

Real-Time Simulations of Open Quantum Spin Systems Driven by Measurement

Masterarbeit

der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Manes Hornung

2015

Leiter der Arbeit

Prof. Dr. Uwe-Jens Wiese

Albert Einstein Center for Fundamental Physics
Institut für Theoretische Physik
Universität Bern

Abstract

We study the real-time evolution of open quantum spin systems driven by dissipative measurement processes. The internal Hamiltonian dynamics are neglected. We consider the XY-Model, the Heisenberg antiferromagnet and the Heisenberg ferromagnet and three different measurement processes with different symmetry properties.

Numerical simulations show that the systems are driven into a new equilibrium by the measurement processes. We find that the dynamics of this equilibration depends strongly on the symmetry of the respective process.

In addition, the conditions on a measurement process to be simulatable are investigated. We find indications that the set of simulatable processes corresponds to a discrete subset of all processes.

Contents

Introduction	6
1. Simulations in Thermal Equilibrium	8
1.1. Considered Systems	8
1.2. From the Density Matrix to the Path Integral	9
1.3. Sampling the Probability Density	14
1.4. Detailed Balance	15
1.5. Deriving Cluster Rules	18
1.6. A Different Basis	22
1.7. Cluster Rules for the Considered Cases	24
2. Real-Time Simulations	26
2.1. Problems of Real-Time Simulations	26
2.2. Theory of Measurement in Quantum Mechanics	27
2.2.1. Orthogonal Measurements	27
2.2.2. Continuous Measurements	28
2.3. Simulating Measurement Processes	29
2.3.1. Discrete Measurements	29
2.3.2. Continuous Measurements	33
2.4. Simulated Measurement Processes and their Cluster Rules	33
2.4.1. Total Spin	34
2.4.2. $S_x^1 S_y^1$	37
2.4.3. $S_x^+ S_y^+ + S_x^- S_y^-$	39
2.5. Cluster Rules for Continuous Measurement Processes	42
2.6. What can be Simulated	42
2.6.1. The \mathbb{R}^2 Case	44
2.6.2. The \mathbb{C}^4 Case	46
3. Numerical Studies	50
3.1. Discrete Measurements	50

3.2. Continuous Measurements	52
3.2.1. AFM Initial Density Matrix	52
3.2.2. XY Initial Density Matrix	52
3.2.3. FM Initial Density Matrix	53
3.2.4. Equilibration Times	54
4. Conclusion and Outlook	59
Appendix	61
A. Statistics	61
A.1. Averages and Uncertainties	61
A.2. Jackknife	62
B. Units	64
C. Notes on the Implementation	65
Acknowledgements	67

Introduction

Simulations have become an important part of modern physics. Whenever a problem cannot be solved analytically one can try to implement it in a computer simulation and in many cases find a numerical solution. The physics of strongly coupled systems is such a case: Analytic solutions are mainly restricted to 1-d systems, but Monte Carlo simulations provide a way to explore various such systems in arbitrary dimensions in equilibrium, be they quantum or classical. However, simulating these systems far from equilibrium e.g. their evolution in time, seems to be a very hard problem and is considered a major challenge of theoretical physics today [1]. All approaches which have been made so far are limited: Monte Carlo techniques suffer from a severe sign or complex phase problem. With exact diagonalization techniques one can calculate the real-time evolution of small systems. However, since the Hilbert space usually grows exponentially with the system size, this method becomes inapplicable for larger systems. Another approach is the density matrix renormalization group [2, 3], which can be used to examine the real-time dynamics of some 1-dimensional systems but only for short time intervals.

What makes it so hard to do such simulations? One reason is that closed quantum systems tend to evolve into strongly entangled states, so called Schrödinger cat states. The quantum mechanical effects become very strong, they behave very differently from classical systems. Classical computers on the other hand work according to the laws of classical physics and thus deterministically. It is therefore no surprise that they are not well suited to tackle such quantum problems which are non-deterministic in nature. This was already pointed out by Feynman in 1982 [4]. He proposed to use quantum simulators, very well controlled quantum devices which mimic the behaviour of other quantum systems. In fact, such devices can be realized with ultra-cold atoms in optical lattices [5]. However, the development of such devices is still in the beginnings. There is no such thing as a universal quantum simulator (not yet at least). Therefore the problem of simulating the real-time evolution of a quantum system on a classical computer remains an interesting one.

Instead of making the classical computer more quantum one can also try to make the quantum problem more classical to tackle the difficulties stated above. In fact this

approach is quite natural since systems which occur in nature are usually not closed but coupled to their environment. The strongly entangled Schrödinger cat states then do not occur since the coupling to the environment usually decoheres the system. The interaction between the system and the environment is well described by the quantum theory of open systems (see for example [6]) and can be modeled as a continuous measurement process. Here we will pursue this approach which was first described in [7]. The continuous measurement process is then described by a Lindblad process, which can be simulated with an efficient loop cluster algorithm [8, 9]. Using this method one can simulate the real-time evolution driven by this measurement process over an arbitrary length of time for systems in arbitrary dimensions.

Here we study the real-time evolution of the Fourier modes of the magnetization of 2-d quantum spin systems initially prepared in thermal equilibrium. The initial state is prepared with the same loop cluster algorithm using the Hamiltonians for the Heisenberg antiferromagnet, the Heisenberg ferromagnet and the quantum XY-model. We find that the system is driven into a new equilibrium by the measurement process, where the timescale of this equilibration depends strongly on the symmetries of the process. In addition, we examine what conditions a measurement process must fulfil to be simulatable with the loop cluster algorithm.

This thesis is organized as follows: In chapter 1 we define the studied systems and give a description of the loop cluster algorithm used to prepare the initial state density matrix in thermal equilibrium. In chapter 2 we give a very short introduction to the theory of measurements and show how the loop cluster algorithm can be used to simulate Lindblad processes. Furthermore, we describe the measurement processes used in the simulations, derive their cluster rules and investigate what processes can be simulated i.e. make the sign problem disappear. The numerical results are presented in chapter 3, and finally conclusion and outlook are given in chapter 4.

1. Simulations in Thermal Equilibrium

With the loop cluster algorithm one can simulate expectation values of observables for a certain class of quantum spin systems in thermal equilibrium. In the first part of this chapter, we will, after clarifying the considered systems, derive path integral expressions for expectation values of observables and relate them to a probability distribution. This probability distribution can be sampled to calculate these expectation values with the loop cluster algorithm. This is explained in the second part. This chapter is based on [8, 9, 10].

1.1. Considered Systems

The studied systems are d -dimensional quantum spin lattices. They consist of a d -dimensional bi-partite lattice with a quantum spin $s = 1/2$ attached to each point. Periodic boundary conditions are assumed. Although the methods described in the following sections work with arbitrary dimensional systems, we will concentrate on 2-dimensional square lattices. We will denote the spacial length of the systems as L . Lattice sites will be denoted as x and y . The Hamiltonian of all considered systems can be written as

$$\mathcal{H} = \sum_{\langle x, y \rangle} J (S_x^1 S_y^1 + S_x^2 S_y^2) + J' S_x^3 S_y^3, \quad (1.1)$$

where $\langle x, y \rangle$ denotes a nearest-neighbour pair of sites x and y on the lattice. The S_x^i 's are the spin operators in direction $i \in 1, \dots, 3$ at location x . They fulfill the commutation relations

$$[S_x^i, S_y^j] = i\delta_{xy}\epsilon_{ijk}S_x^k, \quad (1.2)$$

with \hbar set to 1. For $J = J' < 0$ this corresponds to a Heisenberg antiferromagnet, for $J = J' > 0$ to a Heisenberg ferromagnet, for $J' = 0$ to the quantum XY-model, and for $J = 0$ to the Ising model.

A basis of the Hilbert space can be written as

$$\{|n\rangle\} = \left\{ \left| s_{x_1} s_{x_2} s_{x_3} \dots s_{x_{ld}} \right\rangle \right\}, \quad (1.3)$$

where $s_{x_i} \in \{\uparrow, \downarrow\}$ is the spin along a certain quantization axis j , at site x_i and

$$S_x^j |\uparrow\rangle = \frac{1}{2} |\uparrow\rangle, \quad S_x^j |\downarrow\rangle = -\frac{1}{2} |\downarrow\rangle. \quad (1.4)$$

We will use the 3- and the 1-direction as quantization axes.

1.2. From the Density Matrix to the Path Integral

A quantum mechanical ensemble is completely characterized by its density matrix

$$\rho = \sum_m |m\rangle \langle m| p_m, \quad (1.5)$$

where the sum is over a basis, in which ρ is diagonal, and p_m is the probability to find the system in the state $|m\rangle$. For an ensemble in thermal equilibrium, a canonical ensemble, this density matrix is given by

$$\rho = \frac{1}{Z} \exp(-\beta \mathcal{H}), \quad (1.6)$$

with the partition function being $Z = \text{Tr} \exp(-\beta \mathcal{H})$. The expectation value of an observable in a quantum mechanical ensemble is given by

$$\langle \mathcal{O} \rangle = \text{Tr} \mathcal{O} \rho \quad (1.7)$$

where \mathcal{O} is the operator representing the observable. For $\langle \mathcal{O} \rangle$ we find

$$\langle \mathcal{O} \rangle = \sum_n \langle n | \mathcal{O} \rho | n \rangle = \sum_n o_n \langle n | \rho | n \rangle = \frac{1}{Z} \sum_n o_n \langle n | \exp(-\mathcal{H}\beta) | n \rangle, \quad (1.8)$$

the sum being over a complete set of states. Here we assume that \mathcal{O} is diagonal in the basis $\{|n\rangle\}$ and denote the eigenvalue to the state $|n\rangle$ as o_n . In principle we could calculate this expression, and we indeed can for small systems. However, the Hilbert space of a d -dimensional spin lattice of extent L has dimension 2^{L^d} , so the number of terms which one has to sum grows exponentially with the system size, making it impossible to perform exact calculations for larger systems. Instead we will do the following. If all terms $\langle n | \exp(-\mathcal{H}\beta) | n \rangle$ are non-negative, we can interpret $\langle n | \exp(-\mathcal{H}\beta) | n \rangle / Z$ as a probability distribution of the system states $|n\rangle$. This distribution can then be sampled via Monte Carlo techniques, giving us a way to estimate $\langle \mathcal{O} \rangle$. The problem which remains in this case is the exponentiation of \mathcal{H} . Since the dimension of the matrix representing

\mathcal{H} again grows exponentially with the system size, this is a really hard problem. We will therefore further simplify this expression.

Let us now discretize Euclidean time and introduce the corresponding lattice spacing a . The inverse temperature β can then be expressed as aN with N being a dimensionless integer, i.e.

$$\langle \mathcal{O} \rangle = \sum_n o_n \langle n | \exp(-\mathcal{H}aN) | n \rangle \quad (1.9)$$

$$= \sum_n o_n \langle n | \prod_{j=1}^N \exp(-\mathcal{H}a) | n \rangle \quad (1.10)$$

Now we split the Hamiltonian in parts, such that each spin appears only once in each part. Thus the resulting parts consist only of commuting summands. For simplicity we first look at one-dimensional systems (spin chains) with spacial extent L . The generalisation to higher dimensions will be straightforward. In the 1-d case we split \mathcal{H} into two parts,

$$\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2 \quad (1.11)$$

with

$$\begin{aligned} \mathcal{H}_1 &= \sum_{x \in 2\mathbb{Z}} J (S_x^1 S_{x+1}^1 + S_x^2 S_{x+1}^2) + J' S_x^3 S_{x+1}^3 = \sum_{x \in 2\mathbb{Z}} h_{xx+1}, \\ \mathcal{H}_2 &= \sum_{x \in 2\mathbb{Z}+1} J (S_x^1 S_{x+1}^1 + S_x^2 S_{x+1}^2) + J' S_x^3 S_{x+1}^3 = \sum_{x \in 2\mathbb{Z}+1} h_{xx+1}. \end{aligned} \quad (1.12)$$

where we have defined

$$h_{xy} = J (S_x^1 S_y^1 + S_x^2 S_y^2) + J' S_x^3 S_y^3. \quad (1.13)$$

Using the Suzuki-Trotter decomposition formula

$$\exp(\mathcal{A}) = \lim_{N \rightarrow \infty} (\exp(\mathcal{B}/N) \exp(\mathcal{C}/N))^N \quad (1.14)$$

for any operators \mathcal{A} , \mathcal{B} and \mathcal{C} such that $\mathcal{A} = \mathcal{B} + \mathcal{C}$, we get for the observable

$$\langle \mathcal{O} \rangle = \lim_{\substack{N \rightarrow \infty \\ a \rightarrow 0 \\ Na = \beta}} \sum_n \frac{1}{Z} o_n \langle n | \prod_{j=1}^N \exp(-a\mathcal{H}_1) \exp(-a\mathcal{H}_2) | n \rangle \quad (1.15)$$

By inserting independent identity operators ($\mathbf{1} = \sum_{n_j} |n_j\rangle \langle n_j|$) between all exponential functions we arrive at the expression

$$\langle \mathcal{O} \rangle = \lim_{\substack{N \rightarrow \infty \\ a \rightarrow 0 \\ Na = \beta}} \sum_{n, n_1, \dots, n_{2N}} \frac{1}{Z} o_n \langle n | \exp(-a\mathcal{H}_1) | n_1 \rangle \langle n_1 | \exp(-a\mathcal{H}_2) | n_2 \rangle \langle n_2 | \dots \\ \dots | n_{2N-1} \rangle \langle n_{2N-1} | \exp(-a\mathcal{H}_2) | n \rangle . \quad (1.16)$$

We can think of this expression as a path integral on a (1+1)-dimensional space-time lattice. The additional dimension is, because of the strong analogy to unitary time evolution, called Euclidean time. In each Euclidean time step only one of the two Hamiltonians is active, connecting spin pairs (s_x, s_{x+1}) with x being either even or odd. We can picture this as a checkerboard covering the lattice, consisting of two different kinds of plaquettes, representing interactions governed by the two different Hamiltonians (see figure 1.1). Still it seems that we have not gained much. The matrices representing the

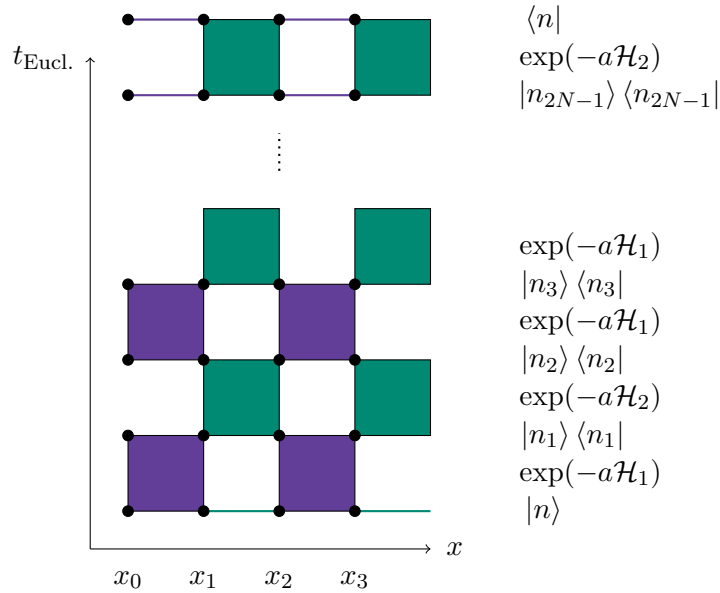


Figure 1.1.: Schematic picture of the checkerboard covering of the (1+1)-dimensional lattice. The lattice sites are represented by black dots. The coloured plaquettes mark which spins interact with the respective Hamiltonian. On the right-hand side, the correspondence between the expressions in the path integral and the 1+1 dimensional lattice is indicated.

operators \mathcal{H}_1 and \mathcal{H}_2 have the same large dimension as the one representing \mathcal{H} . However, they now consist of commuting summands h_{xy} . Thus we can use

$$\begin{aligned}\exp(-a\mathcal{H}_1) &= \prod_{x \in 2\mathbb{Z}} \exp(-ah_{xx+1}) , \\ \exp(-a\mathcal{H}_2) &= \prod_{x \in 2\mathbb{Z}+1} \exp(-ah_{xx+1}) .\end{aligned}\tag{1.17}$$

Each h_{xy} involves only two spins, and can be represented by a 4×4 matrix, which is easy to exponentiate. In addition, each h_{xy} belongs to exactly one plaquette. Writing out the states $|n_i\rangle$ as

$$|n_i\rangle = |s_{(x_1,i)} s_{(x_2,i)} \cdots s_{(x_L,i)}\rangle ,\tag{1.18}$$

we can write equation (1.16) as

$$\begin{aligned}\langle \mathcal{O} \rangle &= \sum_{\mathcal{S}} \frac{1}{Z} o_n \prod_{\substack{k=1 \\ k \text{ odd}}}^{2N} \prod_{\substack{x=1 \\ x \text{ odd}}}^L \langle s_{(x,k)} s_{(x+1,k)} | \exp(-ah_{xx+1}) | s_{(x,k+1)} s_{(x+1,k+1)} \rangle \\ &\quad \langle s_{(x-1,k+1)} s_{(x,k+1)} | \exp(-ah_{x-1x}) | s_{x-1,k+2} s_{x,k+2} \rangle .\end{aligned}\tag{1.19}$$

The sum is now over all configurations of the (1+1)-d extended system, $\mathcal{S} = \{s_{(x,k)}\}$. We can decompose the configuration of the extended system, in configurations of individual plaquettes,

$$\mathcal{S} = \{\mathcal{S}_p\} , \quad \mathcal{S}_p = \{s_{x_p}, s_{y_p}, s_{x'_p}, s_{y'_p}\} ,\tag{1.20}$$

where $x_p^{(i)}$ and $y_p^{(i)}$ denote the sites belonging to the plaquette p . Defining the plaquette weight function

$$W_p(\mathcal{S}_p) = \langle s_{x_p} s_{y_p} | \exp(-ah_{xy}) | s_{x'_p} s_{y'_p} \rangle ,\tag{1.21}$$

we can write equation (1.19) in the compact form

$$\langle \mathcal{O} \rangle = \sum_{\mathcal{S}} \prod_p \frac{1}{Z} o_n W_p(\mathcal{S}_p) ,\tag{1.22}$$

with the product being over all plaquettes.

