UNIVERSITAT DE BARCELONA

ADVANCED MATHEMATICS
MASTER'S FINAL PROJECT

Markov chains for non abelian gauge theories

Author:
Juan Carlos Díaz Rodríguez

Supervisor:
Sofyan Iblisdir
Bruno Julia Díaz
Marina Gonchenko

Facultat de Matemàtiques i Informàtica

June 28, 2024

Abstract

Markov chains for non abelian gauge theories

We present a Metropolis-Hastings Markov chain for correlated systems of continuous variables ruled by the Boltzmann distribution. An initial introduction to the algorithm with discrete variables is given and then extended to variables in a compact group. We initially work with the $SU(2)$ Higgs model with frozen matter term on a 2D lattice. The algorithm is then extended to a 3D lattice. The configuration updates are done though tensor network renormalization schemes that approximate the partition function of the system. We evaluate the performance of the chains based on equilibration and autocorrelation times. The mean value of different observables are compared to existing literature and additional computations are done with the groups $SU(3)$, $SU(5)$ and $SU(10)$.

Keywords

Monte Carlo, Metropolis-Hastings, tensor network, Ising model, Wilson action, Higgs model, 3D lattice.

Contents

List of figures

Chapter 1

Introduction

There is a fundamental concept that has paved our understanding of fundamental physics during the last century: gauge symmetries. Through its study, physicists have developed theories explaining the four basic forces of nature, among other results (Lancaster and Blundell, 2014; Tong, 2018). These theories are described in terms of Lagrangian mechanics, where a system is said to be gauge symmetric if its Lagrangian is invariant under a set of transformations, the gauge symmetry group. Many of the interesting systems formulated in this regime, such as Yang-Mills theories, cannot be treated analytically. To face this problem, scientists have turned their attention to Markov chain Monte Carlo.

To study these models, a discretization of space-time into an Euclidean lattice is taken, where the degrees of freedom are placed on the vertices or the edges of the lattice. These degrees of freedom make up what is known as a configuration, and one uses Markov chains to sample these configurations according to the Boltzmann distribution. A numerical approach such as Monte Carlo is needed because the computational cost of Boltzmann sampling scales exponentially with the lattice size. Typically, the updates of the stochastic process are done at one degree of freedom at a time. In this project, we present a Markov chain with collective updates. We do this with the aid of tensor networks, which help us in sampling approximately the Boltzmann distribution at each step. Collective updates are desirable because when the energetic landscape of the system has many local minima, overcoming energy barriers with local changes can be hard. Global updates are less likely to get stuck in these local extrema, hence moving more effectively through the configuration set.

Monte Carlo sampling has been a very successful tool in the last six decades on this field, and more recently tensor networks have been introduced to study the structure of these theories. The fusion of both strategies in what we know as Tensor Network Metropolis-Hastings (TNMH) has led to good results in systems with a finite set of values for the degrees of freedom (Frías Pérez et al., 2023). In that same work, a suggestion to adapt the chain to theories with compact gauge groups is given. The present work is the continuation of the research done during my Bachelor's final project on the extension of the algorithm to compact groups (Díaz Rodríguez, 2023).

1.1 Setting

To give the reader some context, this section contains an introduction to the physical models treated throughout the project and an idea on how they are treated in Physics. We also give a brief review of the main mathematical tools that will be used: Monte Carlo algorithms and tensor networks.

1.1.1 Ising model

In Statistical Mechanics, we study a system of many interacting particles through the possible microstates it could be in. All of those states share some common properties (Schwartz, 2019). They can be aspects like the number of particles, energy, temperature or pressure. The set of possible states is known as *ensemble*, and depending on which common properties are fixed there are different kinds of them. For instance, the *microcanonical ensemble* can be found in an isolated system with constant number of particles and energy (Reichl, 1998). We will be focusing on the *canonical ensemble*, the set of states of a system with fixed number of particles connected to a heat bath acting as an infinite source or sink of energy that keeps the system at constant temperature T. In this regime, the probability of the system being at a particular state i with energy ε_i is given by the Boltzmann distribution

$$P_i = \frac{1}{Z} e^{-\beta \varepsilon_i}, \tag{1.1}$$

where $\beta = \frac{1}{k_B T}$ and $Z = \sum_i e^{-\beta \varepsilon_i}$ is known as the *partition function*, where the sum is taken over all states in the ensemble. Knowing the partition function exactly has a lot of value, as it allows us to calculate many quantities of the system. However, we will almost never be able to calculate this function accurately, even less take derivatives, so we will not go into its analytical uses. The reason why it is in general hard to have access to an expression of Z, or even a mean to compute it, is that the number of states increases exponentially with the system size. So even numerical methods scale very badly for large systems.

The Ising model is a paradigmatic many-body system of Statistical Physics due to its simplicity yet rich behavior. It is used as the testing ground of sampling algorithms as well as a source of many interesting physical discoveries. It was initially introduced by Ernst Ising and Wilhelm Lenz in 1924 to study the ferromagnetic properties of materials (Ising, 1924). The behavior of a many-body system is given by its Hamiltonian, for now just an energy function. The configurations are given by a tuple of values, each one associated with a point in the lattice. In the Ising model, this lattice is usually of square shape, as can be seen in Figure 1.1. The set of values for each site can differ from model to model, depending on its physical considerations. For now, the particles are highly anisotropic spins, i.e. they only take two values: 1 and -1.

Particles interact with each other in many different ways, but due to the intensity decay with distance, the most common representation of the dynamics in the Hamiltonian is given by nearest neighbor interactions, represented as the edges in Figure 1.1. The model is also equipped with the effect of an external magnetic field.

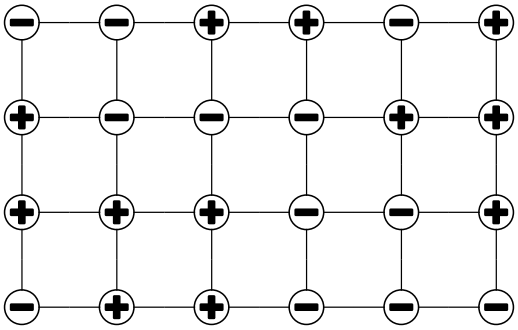


Figure 1.1: Example of a 4×6 square lattice for the Ising model

The Hamiltonian of an Ising model is given by

$$H(\omega) = -\sum_{\langle i,j\rangle \in E} J_{ij}\sigma_i\sigma_j - \sum_{i \in V} h_i\sigma_i, \tag{1.2}$$

where the spins are $\sigma_i \in \{-1, 1\}$, J_{ij} represents the coupling between particles and h_i is the field acting on site i. The set of all possible configurations ω, or state space, is denoted by Ω, previously referred to as the ensemble. In the case of the Ising model, $\Omega = \{-1, 1\}^{|V|}$. Despite all the interesting phenomenology observed in this model, this is all that is needed to describe its dynamics.

1.1.2 Gauge symmetries

A system is said to be *gauge symmetric* if the measurable quantities of the system do not change under certain transformations of the degrees of freedom. These transformations have a group structure and they are known as the *gauge group*. The transformation of each degree of freedom can be parametrized by its space-time coordinates, in which case it is known as a local symmetry, or rather acts in the same way at every site, known as global symmetry. The usual way to work with these symmetries is to introduce them into the Lagrangian in the form of a new field. This gauge field is a mathematical tool to study certain systems that have been observed to be invariant under these transformations. In Physics, it is referred to as *field theory* the description of the dynamics of a physical field, such as the magnetic field for instance. A *gauge theory* is then a field theory whose Lagrangian presents gauge symmetries.

Probably, the best way to introduce gauge symmetries is by quoting the introduction of Tong (2018).

> Gauge symmetry is, in many ways, an odd foundation on which to build our best theories of physics. It is not a property of Nature, but rather a property of how we choose to describe Nature. Gauge symmetry is, at heart, a redundancy in our description of the world. Yet it is a redundancy that has enormous utility, and brings a subtlety and richness to those theories that enjoy it.

Gauge symmetries are, in fact, just a property of the Lagrangian describing the dynamics of whatever physical reality it describes. The importance of this concept

rests in physicists designing the theory with these invariances as present in their equations as possible. The best well-known example that presents a gauge symmetry is the electrostatic potential. A basic study of electromagnetism reveals that adding any given quantity to the potential at every point in space makes no change in the system. This would be a global symmetry, but in fact, the quantity added does not need to be the same at every point. One can add a gradient of an arbitrary scalar field and still leave the physical observables unchanged. This is an example of a local symmetry and they are usually the most interesting ones to study. The potential is just a mathematical tool, not a physical quantity that can be measured. However, if one bases its derivations on the fact that a local change in the potential leaves unchanged all physical observables, one can find elegant derivations of the same theory and even move into the quantum description with more ease.

It turns out, that the Standard Model is written in terms of gauge theories. Understanding how gauge fields behave, even though they represent no physical realities on their own, is a crucial step in studying Quantum Field Theories or Condensed Matter Physics. In this context, the Wilson action captures many of the properties observed in the Standard Model and is of special interest to understanding Yang-Mills theories. In fact, the Wilson action is the lattice discretization of pure gauge Yang-Mills theories. Moreover, to get an action that does represent some physical reality, like the weak or strong forces, it is only necessary to add some extra terms to the action (Kogut, 1979).

We will give a description of this action in Chapters 3 and 4, where we first work with the pure gauge action and later add additional terms to describe what is known as Higgs model.

One of the simplest models with a local symmetry was proposed by F. Wegner, where he introduced a local symmetry to the Ising model (Wegner, 1971). The model is placed again in a 2D square lattice, and the minimal squares of the lattice are called *plaquettes*. Each spin now interacts with those that belong to the same plaquette. We can denote, for a plaquette P, the spins of that plaquette as $\sigma_{P(1)}, \sigma_{P(2)}, \sigma_{P(3)}$ and $\sigma_{P(4)}$, the Hamiltonian is given by

$$H(\omega) = -\sum_{P} J_P \sigma_{P(1)} \sigma_{P(2)} \sigma_{P(3)} \sigma_{P(4)}, \tag{1.3}$$

where the sum is taken over all plaquettes of the lattice. In this project we will not be using the spin model of Wegner, but a similar one where spins will be placed on the edges and the interaction are also plaquette-like.

1.1.3 Physical problem

We want to give an intuition on what we are trying to solve and what is the interest behind it. Our setting starts with a Hamiltonian operator H that generates the evolution of a quantum system. We are interested in finding a pair $(\varepsilon_0, |\psi_0\rangle)$ such that

$$\varepsilon_0 = \frac{\langle \psi_0 | H | \psi_0 \rangle}{\langle \psi_0 | \psi_0 \rangle} = \min_{\{|\psi\rangle\}} \left\{ \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} \right\}. \tag{1.4}$$

This is, we are looking for the minimum eigenvalue of our Hamiltonian and its associated eigenstate, usually known as ground state. We say that the ground state is degenerate if the multiplicity of the minimum eigenvalue is higher than one.

The Schrödinger equation in its simplest form (omitting constants) is given by

$$iH\left|\phi\right\rangle = \partial_t \left|\phi\right\rangle . \tag{1.5}$$

If the Hamiltonian does not depend on time, this is a simple linear differential equation of known solution $\left|\phi\left(t\right)\right\rangle = e^{-itH}\left|\phi_0\right\rangle$. If instead we take a Wick rotation into imaginary time as $t = -i\tau \in i\mathbb{R}$ the Schrödinger equation becomes

$$H\left|\phi\right\rangle = \partial_\tau \left|\phi\right\rangle \tag{1.6}$$

with a decaying exponential $\left|\phi\left(\tau\right)\right\rangle = e^{-\tau H}\left|\phi_0\right\rangle$ for solution, whose study is more convenient.

Let us go back to finding the ground state, which we suppose is non-degenerate. Evolution operators are hermitian, hence they are diagonalizable and their eigenvalues are real. So, we can represent H as $\sum_i \varepsilon_i \left|\psi_i\right\rangle \left\langle\psi_i\right|$, with $\varepsilon_i \in \mathbb{R}$, $\left\langle\psi_i|\psi_j\right\rangle = \delta_{ij}$, which transforms the expression for the time evolution of the state to

$$e^{-\tau H} = \sum_i e^{-\tau\varepsilon_i}\left|\psi_i\right\rangle\left\langle\psi_i\right| = e^{-\tau\varepsilon_0}\left(\left|\psi_0\right\rangle\left\langle\psi_0\right| + \sum_{i>0} e^{-\tau(\varepsilon_i-\varepsilon_0)}\left|\psi_i\right\rangle\left\langle\psi_i\right|\right). \tag{1.7}$$

There is a very important concept in this equation, the energy gap. This is the difference between the lowest eigenvalue ε_0 and the second smallest one, let us call it ε_1. Hence, the time evolution operator tends to a projector into the eigenspace of the ground state, exponentially fast in a factor depending on the energy gap. As we assumed this eigenspace to have dimension one because the ground state is non-degenerate, given any initial condition $\left|\phi_0\right\rangle$ with overlap with the ground state, the evolution tends to the ground state as

$$\left|\psi_0\right\rangle = \lim_{\tau\to\infty} \frac{e^{-\tau H}\left|\phi_0\right\rangle}{\left\|e^{-\tau H}\left|\phi_0\right\rangle\right\|}. \tag{1.8}$$

Another important aspect is the computation of expectation values of observables. Given an observable X, its expected value when the system is in the ground state is given by

$$\left\langle X\right\rangle = \lim_{\tau\to\infty} \frac{\left\langle\phi_0|e^{-\tau H}Xe^{-\tau H}|\phi_0\right\rangle}{\left\langle\phi_0|e^{-2\tau H}|\phi_0\right\rangle}. \tag{1.9}$$

This identity is the starting point of correspondence between Quantum Mechanics in d dimensions and Statistical Mechanics in $d+1$ dimensions. To complete such relation, one can make use of the Suzuki-Trotter identity. An example of such calculations is given in Appendix A.

1.1.4 Monte Carlo sampling

We have already seen that knowledge of the partition function, on which Statistical Mechanics relies, is usually limited. For this reason, numerical methods are employed. Probably, the most successful of them all across a vast amount of scientific fields is that of Monte Carlo sampling (Katzgraber, 2011). The main objective of this type of algorithms is to compute integrals of the form

$$\mathbb{E}_\rho [f] = \int_\Omega f(\omega)\, \rho(\omega)\, d\omega. \tag{1.10}$$

In the following, we will be interested in computing expectation values of observables, focusing for now on the case of finite Ω and π a probability instead of a density of probability. The expectation value of and observable $\mathcal{O}$ takes the form

$$\langle \mathcal{O} \rangle \coloneqq \mathbb{E}_\pi [\mathcal{O}]. \tag{1.11}$$

If one is able to draw N independent and identically distributed samples $\omega^{(n)} \sim \pi$, the Law of Large Numbers gives us a good approximation to $\langle \mathcal{O} \rangle$ as

$$\langle \mathcal{O} \rangle \approx \overline{\mathcal{O}} \coloneqq \frac{1}{N} \sum_{n=1}^{N} \mathcal{O}\left(\omega^{(n)}\right). \tag{1.12}$$

In such case, the variance of the estimator is given by

$$\sigma^2 = \frac{1}{N} \mathrm{Var}_\pi [\mathcal{O}], \tag{1.13}$$

where the variance of an observable θ from a sample space with distribution p is defined as

$$\mathrm{Var}_p [\theta] = \mathbb{E}_p \left[(\theta - \mathbb{E}_p [\theta])^2 \right]. \tag{1.14}$$

We will use Markov chain Monte Carlo to sample our state space, effectively sampling the observables (Katzgraber, 2011).

1.1.5 Tensor Networks

Monte Carlo sampling has been around for almost 70 years and has been the main tool for studying statistical mechanics models. In more recent years, other mathematical objects have been designed to help our understanding of these strong correlated systems. It is the representation of the partition function as the ensemble obtained from wiring mathematical objects that we will later call tensors. These constructions are known as tensor networks and they have gained a lot of importance in this field (Perez-Garcia et al., 2006; Kuramashi and Yoshimura, 2019; Bazavov et al., 2019). The mere representation of the partition function as a tensor network does not take away the exponential cost by itself. To face that, we use what is known as the renormalization group. There are different strategies like MPS truncation (Murg et al., 2005) or restructuring the network with fewer operators each time (Levin and Nave, 2007). In general, all these methods are based on the reduction

of the tensors employing singular values truncation, with the consequent loss of information, hoped to be irrelevant. Depending on the topology of the network, these decompositions can support different approaches. It is nonetheless a very common technique for different models (Bazavov et al., 2019; Yu et al., 2014). The main problem is the introduction of systematical errors coming from the deleted singular values.

1.2 Objectives

Once we have a better idea of the project setting, we set a collection of objectives to better judge the reach of the project:

I) Study the techniques used to judge the performance of sampling algorithms like correlation and convergence times of Markov chains.

II) Evaluate under which setting the algorithm presents an advantage over known sampling methods.

III) Expand the algorithm to work with a more complex action similar to the Higgs model.

IV) Move from a 2-dimensional lattice to a 3-dimensional one.

V) Build an expandable code base for future research related with TNMH.

1.3 Structure of this document

In Chapter 2, we will review a description of the mathematical concepts to be used. Much of the material presented in this chapter intersects with that written in my bachelor's final project but with significant revision. In Chapter 3 we look at the historical approach to the problem of sampling with Boltzmann probability in lattice gauge theories. Chapter 4 contains the results obtained in this year's research, finishing the document in Chapter 5 with an overview of the ideas collected for future work and the conclusions drawn.

