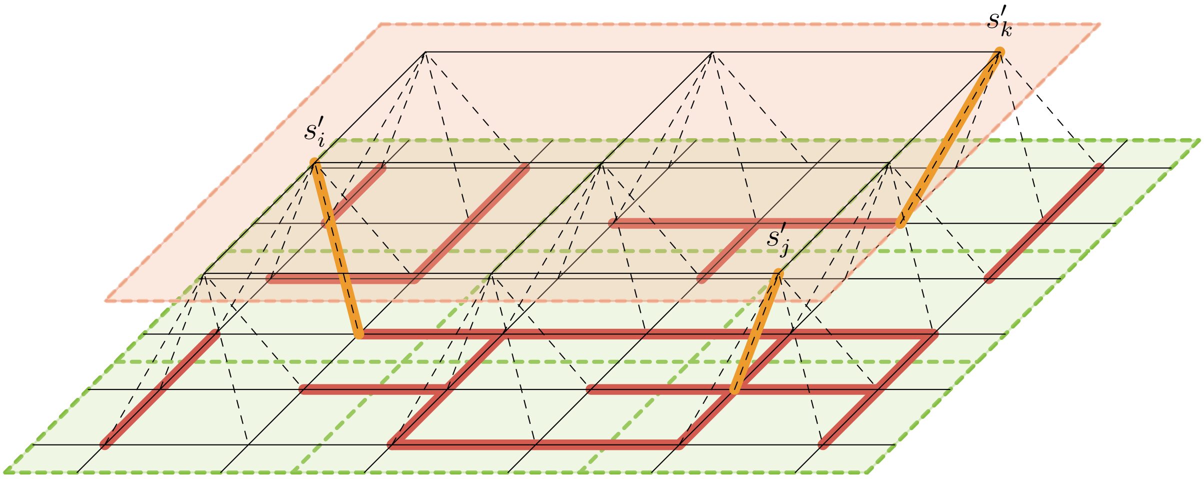


Introduction to Cluster-Inspired RG on the Ising Model

Master thesis, Faculty of Science, University of Bern

handed in by

Andrin Sprecher

2024

Supervisor

Prof. U.J. Wiese

Contents

Acknoledgements

I thank Prof. Uwe-Jens Wiese for his dedicated, supportive and cooperative supervision of the project. I thank Joel Wittwer and Samuel Jullierat for doing the project together. I thank Lukas Born for occasionally providing an office space and his company. I thank Valentin Wyss for his support with algorithmic and programming issues. I thank Binia Marti for her flexibility regarding office seating. I thank Adriana Pisani, Elisabeth Sprecher, Erwin Gisler, and all my friends for their support.

Introduction

The foundation of this research project was Prof. Uwe-Jens Wiese's idea to develop a new renormalization group (RG) transformation on the Ising model that could be solved numerically by the means of the Monte Carlo cluster algorithm, therefore we call it cluster-inspired RG. The cluster algorithm has the power to maintain a high efficiency at the critical point, where other algorithms fail. The critical point is of special interest when doing RG. The flow of an RG transformation starting at the critical point will remain on the critical surface and may approach a fixed point in theory space. Fixed points are interesting for many reasons. At lattice theory living at a fixed point loses all its lattice artifacts and is scale invariant. The critical surface related to the fixed point forms a universality class of theories which all show universal behavior. This holds for seemingly unrelated theories like the Ising model and the thermodynamical model of water. The ultimate goal of the development of cluster-inspired RG will be to approach fixed points and generate data to determine their Hamilton functions numerically. The hope is to parametrize the fixed point of the 2D Ising model with higher accuracy than before and perhaps to parametrize the fixed point of the 3D Ising model. Even though this project has not yet progressed to that point, cluster-inspired RG may have the potential to do so. It is not known to have been done before and the drastic increase in computational power within the last decades can be an augmenting factor for its success.

Chapter 1 provides background information on the Ising model and general Ising-type lattice models, shows ways to solve the 1D Ising model, and gives an introduction on phase transitions and critical phenomena. Chapter 2 introduces the Monte Carlo cluster algorithm for the Ising model, presents results obtained on the 2D square lattice, and discusses error analysis. Chapter 3 explains the concept of RG applied to the Ising model and introduces the concept of cluster-inspired RG. Chapter 4 discusses our analytical work to it on the 1D Ising model, which is the first major part of research we did on the project. And Chapter 5 discusses the application and verification of cluster-inspired RG on the 2D Ising model, which is the second major part of the project.

1 The Ising Model

1.1 Ising-Type Models

Ising-type models are *lattice* models. A lattice is a regular structure of points in space, called *lattice sites*, with discrete translational, rotational and reflectional symmetries. Transformations are symmetries of a lattice if they map all lattice sites onto each other in such a way that their relative positions remain unchanged.

A section of a 2D square lattice with lattice spacing a is displayed in Figure 1.1. An infinite lattice of that kind has a discrete translational symmetry by multiples of a and a 90° rotational and reflectional symmetry. These symmetries extend to a finite lattice if the shape of the lattice also obeys the rotational and reflectional symmetries and periodic boundary conditions are chosen. The shape of a finite 2D square lattice that obeys the symmetries, is also a square. *Periodic boundary conditions* imply that the lattice repeats itself at its boundary in each translationally symmetric direction, allowing a finite lattice to maintain translational symmetry; whereas *open boundary conditions* would refer to a lattice that ends at its boundary.

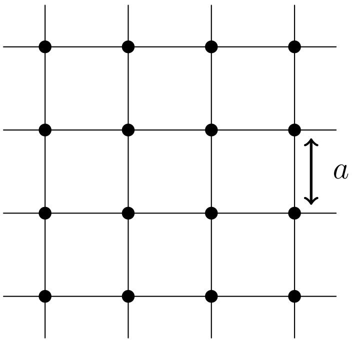


Figure 1.1: Section of a 2D square lattice with lattice spacing a.

Ising-type models are based on identical *spin variables*, or *spins*, that live on a lattice. Each spin has its own lattice site and can take two discrete values of *spin up* and *spin down*, which are sometimes illustrated as $\uparrow$ and $\downarrow$ or as two different colors. We choose to set spin up to $+1$ and spin down to -1. If two spins have the same value, they are called *parallel* and otherwise *anti-parallel.* A section of a 2D square lattice Ising-type system is illustrated in Figure 1.2.

1.1.1 The Hamilton Function, the Boltzmann Weight and the Probability

Ising-type models belong to the field of *statistical physics*, a field that describes physical systems in a statistical way instead of analysing their dynamics. The state of an entire Ising-type system is called *configuration* and denoted $[s]$. It consists of all spins' states together, thus

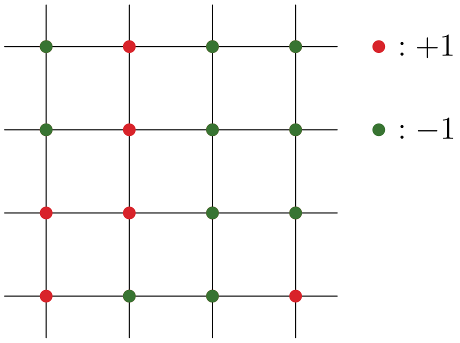


Figure 1.2: Illustration of a 2D square lattice section of an Ising-type system. The red and green dots stand for spin up and spin down.

$[s] = [s_1, s_2, \dots, s_n]$, where s_i is a single spin, with $i \in \{1, \dots, n\}$ being the spatial index related to the spin's lattice site. Instead of describing the system's dynamics, one assigns a probability to each configuration, predicting how frequently the system will be found in that configuration when measured.

The probability of a configuration is determined by the *Hamilton function* $\mathcal{H}[s]$, a real quantity describing the energy of the system, as well as the temperature, usually converted to its inverse $\beta = \frac{1}{k_B T}$, called the *inverse temperature*, where k_B is Boltzmann's constant. A configuration's probability follows the *Boltzmann distribution*, meaning that it is proportional to its *Boltzmann weight*

$$exp(-\beta\mathcal{H}[s]), \tag{1.1}$$

that must be normalized by the partition function

$$Z = \sum_{[s]} exp(-\beta\mathcal{H}[s]), \tag{1.2}$$

to become a a probability. The probability of a configuration then is

$$P[s] = \frac{1}{Z} exp(-\beta H[s]) = \frac{exp(-\beta H[s])}{\sum_{[\tilde{s}]} exp(-\beta\mathcal{H}[\tilde{s}])}, \tag{1.3}$$

such that the sum of all probabilities $\sum_{[s]} P[s]$ correctly adds up to 1.

The Boltzmann distribution implies that at finite temperatures, low energy configurations are more probable than high energy configurations. At zero temperature, or $\beta \to \infty$, the configurations with the lowest energy are the only ones that have a non-zero probability. With increasing temperature, or decreasing β, the probability of high energy configurations increases more and more, and in the high temperature limit, where $\beta \to 0$, the Boltzmann distribution converges towards the uniform distribution, where all configurations have equally probability.

The term *Boltzmann weight* is often abbreviated as *weight* in this thesis.

1.1.2 Physical Observables and Their Expectation Values

One can measure physical observables on Ising-type models. They are functions of the configuration that are typically of *macroscopic* nature, relating to the entire system and respecting the lattice symmetries. We denote them $\mathcal{O}[s]$. In the best case, one can measure an observable's expectation value, that is

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \sum_{[s]} \mathcal{O}[s] exp(-\beta \mathcal{H}[s]). \tag{1.4}$$

However, in experiment one will instead measure the empirical mean value over a sample, which is

$$\bar{\mathcal{O}} = \frac{1}{Z} \sum_{[\tilde{s}]} \mathcal{O}[\tilde{s}] exp(-\beta \mathcal{H}[\tilde{s}]), \tag{1.5}$$

where $[\tilde{s}]$ symbolize the sample configurations. One is also interested in the variance

$$(\Delta \mathcal{O})^2 = \langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2, \tag{1.6}$$

and the standard deviation $\Delta \mathcal{O}$.

Important observables of Ising-type models are the energy, the magnetization, the susceptibility, the correlation function and the correlation length. The energy is represented by the Hamilton function $\mathcal{H}[s]$, the quantity which defines the interactions of the system through the energies they induce. It will be discussed in more detail in the next section.

The *magnetization* $\mathcal{M}[s]$ is defined as the sum over all spin values

$$\mathcal{M}[s] = \sum_i s_i. \tag{1.7}$$

It is responsible for the induced magnetic field of a ferromagnet. It is an *extensive* quantity, as it is proportional to the system size. The related *intensive* quantity, which is independent of the system size, is the average spin value $\bar{s}$, through which the magnetization can be written as

$$\mathcal{M}[s] = V\bar{s}, \tag{1.8}$$

where V is the volume or number of spins in the system. The translational symmetry of the lattice implies that expectation values of individual spin values must all be equal, such that $\langle s_i \rangle = \langle s_j \rangle$ for any two lattice sites i, j. It follows that these must also be equal to the expectation value of the average spin value. Therefore, the expectation value of the magnetization can be written as

$$\langle \mathcal{M} \rangle = \sum_i \langle s_i \rangle = V \langle s_i \rangle = V \langle \bar{s} \rangle. \tag{1.9}$$

The *susceptibility* measures how strongly local fluctuations in the magnetization are carried through the system and is related to the interactions between more distant spins in the system.

It is defined as

$$\chi = \frac{1}{V}(\Delta\mathcal{M})^2 = \frac{1}{V}(\langle\mathcal{M}^2\rangle - \langle\mathcal{M}\rangle^2). \tag{1.10}$$

The susceptibility is also an extensive quantity, as $\chi = \frac{1}{V}(\Delta\mathcal{M})^2 = V(\Delta s_i)^2$, and using again the translational symmetry of the lattice, it can be rewritten as

$$\chi = \frac{1}{V}\sum_i\sum_j(\langle s_i s_j\rangle - \langle s_i\rangle\langle s_j\rangle) = \sum_j\langle s_i s_j\rangle - V\langle\bar{s}\rangle^2. \tag{1.11}$$

$\langle s_i s_j\rangle$ is known as the *(two-spin) correlation function*. It describes the correlation between the two spins s_i and s_j. Positive values imply that the spins s_i and s_j tend to align, negative values that they tend to anti-align and a zero value implies that they are uncorrelated. One defines the connected correlation function $\langle s_i s_j\rangle_c = \langle s_i s_j\rangle - \langle\bar{s}\rangle^2$, which subtracts the influence of overall magnetization from the correlation function. One can then express the susceptibility through the connected correlation function as

$$\chi = \sum_j\langle s_i s_j\rangle_c. \tag{1.12}$$

The connected correlation function typically decays exponentially with the distance for large distances $|i-j|$ and allows the definition of a *correlation length* ξ through

$$\langle s_i s_j\rangle_c \sim \exp(-|i-j|/\xi). \tag{1.13}$$

The correlation length is a limit for the range in which physical phenomena may appear in the system. Physical phenomena can be seen as patterns of correlated spin values. They cannot adopt length scales significantly beyond the correlation length, as spins at further distances become largely uncorrelated and their relative behavior becomes random. Only if a system is at a critical point, which is described in Section 1.4, its correlation length diverges, and the connected correlation function does not decrease exponentially but only with a power-law. The susceptibility then diverges as well and phenomena of arbitrarily long range appear in the system. In case of a finite system, the correlation length will not diverge but instead approach the size of the system.

1.1.3 Structure of the Hamilton Function and Locality

An Ising-type Hamilton function typically consists of a sum of interaction terms. These terms are typically products of spins, known as *n-spin interactions*, that are summed over the whole system. Each term has a coefficient called the corresponding *coupling*. Interactions can be classified as ultra-local, local and non-local. *Ultra-local* means that the interaction occurs only between spins up to a certain distance, *local* means that it decreases exponentially with distance and *non-local* means that it decreases only up to a power-law.

A simple ultra-local interaction term is the 1-*spin interaction*

$$K_1 \sum_i s_i, \tag{1.14}$$

appearing in the Ising model as an external magnetic field. It implies that individual spins have states with higher and lower energies. The other ultra-local interaction term appearing in the Ising model is the *(*2*-spin) nearest-neighbor interaction*

$$K_2 \sum_{\langle i,j \rangle} s_i s_j, \tag{1.15}$$

with $\sum_{\langle i,j \rangle}$ being the sum over all neighboring sites i and j. A local, yet not ultra-local 2-spin interaction would instead be

$$K_3 \sum_{i,j\colon i \neq j} s_i s_j e^{-|i-j|}. \tag{1.16}$$

and two non-local versions of it would be

$$K_4 \sum_{i,j\colon i \neq j} \frac{s_i s_j}{|i-j|}, \quad K_5 \sum_{i,j\colon i \neq j} s_i s_j. \tag{1.17}$$

Another ultra-local interaction term involving more spins would be the 3-*spin interaction*

$$K_5 \sum_{i,j,k\colon i \neq j \neq k,\ |i-j|^2+|j-k|^2+|k-i|^2<6a^2} s_i s_j s_k, \tag{1.18}$$

which would also include spin triples that are slightly more distant than being pairwise nearest neighbors.

Ising-type models are generally local, meaning that they are built from local interactions. If they were not, they could not be related to known physics.

Considerations of how to parametrize the interactions of an unknown Hamilton function will become important at some stage in the search for the fixed point Hamilton function.

1.2 The Ising Model

1.2.1 Basic Properties of the Ising Model

The Ising model was originally designed as a model for a ferromagnet and because of that, the variables are called *spins*. Its applications were then extended to the study of phase transitions and critical phenomena and its importance increased with the discovery of universality. Today it still has many other applications. e.g. as a toy model for lattice field theory [1] and in neural network models [2].

The Ising model is an ultra-local Ising-type model with the Hamilton function

$$\mathcal{H}[s] = -J \sum_{\langle ij \rangle} s_i s_j - B \sum_i s_i. \tag{1.19}$$

Its first term represents nearest-neighbor interactions and the second term represents the influence of an external homogeneous magnetic field B. This thesis focuses on the ferromagnetic Ising model, assuming a positive value of J such that parallel spins are energetically favored over anti-parallel spins[1].

Of special interest in this theses is the Ising model with $B = 0$, which only includes nearest-neighbor interactions and obeys an additional $\mathbb{Z}(2)$ spin-flip symmetry, meaning that any expectation values are invariant under the global spin-flip transformation, changing the values of all spins. The magnetic field explicitly breaks this symmetry by favoring spins that are aligned with it. But it is zero at the critical point of any n-dimensional Ising model and an RG transformation, which will be introduced in Chapter 3, usually starts at the critical point of a system to approach a fixed point in theory space. Therefore, it is sufficient for this project to set it to zero, even though most concepts explained in this thesis could in principle be extended to a non-zero magnetic field.

The Ising Boltzmann weight, here denoted $W[s]$, is

$$W[s] = \exp(-\beta \mathcal{H}[s]) = \exp\left(\beta J \sum_{\langle ij \rangle} s_i s_j + \beta B \sum_i s_i \right), \tag{1.20}$$

and the partition function is

$$Z = \sum_{[s]} W[s] = \sum_{[s]} \exp(-\beta \mathcal{H}[s]) = (\prod_i \sum_{s_i = \pm 1}) \exp\left(\beta J \sum_{\langle ij \rangle} s_i s_j + \beta B \sum_i s_i \right). \tag{1.21}$$

The parameter J can be set to 1 w.l.o.g. when restricting to the ferromagnetic Ising model. This is possible because J only appears in the combination βJ and when J is set to 1, no parameters are actually lost as β still remains as a coupling parameter for the nearest-neighbor coupling. When we also set B to zero, the Hamilton function and the Boltzmann weight reduce to

$$\mathcal{H}[s] = -\sum_{\langle ij \rangle} s_i s_j, \quad W[s] = \exp\left(\beta \sum_{\langle ij \rangle} s_i s_j \right). \tag{1.22}$$

The Boltzmann weight can further be factored to local Boltzmann weights $w[s_i, s_j]$, one for

[1] An anti-ferromagnetic Ising model assuming negative values of J would yield completely different characteristics.

each combination of neighboring sites i, j, as

$$W[s] = \exp\left(\beta \sum_{\langle ij \rangle} s_i s_j\right) = \prod_{\langle ij \rangle} \exp(\beta s_i s_j) = \prod_{\langle ij \rangle} w[s_i, s_j], \quad \text{with} \quad w[s_i, s_j] = \exp(\beta s_i s_j). \tag{1.23}$$

1.2.2 The Link Representation and Loop Complexity

The term *link* may refer to the connections between neighboring lattice sites, commonly illustrated as lines. But, it is also used here to describe *link variables* that can be defined from two neighboring spins as $l_{\langle ij \rangle} = s_i s_j$. They take the values $+1$ for parallel and -1 for anti-parallel spins.

When B is set to zero, links can fully replace the spins[2], such that the Hamilton function, Boltzmann weight and partition function turn to

$$\mathcal{H}[l] = -\sum_{\langle ij \rangle} l_{\langle ij \rangle}, \quad W[l] = \exp\left(\beta \sum_{\langle ij \rangle} l_{\langle ij \rangle}\right), \quad Z = \sum_{[l]} W[l] = \sum_{[l]} \exp\left(\beta \sum_{\langle ij \rangle} l_{\langle ij \rangle}\right). \tag{1.24}$$

One may factor $W[l]$ to local weights $w[l_{\langle ij \rangle}]$. However, the partition function cannot be solved by simply applying the local sums $\sum_{l_{\langle ij \rangle}}$ to the local weights, because links are not independent variables and not all link configurations are physical. They originate from spins and their configurations must follow certain constraints that are related to loops in the lattice. Let us illustrate this issue on the example of a 2D square lattice and look at a section of four spins, displayed in Figure 1.3.

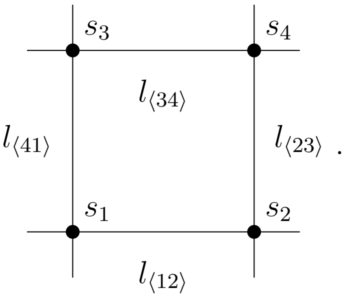


Figure 1.3: Elementary loop on a 2D square lattice.

One can follow a loop along the four links, called *elementary loop*, to end up back at the same site. A link configuration, with the first three links being $+1$ and the last being -1 would result in a contradiction in the spin values. One can identify the rule that products of links across each loop in the system must equal $+1$ to obtain a physical configuration. Larger loops within the system can then be reduced to elementary loops.

[2] In fact this also works for nonzero B, but then it is more complicated.

A finite 1D Ising system with open boundary conditions, consisting of n spins, does not contain any loops, such that any link configuration is physical and the partition function can indeed be solved by applying the local sums to the local weights as

$$Z = 2\prod_{\langle ij\rangle}\Big(\sum_{l_{\langle ij\rangle}=\pm 1}\exp\big(\beta l_{\langle ij\rangle}\big)\Big) = 2(2\cosh\beta)^{n-1}. \tag{1.25}$$

The overall factor 2 accounts for the fact that each link configuration corresponds to two different spin configurations related by spin-flip transformation. The power $n-1$ relates to the number of links in the system.

If periodic boundary conditions are chosen instead, there is one loop going through the whole system, adding complexity to the system. Solving the partition function takes some more transformation steps, as will be seen in the next section. The two 1D Ising systems with open and periodic boundary conditions are illustrated in figure 1.4.

In 2D, loops become omnipresent, and with increasing dimension also the number of loops per spin increases. This imposes a major difficulty on the analytical solutions of higher dimensional Ising models. Currently even the 3D Ising model remains unsolved.

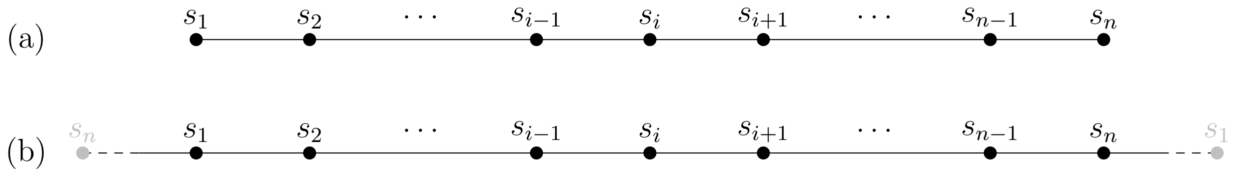


Figure 1.4: Illustration of two finite 1D Ising systems, each consisting of n spins. (a) shows the system with open and (b) with periodic boundary conditions.

1.3 Exact Solutions for the 1D Ising Model

The 1D and 2D Ising models can be solved analytically. This section shows two ways to solve the 1D Ising model with $B = 0$ and periodic boundary conditions.[3] Solving an Ising model means solving its partition function, which will allow to determine the configurations' probability distribution.

1.3.1 First Approach: Solution in the Link Representation

The 1D Ising model with periodic boundary conditions contains an equal number of spins and links, therefore the notation $l_{\langle ij\rangle}$ can be simplified to l_i where i refers to the spin to the left of

[3] These methods could be extended to the inclusion of a magnetic field.

the link, such that we obtain

$$\mathcal{H}[l] = -\sum_i l_i, \quad W[l] = \exp\left(\beta \sum_i l_i\right) = \prod_i \exp(\beta l_i). \tag{1.26}$$

The periodic boundary condition

$$\prod_i l_i = 1, \tag{1.27}$$

is related to the loop going through the whole system and limits the configurations to physical ones, ensuring that $s_{n+1} = s_1$. It can be included in the partition function by adding a Kronecker delta

$$\delta_{(\prod_i l_i),1} = \delta_{a[l],1}, \quad \text{with} \quad a[l] = \prod_i l_i. \tag{1.28}$$

With $a[l] \in \{-1, +1\}$, the Kronecker delta can be realized as

$$\delta_{a[l],1} = \frac{1}{2}(1 + a[l]). \tag{1.29}$$

The partition function can then be written as

$$Z = 2\sum_{[l]} \delta_{a[l],1} W[l], \tag{1.30}$$

and solved as

$$\begin{aligned} Z &= (\prod_i \sum_{l_i=\pm 1})(1 + \prod_i l_i) \prod_i \exp(\beta l_i) = (\prod_i \sum_{l_i=\pm 1}) \left((\prod_i \exp(\beta l_i) + \prod_i l_i \exp(\beta l_i) \right) \\ &= \prod_i \sum_{l_i=\pm 1} \exp(\beta l_i) + \prod_i \sum_{l_i=\pm 1} l_i \exp(\beta l_i) \\ &= \prod_i (\exp(\beta) + \exp(-\beta)) + \prod_i (\exp(\beta) - \exp(-\beta)) \\ &= (2\cosh\beta)^n + (2\sinh\beta)^n. \end{aligned} \tag{1.31}$$

Similarly, the correlation function [3] results in

$$\langle s_i s_j \rangle = \frac{1}{Z}\left((2\cosh\beta)^{n-|i-j|}(2\sinh\beta)^{|i-j|} + (2\cosh\beta)^{|i-j|}(2\sinh\beta)^{n-|i-j|} \right), \tag{1.32}$$

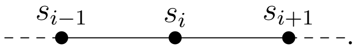
and the susceptibility becomes

$$\chi = \frac{1 - (\tanh\beta)^n}{1 + (\tanh\beta)^n} \exp(2\beta). \tag{1.33}$$

1.3.2 Second Approach: Transfer Matrix Method

The transfer matrix method is another way to formulate and solve the partition function of the 1D Ising model with periodic boundary conditions. Its core idea is to replace the summation over configurations in the partition function by matrix multiplications, such that one can use linear algebra to solve the partition function by diagonalization.