For an originally two-dimensional spin system we can perform the exact same steps, except that we have to split the Hamiltonian into the four parts,

$$\begin{aligned}
\mathcal{H}_1 &= \sum_{x_1 \text{ even}} \sum_{x_2} h_{(x_1, x_2)(x_1+1, x_2)}, \\
\mathcal{H}_2 &= \sum_{x_1} \sum_{x_2 \text{ odd}} h_{(x_1, x_2)(x_1, x_2+1)}, \\
\mathcal{H}_3 &= \sum_{x_1 \text{ odd}} \sum_{x_2} h_{(x_1, x_2)(x_1+1, x_2)}, \\
\mathcal{H}_4 &= \sum_{x_1} \sum_{x_2 \text{ even}} h_{(x_1, x_2)(x_1, x_2+1)},
\end{aligned} \tag{1.23}$$

where a site on the lattice is denoted as $x = (x_1, x_2)$. The Euclidean time extended system is then (2+1)-dimensional and the plaquette division is no longer a simple checkerboard (see figure 1.2).

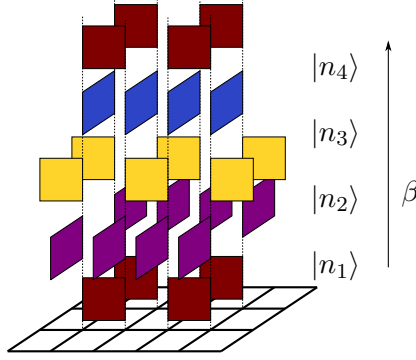


Figure 1.2.: Schematic picture of the plaquette decomposition of the (2+1)-dimensional extended system.

Performing the exact same steps as in (1+1) dimensions yields again

$$\langle \mathcal{O} \rangle = \sum_{\mathcal{S}} \prod_p \frac{1}{Z} o_n W_p(\mathcal{S}_p), \tag{1.24}$$

with the same plaquette weight function. The differences are hidden in the plaquette decomposition and the different sets of configurations. The same procedure could be applied to a system in an arbitrary number of dimensions.

If all entries of the matrix $\exp(-ah_{xy})$ are non-negative which is the case for the models considered here, as we will see later, $W_p(\mathcal{S}_p)$ will always be non-negative. We can then interpret the right-hand side of equation (1.24) as a probability distribution,

$$\langle \mathcal{O} \rangle = \sum_{\mathcal{S}} \frac{1}{Z} W(\mathcal{S}) o_n, \quad (1.25)$$

with $W(\mathcal{S}) = \prod_p W_p(\mathcal{S}_p)$ as the weight function of the whole system. It is properly normalized since

$$Z = \sum_{\mathcal{S}} W(\mathcal{S}) \quad (1.26)$$

The loop cluster algorithm provides a way to efficiently sample this probability distribution, and thus to calculate observables.

1.3. Sampling the Probability Density

The basic principle behind the loop cluster algorithm is to create a Markov chain, a procedure to create a new configuration out of a previous one, which follows certain probabilistic rules. The process of creating a new configuration will be called a Monte Carlo update or simply an update. If these updates are performed respecting the detailed balance condition (see below), the resulting configurations will be distributed according to the correct distribution.

One possibility to do this would be to use the Metropolis algorithm. As we will see, flipping a single spin results in a configuration with weight 0, and is thus forbidden. A possible way out would be using a Metropolis algorithm not with single spins but with plaquettes. This, however, turns out to be slow and inefficient. The loop cluster algorithm is a procedure to perform not only local updates as in a Metropolis algorithm, but global ones. A loop cluster update works in the following way:

1. The $(d+1)$ -dimensional lattice is divided into active plaquettes, according to the path integral derived in equation (1.16).
2. A breakup is assigned to all plaquettes. These breakups connect the spins of a plaquette pairwise. This is done in a probabilistic way, in order to fulfill the detailed balance condition.
3. From a random starting point x a loop is created, by following the breakups. With periodic boundary conditions this will always result in a closed loop.

Label	A	B	C	A'	B'	C'
Breakup						

Table 1.1.: The different breakups and their labels.

4. All spins in the loop are flipped simultaneously.

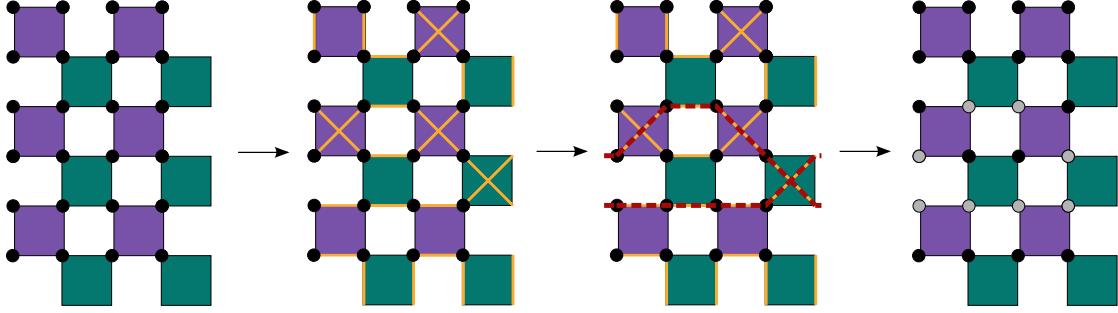


Figure 1.3.: Illustration of the 4 steps of a loop cluster update.

A crucial step here is step two, the assignment of the breakups according to the cluster rules. The cluster rules define which model is simulated. They ensure detailed balance and ergodicity. As we will see, there are six kinds of breakups. Two connecting the spins vertically (in the Euclidean time direction), called the A - and A' -breakup, two connecting horizontally, the B - and B' -breakup, and connecting diagonally, the C - respectively C' -breakup. The unprimed breakups connect parallel, and the primed antiparallel spins. One can think of these breakups as a new set of variables.

1.4. Detailed Balance

In the following we will motivate that the loop cluster algorithm fulfills detailed balance. From this we will be able to derive the cluster rules. To do so we start again at a very general level. The partition function of the considered spin system is given as

$$Z = \sum_{\mathcal{S}} W(\mathcal{S}), \quad (1.27)$$

where the sum is over all possible configurations of the system. The detailed balance condition can be stated as

$$W(\mathcal{S})p(\mathcal{S} \rightarrow \mathcal{S}') = W(\mathcal{S}')p(\mathcal{S}' \rightarrow \mathcal{S}), \quad (1.28)$$

where $p(\mathcal{S} \rightarrow \mathcal{S}')$ denotes the probability to go from configuration \mathcal{S} to configuration \mathcal{S}' in a Monte Carlo update. In addition to the spins we introduce the breakups as new variables. A configuration of the system is now described by the spin configuration \mathcal{S} and the breakup configuration \mathcal{G} .

A new weight function $W(\mathcal{S}, \mathcal{G})$ is chosen such that

$$W(\mathcal{S}) = \sum_{\mathcal{G}} W(\mathcal{S}, \mathcal{G}). \quad (1.29)$$

The detailed balance condition of this extended system is now

$$W(\mathcal{S}, \mathcal{G})p((\mathcal{S}, \mathcal{G}) \rightarrow (\mathcal{S}', \mathcal{G})) = W(\mathcal{S}', \mathcal{G})p((\mathcal{S}', \mathcal{G}) \rightarrow (\mathcal{S}, \mathcal{G})), \quad (1.30)$$

where we have restricted ourselves to updates changing only the spin but not the breakup configuration. As already mentioned above, updating the configuration of the system (\mathcal{S}) consists of two steps. To the system configuration \mathcal{S} a breakup configuration \mathcal{G} is chosen according to the probability

$$p(\mathcal{S} \rightarrow (\mathcal{S}, \mathcal{G})) = \frac{W(\mathcal{S}, \mathcal{G})}{W(\mathcal{S})}. \quad (1.31)$$

Then the spin configuration is updated according to the detailed balance condition, stated in (1.30). It turns out that if we chose this probability, detailed balance on the spin configuration level (equation (1.35)) is automatically fulfilled regardless of the chosen weight function for the breakups.

Say we start with the spin configuration \mathcal{S} and chose the breakup configuration \mathcal{G} . The whole lattice is now covered with breakups forming closed loops. Then one of the loops is chosen at random by choosing a starting point at random. The probability to choose a specific loop l depends thus only on the breakup configuration \mathcal{G} and not on the spin configuration \mathcal{S} . We denote it as $p_l(\mathcal{G})$. If we flip all spins in the loop l we end up at a different spin configuration \mathcal{S}' . The detailed balance condition for this scenario is

$$\begin{aligned} W(\mathcal{S}, \mathcal{G})p((\mathcal{S}, \mathcal{G}) \rightarrow (\mathcal{S}', \mathcal{G})) &= W(\mathcal{S}', \mathcal{G})p((\mathcal{S}', \mathcal{G}) \rightarrow (\mathcal{S}, \mathcal{G})), \\ W(\mathcal{S}, \mathcal{G})p_l(\mathcal{G})p_{\text{flip } l}(\mathcal{S}, \mathcal{G}) &= W(\mathcal{S}', \mathcal{G})p_l(\mathcal{G})p_{\text{flip } l}(\mathcal{S}', \mathcal{G}), \\ W(\mathcal{S}, \mathcal{G})p_{\text{flip } l}(\mathcal{S}, \mathcal{G}) &= W(\mathcal{S}', \mathcal{G})p_{\text{flip } l}(\mathcal{S}', \mathcal{G}). \end{aligned} \quad (1.32)$$

Detailed balance can be realized with the heat bath probability

$$p_{\text{flip } l}(S, \mathcal{G}) = \frac{W(\mathcal{S}', \mathcal{G})}{W(\mathcal{S}, \mathcal{G}) + W(\mathcal{S}', \mathcal{G}) + \text{const.}} \quad (1.33)$$

Since the weight function of the extended system is somewhat arbitrary we can impose

$$W(\mathcal{S}, \mathcal{G}) = W(\mathcal{S}', \mathcal{G}) \quad (1.34)$$

If we do this, the heat bath probability is particularly simple, namely constant, and we can set it to 1. So if we can (and we will) find a weight function which respects (1.34) the loop cluster algorithm will somewhat trivially fulfill detailed balance. The cluster rules are then given by (1.31).

Previously we derived an expression for $W(\mathcal{S})$. It turned out that one can write the weight of a configuration of the whole system as a product of weights of plaquette configurations,

$$W(\mathcal{S}) = \prod_p W_p(\mathcal{S}_p). \quad (1.35)$$

We can implement the setup described above locally on the level of plaquettes in this way,

$$\begin{aligned} W(\mathcal{S}, \mathcal{G}) &= \prod_p W_p(\mathcal{S}_p, \mathcal{G}_p), \\ W_p(\mathcal{S}_p) &= \sum_{\mathcal{G}_p} W_p(\mathcal{G}_p, \mathcal{S}_p), \\ p(\mathcal{S}_p \rightarrow (\mathcal{S}_p, \mathcal{G}_p)) &= \frac{W_p(\mathcal{S}_p, \mathcal{G}_p)}{W_p(\mathcal{S}_p)}, \\ W_p(\mathcal{S}_p, \mathcal{G}_p) &= W_p(\mathcal{S}'_p, \mathcal{G}) . \end{aligned} \quad (1.36)$$

Here \mathcal{S}_p is the spin configuration of the plaquette and \mathcal{G}_p the breakup assigned to the plaquette. The spin configuration weight function $W_p(\mathcal{S}_p)$ is specified by (1.21). If we choose a concrete basis for the spin states, $\exp(-ah_{xy})$ is represented by a 4×4 matrix. The weight function takes the entries of this matrix as values. As mentioned above, there are six different breakups called $A^{(\prime)}$ -, $B^{(\prime)}$ - and $C^{(\prime)}$ -breakup. \mathcal{G}_p is therefore either $A^{(\prime)}$, $B^{(\prime)}$ or $C^{(\prime)}$. With this we can express the weight function $W_p(\mathcal{S}_p, \mathcal{G}_p)$ as six different 4×4 matrices, one for each breakup. The unprimed breakups connect equal spins, and the primed only unequal spins. We will call these matrices $\mathbf{A}^{(\prime)}$, $\mathbf{B}^{(\prime)}$ and $\mathbf{C}^{(\prime)}$. Equations (1.36) will provide conditions on these matrices, namely:

1. The spin breakup weight after the spin flip must stay the same.
2. The matrices \mathbf{A} , \mathbf{B} and \mathbf{C} must add up to the matrix representing $\exp(-ah_p)$.

Condition 1 is somewhat trivially fulfilled. After flipping one or two branches of the breakup, the resulting spin configuration allows the same breakup, since the spins on each branch are still either parallel or antiparallel. Thus, all spin breakup configurations with an equal breakup having the same weight, is enough to fulfill this condition.

1.5. Deriving Cluster Rules

To find concrete cluster rules, one has to choose a basis for the spin configurations. The cluster rules for the considered models are obtained as follows. In section 1.2 we identified the plaquette spin configuration weight function as

$$W_p(\mathcal{S}_p) = \left\langle s_{x_p}, s_{y_p} \left| \exp(-ah_{x_p y_p}) \right| s_{x'_p}, s_{y'_p} \right\rangle, \quad (1.37)$$

with $\mathcal{S}_p = \{s_{x_p}, s_{x'_p}, s_{y_p}, s_{y'_p}\}$. To proceed further we choose the basis for the spin states in which S_x^3 is diagonal

$$S_x^1 = \frac{1}{2}\sigma^1 = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (1.38)$$

$$S_x^2 = \frac{1}{2}\sigma^2 = \frac{1}{2} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, \quad (1.39)$$

$$S_x^3 = \frac{1}{2}\sigma^3 = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.40)$$

This basis will be called the 3-basis. The basis vectors of a single spin Hilbert space are

$$|\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (1.41)$$

and for the product space of two spins

$$|\uparrow\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |\uparrow\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |\downarrow\uparrow\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |\downarrow\downarrow\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (1.42)$$

In this basis the single plaquette Hamiltonian becomes

$$h_{xy} = J(S_x^1 S_y^1 + S_x^2 S_y^2) + J' S_x^3 S_y^3 = \frac{1}{4} \begin{pmatrix} J' & 0 & 0 & 0 \\ 0 & -J' & 2J & 0 \\ 0 & 2J & -J' & 0 \\ 0 & 0 & 0 & J' \end{pmatrix}. \quad (1.43)$$

Exponentiating this matrix we find for the matrix representing the weight function of a single plaquette

$$\exp(-ah_{xy}) = \begin{pmatrix} e^{-aJ'} & 0 & 0 & 0 \\ 0 & e^{aJ'} \cosh(2aJ) & -e^{aJ'} \sinh(2aJ) & 0 \\ 0 & -e^{aJ'} \sinh(2aJ) & e^{aJ'} \cosh(2aJ) & 0 \\ 0 & 0 & 0 & e^{-aJ'} \end{pmatrix}. \quad (1.44)$$

Each entry of this matrix corresponds to the weight belonging to a spin configuration of a single plaquette. There are six non-zero entries, corresponding to the allowed plaquette configurations:

$$\begin{matrix} s_{x'_p} & s_{y'_p} \\ s_{x_p} & s_{y_p} \end{matrix} \in \left\{ \begin{matrix} \uparrow & \uparrow \\ \uparrow & \downarrow \\ \downarrow & \uparrow \\ \downarrow & \downarrow \end{matrix}, \begin{matrix} \uparrow & \downarrow \\ \downarrow & \uparrow \\ \downarrow & \downarrow \\ \uparrow & \downarrow \end{matrix}, \begin{matrix} \downarrow & \uparrow \\ \uparrow & \downarrow \\ \downarrow & \downarrow \\ \uparrow & \downarrow \end{matrix} \right\} \quad (1.45)$$

In this basis the breakup weight matrices are given as

$$\mathbf{A} = A \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B} = B \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{C} = C \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{A}' = A' \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}' = B' \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{C}' = C' \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \quad (1.46)$$

Condition 2. gives us equations for A, B, C, A', B', C' . We can set $A' = B = C' = 0$ and still find a consistent solution for the remaining weights. Doing this the equations from condition 2. reduce to

$$\begin{pmatrix} A+C & 0 & 0 & 0 \\ 0 & A+B' & B'+C & 0 \\ 0 & B'+C & A+B' & 0 \\ 0 & 0 & 0 & A+C \end{pmatrix} = \begin{pmatrix} e^{-aJ'} & 0 & 0 & 0 \\ 0 & e^{aJ'} \cosh(2aJ) & -e^{aJ'} \sinh(2aJ) & 0 \\ 0 & -e^{aJ'} \sinh(2aJ) & e^{aJ'} \cosh(2aJ) & 0 \\ 0 & 0 & 0 & e^{-aJ'} \end{pmatrix}. \quad (1.47)$$

