UNIVERSITÀ DEGLI STUDI DI PADOVA
Dipartimento di Fisica e Astronomia “Galileo Galilei”
Master Degree in Physics

Final Dissertation

Real-space Renormalization Group Techniques for Lattice Systems

Thesis supervisors
Prof. Amos Maritan
Prof. Samir Suweis

Candidate
Monisha Gopalan

Academic Year 2022/2023

Real-space Renormalization Group Techniques for Lattice Systems

Monisha Gopalan

Abstract

This thesis explores the application of real-space renormalization group (RG) techniques in the study of critical phenomena in statistical physics. Critical phenomena are characterized by diverging length-scales that manifest with the emergence of correlation functions decaying as power laws both at large spatial and temporal distances and more in general by the presence of singularities in the free energy in the thermodynamic limit. This implies that a macroscopic system close to or at criticality cannot be understood in terms of the properties of its finite subparts. The renormalization group provides a general framework to explain the emergence of singularities in the thermodynamic limit using an iterative procedure involving only an analytic recursion equation. Typically, the implementation of RG at Wilson leads to the so-called proliferation of interactions between degrees of freedom, which is difficult to handle. Some kind of approximation, such as perturbation theory or brute force truncation, is used to simplify the analysis and these approximations are justified a posteriori on the basis of the results obtained. This thesis aims to comprehensively explore and analyze existing RG methods, including decimation, Migdal-Kadanoff bond moving approximation, cumulant approximation, and Monte Carlo renormalisation group methods, in order to gain valuable insights into the critical behavior of various lattice models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Critical phenomena are fascinating phenomena that arise in various fields of physics, including condensed matter physics, statistical mechanics, and quantum field theory. These phenomena occur near phase transitions, where a system undergoes a dramatic change in its macroscopic properties, such as magnetization, conductivity, or phase coexistence. The behavior of a system at criticality is characterized by the emergence of scaling laws, universal behavior, and diverging correlation lengths. Understanding the underlying mechanisms and properties of critical phenomena is of great importance in both theoretical and experimental physics.

The renormalization group (RG) technique provides a powerful framework for studying critical phenomena and capturing the universal behavior near phase transitions. Developed by Kenneth Wilson and others in the 1970s [16], the RG technique allows us to systematically coarse-grain a system by integrating out degrees of freedom at different length scales. Through this process, the RG flow captures the essential physics of the system and reveals the critical behavior and fixed points that govern the universality classes. The RG approach has had significant success in explaining and predicting critical phenomena in a wide range of physical systems.

When Wilson first developed the renormalization-group theory of critical phenomena, extensive use of techniques borrowed from quantum field theory was made. In the meantime, the basic ideas of the renormalization group became quite clear, and attempts were made to implement the ideas directly without recourse to field theory. Such efforts resulted in the so-called “real-space renormalization group”. In this approach, calculations are performed directly in position space, in contradistinction to the field-theoretic ϵ-expansion in momentum space. The advent of the real-space renormalization group not only rendered possible a transparent and elegant implementation of the basic ideas of the renormalization group but also introduced new and effective methods of calculations.

The primary goal of this thesis is to explore different real-space renormalization group techniques. By applying these methods to lattice models, such as the Ising model and the Potts model, we aim to analyze their critical behavior, compute critical exponents, and investigate the scaling properties of these models near the critical point. Furthermore, we seek to understand their strengths and limitations in capturing the essential physics of critical phenomena.

This thesis is structured as follows: In Chapter 2, we provide a comprehensive overview of the renormalization group technique, including its theoretical foundations, phase transitions, critical exponents, and scaling properties. We discuss the concept of universality and how the RG framework captures the universal behavior near critical points. Chapter 3 focuses on real-space renormalization methods, discussing various techniques, such as decimation, Migdal-Kadanoff bond moving approximation, Niemeijer-van Leeuwen cumulant approximation, and their application to specific lattice models. In Chapter 4, we delve into Monte Carlo renormalization group methods, exploring the general theory and discuss the algorithms behind Monte Carlo simulations, such as the Wolff Cluster algorithm. Finally, Chapter 5 presents our conclusions and suggests future directions for research.

Through this thesis, we aim to contribute to the understanding of critical phenomena and the computational techniques used in real-space renormalization group analysis. By deeply studying these methods, we hope to shed light on the rich physics underlying phase transitions and provide valuable insights into the behavior of complex systems near criticality. These topics may have implications for various fields, including condensed matter physics, statistical mechanics, and quantum field theory, polymer physics. The versatility and ability to capture emergent phenomena make real-space renormalisation group methods an invaluable tool for understanding complex systems.

Chapter 2

The Renormalization group

2.1 Phase Transitions and Critical Phenomena

Phase transitions are phenomena in which the physical properties of a material change abruptly as a result of small changes in temperature, pressure, or other external parameters. They are a fundamental concept in condensed matter physics, as they describe how materials can change their properties from solid to liquid, or from magnetic to non-magnetic. An order parameter is a measure of the degree of order across the boundaries in a phase transition system.

In magnetic materials such as iron, the magnetism results from the alignment of electron magnetic moments due to the exchange interaction. In the paramagnetic phase, the magnetic moments of the atoms in the material are randomly oriented, resulting in a net magnetic moment of zero. In this phase, there is no long-range order in the orientation of the magnetic moments, and the material does not exhibit any magnetic properties. In contrast, in the ferromagnetic phase, the magnetic moments of the atoms become aligned with each other, resulting in a net magnetic moment that is non-zero. In this phase, there is long-range order in the orientation of the magnetic moments, and the material exhibits magnetic properties such as magnetization and magnetic susceptibility. The transition between the paramagnetic and ferromagnetic phases occurs at a critical temperature called the Curie temperature T_c. The magnetisation M is an order parameter of the paramagnetic-ferromagnetic phase transition.

Phase transitions can be classified into two main categories: first-order transitions and second-order transitions. First-order transitions involve a discontinuity in the order parameter or more generally a discontinuity in one or more partial derivatives of the thermodynamic potentials. Second-order transitions exhibit a continuous change in the order parameter but the second derivatives are discontinuous. A typical critical behavior of a magnetic material

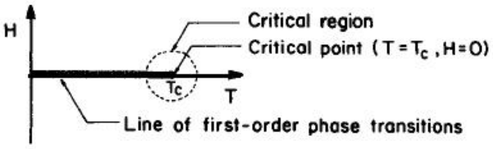
H
Critical region
Critical point (T = T_c , H = 0)
T_c
T
Line of first-order phase transitions

Figure 2.1: Phase diagram of the paramagnetic-ferromagnetic transition [4]

as a function of magnetic field (H) and temperature (T) is illustrated in Figure 2.1. The phase diagram shows a line of discontinuous first-order phase transitions that separates the ferromagnetic and paramagnetic states. Along this line, the magnetization undergoes an abrupt reversal. The line of first-order transitions terminates at the critical point T_c, where the system undergoes a second-order phase transition. Near T_c, a continuous transition occurs from a paramagnetic state ($T > T_c$) to a ferromagnetic state ($T < T_c$). This second-order transition is characterized by a continuous change in magnetization without any latent heat.

2.1.1 Critical exponents

The neighborhood of the critical point is characterized by singular behaviors. The critical exponents are mathematical parameters that describe the behavior of various physical quantities near the critical point. These exponents describe the power-law behavior of various thermodynamic quantities, such as the magnetization, specific heat, susceptibility, and correlation length, and they provide valuable information about the nature of the phase transition. Let us define the adimensional parameter measuring the distance from the critical point, $t = (T - T_c)/T_c$

1. Magnetization (β): The magnetization is a measure of the average spin alignment in the system.
$$M = \left[\frac{\partial F}{\partial H}\right]_T \qquad (T < T_c) \tag{2.1}$$
At the critical point, the magnetization scales with the external magnetic field as a power law with an exponent β:
$$M \sim (-t)^{\beta} \tag{2.2}$$

2. Susceptibility (γ, γ'): The susceptibility measures the response of the system to an external magnetic field.
$$\chi = \left[\frac{\partial M}{\partial H}\right]_{H=0} \tag{2.3}$$
At the critical point, the susceptibility diverges as a power law with the exponents γ, γ':
$$\chi \sim (t)^{-\gamma} \qquad (T > T_c) \tag{2.4}$$
$$\chi \sim (-t)^{-\gamma'} \qquad (T < T_c) \tag{2.5}$$

3. Specific heat (α, α'): The specific heat measures the energy required to heat the system up by a small amount.
$$C = -T\frac{\partial^2 F}{\partial T^2} \tag{2.6}$$
At the critical point, the specific heat diverges as a power law with the exponent α, α':
$$C \sim (t)^{-\alpha} \qquad (T > T_c) \tag{2.7}$$
$$C \sim (-t)^{-\alpha'} \qquad (T < T_c) \tag{2.8}$$

4. Magnetization vs applied field (δ): The critical exponent δ describes how the magnetization scales near the critical point as the magnetic field approaches zero $H = 0$.
$$|M| \sim |H|^{1/\delta} \qquad (T = T_c) \tag{2.9}$$

5. Correlation function(η): The correlation function is a measure of how strongly the spins at two different lattice sites are correlated.

$$\Gamma(x) = \langle s(x)s(0)\rangle - \langle s(x)\rangle \langle s(0)\rangle \tag{2.10}$$

where $s(x)$ is the spin at lattice site at a distance x, and $\langle \cdot \rangle$ denotes the thermal average over all possible spin configurations. The critical exponent associated with the decay of the correlation function is η:

$$\Gamma(x) \sim |x|^{-(d-2+\eta)} \tag{2.11}$$

6. Correlation length (ν, ν'): The correlation length is a measure of how strongly correlated spins are as a function of distance. When $T \neq T_c$,

$$\Gamma(x) \sim \exp(-|x|/\xi) \qquad |x| \longrightarrow \infty \tag{2.12}$$

At the critical point, the correlation length diverges as a power law with the exponents ν, ν':

$$\xi \sim (\tau)^{-\nu} \qquad (T > T_c) \tag{2.13}$$

$$\xi \sim (-\tau)^{-\nu'} \qquad (T < T_c) \tag{2.14}$$

2.1.2 Scaling and Universality

There are a total of nine critical exponents associated with this paramagnetic-ferromagnetic phase transition, but not all of them are independent. The scaling hypothesis provides relations among these exponents, reducing the number of independent ones to two. Let's explore how this is derived. [14]

During a phase transition, the free energy of the system exhibits singularities. The free energy density could be decomposed into two distinct parts and written as:

$$f(t, h) = f_r(t, h) + f_s(t, h) \tag{2.15}$$

where $t = (T - T_c)/T_c$ and $h = (H - H_c)/k_B T$. The term $f_r(t, h)$ is the "regular" part of the free energy density which remains analytic and does not change significantly near the critical point and $f_s(t, h)$ is the "singular" part of the free energy density which captures the non-analytic behaviour of the system near the critical point.

According to Widom's static scaling hypothesis, the singular part of the free energy density, f_s is assumed to be a generalized homogenous function:

$$f_s(\lambda^{p_1} t, \lambda^{p_2} h) = \lambda f_s(t, h) \qquad \lambda \in \mathbb{R}^+ \tag{2.16}$$

where p_1 and p_2 are the degrees of homogeneity. The exponents p_1 and p_2 are not specified by the scaling hypothesis but all the critical exponents of the system can be expressed in terms of p_1 and p_2. Since f_s is a generalized homogenous function, it is always possible to choose λ to remove the dependence on one of the arguments. For instance, choose $\lambda = h^{-1/p_2}$. Then we have,

$$f_s(t, h) = h^{1/p_2} f_s(h^{-p_1/p_2} t, 1) \tag{2.17}$$

and the ratio

$$\Delta \equiv \frac{p_2}{p_1} \tag{2.18}$$

is called the *gap exponent.*

Let us now explore the consequences of Widom's assumption on the critical exponents.

Exponent β: Starting with the scaling hypothesis Eq. 2.16 and taking the derivative with respect to h on both sides

$$f_s(\lambda^{p_1}t, \lambda^{p_2}h) = \lambda f_s(t,h) \tag{2.19}$$

$$\lambda^{p_2}\frac{\partial f_s(\lambda^{p_1}t, \lambda^{p_2}h)}{\partial h} = \lambda\frac{\partial f_s(t,h)}{\partial h} \tag{2.20}$$

$$\lambda^{p_2}M_s(\lambda^{p_1}t, \lambda^{p_2}h) = \lambda M_s(t,h) \tag{2.21}$$

We know that for $h=0$ and $t \to 0^-$, $M_s(t) \sim (-t)^\beta$. So, setting $h=0$, we have

$$M_s(t,0) = \lambda^{p_2-1}M_s(\lambda^{p_1}t, 0) \tag{2.22}$$

Now, we choose $\lambda^{p_1}t = -1$, to eliminate the dependence on t. This implies $\lambda = (-t)^{-1/p_1}$ Hence,

$$M_s(t,0) = (-t)^{(1-p_2)/p_1}M_s(-1,0) \tag{2.23}$$

and by the definition of β, we have

$$\beta = \frac{1-p_2}{p_1} \tag{2.24}$$

Exponent δ Consider the relation

$$\lambda^{p_2}M_s(\lambda^{p_1}t, \lambda^{p_2}h) = \lambda M_s(t,h) \tag{2.25}$$

By setting $t=0$ ($T=T_c$),

$$M(0,h) = \lambda^{p_2-1}M(0, \lambda^{p_2}h) \tag{2.26}$$

Again, following the same property of generalized homogenous functions, we choose $\lambda^{p_2}h = 1$ implying $\lambda = h^{-1/p_2}$, and we have

$$M(0,h) = h^{(1-p_2)/p_2}M(0,1) \tag{2.27}$$

Since $M_s \sim h^{1/\delta}$ as $h \to 0^+$,

$$\delta = \frac{1-p_2}{p_2} \tag{2.28}$$

Now, expressing p_1 and p_2 in terms of β and δ, we write the gap exponent Δ

$$p_1 = \frac{1}{\beta(\delta+1)} \tag{2.29}$$

$$p_2 = \frac{\delta}{\delta+1} \tag{2.30}$$

$$\Delta \equiv \frac{p_2}{p_1} = \beta\delta \tag{2.31}$$

Exponent γ, γ' To obtain the magnetic susceptibility, we take the second derivative of the scaling hypothesis relation with respect to h,

$$\lambda^{2p_2}\chi_T(\lambda^{p_1}t, \lambda^{p_2}h) = \lambda\chi_T(t,h) \tag{2.32}$$

- Case $t \to 0^-$: setting $h = 0$ and $\lambda = (-t)^{-1/p_1}$ we get

$$\chi_T(t,0) = (-t)^{-\frac{2p_2-1}{p_1}} \chi_T(-1,0) \tag{2.33}$$

and by the definition of γ',

$$\chi_T(t,0) \overset{t\to 0^-}{\sim} (-t)^{-\gamma'} \tag{2.34}$$

we get

$$\gamma' = \frac{2p_2 - 1}{p_1} = \beta(\delta - 1) \tag{2.35}$$

- Case $t \to 0^+$: setting $h = 0$ and $\lambda = (t)^{-1/p_1}$ we get

$$\chi_T(t,0) = t^{-\frac{2p_2-1}{p_1}} \chi_T(1,0) \tag{2.36}$$

and by the definition of γ

$$\chi_T(t,0) \overset{t\to 0^+}{\sim} t^{-\gamma} \tag{2.37}$$

we get

$$\gamma = \frac{2p_2 - 1}{p_1} = \beta(\delta - 1) \tag{2.38}$$

Therefore, we have

$$\gamma' = \gamma = \frac{2p_2 - 1}{p_1} = \beta(\delta - 1) \tag{2.39}$$

Exponent α, α' To determine the specific heat, we take the second derivative of the Widom's relation with respect to the temperature t, so that:

$$\lambda^{2p_1} C(\lambda^{p_1} t, \lambda^{p_2} h) = \lambda C(t,h) \tag{2.40}$$

- Case $t \to 0^-$: setting $h = 0$ and $\lambda = (-t)^{-1/p_1}$ we get

$$C(t,0) = (-t)^{-\left(2-\frac{1}{p_1}\right)} C(-1,0) \tag{2.41}$$

and by the definition of α'

$$C(t,0) \overset{t\to 0^-}{\sim} (-t)^{-\alpha'} \tag{2.42}$$

we get

$$\alpha' = 2 - \frac{1}{p_1} \tag{2.43}$$

- Case $t \to 0^+$: setting $h = 0$ and $\lambda = (t)^{-1/p_1}$ we get

$$C(t,0) = t^{-\left(2-\frac{1}{p_1}\right)} C(1,0)$$

and by the definition of α

$$C(t,0) \overset{t\to 0^-}{\sim} t^{-\alpha} \tag{2.44}$$

we get

$$\alpha = 2 - \frac{1}{p_1} \tag{2.45}$$

Critical Exponent	Exact Ising in 2D	Mean field theory
β	1/8	1/2
γ, γ'	7/4	1
δ	15	3
α, α'	0	0
ν, ν'	1	1/2
η	1/4	0

Table 2.1: Critical exponents for the Ising model

Therefore,

$$\alpha' = \alpha = 2 - \frac{1}{p_1} \tag{2.46}$$

If we substitute $p_1 = \frac{1}{\beta(\delta+1)}$ into $\alpha = 2 - \frac{1}{p_1}$, we get:

$$\alpha + \beta(\delta + 1) = 2 \tag{2.47}$$

This is known as the *Griffiths equality.*

If we combine the Griffith equality with the relation $\gamma = \beta(\delta - 1)$, we get the *Rushbrooke's equality*:

$$\alpha + 2\beta + \gamma = 2 \tag{2.48}$$

The concept of universality is closely related to scaling. It refers to the fact that many different physical systems, which may have completely different microscopic details, exhibit the same critical behavior near the critical point. This means that the scaling exponents are the same for all systems that belong to the same universality class, regardless of the specific details of the system. Physical systems can be divided into universality classes according to the dimensionality of space, the number of components of the order parameter, and the symmetry of the system. For example, the Ising model, which is a simple model of a ferromagnet, belongs to the same universality class as models of liquid-gas transitions, superfluid transitions.