Let us consider a short segment of a 1D Ising model consisting of two links



(1.34)

We remain in the spin representation such that the two links relate to local weights $w[s_{i-1}, s_i]$ and $w[s_i, s_{i+1}]$. These are the only local weights that depend on the spin s_i. We can therefore combine the two weights and the sum over the values of spin s_i that appears in the partition function to form a new weight factor that depends on second nearest-neighbor spins

$$w^{(2)}[s_{i-1}, s_{i+1}] = \sum_{s_i=\pm1} w[s_{i-1}, s_i]w[s_i, s_{i+1}]. \tag{1.35}$$

The term resembles a matrix multiplication

$$(AB)_{\alpha\gamma} = \sum_{\beta} A_{\alpha\beta}B_{\beta\gamma} \tag{1.36}$$

of two matrices A and B. We can define a matrix τ containing the local weights of all combinations of spin values. Its rows correspond to the left spin and its columns to the right spin, such that

$$\tau = \begin{pmatrix} w[+1,+1] & w[+1,-1] \\ w[-1,+1] & w[-1,-1] \end{pmatrix} \quad \Leftrightarrow \quad (\tau)_{s_i,s_{i+1}} = w[s_i, s_{i+1}]. \tag{1.37}$$

We use τ to rewrite equation (1.35) to

$$w^{(2)}[s_{i-1}, s_{i+1}] := \sum_{s_i=\pm1} \tau_{s_{i-1},s_i}\tau_{s_i,s_{i+1}} = (\tau\tau)_{s_{i-1},s_{i+1}}, \tag{1.38}$$

where the sum $\sum_{s_i=\pm1}$ was integrated in the matrix product. We can add another neighboring spin, e.g. s_{i+2}, and iterate the procedure to obtain

$$w^{(3)}[s_{i-1}, s_{i+2}] := \sum_{s_i=\pm1}\sum_{s_{i+1}=\pm1} \tau_{s_{i-1},s_i}\tau_{s_i,s_{i+1}}\tau_{s_{i+1},s_{i+2}} = (\tau\tau\tau)_{s_{i-1},s_{i+2}}. \tag{1.39}$$

Another $(n-3)$ iterations of this procedure, when n is the number of spins in the system, yield the weight

$$w^{(n)}[s_1, s_{n+1}] := (\underbrace{\tau\ldots\tau}_{n})_{s_1,s_{n+1}} = (\tau^n)_{s_1,s_{n+1}}, \tag{1.40}$$

that includes all configurations at given s_1 and s_{n+1}. Implementing the periodic boundary conditions through $s_1 = s_{n+1}$ leaves us with the partition function being

$$Z = \sum_{s_1=\pm 1} (\tau^n)_{s_1,s_1} = \mathrm{Tr}\,\tau^n. \tag{1.41}$$

where Tr stands for the trace of a matrix.

τ can be diagonalized to obtain the eigenvalues λ_0 and λ_1

$$diag(\tau) = \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix} = \begin{pmatrix} 2\sinh\beta & 0 \\ 0 & 2\cosh\beta \end{pmatrix}, \tag{1.42}$$

such that the partition function is solved as

$$Z = (2\cosh\beta)^n + (2\sinh\beta)^n. \tag{1.43}$$

The transfer matrix method is a general and powerful that finds many applications in physics. It can be used to solve other statistical mechanical models, such as the Potts model [4], but it is also used in optics, e.g. when working with light rays propagating through a series of lenses [5]. Onsager managed to solve the 2D Ising model in 1944 [6] using the transfer matrix method. On the 2D square lattice, he constructed a large transfer matrix such that all configurations of an entire column were included in a single variable and the transfer matrix then related to two neighboring columns.

An approach inspired by the transfer matrix method is used in this thesis in Chapter 4 to formalize cluster-inspired RG in the 1D bond configuration.

1.4 Phase Transitions

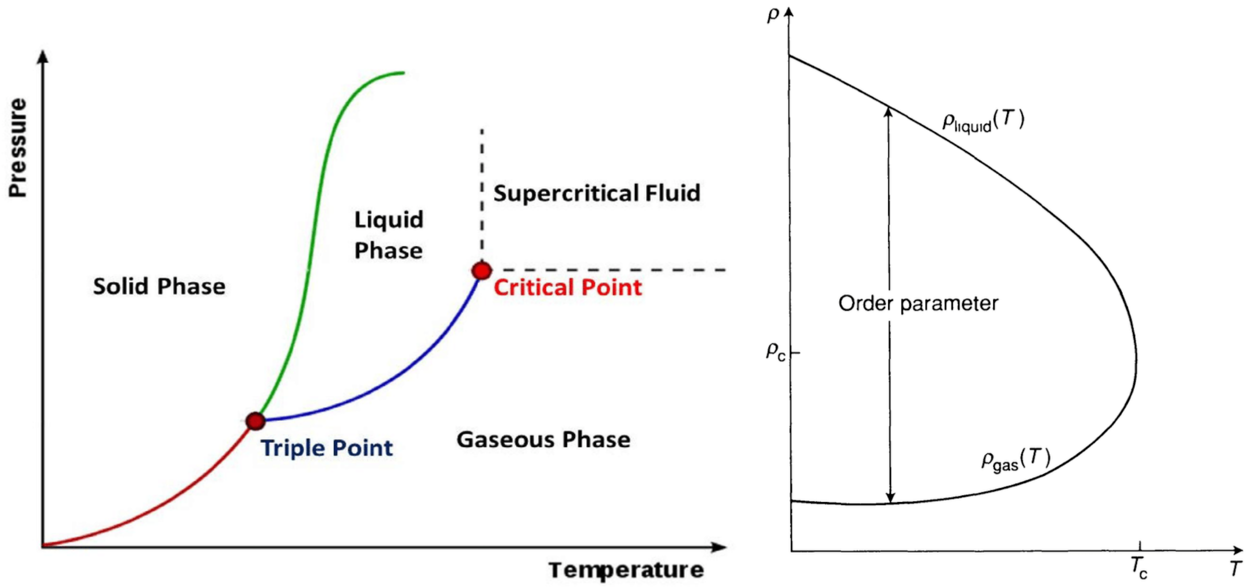
In thermodynamics, which has its foundations in statistical physics, phase transitions describe a drastic change in an infinite system's characteristics when a particular value of a *state variable* is crossed.

A most commonly known example of a phase transition in everyday life is boiling water [4]. Water, like most other substances, can be modelled as a thermodynamical system. It turns from a liquid to a gas when crossing a temperature of 100° C at a pressure of 1 bar, where temperature and pressure are two state variables causing the density to change in a discrete way. The density's first derivative diverges, which classifies the phase transition as *first order.* The finite difference in density between liquid and gas is called *order parameter.* A first order phase transition is also characterized by a latent heat[5] that is absorbed or set free while the state variable remains constant. In the case of boiling water, a finite amount of heat is absorbed from the burner while the temperature remains at 100° C.

[4] Of course boiling water is not an infinite system, but it can be approximated such.

[5] a finite amount of energy per volume

The liquid/gas phase transition in the thermodynamical model of water forms a continuous line of temperature and pressure values in the phase diagram, which is displayed in Figure 1.5 (a), while Figure 1.5 (b) shows the behavior of the order parameter. Water e.g. also boils a temperature of 180° C and a pressure of around 10 bar. However, the line ends at a point, the so-called *critical point*. Beyond the critical point, only a single phase exists, the so-called *high temperature phase*, also referred to as the *supercritical fluid phase*. Its properties change only continuously when temperature or pressure are varied.



(a)

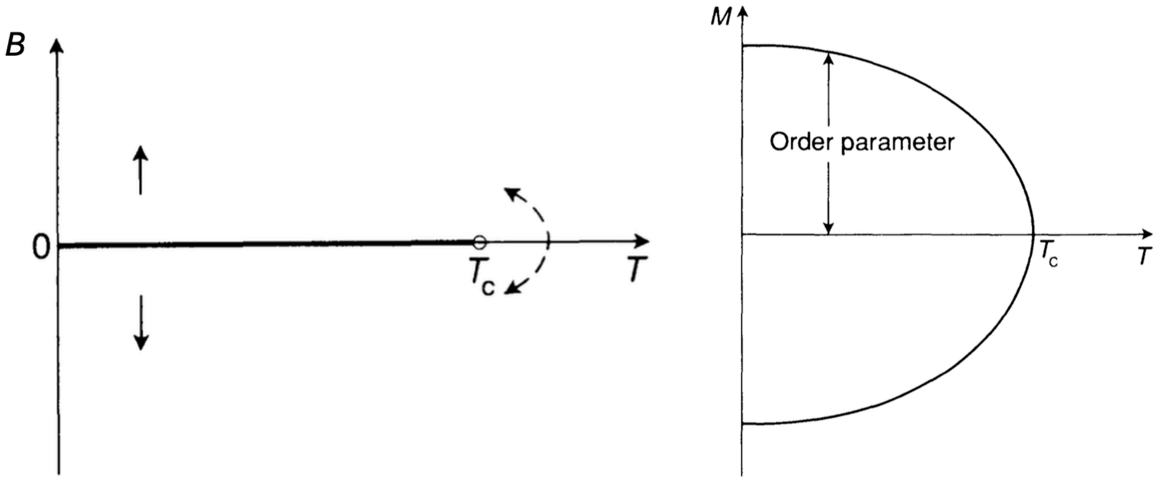
(b)

Figure 1.5: (a) shows the phase diagram of water [7]. The blue line represents its liquid/gas phase transition ending in the critical point when temperature and pressure are increased. (b) shows the behavior of the order parameter in the liquid/gas phase transition, which is the discrete change in density that occurs when the phase changes between liquid and gas [8]. ρ_c and T_c symbolize the critical density and temperature.

The ferromagnetic Ising model also has a first order phase transition in its continuum limit where the lattice spacing goes to zero and the number of sites becomes infinite. It happens at $B = 0$. B is its state variable. When it crosses zero, it causes a discrete change in magnetization.

The phase transition of the 1D Ising model does not end in a critical point but continues up to infinite temperatures.

The phase transitions of the 2D and higher dimensional Ising models do end in critical points. A phase diagram of a higher dimensional Ising model is displayed in Figure 1.6 (a). The critical point of the Ising model on the 2D square lattice is found at $\beta = \frac{1}{2}\ln\left(1+\sqrt{2}\right) \approx 0.440687$ [9], whereas that on the 2D triangular lattice is found at $\beta = \frac{1}{2}\ln\left(\sqrt{3}\right) \approx 0.274653$ [10]. The high-temperature phase above the critical temperature T_c is called *disordered phase*, while the two phases below T_c are called *ordered phases* and relate to equal positive or negative finite values of magnetization. The order parameter is the absolute value of the magnetization in any of the two ordered phases. Its behavior is displayed in Figure 1.6 (b).



(a) (b)

Figure 1.6: (a) shows the phase diagram of a two- or higher-dimensional Ising model [8], where the first order phase transition line ends in a critical point. T_c symbolized the critical temperature. The arrows symbolize the option to switch between the two ordered phases without going through the first order phase transition, by following a path around the critical point that involves nonzero values of B. (b) shows the behavior of the order parameter of Ising model phase transition [8], which is the absolute value of the magnetization in any of the two ordered phases.

As the Ising model at $B = 0$ obeys spin-flip symmetry, the two ordered phases are energetically equivalent. When the temperature crosses from a value above T_c to a value below T_c, it has to choose one of the two ordered phases and the spin-flip symmetry will be broken, as the magnetization's expectation value will then differ from zero. This phenomenon is called spontaneous symmetry breaking.

Critical points are classified as a *continuous* or *higher order* phase transitions. The order parameter is zero and the quantity related to it, which is the magnetization for the Ising model, is continuous. A divergency occurs only in one of the higher order derivatives of that quantity. However, at critical points the correlation length diverges and phenomena of arbitrary high length scales appear in the system.

Universality is an important concept related to phase transitions that allows us to relate the behavior of different physical models close to criticality. It was found that physical models that include phase transitions can be classified to universality classes [8, 11].

Two models in the same universality class show universal behavior in the infinitesimal neighborhood of the critical point. Many of their observables can be described by the same critical exponents. In Ising-type models, the correlation length, the expectation value of the magnetization and the susceptibility can close to the critical point be described as

$$\begin{aligned} \xi &\propto |T - T_c|^{-\nu} \\ \langle \mathcal{M} \rangle &\propto |T - T_c|^{\beta} \\ \chi &\propto |T - T_c|^{-\gamma}, \end{aligned} \tag{1.44}$$

with ν, β and γ being their *critical exponents*. The thermodynamical model of water with its liquid/gas phase transition belongs to the same universality class as the Ising model. Its observables have different physical meanings and yet they share the same critical exponents. The correlation length relates to that between the molecules, the magnetization corresponds to the density difference and the susceptibility relates to fluctuations in the density difference.

2 The Cluster Algorithm

2.1 The Concept of the Cluster Algorithm

2.1.1 The Monte Carlo Method

Monte Carlo is a method to simulate statistical systems numerically. It generates a sample that approximates the system's Boltzmann distribution. In case of Ising-type models, a *Monte Carlo sample* consists of a sequence of configurations which are generated by a *Monte Carlo algorithm.* A new a configuration is generated by modification of a previous one, usually with the use of random numbers. This type of sequence is called a *Markov chain.*

Expectation values of observables can be approximated by their mean values over the sample configurations, as described in equation (1.5). Convergence is ensured if the algorithm obeys two conditions known as *ergodicity* and *detailed balance.*

Ergodicity implies that an algorithm can reach any configuration in a finite number of steps. *Detailed balance* is defined as the requirement

$$W[s]P[s \rightarrow s'] = W[s']P[s' \rightarrow s], \tag{2.1}$$

where $P[s \rightarrow s']$ is the probability of the algorithm to switch from a configuration s to s'.

To estimate errors of observables mean values, the correlation between consecutive configurations must be considered. These are certainly correlated. Their correlation is transferred to more distant configurations in time with an exponential decrease. It can be described by the *autocorrelation time* τ through

$$\langle \mathcal{O}^{(t)} \mathcal{O}^{(t+n)} \rangle \propto \exp\Big(-\frac{n}{\tau}\Big), \tag{2.2}$$

where t is a time index relating to an arbitrary configuration in the Markov chain and n is the number of steps taken to arrive at configuration $t+n$.

τ provides a magnitude for the number of steps beyond which the the values of the observable become largely uncorrelated. It is specific to an observable and a used algorithm and must be considered in error estimation, described in Section 2.3. A short autocorrelation time implies a fast convergence, while a long autocorrelation time implies a slow convergence.

The autocorrelation time tends to increase with the correlation length in the system. Near a critical point, it can be described by a power-law [3]

$$\tau \propto \xi^z, \tag{2.3}$$

where z is called the *dynamical critical exponent* specific to an algorithm.

Monte Carlo algorithms do not need to follow a physical real-time dynamics. The cluster algorithm, for example, may at low temperatures flip large parts of the entire system at once. This would never happen in real-time in a physical system approximated by the Ising model such as a magnet, but it is legitimate to happen in a Monte Carlo simulation as long as it obeys detailed balance.

2.1.2 The Metropolis Algorithm

The Metropolis algorithm is a simple algorithm that flips a single spin at a time. It chooses a spin at random, compares the Boltzmann weight of the current configuration with that of the potentially new one and accepts the flip with a probability that obeys detailed balance. If it does, the new configuration is added to the Markov chain.

The Metropolis algorithm suffers from long autocorrelation times when the correlation length becomes large. Spins are then likely to be surrounded mostly by parallel spins and the Boltzmann weight of a potentially new configuration will in most cases be lower. The algorithm then hesitates to flip spins and its dynamics slows down. This manifests itself in two phenomena, called *critical slowing down* and *topological freezing.*

Critical slowing down [12] arises near the critical point where the correlation length becomes large. The dynamical critical exponent z was found to be around 2 by numerical evidence [3], implying that the autocorrelation time τ increases with the second power of the correlation length. In case of a finite system, the correlation length does not become infinite at the critical point but approaches the size of the system. Therefore, the dynamics of the Metropolis algorithm slows down with increasing system size at the critical point.

Topological freezing [13], also-called *supercritical slowing down*, occurs in the low temperature regime, where the system chooses one of the two ordered phases by spontaneous symmetry breaking. These phases of positive or negative magnetization can be regarded as topological sectors in the continuum limit. Once a phase is chosen, the system has no chance to escape from it and enter the other one. On a finite lattice system, that probability decreases exponentially with the number of spins, while the autocorrelation time increases exponentially and the performance of the algorithm breaks down.

2.1.3 The Cluster Algorithm

The cluster algorithm can flip multiple spins at a time. It does this by grouping the spins to clusters and then flipping clusters as a whole. This way, it is able to avoid the issues of critical slowing down and topological freezing.

In fact, numerical studies have found no significant deviation of the dynamical critical exponent from zero [3]. Achieving this characteristic requires the cluster algorithm to flip large areas of parallel spins simultaneously. The effective performance of the cluster algorithm at large correlation lengths depends on satisfying two conditions, which we term cluster logic:

1. Clusters only contain parallel spins.

2. Clusters may flip freely, independent of other clusters' orientations.

To introduce the cluster algorithm, we introduce additional variables known as *bond variables*, which enable neighboring spins to form *bonds* with each other. These bonds automatically partition the lattice into disjoint sets of bonded spins called *clusters*. A cluster originates at a given spin and includes all spins bonded to it.

Bond variables are denoted as $b_{\langle ij \rangle}$, where i and j are neighboring lattice sites. We define an activated bond as $b_{\langle ij \rangle} = 1$, meaning spins s_i and s_j belong to the same cluster, and a deactivated bond as $b_{\langle ij \rangle} = 0$, which does not necessarily imply the two spins are in different clusters. Spins s_i and s_j may still belong to the same cluster if connected via a series of activated bonds passing through other sites. This allows any two spins in the system to be bonded and belong to the same cluster.

While bonds can be defined as local variables between neighboring spins, the formation of clusters and the question of whether two distant spins are bonded are global issues that cannot generally be reduced to a local description. Clusters can be very large or consist of just a single spin. Figure 2.1 illustrates a section of a 2D square lattice Ising-type system with spins and bonds.

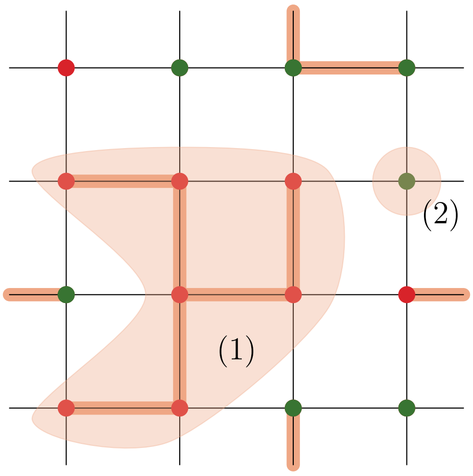


Figure 2.1: Illustration of a section of a 2D square lattice Ising-type system with spins and bonds. Red dots are up spins, green dots are down spins and orange lines are activated bonds. The two highlighted regions symbolize clusters, (1) consisting of seven spins, and (2) consisting of a single spin.

When introducing bonds as artificial variables, we must ensure that they do not alter the physics of the system. With bonds, we will introduce new weights that are functions of both the spins and the bonds. It is essential to ensure that the original weights, which are functions only of the spins, can be reproduced by summing the new weights over all possible bond states. Locally, this means

$$w[s_i, s_j] = \sum_{b_{\langle ij \rangle} = \pm 1} w[s_i, s_j, b_{\langle ij \rangle}], \tag{2.4}$$

where $w[s_i, s_j]$ is the original local weight as a function of the spins, and $w[s_i, s_j, b_{\langle ij \rangle}]$ is the new weight as a function of the spins and bonds, which needs to be defined in a way that satisfies the cluster logic.

The first condition of cluster logic can be fulfilled by ensuring that bonds are only activated between parallel spins. This is achieved by setting the weight of an activated bond to zero if the spins are anti-parallel.

The second condition of cluster logic is fulfilled if the weights resulting from the boundaries of the cluster remain invariant under cluster flipping. This condition can be satisfied by requiring that the weight of a deactivated bond is the same for both parallel and anti-parallel spins. A cluster is by definition surrounded by a boundary of deactivated bonds. Therefore, if the weight of a deactivated bond is the same for both parallel and anti-parallel spins, the weight of the boundary will remain unchanged.

With these conditions, the local Boltzmann weights as functions of the spin and bond configurations are uniquely defined to be

$$\begin{aligned} w[s_i s_j = +1, b_{\langle ij \rangle} = 1] &= e^{\beta} - e^{-\beta} := w_1, \\ w[s_i s_j = +1, b_{\langle ij \rangle} = 0] &= e^{-\beta} := w_0, \\ w[s_i s_j = -1, b_{\langle ij \rangle} = 1] &= 0, \\ w[s_i s_j = -1, b_{\langle ij \rangle} = 0] &= e^{-\beta} = w_0. \end{aligned} \tag{2.5}$$

or

$$w[s_i, s_j, b_{\langle ij \rangle}] = \delta_{b_{\langle ij \rangle},0} w_0 + \delta_{s_i,s_j} \delta_{b_{\langle ij \rangle},1} w_1. \tag{2.6}$$

This construction not only satisfies cluster logic but also allows for almost complete freedom in flipping clusters. Due to spin-flip symmetry, the Boltzmann weight remains unchanged when a cluster is flipped. Consequently, the condition of detailed balance is reduced to the simple requirement that clusters must be flipped with spin-flip symmetric probabilities.

The description of the Ising model with spin and bond variables is called the *cluster representation.* The cluster algorithm begins with an initial spin configuration, sets bonds according to the defined weights, and then determines which clusters to flip based on the specific version of the cluster algorithm, resulting in a new spin configuration. The bonds are then deleted, the spin configuration is added to the sample and the algorithm proceeds to the next step.

2.1.4 The Bond Representation of the Ising Model and its Solution in the 1D

We now aim to find the Boltzmann weights of bond configurations without specifying the spins, assuming that the spins follow their Boltzmann distribution. This requires summing over all spin configurations for a given bond configuration, a process called *integrating out* the spins. This is analogous to the condition implied in equation 2.4 when we introduced bonds as additional variables to the spins. There, transitioning from spins and bonds to only spins meant integrating out the bonds. Here, we want to retain only the bonds. However, the spins cannot be integrated out locally because their freedom to flip is constrained by the clusters, which are non-local objects. Therefore, we need to reconsider the cluster logic and its implications for a bond configuration:

- An activated bond corresponds to the weight w_1.
- A deactivated bond corresponds to the weight w_0.
- Clusters can flip as a whole to yield the same bond configuration again. Therefore, each cluster increases the weight of a bond configuration by a factor of 2.

From these arguments follows the weight of a bond configuration as

$$W[b] = 2^{\#Cl[b]} \prod_{\langle ij \rangle} (\delta_{b_{\langle ij \rangle},0} w_0 + \delta_{b_{\langle ij \rangle},1} w_1). \tag{2.7}$$

where $\#Cl[b]$ is the number of clusters in the bond configuration. The related partition function

$$Z = \sum_{[b]} 2^{\#Cl[b]} \prod_{\langle ij \rangle} (\delta_{b_{\langle ij \rangle},0} w_0 + \delta_{b_{\langle ij \rangle},1} w_1) \tag{2.8}$$

is hard to solve analytically due to the non-local character of $\#Cl[b]$. In 1D, however, the Ising model can actually be solved. This is feasible because in 1D, each deactivated bond (except the first one in the case of periodic boundary conditions) creates an additional cluster. Therefore, the number of clusters can be expressed as an analytical function of the bond configuration.