Solving these equations for A, B' and C yields

$$\begin{aligned} A &= \frac{1}{2} \exp\left(-\frac{aJ'}{4}\right) + \frac{1}{2} \exp\left(\frac{aJ'}{4}\right) \exp\left(\frac{aJ}{2}\right), \\ B' &= -\frac{1}{2} \exp\left(-\frac{aJ'}{4}\right) + \frac{1}{2} \exp\left(\frac{aJ'}{4}\right) \exp\left(\frac{-aJ}{2}\right), \\ C &= \frac{1}{2} \exp\left(-\frac{aJ'}{4}\right) - \frac{1}{2} \exp\left(\frac{aJ'}{4}\right) \exp\left(\frac{aJ}{2}\right). \end{aligned} \quad (1.48)$$

From these weights we can now calculate the cluster rules. Since the matrix $\exp(-ah)$ is symmetric we can consider only three of the allowed plaquette configurations. The probabilities according to equation 1.36, the cluster rules, are summarized in table 1.2

Entering the weights we find:

$$\begin{aligned} \frac{A}{A+C} &= \frac{1 + \exp\left(\frac{a}{2}(J+J')\right)}{2}, & \frac{C}{A+C} &= \frac{1 - \exp\left(\frac{a}{2}(J+J')\right)}{2}, \\ \frac{A}{A+B'} &= \frac{\exp\left(\frac{aJ}{2}\right) + \exp\left(\frac{-aJ'}{2}\right)}{2 \cosh\left(\frac{aJ}{2}\right)}, & \frac{B'}{A+B'} &= \frac{\exp\left(\frac{aJ}{2}\right) - \exp\left(\frac{-aJ'}{2}\right)}{2 \cosh\left(\frac{aJ}{2}\right)}, \end{aligned}$$




\mathcal{S}_p	Probabilities			
				
$\uparrow \uparrow$ $\uparrow \uparrow$	$\frac{A}{A+C}$	0	$\frac{C}{A+C}$	
$\uparrow \downarrow$ $\uparrow \downarrow$	$\frac{A}{A+B'}$	$\frac{B'}{A+B'}$	0	
$\uparrow \downarrow$ $\downarrow \uparrow$	0	$\frac{B'}{B'+C}$	$\frac{C}{B'+C}$	

Table 1.2.: Cluster rules for the Euclidean time loop cluster algorithm in the 3-basis expressed through the weights A, B' and C . The weights are given in equations (1.48).

$$\frac{B'}{B' + C} = \frac{\exp\left(\frac{-aJ}{2}\right) - \exp\left(\frac{-aJ'}{2}\right)}{2 \sinh\left(\frac{aJ}{2}\right)}, \quad \frac{C}{B' + C} = \frac{-\exp\left(\frac{aJ}{2}\right) + \exp\left(\frac{-aJ'}{2}\right)}{2 \sinh\left(\frac{aJ}{2}\right)}. \quad (1.49)$$

It is not clear a priori that these expressions are probabilities and thus between 0 and 1. To guarantee this, the conditions

$$\begin{aligned} 0 \leq \frac{A}{A + C} \leq 1 &\Leftrightarrow 0 \leq \exp\left(\frac{a}{2}(J + J')\right) \leq 1 \\ &\Leftrightarrow J + J' \leq 0 \end{aligned} \quad (1.50)$$

and

$$\begin{aligned} 0 \leq \frac{A}{A + B'} \leq 1 &\Leftrightarrow \cosh\left(\frac{aJ}{2}\right) \geq \frac{\exp\left(\frac{Ja}{2}\right) + \exp\left(\frac{-aJ'}{2}\right)}{2} \\ &\Leftrightarrow \exp\left(\frac{-aJ}{2}\right) \geq \exp\left(\frac{-aJ'}{2}\right) \\ &\Leftrightarrow J \leq J' \end{aligned} \quad (1.51)$$

must be fulfilled. They reduce to

$$J \leq -|J'|. \quad (1.52)$$

It can easily be checked that this is sufficient to ensure that all probabilities are indeed between 0 and 1. Now this seems like a strong condition on the allowed Hamiltonians. This would exclude us from simulating an antiferromagnetic model, since J must be smaller than 0. However, on a bipartite lattice we can perform the transformation

$$S_x^1 \rightarrow -S_x^1, \quad S_x^2 \rightarrow -S_x^2, \quad S_x^3 \rightarrow S_x^3, \quad \forall x \text{ in one sublattice.} \quad (1.53)$$

This transformation is unitary and just a coordinate change. It does not affect S_x^3 and therefore also not its eigenstates, but it changes the sign of J in the Hamiltonian. Therefore the sign of J has no effect on the physics and condition (1.52) becomes

$$|J| \geq |J'|, \quad (1.54)$$

which is true for all models we are interested in here.

1.6. A Different Basis

Already in equation (1.8) we have used that the observable we want to simulate is diagonal in the basis we use. For the case of the XY-model the order parameter is the total spin in a direction inside the XY-plane. In the 3-basis used in the previous section this observable is not diagonal. We therefore have to reformulate the loop cluster algorithm in another basis. We choose the basis in which S_x^1 is diagonal. The unitary transformation

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} \quad (1.55)$$

maps σ^1 to σ^3 , σ^3 to σ^2 and σ^2 to σ^1 . In the previously used 3-basis S_x^1 is given as

$$S_x^1 = \frac{1}{2}\sigma^1 = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (1.56)$$

After the basis change S_x^1 is indeed diagonal, namely

$$US_x^1U^\dagger = \frac{1}{2}U\sigma^1U^\dagger = \frac{1}{2}\sigma^3. \quad (1.57)$$

We denote its eigenvectors as $|\rightarrow\rangle$ with eigenvalue $+1/2$ and $|\leftarrow\rangle$ with eigenvalue $-1/2$. In the two spin Hilbert space the basis change matrix is given as

$$\tilde{U} = U \otimes U = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ i & -i & i & -i \\ i & i & -i & -i \\ -1 & 1 & 1 & -1 \end{pmatrix}. \quad (1.58)$$

We can now simply transform the weight function matrices \mathbf{A} , \mathbf{B}' and \mathbf{C} as well as $\exp(-ah)$. Since it is a unitary transformation, the overall weight factors A , B' and C do not change. From $\tilde{U}(\mathbf{A} + \mathbf{B}' + \mathbf{C})\tilde{U}^\dagger = \tilde{U} \exp(-ah) \tilde{U}^\dagger$ we can read off the cluster rules, summarized in table 1.3. Note that $\tilde{U}\mathbf{A}\tilde{U}^\dagger = \mathbf{A}$, $\tilde{U}\mathbf{C}\tilde{U}^\dagger = \mathbf{C}$ but $\tilde{U}\mathbf{B}'\tilde{U}^\dagger \sim \mathbf{B}$. The B' breakup has become the B breakup under this coordinate change, with the weight B' . For simplicity we will call its weight now B as well. Thus in the 1-basis a loop always connects parallel spins. At this point we point out that in the simulation there is no difference between the primed and unprimed breakups. They are only introduced to give a consistent description of the loop cluster algorithm and to derive the cluster rules. The




\mathcal{S}_p		Probabilities		
				
\rightarrow	\rightarrow	$\frac{A}{A+B+C}$	$\frac{B}{A+B+C}$	$\frac{C}{A+B+C}$
\rightarrow	\leftarrow	1	0	0
\leftarrow	\rightarrow	0	0	1
\rightarrow	\leftarrow	0	1	0

Table 1.3.: Cluster rules for the Euclidean time loop cluster algorithm in the 1-basis. Explicit expressions for the probabilities are given in equations 1.59 .

explicit values for the probabilities now are:

$$\frac{A}{A+B+C} = \frac{\exp\left(\frac{-aJ'}{4}\right) + \exp\left(\frac{aJ'}{4} + \frac{aJ}{2}\right)}{\exp\left(\frac{-aJ'}{4}\right) + \exp\left(\frac{aJ'}{4} - \frac{aJ}{2}\right)}, \quad (1.59)$$

$$\frac{B}{A+B+C} = \frac{-\exp\left(\frac{-aJ'}{4}\right) + \exp\left(\frac{aJ'}{4} + \frac{aJ}{2}\right)}{\exp\left(\frac{-aJ'}{4}\right) + \exp\left(\frac{aJ'}{4} - \frac{aJ}{2}\right)}, \quad (1.60)$$

$$\frac{C}{A+B+C} = \frac{\exp\left(\frac{-aJ'}{4}\right) - \exp\left(\frac{aJ'}{4} + \frac{aJ}{2}\right)}{\exp\left(\frac{-aJ'}{4}\right) + \exp\left(\frac{aJ'}{4} - \frac{aJ}{2}\right)}. \quad (1.61)$$

1.7. Cluster Rules for the Considered Cases

In the previous section we have found consistent cluster rules for the quite general Hamiltonian $\mathcal{H} = \sum_{\langle x,y \rangle} J (S_x^1 S_y^1 + S_x^2 S_y^2) + J' S_x^3 S_y^3$. We are interested mainly in three special cases:

1. $J' = 0$ - the XY-model.
2. $J' = J < 0$ - the Heisenberg ferromagnet, abbreviated FM.
3. $J' = J > 0$ - the Heisenberg antiferromagnet, abbreviated AFM.

To have access to the order parameter, the XY-model will be quantized in the 1-direction, the FM and the AFM in the 3-direction. As mentioned before, in order to obtain consistent probabilities for the AFM model where $J > 0$ we will have to perform the transformation $J \rightarrow -J$. The resulting simplified cluster rules are summarized in table 1.4





\mathcal{S}_p	Probabilities			
	 A	 B	 C	 B'
XY				
$\rightarrow \rightarrow$	$e^{\frac{-aJ}{2}}$	$\tanh\left(\frac{aJ}{4}\right)$	$e^{\frac{-aJ}{2}} \tanh\left(\frac{aJ}{4}\right)$	0
$\rightarrow \leftarrow$	1	0	0	0
$\leftarrow \rightarrow$	0	0	1	0
$\leftarrow \leftarrow$	0	1	0	0
FM				
$\uparrow \uparrow$	$e^{\frac{aJ}{2}} \cosh\left(\frac{a J }{2}\right)$	0	$e^{\frac{aJ}{2}} \sinh\left(\frac{a J }{2}\right)$	0
$\uparrow \downarrow$	1	0	0	0
$\downarrow \uparrow$	0	0	1	0
AFM				
$\uparrow \uparrow$	1	0	0	0
$\uparrow \downarrow$	$1 - \tanh\left(\frac{aJ}{2}\right)$	0	0	$\tanh\left(\frac{aJ}{2}\right)$
$\downarrow \uparrow$	0	0	0	0

Table 1.4.: Summary of the cluster rules for the loop cluster algorithm in Euclidean time. The Breakups C' and A' do not occur and are therefore not listed in this table. Note that only half of the allowed plaquette configurations are listed. Flipping all spins results in a new allowed configuration with equal probabilities for the different breakups.

2. Real-Time Simulations

In this section we will define the loop cluster algorithm in real-time. First, we take a closer look at the problems one encounters with real-time simulations, then we will provide a very quick introduction to the theory of measurement in quantum mechanics and derive expressions for the evolution of the density matrix driven by measurement. These expressions are then converted into a path integral, which can be solved using the same efficient loop cluster algorithm as described in chapter 1. We then describe explicitly the measurement processes we simulate, and finally we discuss the conditions on a measurement process to make the sign problem disappear.

2.1. Problems of Real-Time Simulations

The real-time evolution of a closed quantum system is governed by the Schrödinger equation. The density matrix evolves as

$$\rho(t) = U(t, t_0) \rho(t_0) U^\dagger(t_0, t) . \quad (2.1)$$

For a time-independent Hamiltonian, \mathcal{H} , the unitary time evolution operator $U(t, t_0)$ is given as

$$U(t, t_0) = \exp(-i\mathcal{H}(t - t_0)) . \quad (2.2)$$

Thus for the time evolution of the expectation value of an observable \mathcal{O} , we obtain

$$\langle \mathcal{O} \rangle(t) = \text{Tr}(\mathcal{O} U(t, t_0) \rho(t_0) U^\dagger(t_0, t)) \quad (2.3)$$

This expression looks very similar to the starting point of the derivation of the Euclidean-time path integral, which can be simulated with the loop cluster algorithm. The main and crucial difference is the i factor in front of the Hamiltonian. If we would transform this expression to a path integral analogously, we would end up with a weight function taking entries of the $U(t, 0) = \exp(-i\mathcal{H}t)$ as values. Since these entries are complex, we cannot interpret them as probabilities and sample them in the same way. We encounter a

severe sign problem, which prevents us from performing numerical calculations on larger systems.

One reason for this is the fact that closed quantum systems tend to evolve into complicated entangled states (Schrödinger cat states). One possible way out is to simulate not closed but open quantum systems, which interact with the environment via a measurement process. This interaction usually decoheres the system. As described below it is possible to simulate such a process if we completely switch off the internal dynamics driven by the Hamiltonian.

2.2. Theory of Measurement in Quantum Mechanics

This section is meant to give a quick overview of the theory of measurement used in the formulation of the loop cluster algorithm in real-time. The material presented here is based on [6].

2.2.1. Orthogonal Measurements

In quantum mechanics an observable is described by a Hermitean Operator \mathcal{O} acting on the Hilbert Space of the System. The eigenvalues of \mathcal{O} , o_i , are the possible measurement outcomes. If a measurement is performed the state of the system is projected to an eigenstate of \mathcal{O} . \mathcal{P}_i is the projection operator projecting to the eigenstate with eigenvalue o_i . If the system is initially in a state $|\psi\rangle$, the outcome o_i is measured with probability

$$\text{Prob}(o_i) = \langle \psi | \mathcal{P}_i | \psi \rangle . \quad (2.4)$$

If we obtain the outcome o_i the state of the system after the measurement is

$$|\psi'\rangle = \frac{\mathcal{P}_i |\psi\rangle}{(\langle \psi | \mathcal{P}_i | \psi \rangle)^{1/2}} . \quad (2.5)$$

If the initial state is given by a density matrix ρ the measurement acts as

$$\rho' = \sum_i \mathcal{P}_i \rho \mathcal{P}_i . \quad (2.6)$$

Under this transformation, a pure state density matrix in general evolves into a mixed state density matrix. This represents the probabilistic nature of the measurement process.

2.2.2. Continuous Measurements

A quantum system interacting with its environment is described by the theory of open systems. See again [6] for more details. The setup of the processes we are interested in is as follows: We consider a (closed) quantum system with a Hilbert space which can be described by the tensor product Hilbert space

$$H = H_S \otimes H_E . \quad (2.7)$$

Here H_S is the Hilbert space of the system we are interested in, and H_E the Hilbert space of the environment. With the theory of open systems, we can describe how the density matrix of the system ($\rho = \text{Tr}_E \rho_{\text{tot}}$), evolves under processes including interactions between the system and the environment. We are interested in processes which fulfill the Markovian limit, namely processes taking place at a time scale much longer than the time scale over which the environment forgets information acquired from the system. This guarantees that the evolution of the density matrix is local in time. The dynamics of such processes can be described with a first order differential equation, the Lindblad equation

$$\dot{\rho} = -i [\mathcal{H}, \rho] + \sum_k \left(\mathcal{L}_k \rho \mathcal{L}_k^\dagger - \frac{1}{2} \mathcal{L}_k^\dagger \mathcal{L}_k \rho - \frac{1}{2} \rho \mathcal{L}_k^\dagger \mathcal{L}_k \right) . \quad (2.8)$$

The first term drives unitary evolution and corresponds to the standard Schrödinger equation. The \mathcal{L}_k 's in the second term are the so-called Lindblad or quantum jump operators. We can use this equation to describe a continuous measurement process using the Lindblad operators

$$\mathcal{L}_k = \sqrt{\gamma} \mathcal{P}_k , \quad (2.9)$$

where the \mathcal{P}_k 's are again the projection operators to the eigenstates of the observable. To obtain a simulatable process without a sign problem we set the Hamiltonian to zero. The real-time dynamics of the system are then entirely driven by measurements. Using this, the Lindblad equation simplifies to

$$\dot{\rho} = \gamma \left(\sum_k \mathcal{P}_k \rho \mathcal{P}_k^\dagger - \rho \right) . \quad (2.10)$$

To better interpret this process we can solve this equation for an infinitesimal time step δt and find

$$\rho(t + \delta t) = (1 - \delta t \gamma) \rho(t) + \delta t \gamma \sum_k \mathcal{P}_k \rho(t) \mathcal{P}_k^\dagger. \quad (2.11)$$

We can interpret this in the following way: With probability $1 - \delta t \gamma$ the system does nothing (the first term) and with probability $\delta t \gamma$ an orthogonal measurement is performed.

2.3. Simulating Measurement Processes

In section 2.2 we showed how discrete and continuous measurement processes act on a density matrix of a quantum system. These expressions can be transformed to a path integral, which can be simulated, if no sign problem occurs. We will look at two kinds of processes, discrete measurements and continuous measurements. In the discrete measurement process we start with an initial density matrix in thermal equilibrium and then evolve the system by performing a series of orthogonal measurements. The system Hamiltonian is switched off. In the continuous process we perform a continuous observation of the system according to the previously discussed Lindblad process. Here as well we have to switch off the Hamiltonian. The loop cluster algorithm in real-time was first introduced in [7]. This section is roughly based on [11].