The Ising model can be exactly solved only in two dimensions [13], posing analytical challenges in higher dimensions due to increased complexity. To overcome this, mean field theory provides a useful approximation approach by considering average field interactions rather than individual spin interactions. Table 2.1 shows the values for different critical exponents.

It is important to note that while Widom's scaling allows us to determine the exact relations between critical exponents, it does not provide insight into the physical origin of these scaling laws. Additionally, it does not involve correlation lengths or the exponents ν and η. The concept of renormalization group offers a framework to explain these phenomena and provides a deeper understanding of the underlying mechanisms behind the scaling laws.

2.2 General Theory of Renormalization Group

Consider a d-dimensional lattice with lattice spacing l and with a spin $s_i = \pm 1$ assigned on every site. The subscript i labels the N lattice sites. For a subset a of sites, let's define s_a as the product of spins:

$$s_a = \prod_{i \in a} s_i \tag{2.49}$$

The most general Hamiltonian for an Ising-like system can be written as [12]:

$$\mathscr{H}(s) = \sum_a K_a s_a \tag{2.50}$$

where K_a are the interaction parameters that characterizes the interaction between spins on the subset of sites a. Also, the Boltzmann factor $-\beta = -1/k_B T$ has been already absorbed in the Hamiltonian and the actual interaction parameters are $J_a = K_a / k_B T$.

The interaction parameters K_a can be formally obtained from $\mathscr{H}(s)$ using the relation:

$$K_a = 2^{-N} \sum_{\{s\}} s_a \mathscr{H}(s) \tag{2.51}$$

This equation recovering the interaction parameters from the Hamiltonian is of importance and is used in actual numerical calculations. The symbol $\sum_{\{s\}}$ means a summation over all the spin configurations, and it is also represented by the trace operation:

$$\mathrm{Tr} \equiv \sum_{\{s\}} \tag{2.52}$$

Let us consider homogeneous Hamiltonians, which are characterized by having the same coupling constant K_α for all subsets of sites $a \in \alpha$, where α is the class of all subsets of sites that can be identified by a symmetry operation of the lattice.
This leads to a simplified expression for the Hamiltonian:

$$\mathscr{H}(s) = \sum_a K_a s_a = \sum_\alpha K_\alpha \sum_{a \in \alpha} s_a \tag{2.53}$$

In a more intuitive way, this can be expressed as

$$\mathscr{H}(s) = H \sum_i s_i + K \sum_{<i,j>} s_i s_j + \dots \tag{2.54}$$

where the symbol $< i, j >$ means that the sites i and j are nearest neighbours.
The familiar form of the Ising Hamiltonian with nearest-neighbour interactions can be now recognized by identifying :

$$\mathscr{H}(s) = -\beta \mathscr{H} = -\beta h \sum_i s_i - \beta J \sum_{<i,j>} s_i s_j \tag{2.55}$$

$$K_1 = H = h/k_B T \tag{2.56}$$

$$K_2 = K = J/k_B T \tag{2.57}$$

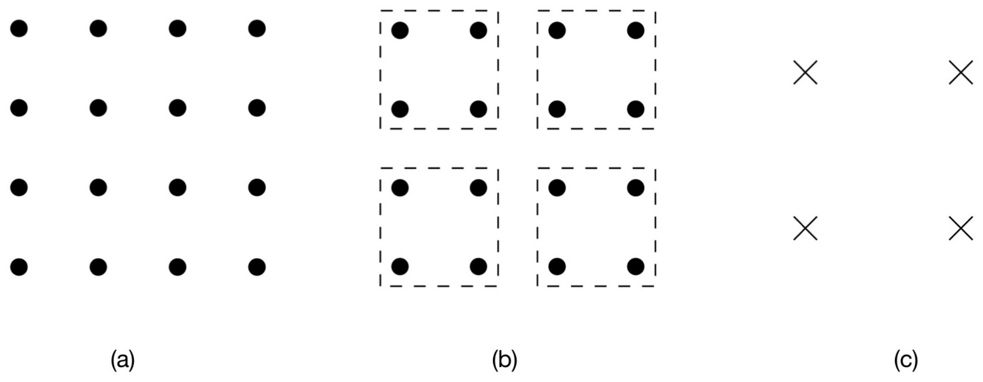

Figure 2.2: Kadanoff's block-spin transformation: (a) original lattice (b) four spins grouped into a block (c) new lattice

2.2.1 Block-spin transformation

Now consider a coarse-graining procedure [5] to the lattice system in which a set of spins are grouped into a block or a cell as shown in Figure 2.2.

Let N' denote a set of cells with cell spins $s'_{i'} = \pm 1$. The index i' refers to the i'-th cell of a lattice isomorphic to the original lattice. All quantities pertaining to the cell system shall be labeled with primes.

We start by writing down the partition function of the original system, in terms of the coupling constants K_a

$$Z_N(K_a) = \sum_{\{s\}} e^{\mathscr{H}(s)} = \operatorname{Tr} e^{\mathscr{H}(s)} \tag{2.58}$$

The block-spin transformation reduces the number of degrees of freedom by a factor l^d:

$$N' = \frac{N}{l^d} \tag{2.59}$$

This is accomplished by making a partial trace over the sites degrees of freedom, keeping the cell degrees of freedom fixed:

$$e^{\mathscr{H}'(s')} = \operatorname{Tr}' e^{\mathscr{H}(s)} \tag{2.60}$$

$$= \operatorname{Tr} P(s', s) e^{\mathscr{H}(s)} \tag{2.61}$$

where Tr′ is the constrained trace, while $P(s', s)$ is the projection operator, which ”incorporates” the constraints and allows us to write an unconstrained trace. The projection operator is constructed so that the coarse-grained degrees of freedom $s'_{i'}$ have the same range of values as s_i.

2.2.2 Renormalization Transformation

The Hamiltonian for the cell system $\mathscr{H}'(s')$ can now be defined by:

$$e^{G+\mathscr{H}'(s')} = \sum_{\{s\}} P(s', s) e^{\mathscr{H}(s)} \tag{2.62}$$

where the projection operator $P(s', s)$ is now referred to as a weight factor that depends on the cell spin configurations $\{s'\}$ and the site spin configurations $\{s\}$.

G is defined by imposing the following condition on the cell spin Hamiltonian $\mathscr{H}'(s')$:

$$\sum_{\{s'\}} \mathscr{H}'(s') = 0 \tag{2.63}$$

and G is independent of the cell spin configurations $\{s'\}$.

The weight factor $P(s', s)$ must satisfy the following three conditions:

1. $P(s', s) \geq 0 \quad \forall s', s$
2. $P(s', s)$ reflects the symmetries of the system
3. $\sum_{\{s'\}} P(s', s) = 1$

Condition 1: This guarantees that $\exp[G + \mathscr{H}'(s')] \geq 0$, so that we can safely identify $\mathscr{H}'(s')$ with the effective Hamiltonian for the cell system.

Condition 2: This implies that the cell spin Hamiltonian $\mathscr{H}'(s')$ has the same symmetries as the site spin Hamiltonian $\mathscr{H}(s)$. We require $P(s', s)$ such that for subsets a' of a certain class α, the cell interaction parameters K'_a are all equal to a certain value K'_α. That is, $\mathscr{H}'(s')$ can again be written as:

$$\mathscr{H}'(s') = \sum_\alpha K'_\alpha \sum_{a' \in \alpha} s'_{a'} \tag{2.64}$$

Condition 3: This guarantees that

$$\begin{aligned} Z_{N'}(K'_a) &\equiv \sum_{\{s'\}} e^{\mathscr{H}'(s')+G} && (2.65) \\ &= \sum_{\{s'\}} \sum_{\{s\}} P(s', s) e^{\mathscr{H}(s)} && (2.66) \\ &= \sum_{\{s\}} e^{\mathscr{H}(s)} \cdot 1 && (2.67) \\ &= Z_N(K_a) && (2.68) \end{aligned}$$

The Eq. 2.62 is called a renormalization transformation. Thus, the partition function is invariant under a renormalization transformation. This also leads to an important relationship between the free energy F' of the cell system and the free energy F of the site system.

$$\begin{aligned} G + F' = \ln \sum_{\{s'\}} e^{G+\mathscr{H}'(s')} &= \ln \sum_{\{s'\}} \sum_{\{s\}} P(s', s) e^{\mathscr{H}(s)} \\ &= \ln \sum_{\{s\}} e^{\mathscr{H}(s)} = F && (2.69) \end{aligned}$$

The three restrictions on $P(s', s)$ still allow an enormous variety in choice for $P(s', s)$, each leading to a different renormalization transformation. A trivial choice would be to take $P(s', s)$ independent of s', and then we have

$$P(s', s) = 2^{-N'} \tag{2.70}$$

In Eq. 2.62, the right-hand side would be independent of s' and therefore we have:

$$\mathscr{H}'(s') = 0 \tag{2.71}$$
$$G = F \tag{2.72}$$

Let us denote the set of coupling constants K_a by $[K]$. Now consider F' and F as functions of the interaction parameters $[K']$ and $[K]$. In the thermodynamic limit, when no long-range forces are present, F' and F assume the form:

$$\begin{aligned} F' &= N' f(K') \\ F &= N f(K) \end{aligned} \tag{2.73}$$

with the same function f in both cases of the cell system and the site system. G also becomes an extensive function $G = Ng(K)$ in the thermodynamic limit with $N/N' = l^d$, where d is the dimensionality of the system.

Substituting these relations into Eq. 2.69,

$$f(K) = g(K) + l^{-d} f(K') \tag{2.74}$$

$g(K)$ is the contribution to the free energy from the degrees of freedom eliminated in the renormalization step. [2]

2.2.3 Recursive Relation

We could see Eq. 2.62 as a renormalization group transformation R_l [3] from the interaction parameters $K_a = [K]$ to the renormalized parameters $K'_a = [K']$,

$$[K'] \equiv R_l[K] \qquad (l > 1) \tag{2.75}$$

This relation is referred to as recursive relation.

R_l describes how the coupling constants change as the length scale is varied. R_l, in general, is a very complicated, non-linear transformation. Since $l > 1$, there is no inverse transformation and the set of transformations R_l for different $l > 1$ form a semi-group:

If we apply two successive transformations R_{l_1} and R_{l_2} on two different length scales l_1 and l_2,

$$[K'] = R_{l_1}[K] \tag{2.76}$$
$$[K''] = R_{l_2}[K'] \tag{2.77}$$
$$= R_{l_2} \circ R_{l_1}[K] \tag{2.78}$$

and hence we have

$$R_{l_1 l_2}[K] = R_{l_2} \circ R_{l_1}[K] \tag{2.79}$$

There is no general way to calculate R_l. Many different RG transformations can be constructed for a given problem. The above framework captures the essence of the Kadanoff block spin transformation, but it differs in that it allows for the generation of new operators during the RG transformation. While the formalism is exact, it may not be immediately useful. An important observation is that while the calculation of the transformed coupling constants $[K']$ as functions of the original couplings $[K]$ can be challenging, these functions are expected to be analytic. This is because the RG transformation integrates over only a finite number of degrees of freedom. The strength of the RG approach lies in the fact that approximating the transformed couplings $[K']$ is generally easier than directly evaluating the partition function itself. Nevertheless, to fully eliminate all degrees of freedom in a thermodynamic system in the thermodynamic limit $N \longrightarrow \infty$, an infinite number of iterations of the RG transformations is required. It is through this iterative process that singular behavior can arise, leading to the emergence of critical phenomena.

After n iterations of the RG transformations, the system undergoes a coarse-graining process, resulting in a length scale of l^n. At this stage, the system's behavior is characterized by the coupling constants $K_0^{(n)}, K_1^{(n)}, \ldots$. As n varies, the system may be thought of as represented by a point moving in a space whose axes are the coupling constants $K_0, K_1, \ldots$. On iterating the RG transformation, a given system represented by its initial set of coupling constants, traces out a trajectory in coupling constant space. The set of all such trajectories, generated by different initial sets of coupling constants generates a renormalisation group flow in coupling constant space. While it is theoretically conceivable for the representative point's trajectory to exhibit complex patterns such as limit cycles or strange attractors, in practice, it is observed that the trajectory tends to converge towards fixed points. In the following section, we will explore how the scaling behaviour is associated with the dynamics near a specific type of fixed point.

2.2.4 Fixed points

Suppose that we know the RG transformation $R_l[K]$. The fixed point of the RG transformation is a point $[K^*]$ in the coupling constant space satisfying

$$[K^*] = R_l[K^*] \tag{2.80}$$

The renormalization transformation can in general have several fixed points, but not every one of them will have physical significance for critical phenomena. The correlation length ξ transforms under R_l as

$$\xi[K'] = \frac{\xi[K]}{l} \tag{2.81}$$

At a fixed point, we have

$$\xi[K^*] = \frac{\xi[K^*]}{l} \tag{2.82}$$

This implies two cases:

$$\xi[K^*] = \begin{cases} 0 & \text{trivial} \\ \infty & \text{critical} \end{cases} \tag{2.83}$$

A fixed point with $\xi = \infty$ is called *critical*, while if $\xi = 0$ it is *trivial.*

Each fixed point has its own basin of attraction or domain: all points $[K]$ in the coupling constant space which lie within the basin of attraction of a given fixed point $[K^*]$ flow towards and ultimately reach the fixed points after an infinite number of iterations of R_l

$$R_l^{(n)}[K] \xrightarrow{n\to\infty} [K^*] \tag{2.84}$$

An important theorem that follows this is that all points in the basin of attraction of a critical fixed point have infinite correlation length. The set of points that forms the basin of attraction of a critical fixed point is also called critical manifold. The critical surface divides the region of space as shown in Fig 2.3.