Chapter 2

Mathematical concepts

This chapter covers the main mathematical concepts needed for the rest of the project. We start studying Markov chains, their definition and important properties that make them viable tools for our sampling objective. Next, we introduce autocorrelation times. They are of great importance to estimate the error given by Markov chain sampling. The following section will summarize the original algorithm on which the project is based, *Tensor Network Metropolis-Hastings*. Finally, an introduction to Tensor Networks is given, as their use is one of the distinctive characteristics of TNMH.

2.1 Markov chains

Definition 2.1.1 (Stochastic process and Markov chain)**.** *A* stochastic process $\boldsymbol{X} = \{X(t) : t \in T\}$ *is a collection of random variables indexed by* t*, that generally represents time.*

If $T = \mathbb{N}$*, we will call* $\boldsymbol{X}$ *a* discrete time process*.*

A discrete time process $\boldsymbol{X} = \{X_0, X_1, X_2, \dots\}$ *is a* Markov chain *if the state at time* t *only depends on the state at* $t-1$*. In other words*

$$P(X_t = a_t \mid X_{t-1} = a_{t-1}, \dots, X_0 = a_0) = P(X_t = a_t \mid X_{t-1} = a_{t-1}) \tag{2.1}$$

for every value $a_0, a_1, \dots, a_t$ *and every* $t \geq 1$*.*

Definition 2.1.2 (Homogeneity)**.** *A Markov chain is* homogeneous *if the transition probabilities are independent of time. These probabilities will be denoted by*

$$P_{i,j} \coloneqq P(X_t = j \mid X_{t-1} = i)$$

for every t*.*

Definition 2.1.3 (Finite Markov chain)**.** *A Markov chain is* finite *if the set of values taken by the random variables is finite.*

Definition and Remark 2.1.4. *Let $\boldsymbol{X}$ be a homogeneous finite Markov chain, the* transition matrix *is defined as*

$$\boldsymbol{P} = \begin{pmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,n} \\ P_{2,1} & P_{2,2} & \dots & P_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n,1} & P_{n,2} & \dots & P_{n,n} \end{pmatrix}.$$

If we define $P_{i,j}^m$ as the probability to go from state i to state j in m steps, then

$$P_{i,j}^m := P\left(X_{t+m} = j \mid X_t = i\right) = \left(\boldsymbol{P}^m\right)_{i,j}. \tag{2.2}$$

Definition 2.1.5 (Irreducibility)**.** *We will say that the state j is* accessible *from state i if there exists a finite $n \geq 0$ such that $P_{i,j}^m > 0$. If two states are accessible from each other we will say they are* communicated*. Finally, we call a chain* irreducible *if all states are communicated with each other.*

Definition 2.1.6 (Aperiodic)**.** *The* period *$d(k)$ of a state k in a homogeneous Markov chain is given by $d(k) = \gcd\{m \geq 1 \colon P_{k,k}^m > 0\}$. If $d(k) = 1$ we say that k is* aperiodic*. An* aperiodic *Markov chain is one whose states are all aperiodic.*

To illustrate the use of these definitions we give the following result.

Proposition 2.1.7. *Let $\boldsymbol{X}$ be an irreducible, finite and aperiodic Markov chain with transition matrix $\boldsymbol{P}$. Then, there exists $M < \infty$ such that $\left(\boldsymbol{P}^n\right)_{i,j} > 0$ for every state i, j and every $n \geq M$.*

This is, under these conditions, there is a sufficiently large number of steps after which a transition between any pair of states has a positive probability of happening.

Definition 2.1.8 (Stationary distribution)**.** *Let $\boldsymbol{X}$ be a finite Markov chain with transition matrix $\boldsymbol{P}$. A row vector $\pi = (\pi_1, \dots, \pi_k)$ is a* stationary distribution *if:*

- $\pi_i \geq 0 \; \forall \, i = 1, \dots, k$ *and* $\sum_{i=1}^k \pi_i = 1$.
- $\pi \boldsymbol{P} = \pi$.

The unique existence and convergence of the chain to this distribution is ensured by the following results:

Theorem 2.1.9. *Every finite, irreducible and aperiodic Markov chain has a unique stationary distribution.*

Definition 2.1.10. *Let $\nu = (\nu_1, \dots, \nu_k)$ and $\nu' = (\nu'_1, \dots, \nu'_k)$ be probability distributions over $S = \{s_1, \dots, s_k\}$, we define the* total variation distance *between ν and ν' as*

$$d_{TV}\left(\nu, \nu'\right) = \frac{1}{2} \sum_{i=1}^{k} \mid \nu_i - \nu'_i \mid .$$

If $\nu, \nu^{(1)}, \nu^{(2)}, \dots$ are distributions over S, we say that the sequence $\{\nu^{(n)}\}_{n \in \mathbb{N}}$ converges to ν in total variation *if*

$$\lim_{n \to \infty} d_{TV}\left(\nu^{(n)}, \nu\right) = 0.$$

Theorem 2.1.11. *Let $\boldsymbol{X}$ be a finite, irreducible and aperiodic Markov chain with transition matrix $\boldsymbol{P}$. If $\nu^{(0)}$ is a given initial distribution and ν is the stationary distribution, then $\nu^{(m)} = \nu^{(0)}\boldsymbol{P}^m$ converges in total variation to ν.*

Let us introduce a final concept that very useful tool for identifying the stationary distribution.

Definition 2.1.12 (Reversibility)**.** *Let $\boldsymbol{X}$ be a finite Markov chain with transition matrix $\boldsymbol{P}$. A distribution π is* reversible *is*

$$\pi_i \boldsymbol{P}_{i,j} = \pi_j \boldsymbol{P}_{j,i} \tag{2.3}$$

for every state i, j. A Markov chain is reversible *if there exists a reversible distribution for such a chain.*

Observe that, under these conditions, π is a fixed point of the map defined by $\mathbf{P}$.

Theorem 2.1.13. *Let $\boldsymbol{X}$ be a finite Markov chain. If π is reversible then it is also a stationary distribution.*

An example of a Markov chain is that proposed by Metropolis in his celebrated algorithm. Suppose we have a function $f(x)$ proportional to a target distribution $P(x)$. One can build a Markov chain with all the desirable properties exposed above:

(I) Propose a new state y with probability $g(y \mid x)$. In the original algorithm, this probability needs to be symmetric, i.e, $g(y \mid x) = g(x \mid y)$.

(II) Accept the state change $x \to y$ with probability $\min\{1, f(y)/f(x)\}$.

2.2 Autocorrelation times

In this section, we follow the lecture notes of Sokal (1997), which make an analysis of errors in Monte Carlo algorithms and convergence bounds. In the previous chapter we saw how the variance of the estimator $\overline{\mathcal{O}}$ is given by $\mathrm{Var}_\pi[\mathcal{O}]/N$, hence the error decreases as as $1/\sqrt{N}$. As MCMC algorithms do not generate independent samples, the previous error does not apply.

There are two different solutions to this. We can generate independent samples by simply restarting the Markov chain. This would mean waiting for equilibration every time we want a new sample. This approach is not practical at all. The other option is to study after equilibration how many steps we need to wait for samples to be uncorrelated. It is important to remark that no two random variables from the same Markov chain are independent. It is only through the conditioning to another previous step how we can achieve that, but that is not a valid condition for the previous error to hold.

We turn our study to correlation functions and how they relate to the error of the estimator computed with a Markov Chain.

Definition 2.2.1 (Autocorrelation functions)**.** *The* unnormalized autocorrelation function *of* $\mathcal{O}$ *is given by*

$$C_{\mathcal{O}}(t) = \langle \mathcal{O}_s \mathcal{O}_{s+t} \rangle - \langle \mathcal{O}_s \rangle \langle \mathcal{O}_{s+t} \rangle = \sum_{x,y \in \Omega} \mathcal{O}(x) \left[\pi(x) \left(\boldsymbol{P}^t \right)_{x,y} - \pi(x)\, \pi(y) \right] \mathcal{O}(y). \tag{2.4}$$

The normalized autocorrelation function *is then*

$$\rho_{\mathcal{O}}(t) = C_{\mathcal{O}}(t) / C_{\mathcal{O}}(0). \tag{2.5}$$

It is common practice to assume that the previous function behaves as a decaying exponential, and in fact, most of the observables do behave that way. However, this is not true in general.

It has been observed that autocorrelation functions do not behave as a decaying exponential near phase transition. In fact, this is a characteristic of such points in the parametric space, where very long autocorrelation times appear, known as critical slowing down.

Let us assume the behavior of $\rho_{\mathcal{O}}$ goes as a decaying exponential. This suggests the definition of the *exponential autocorrelation time*

$$\tau_{exp,\mathcal{O}} = \limsup_{t \to \infty} \frac{t}{-\log |\rho_{\mathcal{O}}(t)|} \tag{2.6}$$

and

$$\tau_{exp} = \sup_{\mathcal{O}} \tau_{exp,\mathcal{O}}. \tag{2.7}$$

From the definition one can see that knowing τ_{exp} will not be possible in general; it can even be unbounded.

One may define a similar value, the *integrated autocorrelation time* for a given observable $\mathcal{O}$, which will be much easier to compute

$$\begin{aligned} \tau_{int,\mathcal{O}} &= \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho_{\mathcal{O}}(t) \\ &= \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{\mathcal{O}}(t). \end{aligned} \tag{2.8}$$

The factor of $\frac{1}{2}$ is a convention, so that $\tau_{exp\mathcal{O}} \approx \tau_{int,\mathcal{O}}$ when $\rho_{\mathcal{O}}(t) \sim e^{-|t|/\tau}$. This new value is interesting as it is related to the error of a Monte Carlo measurement of $\langle \mathcal{O} \rangle$. The variance of the estimator $\overline{\mathcal{O}}$ at Eq. 1.12 can be approximated as

$$\begin{aligned} \operatorname{Var}\left[\overline{\mathcal{O}}\right] &= \frac{1}{N^2} \sum_{r,s=1}^{N} C_{\mathcal{O}}(r-s) \\ &= \frac{1}{N} \sum_{t=1-N}^{N-1} \left(1 - \frac{|t|}{N}\right) C_{\mathcal{O}}(t) \\ &\approx \frac{1}{N} \left(2\tau_{int,\mathcal{O}}\right) C_{\mathcal{O}}(0), \quad \text{for } N \gg \tau. \end{aligned} \tag{2.9}$$

As $\mathrm{Var}_\pi[\mathcal{O}] = C_\mathcal{O}(0)$, then the variance of $\mathcal{O}$ is a factor of $2\tau_{int,\mathcal{O}}$ times larger, i.e. the effective number of independent samples in a chain of length N is approximately $N/2\tau_{int,\mathcal{O}}$.

Observe how the definition of the autocorrelation functions in Eq. 2.4 is done with π as the distribution sampling each step. Hence, this computation should only be done for a time greater than the equilibration time of the chain. To judge this equilibration time, in practice, one would plot the observables under study and note when the initial transient appears to end.

Although we will not discuss much the following, it should be noted that this empirical method has its dangers. It may happen that a system seems to be reaching equilibrium when it has actually only settled down on a long-lived region of the configuration space that can be far from equilibrium. This phenomenon is known as metastability. It is desirable to identify this metastable regions and design some techniques to either detect them or give reasonable arguments to their unlikeliness, such as bounding the convergence time. This last solution, again, is hardly ever feasible.

2.3 TNMH algorithm

In this section, we will have an introduction to the original algorithm of TNMH. It was first proposed by Frías Pérez et al. (2023) and it combines the Metropolis-Hastings algorithm (Hastings, 1970) with an approximate first sampling done with tensor networks. The states in this section will be those corresponding to an Ising model, so $\omega = (\sigma_1, \sigma_2, \dots, \sigma_N) \in \{-1, 1\}^N$, where N is the size of the system.

The Metropolis-Hastings algorithm works as follows:

(I) Given an initial state ω, propose a new one ω' with probability $g^{(\beta)}(\omega' \mid \omega)$.

(II) The change $\omega \to \omega'$ is accepted with probability

$$P_{acc}(\omega \to \omega') = \min\{1, \frac{g^{(\beta)}(\omega \mid \omega')}{g^{(\beta)}(\omega' \mid \omega)} \times \frac{\pi^{(\beta)}(\omega')}{\pi^{(\beta)}(\omega)}\}. \tag{2.10}$$

When the prior distribution $g^{(\beta)}$ is symmetric, this algorithm reduces to the original one of Metropolis. However, the absence of symmetry on the prior distribution can lead to a better exploration of the state space.

The probabilities of the transition matrix will then be given by

$$\mathcal{T}(\omega \to \omega') = g^{(\beta)}(\omega' \mid \omega) \times P_{acc}(\omega \to \omega'). \tag{2.11}$$

Observe how, if $g^{(\beta)}(\omega' \mid \omega) = \pi^{(\beta)}(\omega')$, then the probability of accepting the change is 1. This makes sense, as our prior distribution would already be the target one. In the following, we will look for a prior distribution $g^{(\beta)}(\omega' \mid \omega) \approx \pi^{(\beta)}(\omega')$.

Suppose we index the vertices of the Ising model from 1 to N. By Bayes, we can express the Boltzmann distribution as

$$\pi(\omega) = \pi_1(\sigma_1)\,\pi_2(\sigma_2 \mid \sigma_1)\,\pi_N(\sigma_N \mid \sigma_1, \dots, \sigma_{N-1}), \tag{2.12}$$

where $\pi_i(\sigma_i \mid \sigma_1, \dots, \sigma_{i-1}) = \sum_{\sigma_{i+1},\dots,\sigma_n} \pi(\sigma_i, \sigma_{i+1}, \dots, \sigma_n \mid \sigma_1, \dots, \sigma_{i-1})$.

These marginal distributions can be expressed as

$$\begin{aligned} \pi_1(\sigma_1) &= Z(\beta \mid \sigma_1)/Z(\beta) \\ \pi_i(\sigma_i \mid \sigma_1, \dots, \sigma_{i-1}) &= Z(\beta \mid \sigma_1, \dots, \sigma_{i-1}, \sigma_i)/Z(\beta \mid \sigma_1, \dots, \sigma_{i-1}), i = 2, \dots, n. \end{aligned} \tag{2.13}$$

The conditioned partition functions $Z(\beta \mid \sigma_1, \dots, \sigma_i)$ can be computed from the original hamiltonian by fixing the indicated σ-values.

This reduces the problem of sampling N spin values according to the Boltzmann distribution, to sampling one spin at a time according to a Bernoulli distribution of $p = \pi_i(\sigma_1, \dots, \sigma_{i-1})$. Each new sampled spin will condition the following partition functions computed. When the process is finished we will have obtained $\omega \sim \pi^{(\beta)}(\omega)$. If we design a method to compute an approximation of these partition functions, we will be able to obtain our desired $g^{(\beta)}(\omega' \mid \omega) = \widetilde{\pi}^{(\beta)}(\omega') \approx \pi^{(\beta)}(\omega')$.

Let us close this section by checking that $\pi^{(\beta)}$ is the stationary distribution of this Markov chain. From Eq. 2.10 and 2.11, we can see that $\pi^{(\beta)}$ is reversible. Moreover, if our approximations of $Z(\beta \mid \sigma_1, \dots, \sigma_i)$ are good enough such that they are all positive, then $\widetilde{\pi}^{(\beta)}(\omega') > 0 \; \forall \omega' \in \Omega$, hence $\mathcal{T}(\omega \to \omega') > 0 \; \forall \omega, \omega' \in \Omega$. This implies that the Markov chain is irreducible and aperiodic and by Theorems 2.1.9 and 2.1.13 $\pi^{(\beta)}$ is the unique stationary distribution.

2.4 Tensor Networks

Definition 2.4.1 (Tensor Product)**.** *Let $V_1, \dots, V_n$ be K vector spaces and L the K vector spaces whose base is $V_1 \times \cdots \times V_n$, i.e L is the set of linear combinations over K whose elements are of the form $(v_1, \dots, v_n)$, $v_i \in V_i$.*

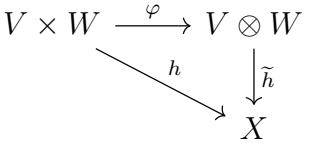
Let $\mathcal{R}$ be the linear subspace of L generated by the following relation R

$$(v_1, \dots, \alpha v_i, \dots, v_n) \sim \alpha(v_1, \dots, v_i, \dots, v_n) \; \forall i = 1, \dots, n \text{ and}$$
$$(v_1, \dots, v_i + u_i, \dots, v_n) \sim (v_1, \dots, v_i, \dots, v_n) + (v_1, \dots, u_i, \dots, v_n) \; \forall i = 1, \dots, n \text{ and}$$

The tensor product *$V_1 \otimes \cdots \otimes V_n$ is defined as the quotient $L/\mathcal{R}$ and the image of $(v_1, \dots, v_n)$ is denoted by $|v_1\rangle \otimes \cdots \otimes |v_n\rangle$ or by $|v_1 \dots v_n\rangle$.*

A discussion on the mathematical interest of this definition is given by Gowers, where the focus is placed on studying multilinear forms as linear forms. This is best seen in the following result.

Theorem 2.4.2 (Universal Property)**.** *Given two finite dimension vector spaces V and W, and for every bilinear map $h\colon V \times W \to X$ there exists a unique linear map $\widetilde{h}\colon V \otimes W \to X$ such that the following diagram conmutes.*



The bilinear map φ is constructed in a natural way as $\varphi(v_1, v_2) = |v_1\rangle \otimes |v_2\rangle$.