2.3.1. Discrete Measurements

We start with an initial density matrix $\rho_i = \exp(-\beta \mathcal{H})$ in thermal equilibrium. After performing N_{rt} orthogonal measurements of some observables we arrive at the final density matrix

$$\rho_f = \sum_{o_1, \dots, o_N} \mathcal{P}_{o_N} \dots \mathcal{P}_{o_2} \mathcal{P}_{o_1} \rho_i \mathcal{P}_{o_1}^\dagger \mathcal{P}_{o_2}^\dagger \dots \mathcal{P}_{o_N}^\dagger. \quad (2.12)$$

The probability to end up in a final state $|f\rangle$, starting with initial density matrix $\rho_i = \sum_i p_i |i\rangle \langle i|$ is given as

$$p_{\rho_i f} = \sum_i \sum_{o_1, \dots, o_N} \langle f | \mathcal{P}_{o_N} \dots \mathcal{P}_{o_1} | i \rangle \langle i | \mathcal{P}_{o_1}^\dagger \dots \mathcal{P}_{o_N}^\dagger | f \rangle p_i. \quad (2.13)$$

This can be written as a tensor product

$$p_{\rho_i f} = \sum_i \sum_{o_1, \dots, o_N} \langle f f | \mathcal{P}_{o_N} \otimes \mathcal{P}_{o_N}^\dagger \dots \mathcal{P}_{o_1} \otimes \mathcal{P}_{o_1}^\dagger | i i \rangle p_i. \quad (2.14)$$

Defining $\tilde{\mathcal{P}}_k = \sum_{o_k} \mathcal{P}_{o_k} \otimes \mathcal{P}_{o_k}^\dagger$ and inserting complete sets of states we arrive at the expression

$$p_{\rho_i f} = \sum_i \sum_{\substack{n'_1, \dots, n'_{N-1} \\ n_1, \dots, n_{N-1}}} \langle f f | \tilde{\mathcal{P}}_N | n_{N-1} n'_{N-1} \rangle \dots \langle n_1 n_1 | \tilde{\mathcal{P}}_1 | i i \rangle p_i. \quad (2.15)$$

This expression can be viewed as a path integral along the Keldysh contour shown in figure 2.1. The path along the Euclidean time axis corresponds to the preparation of the initial state $|ii\rangle$ as described in chapter 1. Along the real-time axis, the states $|n_i n'_i\rangle$ encompass both branches of the Keldysh contour.

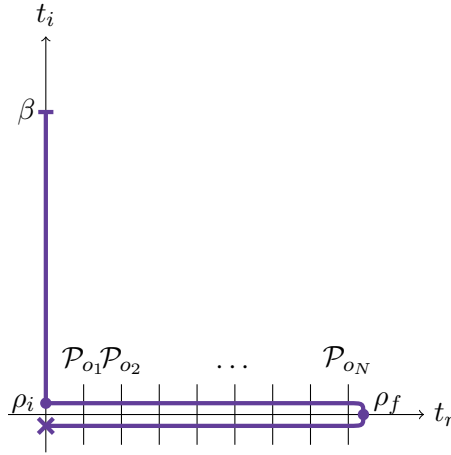


Figure 2.1.: The Keldysh contour of the real-time path integral. The real-time path is directly along the real-time axis, the offset is drawn to visualize that it goes forward and backward. The Euclidean time part along the imaginary time axis corresponds to a preparation of the initial state in thermal equilibrium. At each real-time step a measurement is performed visualized by the \mathcal{P}_k 's

The expectation value of an observable O of the entire system, (e.g. the 3-component of the total magnetization, $M = \sum_x S_x^3$) at time t is given as

$$\langle O \rangle = \sum_f o_f p_{\rho_i f}. \quad (2.16)$$

Here we have assumed that O is diagonal in the basis $\{|f\rangle\}$, and denoted its eigenvalues as o_f . We have now arrived at an expression similar to equation (1.16). The $\tilde{\mathcal{P}}_k$'s replace the $\exp(-a\mathcal{H}_i)$'s, and the Hilbert space is replaced by a larger tensor product Hilbert space. Up to now we have been very general. To proceed further we must specify the measured observables in more detail. In the derivation of the loop cluster algorithm in Euclidean

time, it was important to split the interaction up into terms involving only two spins. To proceed analogously we will consider measurements of pairs of spins, for example the total spin of two neighbouring spins. Then we measure the spin pairs according to the plaquette decomposition, as in Euclidean time, such that in a single measurement step each spin appears only in one measured pair. We will denote these spin pairs as $(s_{x,k}, s_{y,k})$ where x and y denote the spacial coordinates and k the measurement step. In 2 spacial dimensions this can be realized as follows: A single real-time step is divided into four measurement steps, so $k \in \{1, 2, \dots, 4N_{rt}\}$. At each measurement step spin pairs $(s_{x,k}, s_{y,k})$ and $(s'_{x,k}, s'_{y,k})$ with x and y fulfilling the respective condition C_k ,

$$\begin{aligned} C_k &= \{x = (x_1, x_2), y = (x_1 + 1, x_2) \text{ with } x_1 \text{ even}\} & \text{for } k \equiv 1 \pmod{4}, \\ C_k &= \{x = (x_1, x_2), y = (x_1, x_2 + 1) \text{ with } x_2 \text{ even}\} & k \equiv 2 \pmod{4}, \\ C_k &= \{x = (x_1, x_2), y = (x_1 + 1, x_2) \text{ with } x_1 \text{ odd}\} & k \equiv 3 \pmod{4}, \\ C_k &= \{x = (x_1, x_2), y = (x_1, x_2 + 1) \text{ with } x_2 \text{ odd}\} & k \equiv 0 \pmod{4}, \end{aligned} \quad (2.17)$$

are measured. $\tilde{\mathcal{P}}_k$ then factorizes to

$$\tilde{\mathcal{P}}_k = \prod_{\langle xy \rangle \in C_k} \tilde{\mathcal{P}}_{k,xy} \quad (2.18)$$

where the projectors on the right-hand side only act on the spins of a single plaquette, according to the scheme described above. Putting all this together yields

$$\langle \mathcal{O} \rangle = \sum_{i,f,\tilde{n},\tilde{n}'} \prod_{k=1}^{4N_{rt}} \prod_{\langle xy \rangle \in C_k} \text{of } p_i \left\langle s_{x,k-1} s_{y,k-1} s'_{x,k-1}, s'_{y,k-1} \left| \tilde{\mathcal{P}}_{k,xy} \right| s_{x,k} s_{y,k} s'_{x,k}, s'_{y,k} \right\rangle, \quad (2.19)$$

where we have introduced a compact notation for the sum over the sets of states

$$\begin{aligned} \tilde{n} &= \{n_1, n_2, \dots, n_{4N_{rt}-1}\}, \\ \tilde{n}' &= \{n'_1, n'_2, \dots, n'_{4N_{rt}-1}\}, \\ n_k &= \{s_{x_1,k}, s_{x_2,k}, \dots\}, \\ n'_k &= \{s'_{x_1,k}, s'_{x_2,k}, \dots\}, \\ i &= \{s_{x_1,0}, s_{x_2,0}, \dots\} = \{s'_{x_1,0}, s'_{x_2,0}, \dots\}, \\ f &= \{s_{x_1,4N_{rt}}, s_{x_2,4N_{rt}}, \dots\} = \{s'_{x_1,4N_{rt}}, s'_{x_2,4N_{rt}}, \dots\}. \end{aligned} \quad (2.20)$$

This expression is now completely analogous to the one derived in section 1.2. If all matrix elements of $\tilde{\mathcal{P}}_{k,xy}$ are non-negative, we can again interpret it as a probability density and sample it in the same way as the Euclidean time version, with the matrix elements of $\tilde{\mathcal{P}}_{k,xy}$ as the plaquette weight function. Concretely this is done by attaching a real-time lattice to both ends of the Euclidean time lattice, which is used to create the initial states $|i\rangle$ in thermal equilibrium. A Monte Carlo update of the full system consists of the following steps:

1. The Euclidean and real-time lattices are divided into active plaquettes.
2. A breakup is assigned to all active plaquettes. This is done according to the respective cluster rules. They are different in the real and Euclidean time part since the plaquette weight function is different.
3. From a random starting point x in the Euclidean time lattice a loop is created by following the breakups. When the loop hits the end of the Euclidean time lattice, it is continued in the real-time lattice. Periodic boundary conditions are used between the outer borders of the two attached real-time lattices.
4. All spins on the loop are flipped.

We can apply one more simplification. For the measurement processes that we study some matrix elements will vanish, namely

$$\left\langle s_{x,k-1}s_{y,k-1}s'_{x,k-1},s'_{y,k-1} \left| \tilde{\mathcal{P}}_{k,xy} \right| s_{x,k}s_{y,k}s'_{x,k},s'_{y,k} \right\rangle = 0, \quad (2.21)$$

for $s_{x,k-1} = s'_{x_{k-1}}$, $s_{y,k-1} = s'_{y_{k-1}}$ and $s_{x,k} \neq s'_{x_k}$ or $s_{y,k} \neq s'_{y_k}$. This means that plaquette configurations, where the configurations on the first time slice are equal in both branches of the Keldysh contour, and unequal on the second time slice, are forbidden. Since the initial time slice configuration $|ii\rangle$ is equal on both branches, the two branches are exact copies of each other. In the simulation we can therefore restrict us to one branch, the other is given as exact copy. Thus the operator $\tilde{\mathcal{P}}_k$ which is represented by a 16×16 matrix can be reduced to a 4×4 matrix, containing the entries corresponding to configurations which are equal on both branches of the Keldysh contour. We will denote this matrix as $\tilde{\mathcal{P}}_k^{\text{red}}$. Concretely this can be realized, by attaching only one real-time lattice. The loop is then first created in the Euclidean time lattice only, assuming periodic boundary conditions. From each point on the loop on the intersection between real and Euclidean time, a real-time loop is created, ending when it hits the last real-time slice. This is equivalent to the procedure described above, if the two real-time lattices are identical.

2.3.2. Continuous Measurements

Previously we have implemented the Lindblad equation for a finite time step. We can translate this solution to discrete time, using the lattice spacing ϵ in the real-time direction as δt . This yields

$$\rho(t + \epsilon) = (1 - \epsilon\gamma)\rho(t) + \epsilon\gamma \sum_k \mathcal{P}_k \rho(t) \mathcal{P}_k^\dagger \quad (2.22)$$

$$= \sum_k \left(\frac{1 - \sqrt{\epsilon\gamma}}{\sqrt{N_k}} \mathbb{1} + \sqrt{\epsilon\gamma} \mathcal{P}_k \right) \rho(t) \left(\frac{1 - \sqrt{\epsilon\gamma}}{\sqrt{N_k}} \mathbb{1} + \sqrt{\epsilon\gamma} \mathcal{P}_k \right). \quad (2.23)$$

Where N_k denotes the number of different projection operators \mathcal{P}_k . Iterating this equation and again using the tensor product notation of the previous section, we find this expression for the probability to reach a certain final state f :

$$p_{\rho_i f} = \sum_i \left\langle ff \left| \left((1 - \epsilon\gamma) \mathbb{1} \otimes \mathbb{1} + \epsilon\gamma \tilde{\mathcal{P}}_N \right) \right| n_{N-1} n'_{N-1} \right\rangle \dots \left\langle n_1 n'_1 \left| \left((1 - \epsilon\gamma) \mathbb{1} \otimes \mathbb{1} + \epsilon\gamma \tilde{\mathcal{P}}_1 \right) \right| ii \right\rangle p_i \quad (2.24)$$

This looks exactly like equation (2.15). Only the operator $\tilde{\mathcal{P}}_k$ is replaced by $(1 - \epsilon\gamma) \mathbb{1} \otimes \mathbb{1} + \epsilon\gamma \tilde{\mathcal{P}}_k$. From this point on we can proceed exactly like in the discrete measurement case.

2.4. Simulated Measurement Processes and their Cluster Rules

In this section we will discuss the three different measurement processes considered, and derive their cluster rules for the simulation of discrete measurements. As already mentioned, in each case the measurement acts on a pair of two spins. The sites of the measured pairs are denoted as x and y , their spin operators as $\vec{S}_x = (S_x^1, S_x^2, S_x^3)$. Simulating the processes will require the choice of a basis. We will choose the basis in which S_x^3 is diagonal, exactly as in the Euclidean time case. For each measurement process we will derive the reduced projection matrix $\tilde{\mathcal{P}}_k^{\text{red}}$. From this matrix we can then calculate the cluster rules.

2.4.1. Total Spin

The first observable we consider is the total spin of two neighbouring spins. The operator representing this observable is

$$\begin{aligned}\vec{S}^2 &= (\vec{S}_x + \vec{S}_y)^2 = \vec{S}_x^2 + \vec{S}_y^2 + 2\vec{S}_x \cdot \vec{S}_y \\ &= \frac{3}{2}\mathbb{1} + 2\vec{S}_x \cdot \vec{S}_y \\ &= \frac{3}{2}\mathbb{1} + 2(S_x^1 S_y^1 + S_x^2 S_y^2 + S_x^3 S_y^3) .\end{aligned}\tag{2.25}$$

This operator commutes with the components of the total spin vector $\vec{S} = \vec{S}_x + \vec{S}_y$.

$$[S^i, \vec{S}^2] = [S_x^i, \vec{S}^2] + [S_y^i, \vec{S}^2] = i\epsilon_{ijk}S_x^k S_y^j + i\epsilon_{ijk}S_y^k S_x^j = 0\tag{2.26}$$

This process therefore conserves all components of the total spin of two neighbouring spins and thus also the 3-component of the total magnetization of the full system, the order parameter of the FM. The operators \vec{S}^2 and S^3 can be simultaneously be diagonalized. We denote the eigenstates of \vec{S}^2 by their eigenvalues as $|SS^3\rangle$. There is the singlet state belonging to the eigenvalue 0,

$$|00\rangle = (|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle) \frac{1}{\sqrt{2}},\tag{2.27}$$

and the triplet states with eigenvalue 1,

$$\begin{aligned}|11\rangle &= |\uparrow\uparrow\rangle , \\ |10\rangle &= (|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle) \frac{1}{\sqrt{2}}, \\ |1-1\rangle &= |\downarrow\downarrow\rangle .\end{aligned}\tag{2.28}$$

Expressed as matrices in the basis $\{|\uparrow\uparrow\rangle, |\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle, |\downarrow\downarrow\rangle\}$ one obtains

$$\vec{S}^2 = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix},\tag{2.29}$$

$$|00\rangle = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix} \frac{1}{\sqrt{2}}, \quad |11\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \frac{1}{\sqrt{2}}, \quad |1-1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (2.30)$$

Since there are two different eigenvalues, there will be two distinct projection operators, namely

$$\mathcal{P}_0 = |00\rangle \langle 00| = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.31)$$

and

$$\mathcal{P}_1 = |11\rangle \langle 11| + |10\rangle \langle 10| + |1-1\rangle \langle 1-1| = \frac{1}{2} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}. \quad (2.32)$$

From these matrices we calculate $\tilde{\mathcal{P}} = \mathcal{P}_0 \otimes \mathcal{P}_0 + \mathcal{P}_1 \otimes \mathcal{P}_1$. We point out that even though the matrix $\mathcal{P}_0 \otimes \mathcal{P}_0$ has negative entries, the matrix $\tilde{\mathcal{P}}$ is non-negative,

$$\tilde{\mathcal{P}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.33)$$

One can also see that indeed all entries corresponding to transitions from states with equal configurations on both branches of the Keldysh contour to unequal configurations are zero. We can thus reduce the matrix to

$$\tilde{\mathcal{P}}^{\text{red}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.34)$$

From this matrix we can derive the cluster rules exactly as in the Euclidean time case. In fact we only need the breakups A and C to find consistent cluster rules. Solving $\tilde{\mathcal{P}}^{\text{red}} = \mathbf{A} + \mathbf{C}$ yields $A = \frac{1}{2}$ and $C = \frac{1}{2}$ for the weights.

Note that for the XY-model initial state we work in a different basis. There we use the 1-direction as quantization axis. However, the matrix representing \vec{S}^2 remains invariant under the basis change U . We can therefore use the cluster rules derived here for the 3-basis also in the 1-basis to simulate the process \vec{S}^2 .

