of the maximum loop length on a XY lattice at T_c. The result from this thesis is labeled 'box dimension' and yields the result $D_H = 1.7747(8)$. This is compared to data from Prokof'evs and Svistunovs paper $D_H^P = 1.7655 \pm 2 \cdot 10^{-3}$ [4] and Hove, Mo and Sudbo's paper $D_H^S = 2.287 \pm 2 \cdot 10^{-3}$ [2].

Finally, the total energy of the XY lattice is plotted for a sequence of system sizes at T_c. The total energy is calculated in the Villain approximation as shown in Eq. (4.2). This scaling coefficient of the energy here is 2.91(9), and is therefore close to scaling as the spatial dimension of the system. This is expected since

$$E = L^d(at^{1-\alpha} + b) \propto L^d, \text{ at } T = T_c \tag{4.4}$$

where $t = T - T_c$, and a and b are some unknown constants.

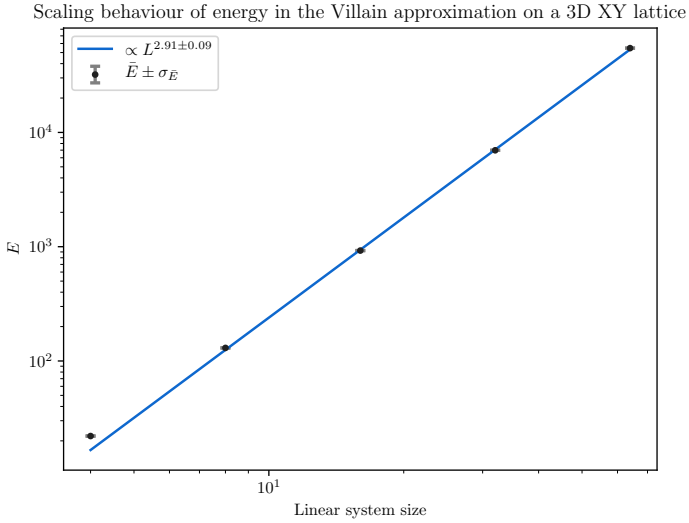


Figure 4.11: Log-log plot of the energy at T_c on an 3D XY lattice of varying sizes. The measured scaling factor is 2.91(9). The energy in the Villain approximation is proportional to the sum of the squares of flux flowing through the lattice.

Chapter 5

Summary and discussion

5.1 Summary

The scaling exponent $\eta_{Ising} = 0.25$ of the susceptibility on a $2D$ Ising lattice was examined with the Worm algorithm. This gave close results of $\eta = 0.262(2)$. Two different ways of determining the Hausdorff dimension of largest clusters at T_c using the Worm algorithm were examined. The scaling method examines the behaviour of the largest cluster as the system size increases. This gave the closest results to the geometrical Hausdorff dimension for a $2D$ Ising cluster of $D_H = 1.386(2)$ compared to the analytical answer $D_H = 1.375$ [3].

The second method was to approximate the Hausdorff dimension using the box dimension. This gave close results, $D_H = 1.35193(5)$, to the scaling method but diverges from the correct result for small boxes. The resulting Hausdorff dimension of the 2D Ising lattice lies between the fractal dimension of a self avoiding walk with $D_{SAW} = 4/3$ [19], and a pure random walk with $D_{RW} = 2$. This suggests a new type of self avoiding walk where 'twisted loops' (see Sec. 5.2) can occur. A full comparison can be seen in Fig. (4.4).

The box counting method was then applied to a $3D$ XY lattice and gave the result $D_H = 1.7747(8)$. This was then compared to previous results given by Hove, Mo and Sudbo as $D_H^S = 2.287(2)$ [2], and the result from the article disputing that claim by Prokof'ev and Svistunov with the value $D_H^P = 1.7655(2)$ [4]. This strengthens the claim by Prokof'ev and Svistunov. Due to time constraints the scaling method was not applied to the $3D$ XY model, which could have improved the claim of this thesis.

To simplify the simulations, the Villain approximation was used (see Sec. 2.2.3). The resulting new critical temperature was calculated by measuring the winding number on a range of temperatures. Since the winding number is proportional to the superfluid density, a sharp decrease can be seen around the critical temperature.

By varying the system size, the intersections of all the drops can be taken as the critical temperature and was calculated to be $T_c = 0.3331(1)$.

5.2 Discussion

The Worm algorithm has proven to be a suitable method for generating cluster configurations which can be used for determining the Hausdorff dimension of a $2D$ Ising cluster. Scaling loop lengths gave the closest value $D_H = 1.386(2)$ to the analytical answer $D_H = 1.375[3]$.