Let $\mathcal{B}_i = \{|v_{s_i}^{(i)}\rangle : s_i = 1, \dots, \dim V_i\}$ be a basis of V_i. Let us construct from them a basis for $V_{tot} \coloneqq V_1 \otimes \cdots \otimes V_n$. Define

$$\mathcal{B}_{tot} = \{|v_{s_1}^{(1)}\rangle \otimes \cdots \otimes |v_{s_n}^{(n)}\rangle : s_i = 1, \dots, \dim V_i,\ j = 1, \dots, n\}. \tag{2.14}$$

For a shorter notation, $|v_{s_1}^{(1)}\rangle \otimes \cdots \otimes |v_{s_n}^{(n)}\rangle \equiv |s_1 \dots s_n\rangle$. Constructed this way, $\mathcal{B}_{tot}$ is a basis of V_{tot}. The elements $|\psi\rangle \in V_{tot}$ can be expressed as

$$|\psi\rangle = \sum_{s_1,\dots,s_n} \psi_{s_1,\dots,s_n} |s_1 \dots s_n\rangle . \tag{2.15}$$

Making use of Riesz Fundamental Representation Theorem, we can write

$$\psi_{s_1,\dots,s_n} = \langle s_1 \dots s_n|\psi\rangle . \tag{2.16}$$

The dimension of these vector spaces is the product of the original spaces' dimensions. This implies that the necessary number of coefficients required to represent a general tensor grows exponentially with the number of starting spaces. Further down, we will look for special families of tensors that do not require as many coefficients, while trying to keep as much information as possible.

In the same way as vectors can be considered maps from a finite set of indices to a field, tensors can be thought of as maps from a cartesian product of indices sets. This gives us an equivalent but more practical definition to feed into a computer of a tensor.

Definition and Examples 2.4.3 (Tensor)**.** *A* tensor *is a map*

$$T\colon Q_1 \times Q_2 \times \cdots \times Q_n \to K,$$

where Q_i are finite indices sets and K is a field. The sum and scalar product operations are defined pointwise.

The simplest tensor is that with a single set of indices, i.e., a regular vector. A vector in $\mathbb{R}^n$ can be seen as

$$\begin{aligned} v\colon \{1, \dots, n\} &\to \mathbb{R} \\ i &\mapsto v_i \end{aligned}$$

In the same way, a matrix is a tensor with two indices. A complex $m \times n$ matrix is a map

$$\begin{aligned} M\colon \{1, \dots, m\} \times \{1, \dots, m\} &\to \mathbb{C} \\ (i, j) &\mapsto M_{ij} \end{aligned}$$

The following is the main operation with tensors we will be using.

Definition 2.4.4 (Contraction)**.** *Given two tensors*

$T_1\colon Q_1^{(1)} \times \cdots \times Q_{j_1-1}^{(1)} \times \widetilde{Q} \times Q_{j_1+1}^{(1)} \times \cdots \times Q_{l_1}^{(1)} \to K$

$T_2\colon Q_1^{(2)} \times \cdots \times Q_{j_2-1}^{(2)} \times \widetilde{Q} \times Q_{j_2+1}^{(2)} \times \cdots \times Q_{l_2}^{(2)} \to K$

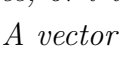
We define the contraction *of the index j_1 from T_1 with the index j_2 of T_2 as the tensor*

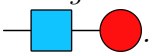
$$\widetilde{T}_{m_1,\dots,m_{j_1-1},m_{j_1+1},\dots,m_{l_1},n_1,\dots,n_{j_2-1},n_{j_2+1},\dots,n_{l_2}} =$$

$$\sum_{\alpha=1}^{|\widetilde{Q}|} (T_1)_{m_1,\dots,m_{j_1-1},\alpha,m_{j_1+1},\dots,m_{l_1}} \cdot (T_2)_{n_1,\dots,n_{j_2-1},\alpha,n_{j_2+1},\dots,n_{l_2}}$$

This definition not only shows a very useful operation, but also how tiring the notation is. That is why the following visual notation is often used.

Notation and examples 2.4.5. *Diagrammatic notation follows these two rules:*

i) A tensor of l indices, or l-tensor, is represented as a geometrical shape and l outgoing legs. Ex: A vector is represented as  *and a matrix as*  .

ii) A contraction is represented as the union of the indices' legs that are being contracted. Ex: The application of a matrix on a vector is pictured as  .

Tensor contraction allows us to build structures that will be of our interest. Moreover, the use of diagrammatic notation is the origin of the term *tensor network*.

Let $V_{tot} = V_1 \otimes \cdots \otimes V_N$ and consider its dual space V_{tot}^* and the set of linear transformations from V_{tot} to V_{tot}^*, denoted by $\mathcal{L}(V_{tot}, V_{tot}^*) \cong V_{tot} \otimes V_{tot}^*$. We are interested in a special family of tensors called Matrix Product States (MPS).

Given $D \in \mathbb{N}$, suppose we have n 3-tensor A_i, which we will treat as maps from the set of indices $Q_i = \{1, \dots, \dim V_i\}$ to the space of matrices $\mathcal{M}_{D\times D}(K)$, but for $A_1^s \in \mathcal{M}_{1\times D}(K)$ and $A_N^s \in \mathcal{M}_{D\times 1}(K)$. An MPS is a tensor that can be expressed as

$$\psi_{s_1,\dots,s_n} = \mathrm{Tr}\left[A_1^{s_1} A_2^{s_2} \dots A_N^{s_N}\right]. \tag{2.17}$$

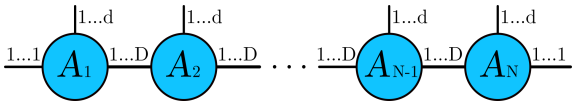


Figure 2.1: MPS in V_{tot}

Example 2.4.6. *The Greenberger-Horne-Zeilinger state (GHZ) of N qubits can be represented as the tensor*

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}\right), \tag{2.18}$$

which can be expressed as an MPS *by contracting*

$$\mathrm{GHZ}_{s_1 s_2 \dots s_N} = \mathrm{Tr}\left[\frac{1}{\sqrt{2}} A_1^{s_1} A_2^{s_2} \dots A_N^{s_N}\right], \tag{2.19}$$

where

$$\begin{gathered}\left(A_N^0\right)^T = A_1^0 = \begin{pmatrix}1 & 0\end{pmatrix}, \ \left(A_N^1\right)^T = A_1^1 = \begin{pmatrix}0 & 1\end{pmatrix}, \\ A_i^0 = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}, \ A_i^0 = \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}, \ 1 < i < N.\end{gathered} \tag{2.20}$$

Observe how this product coincides with the expected coefficients in Eq. 2.18

$$\mathrm{Tr}\left[A_1^{s_1} A_2^{s_2} \dots A_n^{s_N}\right] = \begin{cases}1 & \text{if } s_1 = s_2 = \dots = s_N \\ 0 & \text{otherwise}\end{cases} \tag{2.21}$$

In the same way, let $\mathcal{B}_i = \{l_{t_i}^{(i)} : t_i = 1, \dots, \dim V_i^*\}$ be a basis of the space V_i^* and $\mathcal{B}_{tot}^* = \{l_{t_1}^{(1)} \otimes \cdots \otimes l_{t_n}^{(n)} : t_i = 1, \dots, \dim V_i,\ j = 1, \dots, n\}$ a basis of V_{tot}^*. By Riesz Fundamental Representation Theorem, the elements $\phi \in V_{tot}^*$ can be expressed, for some $y_{t_i}^{(j)} \in V_{tot}$ as

$$\begin{aligned} \phi &= \sum_{t_1,\dots,t_n} \phi_{t_1,\dots,t_n} l_{t_1}^{(1)} \otimes \cdots \otimes l_{t_n}^{(n)} \\ &= \sum_{t_1,\dots,t_n} \phi_{t_1,\dots,t_n} \langle y_{t_1}^{(1)}, \cdot \rangle \otimes \cdots \otimes \langle y_{t_n}^{(n)}, \cdot \rangle \equiv \sum_{t_1,\dots,t_n} \phi_{t_1,\dots,s_n} \langle t_1 \dots t_n | \,. \end{aligned} \tag{2.22}$$

Again, given D the MPS in V_{tot}^* are of the form

$$\phi_{t_1,\dots,t_n} = \mathrm{Tr}\left[A_1^{t_1} A_2^{t_2} \dots A_n^{t_n}\right]. \tag{2.23}$$

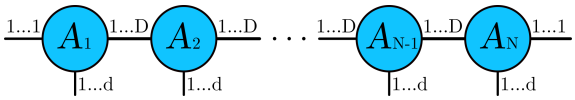


Figure 2.2: MPS in V_{tot}^*

Lastly, from the basis $\mathcal{B}_i = \{|s_i\rangle : j = 1, \dots, \dim V_i$ and $\mathcal{B}_i^* = \{\langle t_i| : j = 1, \dots, \dim V_i^*\}$ one can construct a basis for $\mathcal{L}(V_{tot}, V_{tot}^*) \cong V_{tot} \otimes V_{tot}^*$, taking into account that the tensor product is commutative and associative,

$$\mathbb{B} = \{[|s_1\rangle \otimes \langle t_1|] \otimes \dots \otimes [|s_n\rangle \otimes \langle t_n|] : t_i, s_i = 1, \dots, \dim V_i,\ j = 1, \dots, n\}. \tag{2.24}$$

With this, elements $\Phi \in \mathcal{L}(V_{tot}, V_{tot}^*)$ are given by

$$\Phi = \sum_{\substack{s_1,\dots,s_n \\ t_1,\dots,t_n}} \Phi_{s_1\dots s_n}^{t_1\dots t_n} [|s_1\rangle \otimes \langle t_1|] \otimes \dots \otimes [|s_n\rangle \otimes \langle t_n|] \tag{2.25}$$

Considering n 4-tensors, analogous to the previous 3-tensors, we can define a Matrix Product Operator (MPO) as

$$\Phi_{s_1\dots s_n}^{t_1\dots t_n} = \mathrm{Tr}\left[A_1^{s_1,t_1} A_2^{s_2,t_2} \dots A_n^{s_n,t_n}\right]. \tag{2.26}$$

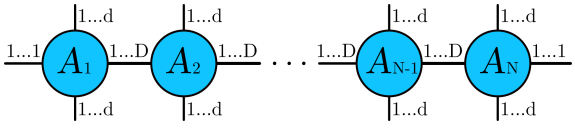


Figure 2.3: MPO in $V_{tot} \otimes V_{tot}^*$

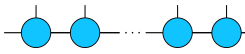
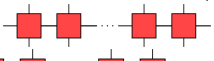
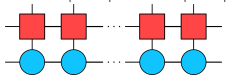
An example of these structures can be seen in the explanation for the contraction of a tensor network associated with the Ising model further down the section.

Having defined these elements, we can see how an MPO naturally acts on an MPS. We denote it by $MPO\,|MPS\rangle$ or $\langle MPS|\,MPO$ depending on whether the MPS

is considered in the dual space or not and we define it as the contraction of the indices s_i with t_i following the origin of the MPS.

We are interested in studying how much information is required to represent a tensor. This is given by the number of coefficients of the linear combination that defines it. We know that, in general, it grows exponentially with the number of vector spaces in the tensor product. If we have some extra information on the structure of a particular tensor, we can avoid storing every coefficient, as many of them will be zero. For example, the simplest tensors are those expressed as $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$, where $|\psi_i\rangle = \sum_{s_i} \psi_i^{s_i} |s_i\rangle \in V_i$. If we have a tensor product of n d-dimensional vector spaces V_i, we only need to store $n \cdot d$ coefficients. This kind of tensor has big limitations, as they cannot encode correlations when measuring in quantum systems, also known as entanglement.

We can encode entanglement however with MPS. Their representation requires $n \cdot d \cdot D^2$ coefficients, which may seem to avoid the exponential growth on n at first, as a tensor in general is given by d^n coefficients. However, this growth is hidden inside D, which we call the bond dimension. The next result illustrates how this behavior manifests.

Proposition 2.4.7. *Let* $|bot\rangle$  *be an MPS with bond dimension* D_1 *and* M  *and MPO with bond dimension* D_2*. Then, the result from* $M\,|bot\rangle$  *can be expressed as an MPO with bond dimension* $D_1 \cdot D_2$*.*

Proof. We denote the k^{th} 4-tensor  as $B_k^{h_k h_{k+1} s_k t_k}$ and the k^{th} 3-tensor  as $A_k^{\eta_k \eta_{k+1} s_k}$. We can express the components of the MPS and the MPO respectively as:

$$\langle s_1 \dots s_n | bot \rangle = \sum_{\eta_2=1}^{D_1} \sum_{\eta_3=1}^{D_1} \cdots \sum_{\eta_n=1}^{D_1} A_1^{\eta_2 s_1} A_2^{\eta_2 \eta_3 s_2} \dots A_{n-1}^{\eta_{n-1} \eta_n s_{n-1}} A_n^{\eta_n s_n} = \sum_{\{\eta\}}^{D_1} A_1^{\eta_2 s_1} \dots A_n^{\eta_n s_n}$$

$$M_{s_1 \dots s_n}^{t_1 \dots t_n} = \dots = \sum_{\{h\}}^{D_2} B_1^{h_2 s_1 t_1} \dots B_n^{h_n s_n t_n}$$

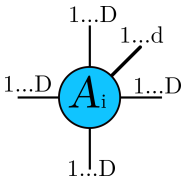
Developing the contraction $M\,|bot\rangle$ we obtain

$$\begin{aligned}
\langle s_1 \dots s_n | M | bot \rangle &= \sum_{t_1} \sum_{t_2} \dots \sum_{t_n} \left[\left(\sum_{\{h\}}^{D_2} B_1^{h_2 s_1 t_1} \dots B_n^{h_n s_n t_n} \right) \cdot \left(\sum_{\{\eta\}}^{D_1} A_1^{\eta_2 t_1} \dots A_n^{\eta_n t_n} \right) \right] \\
&= \sum_{\{h\}}^{D_2} \sum_{\{\eta\}}^{D_1} \left[\sum_{\{t\}} B_1^{h_2 s_1 t_1} \dots B_n^{h_n s_n t_n} A_1^{\eta_2 t_1} \dots A_n^{\eta_n t_n} \right] \\
&= \sum_{h_2=1}^{D_2} \sum_{\eta_2=1}^{D_1} \sum_{h_3=1}^{D_2} \sum_{\eta_3=1}^{D_1} \dots \sum_{h_n=1}^{D_2} \sum_{\eta_n=1}^{D_1} \left(\sum_{\{t\}} B_1^{h_2 s_1 t_1} \dots A_n^{\eta_n t_n} \right) \\
&= \sum_{\nu_2=1}^{D_2 D_1} \sum_{\nu_3=1}^{D_2 D_1} \dots \sum_{\nu_n=1}^{D_2 D_1} C_1^{\nu_2 s_1} C_2^{\nu_2 \nu_3 s_2} \dots C_n^{\nu_n s_n},
\end{aligned}$$

where $C_k^{h_k\eta_k h_{k+1}\eta_{k+1}t_k} = \sum_{t_k} B_k^{h_k h_{k+1} s_k t_k} A_k^{\eta_k \eta_{k+1} t_k}$ and the subindices ν are just a reindexing of any indices pair (h, η). □

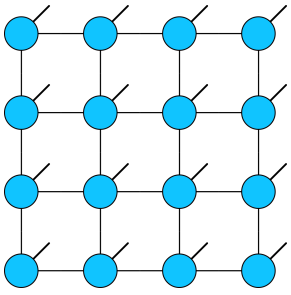
The last expression shows that $M\,|bot\rangle$ is an MPS with bond dimension $D_2 D_1$. Then, the state $M_n \dots M_2\,|bot\rangle$ is an MPS with bond dimension $D_n \dots D_2 D_1$, that is exponentially growing. The renormalization schemes we will be employing are based on keeping this bond dimension below a certain bound. Of course, this leads to a loss of information, and this is the source of our approximation. However, keeping the bond dimension bounded allows the size of the resulting MPS to scale linearly with the number of starting vector spaces.

MPS are very useful to describe systems whose particles are placed in a line and whose interactions are to close neighbors. It is only natural to extend to higher dimensions. We can focus on a 2-dimensional square grid, which takes us to the concept of *Projected entangled-pair states* (PEPS). This name comes from the way in which they are defined (Verstraete and Cirac, 2004). The original description takes a more general approach, but we will state it in a simpler way that resembles the previous constructions. To construct a PEPS we take 5-tensors,

$A_i \equiv$ 

(2.27)

The dimension of the indices at the non-diagonal legs does not need to be the same, but it simplifies the discussion, as now we can refer to D as the bond dimension of the PEPS. The diagonal leg at Eq. 2.27 is called the physical index, while the rest are virtual indices. To build the PEPS we have to contract the virtual indices with the corresponding one of each neighbor tensor. The diagramatic representation of a PEPS would be

$|\Psi\rangle =$ 
(2.28)

Observe that for the tensors at the edge of the structure, we can simply delete the index that would have dimension 1. In the same way as we talked about MPS and MPO, we can talk about PEPS and PEPO, which stands for *Projected entangle-pair operator*. They can be defined as PEPS, but this time taking the elemental tensors with 6 indices, 2 physical and 4 virtual. The proof would be most tedious, but it is clear that one can extend Proposition 2.4.7 to the contraction of a PEPS with a PEPO. If the bond dimensions are D_1 and D_2 respectively, the resulting PEPS will have bond dimension $D_1 D_2$.

We will not be using these constructions until later, but this has been a very successful tool in representing ground states of 2D lattices or even higher dimensions. Those are constructed by considering the natural graph given by the vertices and edges of the lattice at play.