2.4.2. $S_x^1 S_y^1$

The second measurement process we consider corresponds to measuring the product of the 1-component of two neighbouring spins. The observable is represented by the operator $S_x^1 S_y^1$. It has the eigenvalues $1/4$ and $-1/4$ for the parallel or anti-parallel spins in the 1-direction. In the chosen 3-basis these states are given by

$$||1\rangle = \frac{1}{\sqrt{2}} (|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle), \quad ||2\rangle = \frac{1}{\sqrt{2}} (|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle), \quad (2.35)$$

$$|\kern-0.25ex|\kern-0.25ex|1\rangle = \frac{1}{\sqrt{2}} (|\uparrow\uparrow\rangle - |\downarrow\downarrow\rangle), \quad |\kern-0.25ex|\kern-0.25ex|2\rangle = \frac{1}{\sqrt{2}} (|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle). \quad (2.36)$$

These states are sometimes referred to as the Bell states $|\phi^\pm\rangle$ and $|\psi^\pm\rangle$. They correspond to the maximally entangled states of two spins. As matrices they are given by

$$S_x^1 S_y^1 = \frac{1}{4} \sigma^1 \otimes \sigma^1 = \begin{pmatrix} 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 \end{pmatrix} \quad (2.37)$$

$$||1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad ||2\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad |\kern-0.25ex|\kern-0.25ex|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}, \quad |\kern-0.25ex|\kern-0.25ex|2\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}. \quad (2.38)$$

For the projectors we find

$$\mathcal{P}_{\frac{1}{4}} = ||1\rangle \langle ||1| + ||2\rangle \langle ||2| = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad (2.39)$$

$$\mathcal{P}_{-\frac{1}{4}} = |\mathbb{H}_1\rangle \langle \mathbb{H}_1| + |\mathbb{H}_2\rangle \langle \mathbb{H}_2| = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}. \quad (2.40)$$

The large matrix $\tilde{\mathcal{P}} = \mathcal{P}_{\frac{1}{4}} \otimes \mathcal{P}_{\frac{1}{4}} + \mathcal{P}_{-\frac{1}{4}} \otimes \mathcal{P}_{-\frac{1}{4}}$ is then given by

$$\tilde{\mathcal{P}} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}. \quad (2.41)$$

Here again only configurations with equal spins on both branches of the Keldysh contour contribute. We can reduce this matrix to the relevant matrix

$$\tilde{\mathcal{P}}^{\text{red}} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad (2.42)$$

To find consistent cluster rules we need the breakups A, C, A' and C' . Solving $\tilde{\mathcal{P}}^{\text{red}} = \mathbf{A} + \mathbf{B} + \mathbf{A}' + \mathbf{C}'$ for the weights yields $A = C = A' = C' = 1/4$. The resulting probabilities are summarized in table 2.1. We point out that this process does not conserve the 3-component of the total magnetization nor the 3- component of the staggered magnetization,

$$\begin{aligned} [S_x^1 S_y^1, S_x^3 + S_y^3] &= -i(S_x^2 S_y^1 + S_x^1 S_y^2), \\ [S_x^1 S_y^1, S_x^3 - S_y^3] &= -i(S_x^2 S_y^1 - S_x^1 S_y^2). \end{aligned} \quad (2.43)$$

We will use the cluster rules derived here also for the XY-model initial state although, there we work in a different basis. The simulated measurement process for the XY-model therefore corresponds to a different measurement process, namely to $S_x^2 S_y^2$. However, we will refer to it generally as $S_x^1 S_y^1$.

2.4.3. $S_x^+ S_y^+ + S_x^- S_y^-$

The last measurement process that we consider can be expressed via the spin raising and lowering operators $S_x^\pm = S_x^1 \pm iS_x^2$ as

$$S_x^+ S_y^+ + S_x^- S_y^- = 2(S_x^1 S_y^1 - S_x^2 S_y^2). \quad (2.44)$$

Again there is a quantity conserved by this process, namely $S_x^3 - S_y^3$ which is the contribution of a single spin pair to the staggered magnetization, the order parameter of the AFM,

$$\begin{aligned} [S_x^+ S_y^+ + S_x^- S_y^-, S_x^3 - S_y^3] &= [2(S_x^1 S_y^1 - S_x^2 S_y^2), S_x^3 - S_y^3] \\ &= 2([S_x^1 S_y^1, S_x^3 - S_y^3] - [S_x^2 S_y^2, S_x^3 - S_y^3]) \\ &= 2i(-S_x^2 S_y^1 + S_x^1 S_y^2 - S_x^1 S_y^2 + S_x^2 S_y^1) = 0. \end{aligned} \quad (2.45)$$

It has three eigenvalues, 1, -1 and 0, where there are two eigenstates with the eigenvalue 0. The corresponding eigenstates are

$$|+\rangle = \frac{1}{\sqrt{2}}(|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|\uparrow\uparrow\rangle - |\downarrow\downarrow\rangle), \quad (2.46)$$

$$|0_+\rangle = |\uparrow\downarrow\rangle, \quad |0_-\rangle = |\downarrow\uparrow\rangle \quad (2.47)$$

or in matrix form

$$S_x^+ S_y^+ + S_x^- S_y^- = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad (2.48)$$

$$|+\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad |-\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad |0_+\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |0_-\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}. \quad (2.49)$$

The three projectors are then given by

$$\mathcal{P}_1 = |+\rangle \langle +|, \quad \mathcal{P}_{-1} = |-\rangle \langle -|, \quad \mathcal{P}_{-1} = |0_+\rangle \langle 0_+| + |0_-\rangle \langle 0_-|, \quad (2.50)$$

or expressed as matrices by

$$\mathcal{P}_1 = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{P}_{-1} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{P}_{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2.51)$$

The matrix representing $\tilde{\mathcal{P}} = \mathcal{P}_1 \otimes \mathcal{P}_1 + \mathcal{P}_{-1} \otimes \mathcal{P}_{-1} + \mathcal{P}_0 \otimes \mathcal{P}_0$ is

$$\tilde{\mathcal{P}} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}. \quad (2.52)$$

Here again we can restrict ourselves to one branch of the Keldysh contour. The reduced matrix is

$$\tilde{\mathcal{P}}^{\text{red}} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}. \quad (2.53)$$

It is possible to find consistent cluster rules using the breakups A and C' . Solving equation $\tilde{\mathcal{P}}^{\text{red}} = \mathbf{A} + \mathbf{C}'$ for the weights yields $A = C' = \frac{1}{2}$. For the cluster rules see again table 2.1. Again we point out that for the XY-model initial state we use a different basis, and the cluster rules derived above correspond to a different measurement process. Under the basis change from the 3-basis to the 1-basis, the operators $S_x^\pm = \frac{1}{2}(S_x^1 \pm iS_x^2)$ transform to $S_x'^\pm = \frac{1}{2}(S_x^2 \pm iS_x^3)$ and with that the operator $S_x^+ S_y^+ + S_x^- S_y^-$ to $S_x'^+ S_y'^+ + S_x'^- S_y'^-$. However, we will again generally refer to it as $S_x^+ S_y^+ + S_x^- S_y^-$.

2.5. Cluster Rules for Continuous Measurement Processes

In the previous section we have seen how to find cluster rules for the different measurement processes for simulating discrete measurements. As derived in section 2.3.2 in the simulation of the continuous process the matrix $\tilde{\mathcal{P}}$ is replaced with $(1 - \epsilon\gamma) \mathbb{1} \otimes \mathbb{1} + \epsilon\gamma\tilde{\mathcal{P}}$. The reduced matrix which we can consider to find cluster rules is therefore $(1 - \epsilon\gamma) \mathbb{1} + \epsilon\gamma\tilde{\mathcal{P}}^{\text{red}}$. This will result in a change of the weights of the different breakups.

For the process \vec{S}^2 solving $(1 - \gamma\epsilon)\mathbb{1} + \epsilon\gamma\tilde{\mathcal{P}}^{\text{red}} = \mathbf{A} + \mathbf{C}$ for the weights yields

$$A = 1 - \frac{\epsilon\gamma}{2}, \quad C = \frac{\epsilon\gamma}{2}. \quad (2.54)$$

For $S_x^1 S_y^1$ we find for the solution of $(1 - \gamma\epsilon)\mathbb{1} + \epsilon\gamma\tilde{\mathcal{P}}^{\text{red}} = \mathbf{A} + \mathbf{A}' + \mathbf{C} + \mathbf{C}'$

$$A = 1 - \frac{3\epsilon\gamma}{4}, \quad A' = \frac{\epsilon\gamma}{4}, \quad C = \frac{\epsilon\gamma}{4}, \quad C' = \frac{\epsilon\gamma}{4}. \quad (2.55)$$

In the case of $S_x^+ S_y^+ + S_x^- S_y^-$ we have to solve $(1 - \gamma\epsilon)\mathbb{1} + \epsilon\gamma\tilde{\mathcal{P}}^{\text{red}} = \mathbf{A} + \mathbf{C}'$ and find for the weights

$$A = 1 - \frac{\epsilon\gamma}{2}, \quad C' = \frac{\epsilon\gamma}{2}. \quad (2.56)$$

The cluster rules for the continuous measurement simulations are also summarized in table 2.1.

2.6. What can be Simulated

Previously we discussed the three measurement processes we study in the numerical simulations. For all these processes the matrices $\tilde{\mathcal{P}}$ has only non-negative entries. This is the crucial feature for a measurement process to be simulatable. However for a general measurement process, this is not the case and simulations suffer from a sign or even complex phase problem. In this section we will investigate for what kind of measurement processes the sign problem disappears.

As discussed in section 2.2, a measurement is characterized by an Hermitean operator \mathcal{O} . In this case it acts on the Hilbert space of two spins which is isomorphic to \mathbb{C}^4 and can therefore be represented by a 4×4 Hermitean complex matrix. A general Hermitean matrix can be parametrized as

$$H = \sum_{i=0}^{15} a_i \lambda_i, \quad (2.57)$$









\mathcal{S}_p	Discrete Measurements				Continuous Measurements			
								
	A	C	A'	C'	A	C	A'	C'
\vec{S}^2								
$\uparrow \uparrow$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$1 - \frac{\epsilon\gamma}{2}$	$\frac{\epsilon\gamma}{2}$	0	0
$\uparrow \downarrow$	1	0	0	0	1	0	0	0
$\downarrow \uparrow$	0	1	0	0	0	1	0	0
$\downarrow \downarrow$	0	0	1	1	0	0	1	1
$S_x^1 S_y^1$								
$\uparrow \uparrow$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{\epsilon\gamma-2} + \frac{3}{2}$	0	$\frac{\epsilon\gamma}{4-2\epsilon\gamma}$	0
$\uparrow \downarrow$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$
$\downarrow \uparrow$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{3}{2} + \frac{1}{\epsilon\gamma-2}$	0	0	$\frac{\epsilon\gamma}{4-2\epsilon\gamma}$
$\downarrow \downarrow$	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0
$S_x^+ S_y^+ + S_x^- S_y^-$								
$\uparrow \uparrow$	1	0	0	0	1	0	0	0
$\uparrow \downarrow$	0	0	0	1	0	0	0	1
$\downarrow \uparrow$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$1 - \frac{\epsilon\gamma}{2}$	0	0	$\frac{\epsilon\gamma}{2}$
$\downarrow \downarrow$	0	0	1	1	0	0	1	1

Table 2.1.: Summary of cluster rules for the real-time loop cluster algorithm.

where λ_i are the Gellmann matrices and λ_0 is the identity matrix. The Hermitean 4×4 matrices form a 16-dimensional vector space. Hermitean matrices can be unitarily diagonalized, therefore their eigenvectors form an orthonormal basis. In addition they have real eigenvalues. The matrix $\tilde{\mathcal{P}}$ depends only on the eigenvectors and the degree of degeneracy of the eigenvalues of \mathcal{O} . We can therefore get some insight about the structure of the $\tilde{\mathcal{P}}$'s by looking not at different Hermitean matrices but at orthonormal bases of \mathbb{C}^4 .

2.6.1. The \mathbb{R}^2 Case

For simplicity let us first consider the problem not on \mathbb{C}^4 but on \mathbb{R}^2 . Each orthonormal basis is here of the form

$$v_1 = \begin{pmatrix} \sin(\alpha) \\ \cos(\alpha) \end{pmatrix}, \quad v_2 = \begin{pmatrix} \cos(\alpha) \\ -\sin(\alpha) \end{pmatrix}, \quad (2.58)$$

for $\alpha \in [0, 2\pi]$. The corresponding projection matrices $\mathcal{P}_i (= v_i v_i^\top)$ are:

$$\mathcal{P}_1 = \begin{pmatrix} \sin^2(\alpha) & \cos(\alpha) \sin(\alpha) \\ \cos(\alpha) \sin(\alpha) & \cos^2(\alpha) \end{pmatrix}, \quad (2.59)$$

$$\mathcal{P}_2 = \begin{pmatrix} \cos^2(\alpha) & -\cos(\alpha) \sin(\alpha) \\ -\cos(\alpha) \sin(\alpha) & \sin^2(\alpha) \end{pmatrix}. \quad (2.60)$$

As expected, their sum is the identity matrix. Only if v_1 and v_2 belong to different eigenvalues the resulting matrix $\tilde{\mathcal{P}} = \mathcal{P}_1 \otimes \mathcal{P}_1 + \mathcal{P}_2 \otimes \mathcal{P}_2$ will be non-trivial, namely

$$\tilde{\mathcal{P}} = \begin{pmatrix} \cos^4(\alpha) + \sin^4(\alpha) & -\frac{1}{4} \sin(4\alpha) & -\frac{1}{4} \sin(4\alpha) & 2 \cos^2(\alpha) \sin^2(\alpha) \\ -\frac{1}{4} \sin(4\alpha) & 2 \cos^2(\alpha) \sin^2(\alpha) & 2 \cos^2(\alpha) \sin^2(\alpha) & \frac{1}{4} \sin(4\alpha) \\ -\frac{1}{4} \sin(4\alpha) & 2 \cos^2(\alpha) \sin^2(\alpha) & 2 \cos^2(\alpha) \sin^2(\alpha) & \frac{1}{4} \sin(4\alpha) \\ 2 \cos^2(\alpha) \sin^2(\alpha) & \frac{1}{4} \sin(4\alpha) & \frac{1}{4} \sin(4\alpha) & \cos^4(\alpha) + \sin^4(\alpha) \end{pmatrix} \quad (2.61)$$

The only terms in this matrix, which could become negative are the terms $\pm \frac{1}{4} \sin(4\alpha)$. All others consist only of squares and fourth powers and will therefore always be non-

negative. The remaining condition on α for the matrix to have only non-negative elements is:

$$\pm \frac{1}{4} \sin(4\alpha) \geq 0 \quad (2.62)$$

which is equivalent to

$$\sin(4\alpha) = 0 \quad (2.63)$$

and therefore

$$\alpha = \frac{n\pi}{4} \quad n \in 1, 2, \dots, 8 \quad (2.64)$$

The only possible bases (modulo exchange of v_1 and v_2 and overall minus signs, which both have no influence on the resulting matrix $\tilde{\mathcal{P}}_i$) are then:

$$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (2.65)$$

and

$$v_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad v_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \quad (2.66)$$

The matrix $\tilde{\mathcal{P}}$ corresponding to the first basis is diagonal and thus not interesting for our purposes. This leaves us with one single non-diagonal $\tilde{\mathcal{P}}$ with non-negative entries:

$$\tilde{\mathcal{P}} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}. \quad (2.67)$$

Let us now generalize this result to the case of $\mathcal{H} = \mathbb{C}^2$. The general orthonormal basis of \mathbb{C}^2 can be written as

$$v_1 = \begin{pmatrix} \sin(\alpha) \exp(i\phi) \\ \cos(\alpha) \exp(i\psi) \end{pmatrix} \quad v_2 = \begin{pmatrix} \cos(\alpha) \exp(i\mu) \\ -\sin(\alpha) \exp(i(\mu + \psi - \phi)) \end{pmatrix} \quad (2.68)$$

However, the overall phase of a vector has no influence on the corresponding subspace, and therefore also not on the corresponding projection matrix \mathcal{P}_i . We can restrict ourselves

to the cases $\phi = \mu = 0$. Again the sum $\mathcal{P}_1 + \mathcal{P}_2$ is the identity matrix, and the only non-trivial matrix $\tilde{\mathcal{P}}$ will, of course, be $\mathcal{P}_1 \otimes \mathcal{P}_1 + \mathcal{P}_2 \otimes \mathcal{P}_2$, given as

$$\tilde{\mathcal{P}} = \begin{pmatrix} \cos^4(\alpha) + \sin^4(\alpha) & -\frac{1}{4}e^{-i\psi} \sin(4\alpha) & -\frac{1}{4}e^{-i\psi} \sin(4\alpha) & 2e^{-2i\psi} \cos^2(\alpha) \sin^2(\alpha) \\ -\frac{1}{4}e^{i\psi} \sin(4\alpha) & 2\cos^2(\alpha) \sin^2(\alpha) & 2\cos^2(\alpha) \sin^2(\alpha) & \frac{1}{4}e^{-i\psi} \sin(4\alpha) \\ -\frac{1}{4}e^{i\psi} \sin(4\alpha) & 2\cos^2(\alpha) \sin^2(\alpha) & 2\cos^2(\alpha) \sin^2(\alpha) & \frac{1}{4}e^{-i\psi} \sin(4\alpha) \\ 2e^{2i\psi} \cos^2(\alpha) \sin^2(\alpha) & \frac{1}{4}e^{i\psi} \sin(4\alpha) & \frac{1}{4}e^{i\psi} \sin(4\alpha) & \cos^4(\alpha) + \sin^4(\alpha) \end{pmatrix}. \quad (2.69)$$

The condition “all entries of $\tilde{\mathcal{P}} \in \mathbb{R}$ ” is then equivalent to $\psi = n\pi$ with $n \in \mathbb{Z}$. For n even, we have the exact same case as in the discussion for \mathbb{R}^2 . For n odd all potentially negative entries of $\tilde{\mathcal{P}}$ change sign. However this does not affect the condition on α to make all entries of $\tilde{\mathcal{P}}$ non-negative. Therefore we arrive again at the same result as in \mathbb{R}^2 (modulo a complex phase for each basis vector, which again does not affect the resulting matrix $\tilde{\mathcal{P}}$).

This result is somewhat remarkable. There exists only one matrix $\tilde{\mathcal{P}}$ over \mathbb{C}^2 which has only real and non-negative entries and which is not diagonal. Namely the matrix specified in equation (2.67).

