Daniel Bauernfeind, BSc

Stochastic Sampling of Reduced Density Matrices with Quantum Monte Carlo

MASTER THESIS

For obtaining the academic degree
Diplom-Ingenieur

Master Programme of Technical Physics

Graz University of Technology

Supervisor:
Ao. Univ.-Prof. Dipl.-Phys. Dr .rer .nat. Hans Gerd Evertz
Institute of Theoretical and Computational Physics

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit/Diplomarbeit/Dissertation identisch.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis/diploma thesis/doctoral dissertation.

______________________________ ______________________________

Datum / Date Unterschrift / Signature

Contents

Abstract

This thesis introduces a new way to perform quantum monte carlo which has access to reduced density matrices or even full density matrices if the total system is very small. The model with which the performance and limits of this method will be tested is the isotropic Heisenberg antiferromagnet. For this model the Loop Algorithm [1] is a convenient way to perform the updates in the path integral. Therefore it will be the core algorithm which we will adapt for our purposes. Nevertheless the idea introduced in this work can be used with any other monte carlo method which samples from the path integral.

Once reduced density matrices, describing some sub-region of the total lattice, are obtained one can extract correlation functions, the von Neumann entropy of the reduced density matrix or any expectation value of some operator acting on those sites. The von Neumann entropy turns out to be a measure of entanglement between the region described by the reduced density matrix and the rest of the lattice if the total state is pure. We will see, that correlation functions and entanglement measures are closely related to each other, which make the latter a useful tool in detecting (quantum) phase transitions.

Since in monte carlo simulations pure states only appear in the ground state we will also look at ways to quantify thermal entanglement. Unfortunately in these cases entanglement depends on the total state (as opposed to being dependent on the reduced density matrix) which forces us to think about the reduced state as the new state of the system in thermal equilibrium with the environment. With this we can then calculate the Entanglement between 2 regions both described by the reduced density matrix. Furthermore thermal

entanglement is in general very difficult to calculate, and we have to use measures which do not quantify it perfectly like the logarithmic negativity. We will present the logarithmic negativity for 1 dimensional and a 2 dimensional systems.

The von Neumann entropy is not only interesting as a measure of entanglement, but also because it exhibits an interesting scaling if the size of the subsystem (described by the reduced density matrix) is increased. If the system can be described by conformal field theory the scaling of the von Neumann entropy gives access to the central charge.

Since there is a broad interest in calculating von Neumann entropies the so called Valence Bond entropies [2] [3] were introduced. They should approximate the (scaling) of the von Neumann entropy. The algorithm presented in this thesis has also access to these quantities. Therefore we thought of some ways to generalize the idea of the Valence Bond entropies and introduce the so called cluster entropies. As a result we will compare the cluster- and Valence Bond entropies with each other and put them in relation to the von Neumann entropy.

Zusammenfassung

In dieser Masterarbeit wird eine neue Quanten Monte Carlo Methode vorgestellt, die sowohl reduzierte Dichtematrizen als auch, wenn das Gesamtsystem klein ist, vollständige Dichtematrizen berechnen kann. Die Leistungsfähigkeit und die Grenzen dieser Methode werden am isotropen, antiferromagnetischen Heisenberg Modell getestet. Für dieses Modell ist der Loop Algorithmus [1] eine praktische Methode um Updates im Pfadintegral durchzuführen. Deshalb wird dieser als Ausgangspunkt gewählt um die nötigen Modifikationen durchzuführen. Dies bedeutet nicht, dass die selbe Idee nicht auch auf andere Monte Carlo Methoden angewandt werden kann, solange diese, wenn auch nur im Prinzip, vom Pfadintegral ausgehen.