At the end of Section 2.3 we wanted to obtain approximations of $Z(\beta \mid \sigma_1, \dots, \sigma_i)$. The following is the explanation given in Appendix A of Frías Pérez et al. (2023), that we repeat here for convenience. Let us focus first on the partition function without conditioning the value of any spin. It is given by

$$Z(\beta) = \sum_{\omega \in \Omega} e^{-\beta H(\omega)} = \sum_{\omega \in \Omega} e^{-\beta \sum_{\langle i,j \rangle \in E} \phi_{ij}(\sigma_i, \sigma_j)} = \sum_{\omega \in \Omega} \prod_{\langle i,j \rangle \in E} W_{ij}(\sigma_i, \sigma_j). \tag{2.29}$$

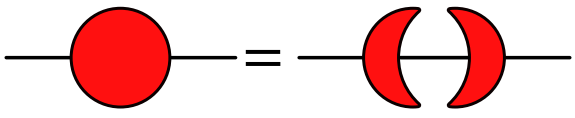
This last expression very much resembles that of a contraction if we consider W_{ij} as a matrix of coefficients $W_{ij}(\sigma_i, \sigma_j)$. For example, if the external field is null, these matrices are of the form

$$W_{ij} = \begin{pmatrix} e^{\beta J_{ij}} & e^{-\beta J_{ij}} \\ e^{-\beta J_{ij}} & e^{\beta J_{ij}} \end{pmatrix}. \tag{2.30}$$

In this way, the edges of our Ising model can be associated a tensor. To be able to fix the spin value and reuse the tensor network, we would like the tensors to be associated with the vertices of the original lattice instead. To that end, we use the Singular Value Decomposition Theorem. This is of paramount importance, as it allows us to manipulate the structure of the network, and by truncating singular values, we can control de size of the tensors as well. Let i be a vertex, and $e(i)$ its right-side neighbor. Given a tensor $W_{ie(i)}$ we take its SVD decomposition $W_{ie(i)} = U_{ie(i)} \Sigma_{ie(i)} V^{\dagger}_{ie(i)}$. As the singular values matrix is diagonal and positive, we can denote by $\sqrt{\Sigma}_{ie(i)}$ the pointwise square root and define $L_i := U_{ie(i)} \sqrt{\Sigma}_{ie(i)}$ and $R_i := \sqrt{\Sigma}_{ie(i)} V^{\dagger}_{ie(i)}$. We have constructed a decomposition

$$W_{ie(i)}(\sigma, \sigma') = \sum_{\nu=1}^{2} L_i(\sigma, \nu) R_{e(i)}(\nu, \sigma'), \tag{2.31}$$

which in diagrammatic notation can be seen as



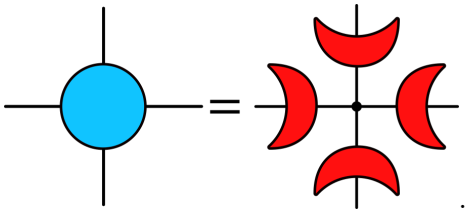
We can do the same for $W_{w(i)i}, W_{in(i)}, W_{s(i)i}$, obtaining

$$W_{w(i)i}(\sigma,\sigma') = \sum_{\nu=1}^{2} L_{w(i)}(\sigma,\nu)\, R_i(\nu,\sigma')$$
$$W_{in(i)}(\sigma,\sigma') = \sum_{\nu=1}^{2} B_i(\sigma,\nu)\, T_{n(i)}(\nu,\sigma')$$
$$W_{s(i)i}(\sigma,\sigma') = \sum_{\nu=1}^{2} B_{s(i)}(\sigma,\nu)\, T_i(\nu,\sigma').$$

Observe how each tensor  is associated with a single vertex. We only need to contract the 4 vertices associated with the same vertex (2 if the vertex is at a corner of the lattice or 3 if it is on the edge) into a single one denoted by A^i, whose elements are given by

$$A^i_{\mu\nu\rho\tau} = \sum_{\sigma=1}^{2} L_i(\sigma,\mu)\, R_i(\nu,\sigma)\, B_i(\sigma,\rho)\, T_i(\tau,\sigma), \tag{2.32}$$

and its graphical representation is



Tensors at corners or edges are constructed in the analogous way taking into account what neighbors they have. The advantage of this representation is that the partition function is still given by the contraction of the network, but if we want to fix the value of a spin σ_i we only need to replace the tensor A^i constructed from a sum in Eq. 2.32 by the corresponding summand associated with σ_i. The whole construction of the new tensor network can be seen in Figure 2.4.

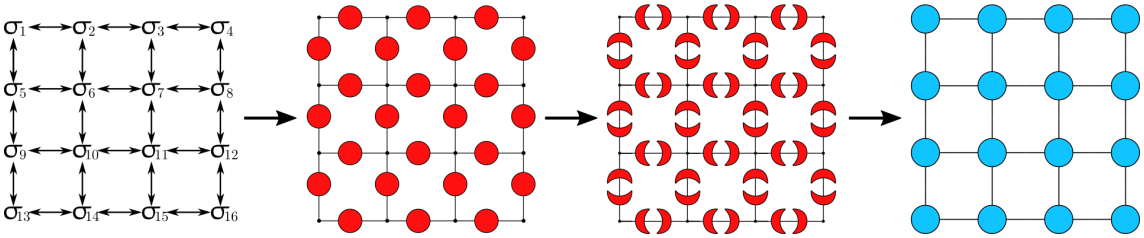


Figure 2.4: Graphical representation of the tensor network construction

To approximate the contraction of an $L \times L$ network, a simple renormalization scheme is used based on the truncation of singular values. If we recall the concept of MPS and MPO, we can see how the row contraction of our networks gives rise to these structures. Let us call $\langle top|$ the top row contraction, which is an MPS; M_k

the contraction of the k^{th} row, $1 < k < L$, which is an MPO; and $\langle bot|$ the bottom row contraction, again another MPS. Our partition function is now given by

$$Z(\beta) = \langle top \mid M_{L-1} ... M_2 \mid bot \rangle \tag{2.33}$$

The bond dimension of these operators is 2, however, Proposition 2.4.7 indicates that it will scale as 2^L. As the computational cost scales linearly with the bond dimension, it is not practical to contract all the MPS and MPO without reducing the bond dimension in between steps. When the bond dimension goes over our chosen bound, we need to reduce it by truncating the singular values of the tensors conforming the resulting MPS. A more detailed explanation can be found in Section 3.4 of (Ran et al., 2020).

This network will have to be contracted every time a partition function needs to be computed having fixed an additional spin value. This has to be done N times. However, only a single tensor will change each time. If we store every intermediate MPS and some additional tensors resulting from the row contractions, we can save a big computational effort in exchange for some memory.

The most important aspect of this section is that we can approximate the contraction of a network with the topology seen in Figure 2.4, where the tensors are associated with a single spin. We will reuse this scheme in the algorithm designed in this project.

One final remark. We have made no comments on how much entanglement information we are losing in each renormalization process. This depends on the bound for the bond dimension we choose, but determining the effects of this value on the contraction error is far from an easy task. It is, however, of great interest to study if we were to analytically treat important aspects of our Markov chain like mixing or correlation times.

Chapter 3

Previous work

In this chapter, we give a brief overview of some numerical methods used in Quantum Field Theories and then explain the work carried out in the project preceding this one.

3.1 Classical and modern approach

The birth of Quantum Field Theories is usually dated in 1927, with the work from Dirac "The quantum theory of the emission and absorption of radiation". Since then, it gained importance during the past century, and various methods for its study have been proposed. One of the most fruitful approaches to Quantum Field Theories is the path integral, developed by Feynman during his PhD thesis (Feynman, 1948).

In most cases, the systems are too complex to study analytically, so numerical methods are the only way to treat these theories. A very common one in Quantum Mechanics is perturbation theory, where in the time-independent approach a Hamiltonian is separated into two terms $H = H_0 + \lambda V$. If we can get an exact solution to H_0, we can study the properties of H as expansions in terms of λ. This method can fail in many scenarios, the simplest of them due to the perturbation being too large. That is why there is a need for non-perturbative methods.

The use of Monte Carlo use took off once computers had enough power to handle large numerical computations. These methods were typically employed in the classical transformation that we saw in Section 1.1.3. There is an ample number of reviews covering the different approaches to Monte Carlo in Lattice Gauge Theories in the past century (Morningstar, 2007; Creutz et al., 1983).

In more recent years other techniques have appeared. Tensor networks have been used to study the properties of the ground state. This is born from the hypothesis that the ground state of some models can be represented with a PEPS (Verstraete et al., 2008). From there, all the renormalization schemes to contract PEPS can be used to either compute the partition function or map the observables we are interested in into these PEPS. Tensor networks have also been combined with Monte Carlo, although not in the same way TNMH does. The idea is to first design a PEPS that represents our system and then map the observable of interest into an integral

in terms of the elements of the tensor network that we can compute with Monte Carlo methods (Zohar and Cirac, 2018).

3.2 Continuous TNMH

As mentioned in the introduction, this project is the continuation of the research carried out last year. To give the reader an idea of the starting point of this year's work, we give an overview of the algorithm developed until July 2023.

Remember we are able to sample with TNMH the Ising model. An extension to a model with any finite number of possible values for the degrees of freedom is nearly immediate. It is also possible to adapt the tensor network to sample spin systems whose interactions do not have the same structure as the Ising model but are of plaquette kind. We would like to take this concept to degrees of freedom in a compact and infinite group. We restrict to compact groups because the properties of Section 2.1 extend without any problem to random variables taking values in compact sets, but it is more delicate when it comes to sets like $\mathbb{R}^n$.

The main idea will be to propose a set of changes on each site and create an associated spin model whose binary values determine whether the change at each site is applied. It can be summarized as follows:

(I) Choose a value m in $\{0, \dots, m_{\max}\}$ with a suitable criteria.

(II) Draw N i.i.d. elements of the gauge group according to some distribution λ_m, where m is controlling the variance and λ_m is centered on the identity. Call these elements the differences $\Psi = \{\Psi_i \sim \lambda_m : i = 1, \dots, N\}$.

(III) Construct a plaquette spin system with N spins, such that $\sigma_i = 1$ has the same effect in the new Hamiltonian as the variable x_i in the original system and $\sigma_i = -1$ is equivalent to $x_i \cdot \Psi_i$.

(IV) Sample a spin system configuration using TNMH.

(V) Apply the changes $x_i \leftarrow x_i \cdot \Psi_i$ suggested by the variables σ_i.

3.2.1 XY model

In our previous work, we implemented these ideas in two different models: the XY model, whose symmetry group is $U(1)$; and the Wilson action, with symmetry group $SU(2)$. We will focus our attention on the latter one, as that is similar to the action we will work with. However, it is worth mentioning the differences in the structure of the hamiltonian of these two systems, as what was learned with XY model will prove useful for future work.

The XY model is studied in a squared lattice, like Ising's. Its hamiltonian is given by

$$H(\omega) = \sum_{\langle i,j \rangle \in E} J_{ij} \cos(\theta_i - \theta_j - A_{ij}). \tag{3.1}$$

As in the Ising model, the degrees of freedom sit on the vertices of the lattice. Once given the differences Ψ, the translation of this kind of model to its Ising equivalent is computationally cheap and simple. It is based on the following lemma (Frías Pérez et al., 2023).

Lemma 3.2.1. *Any function of two binary variables, B, can be expressed as an energy Ising model plus a constant*

$$B(\sigma, \sigma') = J\sigma\sigma' + h\sigma + h'\sigma' + K. \tag{3.2}$$

Proof. It is enough to solve the following linear system

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} J \\ h \\ h' \\ K \end{pmatrix} = \begin{pmatrix} B(+1,+1) \\ B(+1,-1) \\ B(-1,+1) \\ B(-1,-1) \end{pmatrix} \tag{3.3}$$

□

To construct the new Ising model one only needs to solve this system for each edge of the lattice and sum the resulting fields. The constant can be ignored, as it plays no role in the Boltzmann probability. Hence, whenever we have a model with degrees of freedom sitting on the vertices and interaction to nearest neighbors, we are able to reuse TNMH. This proved to be possible in last year's results (Díaz Rodríguez, 2023).

3.2.2 Wilson action

We now turn our attention to the Wilson model. Before jumping to what the action looks like, we will make a comment on the notation we will be employing. Our model sits on a D-dimensional euclidean lattice. Each point in the lattice also referred to as a vertex, will be indexed by a natural number x. Directions vectors in the lattice will be represented by letters μ and ν and they correspond to the canonical directions in $\mathbb{R}^D$. The correspondence between D-dimensional coordinates and the indexing of the vertices is not important, but the vertex $x + \hat{\mu}$ should be understood as the one corresponding to moving one position in the μ direction from vertex x. Our model variables may sit on a vertex, then labeled by x, or on the edge between two vertices x and $x + \hat{\mu}$, labeled by x, μ.

Let us now move on to the Wilson action. For now, we will restrict to dimension 2 and only consider degrees of freedom sitting on the edges of the lattice. The variables belong to the gauge group $SU(2)$. The Wilson action reads

$$S = -\frac{\beta}{2} \sum_{x=1}^{N} \mathrm{Re}\,\mathrm{Tr}\left[U_{x,\mu}\, U_{x+\hat{\mu},\nu}\, U^{\dagger}_{x+\hat{\nu},\mu}\, U^{\dagger}_{x,\nu}\right], \tag{3.4}$$

where μ and ν are the two directions on the lattice plane.

The study was done under open boundary conditions, so in reality, the sum is not taken in all vertices, and only in those that sit at the bottom left of a plaquette. We ignore this in the action given to simplify the notation.

Let us be more specific with steps II to V from the algorithm given before. Comments on step I will be made in the next section.

To draw configuration differences we need to obtain random independent matrices in $SU(2)$. The following is a scheme to produce random matrices in $SU(N)$.

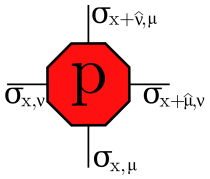
1. Draw a unitary matrix $W \in U(N)$ distributed according to the Haar measure (Mezzadri, 2007).
2. Draw $N-1$ phases λ_i uniformly in $[-\alpha/2, \alpha/2]$. The parameter $\alpha \in (0,1]$ allows us to control how far from the identity we are expected to draw the next matrices. We choose $\alpha = 2^{-m}$, where m is the value given by step (I).
3. The last phase λ_N is taken as $\lambda_N = \left\lfloor \sum_{j=1}^{N-1} \lambda_j \right\rfloor - \sum_{j=1}^{N-1} \lambda_j$.
4. Last, obtain the matrix

$$U = W \begin{pmatrix} e^{2\pi\lambda_1 i} & & \\ & \ddots & \\ & & e^{2\pi\lambda_N i} \end{pmatrix} W^\dagger. \tag{3.5}$$

We now construct the associated spin model. Let Ψ be the set of matrices $\Psi_{x,\mu}$ drawn as described. Start by considering a set of spins Σ in each edge of the lattice and define $W_{x,\mu}(\sigma_{x,\mu}) := \frac{1+\sigma_{x,\mu}}{2} U_{x,\mu} + \frac{1-\sigma_{x,\mu}}{2} U_{x,\mu}\Psi_{x,\mu}$. Then, we can define the action

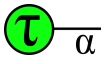
$$H_S(\Sigma) = -\frac{\beta}{2} \sum_{x=1}^{N} \operatorname{Re}\operatorname{Tr}\left[W_{x,\mu}(\sigma_{x,\mu})\, W_{x+\hat{\mu},\nu}(\sigma_{x+\hat{\mu},\nu})\, W_{x+\hat{\nu},\mu}(\sigma_{x+\hat{\nu},\mu})^\dagger\, W_{x,\nu}(\sigma_{x,\nu})^\dagger \right]. \tag{3.6}$$

In Section 2.3 we studied how to contract tensor networks where spins lie on the vertices. To reuse those structures we define the following tensors:

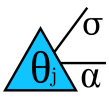


$$= \exp\left\{ \frac{\beta}{2} \operatorname{Re}\operatorname{Tr}\left[W_{x,\mu}(\sigma_{x,\mu})\, W_{x+\hat{\mu},\nu}(\sigma_{x+\hat{\mu},\nu})\, W_{x+\hat{\nu},\mu}(\sigma_{x+\hat{\nu},\mu})^\dagger\, W_{x,\nu}(\sigma_{x,\nu})^\dagger \right] \right\},$$

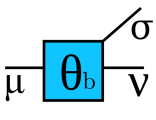
to which we refer as *plaquette tensors*.



$= 1\ \forall\alpha$



$$= \begin{cases} 1 & \text{if } \sigma = \alpha \\ 0 & \text{if } \sigma \neq \alpha \end{cases}$$



$$= \begin{cases} 1 & \text{if } \sigma = \mu = \nu \\ 0 & \text{otherwise} \end{cases}$$

which are simply auxiliary tensors. With these definitions, we can build the tensor network seen in Figure 3.1. Observe how blue tensors allow us to fix the value of the spin that is common in the red tensors associated with each plaquette. The blue diagonal legs correspond to the spins that we need to sum over to compute the partition function.