In the case of open boundary conditions, the number of clusters is equal to 1 plus the number of deactivated bonds. Therefore, all factors of 2 beyond the first one may be attributed to the local weight of a deactivated bond w_0. The Boltzmann weight then is

$$\begin{aligned} W[b] &= 2^{\#Cl[b]} \prod_{\langle ij \rangle} (\delta_{b_{\langle ij \rangle},0} w_0 + \delta_{b_{\langle ij \rangle},1} w_1) \\ &= 2^{1+\#(b=0)[b]} \prod_{i=1}^{n-1} (\delta_{b_i,0} w_0 + \delta_{b_i,1} w_1) = 2 \prod_{i=1}^{n-1} (2\delta_{b_i,0} w_0 + \delta_{b_i,1} w_1), \end{aligned} \tag{2.9}$$

and the partition function is

$$Z = \sum_{[b]} 2 \prod_{i=1}^{n-1} (2\delta_{b_i,0} w_0 + \delta_{b_i,1} w_1) = 2 \prod_{i=1}^{n-1} \sum_{b_i=0,1} ((2\delta_{b_i,0} w_0 + \delta_{b_i,1} w_1)) \tag{2.10}$$

$$= 2 \prod_{i=1}^{n-1} ((2w_0 + w_1)) = 2\left((2w_0 + w_1)\right)^{n-1}. \tag{2.11}$$

In the case of periodic boundary conditions, things are slightly more complicated due to the appearance of a loop in the system. Because of this loop, a configuration with all bonds activated and a configuration with one deactivated bond both yield a single cluster, as illustrated in Figure 2.2. Starting from the second deactivated bond, however, each additional one again corresponds to an additional cluster, and factors of 2 can again be attributed to deactivated bonds. Thus, the partition function can be derived as the sum of all weights with one or more deactivated

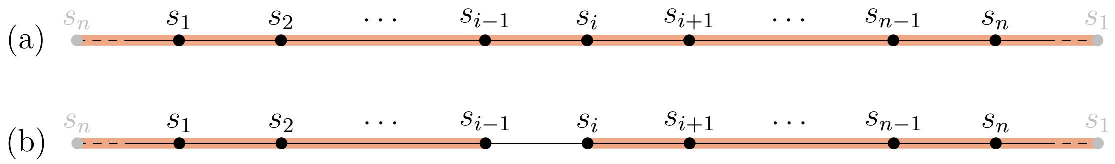


Figure 2.2: Illustration of two 1D bond configurations (a) and (b) with periodic boundary conditions, both yielding a single cluster.

bonds, which can be simplified similarly to the open boundary case, plus a single configuration with all bonds activated. It is

$$Z = \left((2w_0 + w_1)\right)^n + w_1^n. \tag{2.12}$$

2.2 Numerical Results: The Cluster Algorithm for the 2D Ising Model

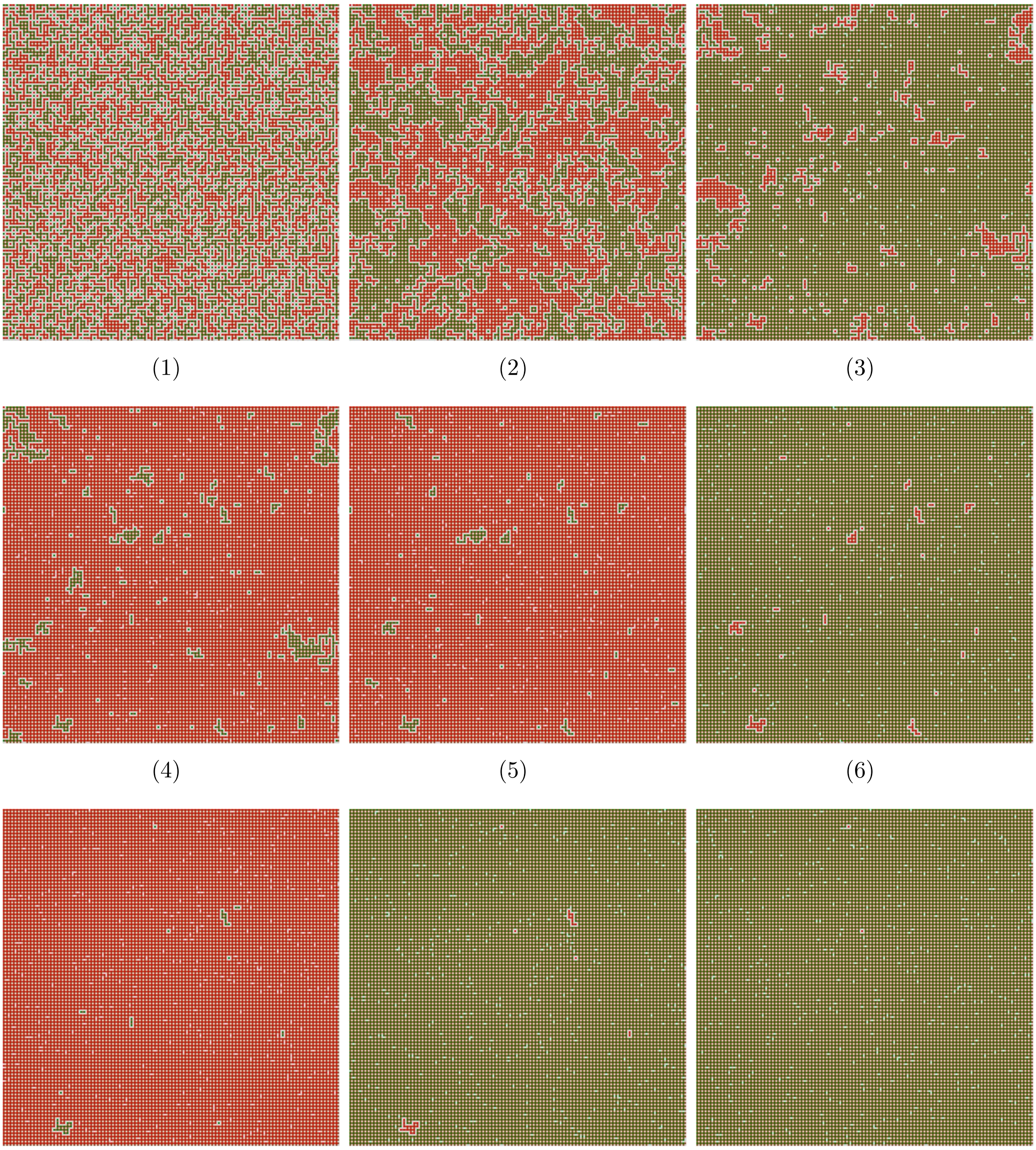
To become familiar with the cluster algorithm, a version of it called *Swendsen-Wang* cluster algorithm [14] was implemented on the square lattice. It flips all clusters with a 50% probability. Its implementation was veryfied using tabulated results of the susceptibility [15] on the 100×100 lattice.

Figures 2.3, 2.4 and 2.5 each show sequential 2D sample configurations generated using the Swendsen-Wang cluster algorithm at three different values of β. Each plot combines spin and bond data: the spin lattice is color-coded with red and green representing up and down spins, respectively. The bond lattice, displayed between spin pairs, is marked in dark brown for activated bonds, while deactivated bonds and empty spaces are marked in light grey. These plots provide an intuitive understanding of the cluster algorithm by comparing consecutive configurations.

Figure 2.3 shows plots generated at low temperature $\beta = 2$ on a 100×100 lattice starting from a random configuration. Initially, small cluster sizes are occur in the transient phase. However, due to the low temperature, spins tend to align, and the system quickly reaches majority alignment. Consequently, spins can form large clusters capable of flipping as a whole, transitioning between positive and negative ordered phases. In contrast, the Metropolis algorithm would typically suffer from topological freezing and remain in one phase.

Figure 2.4 displays plots generated at the critical temperature $\beta = \beta_{crit}$ on a 200×200, after omitting a sufficiently long transient phase. The lattice was chosen to be sufficiently large to enhance visibility of the emerging large-scale clusters. At the critical point, clusters appear across various length scales observable on the lattice.

Figure 2.5 illustrates plots generated at high temperature $\beta = 0.1$ on a 100×100 lattice, starting from a random configuration. At such high temperatures, all Boltzmann weights are similar, resulting in spins behaving almost randomly. As a result, the transient phase does not prominently stand out. The bond activation probability is very low, leading to spins predominantly forming single-spin clusters, represented dominantly by a light grey color across the lattice.

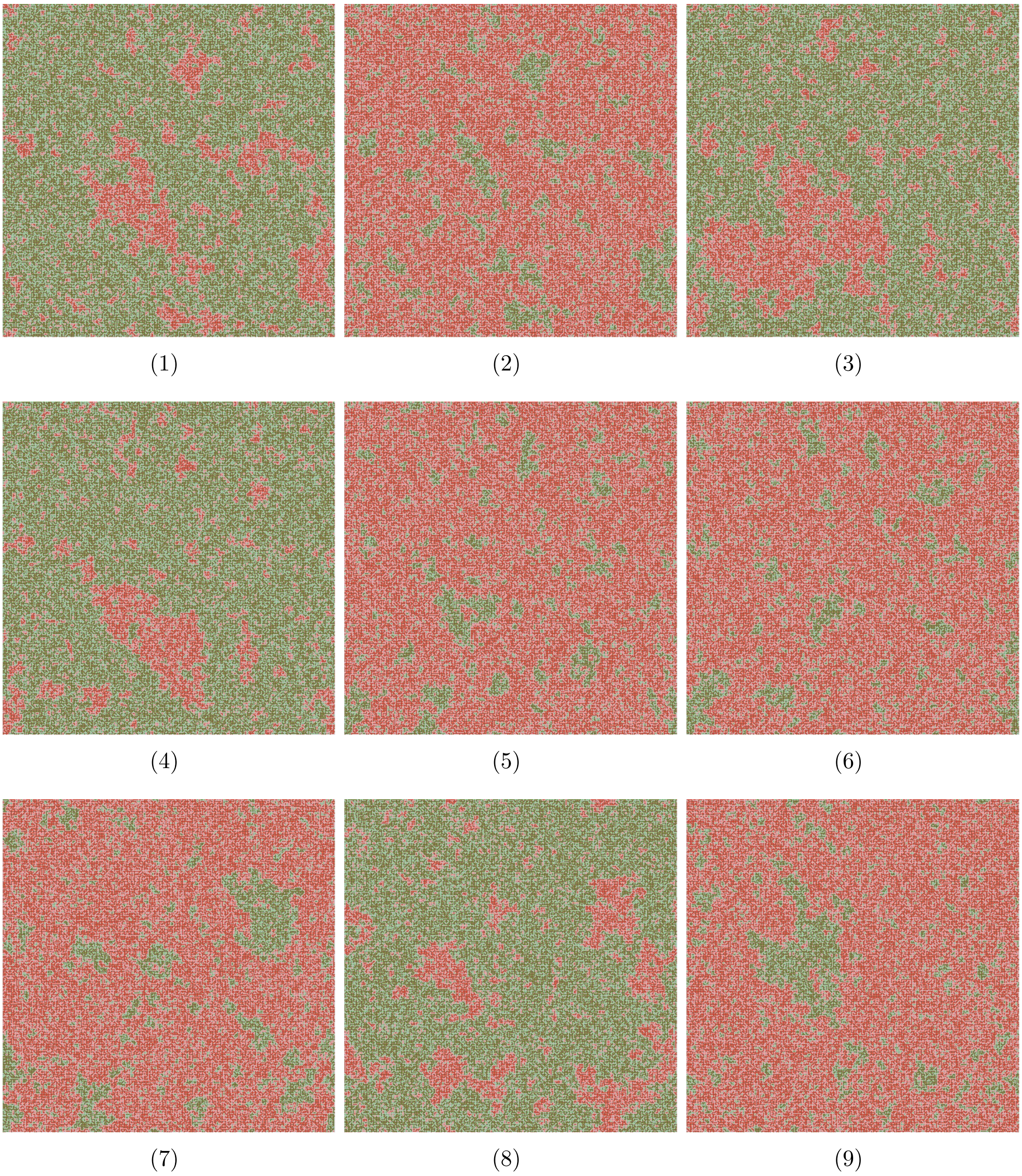


(1) (2) (3)

(4) (5) (6)

(7) (8) (9)

Figure 2.3: Sequential plots of 2D sample configurations generated using the Swendsen-Wang cluster algorithm at low temperature $\beta = 2$ on a 100×100 lattice, starting from a random configuration.

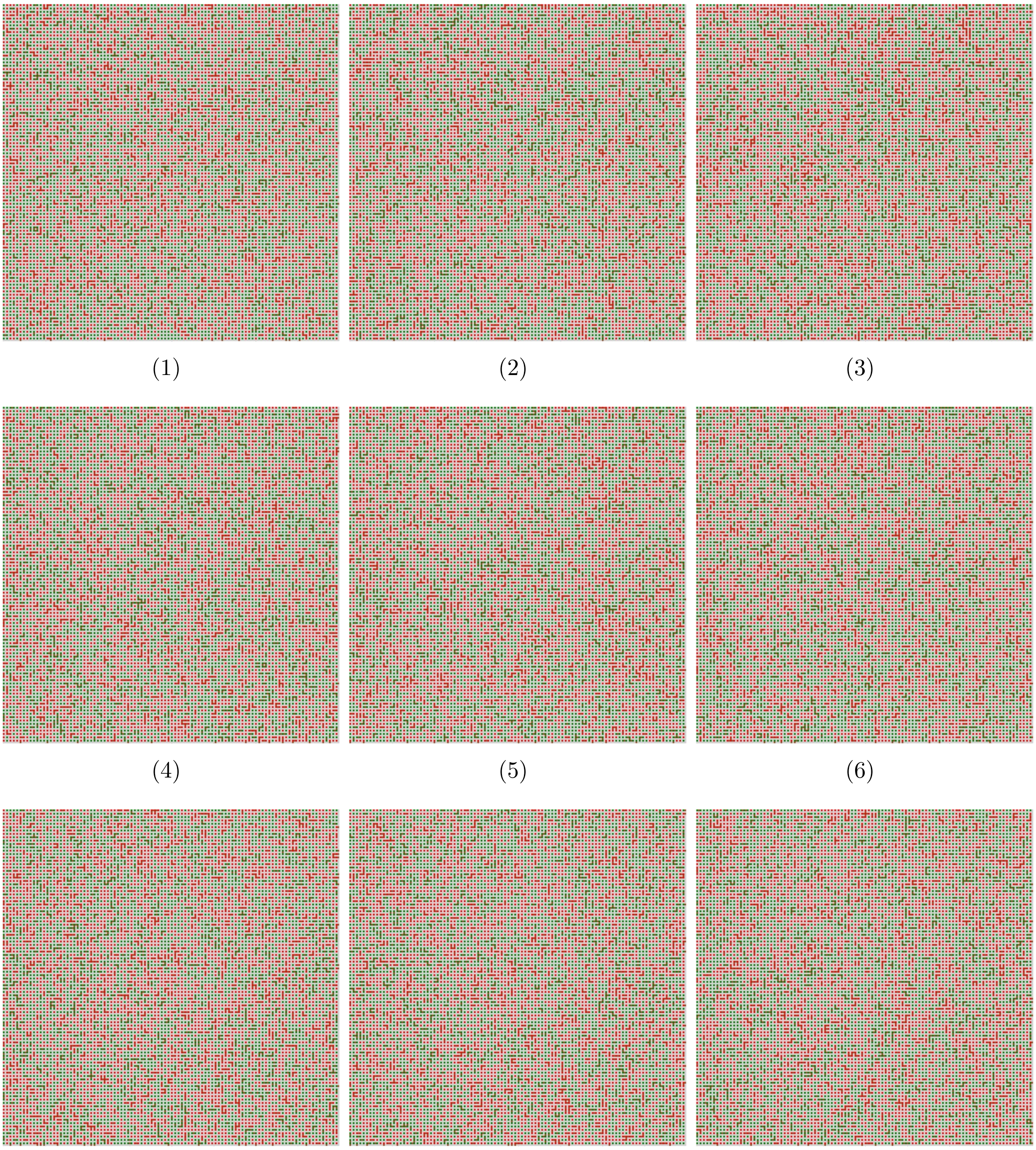


(1) (2) (3)

(4) (5) (6)

(7) (8) (9)

Figure 2.4: Sequential plots of 2D sample configurations generated using the Swendsen-Wang cluster algorithm at critical temperature $\beta = \beta_{crit}$ on a 200×200 lattice after omission of the transient phase.



(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

Figure 2.5: Sequential plots of 2D sample configurations generated using the Swendsen-Wang cluster algorithm at high temperature $\beta = 0.1$ on a 100×100 lattice, starting from a random configuration.

2.3 Error Analysis of Monte Carlo Data

A Monte Carlo simulation generates a sample that converges towards the Boltzmann distribution, such that mean values of observables converge towards their expectation values. However, as in any statistical model, there exists a deviation between the mean value of an observable in the sample and its expectation value, referred to as the *error*. Estimating this error is crucial for assessing the reliability of the results.

2.3.1 Error Estimation of Uncorrelated Data

In the case of uncorrelated data, one may estimate the error of a mean value using the *standard error*, defined as

$$\sigma_{\bar{\mathcal{O}}} = \frac{\sigma_{\mathcal{O}}}{\sqrt{n_s}}, \tag{2.13}$$

where $\sigma_{\mathcal{O}}$ is the standard deviation of the observable $\mathcal{O}$ and n_s is the sample size. The error decreases with sample size, suggesting that increasing the sample size can reduce the error if computational time permits.

Alternatively, one may estimate the mean value's error using the *bootstrap* method, which relies on *resampling*. This method involves generating a number of random *bootstrap samples* from the original sample's distribution, each of the same size as the actual sample. The mean value of the observable is computed for each bootstrap sample, and the error of the mean value is estimated based on their standard deviation [16]. Bootstrap assumes that the sample size is sufficiently large for its distribution to be representative of the entire system. This assumption can be verified, for example, by varying the sample size and comparing the results, generating new bootstrap samples of equal size as the original ones.

Both methods were used here to estimate the error of the mean value, with their outcomes generally similar, although the estimated errors from the bootstrap method tended to be slightly larger.

2.3.2 Binning

Monte Carlo algorithms typically generate correlated data, as configurations are sequentially generated from previous ones. The sequential correlation of an observable can be quantified by its autocorrelation time τ.

Binning is a method used to accurately estimate the error of sequentially correlated data by reducing the number of data points to obtain a smaller, uncorrelated dataset. This process involves grouping consecutive data points into bins of equal size, computing the average within each bin to create new, coarser data points that are less correlated. As the bin size increases relative to the autocorrelation time of the observable, the data points generated become increasingly uncorrelated.

Determining the appropriate bin size does not necessarily require knowing the autocorrelation time. Instead, one can analyze the behavior of the estimated error. Positive correlations result in an underestimation of the error, as correlated data points are more similar to each other than randomly chosen points would be. Increasing the bin size decreases the correlation, causing the estimated error to increase. Once the data become effectively uncorrelated, the estimated error becomes realistic and reaches a plateau. This plateau appears as a flat region in the curve, and its visibility increases when plotting the estimated error against the logarithmic bin size.

Further increasing the bin size beyond this point does not increase the estimated error further. Instead, it remains roughly constant. Conversely, increasing the bin size excessively will reduce the number of bins until the statistical validity of the error estimate is lost. Therefore, the most accurate error estimate is typically taken from the plateau region. Figure 2.6 shows an exemplary plot of an estimated error of a variable against the bin size.

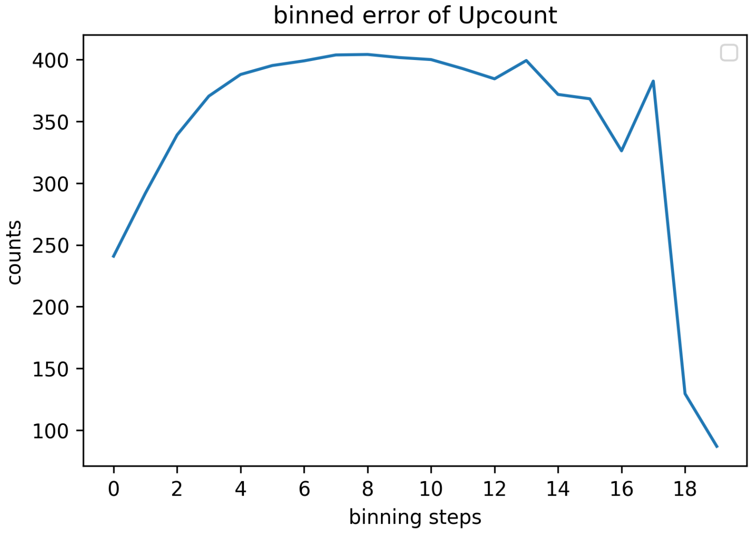


Figure 2.6: Plot showing the estimated error of the *upcount* variable, which counts the number of renormalized configurations with all spins up, generated by a cluster algorithm on a small renormalized triangular lattice (described in Chapter 5). Each binning step combines two bins into one, effectively doubling the bin size. The number of binning steps shown on the x-axis corresponds to $\log_2$ of the bin size. In this plot, the error estimate is approximately 400 counts, whereas the biased error estimate of the correlated data would have been around 250 counts.

2.3.3 The Transient Phase

Monte Carlo simulations encounter the issue of a transient phase, an initial period where errors induced by the choice of the initial configuration can affect observables. This phase is observable-specific and can be mitigated by discarding data from the initial configurations until the system reaches a steady state. The number of initial configurations to discard can be determined by plotting and visually inspecting the observable's values. Figure **??** illustrates the behavior of the energy density at the beginning of a Monte Carlo simulation, highlighting the transient phase.

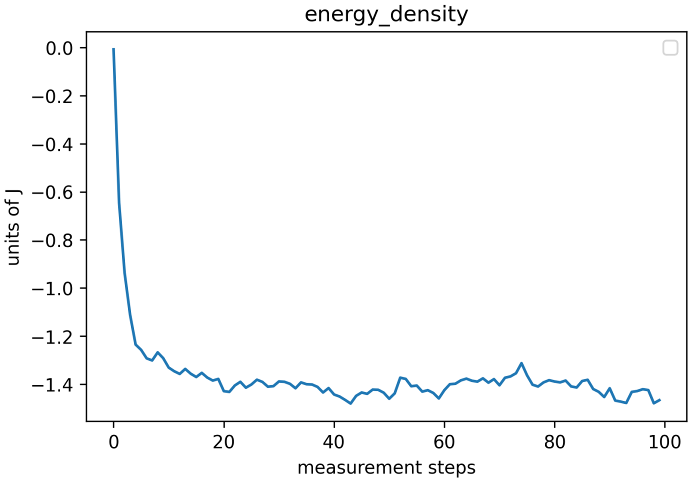


Figure 2.7: Initial values of the energy density in a sample generated by the Swendsen-Wang cluster algorithm on a 100×100 square lattice at $\beta = \beta_{\text{crit}}$. The visible transient phase persists for approximately 20 simulation steps.

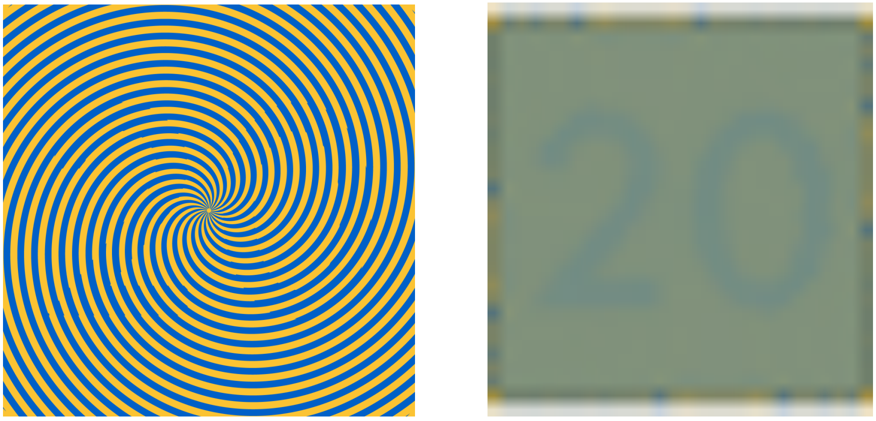
3 Cluster-Inspired RG

3.1 Renormalization Group Theory

The Renormalization Group (RG) is a concept that modifies a theory to emphasize its long-range phenomena [1]. This is achieved by eliminating short-range degrees of freedom while preserving long-range information. By removing dominant short-range effects that might obscure long-range phenomena, the renormalized theory reveals the underlying long-range behavior.

Kadanoff was the first to apply the concept of RG within classical statistical mechanics, particularly to Ising-type lattice models [11]. He introduced it through a blocking transformation, a process of coarse-graining on the lattice. This transformation groups lattice sites into blocks, creating new, coarser lattice sites in a renormalized system. New variables of the same type as the originals are defined on these blocks, replacing the original variables and absorbing some of their information. Despite losing some information as the lattice coarsens and the number of variables decreases, RG ensures that long-range phenomena persist through the transferred information. The blocking process can be iterated to further coarsen the lattice, hence the term 'renormalization group,' although mathematically it forms a semi-group since there is no inverse blocking transformation.

An intuitive analogy of RG on a lattice can be reducing the resolution of a picture. Although a picture lacks physical interactions, it may contain information about patterns at different length scales. Coarse-graining a picture results in the loss of short-range patterns, while long-range patterns remain visible. There is a type of optical illusions that include a dominant short-range pattern obscuring a weaker long-range pattern. This optical illusions may help to provide an intuitive understanding of RG. As resolution decreases and pixels are averaged, the short-range pattern vanishes, revealing the underlying long-range pattern. An example of such an optical illusion is provided in Figure 3.1.



(a) (b)

Figure 3.1: (a) shows an optical illusion with a hidden number 20. The strong contrast of short-range patterns formed by blue and yellow lines obscures the subtler and longer-range phenomenon of hidden number 20. By intentionally defocusing one's eyes to see the picture blurry, the number 20 may become visible. (b) provides an enlarged view of the optical illusion at a much lower resolution, where pixels have been integrated out. The short-range patterns formed by the blue and yellow lines have largely disappeared, leaving the number 20 as the dominant long-range pattern.

Kadanoff introduced RG blocking transformations as a method to identify critical points in Ising-type lattice models [11, 17]. In this approach, the lattice resolution is systematically reduced, leading to an increase in lattice spacing. Subsequently, the renormalized lattice is rescaled to restore the original lattice spacing. This process diminishes the correlation length and reduces the range of long-range phenomena in the system. Systems not at the critical point experience a decrease in correlation length with each blocking iteration, eventually converging to zero. Conversely, systems at the critical point are characterized by an infinite correlation length that persists through RG transformations, thereby remaining on the *critical surface.*

Wilson formalized RG theory and significantly advanced its understanding, along with the concept of universality [18, 19, 20, 21]. On a critical surface, RG theory identifies points of attraction known as fixed points. A theory at a fixed point remains invariant under RG transformations, exhibiting *scale-invariance*. This implies that physical properties remain unchanged when observed at different length scales. The basin of attraction of a fixed point on the critical surface defines a universality class of theories. Theories within the same universality class display identical critical behavior and share common critical exponents, irrespective of their microscopic details.