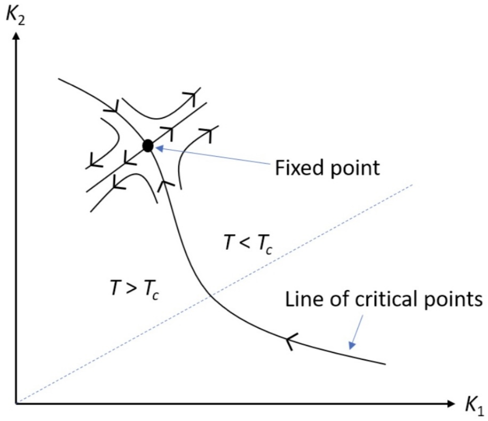

Figure 2.3: Schematic diagram of RG flow. The arrow indicates the direction of flow. The dotted line indicates system with K_1/K_2

As we will see in the next section, the critical fixed points describe the singular critical behaviour. The trivial fixed points, on the other hand, describe the bulk phases of the system. Knowledge of the location and nature of the fixed points of a RG transformation thus enables the phase diagram to be determined, whilst the behaviour of the RG flows near a critical fixed point determines the critical exponents.

2.2.5 Linearized transformation around fixed point

Consider the linearized form of the renormalization transformation, i.e., the matrix

$$\frac{\partial K'_\alpha}{\partial K_\beta} = T_{\alpha\beta} \tag{2.85}$$

If it is linearized at the fixed point, it is denoted by an asterisk

$$\left(\frac{\partial K'_\alpha}{\partial K_\beta}\right)_{K^*} = T^*_{\alpha\beta} \tag{2.86}$$

So, in the neighbourhood of K^*, the transformation can be written as:

$$K'_\alpha - K^*_\alpha = \sum_\beta T^*_{\alpha\beta}(K_\beta - K^*_\alpha) \tag{2.87}$$

The eigenvalues of the matrix $T^*_{\alpha\beta}$ are directly related to the critical exponents. But since the matrix $T^*_{\alpha\beta}$ is not symmetric, it is not guaranteed that its eigenvalues will be real.

It is convenient to introduce a type of normal coordinates in which the transformation takes a simple form. Let $T^*_{\alpha\beta}$ have eigenvalues λ_i with the associated left eigenvectors ϕ^i_α:

$$\sum_\alpha \phi^i_\alpha T^*_{\alpha\beta} = \lambda_i \phi^i_\beta \tag{2.88}$$

Then we construct "normal" coordinates u_i as

$$u_i = \sum_\alpha \phi^i_\alpha (K_\alpha - K^*_\alpha) \tag{2.89}$$

such that the eigenvalue equations are decoupled and u_i transform as

$$u'_i = \sum_\alpha \phi^i_\alpha (K'_\alpha - K^*_\alpha) = \lambda_i \sum_\beta \phi^i_\beta (K_\beta - K^*_\beta) = \lambda_i u_i \tag{2.90}$$

The u_i 's are called Wegner's scaling fields. There are three cases:

1. $\lambda_i > 1$; u_i relevant
2. $\lambda_i < 1$; u_i irrelevant
3. $\lambda_i = 1$; u_i marginal

The values of the relevant fields increase upon performing a transformation, whereas the irrelevant fields decrease. The fixed point corresponds to all $u_i = 0$.

The free energy density

$$f(K) = g(K) + l^{-d} f(K') \tag{2.91}$$

in terms of the scaling fields u_i can be written as:

$$f(u_1, u_2, \ldots) = g(u_1, u_2, \ldots) + l^{-d} f(\lambda_1 u_1, \lambda_2 u_2, \ldots) \tag{2.92}$$

We assume that the scaling fields u_i are regular functions of the interaction parameters K_α and that $g(u_1, u_2, \ldots)$ is a regular function of the scaling fields u_i. Thus, if f has a singular part, it behaves as

$$f_{\text{sing}}(u_1, u_2, \ldots) = l^{-d} f_{\text{sing}}(\lambda_1 u_1, \lambda_2 u_2, \ldots) \tag{2.93}$$

The connection between the eigenvalues λ_i of the matrix $T^*_{\alpha\beta}$ and the critical exponents is most easily established by asking whether a singular behaviour in powers of u_i for the free energy is compatible with Eq. 2.92. For simplicity, let us start by assuming $u_2 = u_3 = 0$. Assume f_{sing} has a powerlike singularity,

$$f_{\text{sing}}(u_1, 0, \ldots) = A u_1^{a_1} \tag{2.94}$$

This leads to

$$A u_1^{a_1} = l^{-d} A (\lambda_1 u_1)^{a_1} \tag{2.95}$$

The exponent a_1 then becomes:

$$a_1 = d \log l / \log \lambda_1 \tag{2.96}$$

The exponent a_1 is a physical quantity and thus λ_1 should vary as a power of l for different renormalization transformations with different l. Therefore, if we write for the relevant eigenvalues (which are assumed to be real and positive)

$$\lambda_1 = l^{y_1} \tag{2.97}$$

the critical exponent y_1 is given by

$$y_1 = d / a_1 \tag{2.98}$$

Now let us consider two scaling fields u_1 and u_2 and determine the singular powers $|u_1|^{a_1}$ and $|u_2|^{a_2}$ allowed by Eq. 2.92. Comparing the exponents on both sides

$$a_1 y_1 + a_2 y_2 = d \tag{2.99}$$

For positive a_i atleast one of the y_i has to be positive. A combination of a relevant scaling field and an irrelevant one leas to a value of a_1 larger than d/y_1 and therefore is a less singular contribution in u_1 than the previously found power d/y_1. So, for two relevant scaling fields, the possible functional forms consistent with Eq. 2.91 are

$$|u_1|^{d/y_1} f_1(u_2/|u_1|^{y2/y_1}) \tag{2.100}$$

$$|u_2|^{d/y_2} f_2(u_1/|u_2|^{y_1/y_2}) \tag{2.101}$$

These are solutions of the scaling relation

$$f_{\text{sing}}(l^{y_1}u_1), l^{y_2}u_2) = l^d f_{\text{sing}}(u_1, u_2) \tag{2.102}$$

Let us now consider the concrete case where $u_1 = t$ is an even scaling field coupled to the temperature and $u_2 = h$ is an odd scaling field coupled to the magnetic field. Then,

$$f_{\text{sing}}(t, h) = |t|^{d/y_T} f_1(h/|t|^{\Delta}) \tag{2.103}$$

$$= |h|^{d/y_H} f_2(t/|h|^{1/\Delta}) \tag{2.104}$$

where Δ is the gap exponent

$$\Delta = \frac{y_H}{y_T} \tag{2.105}$$

Using the scaling laws derived in Section 2.1.2, we can determine all the critical exponents

$$\alpha = 2 - d/y_T \tag{2.106}$$

$$\beta = (d - y_H)/y_T \tag{2.107}$$

$$\gamma = (2y_H - d)/y_T \tag{2.108}$$

$$\delta = y_H/(d - y_H) \tag{2.109}$$

Chapter 3

Real space renormalization methods

3.1 Decimation

Consider a one-dimensional ($d = 1$) Ising model with N spins, denoted as $s_i = \pm 1$. The spins are arranged in a linear chain with nearest-neighbor interactions governed by the coupling constant K. Additionally, we assume that there is no external magnetic field ($H = 0$) and impose periodic boundary conditions, where the first and last spins are also coupled. The Hamiltonian of this system is given by:

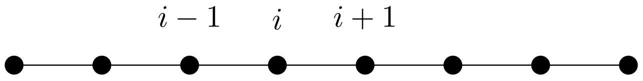

Figure 3.1: One dimensional Ising chain

$$\mathscr{H}(s) = K(s_1 s_2 + s_2 s_3 + \ldots + s_{N-1} s_N + s_N s_1) = K \sum_{i=1}^{N} (s_i s_{i+1}) \tag{3.1}$$

The basic idea is to find a mapping that reduces the number of degrees of freedom in the system by a factor of l, while preserving the partition function:

$$Z_{N'=N/l}(K') = Z_N(K) \tag{3.2}$$

There are several possible mappings, $\{s\} \mapsto \{s'\}$, that satisfy this condition. The choice of transformation is guided by the desire for simplicity in the resulting RG.

When $l = 2$, one option is to group neighboring spins into pairs and define the renormalized spin as their average. However, this ”majority rule” introduces ambiguity as the renormalized spin can have three possible values $(0, \pm 1)$, whereas the original spins are ± 1. This ambiguity can be resolved by assigning one of the two spins, for example $s'_i = s_{2i-1}$. Such an RG procedure effectively removes the even-numbered spins ($s_i = s_{2i}$) and is commonly referred to as *decimation.* [7] So, a coarse graining of $l = 2$ can be obtained by summing over the spins positioned at the even sites.

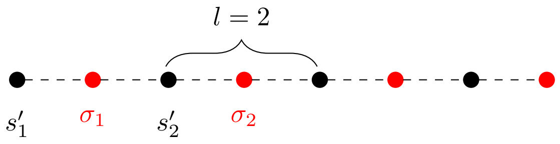

Figure 3.2: Decimated 1-dim Ising chain

Let us relabel the even sites as $s_{2i} = \sigma_i$ and odd sites as $s_{2i-1} = s'_i$. We can split the summation over all spin site configurations into two summations over all the even sites and the odd sites. The partition function can be now written as:

$$
\begin{aligned}
Z_N(K) &= \sum_{\{s\}} e^{K\sum_{i=1}^{N}(s_i s_{i+1})} && (3.3)\\
&= \sum_{\{s\}} \prod_{i=1}^{N} e^{K(s_i s_{i+1})} && (3.4)\\
&= \sum_{\{s'\}} \sum_{\{\sigma\}} \prod_{i=1}^{N/2} e^{K(s'_i \sigma_i)+K(\sigma_i s'_{i+1})} && (3.5)\\
&= \sum_{\{s'\}} \prod_{i=1}^{N/2} e^{K'(s'_i s'_{i+1})} && (3.6)\\
&= Z_{N'}(K') && (3.7)
\end{aligned}
$$

As each σ_i contributes in the same way, we can just consider σ_1. If we average over all possible values of σ_1:

$$
\begin{aligned}
e^{K'(s'_1 s'_2)} &= \sum_{\{\sigma_1 = \pm 1\}} e^{K(s'_1\sigma_1)+K(\sigma_1 s'_2)} && (3.8)\\
&= e^{K(s'_1\cdot 1)+K(1\cdot s'_2)} + e^{K(s'_1\cdot(-1))+K((-1)\cdot s'_2)} && (3.9)
\end{aligned}
$$

The different possible combinations of (s'_1, s'_2) and the different values of the above expression are summarized in the Table 3.1

	s'_1	s'_2	$e^{K'(s'_1 s'_2)}$	
(a)	$+1$	$+1$	$e^{K'}$	$e^{2K} + e^{-2K}$
(b)	-1	-1	$e^{K'}$	$e^{-2K} + e^{2K}$
(c)	$+1$	-1	$e^{-K'}$	2
(d)	-1	$+1$	$e^{-K'}$	2

Table 3.1: Different possible combinations of (s'_1, s'_2)

Multiplying the expressions $\frac{(a)\times(b)}{2\cdot(c)}$, we get:

$$
e^{4K'} = \left(\frac{e^{2K} + e^{-2K}}{2}\right)^2 = \cosh^2(2K) \qquad (3.10)
$$

and we obtain the RG equation:

$$
K' = \frac{1}{2}\log\left(\cosh(2K)\right) \qquad (3.11)
$$

By rearranging we have:

$$e^{2K'} = \cosh(2K) = 2\cosh^2(K) - 1 \tag{3.12}$$

$$e^{2K'} - 1 = 2\Big(\cosh^2(K) - 1\Big) = 2\sinh^2(K) \tag{3.13}$$

$$e^{2K'} + 1 = 2\cosh^2(K) \tag{3.14}$$

Hence,

$$\frac{e^{2K'} - 1}{e^{2K'} + 1} = \tanh(K') = \tanh^2(K) \tag{3.15}$$

Thus, we can rewrite the RG equation as:

$$K' = \tanh^{-1}\Big[(\tanh(K))^2\Big] \tag{3.16}$$

Setting $y \equiv \tanh K$,

$$y' = y^2 \tag{3.17}$$

The fixed points of the RG equation are then given by

$$y^* = y^{*2} \tag{3.18}$$

whose solutions are $y^* = 1$ and $y^* = 0$.

Let us consider the two cases separately:

- Case $y^* = 1^-$ $(K \to \infty, T \to 0^+)$:

  Since $\tanh K < 1 \quad \forall K \in \mathbb{R}$, starting from any initial point $y_0 < 1$, the recursion relation $y' = y^2$ makes y smaller every time, moving it towards the fixed point $y^* = 0$

  $$R_l^{(n)}(y_0) \overset{n\to\infty}{\to} 0^+ \tag{3.19}$$

  We can thus conclude that $K^* \to \infty$ is an unstable fixed point.

- Case $y^* = 0^+$ $(K \to 0^+, T \to \infty)$:
  for all $y_0 < 1$ we have

  $$R_l^{(n)}(y_0) \overset{n\to\infty}{\to} 0^+ \tag{3.20}$$

  Hence, $K^* = 0$ is a stable fixed point.

As anticipated, the one-dimensional Ising model does not exhibit a critical fixed point for $T \neq 0$ (as depicted by the trajectory flow in Figure 3.3). It is worth noting that the convergence of the flow towards $T = \infty$ implies that, on large spatial scales, the system can be effectively described by a Hamiltonian with a high temperature. Consequently, the system will always reside in the paramagnetic phase, except when $T = 0$.

In general, if we scale the lattice spacing by a factor λ, then the recursion relation reads:

$$K' = \tanh^{-1}(\tanh K)^{\lambda} \tag{3.21}$$

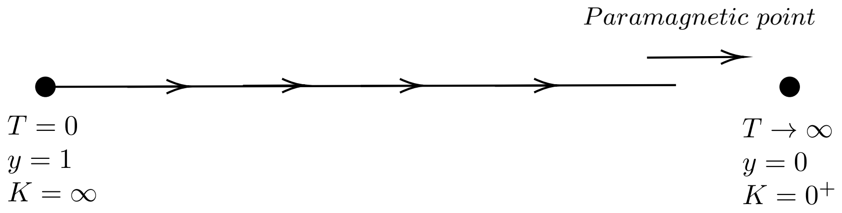

Figure 3.3: One-dimensional flux of trajectories for Ising model for the recursion relation $\tanh(K') = \tanh^2(K)$.

3.1.1 Proliferation of Interactions

In the previous section, we derived the recursion relations for the Ising model in one dimension ($d = 1$), where the determination of the new coupling constants posed no significant challenges and did not introduce new interactions. However, when we consider dimensions higher than one ($d > 1$), the situation becomes more complex. It is observed that the iteration of the RG transformation introduces increasingly complicated couplings, giving rise to what is known as the proliferation of interactions. Resolving this issue requires the use of approximations to simplify the analysis. In the next section, we discuss two such approximations: Migdal-Kadanoff bond-moving approximation and the Niemeijer - van Leeuwen cumulant approximation, and make further progress in our study of the renormalization group.