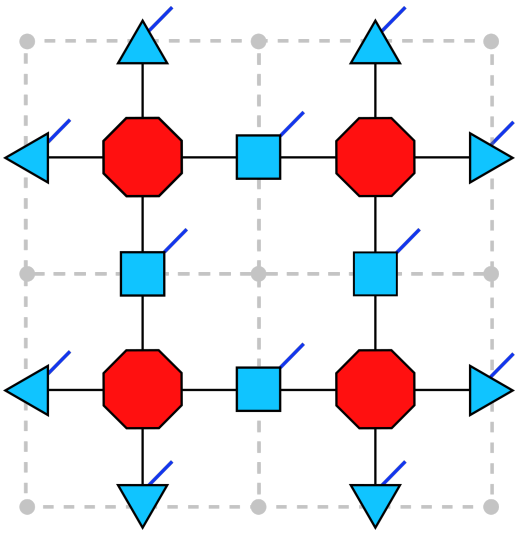


Figure 3.1: Tensor network for the Wilson action on a 3×3 lattice

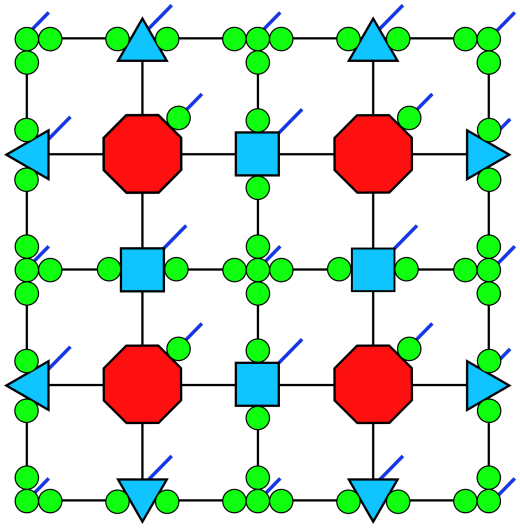


Figure 3.2: Adaptation of a Wilson action Tensor Network to those used in the spin model

We employ the green tensors to give the tensor network a structure similar to the one we worked with in the Ising model. In Figure 3.2 we add these transparent tensors, where glueing tensors together corresponds to a new tensor created by multiplying the coefficients. This is, glueing T_α with S_β is a new tensor of 2 indices $R_{\alpha,\beta} = T_\alpha \cdot S_\beta$. If a diagonal leg points out of a green tensor, it only has one possible value. Moreover, their inclusion does not change the value computed by the contraction. Hence, these fictional spins do not play any role, other than allowing to reuse the renormalization schemes employed in the Ising model.

Once we have the new spin configuration, we can compute the equivalent Wilson configuration by applying the change $U_i \leftarrow U_i \Psi_i$ when $\sigma_i = -1$. Let $\mathbf{1}$ be a spin configuration with every spin $\sigma_i = 1$ and Σ the discrete configuration proposed by the tensor network with probability $\pi_{TN}^{(\beta,\kappa)}(\Sigma \mid \omega, \Psi)$. We accept said change with probability

$$P_{acc}(\omega \to \omega' \mid \Psi) = \min\{1, \frac{\pi_{TN}^{(\beta,\kappa)}(\Sigma \mid \omega, \Psi)}{\pi_{TN}^{(\beta,\kappa)}(\mathbf{1} \mid \omega, \Psi)} \times \frac{\pi^{(\beta,\kappa)}(\omega')}{\pi^{(\beta,\kappa)}(\omega)}\}. \tag{3.7}$$

3.2.3 Amplitude choice

The approach described presents a problem not seen in the finite case: choosing the amplitude of the changes proposed at each step.

If we are sitting in a low energy state, random changes are likely to cause an increase in energy. If the energy is to increase, the larger the energy difference between proposed configurations the lower the Metropolis probability of acceptance. This suggests that small amplitudes in the proposed changes will be beneficial toward acceptance rates. As we are using Metropolis-Hastings acceptance probability, the accuracy of the approximation $\widetilde{\pi}(\omega)$ also determines the acceptance rates. However,

we cannot always rely on high accuracy of such computations, as we have seen how its cost scales badly with the system size.

We are interested in sampling as much of the state space as possible, and small changes in the configurations suppose a very slow sweeping of the overall configurations set. This is in conflict with the previous reasoning, so we have to decide on a criterion to determine what an *optimal* amplitude is and how to choose it at each step. We do this dynamically using the statistics of the running sampling process. More about this in the next section.

3.2.4 Previous results

We now present the main results that are of importance for the current project.

First of all, the described chain is irreducible and aperiodic, and $\pi^{(\beta,\kappa)}(\omega)$ is reversible. Irreducibility and aperiodicity are now seen in terms of $\rho_{pick}(\omega \to \omega') > 0$, where ρ_{pick} is the density of probability of the change $\omega \to \omega'$ being proposed. Proving this implies that $\pi^{(\beta,\kappa)}(\omega)$ is the unique stationary distribution.

Let us denote by $F(\Sigma, \omega, \Psi)$ the transformation of the configuration ω when applying the changes Ψ according to some spin configuration Σ. Given some set of differences Ψ drawn with probability density $\lambda(\Psi)$, the probability of proposing the change $\omega \to \omega'$ is

$$P_{pick}(\omega \to \omega' \mid \Psi) = \sum_{\omega:\, F(\Sigma,\omega,\Psi)=\omega'} \pi_{TN}^{(\beta,\kappa)}(\Sigma \mid \omega, \Psi). \tag{3.8}$$

We can calculate the probability density without conditioning on Ψ as

$$\rho_{pick}(\omega \to \omega') = \sum_{\Sigma \in \{1,-1\}^{|V|}} \int_{D\Psi} \pi_{TN}^{(\beta,\kappa)}(\Sigma \mid \omega, \Psi)\, \lambda(\Psi)\, \delta(\omega' - F(\Sigma, \omega, \Psi))\, d\Psi, \tag{3.9}$$

where the domain $D\Psi$ is $SU(N)^{|V|}$. We are taking $\delta(\omega' - F(\Sigma, \omega, \Psi))$ as the measure in $SU(N)^{|V|}$ centered at 0 defined as

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

The value at Eq. 3.9 is positive because: (i) we take a distribution λ with positive value for every Ψ, (ii) the set $\{\Psi \in SU(N)^{|V|} : \exists\, \Sigma \in \{1,-1\}^{|V|} \colon \omega' = F(\Sigma, \omega, \Psi)\}$ is non-empty, (iii) we assume our approximation $\pi_{TN}^{(\beta,\kappa)}$ is precise enough to be positive for all Σ. As every configuration ω' is susceptible to being proposed from any configuration ω we conclude that the chain is aperiodic and irreducible.

To prove that $\pi^{(\beta,\kappa)}$ is reversible we check the detailed balance condition. Given an initial configuration ω and a set of differences Ψ, this defines an associated spin model. If $\pi_{\omega,\Psi}^{(\beta,\kappa)}$, $\pi^{(\beta,\kappa)}$ are the Boltzmann distributions and Z_S, Z_W the partition functions for the spin model and the Wilson action respectively, it follows that

$$\pi_{\omega,\Psi}^{(\beta,\kappa)}(\Sigma) = \pi^{(\beta,\kappa)}(F(\Sigma, \omega, \Psi)) \times Z_W/Z_S. \tag{3.11}$$

Let $T_{\omega,\Psi}(\Sigma_1 \to \Sigma_2)$ be the transition probability from one spin configuration Σ_1 to another Σ_2 of the Markov chain step defined by fixing Ψ and ω. The reversibility of a chain of this kind for the spin model is already proven in Frías Pérez et al. (2023), hence we can start with

$$\pi^{(\beta,\kappa)}_{\omega,\Psi}(\Sigma_1)\, T_{\omega,\Psi}(\Sigma_1 \to \Sigma_2) = \pi^{(\beta,\kappa)}_{\omega,\Psi}(\Sigma_2)\, T_{\omega,\Psi}(\Sigma_2 \to \Sigma_1)\,. \tag{3.12}$$

From the definition of the accepting probability,

$$T_{\omega,\Psi}(\Sigma_i \to \Sigma_j) = T(\omega_i \to \omega_j \mid \Psi)\,, \tag{3.13}$$

where $\omega_k = F(\Sigma_k, \omega, \Psi)$. Putting these last three equations together

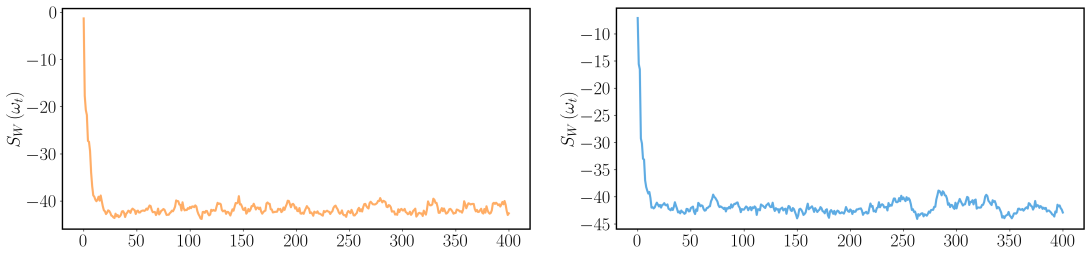
$$\pi^{(\beta,\kappa)}(\omega)\, T(\omega \to \omega' \mid \Psi) = \pi^{(\beta,\kappa)}(\omega')\, T(\omega' \to \omega \mid \Psi) \tag{3.14}$$

The detailed balance condition is obtained by integrating both sides with the suitable δ measure and multiplying both sides by the distribution of Ψ.

As mentioned before, different strategies have to be tested to choose the best amplitude at each step. The strategy used at the end of last year's project can be described as follows:

1. Choose an interval $[0, \dots, m_{max}]$ in which m will take integer values.
2. At each step compute the difference between configurations after accepting or rejecting the proposed change. This is done by summing the distance between each matrix's sites in Frobenius norm.
3. The m value for the next step is chosen with a probability proportional to the mean distance resulting from using each m value in previous steps.

This strategy presented a few drawbacks, like having to spend the first few steps using a random m to gather some data before mean values could be computed. Although it promoted state space exploration, it was very susceptible to the outcomes of the first steps and made convergence times even harder to determine from execution to execution. In the next chapter, we will comment on two new methods used that seem to perform better and with higher stability.



(a) Metropolis (b) TNMH

Figure 3.3: Energy of the Wilson model in a 8×8 lattice with $\beta = 10$ and bond dimension $D = 8$.

The parameter used to determine the performance was the convergence time, sometimes referred to as burning time. This was done by simply plotting observables and eyeballing the time of transitioning from an unstable phase to a stable one. As we said before, metastability can invalidate this simple procedure, but it was enough for the objectives proposed.

On July 2023 we had an algorithm with a convergence time similar to Metropolis, as we can see in the two graphs in Figure 3.3.

Chapter 4

Monte Carlo simulation of lattice gauge theories

In this chapter we present the bulk of our work. We have extended our previous algorithm to work with a more general action. First, we have made a deeper study of two-dimensional LGT. Next, we have moved up to three dimensions. We will review here the tensor network construction and the performance for both settings. In the last section, some brief comments on the computational effort, hardware used and implementation are given.

4.1 Action

In the previous chapter, we have studied the Wilson action, which models the behavior of a gauge field. Now we are going to extend it to what is known as the Higgs model, by adding a scalar kinetic term in representation of matter elements. By adding matter to the system and with $SU(2)$ as the gauge group, this new action is related to the weak nuclear force (Montvay and Münster, 1994). This addition includes a new parameter κ sometimes referred to as the *hopping distance*, which plays a similar role as the known inverse of the temperature β. Let ω englobe all degrees of freedom, i.e. ω is a configuration. The action includes three terms. The first is the standard Wilson action, which only depends on the gauge field

$$\beta S_g(\omega) = -\frac{\beta}{2} \sum_{x=1}^{N} \sum_{\mu<\nu=1}^{D} \mathrm{Re}\,\mathrm{Tr}\left[U_{x,\mu}\, U_{x+\hat{\mu},\nu}\, U^{\dagger}_{x+\hat{\nu},\mu}\, U^{\dagger}_{x,\nu}\right]. \tag{4.1}$$

The second, the gauge-matter term, corresponds to the scalar term in the Yang-Mills theory

$$\kappa S_{\phi}(\omega) = -\frac{\kappa}{2} \sum_{x=1}^{N} \sum_{\mu=1}^{D} \rho_x\, \rho_{x+\hat{\mu}} \mathrm{Re}\,\mathrm{Tr}\left[\alpha^{\dagger}_x\, U_{x,\mu}\, \alpha_{x+\hat{\mu}}\right]. \tag{4.2}$$

The third is to the scalar potential, and only depends on the matter field

$$W(\omega) = \sum_{x=1}^{N} \rho_x^2 + \lambda \left(\rho_x^2 - 1\right)^2. \tag{4.3}$$

The variables $\alpha_x \in SU(2)$ and $\rho_x \in \mathbb{R}_{\geq 0}$, known as the Goldstone mode and the Higgs mode respectively; come from a 2×2 hermitian matrix ϕ_x which can be expressed as $\phi_x = \rho_x \alpha_x$. This matrix at the same time is a reexpression of the actual complex scalar field. We will not dive into the physics at play here, but a detailed explanation can be found in Section 6.1 of Montvay and Münster (1994).

The partition function, which will act as the normalization factor of the distribution is given by

$$Z(\beta, \kappa) = \int D[U]\, D[\rho]\, D[\alpha]\, e^{-\beta S_g(\omega) - \kappa S_\phi(\omega) - W(\omega)}, \tag{4.4}$$

where the integration over U and α is done with the Haar measure in $SU(2)$ and the integration over ρ is given by the measure $\rho_x^3 d\rho_x$ over $[0, \infty)$ and $d\rho_x$ the usual Lebesgue measure.

The expression of the gauge-matter term at Eq. 4.2 suggests that we can express the action in terms of gauge invariant variables: the Higgs mode ρ_x and the gauge invariant link variables given by the map

$$U_{x,\mu} \mapsto V_{x,\mu} = \alpha_x\, U_{x,\mu}\, \alpha^\dagger_{x+\hat{\mu}}. \tag{4.5}$$

If we apply this change to every matrix we can check how the terms in S_g do not change.

$$\begin{aligned}
&\mathrm{Tr}\left[V_{x,\mu}\, V_{x+\hat{\mu},\nu}\, V^\dagger_{x+\hat{\nu},\nu}\, V^\dagger_{x,\nu}\right] = \\
&\mathrm{Tr}\left[\left(\alpha_x U_{x,\mu} \alpha^\dagger_{x+\hat{\mu}}\right)\left(\alpha_{x+\hat{\mu}} U_{x+\hat{\mu},\nu} \alpha^\dagger_{x+\hat{\mu}+\hat{\nu}}\right)\left(\alpha_{x+\hat{\nu}} U_{x+\hat{\nu},\nu} \alpha^\dagger_{x+\hat{\mu}+\hat{\nu}}\right)^\dagger\left(\alpha_x U_{x,\nu} \alpha^\dagger_{x+\hat{\nu}}\right)^\dagger\right] = \\
&\mathrm{Tr}\left[\alpha_x U_{x,\mu} \alpha^\dagger_{x+\hat{\mu}} \alpha_{x+\hat{\mu}} U_{x+\hat{\mu},\nu} \alpha^\dagger_{x+\hat{\mu}+\hat{\nu}} \alpha_{x+\hat{\mu}+\hat{\nu}} U^\dagger_{x+\hat{\nu},\nu} \alpha^\dagger_{x+\hat{\nu}} \alpha_{x+\hat{\nu}} U^\dagger_{x,\nu} \alpha^\dagger_x\right] = \\
&\mathrm{Tr}\left[U_{x,\mu}\, U_{x+\hat{\mu},\nu}\, U^\dagger_{x+\hat{\nu},\nu}\, U^\dagger_{x,\nu}\right].
\end{aligned} \tag{4.6}$$

However, in the gauge-matter term, we do get a simplification by removing the Goldstone modes from the action as

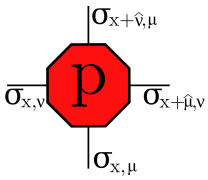
$$S_\phi(\omega) = -\frac{1}{2} \sum_{x=1}^{N} \sum_{\mu=1}^{D} \mathrm{Re}\,\mathrm{Tr}\left[\alpha^\dagger_x\, V_{x,\mu}\, \alpha_{x+\hat{\mu}}\right] = -\frac{1}{2} \sum_{x=1}^{N} \sum_{\mu=1}^{D} \mathrm{Re}\,\mathrm{Tr}\, U_{x,\mu}. \tag{4.7}$$

Hence, our only variables are the matrices on the edges and the non-negative scalars on the vertices. For this study, however, we are going to freeze the matter terms by fixing $\rho_x = 1$ at every site. Fixing the value of ρ allows us not to worry about the parameter λ. W will become a constant, hence we can forget about it when computing the action during the sampling process. From now on, we only work with the $SU(2)$ link variables and the terms S_g and S_ϕ of the action.

4.2 2D

The degrees of freedom are the same as those considered in Chapter 3, hence the tensor network construction will be very similar. The only modification is in the plaquette tensors caused by the gauge-matter contribution. For convenience, we reproduce the definitions from Chapter 3. We denote by $\Psi = \{\Psi_{x,\mu}\}$ the set of differences drawn. $W_{x,\mu}(\sigma_{x,\mu}) = \frac{1+\sigma_{x,\mu}}{2} U_{x,\mu} + \frac{1-\sigma_{x,\mu}}{2} U_{x,\mu}\Psi_{x,\mu}$ are the functions depending on a binary value that either apply or not such change. To simplify the presentation, we will omit the argument of the function $W_{x,\mu}$ in the following expressions. Σ is the set of $\sigma_{x,\mu}$ values that are related with artificially created spin system.

The only tensors that change are the plaquette tensors, now defined as



$$= \exp\left\{ \begin{array}{c} \frac{\beta}{2}\mathrm{Re\,Tr}\left[W_{x,\mu}W_{x+\hat{\mu},\nu}W^{\dagger}_{x+\hat{\nu},\mu}W^{\dagger}_{x,\nu}\right] \\ + \\ \frac{\kappa}{2}\mathrm{Re\,Tr}\left[c_{x,\mu}W_{x,\mu} + c_{x+\hat{\mu},\nu}W_{x+\hat{\mu},\nu} + c_{x+\hat{\nu},\mu}W_{x+\hat{\nu},\mu} + c_{x,\nu}W_{x,\nu}\right] \end{array} \right\},$$

where the coefficients $c_{x,\mu}$ are either 1 or 1/2 depending on whether the edge (x,μ) is on the boundary of the lattice or not. This is just a way of not summing a different amount of times the trace of matrices depending on their position on the lattice. With this change, we can repeat the setup we constructed in last chapter and begin sampling with the new action.