Wilson's work not only provided a deeper theoretical framework for understanding critical phenomena but also highlighted the profound idea that diverse physical systems can exhibit identical behavior near critical points, a concept now fundamental to the study of phase transitions and critical phenomena.

3.2 RG Blocking

The local character of an RG blocking transformation ensures that the renormalized system remains local. Multiple original spins are grouped into blocks of uniform size and shape that cover the entire lattice. The spins are then integrated out to form fewer renormalized spins on a coarser lattice. Each renormalized spin is determined by the spins within its block as well as by random numbers. An RG blocking transformation is characterized by a mathematical entity known as the *blocking kernel* $T[s, s']$, which specifies the weight of a renormalized configuration s' given an original configuration s. The combination of the original lattice and the renormalized lattice is collectively referred to as the *multigrid*, illustrated in Figure 3.2. The blocking kernel can be transformed into a probability by requiring

$$\sum_{s'} T[s, s'] = 1 \quad \Leftrightarrow \quad T[s, s'] = P(s'|s). \tag{3.1}$$

Local blocking implies that the blocking kernel is a local object

$$T[s, s'] = \prod_{\square} t[s_\square, s'_\square], \tag{3.2}$$

where the factor $t[s_\square, s'_\square]$ is called a *local blocking kernel.* $\square$ refers to a specific block in the system and $\prod_\square$ sums over all blocks. $s'_\square$ refers to the renormalized block spin and $s_\square$ is an abbreviation for the set of all original block spins. The local blocking kernel $t[s_\square, s'_\square]$ depends on the renormalized block spin $s'_\square$ and the set of original block spins $s_\square$. It forms the local weight of the renormalized spin $s'_\square$ for a given original block configuration $s_\square$.

The translational symmetry of lattices implies that each local blocking kernel should individually fulfil

$$\sum_{s'_\square} t[s_\square, s'_\square] = 1, \tag{3.3}$$

and therefore also the local blocking kernel can be interpreted as a local probability

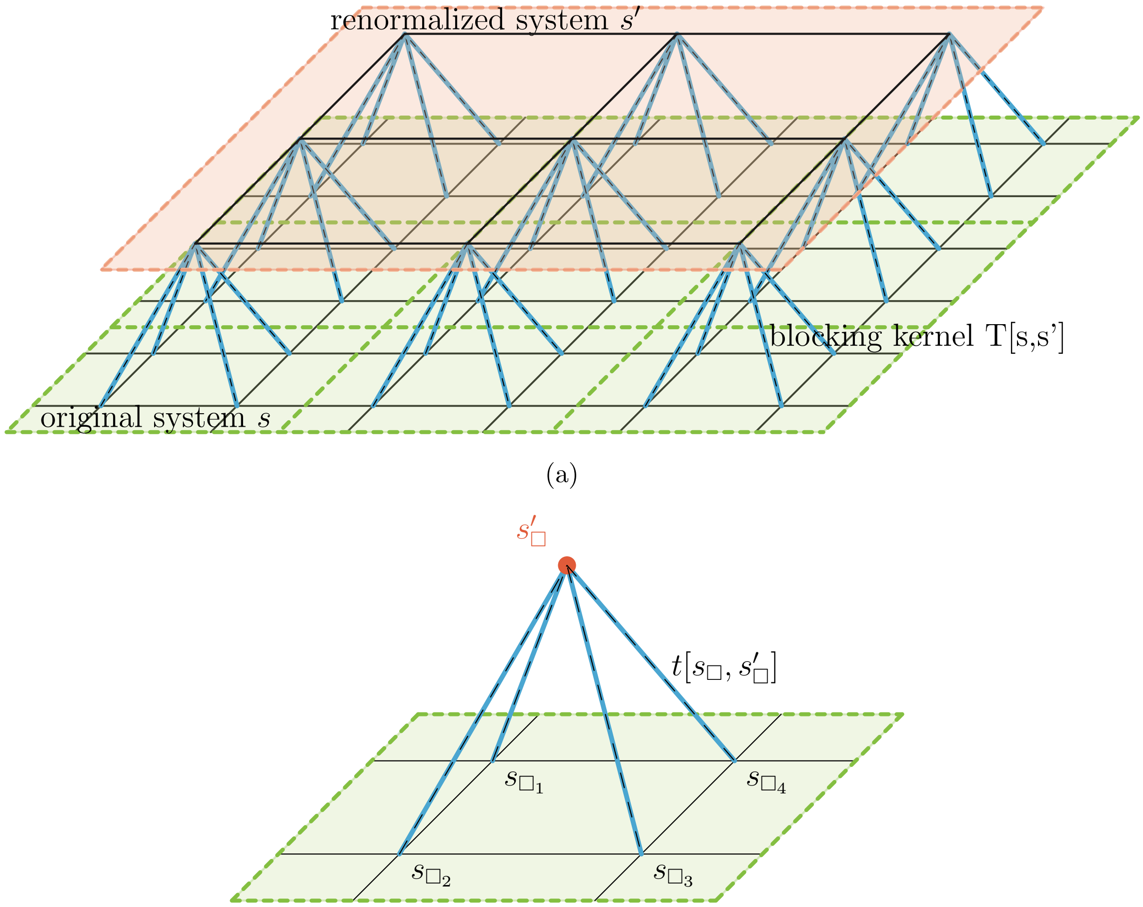
$$t[s_\square, s'_\square] = P(s'_\square|s_\square). \tag{3.4}$$

The weight of a given multigrid configuration consists of the weight of the original configuration times the blocking kernel

$$T[s, s']W[s]. \tag{3.5}$$

We would, however, like to know the weight of the renormalized configuration independently of the original configuration. It follows from integrating out the original configurations, as

$$W'[s'] = \sum_{[s]} T[s, s']W[s]. \tag{3.6}$$



(a)

(b)

Figure 3.2: Schematic illustration of an RG blocking transformation on the 2D square lattice, where (a) shows a section of the multigrid and (b) shows a single square-shaped block.

The renormalized weights $W'[s'] = \exp(-\beta\mathcal{H}'[s'])$ then implicitely determine the renormalized Hamilton function $H'[s']$. The renormalized partition function is equal to the original one if the blocking kernel is defined as a probability, as was done in equation (3.1). Then we obtain

$$\begin{aligned} Z' &= \sum_{[s']} \exp(-\beta H'[s']) \\ &= \sum_{[s]} \sum_{[s']} T[s,s'] \exp(-\beta\mathcal{H}[s]) \\ &= \sum_{[s]} \exp(-\beta\mathcal{H}[s]) = Z. \end{aligned} \tag{3.7}$$

3.3 Cluster-inspired RG

The core idea of this project was to develop and perform RG in a new way that allows to apply the cluster algorithm to a renormalized system. We call this concept *cluster-inspired RG transformation* or short *cluster-inspired RG*. The idea behind it is to pass the concept of clusters from the original system to the renormalized system, which will then have different interactions but still consist of clusters.

Before starting this project in form of multiple master theses, Prof. U.J. Wiese had already found a scheme for a cluster-inspired RG transformation, which formed the starting point of this project. The scheme can easily be iterated and allows to numerically simulate the renormalized system with the Monte Carlo cluster algorithm.

The cluster-inspired RG transformation is introduced in this section. Even if some illustrations are in 1D or in 2D, the concept in principle works for any dimension.

3.3.1 The First Blocking Step

We start operating in the cluster representation and introduce the cluster-inspired RG transformation via a blocking kernel. We denote the cluster-inspired blocking kernel as t_{cluster} and to define it, we introduce another variable type, so-called *kernel bonds*, which we denote b^{kernel}. We define kernel bonds such that they allow renormalized spins to bond to original spins. Kernel bonds will then translate original clusters to renormalized clusters. They may put renormalized spins into the same cluster by adding them to the same original cluster. An illustration on the concept of kernel bonds is given in Figure 3.3.

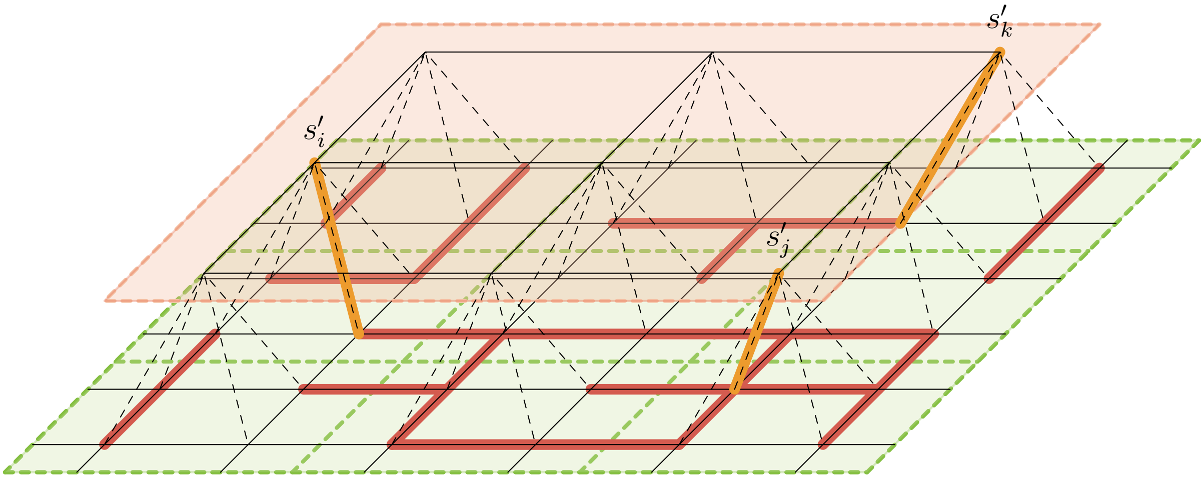


Figure 3.3: Illustration of the logical concept of cluster-inspired RG on a 2D square lattice. Red lines are activated original bonds, orange lines are activated kernel bonds, and all uncoloured lines are deactivated bonds. In this exemplary configuration, s'_i and s'_j belong to the same renormalized cluster, while s'_k does not.

We want to specify the blocking kernel as the weights related to the kernel bonds. In order to later be able to apply the cluster algorithm to the renormalized system we must insure that the kernel bonds respect cluster logic, introduced in Section 2.1.3.

Cluster logic implies that clusters can only contain parallel spins and that different clusters must have the freedom to flip independently.

The first condition implies that a renormalized spin can only be bonded to an original spin if the two are parallel. And when an original spin flips, a bonded renormalized spin must flip as well. This further implies that we should define the kernel bonds such that a renormalized spin can only bond to a single original spin. If we did not, and the kernel bond could be bonded to multiple original spins at once, this would immediately lead to contradictions. A renormalized spin that was bonded to two original spins at the same time, which are not part of the same original cluster, could not obey cluster logic if one of the original spins flipped while the other kept its orientation. If the two original spins were in the same cluster instead, bonding to both would not differ from bonding to just one of them. Therefore, kernel bonds may only bond a renormalized spin to a single original spin it its block.

The second condition is fulfilled, if we demand again that the probability of deactivating a bond is independent of the spins' orientations, such that the overall Boltzmann weight does not change when a cluster is flipped.

The blocking kernel should also respect the lattice symmetries, thus we can further demand that the weight of activating a bond should not depend on the location of the original spin in the block. Thus, we can define the blocking kernel to take only two distinct values related to an activated and a deactivated kernel bond. We define a as the weight of a deactivated kernel bond and c as the weight of an activated kernel bond if the bonded spins are parallel.

And finally, the new cluster-inspired blocking kernel should be consistent with the original one acting in the spin representation. We therefore require

$$t[s_\square, s'_\square] = \sum_{b_\square^{\text{kernel}}} t_{\text{cluster}}(s_\square, s'_\square, b_\square^{\text{kernel}}). \tag{3.8}$$

We now consider a block configuration where all original spins are up. The probabilities of the renormalized spin to be up or down at any given configuration of the original block spins are embodied by the local blocking kernel in the spin representation $t[s_\square, s'_\square]$. We denote the weight of it to be down as A and that of it to be up as $1 - A$, such that the two sum up to 1. A is an unspecified parameter that takes values between 0 and 1 and will later be identified as a tuning parameter of the RG transformation.

From our construction, we obtain the equations

$$\begin{aligned} A &= a, \\ 1 - A &= a + n_\square c, \end{aligned} \tag{3.9}$$

from which follow

$$\begin{aligned} a &= A, \\ c &= \frac{1-2A}{n_\square}. \end{aligned} \tag{3.10}$$

These reduce the domain of A to $[0, \frac{1}{2}]$. If we now define the values of the kernel bond as $b_\square^{\text{kernel}} = 0$ for no activated bond and $b_\square^{\text{kernel}} = 1$ for an activated bond to any one of the original spins, the cluster-inspired blocking kernel is specified as

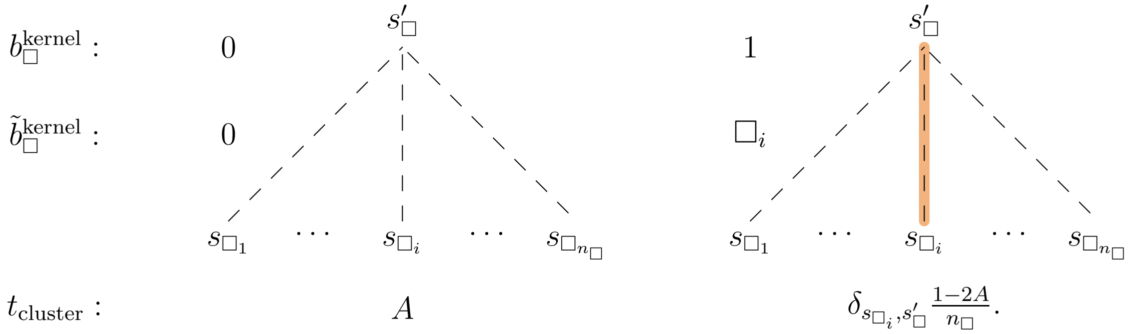
$$\begin{aligned} t_{\text{cluster}}(s_\square, s'_\square, b_\square^{\text{kernel}}) &= \delta_{b_\square^{\text{kernel}},0}\, a + \delta_{b_\square^{\text{kernel}},1} \sum_{i\in\square} \delta_{s_i,s'_\square}\, c \\ &= \delta_{b_\square^{\text{kernel}},0}\, A + \delta_{b_\square^{\text{kernel}},1} \sum_{i\in\square} \delta_{s_i,s'_\square} \frac{1-2A}{n_\square}, \end{aligned} \tag{3.11}$$

with $\delta_{\alpha,\beta}$ the Kronecker delta, where $\delta_{s_i,s'_\square}$ checks whether the original spin s_i and the renormalized spin $s'_\square$ are parallel and $\delta_{b_\square^{\text{kernel}},i}$ checks whether $b_\square^{\text{kernel}}$ was set to i.

If we alternatively define the values of the kernel bond as $\tilde{b}_\square^{\text{kernel}} = 0$ for no activated bond and to $\tilde{b}_\square^{\text{kernel}} = i$ for an activated bond to the original spin s_i within the block, then also its block-internal location is defined by its value. This version of the kernel bond can be useful for its implementation in a programming algorithm. In this case, the blocking kernel is specified as

$$\begin{aligned} t_{\text{cluster}}(s_\square, s'_\square, \tilde{b}_\square^{\text{kernel}}) &= \delta_{\tilde{b}_\square^{\text{kernel}},0}\, a + \sum_{i\in\square} \delta_{\tilde{b}_\square^{\text{kernel}},i}\, \delta_{s_i,s'_\square}\, c \\ &= \delta_{\tilde{b}_\square^{\text{kernel}},0}\, A + \sum_{i\in\square} \delta_{\tilde{b}_\square^{\text{kernel}},i}\, \delta_{s_i,s'_\square} \frac{1-2A}{n_\square}. \end{aligned} \tag{3.12}$$

The results can be illustrated as follows:


$b_\square^{\text{kernel}}$:
$\tilde{b}_\square^{\text{kernel}}$:
t_{cluster} :
0
0
$s'_\square$
$s_{\square_1}$
$\cdots$
$s_{\square_i}$
$\cdots$
$s_{\square_{n_\square}}$
A
1
$\square_i$
$s'_\square$
$s_{\square_1}$
$\cdots$
$s_{\square_i}$
$\cdots$
$s_{\square_{n_\square}}$
$\delta_{s_{\square_i},s'_\square} \frac{1-2A}{n_\square}$.

(3.13)

The case of $A = 0$ implies that each renormalized spin will certainly be bonded to an original spin. A positive value of A allows a renormalized spin to remain unbonded with a certain probability, such that it can flip freely, regardless of the original bond configuration. The limit of $A = \frac{1}{2}$ then corresponds to the trivial RG transformation where all renormalized spins remain unbonded and no information is passed from the original to the renormalized system, such that the renormalized system will become random.

3.3.2 Multiple Blocking Steps

The cluster-inspired blocking kernel can indeed be extended nicely to a relatively simple form describing multiple blocking steps at once.

A single blocking step partitions the original system into blocks of $n_\square$ original sites. Multiple blocking steps do this repeatedly. The renormalized system becomes more and more coarse and each renormalized spin relates to larger and larger blocks, that we call *block sectors*. After a number n_r of blocking steps, the block sectors have a size of $n_\square^{n_r}$ sites. One can imagine each blocking step to generate a new layer of the multigrid in between the original and the final renormalized system. Kernel bonds exist between any two layers of the multigrid. Starting at a renormalized spin, a series of kernel bonds makes its way through the layers of the multigrid until it may or may not reach the original system. If it does, the renormalized spin is bonded to a particular original spin and added to its cluster, just like after a single blocking step.

For positive values of A, the probability that a renormalized spin remains unbonded increases with the number of blocking steps, as in this case, there is a non-zero probability to discontinue the series of bonds added in each additional blocking step. In the case of $A = 0$, series of bonds are certain to continue all the way to the original system and each renormalized spin is certainly bonded to an original spin.

The block sectors that are formed by the iterated blocking are disjoint sets that contain all original spins that a particular renormalized spin can bond to. The bonding happens again with an equal probability for each original spin that is parallel to the renormalized spin, as each layer in the multigrid obeys the lattice symmetries.

We now derive the blocking kernel that describes multiple blocking steps at once. We do so by iterating the blocking kernel of a single blocking step, found in equation (3.11). Since we only care about the original and the renormalized systems and not about the in-between layers, we integrate these out by summing over all configurations of these layers and adding up the kernel weights. This procedure is simplified by the fact that each original spin may only be bonded to the renormalized spin of its own block sector by one particular path of activated kernel bonds. We thus want to know the weight that this particular series of bonds is activated. We denote the spins in the original system s_{i_o}, those in the final renormalized system s'_{i_r} and those in the intermediate layers $s^{\alpha}_{i_\alpha}$, with α the number of blocking steps taken to reach that layer. i_o, i_r, i_α are the lattice sites of the spins in their system. We obtain the probability for a bond between s_{i_o} and s'_{i_r} by

$$\begin{aligned}
&\sum_{s^{(1)}_{i_1}=\pm1}\sum_{s^{(2)}_{i_2}=\pm1}\sum_{s^{(3)}_{i_3}=\pm1}\cdots\sum_{s^{(n_r-1)}_{i_{n_r-1}}=\pm1}(\delta_{s_{i_o},s^{(1)}_{i_1}}c)(\delta_{s^{(1)}_{i_1},s^{(2)}_{i_2}}c)(\delta_{s^{(2)}_{i_2},s^{(3)}_{i_3}}c)\dots(\delta_{s^{(n_r-1)}_{i_{n_r-1}},s'_{i_r}}c)\\
&=c^{n_r}\sum_{s^{(1)}_{i_1}=\pm1}\sum_{s^{(2)}_{i_2}=\pm1}\sum_{s^{(3)}_{i_3}=\pm1}\cdots\sum_{s^{(n_r-1)}_{i_{n_r-1}}=\pm1}\delta_{s_{i_o},s^{(1)}_{i_1}}\delta_{s^{(1)}_{i_1},s^{(2)}_{i_2}}\delta_{s^{(2)}_{i_2},s^{(3)}_{i_3}}\dots\delta_{s^{(n_r-1)}_{i_{n_r-1}},s'_{i_r}}\\
&=c^{n_r}\delta_{s_{i_o},s'_{i_r}},
\end{aligned} \tag{3.14}$$

where we used that

$$\sum_{s_\beta=\pm 1} \delta_{s_\alpha,s_\beta}\delta_{s_\beta,s_\gamma} = \delta_{s_\alpha,s_\gamma}, \tag{3.15}$$

and assumed that s_{i_o} is in the block sector of s_{i_r} and all sites i_α in equation (3.14) are exactly the ones on the *kernel bond path* between s_{i_o} and s_{i_r}. This procedure is illustrated in Figure 3.4.

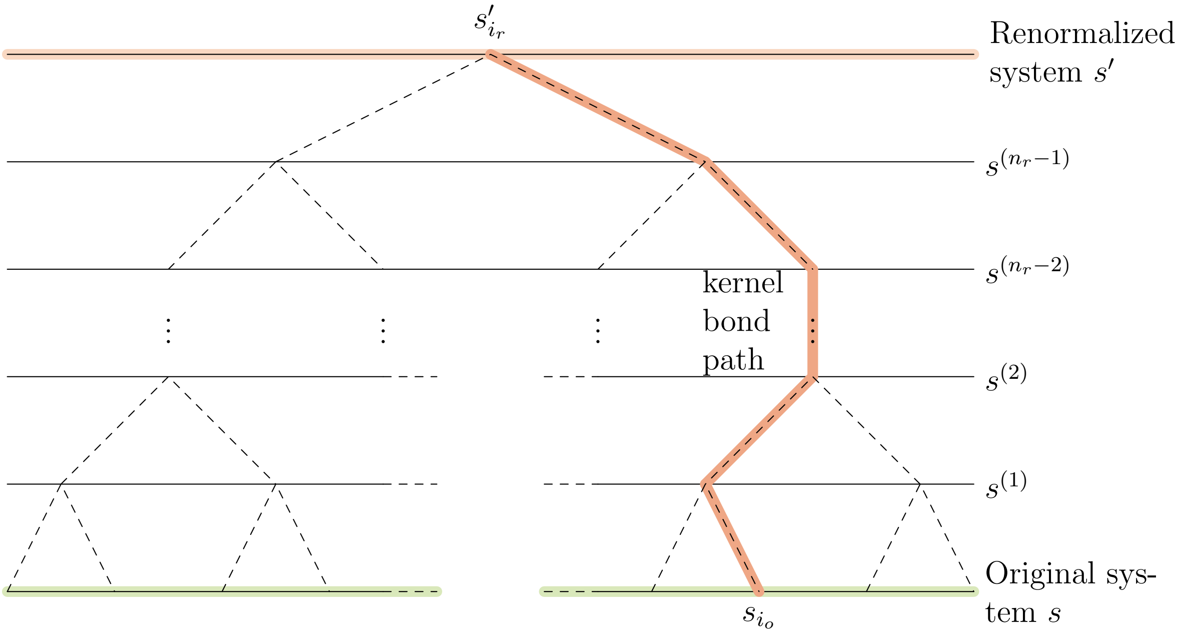


Figure 3.4: Simplified illustration of a multigrid after a number n_r of blocking steps, showing a series of bonds that bonds a renormalized spin s'_{i_r} to an original spin s_{i_o} in its block sector.

The reason why we did not have to include any kernel weights of blocks, nor sums over spins, that are not on the kernel bond path, is because we normalized the local blocking kernel to form a probability, such that

$$\sum_{s'_\square} t[s_\square, s'_\square] = \sum_{s'_\square, b^{\text{kernel}}_\square} t_{\text{cluster}}[s_\square, s'_\square, b^{\text{kernel}}_\square] = 1. \tag{3.16}$$

If we included all local blocking kernels and all sums that are not on the kernel bond path, then these would all simplify to factors of 1. Therefore, we can indeed neglect them.