3.2 Migdal-Kadanoff bond moving approximation

3.2.1 2D Ising Model

Suppose we have an Ising model located on a square lattice. Let us consider an anisotropic lattice with the vertical bonds K_y and horizontal bonds K_x. We perform the following alternate sequence of bond-moving and decimation [11][6](refer Fig. 3.4):

1. Move the vertical bonds horizontally by one unit
$$K'_y = 2K_y \tag{3.22}$$
2. Decimate the rows
$$K'_x = \tanh^{-1}(\tanh K_x)^2 \tag{3.23}$$
3. Move the horizontal bonds vertically by one unit
$$K''_x = 2K'_x = 2\tanh^{-1}(\tanh K_x)^2 \tag{3.24}$$
4. Decimate the columns
$$K''_y = \tanh^{-1}(\tanh K'_y) = \tanh^{-1}(\tanh 2K_y)^2 \tag{3.25}$$

The final recursion relations are:

$$K''_x = 2\tanh^{-1}(\tanh K)^2 = \ln\cosh(2K_x) \tag{3.26}$$
$$K''_y = \tanh^{-1}(\tanh 2K)^2 = \frac{1}{2}\ln\cosh(4K_y) \tag{3.27}$$

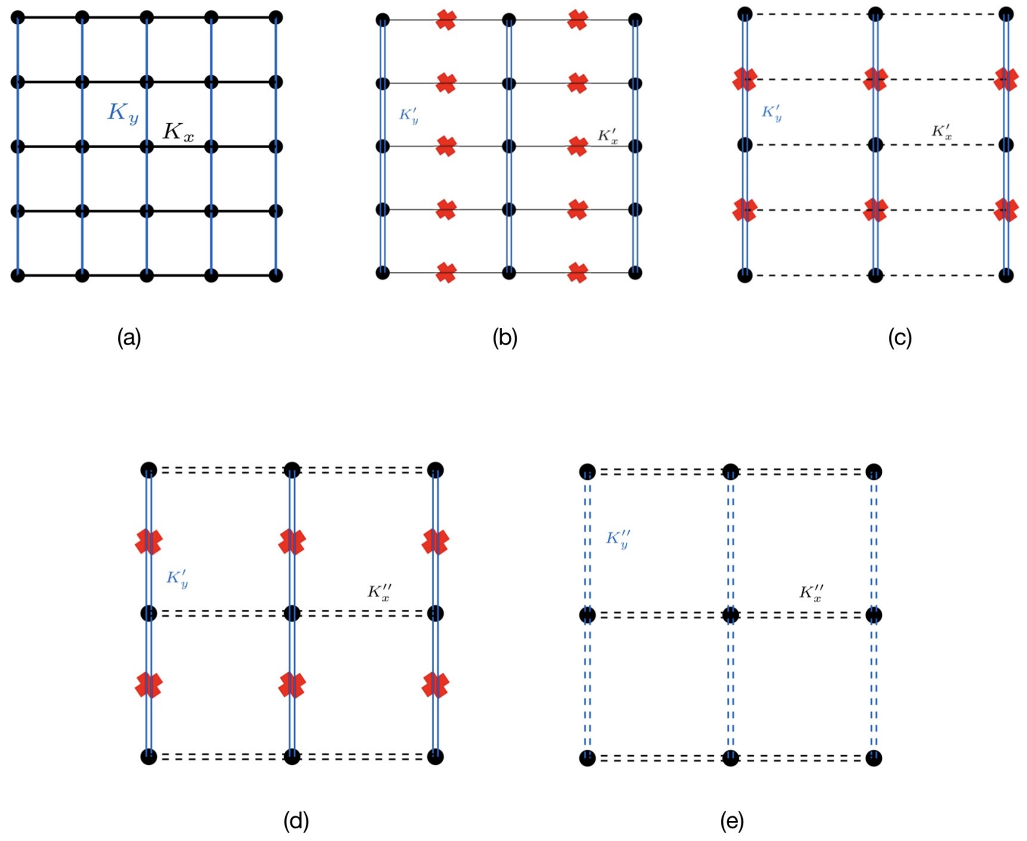

Figure 3.4: Migdal-Kadanoff approximation: (a) original lattice; (b) y-bond moving; (c) decimation in x-direction; (d) x-bond moving; (e) decimation in y-direction

The fixed points are given by:

$$K_x^* = \ln\cosh(2K_x^*) \tag{3.28}$$

$$K_y^* = \frac{1}{2}\ln\cosh\left(4K_y^*\right) \tag{3.29}$$

Case 1: $K_x = K_y = K$

The fixed point relation Eq. 3.27 becomes:

$$K^* = \frac{1}{2}\ln\cosh(4K^*) \tag{3.30}$$

This is similar to the $d = 1$ case. Again, we have $K = 0$ and $K = \infty$ as fixed points:

- For $K \gg 1$ ($T \to 0$): unlike the $d = 1$ we have

$$K' \approx \frac{1}{2}\ln e^{4K} \approx 2K \tag{3.31}$$

  and hence the low temperature fixed point is also stable.

- For $K \ll 1$ ($T \to \infty$): we have

$$K' \approx \frac{1}{2}\ln\left(1 + 16K^2\right) \approx 8K^2 \tag{3.32}$$

so the high temperature fixed point is stable as it is in $d = 1$.

The other fixed point relation Eq. 3.26 becomes:

$$K^* = \ln\cosh(2K^*) \tag{3.33}$$
$$e^{2K^*} = \frac{e^{4K^*} + e^{-4K^*}}{2} \tag{3.34}$$
$$\tag{3.35}$$

Let us take $x = e^{2K^*}$, and now the equation becomes

$$x = \frac{x^2 + x^{-2}}{2} \tag{3.36}$$

Simplifying, we get

$$x^3 - x^2 - x - 1 = 0 \tag{3.37}$$

Solving this cubic equation, gives $x = 1.839$ and

$$K^* = 0.3046 \sim 0.305 \tag{3.38}$$

Linearizing Eq. 3.27 near the fixed point gives

$$l^{y_T} = \left.\frac{\partial K'}{\partial K}\right|_{K=K^*} = 2\tanh 4K^* \sim 1.6786 \tag{3.39}$$

For $l = 2$, this implies

$$y_T = 0.75 \tag{3.40}$$

Case 2: $K_x = 2K_y$.

The couplings K_x and K_y at the fixed point are related by

$$K_x^* = 2K_y^* \tag{3.41}$$

Hence, the values of (K_x^*, K_y^*) are $(0.62, 0.31)$. Linearizing Eqns. (3.26 - 3.27) near the fixed points:

$$\left(\frac{\partial K_x''}{\partial K_x}\right)_{K_x=K_x^*} = 2\tanh 2K_x^* = 2^{y_t} \tag{3.42}$$
$$\left(\frac{\partial K_y''}{\partial K_y}\right)_{K_y=K_y^*} = 2\tanh 4K_y^* = 2\tanh(2K_x^*) = 2^{y_t} \tag{3.43}$$

This also gives $y_t = 0.75$. Note that y_t is the same as the one for the symmetric case. Let us consider the plot in Figure 3.5. We note that all points on the critical surface are critical with a given choice of K_x/K_y. All these $T_c(K_x/K_y)$ flow to the Ising model fixed point (same universality class).

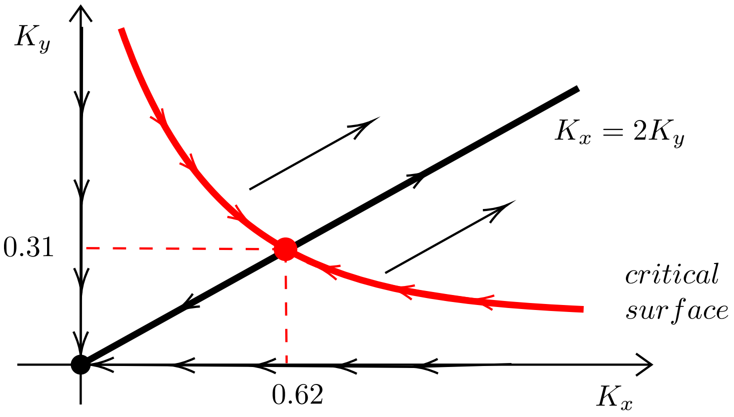

Figure 3.5: (K_x, K_y) plane. The critical surface is represented in red, while the line $K_x = 2K_y$ in black. The intersection point is the critical point $K^* = (0.62, 0.31)$.

3.2.2 2D Potts Model

The q-state Potts model is a generalization of the Ising model where each spin can take one of the q states. Each site of the Potts model can take state $s_i = 1, 2, 3 \ldots, q$. In zero external field, the Hamiltonian of the Potts model is given by

$$\mathscr{H}(s) = K \sum_{<i,j>} \delta_{s_i s_j} \tag{3.44}$$

where the Boltzmann constant k_B and temperature T are absorbed in the coupling K, $< i, j >$ denotes the nearest-neighbor pairs and δ is the Kronecker delta. At $q = 2$, the Potts model becomes the Ising model.

Although the Potts model is relatively well-studied[1], there are interesting unsolved questions regarding its phase transition behavior. Specifically, the Potts model exhibits a first-order phase transition for q values greater than 4, while a second-order phase transition occurs for q values equal to or less than 4. However, the underlying physical mechanism responsible for the first-order phase transition at $q > 4$ remains unclear. To gain insights into the critical phenomena and phase transition of the Potts model, we employ real-space renormalization group transformations on the square-lattice q-state Potts model.

In order to apply the Migdal-Kadanoff bond moving approximation, let us start by finding the recursion relation in $d = 1$ by a $l = 2$ decimation process. Similar to the 1D Ising model, let us relabel the even sites as $s_{2i} = \sigma_i$ and odd sites as $s_{2i-1} = s'_i$. and so we can split the summation over all spin site configurations into two summations over all the even sites and the odd sites. The partition function for the 1D Potts model can be now written as:

$$Z_N(K) = \sum_{\{s\}} e^{K\sum_{i=1}^{N}\delta_{s_i,s_{i+1}}} \tag{3.45}$$

$$= \sum_{\{s\}}\prod_{i=1}^{N} e^{K\delta_{s_i,s_{i+1}}} \tag{3.46}$$

$$= \sum_{\{s'\}}\sum_{\{\sigma\}}\prod_{i=1}^{N/2} e^{K(\delta_{s'_i,\sigma_i})+K(\delta_{\sigma_i,s'_{i+1}})} \tag{3.47}$$

$$= \sum_{\{s'\}}\prod_{i=1}^{N/2} e^{K'\delta_{s'_i,s'_{i+1}}} \tag{3.48}$$

$$= Z_{N'}(K') \tag{3.49}$$

As each σ_i contributes in the same way, we can just consider σ_1. If we average over all possible values of σ_1:

$$e^{K'\delta_{s'_1,s'_2}} = \sum_{\{\sigma_1=1\}}^{q} e^{K(\delta_{s'_1,\sigma_1})+K(\delta_{\sigma_1,s'_2})} \tag{3.50}$$

For $s'_1 = s'_2$,

$$e^{K'\cdot(1)} = e^{K(\delta_{s'_1,\sigma_1})+K(\delta_{\sigma_1,s'_2})} \tag{3.51}$$

$$= e^{K(\delta_{1,\sigma_1})+K(\delta_{1,s'_2})} + e^{K(\delta_{2,\sigma_1})+K(\delta_{\sigma_1,2})} + \dots \tag{3.52}$$

$$+ e^{K(\delta_{q,\sigma_1})+K(\delta_{q,s'_2})} \tag{3.53}$$

Out of q terms, the one term matching the value of $s'_1 = s'_2$ becomes e^{2K}. The other $(q-1)$ terms become $e^0 = 1$. So we have

$$e^{K'} = q - 1 + e^{2K} \quad s'_1 = s'_2 \tag{3.54}$$

For $s'_1 \neq s'_2$,

$$e^0 = e^{K(\delta_{s'_1,\sigma_1})+K(\delta_{\sigma_1,s'_2})} \tag{3.55}$$

$$= e^{K(\delta_{1,\sigma_1})+K(\delta_{1,s'_2})} + e^{K(\delta_{2,\sigma_1})+K(\delta_{\sigma_1,2})} + \dots \tag{3.56}$$

$$+ e^{K(\delta_{q,\sigma_1})+K(\delta_{\sigma_1,q})} \tag{3.57}$$

In this case, out of q terms, the term matching the value of $s'_1 = \sigma_1$ becomes e^K. Similarly for $s'_2 = \sigma_1$. The other $(q-2)$ terms become $e^0 = 1$. So we have

$$1 = q - 2 + 2e^K \quad s'_1 \neq s'_2 \tag{3.58}$$

Now, we divide Eq. 3.54 by Eq. 3.58 to get the recursive realtion:

$$e^{K'} = \frac{q-1+e^{2K}}{q-2+2e^K} \tag{3.59}$$

To find the fixed points, we set $K' = K = K^*$.

Let us take $x = e^{K^*}$. Now the equation we need to solve becomes,

$$x = \frac{q-1+x^2}{q-2+2x} \tag{3.60}$$

$$x^2 + (q-2)x - (q-1) = 0 \tag{3.61}$$

The only meaningful solution to this equation is $x = 1$, resulting in $K^* = 0$. Again, we have $K = 0$ and $K = \infty$ as fixed points:

- For $K \ll 1$ ($T \to \infty$): we have

$$K' = \ln \frac{q + 2K + 2K^2}{q + 2K + K^2} \approx \frac{K^2}{q} \ll K \tag{3.62}$$

  so the fixed point is stable.

- For $K \gg 1$ ($T \to 0$): we have

$$e^{K'} \approx \frac{1}{2} e^K \tag{3.63}$$

$$K' \approx K - \ln 2 < K \tag{3.64}$$

  and hence this fixed point is unstable.

Now let us proceed to $d = 2$ and apply the Midgal-Kadanoff approximation. Consider a Potts model on a 2D square lattice. Performing the bond-moving and decimation as in the Ising model with $l = 2$, we see that moving bonds strengthens the remaining bonds by a factor of 2 in the decimated lattice. Therefore, for the case $K_x = K_y = K$ we have

$$e^{K'} = \frac{q - 1 + e^{2\cdot 2K}}{q - 2 + 2e^{2\cdot K}} \tag{3.65}$$

As before, to find the fixed points, we set $K' = K = K^*$.

Let us take $x = e^{K^*}$. Now the equation we need to solve becomes,

$$x = \frac{q - 1 + x^4}{q - 2 + 2x^2} \tag{3.66}$$

$$x^3(x-2) + (q-2)x - (q-1) = 0 \tag{3.67}$$

By considering the stability of the fixed points at zero and infinite coupling,

- For $K \ll 1$ ($K^* = 0$): we have

$$K' \approx \frac{4K^2}{q} \ll K \tag{3.68}$$

  so the fixed point is stable.

- For $K \gg 1$ ($K^* \to \infty$): we have

$$e^{K'} \approx \frac{1}{2} e^{2K} \tag{3.69}$$

$$K' \approx 2K - \ln 2 \gg K \tag{3.70}$$

  which implies that this fixed point is stable unlike $d = 1$

As a result, there must be a finite K^* fixed point, that separates the flows to the other fixed points.

Let us take $q = 3$. The fixed point relation becomes:

$$(x^3 - 1)(x - 2) = 0 \tag{3.71}$$

This has a solution $x = 2$ and yields a non-trivial fixed point at $K^* = \ln 2 \approx 0.69$.

Linearizing near the fixed point,

$$l^{y_T} = \frac{\partial K'}{\partial K}\bigg|_{K=K^*} = 4\left[\frac{e^{4K^*}}{e^{4K^*}+2} - \frac{e^{2K^*}}{1+2e^{2K^*}}\right] \tag{3.72}$$

$$= \frac{16}{9} \tag{3.73}$$

For $l = 2$, this implies

$$y_T = 0.83 \tag{3.74}$$

This can be compared to the exact values, $K^* = 1.005$ and $y_T = 1.2$

3.3 Niemeijer–van Leeuwen cumulant approximation

This method [12] is based on a splitting of the spin Hamiltonian into a zeroth part $\mathscr{H}_0$ and a small perturbation $\mathscr{V}$

$$\mathscr{H}(s) = \mathscr{H}_0(s) + \mathscr{V}(s) \tag{3.75}$$

$\mathscr{H}_0$ is taken as the set of interactions inside the cells and $\mathscr{V}$ is the set of interactions between the sites of different cells. In other words, in the cumu1ant approximation, the intracell interactions are treated exactly, while the intercell interactions are treated to low order in perturbation theory.