We now address the behavior of the algorithm in the 2-dimensional lattice. We will contrast observable mean values against those obtained by Bazavov et al. (2019) as well as compare mixing and autocorrelation times with the Metropolis algorithm.

We computed three observables: (i) the average plaquette $\langle p\rangle$, (ii) the expectation value of the gauge-matter term $\langle L_\phi\rangle$, and (iii) the susceptibility of L_ϕ, denoted by χ_κ. Let V be the number of vertices of the system, these observables are given by

$$\langle p\rangle = \frac{1}{V}\left\langle \frac{1}{2}\sum_{x=1}^{N}\sum_{\mu<\nu=1}^{D}\mathrm{Re\,Tr}\left[U_{x,\mu}\,U_{x+\hat{\mu},\nu}\,U^{\dagger}_{x+\hat{\nu},\mu}\,U^{\dagger}_{x,\nu}\right]\right\rangle \tag{4.8}$$

$$\langle L_\phi\rangle = \frac{1}{V}\left\langle \frac{1}{2}\sum_{x=1}^{N}\sum_{\mu=1}^{D}\mathrm{Re\,Tr}\,U_{x,\mu}\right\rangle \tag{4.9}$$

$$\chi_\kappa = V\left\langle \left(L_\phi - \langle L_\phi\rangle\right)^2\right\rangle, \tag{4.10}$$

where in the last equation we are taking L_ϕ as

$$L_\phi = \frac{1}{2V}\sum_{x=1}^{N}\sum_{\mu=1}^{D}\mathrm{Re\,Tr}\,U_{x,\mu}. \tag{4.11}$$

The computations have been done taking 5 independent executions with 15000 steps each. The first 2000 steps are discarded, giving enough time for the chain

to reach equilibrium. Mean values are computed by taking a sample every $\widetilde{\tau}$ steps, where $\widetilde{\tau}$ has been taken as the time t for which $\rho_{\mathcal{O}}(t) = 1/e$. This choice is motivated by the behavior of $\rho_{\mathcal{O}}$ being similar to a decaying exponential $e^{-t/\tau_{exp,\mathcal{O}}}$.

The parameters chosen follow those from the comparing literature, where κ is taken from 0 to 2 and the ratio β/V is kept at a constant $c = 0.01$. We find agreement in the values obtained for both the average plaquette and the gauge-matter term, which can be seen in Figures 4.1 and 4.2.

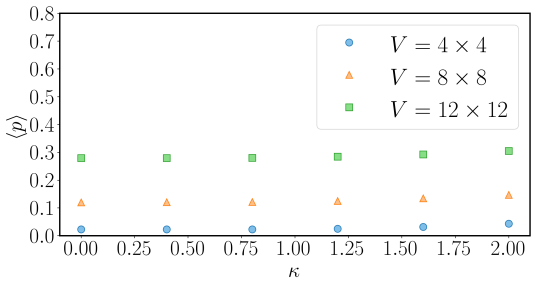


Figure 4.1: Average plaquette for different system size and κ, keeping the ratio $\beta/V = 0.01$.

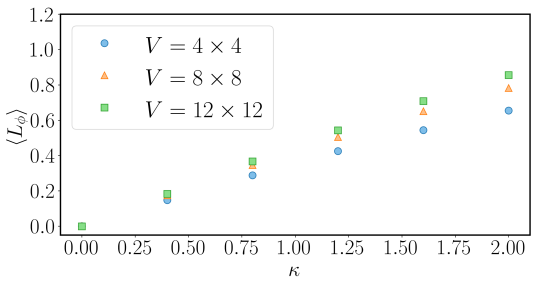


Figure 4.2: Average gauge-matter term for different system size and κ, keeping the ratio $\beta/V = 0.01$.

We do find some differences in our susceptibility results in Figure 4.3. We suspect this is due to the boundary conditions playing a more important role in the susceptibility. The article from Bazavov et al. (2019) uses periodic boundary conditions, while we have chosen to sample with open boundary conditions. In small system sizes this can have a higher impact. Still, the tendency of the curves is similar to that shown in the literature.

We observe that, once the plaquette tensors have been evaluated, the rest of the computations are independent of the group in which the gauge invariant matrices live. Physicists are not only interested in this model with $SU(2)$ as gauge group, but with other special unitary groups $SU(N)$ as well. For instance, the same action but using $SU(3)$ as link variables is related to the strong nuclear force. Other

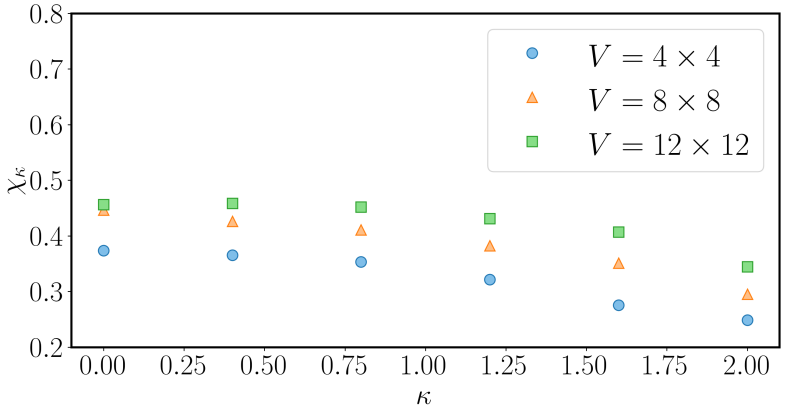
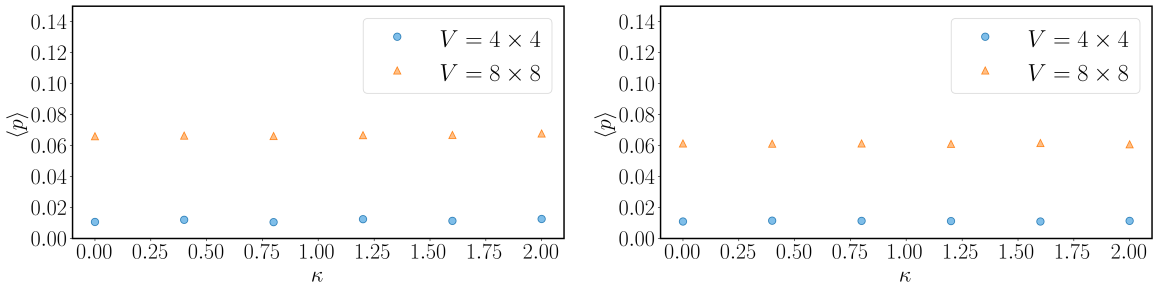


Figure 4.3: Susceptibility of L_ϕ for different system size and κ, keeping the ratio $\beta/V = 0.01$.

theories require working with larger values of N. With our current design, there is no additional effort to change N and observe how the new systems behave. We have done so for values $N = 3, 5, 10$ and worked with the same parameters as before.

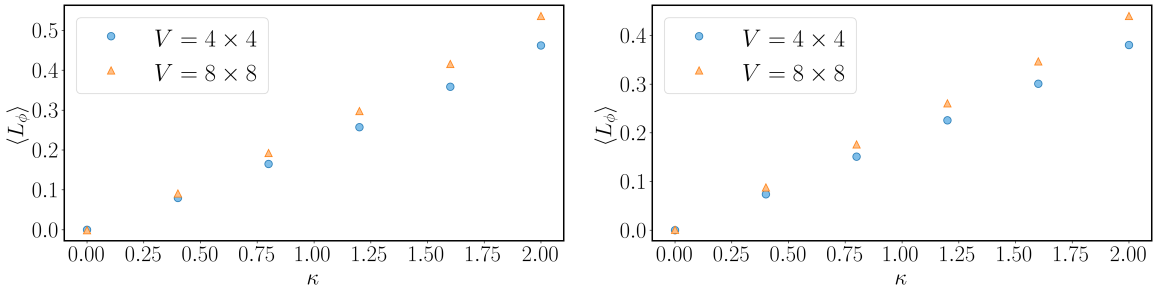
We begin by showing the average plaquette for values $N = 3$ in Figure 4.4a and $N = 10$ in Figure 4.4b. Again, they present an almost constant value when varying κ as we observed with $SU(2)$. The results for $SU(5)$ are very similar to the ones shown. We have not run executions with $V = 12^2$ because they are quite expensive and we opted to employ the computer power for different settings.



(a) $SU(3)$ (b) $SU(10)$

Figure 4.4: Average plaquette for two different groups with $\beta/V = 0.01$ and different values of κ

In Figures 4.5a and 4.5b we can see how the gauge-matter term behaves similarly for both $SU(3)$ and $SU(5)$ respectively. The values vary, but the behavior is the same. We find the same pattern even for $SU(10)$.



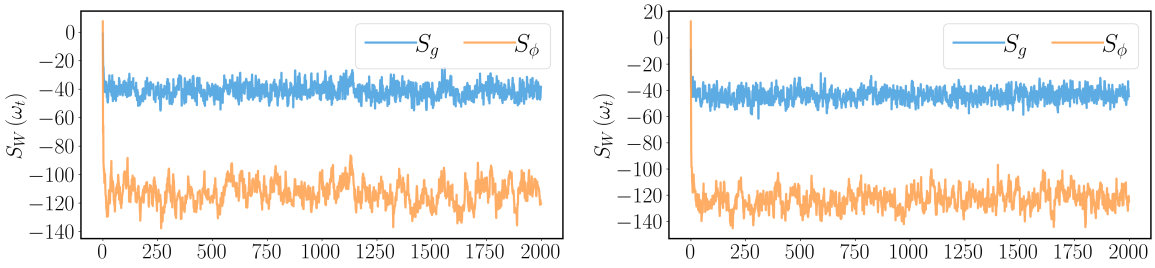
(a) $SU(3)$ (b) $SU(5)$

Figure 4.5: Gauge-matter term for two different groups with $\beta/V = 0.01$ and different values of κ

The susceptibility results are a bit less clear. We can appreciate a good amount of noise that does not allow us to draw further conclusions. Moreover, boundary conditions might play a part in the graphs and increase the gap with the literature chosen. As we have not found literature employing open boundary conditions, we will stay with obtaining similar results with the two first observables and similar behavior with the susceptibility for $SU(2)$ and seeing the pattern propagated to bigger groups with the average plaquette and gauge-matter term.

Let us now divert our attention to the chain performance. We already saw that bounding mixing times is a hard problem for Markov chains. There exist general exponential bounds, but they depend on a value τ_{mix} that is usually unknown. Hence, we judge convergence times by identifying the burn-in period in observable graphs. We have not observed any indication of metastability appearing in TNMH chains. However, all we can say is that we deem them highly unlikely to appear, as no proof of their nonexistence has been sought.

Our findings are that the convergence of TNMH is similar to that displayed by the Metropolis algorithm. As an example, in Figure 4.6 we represent S_g and S_ϕ for a system of size 12^2. The time it takes to observe a relatively uniform behavior may vary between executions, of course, but the general observation is that of similar performance in both cases.



(a) Metropolis (b) TNMH

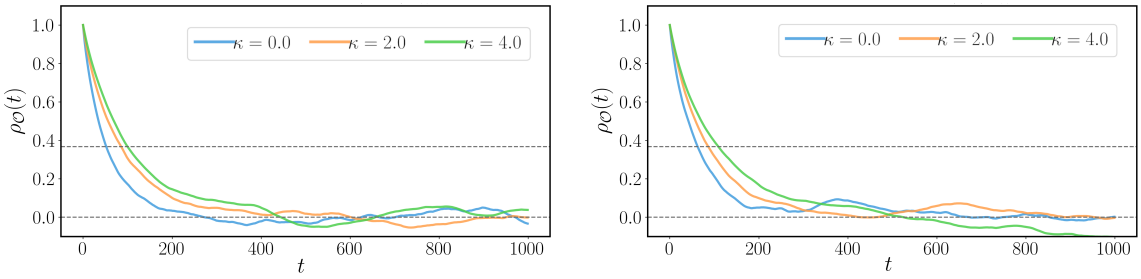
Figure 4.6: Energy of each part of the action of a 12×12 lattice and parameters $\beta = 1.44$ and $\kappa = 2.0$

Results are not as bright as with convergence if we focus on autocorrelation times. Again, we estimate those by taking the time t for which the function $\rho_{\mathcal{O}}(t)$ reaches $1/e$. The finding is that the Metropolis algorithm tends to have slightly shorter autocorrelation times, as we can see in Figure 4.7. It has been marked with a horizontal grey line values $y = 1/e$ and $y = 0$. The value $y = 0$ is important because the function is supposed to be positive, and anything going below it is the result of having noise caused by a lack of samples. The observable used for these computations is L_ϕ, as Bock et al. (1990) hold it is the observable where critical points are easier to find, hence critical slowing down might be most visible.

It is important to note that computing these quantities requires plenty of data, having used for these graphs 5 independent chains with 15000 steps each and discarding the first 2000 steps. Even with this amount of steps we can observe a high amount of noise in the tail of the function. As we are interested in the time t for which the function crosses the top grey line we have considered this amount of noise tolerable, as the function seems to be dominated by non-noisy values at lower times.

Our initial idea was that TNMH would be a better algorithm to obtain uncorrelated samples in fewer steps. Although this method might not be the most accurate to determine the autocorrelation time, it gives us enough information to discard our initial hypothesis. We even used high values for β, where local changes are known to suffer more. Let us discuss the reasons why TNMH is not outperforming Metropolis.

Heuristically, one can identify in TNMH how uncorrelated two configurations are



(a) Metropolis (b) TNMH

Figure 4.7: Normalized autocorrelation function for the gauge-matter term on a 4×4 lattice and variables in $SU(3)$. The parameters used where $\beta = 10$ and κ ranging from 0 to 4.

with how many changes $\Psi_{x,\mu}$ we have applied and how far from the identity those are. We already discussed that low values for m make $\Psi_{x,\mu}$ be far from the identity, and both algorithms do use similar ones. The amount of changes that are applied is conditioned by the following: (i) the change $\omega \to \omega'$ needs to be accepted and (ii) the spin configuration drawn needs to have as many spins $\sigma = -1$ as possible. Condition (i) is related to the Metropolis-Hastings probability

$$P_{acc}^{TN}\left(\omega \to \omega' \mid \Psi\right) = \min\{1, \frac{\pi_{TN}^{(\beta,\kappa)}\left(\Sigma \mid \omega, \Psi\right)}{\pi_{TN}^{(\beta,\kappa)}\left(\mathbf{1} \mid \omega, \Psi\right)} \times \frac{\pi^{(\beta,\kappa)}\left(\omega'\right)}{\pi^{(\beta,\kappa)}\left(\omega\right)}\}, \tag{4.12}$$

which, we have checked that even for low values of the bond dimension, is close to 1. This is possible because the system sizes are relatively small and the tensor network renormalization scheme is still able to give good approximations of the partition function. With respect to condition (ii), the amount of spin $\sigma = -1$ will only be determined by the probabilities $Z\left(\beta, \kappa|\sigma_1, \ldots, \sigma_i\right) / Z\left(\beta, \kappa|\sigma_1, \ldots, \sigma_{i-1}\right)$.

In the Metropolis algorithm, the analysis of how many changes we accept after a full sweep of the lattice is a bit different. We consider a single spin change each time. Moreover, that spin change is always from +1 to -1, i.e. from not applying the change $\Psi_{x,\mu}$ in ω to applying it. Let ω' be a state such that every site is equal to that in ω but for a variable $U'_{x,\mu} = U_{x,\mu}\Psi_{x,\mu}$. We accept this change with Metropolis probability

$$P_{acc}^{M}\left(\omega \to \omega' \mid \Psi\right) = \min\{1, \frac{\pi^{(\beta,\kappa)}\left(\omega'\right)}{\pi^{(\beta,\kappa)}\left(\omega\right)}\}. \tag{4.13}$$

Note that, in this case, every time $\omega \to \omega'$ is accepted, we are automatically applying the change $\Psi_{x,\mu}$, while in TNMH accepting a change where the spin value is +1 makes no difference. We observed that, despite having a higher acceptance probability in TNMH than the mean Metropolis acceptance probability along a full sweep, the number of changes applied with the first algorithm is still lower on average.

This is not an argument that shows an advantage of Metropolis over TNMH, but rather suggests that the asymmetry in how the changes are accepted can benefit one algorithm or the other depending on the setting we are in. TNMH is better suited for

systems with very complex energy landscapes and this 2-dimensional simplified Higgs model is not the appropriate model where the algorithm can outperform Metropolis.

On a final note, we have constructed an algorithm that although it does not perform as we expected in the 2-dimensional regime, yields results that are coherent with the literature.

4.3 3D

State of the art simulations are done in 4-dimensional euclidean lattices, so we are interested in moving past our 2-dimensional lattice. In this section we explain how we have scaled the algorithm to three dimensions and evaluate its performance and limitations.

4.3.1 Tensor structure and contraction

Let us focus on the tensor network construction, as the dimension of the lattice is transparent for the Metropolis-Hastings part of the algorithm. The lattice now has the structure of a cube. In Figure 4.8 we can see an example of a $4 \times 4 \times 4$ cubic lattice. The plaquettes are now the minimal square faces all along the cube and they are characterized by a vertex and two lattice directions.