To determine the blocking kernel of multiple blocking steps, we can look at a block sector configuration with all original spins up again, as we already did in the single blocking step. We denote the probability that the renormalized spin is down as $A^{(n_r)}$ and the probability that it is up as $1 - A^{(n_r)}$. We denote the weight that the renormalized spin remains unbonded as $a^{(n_r)}$. The weight that it is bonded to any one of the original spins in its block sector is c^{n_r} if they are parallel, with $c = \frac{1-2A}{n_\Box}$. We obtain the equations

$$A^{(n_r)} = a^{(n_r)}, \qquad 1 - A^{(n_r)} \;\; = a^{(n_r)} + n_\Box^{n_r} c^{n_r}. \tag{3.17}$$

From these follow

$$a^{(n_r)} = A^{(n_r)} = \frac{1-(n_\Box c)^{n_r}}{2} = \frac{1-(1-2A)^{n_r}}{2}, \tag{3.18}$$

and

$$c^{n_r} = \left(\frac{1-2A}{n_\Box}\right)^{n_r} = \frac{(1-2A)^{n_r}}{n_\Box^{n_r}}. \tag{3.19}$$

We now introduce an overall kernel bond that summarises the kernel bonds of all the n_r blocking steps as $b^{\text{kernel}}_{\Box^{(n_r)}}$. We define its values as it as $b^{\text{kernel}}_{\Box^{(n_r)}} = 0$ if the renormalized spin $s'_{\Box^{(n_r)}}$ remains unbonded and $b^{\text{kernel}}_{\Box^{(n_r)}} = 1$ if it is bonded to an original spin in the block sector $\Box^{(n_r)}$. We can then generalize the blocking kernel to relate to all the n_r blocking steps at once, as

$$\begin{aligned} t^{(n_r)}_{\text{cluster}}(s_{\Box^{(n_r)}}, s'_{\Box^{(n_r)}}, b^{\text{kernel}}_{\Box^{(n_r)}}) &= \delta_{b^{\text{kernel}}_{\Box^{(n_r)}},0}\, a^{(n_r)} + \delta_{b^{\text{kernel}}_{\Box^{(n_r)}},1} \sum_{i\in\Box^{(n_r)}} \delta_{s_i,s'_{\Box^{(n_r)}}}\, c^{n_r} \\ &= \delta_{b^{\text{kernel}}_{\Box^{(n_r)}},0} \frac{1-(1-2A)^{n_r}}{2} + \delta_{b^{\text{kernel}}_{\Box^{(n_r)}},1} \sum_{i\in\Box^{(n_r)}} \delta_{s_i,s'_{\Box^{(n_r)}}} \frac{(1-2A)^{n_r}}{n_\Box^{n_r}}, \end{aligned} \tag{3.20}$$

where $\frac{1-(1-2A)^{n_r}}{2}$ is the weight to leave the renormalized spin unbonded and $\frac{(1-2A)^{n_r}}{n_\Box^{n_r}}$ is the weight to set a bond between the renormalized spin and any one of the block sector spins if the two are parallel.

The found results provide a method to simulate a renormalized system by means of the cluster algorithm. We therefore start with initial original and renormalized spin configurations. We set the original bonds and the kernel bonds according to the found rules, such that bonded renormalized spins become part of the original cluster, and unbonded renormalized spins form their own single-spin cluster. We may then flip the clusters in a spin-flip symmetric way as we like to obtain a new original and renormalized configuration. When this is done repeatedly, the frequency of each renormalized configuration will converge to be proportional to its Boltzmann weight. This then allows, in principle, to determine of the renormalized Hamilton function, even if the way to do so needs to be further specified. This method is further specified and used in Chapter 5.

The derivation could be extended to choosing individual values of the tuning parameter A for each blocking step. The procedure would work analogously, with the difference that A would have to be replaced by a set of tuning parameters $[A] = [A_i, \ldots, A_{n_r}]$, where A_k is the tuning parameter of the k-th blocking step. Also the weight c would have to be replaced by weights c_k, such that

$$c_k = \frac{1-2A_k}{n_\Box}. \tag{3.21}$$

Instead of c^{n_r}, one would obtain

$$c^{(n_r)} = \prod_{k=1}^{n_r} c_k = \frac{\prod_{k=1}^{n_r}(1-2A_k)}{n_\square^{n_r}}, \tag{3.22}$$

and $a^{(n_r)}$ would turn to

$$a^{(n_r)} = A^{(n_r)} = \frac{1-\prod_{k=1}^{n_r}(1-2A_k)}{2}. \tag{3.23}$$

The more general form of the local blocking kernel written as a function of the set of tuning parameters $[A]$ would turn to

$$\begin{aligned} t^{(n_r)}_{\text{cluster}}(s_{\square^{(n_r)}}, s'_{\square^{(n_r)}}, b^{\text{kernel}}_{\square^{(n_r)}}, [A]) &= \delta_{b^{\text{kernel}}_{\square^{(n_r)}},0} a^{(n_r)} + \delta_{b^{\text{kernel}}_{\square^{(n_r)}},1} \sum_{i\in\square^{(n_r)}} \delta_{s_i,s'_{\square^{(n_r)}}} c^{(n_r)} \\ &= \delta_{b^{\text{kernel}}_{\square^{(n_r)}},0} \frac{1-\prod_{k=1}^{n_r}(1-2A_k)}{2} \\ &+ \delta_{b^{\text{kernel}}_{\square^{(n_r)}},1} \sum_{i\in\square_i^{(n_r)}} \delta_{s_i,s'_{\square^{(n_r)}}} \frac{\prod_{k=1}^{n_r}(1-2A_k)}{n_\square^{n_r}}. \end{aligned} \tag{3.24}$$

One could, if necessary, also generalize the blocking kernel even further and vary block sizes in each step. Then also $n_\square$ would need to be labelled with a blocking step index k. However, this was not seen as particularly useful for the purposes here.

4 1D Analytical Cluster-Inspired RG

The first part of this project after a general introduction towards cluster-inspired RG was the attempt to analytically solve it in 1D. The idea was to get started with the simplest case and try to progress analytically to either successfully solve it or at least get to know the limits of an analytical approach.

4.1 General 1D RG Ansatz

Before approaching cluster-inspired RG, we started with a general 1D RG transformation acting in the spin representation. We found a general form of a blocking kernel with blocks consisting of two original spins that are transformed to one renormalized spin as illustrated in Figure 4.1. While the original system is assumed as an Ising system with $B = 0$, the renormalized system has a priori unknown interactions.

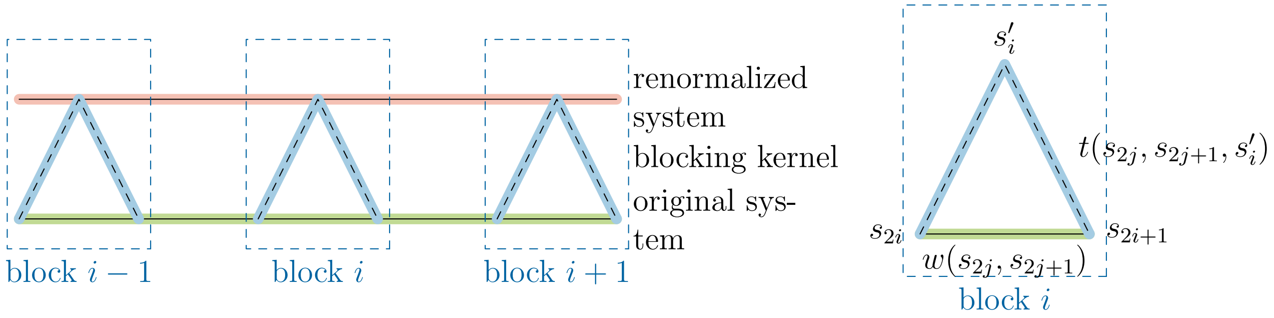
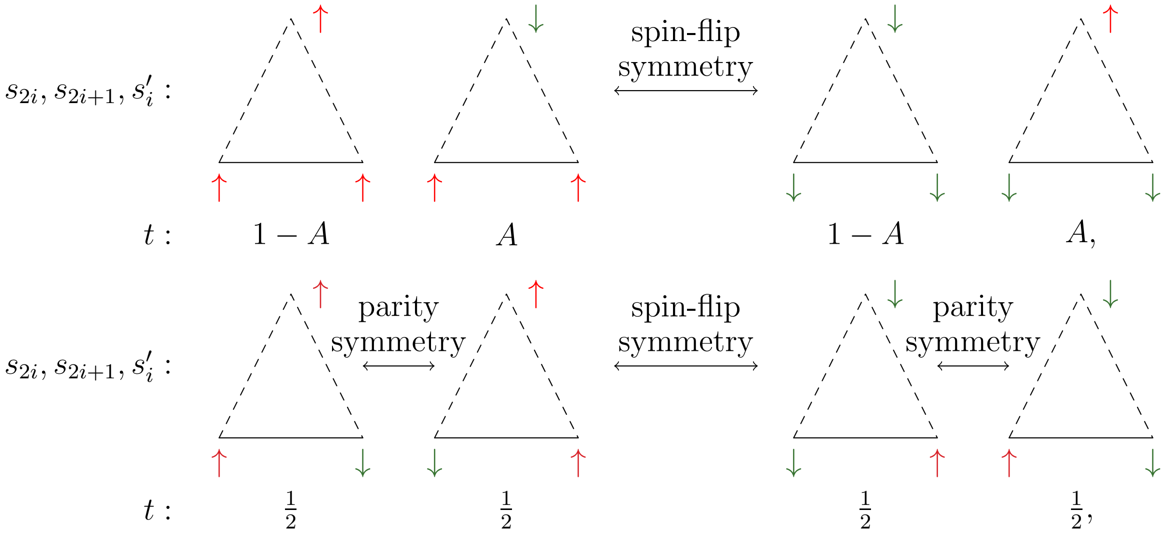


Figure 4.1: Illustration of a single blocking step. The green lines represent the original system, the blue lines represent the blocking kernel and the orange lines represent the renormalized system.

The local blocking kernel is defined as a probability by

$$\sum_{s'_i} t(s_{2i}, s_{2i+1}, s'_i) = 1 \quad \Leftrightarrow \quad t(s_{2i}, s_{2i+1}, s'_i) = P[s'_i | s_{2i}, s_{2i+1}]. \tag{4.1}$$

The degrees of freedom of such a blocking kernel are few. The 1D lattice's parity symmetry and the Ising model's spin-flip symmetry constrain the blocking kernel and leave only one degree of freedom to its definition. The resulting parameter can be identified as the same parameter A that we already encountered in Section 3.3 when performing cluster-inspired RG in general dimensions. The blocking kernel takes the values



(4.2)

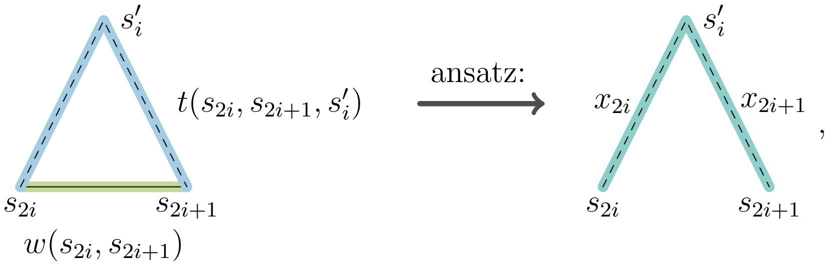
or formulated as an equation

$$t(s_{2i}, s_{2i+1}, s'_i) = (1-A)\delta_{s_{2i},s_{2i+1}}\delta_{s_{2i},s'_i} + A\delta_{s_{2i},s_{2i+1}}(1-\delta_{s_{2i},s'_i}) + \frac{1}{2}(1-\delta_{s_{2i},s_{2i+1}}). \tag{4.3}$$

4.2 1D Ising-Interaction-Preserving RG Ansatz

After constructing a general 1D blocking kernel, we wanted to find an RG transformation that allowed to preserve the Ising interactions by setting the tuning parameter A accordingly. This specific value of A might later be interesting when working on cluster-inspired RG.

Our ansatz to preserve the Ising interactions originated in the idea of factoring the local weight of a block together with the local blocking kernel $t(s_{2i}, s_{2i+1}, s'_i)w(s_{2i}, s_{2i+1})$ into two spatially separate variables $x(s_{2i}, s'_i)$ and $x(s_{2i+1}, s'_i)$ or short x_{2i} and x_{2i+1}, each of them related to one side of the block, illustrated schematically as

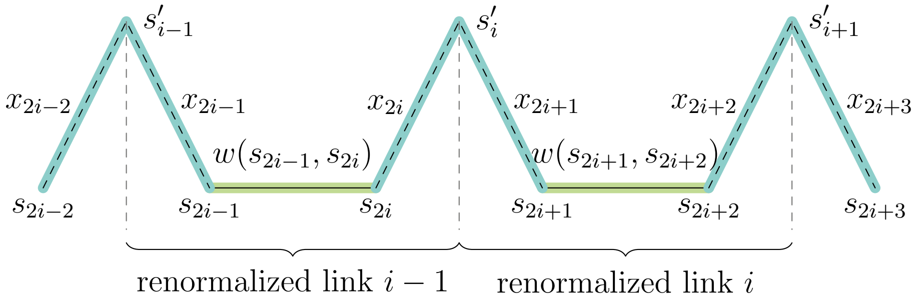


(4.4)

and formulated as an equation

$$t(s_{2i}, s_{2i+1}, s'_i)w(s_{2i}, s_{2i+1}) = x(s_{2i}, s'_i)x(s_{2i+1}, s'_i) = x_{2i}x_{2i+1}. \tag{4.5}$$

The ansatz is inspired by the link representation of the Ising model. It implies that the renormalized system can again be separated into links connecting neighboring spins. This can be visualized as

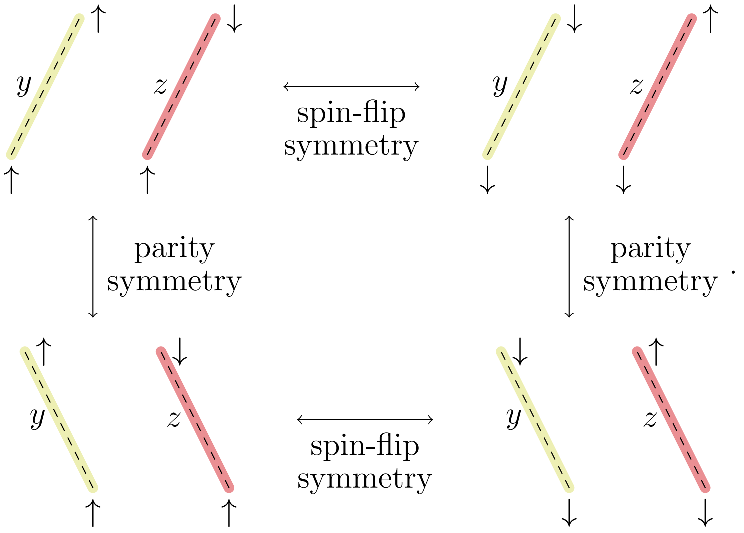


, (4.6)

where a renormalized link consists of the kernel and the original link in between the two neighboring blocks. To determine the renormalized configuration's weight, the original spins need to be integrated out and the local weight of the renormalized link i can be written as a function of s'_i and s'_{i+1} as

$$w'[s'_i, s'_{i+1}] = \sum_{s_{2i+1}} \sum_{s_{2i+2}} x(s_{2i+1}, s'_i) w(s_{2i+1}, s_{2i+2}) x(s_{2i+2}, s'_{i+1}). \tag{4.7}$$

This again represents a nearest-neighbor interaction, thus the ansatz preserves the Ising interactions. We are not finished however, as we have not yet found suitable weights for x_i. To do so, we must again consider the symmetries of the system. Parity implies that it should not matter whether x_i is on the left or on the right of a renormalized block spin, therefore the uniform name x_i makes sense. Spin flip symmetry implies that only the relative orientations of the two spins should matter. This allows for two different weights assigned to x_i, related to parallel and anti-parallel spins, that we call y and z:



. (4.8)

Inserting the different combinations of spins in equation (4.5) yields the set of equations

$$\begin{aligned} 1 - Ae^{\beta} &= y^2, \\ Ae^{\beta} &= z^2, \\ \frac{1}{2}e^{-\beta} &= yz, \end{aligned} \tag{4.9}$$

and specifies A as

$$A = \frac{1}{2}(1 \pm \sqrt{1 - e^{-4\beta}}). \tag{4.10}$$

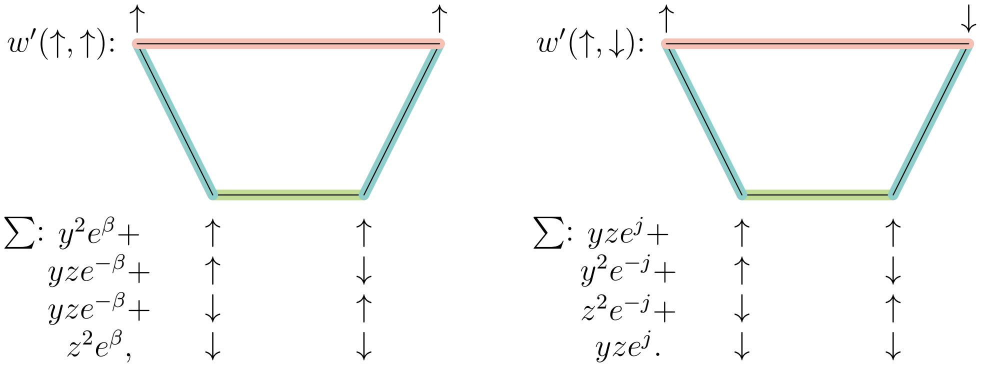
Since our renormalized system is again a standard Ising model, it can again be fully characterized by an inverse temperature β'. We relate β' to β, the inverse temperature of the original system, by comparing the weight ratios of parallel versus anti-parallel spins of the original and the renormalized system. With the local Ising Boltzmann weight specified in equation (1.23) follows

$$\frac{w(\uparrow,\uparrow)}{w(\uparrow,\downarrow)} = e^{2\beta}, \tag{4.11}$$

and thus

$$\frac{w'(\uparrow,\uparrow)}{w'(\uparrow,\downarrow)} = e^{2\beta'}, \tag{4.12}$$

from which we can determine β'. We use equation (4.7) to determine $w'(\uparrow,\uparrow)$ and $w'(\uparrow,\downarrow)$ integrating out the original spins as



(4.13)

The weight ratio is

$$\frac{w'(\uparrow,\uparrow)}{w'(\uparrow,\downarrow)} = \frac{y^2e^{\beta} + yze^{-\beta} + yze^{-\beta} + z^2e^{\beta}}{yze^{\beta} + y^2e^{-\beta} + z^2e^{-\beta} + yze^{\beta}} = e^{2\beta'}, \tag{4.14}$$

and using equations (4.9), β' is found as

$$\beta' = \frac{1}{2}\ln(\cosh 2\beta). \tag{4.15}$$

The value found for A in equation 4.10 depends on β. The fact that β changes in each blocking step implies that also the parameter A needs to be adjusted in every blocking step to preserve the Ising interactions.

4.2.1 1D Ising Model Fixed Points

The RG ansatz to preserve the Ising interactions can be used to find fixed points within the 1D Ising model by iteration of the RG transformation. The behavior of the coupling β with the iteration is displayed in Figure 4.2 as an RG flow in the one-dimensional parameter space. It converges to the *trivial* or *high-temperature fixed point* at $\beta = 0$, or $T \to \infty$, where all Boltzmann weights are 1 and all configurations are equally probable, thus random. This trivial fixed point describes a system without any interactions and correlation length 0.

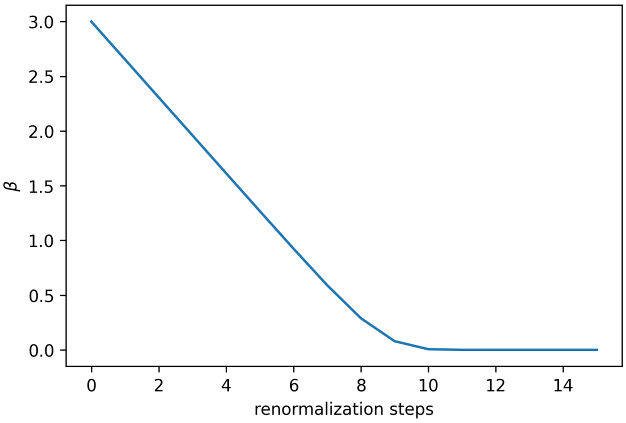


Figure 4.2: RG flow of the Ising-interaction-preserving RG transformation with initial value $\beta = 3$ and converging to $\beta = 0$.

The found RG transformation will approach the trivial fixed point for any finite initial value of β, as equation (4.15) implies that $\beta' < \beta$. It is well-known that the 1D Ising model does not have a critical point as another candidate for a fixed point. Nevertheless, there is another fixed point, the so-called zero-temperature fixed point where $\beta \to \infty$ and $T = 0$. It describes a system where all spins are parallel. It is the only β value where the correlation length is infinite. The correlation length will however remain infinite under the RG transformation, meaning that the system remains at that point. The high-temperature and the low-temperature fixed points are the only two fixed points for the 1D Ising model.

4.3 The Cluster-inspired RG Transformation in 1D

We now want to apply the cluster-inspired RG transformation to the 1D Ising model. We had managed to define a cluster-inspired RG transformation through a blocking kernel that passes the concept of bonds and clusters on to the renormalized system in Section 3.3. While the cluster-inspired RG allows to perform the cluster algorithm on the renormalized system and obtain numerical results, we want to first take an analytical approach. The goal is to analytically find Boltzmann weights of renormalized configurations that might allow to find the renormalized Hamilton function. We do this in the bond representation, where spins have been integrated out. From there, we could then return to the cluster and the spin representations by using cluster logic.

We start with a single blocking step and define the renormalized nearest-neighbor bond variable as $b'_i = 1$ if activated and $b'_i = 0$ if deactivated. Figure 4.3 shows an examplaric 1D configuration on a short renormalized section that leads to an activated and a deactivated renormalized bond.

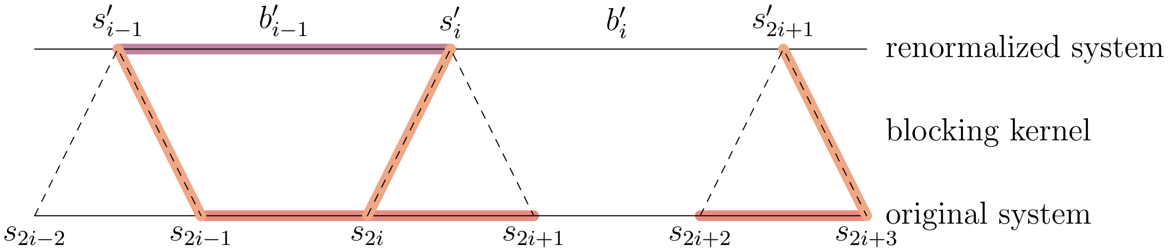
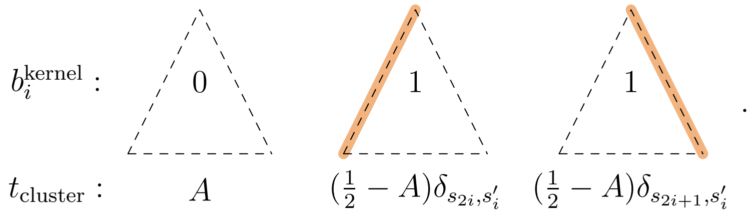


Figure 4.3: Illustration of exemplary section of a renormalized bond configuration. Red lines symbolize activated original bonds, orange lines activated kernel bonds and the purple line an activated renormalized bond, whereas all uncoloured lines are deactivated bonds. The renormalized bond between s'_{i-1} and s'_i is activated, because there is an end-to-end bond connection between them.

The cluster-inspired blocking kernel reduces from equation (3.11) to

$$t_{\text{cluster}}(s_{2i}, s_{2i+1}, s'_i, b_i^{\text{kernel}}) = \delta_{b_i^{\text{kernel}},0} A + \delta_{b_i^{\text{kernel}},1} \left((\delta_{s_{2i},s'_i} + \delta_{s_{2i+1},s'_i})(\frac{1}{2} - A) \right), \tag{4.16}$$

which may be illustrated as


b_i^{kernel} :
0
1
1
t_{cluster} :
A
$(\frac{1}{2} - A)\delta_{s_{2i},s'_i}$
$(\frac{1}{2} - A)\delta_{s_{2i+1},s'_i}$
.

(4.17)

4.4 Renormalized Transfer Matrices

We want to further formalize the concept of renormalized bonds in order to develop a method to specify Boltzmann weights of renormalized configurations. We start by analyzing a single renormalized link displayed in Figure 4.4. A renormalized bond will be activated or deactivated

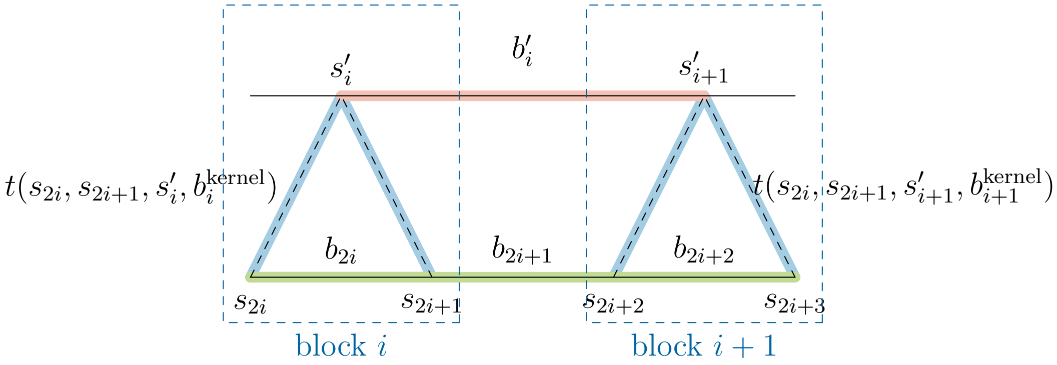


Figure 4.4: Schematic illustration of a renormalized section with two neighboring blocks, consisting of original bonds, kernel bonds and a renormalized bond.