2.6.2. The \mathbb{C}^4 Case

The problem which is really of interest is however not over \mathbb{C}^2 but \mathbb{C}^4 . A straightforward calculation as in the \mathbb{C}^2 case could in principle be done but results in an unresolvable system of inequalities. In the \mathbb{C}^2 case we found only one single matrix $\tilde{\mathcal{P}}$. This suggests that over \mathbb{C}^4 there could be only a limited amount of solutions, perhaps even finitely many. This is confirmed by a random sampling investigation. Out of 30'000 randomly selected Hermitean 4×4 matrices constructed from randomly chosen bases of \mathbb{R}^4 , none had strictly non-negative entries (see figure 2.2). The general basis is constructed according to the iterative scheme described in [12]. It is parametrized by 12 angles and 12 complex phases. The complex phases were all set to zero and the angles were sampled from a uniform distribution.

So let us try a different approach. Each projection matrix \mathcal{P}_i can be written as

$$\mathcal{P}_i = U D_i U^\dagger \quad (2.70)$$

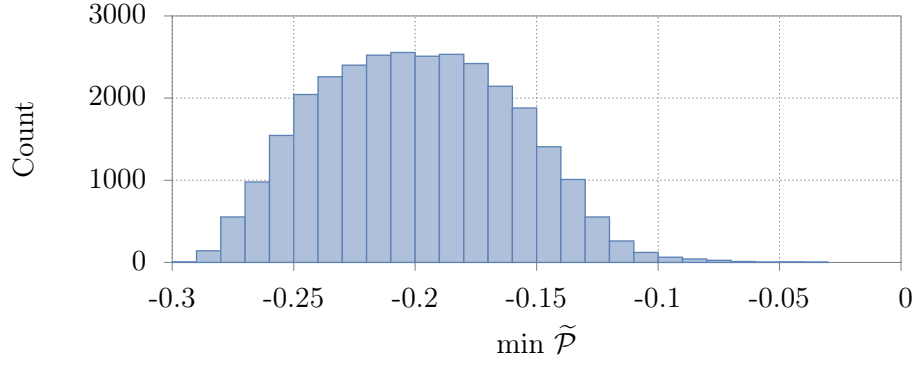


Figure 2.2.: Histogram of the minimal entry of 30'000 randomly chosen matrices $\tilde{\mathcal{P}}$.

with U being the unitary matrix of eigenvectors of \mathcal{O} , and D_i a diagonal matrix with a 1 on the diagonal at each position corresponding to the positions in U of the eigenvectors belonging to the subspace \mathcal{P}_i is projecting to

$$(D_i)_{\alpha\beta} = \sum_{j \text{ s.t. } \mathcal{O}v_j = o_i v_j} \delta_{\alpha j} \delta_{\beta j}. \quad (2.71)$$

Thus we get

$$\begin{aligned} \tilde{\mathcal{P}} &= \sum_i (U D_i U^\dagger) \otimes (U D_i U^\dagger) \\ &= \sum_i (U \otimes U) (D_i \otimes D_i) (U^\dagger \otimes U^\dagger). \end{aligned} \quad (2.72)$$

To further examine this matrix let us introduce some notation. v_j will denote the j -th eigenvector of the observable \mathcal{O} , o_i will denote the different eigenvalues of \mathcal{O} , and $I_i := \{j : \mathcal{O}v_j = o_i v_j\}$ the set of indices of the eigenvectors to the i -th eigenvalue. Furthermore $u_{\alpha\beta}$ denote the coefficients of U and thus $U_{\alpha\beta}^\dagger = u_{\beta\alpha}^*$. With this we can write

$$(\mathbf{D}_i)_{\alpha\beta} = \sum_{j \in I_i} \delta_{\alpha j} \delta_{\beta j}. \quad (2.73)$$

If we now write out the tensor product for $\tilde{\mathcal{P}}$ in index notation we obtain

$$(\tilde{\mathcal{P}})_{\alpha\beta\gamma\delta} = \sum_i (U)_{\alpha\mu} (U)_{\beta\nu} (D_i)_{\mu\lambda} (D_i)_{\nu\eta} (U)_{\lambda\gamma}^\dagger (U)_{\eta\delta}^\dagger$$

$$\begin{aligned}
&= \sum_i u_{\alpha\mu} u_{\beta\nu} \left(\sum_{j \in I_i} \delta_{\mu j} \delta_{\lambda j} \delta_{\nu j} \delta_{\eta j} \right) u_{\gamma\lambda}^* u_{\delta\eta}^* \\
&= \sum_i u_{\alpha\mu} u_{\beta\nu} \delta_{\mu i} \delta_{\lambda i} \delta_{\nu i} \delta_{\eta i} u_{\gamma\lambda}^* u_{\delta\eta}^* + \\
&\quad \sum_{\substack{i,j \text{ s. t. } o_i=o_j \\ i \neq j}} u_{\alpha\mu} u_{\beta\nu} \delta_{\mu i} \delta_{\lambda i} \delta_{\nu j} \delta_{\eta j} u_{\gamma\lambda}^* u_{\delta\eta}^* \\
&= \sum_i u_{\alpha i} u_{\beta i} u_{\gamma i}^* u_{\delta i}^* + \sum_{\substack{i,j \text{ s. t. } o_i=o_j \\ i \neq j}} u_{\alpha i} u_{\beta i} u_{\gamma j}^* u_{\delta j}^*. \tag{2.74}
\end{aligned}$$

We arrive at a quite compact form for the elements of the matrix $\tilde{\mathcal{P}}$. These four indices are related to the position of the matrix element in the 16×16 matrix $\tilde{\mathbf{P}}$ as follows: The indices α and γ denote the position of the 4×4 block in the large matrix and the indices β and δ the position inside the 4×4 block. For simplicity let us now examine the case of no degeneracies in the spectrum, where the second sum does not contribute. Here we have

$$(\tilde{\mathcal{P}})_{\alpha\beta\gamma\delta} = \sum_i u_{\alpha i} u_{\beta i} u_{\gamma i}^* u_{\delta i}^*. \tag{2.75}$$

Even in this form the 16×16 inequalities

$$(\tilde{\mathcal{P}})_{\alpha\beta\gamma\delta} = \sum_i u_{\alpha i} u_{\beta i} u_{\gamma i}^* u_{\delta i}^* \geq 0, \tag{2.76}$$

can still not be simultaneously resolved. However we can find special solutions in the following way. Let us consider only the entries for which $\alpha = \gamma$. For these the inequalities read

$$(\tilde{\mathcal{P}})_{\alpha\beta\alpha\delta} = \sum_i |u_{\alpha i}|^2 u_{\beta i} u_{\delta i}^* \geq 0. \tag{2.77}$$

For $\beta = \gamma$ this holds trivially. The entries $\beta \neq \gamma$ give weight to transitions from equal to unequal configurations on the two branches of the real-time Keldysh contour. If we impose

$$(\tilde{\mathcal{P}})_{\alpha\beta\alpha\delta} = 0 \quad \text{for} \quad \beta \neq \delta, \tag{2.78}$$

the only entries which matter in the simulation are those for which $\alpha = \gamma$ and $\beta = \delta$, and these are non-negative anyway. The sign problem disappears trivially. A basis which fulfills this condition is

$$u_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad u_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}, \quad u_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad u_4 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}. \quad (2.79)$$

Remarkably, these are simultaneous eigenvectors of $\sigma^1 \otimes \sigma^1$, $\sigma^2 \otimes \sigma^2$ and $\sigma^3 \otimes \sigma^3$ and thus of the $S_x^i S_y^i$'s. Written as spin states they correspond to

$$u_{1/2} = \frac{1}{\sqrt{2}} (|\uparrow\uparrow\rangle \pm |\downarrow\downarrow\rangle), \quad u_{3/4} = \frac{1}{\sqrt{2}} (|\uparrow\downarrow\rangle \pm |\downarrow\uparrow\rangle). \quad (2.80)$$

This basis can be used as a starting point for a trial and error investigation to find interesting simulatable measurement processes. All studied measurement processes correspond to this basis, but with different degeneracies in the spectrum. For the process \vec{S}^2 the eigenvalues of u_1, u_2 and u_3 are degenerate. For $S_x^1 S_y^1$ the eigenvalues of u_1, u_3 and u_2, u_4 are pairwise degenerate and for $S_x^+ S_y^+ + S_x^- S_y^-$ only the eigenvalues of u_3 and u_4 are degenerate.

3. Numerical Studies

With the loop cluster algorithm we are able to simulate the real-time evolution of quantum spin systems driven by measurements. With the Euclidean-time part, the system is prepared as a thermal initial density matrix. Here we use the Heisenberg ferromagnet, the Heisenberg antiferromagnet, and the XY-model density matrix. We simulate 2-dimensional $L \times L$ square lattices. The studied observables are the square of the magnetization along the respective quantization axis (1-direction for the XY-model initial density matrix and 3-direction for ferromagnetic and antiferromagnetic initial density matrices), and its Fourier modes. The Fourier modes of the magnetization are given by

$$S(p) = \sum_x \exp(ipx) S_x^j = \sum_x \exp(ip_1 x_1 + ip_2 x_2) S_x^j, \quad (3.1)$$

where j is the quantization axis direction. Note that for $p = (0, 0)$, we retrieve the j -component of the uniform magnetization $M^j = \sum_x S_x^j$, and for $p = (\pi, \pi)$, the j -component of the staggered magnetization $M_s^j = \sum_x (-1)^{x_1+x_2} S_x^j$.

In this section we present the results obtained from these simulations. First, we discuss the real-time evolution driven by discrete measurement processes, where at each time step all spins are measured. Then we consider the real-time evolution driven by continuous measurements in more detail. Finally we study the time scales at which the modes reach the new equilibrium.

3.1. Discrete Measurements

Figure 3.1 shows the real-time evolution of different modes of the magnetization driven by discrete measurements. The lines are included to guide the eye. The initial density matrix is antiferromagnetic. Not all modes are shown to keep the plots clear. We use $\beta J = 40$ on a $L = 16$ square lattice and $aJ = 0.078125$ as Euclidean-time lattice spacing for the preparation of the initial density matrix. Thus we have $4N_{\text{et}} = \beta/a = 512$ Euclidean-time slices. The lattice spacing in real-time, ϵ has no influence on the results in the sporadic measurement case.

Since we start with an antiferromagnetic initial density matrix at low temperature, the (π, π) -mode, the staggered magnetization has a very high initial value ($(M_s^3)_0 = 2965(55)$). The staggered magnetization is the order parameter of the Heisenberg antiferromagnet. We find that the initial order is destroyed quickly. The system is driven into a new equilibrium depending on the simulated process. As one sees in figure 3.1 a), the $(0,0)$ -mode, the uniform magnetization M^3 , is conserved by the measurement process \vec{S}^2 . The conservation of the (π, π) -mode by $S_x^+ S_y^+ + S_x^- S_y^-$ is not visible in the respective figure, since it is outside of the plot range. The measurement process $S_x^1 S_y^1$ does not conserve any of the magnetization modes. Each measurement process drives all except the conserved mode to the same equilibrium value. This new equilibrium value is reached quickly. We find that the conserved mode seems to slow down the neighbouring modes. After a single completed time step, at $4t/\epsilon = 1$, all except the conserved, and the nearest to the conserved mode have equilibrated. To study the real-time dynamics of these measurement processes in more detail, we consider the slowed Lindblad process discussed in section 3.2.

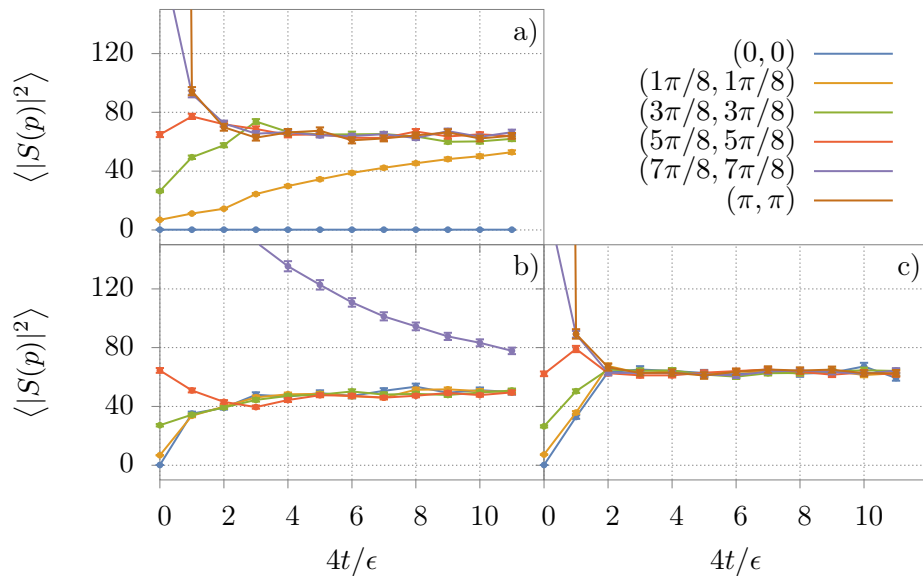


Figure 3.1.: The real-time evolution of different modes of the magnetization driven by sporadic measurements. The initial density matrix is antiferromagnetic with $\beta J = 40$, $aJ = 0.3125$ and $l = 16$. Subfigure a) shows the process \vec{S}^2 , b) $S_x^+ S_y^+ + S_x^- S_y^-$ and c) $S_x^1 S_y^1$. Note that the initial value of the (π, π) -mode is out of the plot range at $M_s^3 = 2965(55)$.

3.2. Continuous Measurements

To study the real-time evolution in more detail, we consider the Lindblad processes. By decreasing the parameter $\epsilon\gamma$ we can make the process arbitrary slow. We prepare the system in three different initial density matrices, ferromagnetic, antiferromagnetic and XY-model. All results shown here correspond to $\beta J = 40$, $aL = 16$ and $\epsilon\gamma = 0.05$. We perform 10^5 Monte Carlo updates.

3.2.1. AFM Initial Density Matrix

The results for the antiferromagnetic initial density matrix are shown in Figure 3.2, for the processes \vec{S}^2 and $S_x^+ S_y^+ + S_x^- S_y^-$, and Figure 3.5 for the process $S_x^1 S_y^1$. Let us first consider the process \vec{S}^2 . We see again that the initial antiferromagnetic order is destroyed, and the system is driven into a new equilibrium. There is a conserved mode, the (0,0)-mode. All other modes are driven to the same new equilibrium value. The modes equilibrate slower, the nearer their momentum is to the momentum of the conserved mode.

The process $S_x^+ S_y^+ + S_x^- S_y^-$ conserves the (π, π) -mode, the order parameter of the AFM. Here again all except the conserved modes are driven to the same new equilibrium value (note that this is not the same value as in the process \vec{S}^2). Here as well, the modes with momentum near the momentum of the conserved mode equilibrate slower.

In the process $S_x^1 S_y^1$ there is no conserved mode and we observe that all modes equilibrate fast, see Figure 3.5.

3.2.2. XY Initial Density Matrix

The real-time evolution of the magnetization modes with the initial density matrix prepared as XY-model are shown in figures 3.3 (\vec{S}^2 and $S_x^+ S_y^+ + S_x^- S_y^-$) and 3.5 ($S_x^1 S_y^1$). First we point out again that for the XY-model initial density matrix, the spins are quantized along the 1-direction in order to have access to the order parameter M^1 . Thus the process $S_x^1 S_y^1$ corresponds to $S_x^2 S_y^2$, and the process $S_x^+ S_y^+ + S_x^- S_y^-$ to $S_x'^+ S_y'^+ + S_x'^- S_y'^-$. Nevertheless we will denote them here as $S_x^1 S_y^1$ and $S_x^+ S_y^+ + S_x^- S_y^-$.

The uniform magnetization is conserved by the process \vec{S}^2 . Here as well as for the antiferromagnetic initial density matrix, all modes except the conserved one are driven to a new equilibrium value. Again the modes equilibrate slower, the nearer their momentum is to the momentum of the conserved mode.

The process $S_x^1 S_y^1$ does not conserve any of the modes and thus, all modes equilibrate fast. We find that the real-time evolution of $S(p)$ driven by the process \vec{S}^2 resembles the real-time evolution of the mode $S((\pi, \pi) - p)$ driven by the process $S_x^+ S_y^+ + S_x^- S_y^-$.