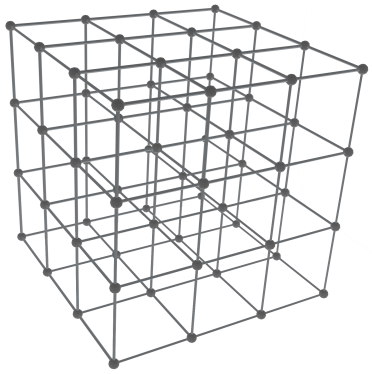


Figure 4.8: $V = 4^3$ cubic lattice.

Plaquette tensors will be now represented by a red geometrical volume. Blue tensors, which come in three different types, help us fix the value of the artificial spins between plaquettes. The structure of the tensor network on an outer face of the lattice very much resembles that of our previous 2D tensor network. In Figure 4.9 we can see a bottom slice along with the tensor corresponding to those plaquettes and edges. 2-tensors T, represented by a blue tetrahedron, are related to spins on an edge of the lattice. 3-tensors S, drawn as a blue cube, represent spins whose edge sits on a face of the lattice. In Figure 4.10 we have hidden the original lattice and replaced it with the legs of the tensors giving the structure to the network.

Besides these tensors S and T, there are 4-tensors I, represented by a blue icosahedron. They are placed on the edges of the lattice belonging to 4 different plaquettes. Following the slicing of the tensor network in planes, we can distinguish two more types. In Figure 4.11 we represent the slices where all plaquettes have the vertical direction as one of their own (vertical as in the long-axis of this paper sheet). We can see how they are placed in between lattice levels. In Figure 4.12 one can find the corresponding legs of the tensors.

Finally, the tensor network slices whose plaquettes are perpendicular to the vertical direction will have a different setup, which can be seen in Figures 4.13 and 4.14.

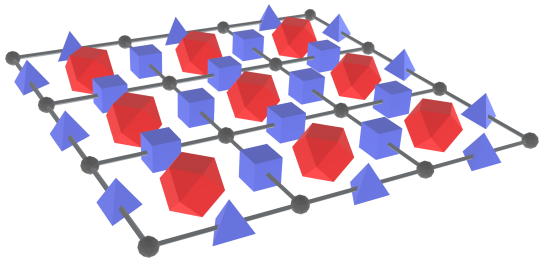


Figure 4.9: Placement of the tensors on a face of the lattice.

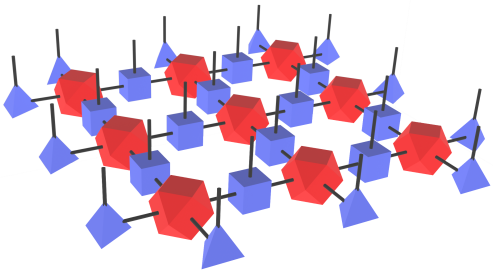


Figure 4.10: Contraction of the tensor network on a face of the lattice.

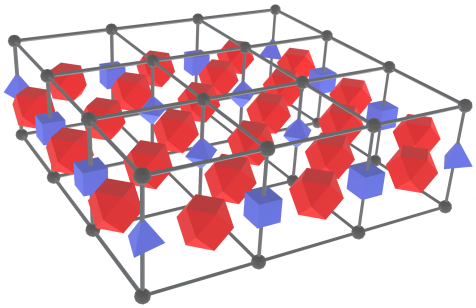


Figure 4.11: Placement of the tensors on an intermediate plane of the lattice.

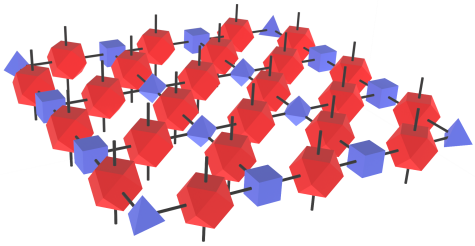


Figure 4.12: Contraction of the tensor network on an intermediate plane of the lattice.

The computation of the plaquette tensors does not change much from those defined in the 2-dimensional lattice. Observe that they are still 4-tensors. The only thing we need to be careful about is choosing the correct matrices given by the vertex and the directions defining the plaquette. Moreover, the coefficients $c_{x,\mu}$ can now take values $1/2, 1/3$ and $1/4$ depending on the edge position in the lattice. Blue tensors are related to the actual spins of the system. They are defined as

$$T_{\mu,\nu} = \sum_{\sigma\in\{-1,+1\}} \delta_{\mu,\nu,\sigma}, \quad S_{\mu,\nu,\xi} = \sum_{\sigma\in\{-1,+1\}} \delta_{\mu,\nu,\xi,\sigma}, \quad I_{\mu,\nu,\xi,o} = \sum_{\sigma\in\{-1,+1\}} \delta_{\mu,\nu,\xi,o,\sigma}, \tag{4.14}$$

where the deltas are Kronecker deltas. Fixing a spin means taking only the corresponding summand of the definition.

To contract this tensor network we will add some transparent tensors as we did in Chapter 3. We glue two transparent tensors $\tau_{\alpha} = 1$ to every tensor of our network with legs pointing in the directions where it lacks one. We of course take into account that tensors at the boundary of the network do not need legs pointing outwards. Finally, add a transparent 6-tensor, result of glueing 6 tensors τ, on each

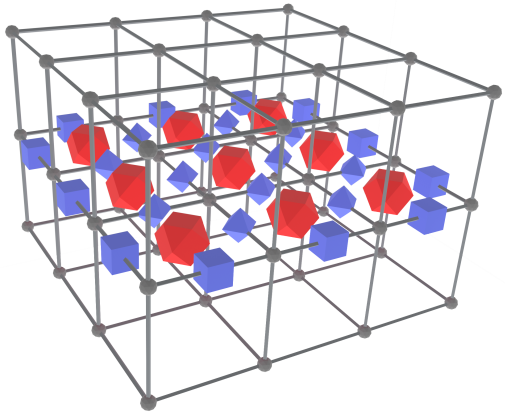


Figure 4.13: Placement of the tensors on an inner plane of the lattice.

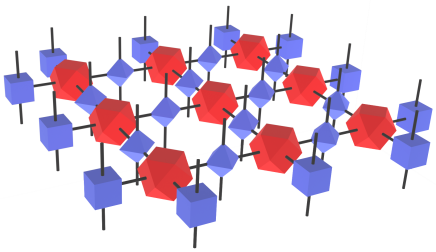


Figure 4.14: Contraction of the tensor network on an inner plane of the lattice.

hole left by the structure of the tensor network at each slide. As an example, in Figure 4.15 we illustrate the transformation done to the bottom layer. This will add additional spins to our network that we will sample for an easier structure of the program, but they are ignored when it comes to computing the resulting configuration of the Higgs model.

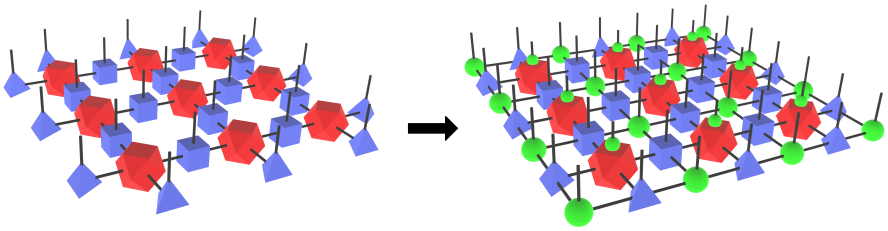


Figure 4.15: Transformation of the bottom layer of the tensor network.

It is with this transformation that each layer now resembles a PEPS in the case of the bottom and top layers and a PEPO for the rest of the inner layers. Renormalization schemes for these structures are similar to those used in the case of MPS. The idea is to contract every PEPO with the PEPS at the top and bottom of the tensor network. We do this by contracting each tensor of the PEPS with the corresponding one in the PEPO along their physical indices pointing to each other. During this process both bond dimensions of the resulting PEPS will start to grow and we may need to renormalize the tensors to keep the tensor ranks under control. After contracting this axis, the resulting structure is the same as the one used for our 2-dimensional tensor network, i.e. 2 MPS at each side and $L-2$ MPO in between that we already know how to work with.

In the same spirit as when we sampled the 2-dimensional lattice, see Section 2.3, storing intermediate tensor structures will help us reduce the computational cost. We start with tensors where no spin has been fixed. Let $|bot\rangle$ be now a PEPS and M_k

a PEPO, we store every PEPS $M_k \dots M_2 |bot\rangle$, which takes $L^2(L-1)$ contractions if done properly. Our first complete contraction is then given by

$$Z(\beta, \kappa) = \langle top | M_{L-1} \dots M_2 | bot \rangle . \tag{4.15}$$

We explain now how we can sample an L^2 spin layer saving some computational effort. Suppose we have sampled the top r layers. We are left with $r \cdot L^2$ tensor with a fixed spin value. We can now contract those r layers into a single PEPS, that we denote as $\langle top_r|$. For the base case of 0 layers sampled, simply take as $\langle top_0|$ a PEPS of transparent tensors. Observe that if we stored $\langle top_{r-1}|$ we only need to contract L^2 tensors. Recover from memory the PEPS $|bot_{L-r-1}\rangle := M_{L-r-1} \dots M_2 |bot\rangle$ and the PEPO M_{L-r}. This last structure holds the tensors corresponding to the spins of the $(r+1)^{\text{th}}$ layer we want to fix. If we have sampled K spins already, this translates to

$$Z(\beta, \kappa | \sigma_1, \dots \sigma_K) = \langle top_r | M_{L-r} | bot_{L-r-1} \rangle . \tag{4.16}$$

After sampling the value of a spin, compute the tensor with the fixed value and replace it in M_{L-r}, creating a new PEPO $\widetilde{M}_{L-r}$ and the new partition function will be given by

$$Z(\beta, \kappa | \sigma_1, \dots \sigma_K, \sigma_{K+1}) = \langle top_r | \widetilde{M}_{L-r} | bot_{L-r-1} \rangle . \tag{4.17}$$

After finishing with the tensors of M_{L-r} we have a new PEPO M'_{L-r} where all the spin values have been fixed. Then $\langle top_{r+1}| = \langle top_r | M'_{L-r}$, and we can continue the process.

This process requires the use of $L^2(L-1)$ contractions to store every $|bot_k\rangle$, plus L^2 contractions after sampling each layer, which we need to do $L-1$ times. In total $2L^2(L-1) = O(L^3)$.

We can reduce the need for $2L^2$ operations to compute the partition function using the same idea but working with MPS and MPO. This would reduce the cost of computing Z to $2L$. The initialization, with a cost of $O(L^2)$, is done once per layer and the storing, with a cost of L, is done $L-1$ times. Taking into account we have L layers, these are an additional $O(L^3)$ contractions.

The final sampling iteration is done with a tensor network of a single dimension, i.e. a line of tensors. In this last stage we have an initializing cost of $O(L)$ and a storing cost of 1 done $L-1$ times per line. Both things are repeated L^2 times, amounting to a total of another $O(L^3)$ operations.

After all these optimizations, the cost of computing the partition function each time is reduced to 6 contraction. Indeed, we first compute the tensor with a fixed spin. Then we contract it with the 2 tensors storing the line information, the 2 tensors from the 2-dimensional optimization and finally the 2 last tensors from $\langle top_r|$ and $|bot_{L-r-1}\rangle$. Hence, we have managed to reduce the computations of each partition function to $O(1)$. As this is repeated L^3 times, together with the cost of storing intermediate steps, the number of contractions is of the order of $O(L^3)$. As the contraction size is limited by the bond dimension bound, the number of multiplications is of the order of $O(DL^3)$, taking D as the maximum of both bond dimensions. If we had not done this extra work, the naive cost would be of the order of $O(L^6)$ contractions to sample a complete spin system.

We would need to include in this analysis the cost of each renormalization, which is more complex and out of the scope of the current work.

4.3.2 Results

Computations for the 3D lattice are more costly than in 2D. Although the scaling from 2 dimensions seems to be only a factor of L, the actual computing cost increases much more. There are many reasons: the additional difficulty of renormalizing PEPS instead of MPS, such process being called more often and technical reasons on the implementation side. As a consequence, we have not been able to extract as much data as is needed to make meaningful observable calculations. We will instead center our attention on comparing the equilibration of the chain compared to that of Metropolis.

At high β, we can think of Boltzmann sampling as minimizing the energy. In some models, minimizing the global energy of the system is achieved by minimizing the energy of each group of interacting degrees of freedom. Take the Ising model without external field as an example.

$$H(\omega) = -\sum_{\langle i,j\rangle \in E} J_{ij}\sigma_i\sigma_j. \tag{4.18}$$

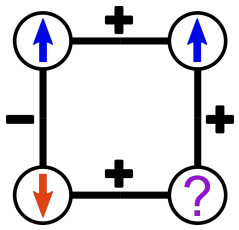


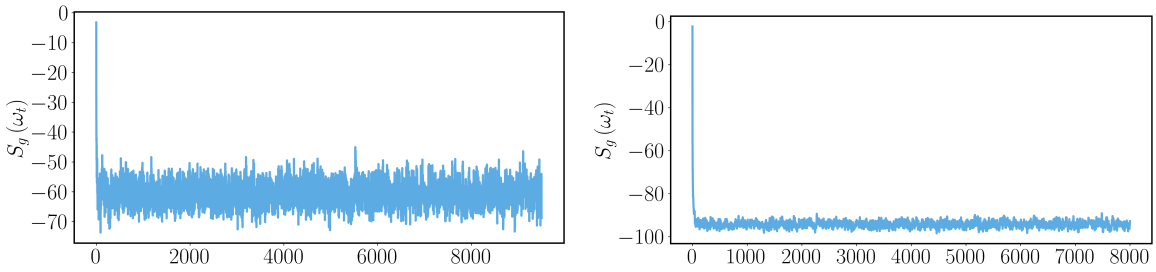
Figure 4.16: Ising model displaying frustration.

We can see how, if $J_{ij} > 0\ \forall i, j$, the easiest way of minimizing $H(\omega)$ is by fixing every spin to the same value, which corresponds to minimizing each $J_{ij}\sigma_i\sigma_j$ term.

Whenever this is not the case and the global minima needs to be found differently, we say the system is *frustrated*. A simple instance would be the previous Ising model, but taking signs for J_{ij} alternated in each row, as in Figure 4.16.

We are interested in finding a setting where TNMH displays an advantage over classical sampling algorithms. The advantage of global updates is mainly visible in systems with high frustration, as minimizing locally might not be the best strategy to reach the lowest energy configuration. Let us go back to the action of the Higgs model, given by $S_H = \beta S_g + \kappa S_\phi + W$. Observe how $S_\phi = -\frac{1}{2}\sum_x \sum_\mu \operatorname{Re}\operatorname{Tr} U_{x,\mu}$ can be easily minimized by taking matrices with higher trace. The interactions of S_g are more complex, and minimizing the energy of each plaquette on its own is not necessarily a means to minimizing S_g.

For this reason, the pure gauge action presents a higher degree of frustration, so we have found the biggest difference in equilibration with parameter $\kappa = 0$. In fact, Metropolis sampling seems to get stuck in local minima, not being able to reach energy levels as low as TNMH does. In Figure 4.17b we can see how for $\beta = 9$ TNMH reaches energy levels below -90, while Metropolis sampling does not go below -70 in Figure 4.17a. The initialization of the algorithms is always done with random matrices drawn according to the Haar measure. This behavior has been observed with other values for β and with smaller sizes than the $V = 4^3$ system displayed here. Hence, it is unlikely to be just a coincidence caused by biased initial conditions.



(a) Metropolis (b) TNMH

Figure 4.17: Energy of a 4^3 pure gauge model with $\beta = 9$ sampled with both algorithms. On the Metropolis execution we can observe how the energy does not manage to reach levels as low as with the sampling from TNMH

The behavior of the Metropolis algorithm in this regime is probably the first example of metastability that we have encountered so far.

One now wonders why, if the frustration is also present in the 2D model, TNMH is has not displayed the same advantage over Metropolis sampling. In 3 dimensions the plaquettes to which a single matrix belongs is 4, while in 2 dimensions stays at 2. This quantities are given assuming the matrix is not placed at the boundary of the lattice. The more plaquettes to which a matrix belongs the more the frustration of the system can make it harder to minimize the energy. Pur main believe is that going from a valence of 2 to 4 might be the turning point for observing this difficulty in Metropolis sampling that TNMH is still able to overcome. An experiment that could back up this idea would be to build a 2-dimensional system with a hexagonal lattice placing the degrees of freedom on the vertices. With this lattice, each matrix could belong to 3 different plaquettes and we could try to observe the same behavior as that displayed in 3 dimensions.

As we have not been able to reach a point with Metropolis where we can assume it has converged to equilibrium, there is no point in comparing autocorrelation times.

4.4 Computations

We would like to make a few comments about the resources used for these computations and the implementation itself. The programming has been done in Python. We consider this a simple testing phase of the algorithm where we are exploring its capabilities and limitations, so we have not deemed it worth employing a faster language or making a big optimization effort.

Another reason why talking about CPU time of each execution is not meaningful is because computations have been run on three different Linux machines. The most powerful of them being a workstation with 32 threads in 8 CPUs running at 2.1GHz, then a laptop with a processor of 12 threads at 4.1 GHz and lastly a PC with 4 threads running at 3.4GHz. The RAM specifications are not as important for this project and the GPU is not even used. To give the reader a rough idea of the computational cost of this project, the executions to obtain Figure 4.17b took

about three days. These were done using a low bond dimension, which in the 3D model can make a big difference in the observed CPU times. However, 10^4 steps for a Markov chain is a very small amount for usual standards. If we focus instead on the executions in 2 dimensions to compute observables and correlations times, the computers were left running for a week. As 5 independent chains were used, one could in principle run different processes at the same time if the computer is powerful enough.