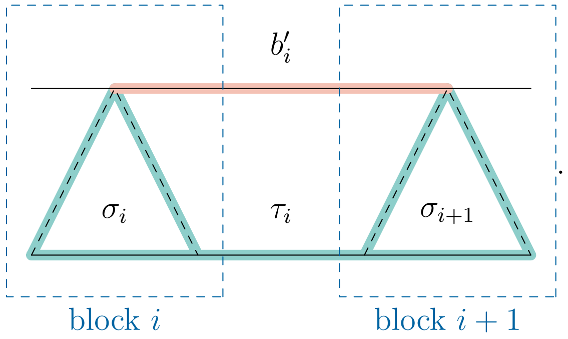
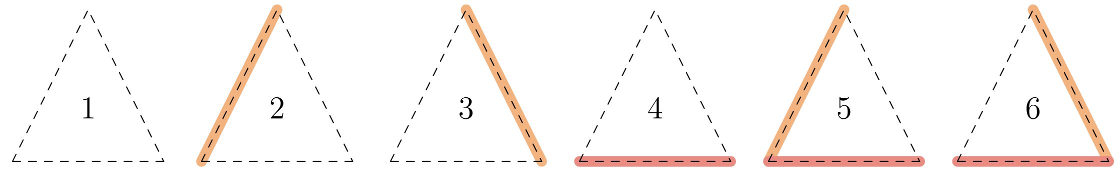


Figure 4.5: Schematic illustration of renormalized section with newly defined variables σ_i and τ_i that have replaced the original and the kernel bonds.

depending on the local link configuration. But how can its Boltzmann weights be described? The idea, that came up during a discussion, was to unify the original and the kernel bonds within a block and replace them by a newly defined single variable. This way, the renormalized system can then be described again as a 1D system, which allows for a transfer matrix approach. A potential matrix product should in this case not describe a partition function but a sum of original and kernel configurations that contribute to a given renormalized configuration. The idea was further elaborated and we chose to define two new variable types, one related to a block, that we call *block bond* σ_i, and the other one connecting the blocks, that we call *in-between bond* τ_i. These two variables will allow us to also integrate factors of 2 that arise from the freedom of cluster flipping when the spins are integrated out, in their definition. They are schematically displayed in Figure 4.5. Block bonds σ_i are constructed from original and kernel bonds, whereas in-between bonds τ_i are constructed only from original bonds. Block bonds σ_i

can take a total of 6 states with according weights $w(\sigma_i)$

σ_i :



$w(\sigma_i)$: $4w_0A$ $\quad 2w_0(\frac{1}{2}-A)$ $\quad 2w_0(\frac{1}{2}-A)$ $\quad 2w_1A$ $\quad w_1(\frac{1}{2}-A)$ $\quad w_1(\frac{1}{2}-A)$, (4.18)

where $w_1 = e^{\beta} - e^{-\beta}$ and $w_0 = e^{-\beta}$ were introduced as weights related to activating or deactivating a bond in the original system in Section 2.1.4. In-between bonds τ_i can take two values with according weights $w(\tau_i)$:

τ_i:

$w(\tau_i)$: $\quad w_1 \quad 2w_0$. (4.19)

In all these weights $w(\sigma_i)$ and $w(\tau_i)$, spins have already been integrated out and factors of 2 that arise when a missing bond causes an additional cluster to form, have been included. This applies when either an original or a renormalized spin remains unbonded.

Having introduced the new variables σ_i and τ_i, we can now indeed regard the renormalized system as a 1D chain again, as displayed in Fig 4.6.

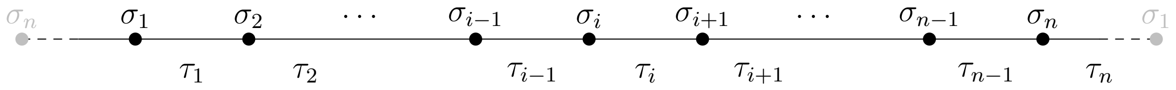


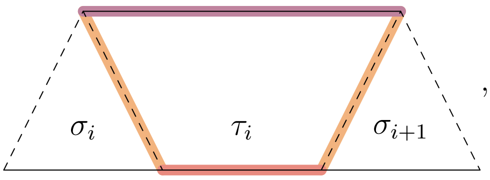
Figure 4.6: 1D renormalized Ising system with n spins and periodic boundary conditions. The original bonds and the kernel bonds have been united and replaced by block bonds σ_i and in-between bonds τ_i.

We now treat a renormalized configuration as a chain of activated and deactivated renormalized bonds, that are defined via the values of the block bonds σ_i and in-between bonds τ_i. To start, we return to the discussion on a single link, displayed in Figure 4.5 and ask ourselves which combinations of σ_i, τ_i and σ_{i+1} will form an activated renormalized bond $b'_i = 1$.

It is clear that in order to set a renormalized bond, τ_i needs to be activated, otherwise the bond will be deactivated in any case. The states of σ_i and σ_{i+1} that yield an activated bond are more involved. Assuming τ_i is activated, we need to check for each combination of states whether s'_i and s'_{i+1} are bonded by an end-to-end connection of original and kernel bonds. For these combinations, we can define a matrix that decides whether the renormalized bond is activated or deactivated. Its rows represent the states of σ_i and its columns those of σ_{i+1}, such that each entry of this matrix represents a combination of the two. We call this matrix W and its elements $W_{\sigma_i,\sigma_{i+1}}$. We fill the matrix with the Boolean values 1 for an activated and 0 for a deactivated renormalized bond, such that

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{4.20}$$

For example, $W_{3,2}$ relates to the local configuration



, (4.21)

and is set to 1 because the combination of $\sigma_i = 3$, $\sigma_{i+1} = 2$ induces an end-to-end bond connection between s'_i and s'_{i+1} in case of an activated τ_i, and therefore activates the renormalized bond.

We now combine the matrix W with the weights related to the three variables σ_i, σ_{i+1} and τ_i to construct a matrix which contains the weights that are related to activating the renormalized bond and call this matrix W_{bond}. As the variables σ_i, σ_{i+1} are each part of two renormalized links and we must only contribute their weights once, we include only the square roots of their weights. When we later multiply matrices, the other square root will be contributed from the neighboring link. The matrix W_{bond} thus is defined as

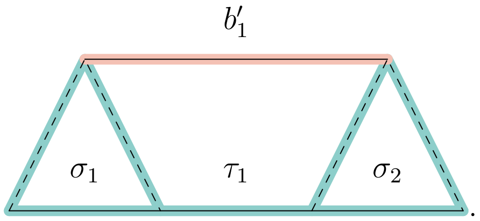
$$(W_{\text{bond}})_{\sigma_i,\sigma_{i+1}} = \sqrt{w(\sigma_i)}\sqrt{w(\sigma_{i+1})}w_1 W_{\sigma_i,\sigma_{i+1}}. \tag{4.22}$$

We now know that all combinations of values of σ_i, σ_{i+1} and τ_i that are not included in W_{bond} will result in a deactivated renormalized bond $b'_i = 0$. We keep the square root ansatz in mind and formulate an according matrix $W_{\text{no-bond}}$ that contains all these combinations. It is

$$(W_{\text{no-bond}})_{\sigma_i,\sigma_{i+1}} = \sqrt{w(\sigma_i)}\sqrt{w(\sigma_{i+1})}(2w_0 + w_1(1 - W_{\sigma_i,\sigma_{i+1}})). \tag{4.23}$$

W_{bond} and $W_{\text{no-bond}}$ are called *renormalized transfer matrices.* One can now multiply these matrices according to the given renormalized configuration consisting of activated and deactivated bonds, and will obtain the weight of the renormalized configuration as the summation of all the contributing weights. The boundary conditions therefore have to be considered. In case of periodic boundary conditions, one must take the trace of the whole matrix product. In case of open boundary conditions, the summation over all elements of the matrix product must be taken and the missing square root factors at the ends need to be contributed.

Let us illustrate the construction of the renormalized weights with some examples. We start with the smallest non-trivial renormalized system consisting of one link with open boundary conditions, which is



(4.24)

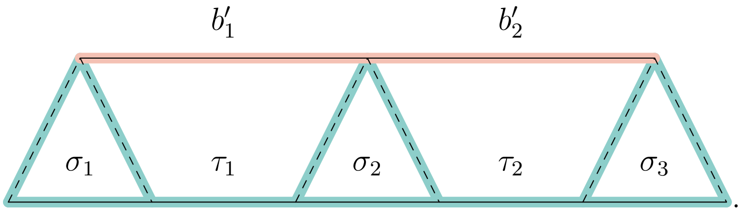
The weight of a renormalized configuration consisting of a single activated bond then is

$$\sqrt{w(\sigma_1)}\sqrt{w(\sigma_2)}\sum_{\sigma_1,\sigma_2}(W_{\text{bond}})_{\sigma_1,\sigma_2}, \tag{4.25}$$

where that of a configuration with a single deactivated bond is

$$\sqrt{w(\sigma_1)}\sqrt{w(\sigma_2)}\sum_{\sigma_1,\sigma_2}(W_{\text{no-bond}})_{\sigma_1,\sigma_2}. \tag{4.26}$$

The benefits of this procedure are not yet visible on such a small system. Let us consider a slightly larger system that contains two renormalized links:

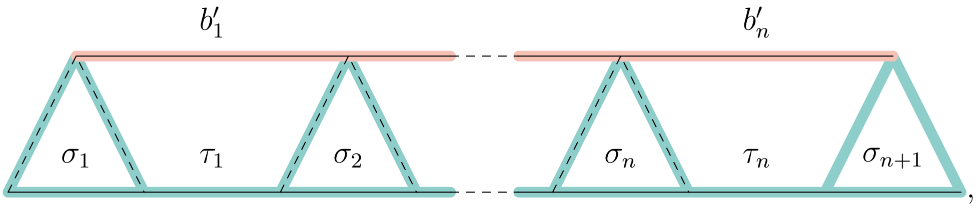


(4.27)

The weight of a renormalized bond configuration $[b'_1 = 1, b'_2 = 0]$ is

$$\sum_{\sigma_1,\sigma_3}\sqrt{w(\sigma_1)}\sqrt{w(\sigma_3)}(W_{\text{bond}}\cdot W_{\text{no-bond}})_{\sigma_1,\sigma_3}. \tag{4.28}$$

More generally, for a system of n renormalized links with open boundary conditions



(4.29)

the weight of an examplaric renormalized configuration $[1, 0, \ldots, 1, 0]$ is

$$\sum_{\sigma_1,\sigma_{n+1}}\sqrt{w(\sigma_1)}\sqrt{w(\sigma_{n+1})}(W_{\text{bond}}\cdot W_{\text{no-bond}}\cdot\cdots\cdot W_{\text{bond}}\cdot W_{\text{no-bond}})_{\sigma_1,\sigma_{n+1}}, \tag{4.30}$$

and the weight of the same configuration with periodic boundary conditions is

$$\mathrm{Tr}(W_{\text{bond}} \cdot W_{\text{no-bond}} \cdot \dots \cdot W_{\text{bond}} \cdot W_{\text{no-bond}}). \tag{4.31}$$

Thus, we have managed to describe the weight of any renormalized configuration, that consists of activated and deactivated nearest-neighbor bonds, by the renormalized transfer matrices.

4.5 Superbonds and True-No-Bonds

One may have noticed a flaw in the so-far construction of renormalized bonds and their weights.

In the bond representation of the original Ising model, bonds were set between neighboring spins only. However, in the general construction of cluster-inspired RG, we saw that bonds could connect any of the two renormalized spins, even if they were not nearest neighbors. This also holds for the 1D case. End-to-end bond connections between distant renormalized spins may skip intermediate renormalized spins via the original system, as displayed in Figure 4.7. To partition the renormalized system into clusters, describing it only with nearest-neighbor bonds is therefore too simplistic. Instead, we must also consider bonds between more distant spins, that we call *superbonds*.

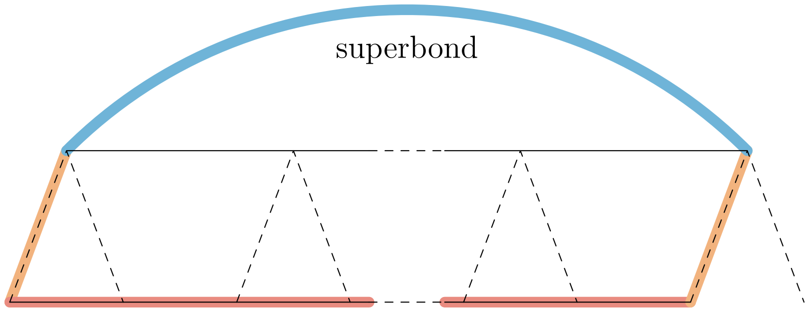


Figure 4.7: Exemplary local configuration yielding a superbond.

One may ask, how the configurations including superbonds could be missed in the so-far construction of renormalized weights. In fact, they were not missed but instead, they were unknowingly included in the matrix $W_{\text{no-bond}}$. $W_{\text{no-bond}}$ was just defined as a trash can for each local configuration that does not activate the nearest-neighbor bond. And that is also the case for a superbond.

By construction, we can identify the following rule for superbonds:

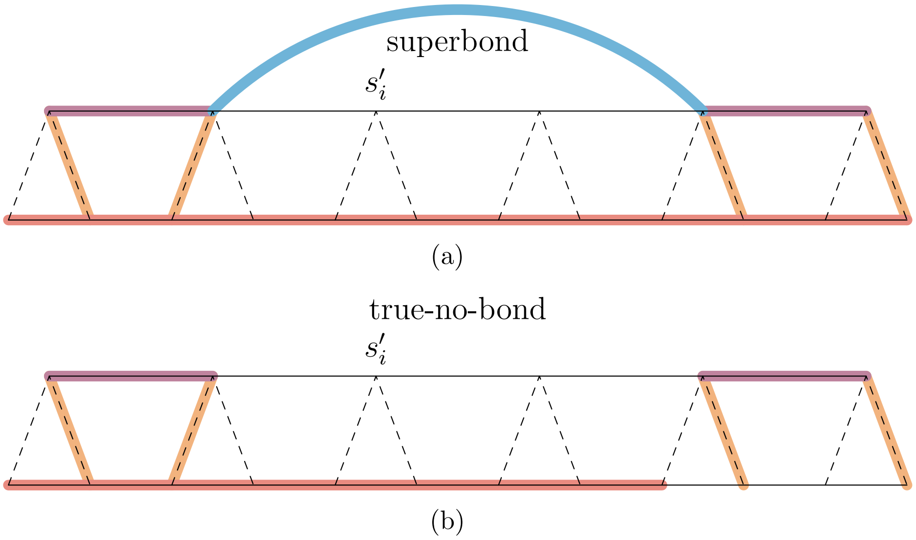
> Superbonds and bonds do not superimpose each other in any way. No shorter superbond or bond appears within the scope of a larger superbond and superbonds do not cross each other.

The rule arises when one considers a local configuration that yields a superbond more carefully. Let us assume a superbond of a given length n, meaning that the two bonded spins are n renormalized sites away from each other. The end-to-end connection that exists between the two spins cannot rise into the renormalized system within the superbond, but only at its ends, as displayed in Figure 4.7. If it did, an additional intermediate spin would be added to the cluster and the superbond would split into two shorter superbonds. This also implies that all renormalized spins in the interior of a superbond cannot be bonded to any other spins, but form their own single-spin clusters. This further rules out the crossing of superbonds and generally any superposition of bonds and superbonds.

We can now define a matrix related to the weight of a superbond of length n that is in fact very similar to the bond matrix. The only differences are that instead of a single τ_i, a series of n τ-variables need to be activated, and that all $(n-1)$ σ-variables within the superbond need to be in state 4. The matrix $W_{\mathrm{superbond(n)}}$ thus is

$$(W_{\mathrm{superbond(n)}})_{\sigma_i,\sigma_{i+n}} = \sqrt{w(\sigma_i)}\sqrt{w(\sigma_{i+n})}w_1^n(2w_1A)^{n-1}W_{\sigma_i,\sigma_{i+n}}. \tag{4.32}$$

The other object that needs to be looked at more closely is that of a series of no-bonds. While a single no-bond, embedded between bonds, cannot include the weights of any superbonds, we now know that a series of no-bonds does. In fact, its weight includes the weights of all combinations of superbonds that may fit within the length of the series of no-bonds, as we defined the no-bond as a trash can, including all local configurations' weights except for those of a nearest-neighbor bond. We would like to define an object that describes a series of no-bonds without the inclusion of superbonds, such that clusters become well-defined. When a spin forms its own cluster, it is not possible to tell by the local configuration, whether the spin is, or is not, in the interior of a superbond. An example of such a situation is given in Figure 4.8, showing two configurations, one with a superbond, and one without. The case of deactivated bonds that are not in the interior of a superbond must therefore be treated in a non-local way. Let us therefore consider a spin that forms the end of a cluster. It is bonded to one side by a bond or a superbond and unbonded to the other side. In this case, because bonds and superbonds cannot superimpose, we know that the deactivated bond on its other side is not within a superbond. To find out whether a deactivated bond is part of superbond or not, it is necessary to know the configuration on both sides of the spin until the end of a bond or superbond is reached and the resulting series of deactivated bonds must therefore be treated as a single object. We call it *true-no-bond* of length n.



(a)

(b)

Figure 4.8: Two similar configurations on the multigrid. The local surroundings within nearest-neighbor distance of s'_i are equivalent, however, when considering the global configuration, it is in the interior of a superbond in configuration (a), where it is part of a true-no-bond in (b).

We want to also define a matrix related to the weights of a true-no-bond of length n. However, this is much harder than for the superbonds. We can start building it by taking series of no-bond matrices. We know that the true-no-bond matrix is a part of it, but the series of no-bond matrices also includes each variation of superbonds and shorter series of no-bonds that fit within its length. We must therefore subtract all these contributions. The number variations increases exponentially with the length n. An algorithm was implemented in Mathematica to construct the true-no-bond matrices. For short lengths, it works well, but as the complexity increases exponentially, it is limited to lengths of up to $n \approx 8$. In a very general notation, $W_{\text{true-no-bond}(n)}$ can formulated as

$$(W_{\text{true-no-bond}(n)})_{\sigma_i,\sigma_{i+n}} = W^n_{\text{no-bond}} - (\text{all variations consisting of superbonds} \\ \text{and shorter series of no-bonds}). \tag{4.33}$$

To be more concrete, a specific example $W_{\text{true-no-bond}(4)}$ is specified as

$$\begin{aligned}(W_{\text{true-no-bond}(4)})_{\sigma_i,\sigma_{i+n}} = {} & W_{\text{no-bond}} \cdot W_{\text{no-bond}} \cdot W_{\text{no-bond}} \cdot W_{\text{no-bond}} \\ & - W_{\text{no-bond}} \cdot W_{\text{no-bond}} \cdot W_{\text{superbond}(2)} \\ & - W_{\text{no-bond}} \cdot W_{\text{superbond}(2)} \cdot W_{\text{no-bond}} \\ & - W_{\text{superbond}(2)} \cdot W_{\text{no-bond}} \cdot W_{\text{no-bond}} \\ & - W_{\text{no-bond}} \cdot W_{\text{superbond}(3)} \\ & - W_{\text{superbond}(3)} \cdot W_{\text{no-bond}} \\ & - W_{\text{superbond}(2)} \cdot W_{\text{superbond}(2)} \\ & - W_{\text{superbond}(4)}. \end{aligned} \tag{4.34}$$

Objects can now be summarized. Bonds can be seen as superbonds of length 1 and no-bonds can be seen as true-no-bonds of length 1. This reduces the objects required to define the renormalized bond configuration to superbonds and true-no-bonds. With these, any renormalized configuration can be described. This finishes the formalization of the first step of cluster-inspired RG in the 1D. It allows to find the Boltzmann weight of any renormalized bond configuration. Figure 4.9 shows an exemplary configuration.

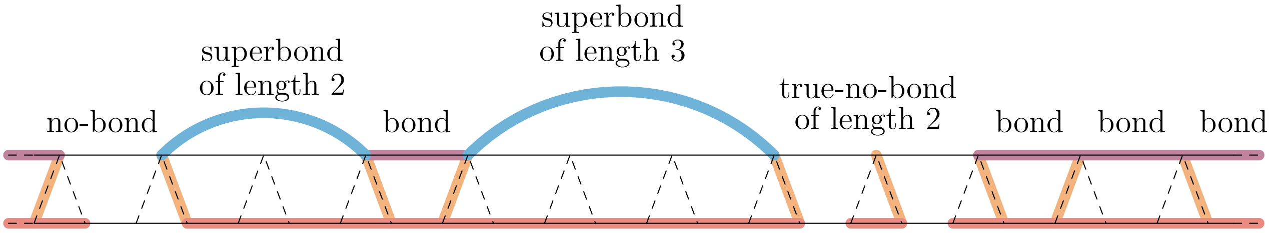


Figure 4.9: Exemplary configuration including bonds, superbonds and true-no-bonds of different lengths.

The Boltzmann weight of the exemplary configuration in Figure 4.9 in case of non periodic boundary conditions can be written as

$$\mathrm{Tr}\big(W_{\text{no-bond}} \cdot W_{\text{superbond(2)}} \cdot W_{\text{bond}} \cdot W_{\text{superbond(3)}} \cdot W_{\text{true-no-bond(2)}} \cdot W_{\text{bond}} \cdot W_{\text{bond}} \cdot W_{\text{bond}}\big). \tag{4.35}$$

4.6 Matrix Properties

We have found a way write down the Boltzmann weight of any renormalized bond configuration after a first blocking step, however, only as a matrix product and including complicated true-no-bonds. If we now want to find the configurations' probabilities, we must find the partition function. And if we want to proceed to iteration of the blocking transformation, we must simplify the results. We try to do so by the means of linear algebra.

Properties of the matrices W_{bond}, $W_{\text{superbond(n)}}$, W_{bond} and $W_{\text{true-no-bond(n)}}$ have been found. Any statements that involve $W_{\text{true-no-bond(n)}}$ could not be verified for general n, due to the complexity that lies within deducing the variations of superbonds. Instead, they were checked iteratively increasing the length of true-no-bonds until the runtime became excessive. The properties, found with Mathematica, are the following:

- W_{bond} is diagonalizable with a single non-zero eigenvalue

$$X := 2w_1^2(\frac{1}{2} - A). \tag{4.36}$$

- $W_{\text{no-bond}}$ is also diagonalizable, however not simultanously with with W_{bond}. It has two very complicated non-zero eigenvalues Y and Z, which are

$$\begin{aligned} Y :=& 4(\tfrac{1}{2}-A)w_0(w_0+w_1)+A(2w_0+w_1)^2 \\ & -\sqrt{-4(\tfrac{1}{2}-A)w_1(2w_0+w_1)((\tfrac{1}{2}-A)w_0^2-Aw_1(2w_0+w_1))+(4(\tfrac{1}{2}-A)w_0(w_0+w_1)+A(2w_0+w_1)^2)^2}, \\ Z :=& 4(\tfrac{1}{2}-A)w_0(w_0+w_1)+A(2w_0+w_1)^2 \\ & +\sqrt{-4(\tfrac{1}{2}-A)w_1(2w_0+w_1)((\tfrac{1}{2}-A)w_0^2-Aw_1(2w_0+w_1))+(4(\tfrac{1}{2}-A)w_0(w_0+w_1)+A(2w_0+w_1)^2)^2}. \end{aligned} \tag{4.37}$$

- All $W_{\text{superbond(n)}}$ are all simultanously diagonalizable, also with a single eigenvalue

$$X_n := 2^n A^{n-1}(\frac{1}{2} - A)w_1^{2n}, \tag{4.38}$$

  where $X_1 \equiv X$. Indeed, $W_{\text{bond}} = W_{\text{superbond(1)}}$.

- $W_{\text{true-no-bond(n)}}$ are diagonalizable, however, it remains unclear, whether they are simultaneously diagonalizable with any other of the matrices. The eigenvalues get more and more complicated and Mathematica quickly reaches its limits with increasing n. The eigenvalues cannot be split up to local factors, that could then be taken to the power of number of links.

This is so far not very satisfactory. At least we know that superbonds (that include also bonds) are simultanously diagonalizable and therefore commute. But they are not simultanously diagonalizable with $W_{\text{no-bond}}$ and $W_{\text{true-no-bond(n)}}$. There is, however, another significant property of $W_{\text{true-no-bond(n)}}$, that appears when one embeds a true-no-bond in between bonds and / or superbonds. And this is by definition always the case. One then finds:

- Any objects $W_{\text{bond}} \cdot W_{\text{true-no-bond(n)}} \cdot W_{\text{bond}}$ or more generally $W_{\text{superbond(m)}} \cdot W_{\text{true-no-bond(n)}} \cdot W_{\text{superbond(l)}}$ commute and are simultanously diagonalizable with $W_{\text{superbond(n)}}$. In other words, all superbonds and true-no-bonds commute, as long as no true-no-bonds are split up or merged.

- $W_{\text{superbond(m)}} \cdot W_{\text{true-no-bond(n)}} \cdot W_{\text{superbond(l)}}$ have only a single non-zero eigenvalue related to the same eigenvector the non-zero eigenvalue of $W_{\text{superbond(n)}}$. However, this eigenvalue gets more complicated with increasing length n and again cannot be split up into local factors that could be taken to the power of n.

All these insights bring the analytical approach to 1D cluster-inspired RG closer to its solution. However, to actually solve the partition function seems unrealistic regarding the high complexity and the loss of ultra-local behavior in objects as superbonds and the resulting true-no-bonds.

4.7 Special Values of A

The construction of analytical cluster-inspired RG on the 1D Ising model was done for general values of the parameter A up to this point. However, complexity coming from superbonds and true-no-bonds diminish the hopes to find the general analytical solution. Therefore, it may be useful to look at certain special values of the parameter A instead.