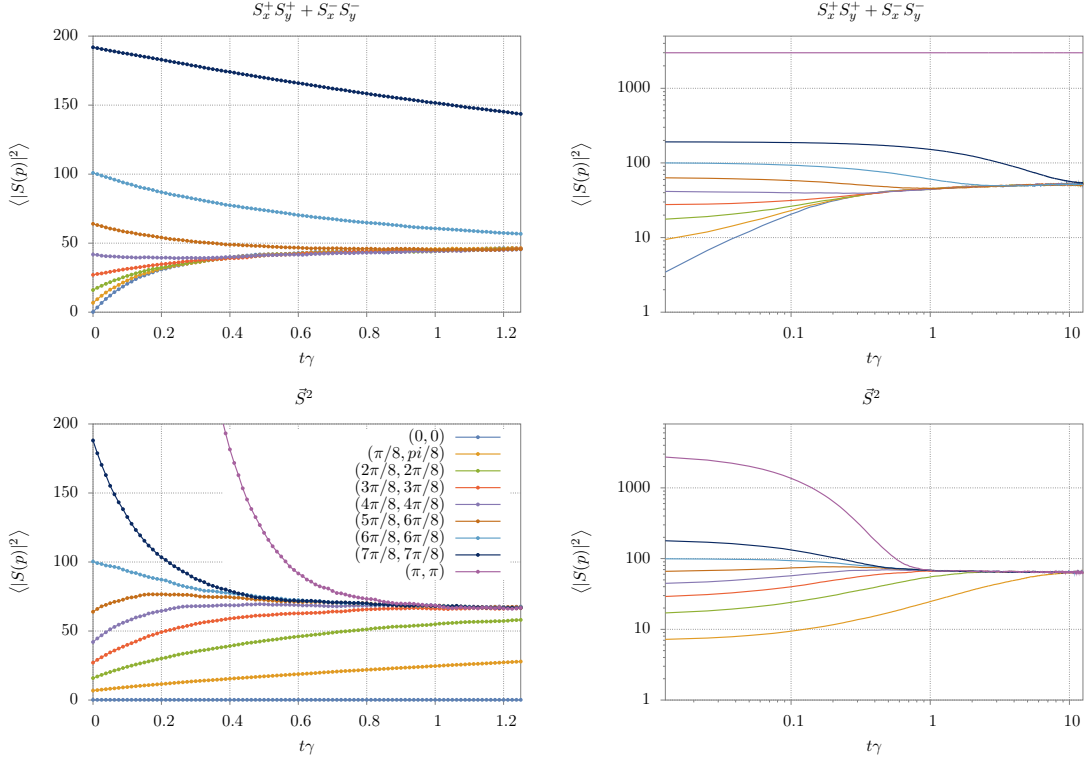


Figure 3.2.: The real-time evolution of different Fourier modes of the magnetization. The initial density matrix is antiferromagnetic. Parameters used: $aJ = 0.3125$ (corresponds to $4N = 512$ Euclidean-time slices), $\beta J = 40$, $l = 16a$ and $\gamma\epsilon = 0.05$. The top figures show the process $S_x^+ S_y^+ + S_x^- S_y^-$ in a linear scale (left) to give a detailed view of the initial phase and in log-log scale (right) to show the long-term evolution. The bottom figures show the process \vec{S}^2 again in linear and in log-log scale

for an antiferromagnetic initial density matrix and vice versa. Both processes conserve the respective order parameter. This suggests that the real-time evolution of the Fourier modes of the magnetization depends strongly on the symmetry properties of the respective process.

3.2.3. FM Initial Density Matrix

Figure 3.4 shows the real-time evolution of the Fourier modes of the magnetization for a ferromagnetic initial density matrix. Again, in the process $S_x^+ S_y^+ + S_x^- S_y^-$, the conserved mode (the (π, π) -mode) slows down the equilibration of the near modes. The process \vec{S}^2 has no influence on the different magnetization modes. They are already in equilibrium,

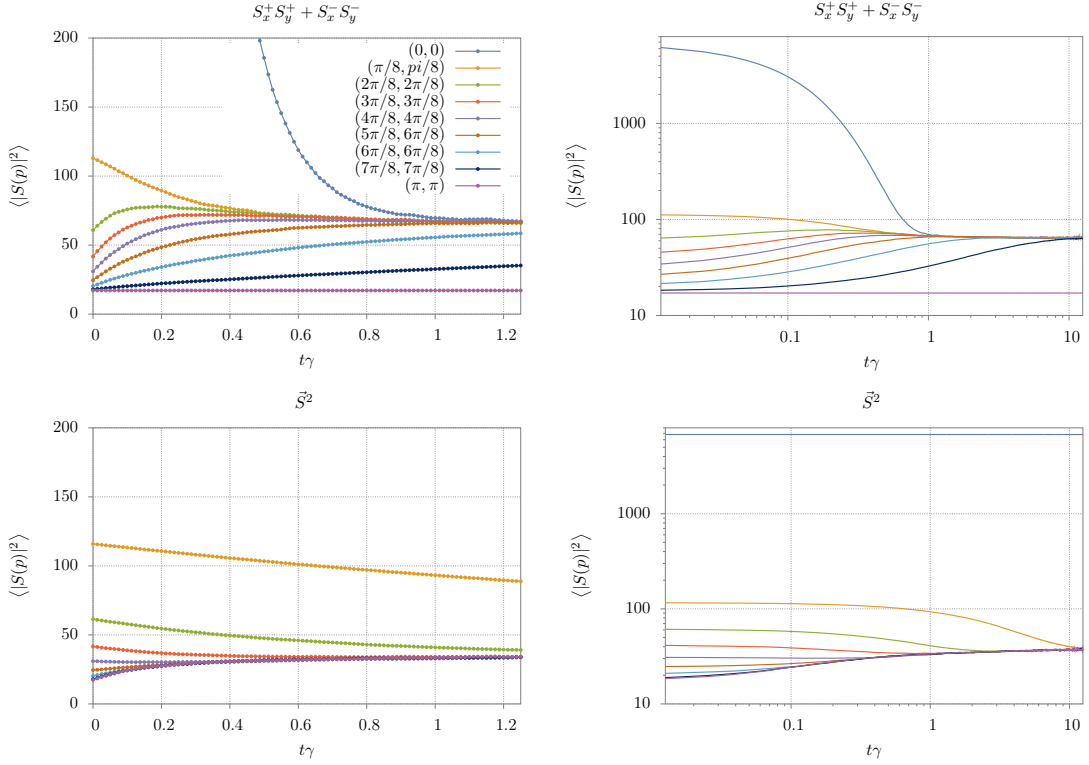


Figure 3.3.: The real-time evolution of various Fourier modes of the magnetization. The initial density matrix is prepared as XY-model. Parameters used: $aJ = 0.3125$ (corresponds to $4N = 512$ Euclidean-time slices), $\beta J = 40$, $l = 16a$ and $\gamma\epsilon = 0.05$. The top figures show the process $S_x^+ S_y^+ + S_x^- S_y^-$ in a linear scale (left), and in a log-log scale (right) to show the long-term evolution. The bottom figures show the process \vec{S}^2 in linear (left) and log-log (right) scale. The error bars are of the order of the point size and the lines are included to guide the eye.

since the cluster rules of this measurement process (for sporadic measurement) correspond to the ferromagnetic cluster rules at low temperature i.e. for $aJ = \beta J/(4N) \rightarrow \infty$. The process $S_x^1 S_y^1$ is shown in figure 3.5. Again it does not conserve any mode, and all modes equilibrate fast.

3.2.4. Equilibration Times

For Fourier modes with low and high momenta, the initial phase of the approach of the equilibrium is well described by an exponential function. The modes with middle momenta show a more complicated behaviour. There is an overshooting, which cannot

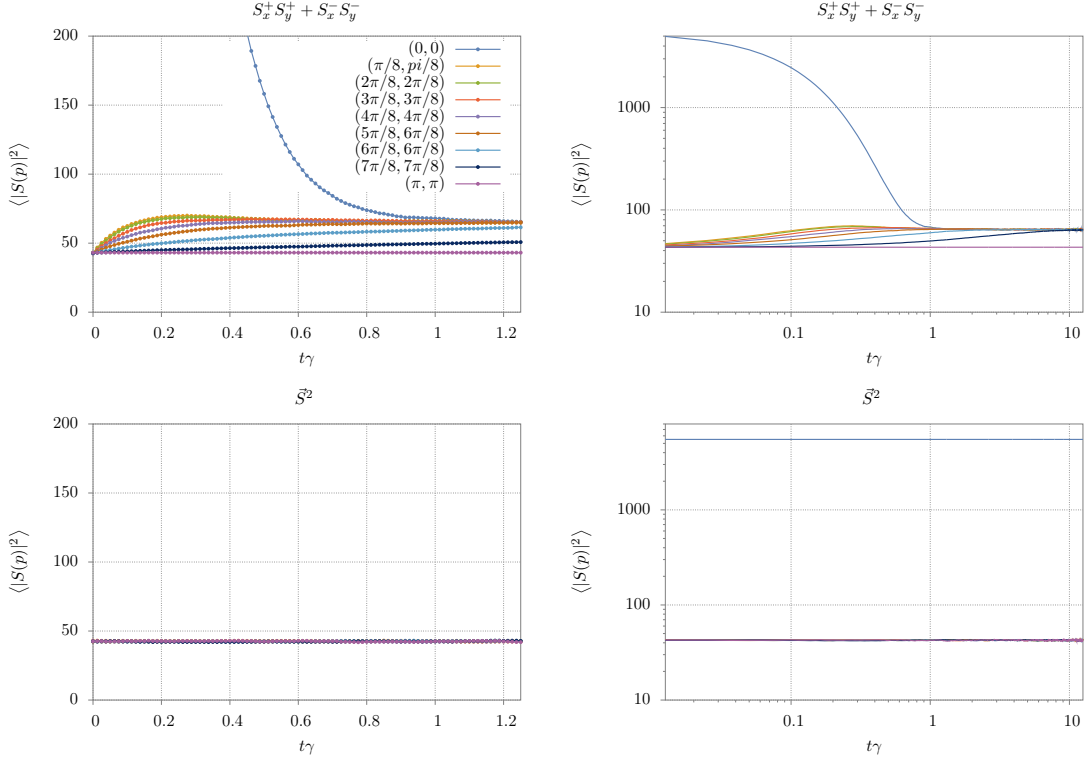


Figure 3.4.: The real-time evolution of various Fourier modes of the magnetization. The initial density matrix is prepared as ferromagnet. Parameters used: $aJ = 0.3125$ (corresponds to $4N = 512$ Euclidean-time slices), $\beta J = 40$, $l = 16a$ and $\gamma\epsilon = 0.05$. The top figures show the process $S_x^+ S_y^+ + S_x^- S_y^-$ in a linear scale (left), and in a log-log scale (right) to show the long-term evolution. The bottom figures show the process \vec{S}^2 in linear (left) and log-log (right) scale. The error bars are of the order of the point size and the lines are included to guide the eye.

be described by a simple exponential function. As can be seen in the log-log plots of the process $S_x^+ S_y^+ + S_x^- S_y^-$, the modes fall on an attractor, which then reaches equilibrium. For a detailed analysis of this behaviour see [11]. Here we focus on the initial phase of the modes with momentum close to $(0,0)$ or (π,π) . In order to have access to more Fourier modes, we simulate a $L = 32$ square lattice, with $\beta J = 40$, $4N_{et} = 512$ and $\gamma\epsilon = 0.05$. We fit the function $a + b \exp(-t/\tau)$ via a , b and τ to the data in the real-time interval $\gamma t \in [0, 2.5]$. The resulting inverse equilibration times for antiferromagnetic as well as XY-model initial density matrices are shown in figures 3.6 and 3.7.

The inverse equilibration time of the modes near the conserved mode ($|p| = \sqrt{2}\pi$ for $S_x^+ S_y^+ + S_x^- S_y^-$ and $|p| = 0$ for \vec{S}^2) show an approximately quadratic dependency on the

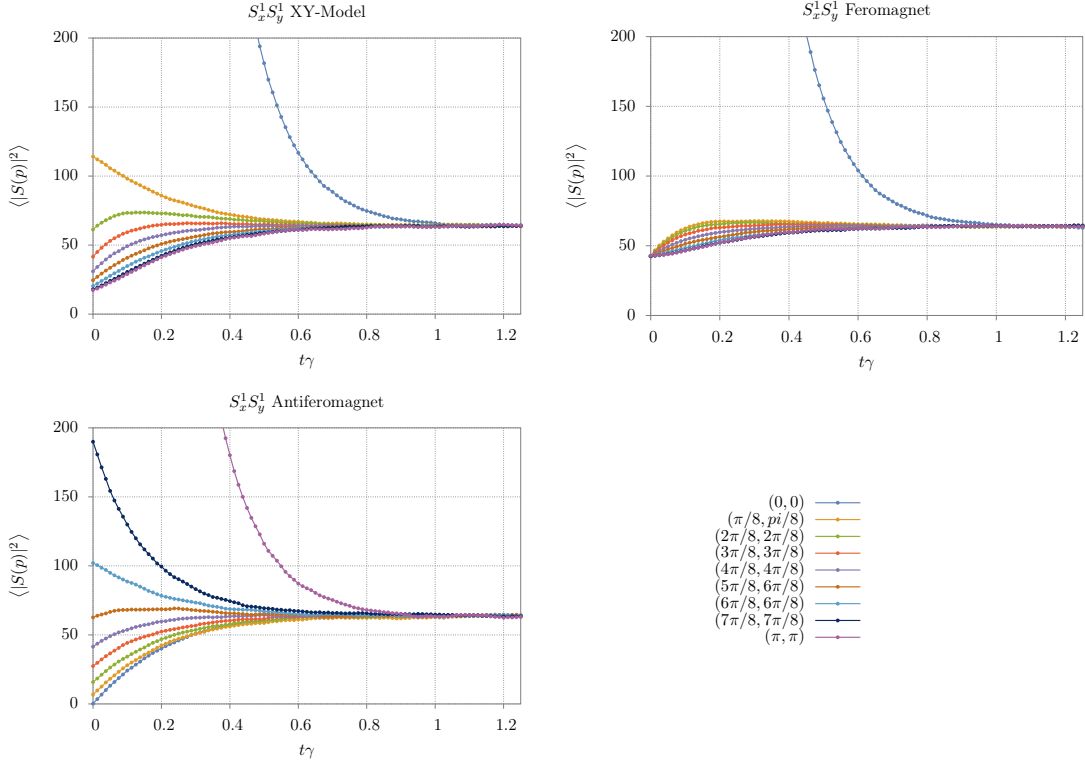


Figure 3.5.: The real-time evolution of various Fourier modes of the magnetization driven by the process $S_x^1 S_y^1$ for initial density matrices prepared as XY-model (top left), ferromagnet (top right) and antiferromagnet (bottom). Parameters used: $aJ = 0.3125$ (corresponds to $4N = 512$ Euclidean-time slices), $\beta J = 40$, $l = 16a$ and $\gamma\epsilon = 0.05$. The error bars are of the order of the point size. The lines are included to guide the eye.

momentum. Fitting the functions $a|p|^b$ and $a(|p| - \sqrt{2}\pi)^b$, yields an exponent consistent with 2, in all except one case: For the XY-initial density matrix, we find $b = 2.03(7)$ for the process \vec{S}^2 and $b = 2.06(11)$ for $S_x^+ S_y^+ + S_x^- S_y^-$. For the antiferromagnetic initial density matrix we find $b = 2.00(8)$ for \vec{S}^2 and $b = 2.13(2)$ for the process $S_x^+ S_y^+ + S_x^- S_y^-$. This square dependency is characteristic for a diffusion process: The diffusion equation in momentum space reads

$$\partial_t S(t, p) = -p^2 D S(t, p), \quad (3.2)$$

with the solution

$$S(t, p) = \exp(-Dp^2 t) S(0, p). \quad (3.3)$$

Note also that for the XY-model initial density matrix $1/(\gamma\tau)$ for the processes $S_x^+ S_y^+ + S_x^- S_y^-$, at momentum $p = \sqrt{2}\pi 15/16$ is inconsistent with the fit. The same is true for

$p = \sqrt{2}\pi/16$ for the process \vec{S}^2 for the antiferromagnetic density matrix. This indicates a systematic error. We point out again that this whole analysis is only approximate. See [11] for more details.

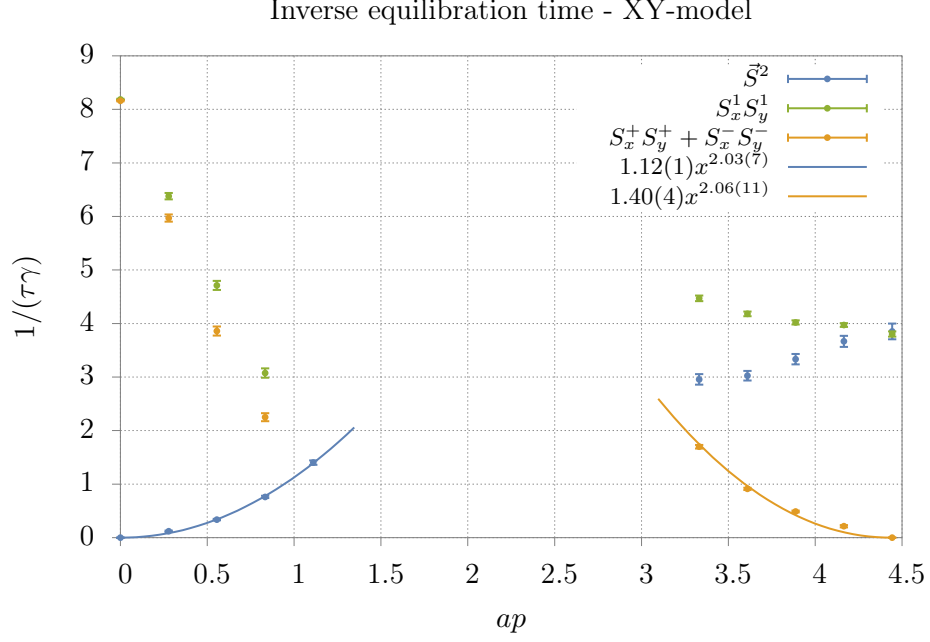


Figure 3.6.: The inverse equilibration times for the XY-model initial density matrix, obtained from fitting $a \exp(-t/\tau) + b$ to the magnetization modes in the real-time interval $\gamma t \in [0, 2.5]$. The lines correspond to the best fit to the functions ax^b and $a(x - \sqrt{2}\pi)^b$.