Mit reduzierten Dichtematrizen, die ein gewisses Teilsystem des gesamten Gitters beschreiben, kann man Korrelationsfunktionen, die von Neumann Entropie beziehungsweise jeden Erwartungswert eines Operators berechnen, der nur auf das Teilsystem wirkt. Es stellt sich heraus, dass, wenn der Gesamtzustand rein ist, die von Neumann Entropie eine Möglichkeit ist die Verschränkung zwischen dem Teilsystem und dem Rest des Gitters zu quantifizieren. Außerdem wird sich zeigen, dass Korrelationsfunktionen und alle Maße von Verschränkung eng miteinander verbunden sind, was letztere zu nützlichen Werkzeugen in der Detektion und der Untersuchung von (Quanten-) Phasenübergängen macht.

Da reine Zustände in Monte Carlo nur auftauchen, wenn sich das System im Grundzustand befindet, wird diese Arbeit auch Möglichkeiten zur Berechnung von thermischer Verschränkung vorstellen. Leider hängt sie, in allen Fällen vom Gesamtzustand ab (wohingegen im Grundzustand nur die reduzierte Dichtematrix benötigt wird). Deshalb muss die reduzierte Dichtem-

atrix als neuer Ausgangszustand eines kleinen Systems betrachtet werden, welches in thermischem Gleichgewicht mit seiner Umgebung ist. Hinzu kommt, dass es im Allgemeinen sehr schwierig ist thermische Verschränkung zu berechnen, weshalb diese Arbeit sich auf Maße beschränken muss, die die Verschränkung nicht perfekt wiedergeben. Ein Beispiel ist die sogenannte *logrithmic negativity.* Diese Arbeit berechnet sie für ein- und zwei-dimensionale Gitter.

Die von Neumann Entropie ist nicht nur als Verschränkungsmaß nützlich, sie zeigt auch ein interessantes Skalierungsverhalten unter Vergrößerung des Teilsystems. Wenn das physikalische System durch eine Konforme Feldtheorie beschrieben werden kann, ermöglicht es das Skalierungsverhalten die sogenannte Zentrale Ladung zu berechnen.

Aus diesen Gründen besteht ein großes Interesse darin, die von Neumann Entropie bestimmen zu können. Deshalb wurden die sogenannten *Valence Bond* Entropien [2] [3] definiert, bei denen es sich um Näherungen zur von Neumann Entropie handelt. Es stellt sich heraus, dass die Methode, die in dieser Arbeit vorgestellt wird auch diese berechnen kann. Dies führte uns zur Definition der Cluster Entropien, die man als Verallgemeinerung der Valence Bond Entropien betrachten kann. Diese Arbeit vergleicht die Cluster Entropien mit den *Valence Bond* Entropien und setzt sie in Bezug zur von Neumann Entropie.

Chapter 1

Introduction and Overview

The purpose of this thesis is to introduce a new way of performing updates in the path integral, which give access to reduced density matrix. Furthermore we will discuss and present quantities calculated from these reduced states. The relation of reduced density matrices with entanglement on the one hand and correlation functions on the other, brings together the fields of quantum computation and condensed matter. Exactly there, somewhere between these two fields this thesis is located. We will use quantum monte carlo stemming from condensed matter theory and determine quantum mechanical states with which entanglement measures are calculated. Entanglement is the strange quantum phenomena which enables quantum computers to perform tasks not possible in classical physics.

In chapter 2, we will introduce the physical model which will be used in this thesis: the isotropic Heisenberg antiferromagnet. Chapter 3 defines reduced density matrices and explains why and how they appear in the path integral.

Reduced states can be used to quantify entanglement, if the total state is pure. Therefore chapter 4 defines and shows ways to quantify entanglement. Since quantum monte carlo very often has not only access to the ground state, but also to thermal states, we will look at ways to calculate the thermal entanglement as well.

The next chapter 5 explains the Loop Algorithm, which I used as a way to perform the actual updates in the worldline configuration. Chapter 6 shows what changes one needs to make to the traditional Loop Algorithm, in order to sample reduced density matrices. This new algorithm will be called density matrix loop algorithm (DMLA)

In Chapter 7 we will introduce the Valence Bond entropies and propose some similar measures, the so called cluster entropies. The reason for dealing with Valence Bond entropies is not only that they appear in literature, but also that they are accessible in the algorithm shown in this thesis. This will be explained in chapter 8.