$A = 0$

Obviously, a special value for A is 0. It restricts the kernel bonds to always be activated and therefore the block bonds σ_i to only take four out of its six states. The states 1 and 4, introduced in equation (4.18), have weight 0. It avoids the existence of superbonds. Therefore, true-no-bonds can be split up locally as they simply correspond to a series of no-bonds. With the diagonalization of W_{bond} and $W_{\text{no-bond}}$, this allows to perform a second blocking step, described in section 4.9.

$A = \exp(-4\beta)$

The value $A = \exp(-4\beta)$ was found as a solution to the requirement of commutation between W_{bond} and $W_{\text{no-bond}}$. It allows to determine the Boltzmann weight of a renormalized configuration consisting of bonds and no-bonds as a function of only the numbers of bonds and no-bonds. This is a strong simplification, however, it is useless as the clusters are not specified only from bonds and no-bonds, but superbonds and true-no-bonds must be specified.

$A = \frac{1}{2}(1 - \sqrt{1 - \exp(-4\beta)})$

The value $A = \frac{1}{2}(1 - \sqrt{1 - \exp(-4\beta)})$ was the value used in the Ising-interaction-preserving RG transformation performed in Section 4.2. The parameter A was first introduced as part of the general 1D blocking kernel acting in the spin representation, that was constructed in Section 4.1. A was later identified again as the parameter that appeared in the cluster-inspired RG transformation. Therefore, when A is set to the value used in Ising-interaction-preserving RG, cluster-inspired RG is expected to also preserve the Ising interactions. However, no pattern could be found that confirms this. How this manifests in the behavior of the renormalized bonds, superbonds, and true-no-bonds appearing in our model remains an unsolved riddle of this project.

4.8 Simplification for $A = 0$

As we could not solve the first blocking step of the cluster-inspired RG transformation on the 1D Ising model generally, we want to try it for the case $A = 0$. All we then need to deal with are bonds and no-bonds related to the matrices W_{bond} and $W_{\text{no-bond}}$. The two matrices are both diagonalizable and their eigenvalues for general A were introduced in Section 4.6.

W_{bond} has a single eigenvalue X that when A is set to zero simplifies to

$$X = w_1^2. \tag{4.39}$$

$W_{\text{no-bond}}$ has two eigenvalues Y and Z, which when A is set to zero simplify to

$$\begin{aligned} Y :=& 2w_0(w_0 + w_1) - \sqrt{2w_0(w_0 + w_1) - w_1 w_0^2 (2w_0 + w_1)}, \\ Z :=& 2w_0(w_0 + w_1) + \sqrt{2w_0(w_0 + w_1) - w_1 w_0^2 (2w_0 + w_1)}. \end{aligned} \tag{4.40}$$

W_{bond} and $W_{\text{no-bond}}$ are not simultanously, but we can introduce transformation matrices U and V, as well as a combined transformation matrix T, which we call *transition matrix*, such that

$$W_{\text{bond}} = U \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & X \end{pmatrix} U^{-1}, \quad W_{\text{no-bond}} = V \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & Y & 0 \\ 0 & \cdots & 0 & Z \end{pmatrix} V^{-1}, \quad T = U^{-1}V. \tag{4.41}$$

The small number of non-zero eigenvalues allows further simplify a configuration's weight. Let us transform the weight of an exemplary configuration that contains one series of bonds and one series of no-bonds into the W_{bond} eigenbasis, such as

$$\begin{aligned} &\operatorname{Tr}\left(\underbrace{W_{\text{bond}} \ldots W_{\text{bond}}}_{n} \underbrace{W_{\text{no-bond}} \ldots W_{\text{no-bond}}}_{m} \underbrace{W_{\text{bond}} \ldots W_{\text{bond}}}_{l} \right) \\ &= \operatorname{Tr}\left(U \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & X^n \end{pmatrix} U^{-1}V \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & Y^m & 0 \\ 0 & \cdots & 0 & Z^m \end{pmatrix} V^{-1}U \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & X^l \end{pmatrix} U^{-1} \right) \\ &= \operatorname{Tr}\left(\left(\begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & X^n \end{pmatrix} T \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & Y^m & 0 \\ 0 & \cdots & 0 & Z^m \end{pmatrix} T^{-1} \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & X^l \end{pmatrix} \right) \right). \end{aligned} \tag{4.42}$$

Generalising this result, W_{bond} and $W_{\text{no-bond}}$ can both be replaced by their diagonals if one adds the transition matrix T at transitions from bond to no-bond and its inverse T^{-1} at transitions from no-bond to bond. Let us refer to a series of no-bonds as a *multi-no-bond* from now on. We assign the transition matrix and its inverse to the diagonalized multi-no-bond matrix, such that a multi-no-bond in a configuration corresponds to a matrix term

$$T \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & Y^m & 0 \\ 0 & \cdots & 0 & Z^m \end{pmatrix} T^{-1}. \tag{4.43}$$

in its weight, appearing inside the trace in between diagonalized W_{bond}, whose only non-zero element X is the sixth element on the diagonal. Thus the only contributing element of the matrix term is

$$\left(T \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & 0 & Y^m & 0 \\ 0 & \cdots & 0 & Z^m \end{pmatrix} T^{-1} \right)_{6,6} = T_{6,5} Y^m T^{-1}_{5,6} + T_{6,6} Z^m T^{-1}_{6,6}. \tag{4.44}$$

Therefore the Boltzmann weight can be written as

$$\begin{aligned} &\mathrm{Tr}\left(\underbrace{W_{\text{bond}} \ldots W_{\text{bond}}}_{n} \underbrace{W_{\text{no-bond}} \ldots W_{\text{no-bond}}}_{m} \underbrace{W_{\text{bond}} \ldots W_{\text{bond}}}_{l} \right) \\ &= X^{n+l} (T_{6,5} Y^m T^{-1}_{5,6} + T_{6,6} Z^m T^{-1}_{6,6}), \end{aligned} \tag{4.45}$$

where $n + l$ is the total number of bonds and m is the length of the one multi-no-bond. A general Boltzmann weight that contains n bonds and m multi-no-bonds of lengths $l_1, \ldots, l_m$ is

$$X^n w_{\text{multi-no-bond}}(l_1) \ldots w_{\text{multi-no-bond}}(l_m), \tag{4.46}$$

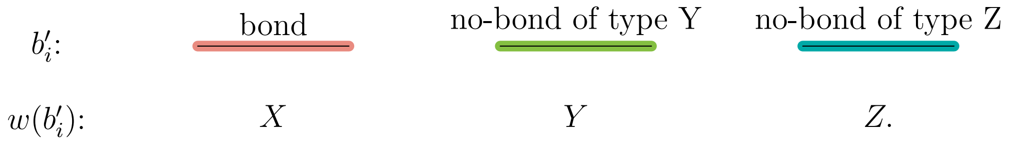
where

$$w_{\text{multi-no-bond}}(n) = T_{6,5} Y^n T^{-1}_{5,6} + T_{6,6} Z^n T^{-1}_{6,6}. \tag{4.47}$$

This is a strong simplification of the Boltzmann weights of renormalized configurations that brings the approach another step closer to the solution. Even though, this was not done here, obtaining a solution for the partition function in the case of $A = 0$ does not seem to be completely unrealistic. One might try to formalize the summation over all bond and nobond combinations using combinatorics. We were however not particularly interested solving only the first blocking step, but instead we tried to continue with the second step instead.

4.9 Second Blocking Step

In order to perform a second renormalization step, we work with the three eigenvalues X, Y and Z that have been found. We treat these as weights of three new renormalized bond states. We therefore redefine the renormalized bond variable b'_i and its weights as



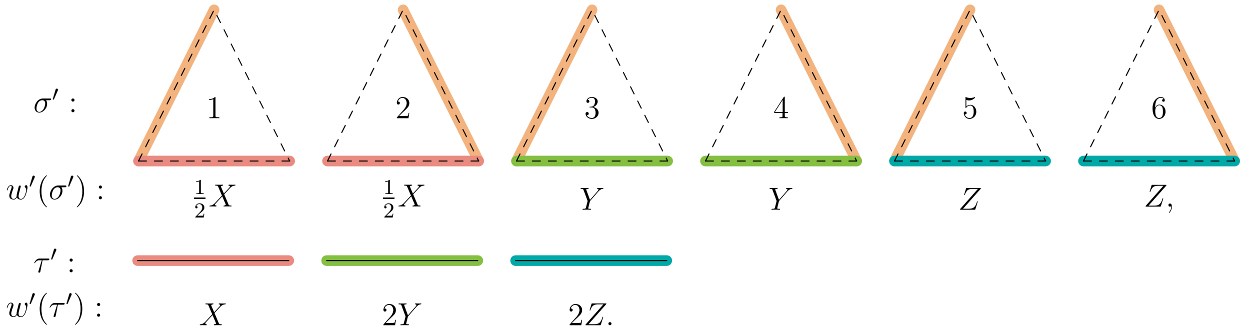
(4.48)

Even though we do not know whether Y and Z can physically be attributed to distinct states, the seperation works mathematically. By it, the Boltzmann weight can be brought to a local form again, that consists of the weights of the renormalized bonds and their transitions, which are the particular entries of the transition matrix T that appear in equation (4.47). We relabel them by type of transition to

$$t_{XY} = T_{6,5}, \tag{4.49}$$
$$t_{YX} = T^{-1}_{5,6}, \tag{4.50}$$
$$t_{XZ} = T_{6,6}, \tag{4.51}$$
$$t_{ZX} = T^{-1}_{6,6}, \tag{4.52}$$

where the two indices relate to the bond values to the left and the right of the transition.

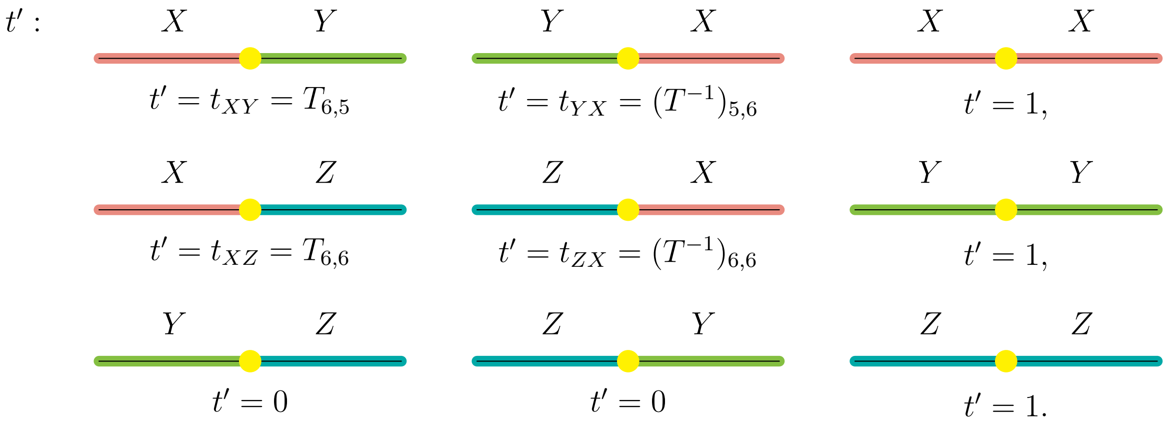
As we already did in the first blocking step, we again define block bonds σ'_i and in-between bonds τ'_i, that take the values and weights



(4.53)

New are the transitions that we include in an object t', which we mark as yellow dot. Between equal values of b'_i such as X and X, no transition occurs, therefore t' is set to 1. Y and Z are two different eigenvalues of the same matrix and appear only separately in the weight of a multi-no-bond, found in equation (4.47). An entire multi-no-bond must therefore decide on type Y or type Z in any configuration. This is achieved by setting the transitions between Y

and Z to 0. Thus, t' takes the values



(4.54)

In the first blocking step, defining transitions was not necessary as these would all have been equal to 1. This is because the weight could be described without the use of matrices and no change of basis was necessary.

With these building blocks, we can construct the bond and no-bond matrices of the second renormalized layer. Schematically, the construction resembles that of the first renormalized layer with the difference of the additional t'. In illustration on it is found in Figure 4.10.

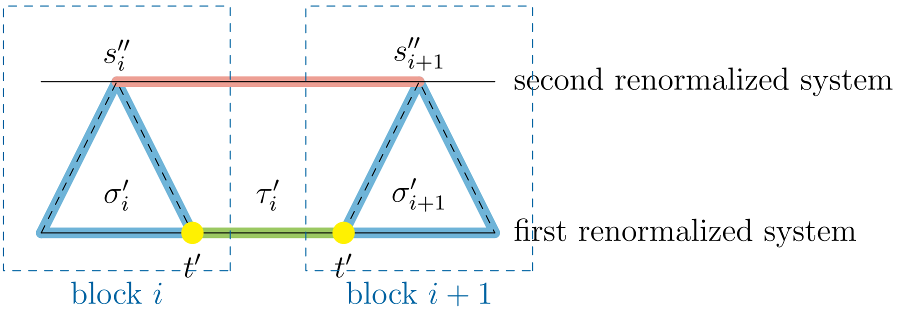


Figure 4.10: Schematic illustration of the second renormalization step on a single link section with relevant variables.

We now construct a W^{second} matrix analogously to the matrix W in the first blocking step, which decides whether a bond is activated for a given combination of values σ'_i and σ'_{i+1}, as long as τ'_i is activated. It also contains the transitions t' that did not exist in the first blocking step. W^{second} is

$$W^{\text{second}} = \begin{pmatrix} 1 & 1 & t_{XY} & 0 & t_{XZ} & 0 \\ 1 & 1 & t_{XY} & 0 & t_{XZ} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ t_{YX} & t_{YX} & t_{YX}t_{XY} & 0 & t_{YX}t_{XZ} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ t_{ZX} & t_{ZX} & t_{ZX}t_{XY} & 0 & t_{ZX}t_{XZ} & 0 \end{pmatrix}. \tag{4.55}$$

Including the square roots of the weights of σ'_i and σ'_{i+1} and an X for the activated τ'_i, the second renormalized bond matrix $W^{\text{second}}_{\text{bond}}$ follows from it as

$$W^{\text{second}}_{\text{bond}}{}_{(\sigma'_i\sigma'_{i+1})} = \sqrt{w'(\sigma'_i)}\sqrt{w'(\sigma'_{i+1})}XW^{\text{second}}_{(\sigma'_i\sigma'_{i+1})} \tag{4.56}$$

In order to construct $W^{\text{second}}_{\text{no-bond}}$, we need one more element, which was also trivial in the first blocking step. There, we simply inverted the Boolean values in the W matrix elementwise as $(1 - W_{\sigma_i,\sigma_{i+1}})$ to obtain the combinations that lead to a no-bond. Now, however, also the transition factors t' need to be included. We define the following matrices

$$\begin{aligned}
W^{\text{XFull}} &= \left(\begin{array}{c|c|c} 1 & t_{XY} & t_{XZ} \\ \hline t_{YX} & t_{YX}t_{XY} & t_{YX}t_{XZ} \\ \hline t_{ZX} & t_{ZX}t_{XY} & t_{ZX}t_{XZ} \end{array}\right), \\
W^{\text{YFull}} &= \left(\begin{array}{c|c|c} t_{XY}t_{YX} & t_{XY} & t_{XY}t_{YZ} \\ \hline t_{YX} & 1 & t_{YZ} \\ \hline t_{ZY}t_{YX} & t_{ZY} & t_{ZY}t_{YZ} \end{array}\right), \\
W^{\text{ZFull}} &= \left(\begin{array}{c|c|c} t_{XZ}t_{ZX} & t_{XZ}t_{ZY} & t_{XZ} \\ \hline t_{YZ}t_{ZX} & t_{YZ}t_{ZY} & t_{YZ} \\ \hline t_{ZX} & t_{ZY} & 1 \end{array}\right).
\end{aligned} \tag{4.57}$$

Each of these matrices stands for the transition factors appearing while having a given τ'_i value X, Y or Z. They are actually 6×6 matrices and each entry in equation (4.57) stands for a 2×2 block filled with identical elements. Each of the matrices entries still refers to a combination of the two variables σ'_i and σ'_{i+1}. The only entries out of these three matrices that lead to an activated renormalized bond, were the ones in W^{second} as part of W^{XFull}, and all other entries will lead to a deactivated renormalized bond.

$W^{\text{second}}_{\text{no-bond}}$ can be constructed by summing over all configurations except the ones leading to a bond, thus it is

$$W^{\text{second}}_{\text{no-bond}(\sigma'_i\sigma'_{i+1})} = \sqrt{w'(\sigma'_i)}\sqrt{w'(\sigma'_{i+1})}\Big(X\Big(W^{\text{XFull}}_{(\sigma'_i\sigma'_{i+1})} - W^{\text{second}}_{(\sigma'_i\sigma'_{i+1})}\Big) - 2YW^{\text{YFull}}_{(\sigma'_i\sigma'_{i+1})} - 2ZW^{\text{ZFull}}_{(\sigma'_i\sigma'_{i+1})}\Big). \tag{4.58}$$

From the two matrices $W^{\text{second}}_{\text{bond}}$ and $W^{\text{second}}_{\text{no-bond}}$, we can again construct the weight of any renormalized configuration, this time relating to second blocking step. An exemplary Boltzmann weight is

$$\text{Tr}\big(W^{\text{second}}_{\text{bond}}\dots W^{\text{second}}_{\text{bond}}W^{\text{second}}_{\text{no-bond}}\dots W^{\text{second}}_{\text{no-bond}}W^{\text{second}}_{\text{bond}}\dots W^{\text{second}}_{\text{bond}}\big). \tag{4.59}$$

We analyze the properties of $W^{\text{second}}_{\text{bond}}$ and $W^{\text{second}}_{\text{no-bond}}$. We try again to diagonalize them in hope for a repetitive pattern in their eigenvalues. However, Mathematica tells us that$W^{\text{second}}_{\text{bond}}$ is not diagonalizable.

We accept that our approach to the analytical solution of cluster-inspired RG has exceeded a reasonable level of complexity and decided at this point to move on to numerical methods.

5 2D Numerical Cluster-inspired RG

5.1 Cluster-inspired RG on the Triangular Lattice

5.1.1 Construction of the Blocking Kernel

In Chapter 3, we constructed the cluster-inspired RG transformation. In Chapter 4, we applied it to the 1D Ising model and tried to solve it analytically. We are now going to apply it to the 2D Ising model and work on it numerically. We proceed on the triangular lattice, where we chose triangular blocks with three original spins, as displayed in Figure 5.1. The structure of blocks in the lattice is displayed in Figure 5.2.

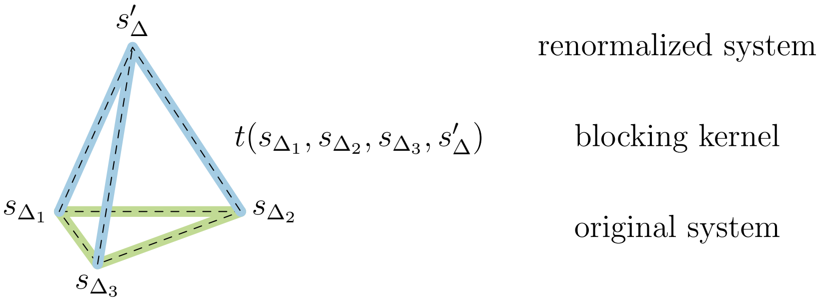


Figure 5.1: Illustration of a single block on the 2D triangular lattice.

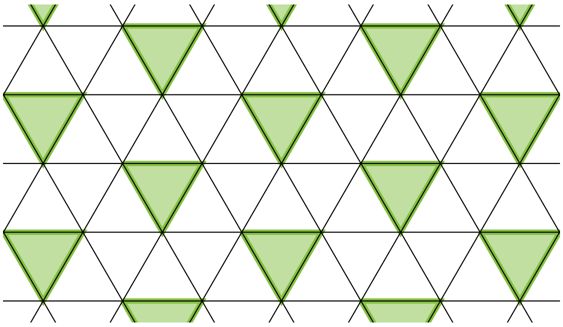


Figure 5.2: Illustration of the structure of 2D triangular blocks in the lattice.

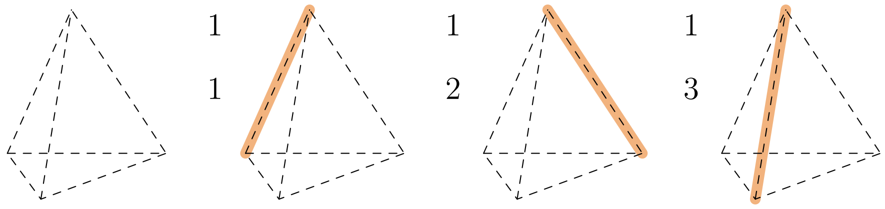
The cluster-inspired blocking kernel follows from its general form in equation (3.11) as

$$t_{\text{cluster}}(s_{\Delta_1}, s_{\Delta_2}, s_{\Delta_3}, s'_\Delta, b_\Delta^{\text{kernel}}) = \delta_{b_\Delta^{\text{kernel}},0} A + \delta_{b_\Delta^{\text{kernel}},1} \frac{1-2A}{3} (\delta_{s_{\Delta_1},s'_\Delta} + \delta_{s_{\Delta_2},s'_\Delta} + \delta_{s_{\Delta_3},s'_\Delta}), \quad (5.1)$$

or alternatively written as a funtion of $\tilde{b}^{\text{kernel}}_{\Delta}$

$$t_{\text{cluster}}(s_{\Delta_1}, s_{\Delta_2}, s_{\Delta_3}, s'_{\Delta}, \tilde{b}^{\text{kernel}}_{\Delta}) = \delta_{\tilde{b}^{\text{kernel}}_{\Delta},0} A + \sum_{i \in \Delta} \delta_{\tilde{b}^{\text{kernel}}_{\Delta},i} \delta_{s_i, s'_{\Delta}} \frac{1-2A}{3}. \tag{5.2}$$

The two versions can be illustrated as



$$\begin{array}{lcccc} b^{\text{kernel}}_{\Delta}: & 0 & 1 & 1 & 1 \\ \tilde{b}^{\text{kernel}}_{\Delta}: & 0 & 1 & 2 & 3 \\ t_{\text{cluster}}: & A & \frac{1-2A}{3}\delta_{s_{\Delta_1},s'_{\Delta}} & \frac{1-2A}{3}\delta_{s_{\Delta_2},s'_{\Delta}} & \frac{1-2A}{3}\delta_{s_{\Delta_3},s'_{\Delta}}. \end{array} \tag{5.3}$$

The iterated blocking kernel on the triangular lattice is

$$\begin{aligned} & t^{(n_r)}_{\text{cluster}}(s_{\Delta^{(n_r)}}, s'_{\Delta^{(n_r)}}, b^{\text{kernel}}_{\Delta^{(n_r)}}) \\ &= \delta_{b^{\text{kernel}}_{\Delta^{(n)}},0} \frac{1-(1-2A)^{n_r}}{2} + \delta_{b^{\text{kernel}}_{\Delta^{(n)}},1} \sum_{i \in \Delta^{(n_r)}} \delta_{s_i, s'_{\Delta^{(n_r)}}} \frac{(1-2A)^{n_r}}{3^{n_r}}. \end{aligned} \tag{5.4}$$

5.1.2 Symmetries and Block Sectors on the Triangular Lattice

We chose to proceed on the triangular lattice in the hope to later be able to parametrize a renormalized Hamilton function with fewer terms due to the larger number of symmetries of the triangular lattice, compared to the square lattice. The downside of the triangular lattice is a higher geometrical complexity that also manifests itself in its RG blocking transformations. The lattice symmetries of the square and the triangular lattices are

symmetries	square lattice	triangular lattice
rotation	90°	60°
translation	4 axes	6 axes
reflection	4 axes	6 axes

. (5.5)

The shapes of a finite, periodic lattice that obey these symmetries are the square for the square lattice and the hexagon for the triangular lattice, displayed in figure 5.3.

Blocks of the triangular lattice consist of three original spins. Unlike the square lattice, the triangular lattice leaves two options to choose the orientation of the blocks, displayed in Figure 5.4 . The triangular shaped blocks can be oriented to point one way or the other.

Renormalizing the triangular lattice also changes its orienation by 90°. When the original lattice was oriented such that it included horizontal lines of links, the renormalized lattice will instead include vertical lines of links. These two different orientations alternate in the RG transformation. They are are displayed in Figure 5.5.