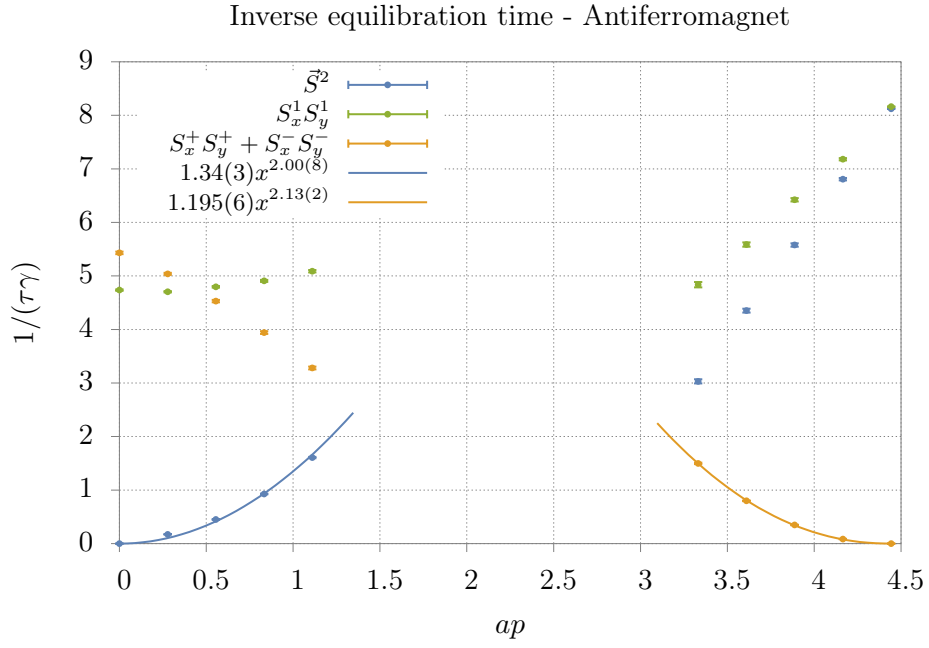


Figure 3.7.: The inverse equilibration times for the antiferromagnetic initial density matrix, obtained from fitting $a \exp(-t/\tau) + b$ to the magnetization modes in the real-time interval $\gamma t \in [0, 2.5]$. The lines correspond to the best fit to the functions ax^b and $a(x - \sqrt{2}\pi)^b$.

4. Conclusion and Outlook

We studied the real-time evolution of 2-d quantum spin systems driven by a dissipative coupling to the environment through a measurement process. This coupling is modeled by a Lindblad process, which is solved by an efficient loop cluster algorithm.

The study of what measurement processes can be simulated did not lead to a definite answer, yet we found some insight on the structure of these processes: We found indication that the simulatable measurement processes, i.e. the measurement processes for which the sign problem disappears, correspond to a discrete subset of all possible measurement processes. This was obtained by a random sampling investigation and a calculation for the 2-dimensional case. This exact calculation shows that on a 2-dimensional Hilbert space, there is only one non-trivial measurement process, for which the sign problem disappears. Although we could not explicitly specify all simulatable measurement processes on a 4-dimensional Hilbert space, we found a good starting point for a trial and error investigation to find new measurement processes. Pursuing this approach, we found two simulatable measurement processes ($S_x^1 S_y^1$ and $S_x^+ S_y^+ + S_x^- S_y^-$) in addition to the process \vec{S}^2 which was introduced in [7].

We further investigated numerically the real-time evolution of the Fourier modes of the magnetization. In order to do this, we simulated three different measurement processes, for three different initial density matrices. The measurement processes have different symmetry properties: The process \vec{S}^2 conserves the (0,0)-mode of the magnetization, the process $S_x^1 S_y^1$ does not conserve any mode and the process $S_x^+ S_y^+ + S_x^- S_y^-$ conserves the (π, π) -mode. We found that the order of the initial density matrices is destroyed, and the system is driven into a new equilibrium. This equilibration depends strongly on the symmetry properties of the measurement processes. We found that the conserved mode slows down their neighbouring modes. An approximate analysis of the equilibration shows that the inverse equilibration time of magnetization modes near the conserved one depends quadratically on the momentum: $1/\tau \propto |p|^2$ for the process \vec{S}^2 which conserves the (0,0)-mode and $1/\tau \propto (|p| - \sqrt{2}\pi)^2$ for the process $S_x^+ S_y^+ + S_x^- S_y^-$, conserving the (π, π) -mode. This dependency is characteristic for a diffusion process. For the process $S_x^1 S_y^1$, which does not conserve any mode, we found no such behaviour.

With the studied method one can simulate the real-time evolution of quantum systems driven by measurements. However, in order to do so one has to completely neglect, the unitary internal dynamics induced by the Hamiltonian. A challenging next step could be to try to bring back the Hamiltonian or parts of it. It would also be interesting, to apply this method to other quantum systems.

Appendix

A. Statistics

In this section we explain how the mean values and their uncertainties are obtained. It is roughly based on [13].

A.1. Averages and Uncertainties

With the loop cluster algorithm we calculate means of observables in the following way: The system starts in a certain configuration C_i . We perform a certain number of Monte Carlo updates to thermalize the system, to make sure the results are not influenced by the initial configuration. After the thermalization the system is in the configuration C_1 and we can calculate the n observables $x_1(1) = O_1(C_1)$, $x_2(1) = O_2(C_1)$, ... , $x_n(1) = O_n(C_1)$. These observables are, for example, the magnetization at different real time steps. The system is updated and then in configuration C_2 . The observables are calculated: $x_1(2) = O_1(C_2)$, $x_2(2) = O_2(C_2)$, ... , $x_n(2) = O_n(C_2)$. This procedure is repeated N_{mc} times. The expectation values for the observables are obtained as

$$\bar{x}_k = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} x_k(i). \quad (.1)$$

Naively one would use the standard deviation of the mean obtained from the sample variance $s_{x_k}^2$,

$$\delta \bar{x}_k = \frac{s_{x_k}}{\sqrt{N_{mc}}} = \sqrt{\frac{\sum_{i=1}^{N_{mc}} (x_k(i) - \bar{x}_k)^2}{N_{mc}(N_{mc} - 1)}} \quad (.2)$$

as the uncertainty. However, this is only valid if the data set $\{x_k(i)\}$ is uncorrelated and normally distributed, which is usually not the case for Monte Carlo simulations. Whether a dataset is autocorrelated can be calculated via the autocorrelation function

$$R(\tau) = \frac{\sum_{i=1}^{N_{mc}-\tau} (x_k(i) - \bar{x}_k) (x_k(i + \tau) - \bar{x}_k)}{\sum_{i=1}^{N_{mc}} (x_k(i) - \bar{x}_k)^2}. \quad (.3)$$

This function behaves approximately exponentially: $R(\tau) \approx \exp(-\tau/\lambda)$ where λ is called the autocorrelation time. The problem of autocorrelated data can be dealt with by binning. If we choose the bin size $m \gg \lambda$, the binned data set,

$$\left\{ \tilde{x}_k(1) = \frac{1}{m} \sum_{i=1}^m x_k(i), \tilde{x}_k(2) = \frac{1}{m} \sum_{i=m}^{2m} x_k(i), \dots, \tilde{x}_k(N/m) = \frac{1}{m} \sum_{i=N-m}^N x_k(i) \right\}, \quad (.4)$$

will no longer be autocorrelated. Due to the central limit theorem, the binned data will be normally distributed if m is large enough. From the binned data one can now calculate the sample variance $s_{\tilde{x}_k}$ to estimate the uncertainty

$$\delta \bar{x}_k = \frac{s_{\tilde{x}_k}}{\sqrt{N_{mc}/m}} = \sqrt{\frac{\sum_{i=1}^{N_{mc}/m} (\tilde{x}_k(i) - \bar{x}_k)^2}{N_{mc}/m (N_{mc}/m - 1)}}. \quad (.5)$$

For a sample data set produced with the loop cluster algorithm we measure a correlation length $\lambda \approx 1$. This is not large but nevertheless has to be taken into account in the error analysis. If we would calculate the uncertainty naively without binning we would underestimate it roughly by a factor 2. For all simulation we use a bin size $m = 100 \gg 1$ (see Figures A.1 and A.2). Thus the binned data is no longer autocorrelated.

A.2. Jackknife

In section 3.2.4 we calculated the equilibration time of the different modes of the magnetization with a non-linear fit. The result of this fit depends highly non-linearly on the measured data. In the last section we have seen how to estimate averages and their uncertainty. This method trivially extends to linear functions of the directly measured quantities but not to non-linear functions, $f(\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_K})$, depending on the true means. If we just use $f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_K)$ as an estimator, the result will be biased, and the uncertainty is usually underestimated. If the data is correlated in k , meaning the sets $\{x_1(i), x_2(i), \dots, x_K(i)\}$ are autocorrelated for a specific i , one has to perform a more sophisticated error analysis. The jackknife method is a tool to estimate uncertainty and bias of such non linear functions.

We will assume the data sets $\{x_k(1), x_k(2), \dots, x_k(N)\}$ to be uncorrelated. This can be obtained by binning as described earlier. The i -th jackknife estimate $x_k^J(i)$ is defined as

$$x_k^J(i) = \frac{1}{N-1} \sum_{j \neq i} x_k(j) = \bar{x}_k + \frac{1}{N-1} (\bar{x} - x_k(i)), \quad (.6)$$

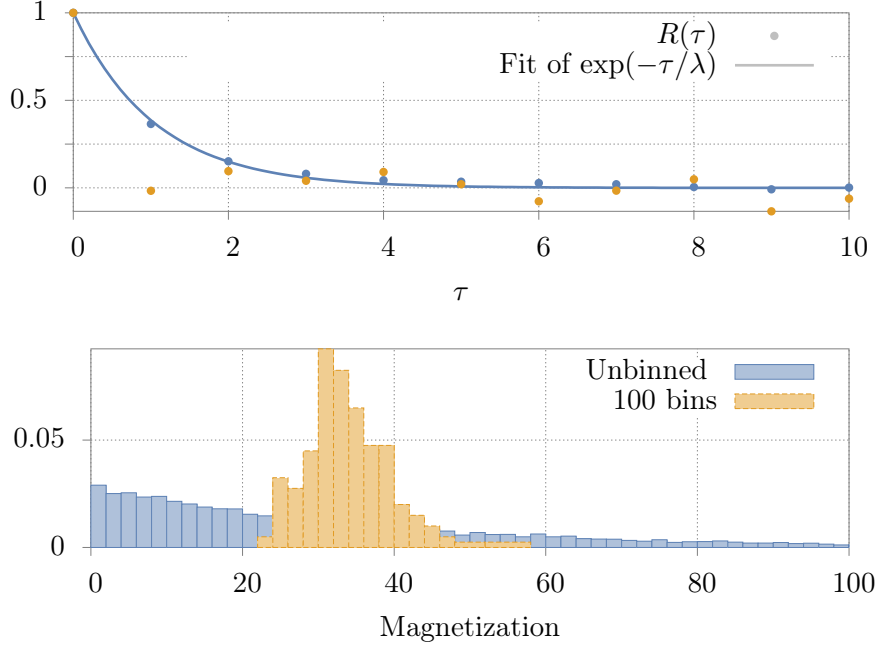


Figure A.1.: The autocorrelation function and histograms of a sample data set for the $(\pi/2, \pi/2)$ -mode of the magnetization in the XY-model at $\beta J = 40$. The blue data set shows unbinned and the orange data set binned data with binsize $m = 100$. For the unbinned data (blue), fitting $\exp(-\tau/\lambda)$ yields $\lambda = 1.05(3)$. For the binned data set (orange) the fit does not lead to a reliable correlation length.

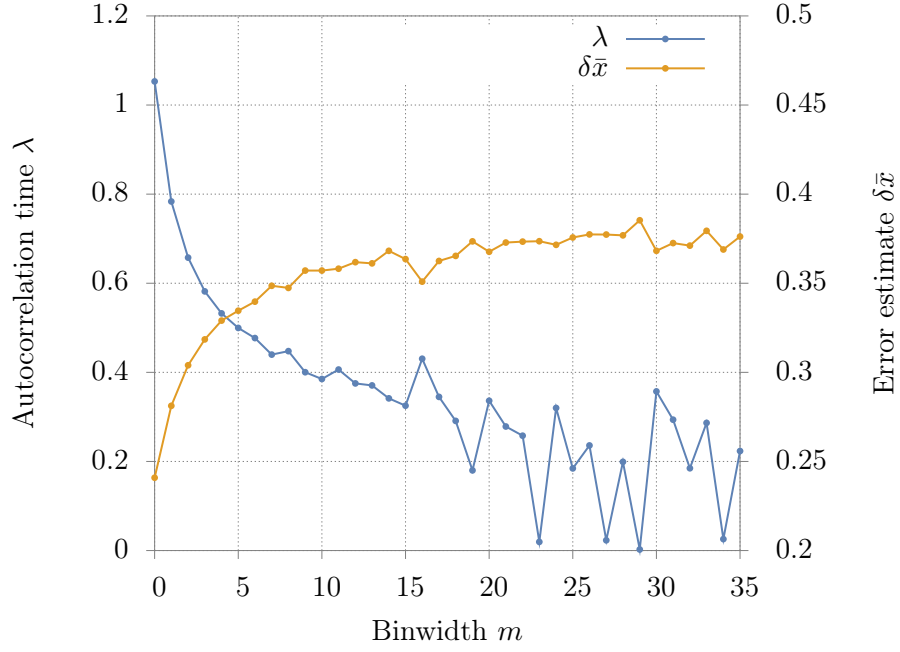


Figure A.2.: The error estimate $\delta\bar{x}$ and the autocorrelation time λ , for different bin widths m , for the same sample data set as in Figure A.1.

the average over all except the i -th values. The i -th jackknife estimate for the function f is defined analogously as

$$f_i^J = f(x_1^J(i), x_2^J(i), \dots, x_K^J(i)). \quad (.7)$$

The average over these N jackknife estimates is denoted $\overline{f^J} = \sum_{i=1}^N f_i^J$ and the variance as $s_{f^J} = \overline{(f^J)^2} - (\overline{f^J})^2$. With this we can find an estimator for $f(\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_K})$, which is unbiased, as

$$Nf(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_K) - (N-1)\overline{f^J}, \quad (.8)$$

and the uncertainty as

$$\delta\bar{f} = \sqrt{N-1} s_{f^J}. \quad (.9)$$

B. Units

We use natural units, where $c = \hbar = \epsilon_0 = 1$. In these units the spin and with it the magnetization is a dimensionless quantity. Time and distance and with it the lattice

spacings, a and ϵ have units of inverse energy. To convert the magnetization back into physical units, one has to multiply it by \hbar .

In lattice simulations we can only deal with dimensionless quantities, e.g. βJ , $\gamma\epsilon$ and so on. The time scale of the Lindblad process is set by the parameter γ . In the simulation it appears only in the dimensionless combinations $\epsilon\gamma$ and $t\gamma$.

C. Notes on the Implementation

The single loop cluster algorithm was implemented as a C-program. The source code can be found at <https://bitbucket.org/mhornung/loop-cluster-algorithm/src>. To be more flexible, the program takes the cluster rules as arguments. A Mathematica notebook is provided to generate files containing cluster rules and other parameters automatically.

The program was checked by comparing the results to the numbers published in [7] and [8], as well as to numerically exact results, calculated on a 2×2 lattice, where one can explicitly sum over all configurations.

Bibliography

- [1] J. Berges, Introduction to Nonequilibrium Quantum Field Theory, in *AIP Conf. Proc.*, volume 739, pages 3–62, AIP, 2004.
- [2] S. R. White, Phys. Rev. Lett **69**, 2863 (1992).
- [3] U. Schollwöck, Rev. Mod. Phys. **77** (2005).
- [4] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).
- [5] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch, and I. Bloch, Nature **415**, 39 (2002).
- [6] J. Preskill, *Lecture Notes for Physics 229: Quantum Information and Computation*, 1998.
- [7] D. Banerjee, F. J. Jiang, M. Kon, and U.-J. Wiese, Phys. Rev. B **90** (2014).
- [8] U.-J. Wiese and H. P. Ying, Z. Phys. B **93**, 7 (1994).
- [9] H. G. Evertz, Adv. Phys. **52**, 1 (2003).
- [10] R. Brower, S. Chandrasekharan, and U.-J. Wiese, Phys. A **261**, 520 (1998).
- [11] F. Hebenstreit et al., arXiv:1502.02980 [cond-mat.str-el] (2015).
- [12] S. J. Akhtarshenas and A. Kheirollahi, Int. J. Theor. Phys. **49**, 402 (2009).
- [13] P. Young, arXiv:1210.3781 [physics.data-an] (2012).

Acknowledgement

First of all I would like to thank Prof. Dr. Uwe-Jens Wiese, the supervisor of this thesis for taking so much time to explain the many things considered here, for including us into discussions with his work group and taking our opinion seriously. Furthermore I want to thank Dr. Debasish Banerjee and Dr. Florian Hebenstreit, for patiently answering our many questions. I also thank Franziska Schranz for the great collaboration. In Addition I want to thank Luca Studer for providing a mathematician's point of view on some of the difficulties I encountered.

Many thanks go to my fellow students who were great companions through the years at university. Thank you for the many inspiring discussions about life, the universe and everything. And last but really not least I would like to thank my friends and family. Thank you for the moral support, and for showing so much interest in the sometimes complicated and mostly far-from-everyday-life things which excited me during the work on this thesis.