The zeroth order transformation is defined as:

$$\exp[G_0 + \mathscr{H}'_0(s')] = \sum_{\{s\}} P(s', s) \exp[\mathscr{H}_0(s)] \tag{3.76}$$

The zeroth average $<>_0$ is defined as:

$$\langle A \rangle_0 = \sum_{\{s\}} P(s', s) A(s) \exp[\mathscr{H}_0(s) - G_0 - \mathscr{H}'_0(s')] \tag{3.77}$$

The RG transformation equation can be rewritten as

$$G + \mathscr{H}'(s') = G_0 + \mathscr{H}'_0(s') + \ln \langle \exp \mathscr{V} \rangle_0 \tag{3.78}$$

We assume that G_0 and $\mathscr{H}'_0(s')$ can be evaluated and $\mathscr{V}$ is considered small as compared to $\mathscr{H}_0(s)$.

The cumulant expansion for $\langle \exp \mathscr{V} \rangle_0$ is :

$$\ln \langle \exp \mathscr{V} \rangle_0 = \langle \mathscr{V} \rangle_0 + \frac{1}{2!}\left\langle [\mathscr{V} - \langle \mathscr{V} \rangle_0]^2 \right\rangle_0 + \frac{1}{3!}\left\langle [\mathscr{V} - \langle \mathscr{V} \rangle_0]^3 \right\rangle_0 + \dots \tag{3.79}$$

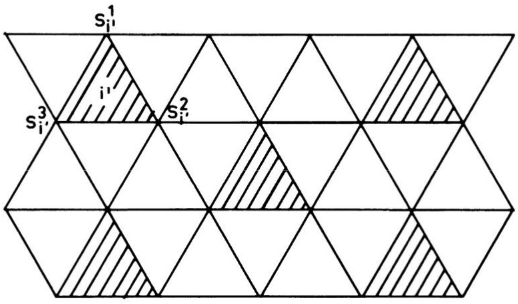

Figure 3.6: Two-dimensional triangular lattice

Let us apply this method to an Ising model on a triangular lattice.

Consider a triangular lattice (Fig.3.6) where cell spins have been placed in the center of three sites. It is noted that the cell spins are again located on a triangular lattice with $l = \sqrt{3}$. Let $\sigma_{i'} \equiv \{s^1_{i'}, s^2_{i'}, s^3_{i'}\}$ denotes the set of sites in cell i'. Define the cell spin $s'_{i'}$ according to the "majority rule":

$$s'_{i'} = \mathrm{sgn}(s^1_{i'} + s^2_{i'} + s^3_{i'}) \tag{3.80}$$

The cell spin values can be only ±1. This can be formally expressed by defining the weight-factor

$$P(s', s) = \prod_{i'} \frac{1}{2}[1 + s'_{i'}(s^1_{i'} + s^2_{i'} + s^3_{i'} - s^1_{i'}s^2_{i'}s^3_{i'})/2] \tag{3.81}$$

$P(s', s) = 1$ when the majority rule is satisfied, and zero otherwise. Each value of cell spin $s'_{i'}$, arises from 4 different configurations of the three neighboring site spins. Let us label the 8 different configurations by $\{\alpha\}$.

The renormalization transformation can be defined as

$$e^{\mathscr{H}'(s')} = \left\langle e^{\mathscr{V}} \right\rangle_0 \sum_{\{\alpha\}} e^{\mathscr{H}'_0(s',\alpha)} \tag{3.82}$$

Let us start with the case of zero external magnetic field $h = 0$. The zeroth order Hamiltonian describing the interactions within a cell can be evaluated as

$$\mathscr{H}'_0 = K \sum_{i'} (s^1_{i'}s^2_{i'} + s^2_{i'}s^3_{i'} + s^3_{i'}s^1_{i'}) \tag{3.83}$$

If N' is the total number of cells in the system, then

$$\sum_{\{\alpha\}} e^{\mathscr{H}_0(s',\alpha)} = z^{N'} \tag{3.84}$$

where z is the partition function of one cell, subject to a given value of $s'_{i'}$:

$$z = \sum_{\sigma_{i'}} e^{K(s^1_{i'}s^2_{i'} + s^2_{i'}s^3_{i'} + s^3_{i'}s^1_{i'})} \tag{3.85}$$

$\{\alpha\}$	$s'_{i'}$	$s^1_{i'}$	$s^2_{i'}$	$s^3_{i'}$	$e^{\mathscr{H}_0}$
(1)	$+1$	$+1$	$+1$	$+1$	e^{3K}
(2)	$+1$	-1	$+1$	$+1$	e^{-K}
(3)	$+1$	$+1$	-1	$+1$	e^{-K}
(4)	$+1$	$+1$	$+1$	-1	e^{-K}
(5)	-1	-1	-1	-1	e^{3K}
(6)	-1	$+1$	-1	-1	e^{-K}
(7)	-1	-1	$+1$	-1	e^{-K}
(8)	-1	-1	-1	$+1$	e^{-K}

Table 3.2: Different possible configurations of spins within a cell, the cell spin value and their contribution to $e^{\mathscr{H}_0}$

Referring Table 3.2,we find that z is independent of $s'_{i'}$ and has the value

$$z = e^{3K} + 3e^{-K} \tag{3.86}$$

Now, the renormalization transformation relation becomes:

$$e^{\mathscr{H}'(s')} = \left\langle e^{\mathscr{V}} \right\rangle_0 z^{N'} \tag{3.87}$$

Using the cumulant expression,

$$\mathscr{H}'(s') = N' \ln z + \langle \mathscr{V} \rangle_0 + \frac{1}{2!} \left\langle [\mathscr{V} - \langle \mathscr{V} \rangle_0]^2 \right\rangle_0 + \mathrm{O}(\mathscr{V}^3) \tag{3.88}$$

The term $N' \ln z$, being the partition function for 3 spins, is clearly regular and does not contribute to singular behaviour.

The interactions between spins in different cells $\mathscr{V}$ can be written as

$$\mathscr{V} = \sum_{i' \neq j'} \mathscr{V}_{i'j'} \tag{3.89}$$

and referring to Fig. 3.7, we have

$$\mathscr{V}_{i'j'} = K(s^2_{i'} + s^3_{i'}) s^1_{j'} \tag{3.90}$$

The thermal average, thus becomes:

$$\langle \mathscr{V}_{i'j'} \rangle_0 = 2K \left\langle s^3_{i'} s^1_{j'} \right\rangle_0 \tag{3.91}$$

Since $\mathscr{H}_0$ by definition, does not couple different cells, the average can be factorised,

$$\langle \mathscr{V}_{i'j'} \rangle_0 = 2K \left\langle s^3_{i'} \right\rangle_0 \left\langle s^1_{j'} \right\rangle_0 \tag{3.92}$$

Let us now calculate the various averages.

$$\left\langle s^3_{i'} \right\rangle_0 = \frac{\sum_{\{\sigma_{i'}\}} e^{\mathscr{H}_0(s', \sigma_{i'})} \cdot s^3_{i'}}{\sum_{\{\sigma_{i'}\}} e^{\mathscr{H}_0(s', \sigma_{i'})}} \tag{3.93}$$

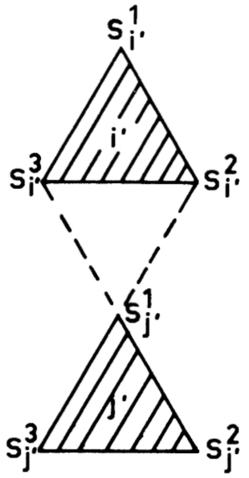

Figure 3.7: Two neighbouring cells in a triangular lattice

For $s'_{i'} = 1$, we find that

$$\left\langle s_{i'}^3 \right\rangle_0 = \frac{e^{3K} + e^{-K}}{e^{3K} + 3e^{-K}} \tag{3.94}$$

whereas for $s'_{i'} = -1$, we find that

$$\left\langle s_{i'}^3 \right\rangle_0 = -\frac{e^{3K} + e^{-K}}{e^{3K} + 3e^{-K}} \tag{3.95}$$

Hence combining these together, we can write,

$$\left\langle s_{i'}^3 \right\rangle_0 = s_{i'}^3 \left[\frac{e^{3K} + e^{-K}}{e^{3K} + 3e^{-K}} \right] = a_1 s_{i'}^3 \tag{3.96}$$

where we define

$$a_1 = \frac{1}{z}(e^{3K} + e^{-K}) \tag{3.97}$$

Similarly, for $s_{j'}^1$, we can write,

$$\left\langle s_{j'}^1 \right\rangle_0 = s_{j'}^1 \left[\frac{e^{3K} + e^{-K}}{e^{3K} + 3e^{-K}} \right] = a_1 s_{j'}^1 \tag{3.98}$$

Thus,

$$\langle \mathscr{V}_0 \rangle = 2a_1^2 K \sum_{<i'j'>} s'_{i'} s'_{j'} \tag{3.99}$$

In summary, the renormalization transformation, to first order in $\mathscr{V}$ is,

$$\mathscr{H}'(s') = N' \ln z + K' \sum_{<i',j'>} s'_{i'} s'_j \tag{3.100}$$

with the recursion relation

$$K' = 2a_1^2 K = 2K \left(\frac{e^{3K} + e^{-K}}{e^{3K} + 3e^{-K}} \right)^2 \tag{3.101}$$

The fixed points of this relation are:

- $K^* = 0$: If $K \ll 1$,
$$K' \approx 2K(2/4)^2 = K/2 < K \tag{3.102}$$
so this fixed point is stable, and has zero correlation length.

- $K^* = \infty$: if $K \gg 1$,
$$K' \approx 2K > K \tag{3.103}$$
This fixed point is also stable with zero correlation length.

- Since both of the above fixed points are stable, there must be atleast one unstable fixed point at finite $K' = K = K^*$. From Eq. 3.101, the fixed point satisfies:
$$\frac{1}{\sqrt{2}} = \frac{e^{3K} + e^{-K}}{e^{3K} + 3e^{-K}} \tag{3.104}$$
This implies
$$\sqrt{2}e^{4K^*} + \sqrt{2} = e^{4K^*} + 3 \tag{3.105}$$
The value of fixed point is given by,
$$K^* = \frac{1}{4} \ln\left(\frac{3 - \sqrt{2}}{\sqrt{2} - 1}\right) \approx 0.3356 \tag{3.106}$$
and this can be compared to the exactly known value of 0.2747 for the triangular lattice.

Linearizing the recursion relation around the non-trivial fixed points,

$$\begin{aligned} l^{y_T} &= \left.\frac{\partial K'}{\partial K}\right|_{K=K^*} \tag{3.107} \\ &= 2\left(\frac{e^{4K^*}+1}{e^{4K^*}+3}\right)^2 + 32K^* e^{4K^*} \frac{e^{4K^*}+1}{(e^{4K^*}+3)^3}\Bigg] \tag{3.108} \\ &\approx 1.624 \tag{3.109} \end{aligned}$$

The thermal eigenvalue for $l = \sqrt{3}$

$$y_T \approx \frac{\ln(1.624)}{\ln\left(\sqrt{3}\right)} \approx 0.883 \tag{3.110}$$

This can be compared to the exactly known value of $y_T = 1$ for the 2D Ising model.

Let us now consider the case $h \neq 0$. Since the fixed point occurs at $h^* = 0$, let us look at how does a small deviation $\delta h = h - h^*$ affects the calculation of $\mathscr{H}'$. Let the change in $\mathscr{H}$ due to a small external field δh be $\delta\mathscr{H}'$. By definition, we have,

$$\begin{aligned} e^{\mathscr{H}'(s')} &= \sum_{\{\alpha\}} e^{\mathscr{H}(s',\alpha)} \tag{3.111} \\ e^{\mathscr{H}'+\delta\mathscr{H}'} &= \sum_{\{\alpha\}} e^{\mathscr{H}(s',\alpha)+\delta\mathscr{H}(s',\alpha)} \tag{3.112} \\ & \tag{3.113} \end{aligned}$$

Subtracting these two equations, and using $e^x = 1 + x + \mathrm{O}(x^2)$,

$$\delta\mathscr{H}'(s') = \frac{\sum_{\{\alpha\}} e^{\mathscr{H}(s',\alpha)}\delta\mathscr{H}(s',\alpha)}{\sum_{\{\alpha\}} e^{\mathscr{H}(s',\alpha)}} \tag{3.114}$$

Also, by definition, [3]

$$\delta\mathscr{H}(s',\alpha) = \delta h \sum_i s_i = \delta \sum_{i'} \sum_{\sigma_{i'}} s'_{i'} \tag{3.115}$$

$$\delta\mathscr{H}' = \delta h' \sum_{i'} s'_{i'} \tag{3.116}$$

Eq. 3.114 implies that to zeroth order in $\mathscr{V}$

$$\delta\mathscr{H}'(s') = \left\langle \delta \sum_{i'} \sum_{\sigma_{i'}} s'_{i'} \right\rangle_0 \tag{3.117}$$

$$= \delta h \sum_{i'} \left\langle s^1_{i'} + s^2_{i'} + s^3_{i'} \right\rangle_0 \tag{3.118}$$

Using Eq. 3.96

$$\left\langle s^1_{i'} \right\rangle_0 = s^1_{i'} a_1 \tag{3.119}$$

we get the recursion relation for the magnetic field:

$$h' = 3h\left(\frac{e^{3K} + e^{-K}}{e^{3K} + 3e^{-K}}\right) \tag{3.120}$$

Near the unstable fixed point, we can write:

$$l^{y_H} = \left.\frac{\partial h'}{\partial h}\right|_{K^*} \tag{3.121}$$

$$= 3\left(\frac{e^{4K^*} + 1}{e^{4K^*} + 3}\right) = \frac{3}{\sqrt{2}} \tag{3.122}$$

The thermal eigenvalue for $l = \sqrt{3}$

$$y_H = \frac{\ln\left(3/\sqrt{2}\right)}{\ln\left(\sqrt{3}\right)} \approx 1.37 \tag{3.123}$$

This can be compared to the exact value of $y_H = 1.875$

Chapter 4

Monte Carlo Renormalisation Group methods

The combination of Monte Carlo simulations and the real space renormalisation group formalism was first suggested by Ma [9] in 1976. The basic idea of this approach is to determine the behavior of the Hamiltonian upon renormalization and by following the 'flow' towards the fixed point Hamiltonian to study critical exponents. By measuring effective interaction parameters between coarse grained blocks of spins, one can extract exponent estimates from this information. Ma's method involved a direct simulation of the fixed-point Hamiltonian, from which he calculated matrix elements for the linearized RG equations. The eigenvalues of these matrices provided estimates of the critical exponents. Ma applied this method to the two-dimensional Ising model with promising results.

However, Ma's method had some limitations that prevented its general application to problems of interest. The direct simulation of the fixed-point Hamiltonian required significant truncation of the coupling constants and involved scanning a large parameter space for the fixed point. A very different approach, was proposed by Swendsen [15], proved to be more effective in finding exponent estimates as it eliminated the need to calculate the renormalized couplings. In this chapter, we first explore Swendsen's approach for MCRG, followed by a brief discussion on the theory of Monte Carlo methods. We then provide a detailed description of our MCRG implementation and present the simulation results.

4.1 Swendsen's Approach

The general Hamiltonian for an Ising-like model given by,

$$\mathscr{H} = \sum_a K_a S_a \tag{4.1}$$

where S_a represents combinations of spins σ_i that are translationally invariant under periodic conditions.