Comparing the $2D$ Ising Hausdorff dimension to the fractal dimension of self avoiding walks, $D_{SAW} = 1.33$ [19], and random walks, $D_{RW} = 2$, the Ising Worm algorithm produces something in between. As can be seen in Fig. (4.5) the cluster ressembles a self avoiding walk but with 'twisted loops' (see Figure (5.1)). This implies that this may be a new type of self avoiding walk that may have similar properties.

Figure 5.1: Examples of twisted loops.

In all simulations trying to measure the length of a loop, the choice was made that whenever there is ambiguity of which path to follow, all paths were followed. This avoids the technique of choosing a random path used by other articles [2], and should in the average case return a longer loop. This can be intuitively understood by following the looping line in the number 8. Whenever the crossing in the middle is reached, one has to follow one of the branching paths. If the stop condition is whenever the starting position is reached, there is a chance of only including one of the loops. This was avoided by recursively following all paths until all loops are explored.

The Hausdorff dimension of the $3D$ XY lattice heavily favoured the result of Prokof'ev and Svistunov over Hove, Mo and Sudbo with this thesis producing $D_H = 1.7747(8)$ compared to Prokof'ev and Svistunovs $D_H = 1.7655(2)$ [4]. The result

would have been strengthened by also comparing to the scaling method described in Sec. 2.4.1.

The winding number method provides a way of determining the critical temperature of a XY lattice. A weighted average of all intersections, weighted with the product of the linear system size of the two intersecting lines, was used to calculate the critical temperature. This method was chosen since the larger system was assumed to provide closer results to an infinite size system.

The choice to use the Hoshen Kopelman algorithm for cluster labeling (discussed in Sec. 3.6) reduced the simulation time considerably from the iterative methods used before. This together with a well chosen graph structure (discussed in Section 3.9.1) improved the data collection speed immensely.

Bibliography

[1] S. H. Strogatz, *Nonlinear Dynamics and Chaos With Applications to Physics, Chemistry, and Engineering*, 2 ed. (Westview Press Inc, Boulder, Colorado, 2014).

[2] J. Hove, S. Mo and A. Sudbø, *Hausdorff Dimension of Critical Fluctuations in Abelian Gauge Theories*, Physical Review Letters **85**, 2368 (2000).

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

[4] N. Prokofev and B. Svistunov, *Comment on Hausdorff Dimension of Critical Fluctuations in Abelian Gauge Theories"*, Physical Review Letters **96** (2006).

[5] N. Prokofev and B. Svistunov, *Worm Algorithms for Classical Statistical Models*, Physical Review Letters **87** (2001).

[6] P. M. Chaikin and T. C. Lubensky, *Principles of condensed matter physics* (Cambridge University Press, 1995).

[7] J. Villain, *Theory of one- and two-dimensional magnets with an easy magnetization plane. II. The planar, classical, two-dimensional magnet*, Journal de Physique **36**, 581 (1975).

[8] J. V. José *et al.*, *Renormalization, vortices, and symmetry-breaking perturbations in the two-dimensional planar model*, Physical Review B **16**, 1217 (1977).

[9] M. Plischke and B. Bergersen, *Equilibrium Statistical Physics* (WORLD SCIENTIFIC, 2006).

[10] J. Heinonen, *Lectures on Analysis on Metric Spaces* (Springer New York, 2001).

[11] W. Rudin, *Principles of Mathematical Analysis (McGraw Hill Publishing Company Ltd., 1964), ix 270 pp., 62s.*, Proceedings of the Edinburgh Mathematical Society **14**, 247 (1965).

[12] K. Falconer, *Fractal Geometry* (John Wiley & Sons, Ltd, 2003).

[13] C. Bandt, N. V. Hung and H. Rao, *On the open set condition for self-similar fractals*, Proceedings of the American Mathematical Society **134**, 1369 (2005).

[14] M. Camarda *et al.*, *Methods to determine the Hausdorff dimension of vortex loops in the three-dimensionalXYmodel*, Physical Review B **74** (2006).

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

[16] J. C. Walter and G. Barkema, *An introduction to Monte Carlo methods*, Physica A: Statistical Mechanics and its Applications **418**, 78 (2015).

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

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

[19] T. Vilgis, *Flory theory of polymeric fractals - intersection, saturation and condensation*, Physica A: Statistical Mechanics and its Applications **153**, 341 (1988).

[20] A. P. Gottlob and M. Hasenbusch, *Critical behaviour of the 3D XY-model: a Monte Carlo study*, Physica A: Statistical Mechanics and its Applications **201**, 593 (1993).