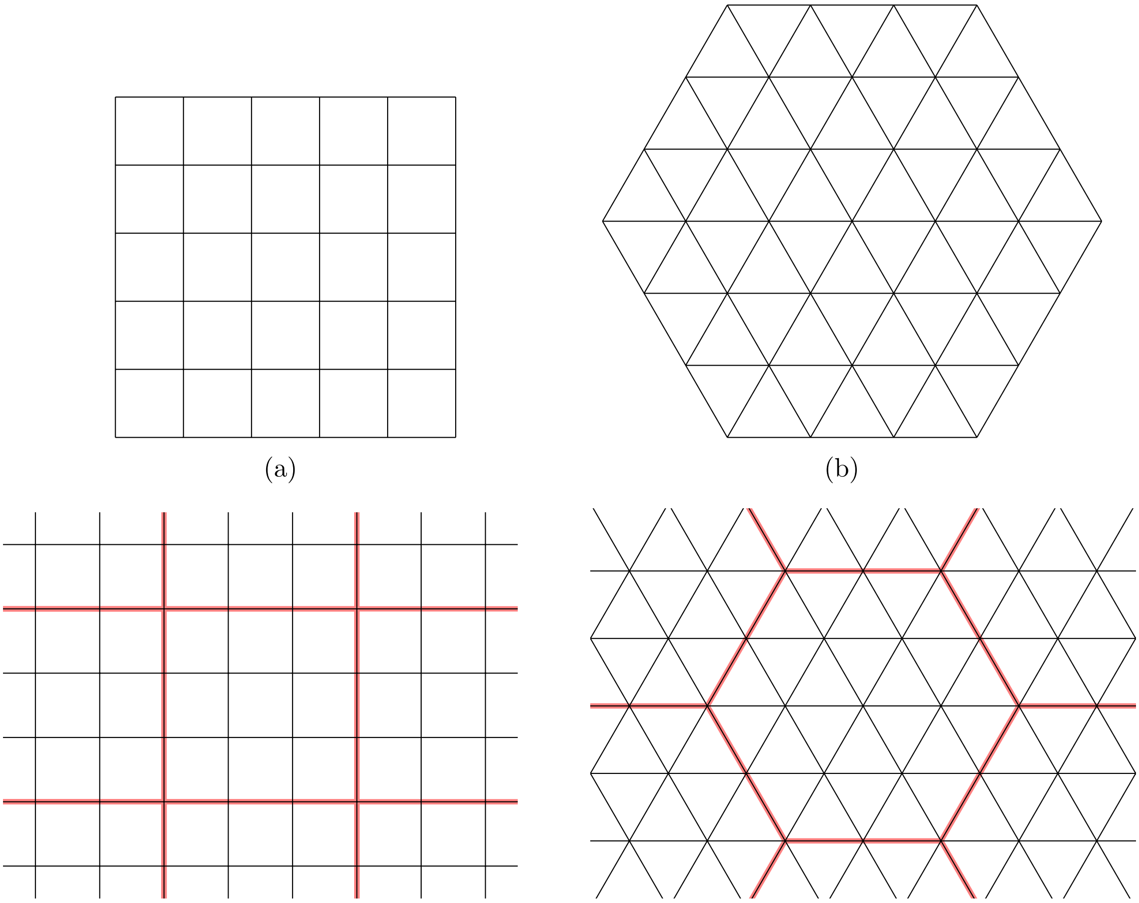


Figure 5.3: Finite lattices of shapes that obey the lattice symmetries. (a) shows a finite square shaped square lattice and (b) a finite hexagonally shaped triangular lattice. (*c*) and (*d*) show the embedding of these finite lattices in a larger lattice, a situation that occurs when periodic boundary conditions are applied, where the lattice repeats itself over and over again in each direction.

Because of the change in orientation, the block sectors happen to obtain a non-trivial shape that converges towards a fractal. The exact shape depends on the chosen sequence of block orientations. Figure 5.6 shows the shapes of the block sectors for two different sequences. The block sectors do not obey all lattice symmetries of the triangular lattice. While they do obey translational symmetry, they explicitly break the 60° rotational symmetry to a 120° rotational symmetry. This happens already in the first blocking step with the triangular shaped blocks. In the case of multiple blocking steps, also the reflectional symmetry is broken. This symmetry breaking puts doubts on the superiority of the triangular lattice over the square lattice.

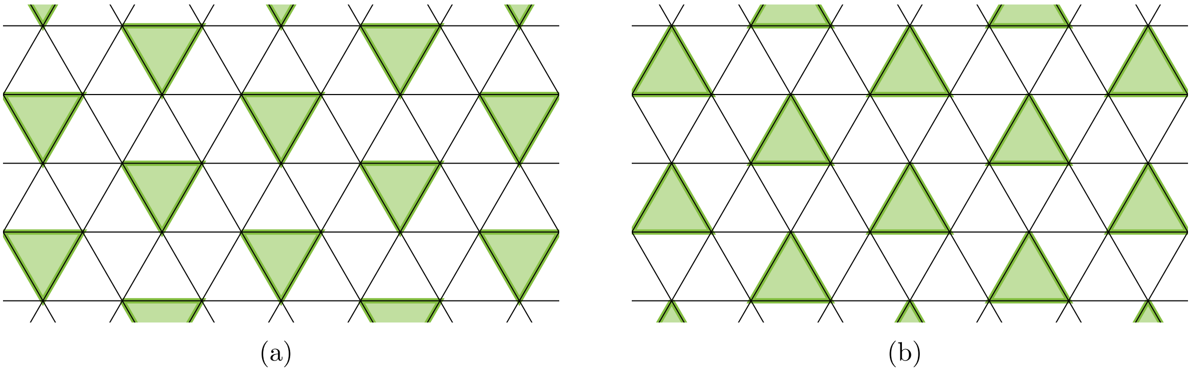


Figure 5.4: This figure shows the two orientations of the blocks that may be chosen in each RG step. The blocks in (a) point downwards, whereas those in (b) point upwards.

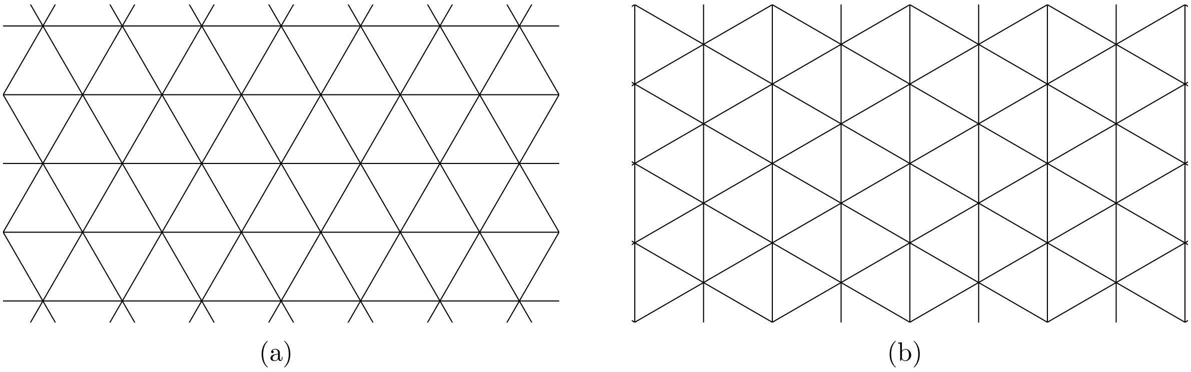


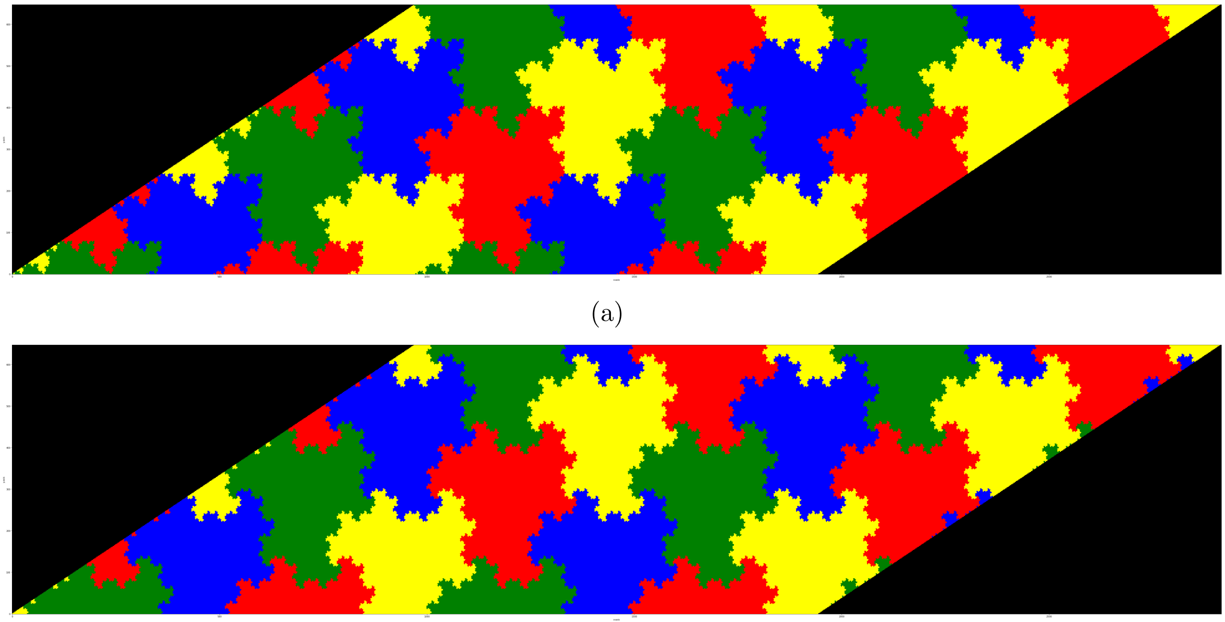
Figure 5.5: Illustration of two different orientations of the triangular lattice that alternate with each blocking step.

5.2 Verification of Cluster-inspired RG on a Small Triangular Lattice

The operation of the cluster-inspired RG was tested on a small triangular multigrid with 12 original spins and 4 renormalized spins, related by one blocking step. This small system was chosen for its small number of configurations that allow to perform exact calculations by summation over all of them. The exact results allow to evaluate the numerical results of the Monte Carlo cluster algorithm by comparison.

A hexagonally shaped lattice can not easily be implemented in an algorithm the way it is, as it cannot be structured by rows and columns of constant length. It can, however, be cut and rearranged into a rhomboidally-shaped lattice in a way that the two systems are equivalent when endowed with periodic boundary conditions. The rhomboidal shape can then be structured by rows and columns. This process is shown in Figure 5.7.

It was found geometrically that the renormalized system with four spins and periodic boundary conditions may only have three configurations with distinct Boltzmann weights. All others are equivalent to these three due to translational symmetry and spin-flip symmetry. We defined



(a)

(b)

Figure 5.6: This figure shows two different types of block sectors that have both been produced from 9 iterated blocking steps on a triangular lattice. The size of the lattice was chosen such that the renormalized lattice consists of four remaining spins, related to four differently coloured block sectors. The plots each show a multiple of four renormalized systems, divided into block sectors, that have been periodically appended to each other in order to better show the equality in shape of the block sectors. (a) shows the block sectors related to the alternation of the two block orientations, in which the blocks point left and down. (b) shows the block sectors related to the alternation of all four block orientations in a cyclic pattern.

the first configuration to consist of four up spins, the second to consist of three up and 1 down spin and the third of two up and two down spins. The three configurations are illustrated in Figure 5.8.

Configurations Boltzmann weights are not directly accessible in Monte Carlo data, as they would first need to be correctly normalized. An entity that can easily be extracted, however, is the ratio between any two configurations' Boltzmann weights, that can be approximated by the configurations' relative frequencies. A system with only three distinct configurations can be characterized by two of their mutual weight ratios, which we used as variables in our verification.

The exact computation iterated through all existing original configurations and summed up the weight contributions to each renormalized configuration under the incorporation of the blocking kernel.

The numerical simulation was based on the cluster algorithm, utilizing its flexibility to flip clusters as desired. To improve accuracy, we designed the algorithm that restricted the renormalized system to only two out of three configurations in a single simulation, to then compare the weights of these two configurations. This was achieved by fixing three out of the four

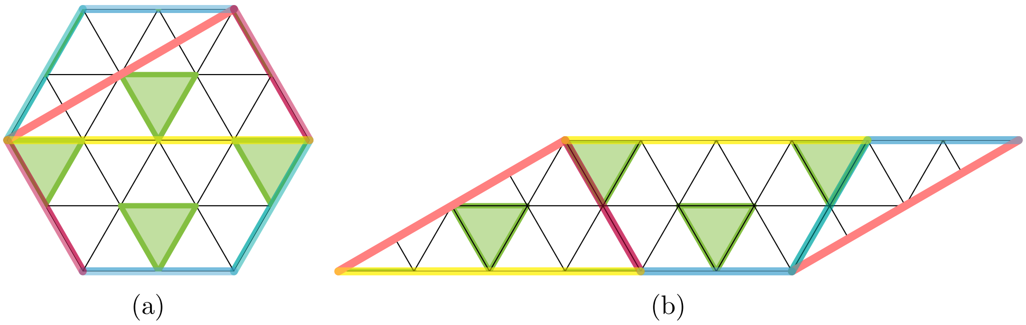


Figure 5.7: Schematic illustration of the 12 spin triangular lattice as a hexagonal shape in (a) and as a rhomboidal shape in (b). The hexagonal shape may be rearranged to a rhomboidal shape by cutting it into three pieces and rearranging these. The colored lines in colors other than green symbolize the cutting lines.

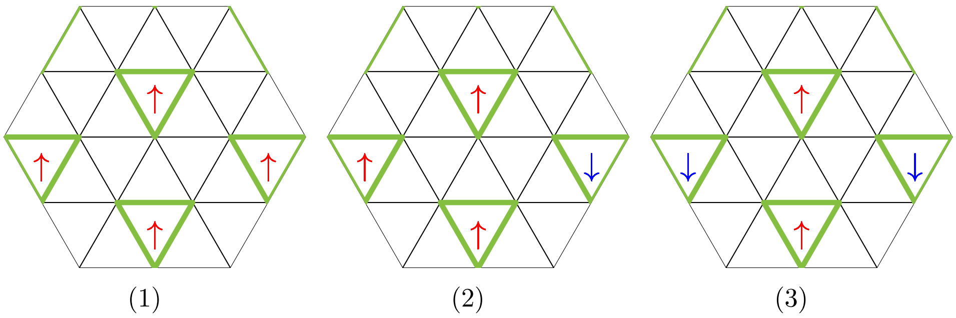


Figure 5.8: Schematic illustration of the three distinct configurations on the renormalized 4-spin triangular lattice with periodic boundary conditions.

renormalized spins to a specific values by ensuring that clusters containing any of these spins were not allowed to flip. This is a legitimate method as it obeys the only restriction of spin-flip symmetry in its design. The original system still obeys ergodicity and detailed balance.

The successful operation of the cluster-inspired RG was verified by satisfactory statistical evidence. The results of the ratio between configuration 1 and configuration 2 for different values of the inverse temperature β as well as the parameter A are found in Figure 5.9. The results of the ratio between configuration 2 and configuration 3 are omitted as they look very similar. The errors throughout the tested range of parameter values β and A were estimated by bootstrap and seemed to match the observed deviations, that could be controlled by a sufficient sample size.

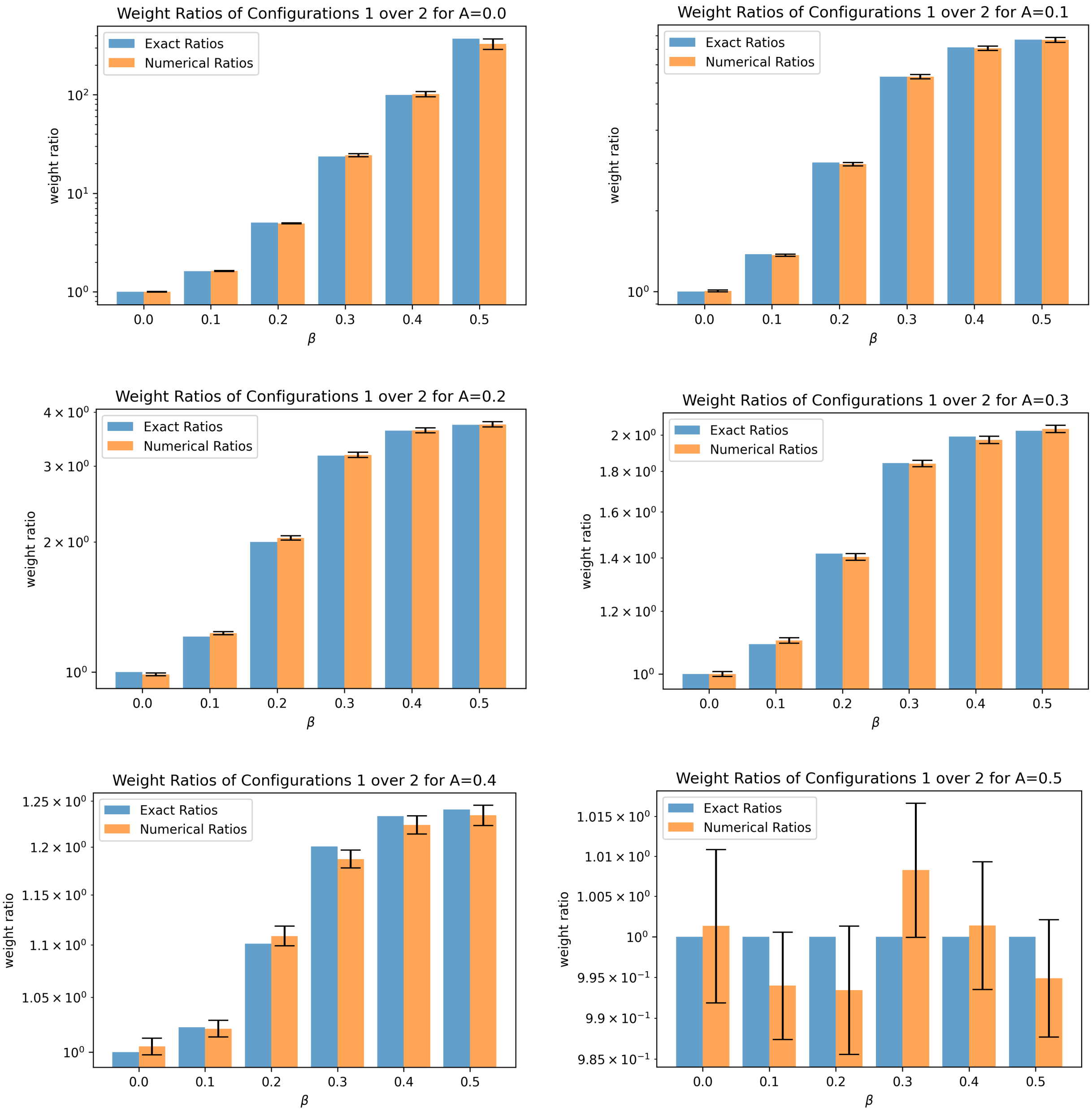


Figure 5.9: Plots showing the results of the weight ratios between configuration 1 and configuration 2. The exact results in blue are compared to the numerical results in orange under the inclusion of error bars for different values of β and A. The y-axis is displayed on a logarithmical scale for better readability.

6 Conclusion

We have introduced the cluster-inspired RG transformation, which operates on Ising models and enables the execution of a cluster algorithm on the multigrid. This transformation employs a blocking kernel that bonds renormalized spins to original spins, integrating them into their original clusters. The concept is applicable to various dimensions and lattice types, with the only parameter affecting the blocking kernel being the number of spins per block.

We applied the cluster-inspired RG transformation to the 1D Ising model, integrating out the spins to operate in the bond representation. Our goal was to formalize the process, iterate it analytically, and ideally solve it. We successfully formalized the first blocking step but did not achieve sufficient formalization of the second step to continue iterating the blocking further. The formalization of the first blocking step revealed that the renormalized system could no longer be described using a bond representation that includes only nearest-neighbor bonds. Instead, bonds also occur between more distant spins, which we term superbonds. These superbonds prevent the renormalized bond system from being described by ultra-local variables, seemingly breaking its ultra-locality. However, this presents an unsolved puzzle. While working with the spin-based RG transformation, we discovered a special value of the parameter A that preserves the Ising interactions. When A is set to this value, the cluster-inspired RG transformation should also yield ultra-local interactions. However, we did not observe any confirmation of this. How can this be compatible with the existence of superbonds? Initially, we hypothesized that setting $A = 0$ would result in nearest-neighbor interactions, as superbonds are absent at that value. However, this hypothesis did not match our results. If our insights are accurate, then the translation from a renormalized bond and superbond representation back to a spin representation must be more complex than we initially expected. While the first blocking step was formalized and simplified to a great extent, the approach used for its iteration seems to be a dead end. It is possible that a different approach might work better, such as one that does not treat the blocking steps separately but instead starts directly with n blocking steps. One might also consider leveraging the fact that the weights of activated kernel bonds are uniform within the block sectors. However, these are just preliminary ideas.

After completing the analytical work on the 1D Ising model, we applied the cluster-inspired RG transformation to the 2D Ising model and executed the Monte Carlo cluster algorithm on the multigrid. We chose to focus on the triangular lattice due to its additional symmetries compared to the square lattice. To verify the successful operation of the concept, we began with a small system consisting of 12 original spins and 4 renormalized spins. This allowed us to compare numerical data to analytical results, which could be obtained for such a small system. We designed the cluster simulation to restrict the renormalized system to occupy only two configurations, taking advantage of the freedom of cluster flipping. We collected numerical data on the weights of the configurations and compared their ratios to the analytically obtained ratios. The data provided clear evidence for the successful operation of the cluster algorithm on the multigrid.

This lays the foundation for further steps towards parametrizing fixed points. The next step in the project should likely involve further consideration of how to best parametrize the Hamilton function and apply the cluster-inspired RG to larger systems.

Bibliography

[1] W. Bietenholz and U.-J. Wiese, *Uncovering Quantum Field Theory and the Standard Model: From Fundamental Concepts to Dynamical Mechanisms.* Cambridge University Press, 2024.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[3] U.-J. Wiese, "Statistical Mechanics." www.wiese.itp.unibe.ch/lectures/thermodyn.pdf, 2010. Lecture notes.

[4] R. J. Baxter, *Exactly solved models in statistical mechanics.* Courier Corporation, 2007.

[5] T. G. Mackay and A. Lakhtakia, *The transfer-matrix method in electromagnetics and optics.* Springer Nature, 2022.

[6] L. Onsager, "Crystal statistics. i. a two-dimensional model with an order-disorder transition," *Physical Review*, vol. 65, no. 3-4, p. 117, 1944.

[7] R. Gupta, A. Mishra, and A. Pathak, "Supercritical fluid technology a boon for pharmaceutical particle manufacturing," *Science & Technology For Human Development*, 2014.

[8] J. M. Yeomans, *Statistical mechanics of phase transitions.* Clarendon Press, 1992.

[9] V. Beffara and H. Duminil-Copin, "Critical point and duality in planar lattice models," *Probability and statistical physics in St. Petersburg*, vol. 91, pp. 51–98, 2014.

[10] U.-J. Wiese, "Renormalization group for the ising model." 2024.

[11] J. J. Binney, N. J. Dowrick, A. J. Fisher, and M. E. Newman, *The theory of critical phenomena: an introduction to the renormalization group.* Oxford University Press, 1992.

[12] J.-S. Wang, "Critical dynamics of the swendsen-wang algorithm in the three-dimensional ising model," *Physica A: Statistical Mechanics and its Applications*, vol. 164, no. 2, pp. 240–244, 1990.

[13] R. H. Swendsen, J.-S. Wang, and A. M. Ferrenberg, "New monte carlo methods for improved efficiency of computer simulations in statistical mechanics," *The Monte Carlo method in condensed matter physics*, pp. 75–91, 2005.

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

[15] B. Bessire, "Ising-modell mit monte carlo methode," may 2007. Bachelor's thesis.

[16] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap.* Chapman and Hall/CRC, 1994.

[17] L. P. Kadanoff, W. Götze, D. Hamblen, R. Hecht, E. Lewis, V. V. Palciauskas, M. Rayl, J. Swift, D. Aspnes, and J. Kane, “Static phenomena near critical points: theory and experiment,” *Reviews of Modern Physics*, vol. 39, no. 2, p. 395, 1967.

[18] K. G. Wilson and M. E. Fisher, “Critical exponents in 3.99 dimensions,” *Physical Review Letters*, vol. 28, no. 4, p. 240, 1972.

[19] K. G. Wilson and J. Kogut, “The renormalization group and the epsilon expansion,” *Physics reports*, vol. 12, no. 2, pp. 75–199, 1974.

[20] K. G. Wilson, “The renormalization group: Critical phenomena and the kondo problem,” *Reviews of modern physics*, vol. 47, no. 4, p. 773, 1975.

[21] K. G. Wilson, “Renormalization group and critical phenomena. i. renormalization group and the kadanoff scaling picture,” *Physical review B*, vol. 4, no. 9, p. 3174, 1971.

Declaration of consent

on the basis of Article 30 of the RSL Phil.-nat. 18

Name/First Name: Sprecher Andrin

Registration Number: 11-920-840

Study program: Master of Science in Theoretical Physics

Bachelor □ Master ☑ Dissertation □

Title of the thesis: Introduction to Cluster-Inspired RG on the Ising Model

Supervisor: U.J. Wiese

I declare herewith that this thesis is my own work and that I have not used any sources other than those stated. I have indicated the adoption of quotations as well as thoughts taken from other authors as such in the thesis. I am aware that the Senate pursuant to Article 36 paragraph 1 litera r of the University Act of 5 September, 1996 is authorized to revoke the title awarded on the basis of this thesis.

For the purposes of evaluation and verification of compliance with the declaration of originality and the regulations governing plagiarism, I hereby grant the University of Bern the right to process my personal data and to perform the acts of use this requires, in particular, to reproduce the written thesis and to store it permanently in a database, and to use said database, or to make said database available, to enable comparison with future theses submitted by others.

Bern, 3.7.2024

Place/Date

A. Sprecher

Signature