After a renormalization transformation $H^{(n+1)} = R_l H^{(n)}$ with a scale factor l, the asymptotic critical properties are determined by the eigenvalues of the linearized RG transformation matrix, $T_{\alpha\beta}$, in the vicinity of the fixed-point.

$$T^*_{\alpha\beta} = \frac{\partial K^{*(n+1)}_\alpha}{\partial K^{*(n)}_\beta} \tag{4.2}$$

In this approach, the elements of the linearized transformation matrix $T^*_{\alpha\beta}$ are expressed in terms of expectation values of correlation functions at different levels of renormalization. These elements can be obtained by solving the chain rule equation:

$$\frac{\partial\left\langle S_\gamma^{(n)}\right\rangle}{\partial K_\beta^{(n-1)}} = \sum_\alpha \frac{\partial K_\alpha^{(n)}}{\partial K_\beta^{(n-1)}} \frac{\partial\left\langle S_\gamma^{(n)}\right\rangle}{\partial K_\alpha^{(n)}} \tag{4.3}$$

To compute the derivatives in Eq. 4.3, a Monte Carlo simulation of the system at criticality is performed to obtain a sequence of configurations. Each configuration undergoes an RG transformation, and correlation functions are calculated from this sequence of configurations. The derivatives can then be evaluated using the correlation functions with the following identities:

$$\frac{\partial\left\langle S_\gamma^{(n)}\right\rangle}{\partial K_\beta^{(n-1)}} = \left\langle S_\gamma^{(n)} S_\beta^{(n-1)}\right\rangle - \left\langle S_\gamma^{(n)}\right\rangle\left\langle S_\beta^{(n-1)}\right\rangle \tag{4.4}$$

$$\frac{\partial\left\langle S_\gamma^{(n)}\right\rangle}{\partial K_\alpha^{(n)}} = \left\langle S_\gamma^{(n)} S_\alpha^{(n)}\right\rangle - \left\langle S_\gamma^{(n)}\right\rangle\left\langle S_\alpha^{(n)}\right\rangle \tag{4.5}$$

Solving Eq. 4.3 numerically yields the elements of the matrix $T_{\alpha\beta}$, and the eigenvalues of this matrix provide estimates of the critical exponents.

4.2 Monte Carlo Methods

The basic principle of Monte Carlo methods is to simulate the system's behavior by sampling its configuration space according to a given probability distribution. The fundamental quantity of interest are the expectation values of some quantity, obtained as

$$\langle A\rangle = \frac{1}{Z}\sum_{\{s\}} A_s e^{\mathscr{H}(s)} \tag{4.6}$$

where Z is the partition function. Monte Carlo procedures enable us to approximate this sum by summing over a representative sample of configurations:

$$\langle A\rangle \approx \frac{1}{M}\sum_{i=1}^{M} A(s_i) \tag{4.7}$$

The key challenge lies in generating configurations that are weighted according to the Boltzmann distribution. By employing suitable algorithms, we can explore the configuration space and generate a diverse set of representative configurations.

4.2.1 Metropolis algorithm

The basic idea is to create a Markov chain of states, i.e., a sequence of states in which each state only depends on the state immediately preceding it. The algorithm begins with an initial configuration s_i with a nonvanishing Boltzmann factor p_i, which serves as the first member of the Markov chain. The next step is to iteratively create trial configurations s_j from the previous state s_i and decide whether to accept or reject the trial configuration based on a comparison with the previous state. There is thus a transition probability from each state s_i to each state s_j , represented by a transition matrix π_{ij} . The goal to find a transition matrix that yields the equilibrium distribution p_j. Evidently, this matrix must satisfy the condition

$$\sum_i p_i \pi_{ij} = p_j \tag{4.8}$$

To determine the transition probabilities between states, the *detailed balance* condition is imposed, which ensures the balance of transitions between states , i.e., on average the number of transitions from a state i to a state j is balanced by the number of transitions from state j to state i.

$$\sum_i p_i \pi_{ij} = p_j \tag{4.9}$$

The matrix elements π_{ij} are determined by the product of two factors: the a priori probability α_{ij} of generating a trial configuration s_j from a configuration s_i, and the acceptance probability P_{ij} of accepting the trial configuration as the new state. The detailed balance condition can thus be written as

$$p_i \alpha_{ij} P_{ij} = p_j \alpha_{ji} P_{ji} \tag{4.10}$$

For the simplest case where, α_{ij} is symmetric and the condition reduces to

$$p_i P_{ij} = p_j P_{ji} \tag{4.11}$$

which can be rewritten as

$$\frac{P_{ij}}{P_{ji}} = \exp\{(-\beta(E_j - E_i))\} \tag{4.12}$$

The acceptance probability is not uniquely defined by this equation. Metropolis et al [10] proposed the solution

$$P_{ij} = \begin{cases} \exp\{(-\beta(E_j - E_i))\} & \text{if} \quad E_j > E_i \\ 1 & \text{if} \quad E_j \leq E_i \end{cases} \tag{4.13}$$

The trial configuration s_j is generated using a *trial move*, which involves a small displacement of one particle in a random direction for an assembly of N particles. In the case of the Ising model, this corresponds to a spin flip (as illustrated in Fig. 4.1). The acceptance of the trial configuration depends on its energy compared to the original configuration. If the energy of the trial configuration is lower than the original, it is always accepted. However, if the energy is higher, it is only accepted with a probability equal to the ratio of the Boltzmann factor of the new configuration and the Boltzmann factor of the original configuration.

4.2.2 Critical slowing down

The Metropolis algorithm, which performs local updates by flipping a single spin at each step, becomes less effective in exploring the configuration space near critical points due to the presence of long-range correlations. In systems near criticality, the relaxation time τ of thermodynamic properties increases as a power law of the correlation length ξ, characterized by the dynamical critical exponent $z \approx 2$ [8].

$$\tau \propto \xi^z \tag{4.14}$$

The correlation length itself diverges as a power law of the difference between the temperature T of the system and the critical temperature T_c ,

$$\xi \propto |T - T_c|^{-\nu} \tag{4.15}$$

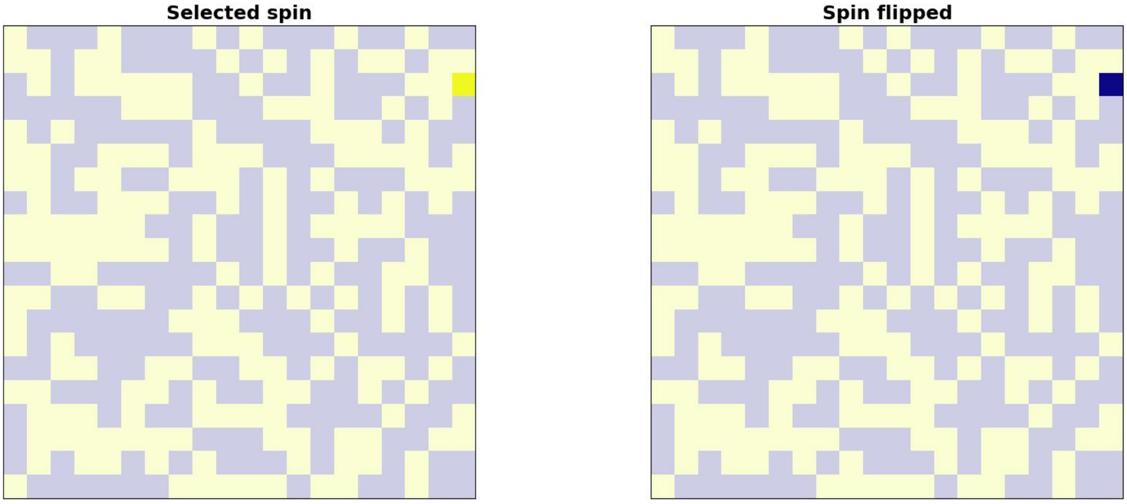

Figure 4.1: Demonstration of single flip in Metropolis algorithm

where ν is a positive exponent.

In finite systems, such as a d-dimensional hypercube of volume L^d, the correlation length is limited by the system size L. Thus, if the temperature approaches T_c , ξ grows according until it reaches a maximum value $\xi_{max} \propto L$, and hence for temperatures sufficiently close to the critical temperature, we have

$$\tau \propto L^z \tag{4.16}$$

We thus encounter a phenomenon called critical slowing down. If a system becomes larger, the correlation time grows very rapidly and it becomes increasingly difficult to generate statistically independent configurations. To overcome the problem of critical slowing down, alternative algorithms and techniques are employed in Monte Carlo simulations. One approach is to use cluster algorithms such as the Wolff algorithm, which exploit the presence of clusters of correlated spins to update multiple spins simultaneously, resulting in more efficient exploration of phase space.

4.2.3 Wolff or Single-Cluster Algorithm

The Wolff algorithm, also known as the single-cluster algorithm, is an efficient approach to perform Monte Carlo simulations for spin systems. It overcomes the problem of critical slowing down by updating spins in clusters rather than individually. The procedure for the Wolff algorithm [17] is as follows:

1. Randomly select a spin i.

2. Add all nearest neighbors j of spin i to the cluster with a probability $p_{ij} = 1 - \exp(-2\beta J)$, provided that spins i and j are parallel, and the bond between i and j has not been considered previously.

3. For each spin j successfully added to the cluster, place it on the stack. Once all neighbors of spin i have been considered for inclusion in the cluster, retrieve a spin from the stack and, in turn, consider all its neighbors for inclusion in the cluster according to step (2).

4. Repeat steps (2) and (3) iteratively until the stack is empty.

5. Once the cluster is complete, invert all spins belonging to the cluster.

A clever trick can be applied to simplify the implementation. In step (2), each added spin j can be immediately inverted, ensuring that a spin is never added more than once. Consequently, step (5) of inverting the cluster spins can be eliminated from the procedure.

The Wolff algorithm is rejection-free, meaning that the cluster is always flipped. Unlike the Metropolis algorithm, the Wolff algorithm does not rely on an acceptance criterion based on the total energy change resulting from a cluster flip. This makes the Wolff algorithm highly efficient for simulations near critical points.

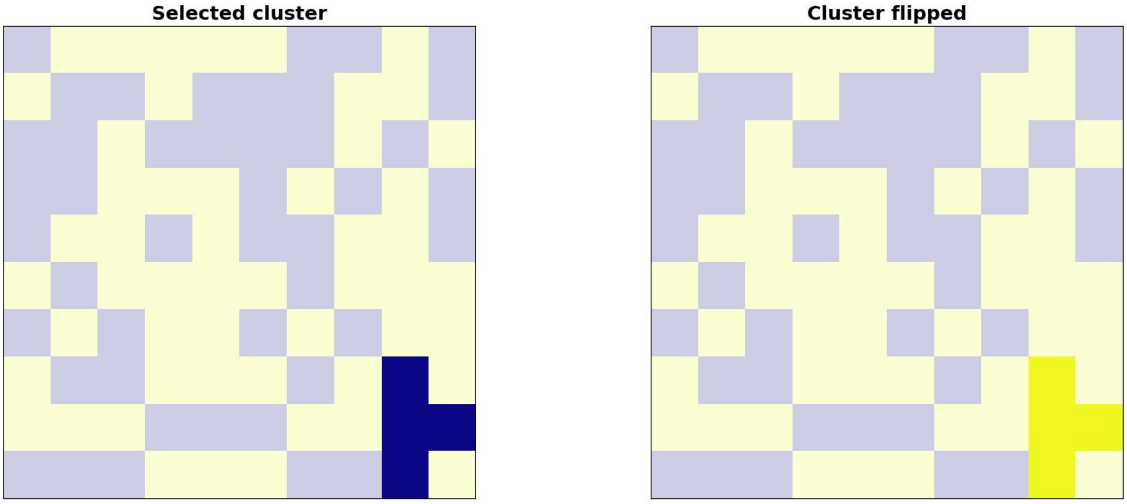

Figure 4.2: Demonstration of cluster flip in Wulff-Cluster algorithm

In Figure 4.2, a demonstration of a cluster flip in the Wolff algorithm is illustrated. In the next section, we discuss how these Monte Carlo methods were utilized to implement Swenden's approach for MCRG analysis.

4.3 Implementation and Results

A set of configurations for an Ising square lattice is simulated using the Wolff algorithm at the critical temperature, and the equilibration of energy and magnetization during the simulation using the Wolff algorithm is shown in Figure 4.3.

Lattice size	500
Beta	0.45
Equilibration steps	1000
Simulation steps	2000

Table 4.1: Simulation Parameters

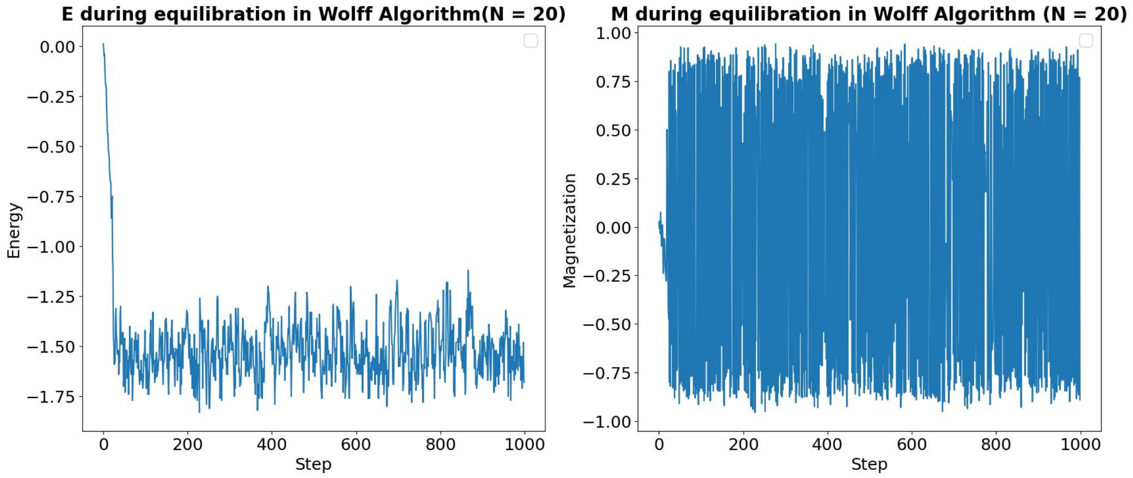

Figure 4.3: Equilibration Energy and Magnetisation in Wolff algorithm

The lattice size is chosen to be $N \times N = 500 \times 500$, and the simulation parameters are summarized in Table 4.1. The Kadanoff-block spin transformation is performed on the simulated configurations, where a block size of $b = 2$ is selected, and the majority rule is used to assign spins to each block. The renormalization transformations are carried out for $n = 3$ levels, resulting in coarse-grained lattices of sizes 250, 125, and 62. The coarse-grained lattices obtained from the renormalization process are illustrated in Figure 4.4.

Next, we consider five couplings in the Hamiltonian:

$$\mathscr{H}(s) = \sum_a K_a S_a = K_1 S_1 + K_2 S_2 + K_3 S_3 + K_4 S_4 \tag{4.17}$$

$$= K_1 S_1 + K_2 S_2 + K_{3x} S_{3x} + K_{3y} S_{3y} + K_4 S_4 \tag{4.18}$$

$$= K_1 \sum_i s_i + K_2 \sum_{<i,j>} s_i s_j + K_{3x} \sum_{i,j,k} s_i s_j s_k + K_{3y} \sum_{i,j,k} s_i s_j s_k + K_4 \sum_{i,j,k,l} s_i s_j s_k s_l \tag{4.19}$$

where we have considered 3 odd couplings with the following spin products:

- S_1 - the sum of single-spin interaction
- S_{3x} - the sum of the 3 - spin interaction along the horizontal direction in the lattice .
- S_{3y} - the sum of the 3 - spin interaction along the vertical direction in the lattice .

and 2 even couplings with the spin products:

- S_2 - the sum of nearest neighbour interactions
- S_4 - the sum of next nearest neighbour interactions

We calculate the spin products and spin correlations for each coupling. These quantities are utilized to solve a matrix equation, which provides the elements of the linear transformation matrix at different renormalization levels. Figure 4.5 displays the resulting matrix.