The amount of data to store and process is an important factor to be taken into account. Models using $SU(2)$ variables can be somewhat lighter, but if one samples with groups like $SU(10)$, storing 15000 steps of an 8×8 lattice can take up around 15GB of memory. For reference, only the data used to create the graphs in this document weights around 100GB.

Chapter 5

Conclusions

We have studied physically motivated lattice gauge theories in two and three dimensions, using a new Monte Carlo method that achieves global updates with high acceptance rates.

We have worked with desirable properties of Markov chains to help us determine if the stochastic process equilibrates. Additionally, we have placed special interest in autocorrelation times, used to assess the sampling behavior once equilibrium has been reached. This has helped us identify the performance not being as expected in 2 dimensions, as we have found autocorrelation times to be similar to Metropolis, contrary to previous belief. On the other hand, equilibration times place TNMH as a good candidate to sample systems in higher dimensions. As far as we have observed, it is able to reach equilibrium in the 3-dimensional setting, where the Metropolis algorithm stagnates in higher energy levels.

For $SU(2)$ theories, the computation of observables has been compared with existing literature, obtaining similar results for the values tested. The method to propose configuration changes lends itself well to changing the group of the degrees of freedom. Similar results have been observed for $SU(3)$, $SU(5)$ and $SU(10)$, reinforcing the fidelity of the computations done with the algorithm. Although we have restricted the system fixing the values $\rho_x = 1$, there is a simple way of including these variables in the sampling with what we already know. We could fix the values ρ_x and sample $U_{x,\mu}$ as we have done here. Then, in the next TNMH step, we would fix $U_{x,\mu}$ and sample ρ_x. This second step is very similar to the XY model, as our degrees of freedom are scalars in the vertices of the lattice and we could follow the suggestions from Frías Pérez et al. (2023).

We have also studied different tensor network structures that help us understand how a 3-dimensional tensor network can be contracted. These structures play an important role in the implementation done during this year's development. The results in 3D are very promising. As expected, the computation in this regime has proven to be very demanding, preventing us from getting as many results as we would have liked with the new lattice. Hence, part of the future work includes optimizing the new tensor network contraction and hopefully make use of higher computational power.

Obtaining high autocorrelation times despite having almost perfect acceptance

rates was a very big surprise for us. For this reason, we have started to look into other ways to propose the new configurations. A fruitful idea with compact group sampling is using representation theory (Bazavov et al., 2019). By doing so, one can map the original problem to one of sampling irreducible representations of the gauge group. This was in fact done with TNMH and the XY model (Frías Pérez et al., 2023). The group used is $U(1)$ which, as it is abelian, has a nicer representation than $SU(2)$. It was observed to perform well in the XY model and, if we put in the work with the representation of $SU(2)$, we believe we can obtain good results too.

Finally, an idea worth exploring is increasing the precision of the computations. There are a lot of interesting physical phenomena at high values of β. However, raising the exponent in the exponential function used to compute the tensors can cause quantities to be rounded to 0 or overflow. This is the reason why we have not gone beyond values of $\beta = 10$. Using multiple or arbitrary precision can make computations a lot slower, as many technical aspects appear within the low-level computer realm. Building a high precision version of TNMH will require utilizing new libraries and optimizing the code, which we plan to explore in future work too. A favorable gap in performance between TNMH and Metropolis for 2D might open up at higher system sizes and values of β, where extra precision will be required.

Bibliography

Bazavov, A., Catterall, S., Jha, R.G., Unmuth-Yockey, J., 2019. Tensor renormalization group study of the non-abelian higgs model in two dimensions. Physical Review D 99, 114507.

Bock, W., Evertz, H.G., Jersák, J., Landau, D.P., Neuhaus, T., Xu, J.L., 1990. Search for critical points in the su(2) higgs model. Phys. Rev. D 41, 2573–2580.

Creutz, M., Jacobs, L., Rebbi, C., 1983. Monte carlo computations in lattice gauge theories. Physics Reports 95, 201–282.

Díaz Rodríguez, J.C., 2023. Collective Monte Carlo updates for continuous correlated systems. Bachelor's final project. Universidad Complutense de Madrid.

Feynman, R.P., 1948. Space-time approach to non-relativistic quantum mechanics. Rev. Mod. Phys. 20, 367–387.

Frías Pérez, M., Mariën, M., Pérez García, D., Bañuls, M.C., Iblisdir, S., 2023. Collective monte carlo updates through tensor network renormalization. SciPost Physics 14.

Gowers, T., . How to lose your fear of tensor products. Available at `https://www.dpmms.cam.ac.uk/~wtg10/tensors3.html` (11/05/2024).

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109.

Iblisdir, S., 2020. Lecture notes in statistical and mathematical techniques.

Ising, E., 1924. Beitrag zur theorie des ferro-und paramagnetismus. Ph.D. thesis. Grefe & Tiedemann Hamburg.

Katzgraber, H.G., 2011. Introduction to monte carlo methods. `arXiv:0905.1629`.

Kogut, J.B., 1979. An introduction to lattice gauge theory and spin systems. Reviews of Modern Physics 51, 659.

Kuramashi, Y., Yoshimura, Y., 2019. Three-dimensional finite temperature z2 gauge theory with tensor network scheme. Journal of High Energy Physics 2019, 1–13.

Lancaster, T., Blundell, S.J., 2014. Quantum field theory for the gifted amateur. OUP Oxford.

Levin, M., Nave, C.P., 2007. Tensor renormalization group approach to two-dimensional classical lattice models. Physical review letters 99, 120601.

Mezzadri, F., 2007. How to generate random matrices from the classical compact groups. arXiv:math-ph/0609050.

Montvay, I., Münster, G., 1994. Quantum fields on a lattice. Cambridge University Press.

Morningstar, C., 2007. The monte carlo method in quantum field theory. arXiv preprint hep-lat/0702020 .

Murg, V., Verstraete, F., Cirac, J.I., 2005. Efficient evaluation of partition functions of inhomogeneous many-body spin systems. Physical review letters 95, 057206.

Perez-Garcia, D., Verstraete, F., Wolf, M.M., Cirac, J.I., 2006. Matrix product state representations. arXiv preprint quant-ph/0608197 .

Ran, S.J., Tirrito, E., Peng, C., Chen, X., Tagliacozzo, L., Su, G., Lewenstein, M., Ran, S.J., Tirrito, E., Peng, C., et al., 2020. Tensor network approaches for higher-dimensional quantum lattice models. Tensor Network Contractions: Methods and Applications to Quantum Many-Body Systems , 87–97.

Reichl, L., 1998. A modern course in statistical physics. University of Texas Press, Austin. chapter 7. pp. 343 – 349.

Schwartz, M., 2019. Lecture notes in statistical mechanics. Online by Harvard University, (11/06/2024).

Sokal, A., 1997. Monte carlo methods in statistical mechanics: foundations and new algorithms, in: Functional integration: Basics and applications. Springer, pp. 131–192.

Tong, D., 2018. Gauge theory. Lecture notes, DAMTP Cambridge 10.

Verstraete, F., Cirac, J.I., 2004. Renormalization algorithms for quantum-many body systems in two and higher dimensions. arXiv:cond-mat/0407066.

Verstraete, F., Murg, V., Cirac, J., 2008. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. Advances in Physics 57, 143–224.

Wegner, F.J., 1971. Duality in Generalized Ising Models and Phase Transitions without Local Order Parameters. Journal of Mathematical Physics 12, 2259–2272.

Yu, J., Xie, Z., Meurice, Y., Liu, Y., Denbleyker, A., Zou, H., Qin, M., Chen, J., Xiang, T., 2014. Tensor renormalization group study of classical xy model on the square lattice. Physical Review E 89, 013308.

Zohar, E., Cirac, J.I., 2018. Combining tensor networks with monte carlo methods for lattice gauge theories. Physical Review D 97, 034510.

Appendix A

Suzuki-Trotter transformation

In this section, we show an example of how one can go from Quantum Mechanics in d dimensions to Statistical Mechanics with an additional dimension. We follow the lecture notes from Iblisdir (2020). These transformations are somewhat ad-hok and have to be designed in each case, although a general guide can be extracted. To illustrate how this can be done we will develop a very simple case, that of a single spin Hamiltonian, i.e. $d = 0$. This is represented by a Hilbert space of dimension 2. Elements of the base are usually denoted by $|+1\rangle$ and $|-1\rangle$, or in general $|\sigma\rangle$.

Let our spin-1/2 system be described by the Hamiltonian

$$H = -\sigma^z - \lambda\sigma^x, \tag{A.1}$$

where $\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ are Pauli matrices. Again we are interested in the properties of the ground state $|\psi_0\rangle$ and, for example, we want to compute the expected value for σ^x:

$$\langle\sigma^x\rangle = \frac{\langle\psi_0|\sigma^x|\psi_0\rangle}{\langle\psi_0|\psi_0\rangle} \tag{A.2}$$

Following Eq. 1.8, we are going to work with the matrix $e^{-\tau H}$. The Suzuki-Trotter identity states:

$$\|e^{\varepsilon(X+Y)} - e^{\varepsilon X/2}e^{\varepsilon Y}e^{\varepsilon X/2}\|_\infty = O\left(\varepsilon^2\|[X,Y]\|_\infty\right). \tag{A.3}$$

Now we split the Hamiltonian as $H = H_A + H_B$, choosing

$$H_A = -\sigma^z, \quad H_B = -\lambda\sigma^x. \tag{A.4}$$

Since $-\tau H/m$ commutes with itself, we can write $e^{-\tau H} = \prod_{j=1}^m e^{-\tau H/m}$. Applying the Suzuki-Trotter identity

$$e^{-\tau H/m} = e^{-\tau H_A/2m}e^{-\tau H_B/m}e^{-\tau H_A/2m} + O\left(\frac{\tau^2}{m^3}\|[H_A,H_B]\|_\infty\right) \tag{A.5}$$

Let $\mathcal{T} := e^{-\tau H_A/2m} e^{-\tau H_B/m} e^{-\tau H_A/2m}$, for m and τ large enough we can approximate $\langle \sigma^x \rangle$ as

$$\langle \sigma^x \rangle \approx \frac{\langle \phi_0 | \mathcal{T}^m \sigma^x \mathcal{T}^m | \phi_0 \rangle}{\langle \phi_0 | \mathcal{T}^{2m} | \phi_0 \rangle}, \tag{A.6}$$

with an error decreasing as $1/m$. Using linearity and $I_2 = \sum_\sigma |\sigma\rangle \langle\sigma|$ we can rewrite the denominator as

$$\sum_{\sigma_1} \cdots \sum_{\sigma_{2m}} \langle \phi_0 | \mathcal{T} | \sigma_1 \rangle \langle \sigma_1 | \mathcal{T} | \sigma_2 \rangle \ldots \langle \sigma_{2m-1} | \mathcal{T} | \sigma_{2m} \rangle \langle \sigma_{2m} | \phi_0 \rangle \tag{A.7}$$

Now consider the operator $T(\beta) = \sum_{\sigma,\sigma'} e^{\beta\sigma\sigma'} |\sigma\rangle \langle\sigma'|$. An equivalent expression is given by

$$\sum_{\sigma,\sigma'} e^{\beta\sigma\sigma'} |\sigma\rangle \langle\sigma'| = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix} = e^{\beta} \begin{pmatrix} 1 & e^{-2\beta} \\ e^{-2\beta} & 1 \end{pmatrix} = e^{\beta} \left(I + e^{-2\beta} \sigma^x \right). \tag{A.8}$$

Given the equality

$$e^{-\tau H_B/m} = e^{\lambda\tau\sigma^x/m} = \cosh\left(\frac{\lambda\tau}{m}\right) \cdot \left(I + \tanh\frac{\lambda\tau}{m}\, \sigma^x \right), \tag{A.9}$$

we can identify $e^{-2\beta} = \tanh\frac{\lambda\tau}{m}$ to obtain

$$e^{-\tau H_B/m} = e^{-\beta} \cosh\left(\frac{\lambda\tau}{m}\right) T(\beta). \tag{A.10}$$

Now, we develop the expression $\mathcal{T} |\sigma\rangle$ as

$$\begin{aligned} e^{\frac{-\tau}{2m}H_A} e^{\frac{-\tau}{m}H_B} e^{\frac{-\tau}{2m}H_A} |\sigma\rangle &= e^{-\beta} \cosh\left(\frac{\lambda\tau}{m}\right) \sum_{\sigma',\sigma''} e^{\frac{-\tau}{2m}H_A} e^{\beta\sigma'\sigma''} |\sigma'\rangle \langle\sigma''| e^{\frac{-\tau}{2m}H_A} |\sigma\rangle \\ &= e^{-\beta} \cosh\left(\frac{\lambda\tau}{m}\right) \sum_{\sigma',\sigma''} e^{\beta\sigma'\sigma''} e^{\frac{\tau}{2m}\sigma^z} |\sigma'\rangle \langle\sigma''| e^{\frac{\tau}{2m}\sigma^z} |\sigma\rangle \\ &= e^{-\beta} \cosh\left(\frac{\lambda\tau}{m}\right) \sum_{\sigma',\sigma''} e^{\beta\sigma'\sigma'' + \frac{\tau}{2m}\sigma' + \frac{\tau}{2m}\sigma''} |\sigma'\rangle \langle\sigma''|\sigma\rangle \\ &= e^{-\beta} \cosh\left(\frac{\lambda\tau}{m}\right) \sum_{\sigma'} e^{\beta\sigma'\sigma + \frac{\tau}{2m}\sigma' + \frac{\tau}{2m}\sigma} |\sigma'\rangle, \end{aligned} \tag{A.11}$$

where we have used

$$e^{\frac{\tau}{2m}\sigma^z} |\sigma\rangle = e^{\frac{\tau}{2m}(|+\rangle\langle+| - |-\rangle\langle-|)} |\sigma\rangle = \left(e^{\frac{\tau}{2m}} |+\rangle \langle+| + e^{\frac{-\tau}{2m}} |-\rangle \langle-| \right) |\sigma\rangle = e^{\frac{\sigma\tau}{2m}} |\sigma\rangle. \tag{A.12}$$

Suppose we choose as starting state $|\phi_0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$. We spare the calculations, but using Eq. A.11, linearity and the orthogonality of the basis, we obtain

$$\begin{aligned}\langle\phi_0|\mathcal{T}^{2m}|\phi_0\rangle &= \frac{1}{2}\sum_{\sigma_0}\cdots\sum_{\sigma_{2m}}\prod_{j=0}^{2m-1}\langle\sigma_j|\mathcal{T}|\sigma_{j+1}\rangle \\ &= \frac{\left(e^{-\beta}\cosh\frac{\lambda\tau}{m}\right)^{2m}}{2}\sum_{\omega}e^{-\beta H_{\mathrm{cl}}(\omega)} \equiv \frac{\left(e^{-\beta}\cosh\frac{\lambda\tau}{m}\right)^{2m}}{2}Z(\beta),\end{aligned} \tag{A.13}$$

where the Hamiltonian H_{cl} corresponds to a 1-dimensional Ising model

$$-\beta H_{\mathrm{cl}}(\omega) = -\sum_{j=0}^{2m}\left(\beta\sigma_j\sigma_{j+1} + \frac{\varepsilon}{2}\sigma_j + \frac{\varepsilon}{2}\sigma_{j+1}\right). \tag{A.14}$$

We also need to work on the numerator of Eq. A.6. We can express it as

$$\langle\phi_0|\left(\prod_{j=1}^{m}\mathcal{T}\sum_{\sigma_j}|\sigma_j\rangle\langle\sigma_j|\right)\sigma^x\left(\prod_{k=1}^{m}\mathcal{T}\sum_{\sigma_{k+m}}|\sigma_{k+m}\rangle\langle\sigma_{k+m}|\right)|\phi_0\rangle. \tag{A.15}$$

Finally, observe that in the previous expression, we always find products of the form $\langle\sigma_j|\mathcal{T}|\sigma_{j+1}\rangle$, but for the element in the middle $\langle\sigma_m|\sigma^x\mathcal{T}|\sigma_{m+1}\rangle$. We can however express it as

$$\langle\sigma_m|\sigma^x\mathcal{T}|\sigma_{m+1}\rangle = g^x(\sigma_m,\sigma_{m+1})\langle\sigma_m|\mathcal{T}|\sigma_{m+1}\rangle. \tag{A.16}$$

Again, using eqs. A.11 and A.15, linearity and the orthogonality of the basis we obtain the equality

$$\langle\phi_0|\mathcal{T}^m\sigma^x\mathcal{T}^m|\phi_0\rangle = \frac{e^{-\beta}\cosh\frac{\lambda\tau}{m}}{2}\sum_{\omega}g^x(\sigma_m,\sigma_{m+1})\,e^{-\beta H_{\mathrm{cl}}(\omega)}. \tag{A.17}$$

Hence, this approximation has given us a translation from the mean value in the quantum context of σ^x, to the mean magnetization of a site in a 1-dimensional Ising model as

$$\frac{\langle\psi_0|\sigma^x|\psi_0\rangle}{\langle\psi_0|\psi_0\rangle} \approx \frac{\sum_\omega g^x(\sigma_m,\sigma_{m+1})\,e^{-\beta H_{\mathrm{cl}}(\omega)}}{Z(\beta)}. \tag{A.18}$$

If this work were to focus on these aspects, we would also need to study the error in this approximation, as well as more complex models than the one shown. However, this is enough to give an intuition on why we are going to work in the context of Statistical Mechanics.