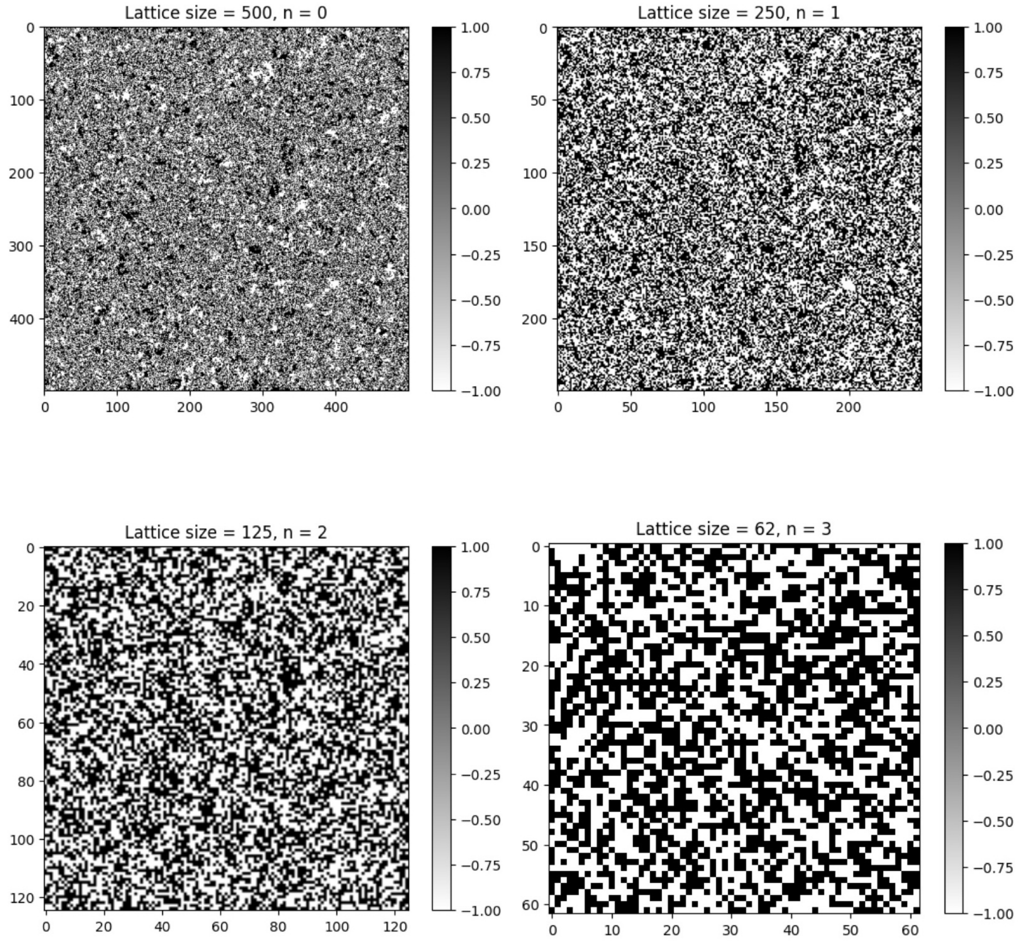

Figure 4.4: Coarse-graining of the simulated lattices

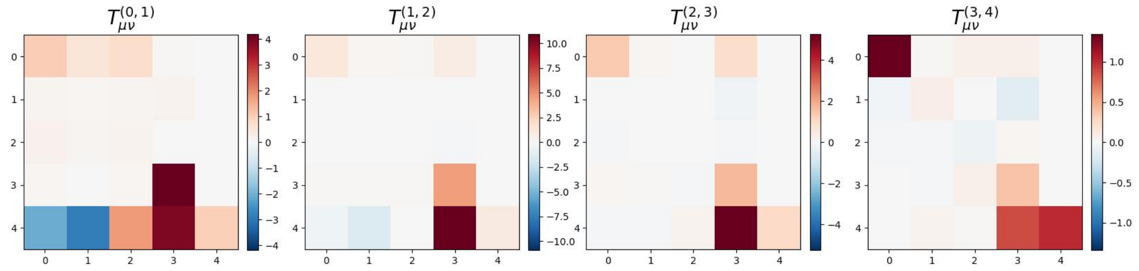

Figure 4.5: Transformation Matrix elements at different renormalisation levels from MCRG

Finally, the eigenvalues of the linear transformation matrix are computed and analyzed in relation to the critical exponent. The details of the code used are given in the Appendix.

The obtained value of the critical exponent does not converge to the expected theoretical value. However, it should be noted that with larger lattice sizes, including a greater number of couplings, and increased simulations, the accuracy of the critical exponent

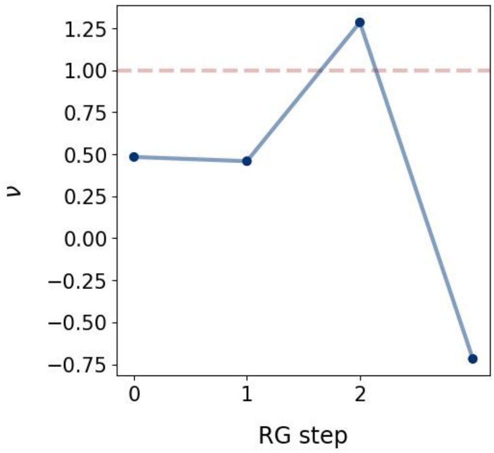

Figure 4.6: Critical Exponent - ν from MCRG

estimation can be significantly improved. Further investigations using these enhancements may lead to a better convergence towards the true value of the critical exponent.

Chapter 5

Conclusions

In this thesis, we have explored the real-space renormalization group (RG) techniques in the study of critical phenomena in statistical physics. In real-space renormalization-group analysis, the problem of an infinite number of interactions is addressed by employing approximations that retain only a finite number of interactions. These ”truncation” approximations are justified on the assumption that the renormalized coupling constants exhibit short-range behavior. While this assumption is supported by successful numerical calculations, it is important to note that there are equally plausible approximations that yield poor results. These truncation approximations introduce uncertainty, making it difficult to assess the reliability of the calculations without comparisons to other methods.

Throughout this thesis, we have explored various RG methods, including decimation, Migdal-Kadanoff bond moving approximation, Niemeijer-van Leeuwen cumulant approximation, and Monte Carlo renormalization group methods, to gain insights into the critical behavior of Ising and Potts models. However, we have encountered limitations with the decimation procedures in higher dimensions due to the proliferation of interactions. On the other hand, the Migdal-Kadanoff approximation offers the advantage of applicability to a broader range of spin systems. Notably, the cumulant approximation has provided improved estimates compared to the Migdal-Kadanoff approximation, particularly in the case of the 2D Ising model. These real-space RG methods boast conceptual clarity and calculational simplicity, and they have demonstrated their capability to achieve good agreement with known results.

In principle, real-space RG approximations can be enhanced by including additional coupling constants in the calculations. However, the computational complexity grows rapidly with the inclusion of more couplings, posing practical challenges. Nevertheless, Monte Carlo simulations offer an alternative avenue for systematic improvement of all approximations without incurring significant additional effort. For the most part, the remaining errors in Monte Carlo simulations are statistical in nature and well understood, allowing for accurate estimation of their magnitude through analysis of the generated data. Moreover, efficient sampling algorithms within Monte Carlo methods enable us to tackle the issue of finite-size effects, a common obstacle when calculating the properties of infinite systems near the critical temperature.

While this thesis has provided significant insights into the application of real-space renormalization group methods, there are several avenues for future research and exploration in this field. Exploring the critical behavior of complex lattice models, such as the XY model and Heisenberg model, would deepen our understanding of exotic phases and phase transitions. Optimization of existing Monte Carlo algorithms through parallel computing techniques, GPU acceleration, and advanced sampling methods holds the potential to further improve their efficiency and accuracy. Additionally, incorporating quantum effects into real-space RG methods, particularly in quantum spin models and interacting quantum field theories, would open new frontiers for investigating quantum phase transitions and emergent phenomena.

In conclusion, the study of critical phenomena using real-space renormalization group methods is a rich and active research area. Continued exploration and development of these methods, along with their applications to various lattice models and experimental systems, hold great potential for advancing our understanding of critical behavior and emergent phenomena in complex systems.

Appendix

Monte Carlo Renormalization Group calculations for 2D Ising Model

```python
def initialize_ising_lattice(size):
  '''
    Returns a random Ising spin configuration in a square lattice.

    Parameters:
    -----------
    size : int
        The size of the square lattice

    Returns:
    --------
    lattice : numpy.ndarray
        An array representing the Ising spin configuration.
        It has shape (size, size) and contains randomly assigned
           spins of -1 or 1.
  '''

  return np.random.choice([-1, 1], size=(size, size))

def visualize_ising_lattice(lattice, n):
    '''
    Plots the Ising spin configuration as a square lattice.

    Parameters:
    -----------
    lattice : numpy.ndarray
        Ising spin configuration as a square lattice.

    n : int
        Renormalization level.

    Returns:
    --------
    None
        This function doesn't return any value. It plots the
           lattice and displays it.

    '''

    size = lattice.shape[0]
    plt.imshow(lattice , cmap='binary')
    plt.colorbar()
    plt.title(f"Lattice size = {size}, n = {n}")
    plt.show()

def energy_ising(state, J):
    '''
    Calculate the energy of an Ising spin configuration in the nearest neighbor model.

    Parameters:
    -----------
    state : numpy.ndarray
        Ising spin configuration as a square lattice.

    J : float
        Interaction strength.

    Returns:
    --------
    energy : float
        The energy of the Ising spin configuration.

    '''
    size = state.shape[0]
    energy = 0.0

    for i in range(size):
        for j in range(size):
            spin = state[i, j]
            neighbors = [
                state[(i+1)%size, j],  # right neighbor
                state[i, (j+1)%size],  # bottom neighbor
                state[(i-1)%size, j],  # left neighbor
                state[i, (j-1)%size]   # top neighbor
            ]
            energy += -J* spin * np.sum(neighbors)

    return energy/2

def single_flip_ising(state, beta, J):
    '''
    Perform a single spin flip in the Ising spin configuration using the Metropolis algorithm.

    Parameters:
        state (ndarray): Ising spin configuration as a square lattice.
        beta (float): Inverse temperature.

    Returns:
        new_state (ndarray): Updated Ising spin configuration after a single spin flip.
    '''
    size = state.shape[0]
    i = np.random.randint(size) # choose a spin site with uniform probability
    j = np.random.randint(size)

    old_spin = state[i, j]
    neighbors = [
        state[(i+1)%size, j],  # right neighbor
        state[i, (j+1)%size],  # bottom neighbor
        state[(i-1)%size, j],  # left neighbor
        state[i, (j-1)%size]   # top neighbor
    ]

    energy_diff = 2 *J * old_spin * np.sum(neighbors)


    # Metropolis
    if energy_diff <= 0 or np.random.rand() < np.exp(-beta * energy_diff):
        new_state = state.copy()
        new_state[i, j] = -old_spin  # perform spin flip
        return new_state
    return state

def add(state, beta, starting, cluster):
    '''
    Recursively add spins to the cluster based on the Wolff algorithm.

    Parameters:
    -----------
    state : numpy.ndarray
        Ising spin configuration as a square lattice.

    beta : float
        Inverse temperature.

    starting : list
        Coordinates of the starting spin.

    cluster : list
        List of spins in the cluster.

    Returns:
    --------
    cluster : list
        Updated list of spins in the cluster.

    '''

    # get lattice size
    L = state.shape[0]
    N = L**2

    # prob to add a spin to the cluster
    p_add = 1. - np.exp(-2*beta)

    # get indexes of starting spin
    i, j = starting[0], starting[1]

    # get neighbors
    nbr = np.array([ [(i - 1)%L, j], [(i + 1)%L, j], [i, (j - 1)%L], [i, (j + 1)%L] ])

    starting_spin = state[i, j]

    # list to store accepted spins
    new_elements = []

    for n in nbr:
        # add spin
        if state[n[0], n[1]] == starting_spin and [n[0], n[1]] not in cluster \
        and np.random.uniform(0., 1.) < p_add:

            new_elements.append([n[0], n[1]])

    # add elements to the cluster
    cluster += new_elements

    return cluster

def wolff(state, beta):
    '''
    Perform a single update step using the Wolff algorithm on an Ising spin configuration.

    Parameters:
    -----------
    state : numpy.ndarray
        Ising spin configuration as a square lattice.

    beta : float
        Inverse temperature.

    Returns:
    --------
    state : numpy.ndarray
        Updated Ising spin configuration after applying the Wolff algorithm.

    '''
    # get lattice size
    L = state.shape[0]
    N = L**2

    # get random spin
    i, j = np.random.randint(L), np.random.randint(L)

    # the random spin is the first element of the cluster
    cluster = [[i,j]]

    # build the cluster
    for elem in cluster:
        cluster = add(state, beta, elem, cluster)
```

```python

    # flip the cluster
    for elem in cluster:
        state[elem[0], elem[1]] *= -1

    return state

def Ising(N, beta, eq_steps, sim_steps, J):
    '''
    Run a Monte Carlo (MC) simulation of a 2D Ising model.

    Parameters:
    -----------
    N : int
        Size of the lattice (N x N).

    beta : float
        Inverse temperature ( ).

    eq_steps : int
        Number of equilibration steps.

    sim_steps : int
        Number of simulation steps.

    J : float
        Interaction strength.

    Returns:
    --------
    sim_results : dict
        Dictionary with simulation results including the current state, all states,
        energy (E), and magnetization (M) during equilibration and simulation.

    '''

    print('Start equilibration...')

    curr_state = initialize_ising_lattice(N)

    E_eq = np.zeros(eq_steps)
    M_eq = np.zeros(eq_steps)

    beta_eq = np.zeros(eq_steps) + beta

    for i in tqdm(range(eq_steps)):
        E_eq[i] = energy_ising(curr_state, J)
        M_eq[i] = np.sum(curr_state)
        curr_state = wolff(curr_state, beta_eq[i])

    print('End equilibration...\n')

    print('Start simulation...')

    all_states = []

    E = np.zeros(sim_steps)
    M = np.zeros(sim_steps)

    for i in tqdm(range(sim_steps)):

            # Store all the simulated states
            all_states.append(curr_state.copy())

            # Compute energy and magnetisation
            E[i] = energy_ising(curr_state, J)
            M[i] = np.sum(curr_state)

            curr_state = wolff(curr_state, beta_eq[-1])

    print('End simulation...')

   # Create dictionary of simulation results
    sim_results = {
        'all_states': all_states,
        'curr_state': curr_state,
        'E_eq': E_eq,
        'M_eq': M_eq,
        'E': E,
        'M': M,
    }

    return sim_results

N = 20
eq_steps = 1000
sim_steps = 100
J = 1
betas = np.array([0.2, 1, 100]) # different values of beta
col_map = np.array(["navy", "red", "green", "orange"])

#create plots
fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols = 2, figsize = (20, 8))
ax1.set_title(r"E during equilibration (N = {})".format(N), fontweight = "bold", fontsize = 20)
ax2.set_title(r"M during equilibration (N = {})".format(N), fontweight = "bold", fontsize = 20)
ax1.set_xlabel("Step", fontsize = 18)
ax2.set_xlabel("Step", fontsize = 18)
ax1.set_ylabel("Energy", fontsize = 18)
ax2.set_ylabel("Magnetization", fontsize = 18)
ax1.tick_params(axis="x",labelsize=18)
ax2.tick_params(axis="x",labelsize=18)
ax1.tick_params(axis="y",labelsize=18)
ax2.tick_params(axis="y",labelsize=18)

#loop over betas and compute values
for i in range(len(betas)):
    E_eq, M_eq, E, M, E_mean, M_mean, E_var, M_var = Ising(N, betas[i], eq_steps, sim_steps, J)

    ax1.plot(E_eq/N**2, label = r"$\beta$ = {}".format(betas[i]), c = col_map[i])
    ax2.plot(M_eq/N**2, label = r"$\beta$ = {}".format(betas[i]), c = col_map[i])

ax1.legend(fontsize = 18)
ax2.legend(fontsize = 18)
plt.show()

# Define system at critical temperature
N = 500
eq_steps = 1000
sim_steps = 2000

J = 1
beta = 0.45

run_simulations = False

if run_simulations:
      # Run Ising simulations for initial (original - og) lattices
      sim_results_og = Ising(N, beta, eq_steps, sim_steps, J)

      all_confs_og = np.array(sim_results_og['all_states'])
      all_confs_og.shape

      np.save('all_spin_config_og.npy', all_confs_og)

else:
      # Load the file with all spin configurations
      all_confs_og = np.load('/content/all_spin_config_og.npy')
      all_confs_og.shape

def block_transform(lattice, block_size):
    '''
    Transform the lattice by grouping spins into blocks and assigning a single spin value to each block acording to the majority rule.

    Parameters:
    -----------
    lattice : 2darray
        Ising spin configuration as a square lattice.

    block_size : int
        Size of the blocks to group the spins.

    Returns:
    --------
    new_lattice : 2darray
        Transformed lattice with spins grouped into blocks.

    '''
```

```python
    lattice_size = lattice.shape[0]
    new_lattice_size = lattice_size // block_size
    new_lattice = np.zeros((new_lattice_size, new_lattice_size))

    for i in range(new_lattice_size):
        for j in range(new_lattice_size):
            block = lattice[i*block_size:(i+1)*block_size, j*block_size:(j+1)*block_size]
            sum_of_spins = np.sum(block)
            if sum_of_spins > 0:
                new_lattice[i, j] = 1
            elif sum_of_spins < 0:
                new_lattice[i, j] = -1
            else:
                # Randomly choose +1 or -1 for the tie-breaker
                new_lattice[i, j] = random.choice([-1, 1])

    return new_lattice

levels = 4
b = 2 # block size
all_confs_cg = [] # list of all configurations at each coarse graining (cg) level
all_confs_cg = [all_confs_og.copy()]

for i in range(levels):
    print("Coarse Graining Step:", i)
    current_confs = np.zeros((all_confs_og.shape[0], all_confs_og.shape[1]//b**(i+1), all_confs_og.shape[2]//b**(i+1)), dtype=np.int8)
    print('current_confs shape', current_confs.shape)
    for idx, conf in enumerate(all_confs_cg[-1]):
        current_confs[idx] = block_transform(conf, b)
    all_confs_cg.append(current_confs.astype(np.int8))


def calculate_S2(lattice):
    '''
    Calculates the sum of nearest-neighbour interaction for an Ising spin configuration.

    Parameters:
    -----------
    lattice : 2darray
        Ising spin configuration as a square lattice.

    Returns:
    --------
    a : float
        Sum of nearest-neighbour interaction for the given lattice.

    '''

    size = lattice.shape[0]
    a = 0

    for i in range(size):
        for j in range(size):
            # Get the value of the spin at the current lattice site
            spin = lattice[i, j]

            # Compute the contribution from the nearest neighbor pairs
            # Assuming periodic boundary conditions
            nn_sum = spin * (lattice[(i+1)%size, j] + lattice[i, (j+1)%size] +
                             lattice[(i-1)%size, j] + lattice[i, (j-1)%size])

            # Add the contribution to the total summation
            a += nn_sum

    return a/2

def calculate_S3_x(lattice):
        '''
        Calculates the sum of the 3-spin interaction along the horizontal direction in the lattice.
        This is an odd coupling.

        Parameters:
        -----------
        lattice : 2darray
            Ising configuration as a square lattice.

        Returns:
        --------
        a : float
            Sum of the 3-spin interaction along the horizontal direction.

        '''
        a = np.pad(lattice, 1, 'wrap')[:-2,:]
        a = a[:,:-2]*a[:,1:-1]*a[:, 2:]
        return np.sum(a)

def calculate_S3_y(lattice):
        '''
        Calculate the sum of the 3-spin interaction along the vertical direction in the lattice.
        This is an odd coupling.

        Parameters:
        -----------
        lattice : 2darray
            Ising configuration as a square lattice.

        Returns:
        --------
        a : float
            Sum of the 3-spin interaction along the vertical direction.

        '''

        a = np.pad(lattice, 1, 'wrap')[:,:-2]
        a = a[:-2,:]*a[1:-1,:]*a[2:,:]
        return np.sum(a)


def calculate_spin_products(states):
        '''
        Calculates the following 5 spin products given a set of Ising configurations.

           (0) -> 1 spin interaction (odd)
           (1) -> 3 spin horizontal interaction (odd)
           (2) -> 3 spin vertical interaction (odd)
           (3) -> nearest neighbors interaction (even)
           (4) -> next nearest neighbors interaction (even)

        Parameters:
        -----------
        states : ndarray
            Set of Ising configurations represented as a 3D array (nsim x N x N), where nsim is the number of configurations.

        Returns:
        --------
        spin_products : ndarray
            Array of spin products of shape (nsim x 5).

        '''

        kernel_nnn = np.array([[1,0,1],
                               [0,0,0],
                               [1,0,1]])

        nsim = states.shape[0]
        spin_products = np.zeros((nsim,5),dtype = np.int64)

        for i in range(1, nsim):
          spin_products[i][0] = np.sum(states[i])
          spin_products[i][1] = calculate_S3_x(states[i])
          spin_products[i][2] = calculate_S3_y(states[i])
          spin_products[i][3] = calculate_S2(states[i])
          spin_products[i][4] = np.sum(conf*convolve2d(conf, kernel_nnn, mode = 'same', boundary = 'wrap'))/2

        return spin_products

spin_products_cg = []
for idx, data in enumerate(all_confs_cg):
    print(idx)
    spin_products_cg.append(calculate_spin_products(data))
    print(len(spin_products_cg), len(spin_products_cg[0]))

```

```python
def calculate_cov(x,y):
        '''
        Returns the covariance between x and y.

        Parameters:
        -----------
        x, y : array-like
            Arrays representing the variables for which covariance needs to be calculated.

        Returns:
        --------
        cov : float
            Covariance between x and y.
        '''

        return np.sum((x - x.mean())*(y-y.mean()))/(x.shape[0])

A_mat_cg = []
B_mat_cg = []

for i in range(len(spin_products_cg) - 1):
    A = np.cov(spin_products_cg[i+1].T, bias = False)
    A_mat_cg.append(A)

    B = np.zeros(A.shape)

    for mu in range(A.shape[0]):
        for nu in range(A.shape[1]):
            B[mu, nu] = calculate_cov(spin_products_cg[i+1][:, mu], spin_products_cg[i][:, nu])

    B_mat_cg.append(B)

T_mat_cg = []

for A, B in zip(A_mat_cg, B_mat_cg):
    T_mat_cg.append(np.linalg.inv(A) @ B)

T_mat_cg = np.array(T_mat_cg)
T_mat_cg.shape

def calculate_even_odd_eigvals(T_mat_cg):
        """
        Finds the eigenvalues of the linearized RG transformation for the even and odd couplings.

        Parameters
        ----------
        T_mat_cg : Nx5x5 array
            Linearized RG transformation.

        Returns
        -------
        eig_even_cg : Nx2 array
            Eigenvalues of the even couplings.
        eig_odd_cg : Nx3 array
            Eigenvalues of the odd couplings.
        """
        eig_even_cg = []
        eig_odd_cg = []

        for i, T in enumerate(T_mat_cg):
            eigvals, eigvecs = np.linalg.eig(T)

            ye = []
            for idx in range(4):
                ye.append(np.linalg.eigvals(T[3:4+idx, 3:4+idx])[0])

            eig_even_cg.append(ye)

            yo = []
            for idx in range(3):
                yo.append(np.linalg.eigvals(T[:1+idx, :1+idx])[0])
            eig_odd_cg.append(yo)

        eig_even_cg = np.array(eig_even_cg)
        eig_odd_cg = np.array(eig_odd_cg)

        return eig_even_cg, eig_odd_cg

eig_even_cg, eig_odd_cg = calculate_even_odd_eigvals(T_mat_cg)

for i in range(levels):
  print('n=', i)
  visualize_ising_lattice(all_confs_cg[i][-1], i)

fig, ax = plt.subplots(1, T_mat_cg.shape[0], figsize = (20,5))

vmin = np.min(T_mat_cg)
vmax = np.max(T_mat_cg)

for i in range(T_mat_cg.shape[0]):
    #center the cmap in zero

    im = ax[i].imshow(T_mat_cg[i], cmap = 'RdBu_r', norm = matplotlib.colors.CenteredNorm())
    ax[i].set_title(r'$T_{\mu \nu}^{(' + str(i) + ', ' + str(i+1) + ')}$', pad = 10, fontsize = 20)
    plt.colorbar(im, ax = ax[i], fraction = 0.046, pad = 0.04)

plt.show()

eig_even_cg, eig_odd_cg = calculate_even_odd_eigvals(T_mat_cg)

fig, ax = plt.subplots(figsize = (5,5))
c = 0
for ye_list in eig_even_cg:
    ax.scatter([c]*len(ye_list), [np.log(b)/np.log(np.real(val)) for val in ye_list],
                c = np.arange(len(ye_list)), cmap = plt.get_cmap('Blues'))
    c += 1
ax.plot(np.arange(len(eig_even_cg)), [np.log(b)/np.log(np.real(val)[-1]) for val in eig_even_cg], alpha = 0.5,
         lw = 3, color = plt.get_cmap('Blues')(0.95))
#ax.set_ylim(0.9, 1.3)
ax.set_xlabel('RG step', fontsize = 17, labelpad = 15)
ax.set_xticks([0,1,2])
ax.set_ylabel(r'$\nu$', fontsize = 17, labelpad = 15)
ax.tick_params(labelsize=15)
plt.axhline(1, ls = '--', lw = 3, color = 'firebrick', zorder = -1, alpha = 0.3)

fig, ax = plt.subplots(figsize = (5,5))
c = 0
for ye_list in eig_odd_cg:
    ax.scatter([c]*len(ye_list), [np.log(np.real(val))/np.log(b)/(2-np.log(np.real(val))/np.log(b)) for val in ye_list],
                c = np.arange(len(ye_list)), cmap = plt.get_cmap('Blues'))
    c += 1
ax.plot(np.arange(len(eig_odd_cg)), [np.log(np.real(val[-1]))/np.log(b)/(2-np.log(np.real(val[-1]))/np.log(b)) for val in eig_odd_cg], alpha = 0.5,
         lw = 3, color = plt.get_cmap('Blues')(0.95))
#ax.set_ylim(0.9, 1.3)
ax.set_xlabel('RG step', fontsize = 17, labelpad = 15)
ax.set_xticks([0,1,2])
ax.set_ylabel(r'$\delta$', fontsize = 17, labelpad = 15)
ax.tick_params(labelsize=15)
plt.axhline(15, ls = '--', lw = 3, color = 'firebrick', zorder = -1, alpha = 0.3)



fig, ax = plt.subplots(figsize = (5,5))
c = 0
for ye_list in eig_odd_cg:
    ax.scatter([c]*len(ye_list), [4 - 2*np.log(np.real(val))/np.log(b) for val in ye_list],
                c = np.arange(len(ye_list)), cmap = plt.get_cmap('Blues'))
    c += 1
ax.plot(np.arange(len(eig_odd_cg)), [4 - 2*np.log(np.real(val[-1]))/np.log(b) for val in eig_odd_cg], alpha = 0.5,
         lw = 3, color = plt.get_cmap('Blues')(0.95))
#ax.set_ylim(0.9, 1.3)
ax.set_xlabel('RG step', fontsize = 17, labelpad = 15)
ax.set_xticks([0,1,2])
ax.set_ylabel(r'$\eta$', fontsize = 17, labelpad = 15)
ax.tick_params(labelsize=15)
plt.axhline(0.25, ls = '--', lw = 3, color = 'firebrick', zorder = -1, alpha = 0.3)
```

Bibliography

[1] D. Andelman and A. N. Berker. q-state potts models in d dimensions: Migdal-kadanoff approximation. *Journal of Physics A: Mathematical and General*, 14(4):L91, apr 1981. doi: 10.1088/0305-4470/14/4/005. URL https://dx.doi.org/10.1088/0305-4470/14/4/005.

[2] T. W. Burkhardt and J. M. J. Leeuwen, editors. *Real-Space Renormalization*. Topics in Current Physics. Springer Berlin, Heidelberg, 1982. ISBN 978-3-642-81827-1. doi: 10.1007/978-3-642-81825-7. URL https://doi.org/10.1007/978-3-642-81825-7. eBook ISBN: 978-3-642-81825-7.

[3] N. Goldenfeld. *Lectures On Phase Transitions And The Renormalization Group*. CRC Press, Boca Raton, 1st edition edition, 1992. doi: 10.1201/9780429493492. URL https://doi.org/10.1201/9780429493492. eBook ISBN: 9780429493492.

[4] B. Hu. Introduction to real-space renormalization-group methods in critical and chaotic phenomena. *Physics Reports*, 91(5):233–295, 1982. ISSN 0370-1573. doi: https://doi.org/10.1016/0370-1573(82)90057-6. URL https://www.sciencedirect.com/science/article/pii/0370157382900576.

[5] L. P. Kadanoff. Scaling laws for ising models near T_c. *Physics Physique Fizika*, 2: 263–272, Jun 1966. doi: 10.1103/PhysicsPhysiqueFizika.2.263. URL https://link.aps.org/doi/10.1103/PhysicsPhysiqueFizika.2.263.

[6] L. P. Kadanoff. Notes on migdal's recursion formulas. *Annals of Physics*, 100(1):359–394, 1976. ISSN 0003-4916. doi: https://doi.org/10.1016/0003-4916(76)90066-X. URL https://www.sciencedirect.com/science/article/pii/000349167690066X.

[7] M. Kardar. *Lattice systems*, page 98–122. Cambridge University Press, 2007. doi: 10.1017/CBO9780511815881.007.

[8] E. Luijten. *Introduction to Cluster Monte Carlo Algorithms*, pages 13–38. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-35273-0. doi: 10.1007/3-540-35273-2_1. URL https://doi.org/10.1007/3-540-35273-2_1.

[9] S.-k. Ma. Renormalization group by monte carlo methods. *Phys. Rev. Lett.*, 37: 461–464, Aug 1976. doi: 10.1103/PhysRevLett.37.461. URL https://link.aps.org/doi/10.1103/PhysRevLett.37.461.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. , 21(6):1087–1092, June 1953. doi: 10.1063/1.1699114.

[11] A. A. Migdal. Recursion equations in gauge field theories. *Soviet Journal of Experimental and Theoretical Physics*, 42:413, Sept. 1975.

[12] T. Niemeijer and J. M. J. Van Leeuwen. Renormalization Theory for Ising Like Spin Systems. In *12th School of Modern Physics on Phase Transitions and Critical Phenomena*, 1976.

[13] L. Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65:117–149, Feb 1944. doi: 10.1103/PhysRev.65.117. URL `https://link.aps.org/doi/10.1103/PhysRev.65.117`.

[14] L. Peliti and M. Epstein. *Phase Transitions*, pages 125–172. Princeton University Press, 2003. ISBN 9780691145297. URL `http://www.jstor.org/stable/j.ctvcm4h83.9`.

[15] R. H. Swendsen. Monte carlo renormalization group. *Phys. Rev. Lett.*, 42:859–861, Apr 1979. doi: 10.1103/PhysRevLett.42.859. URL `https://link.aps.org/doi/10.1103/PhysRevLett.42.859`.

[16] K. G. Wilson and J. Kogut. The renormalization group and the expansion. *Physics Reports*, 12(2):75–199, 1974. ISSN 0370-1573. doi: https://doi.org/10.1016/0370-1573(74)90023-4. URL `https://www.sciencedirect.com/science/article/pii/0370157374900234`.

[17] U. Wolff. Collective monte carlo updating for spin systems. *Phys. Rev. Lett.*, 62:361–364, Jan 1989. doi: 10.1103/PhysRevLett.62.361. URL `https://link.aps.org/doi/10.1103/PhysRevLett.62.361`.