Then we will discuss the results of this thesis. To get a feeling about the performance of the DMLA we will try to emulate the monte carlo in chapter 9.

After this, the the first results of the DMLA will be presented and discussed in chapter 10. There also ways to drastically improve the performance by means of improved estimators will be discussed and results for the cluster and Valence Bond entropies will be presented.

Finally, the last two chapters 11 and 12 show results for thermal entanglement in 1 and 2 dimensional systems respectively. Some plots showing the so called logarithmic negativity (our measure of choice) can be found there.

Chapter 2

The Heisenberg Model

The Heisenberg Model is a lattice model describing (nearest neighbor) spin-interactions. It is governed by the Hamiltonian:

$$H = \sum_{\langle i,j \rangle} J_z \sigma_i^z \sigma_j^z + J_{xy} \big(\sigma_i^x \sigma_j^x + \sigma_i^y \sigma_j^y \big) + h \sum_i \sigma_i^z \tag{2.1}$$

with $\langle i, j \rangle$ implying that the sum is only over nearest neighbor pairs. The operators acting on site i, σ_i^n $(n \in \{x, y, z\})$ are the Pauli Matrices, h is an external magnetic field, chosen to point in z-direction. Here I will write spins states mostly in the σ^z-basis with states $|1\rangle$ for spin up and $|0\rangle$ for spin down. The Heisenberg model describes the physics of the half-filled Hubbard Model in the strong coupling limit [4]. In case of d=1 (one dimension) the Heisenberg Model as well as the Hubbard Model are exactly solvable by Bethe Ansatz [5]. If one wants to go beyond one dimensional systems one has to either use numerical techniques such as Monte Carlo or use approximations. Different values of the parameters J_z and J_{xy} result in different models and physics. $J_{xy} = 0$ describes the classical Ising Model. It being a classical model comes from the fact that there are only σ^z-operators which are diagonal in the local S_z-basis. Therefore also the Hamiltonian is diagonal in this basis i.e.: it is the eigenbasis. This means that every state can be written as $\rho = e^{-\beta H} = \sum_i e^{-\beta E_i} |i\rangle \langle i|$ with $|i\rangle$ being a local basis-state. Note that the word *local* is very important in the last sentence since the density matrix can

always be expanded like this if one takes the eigenstates of the Hamiltonian. This definition of classicality is closely related to entanglement which cannot be found in classical systems as will be explained in section 4.
If $J_z = 0$, one gets the XY-Model with a generally entangled ground state. In one dimension at a critical field $h = 4J$ the XY-Model undergoes a second order phase transition to a factorized ground state (no entanglement). The order parameter is the magnetization in x-direction [6]. Note that in [6] the Hamiltonian is defined differently than it is done here resulting in a different factor in the critical field.

The case $J_z = |J_{xy}| = J$ is the isotropic Heisenberg Model with anti-ferromagnetic couplings if $J > 0$ and ferromagnetic ones for $J < 0$[1].
The sign of the J_{xy} is of no relevance to the model having ferromagnetic or anti-ferromagnetic ordering when the lattice is bipartite. To understand this, assume an even number of sites in every dimension so that the lattice can be separated into 2 sub lattices A and B. Hence every site from A has only nearest neighbors from B and vice versa. A unitary transformation written in S^z-basis:

$$U = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

on all sites of one of these sub lattices results effectively in changing the sign of J_{xy}. Since unitary transformations only change the basis, the sign of J_{xy} has no physical meaning.
In the present work always anti-ferromagnetic couplings i.e.: $J > 0$ were used. By introducing the so called ladder operators $\sigma_i^+ = \sigma_x + i\sigma_y$ and $\sigma_i^- = \sigma_x - i\sigma_y$ the Hamiltonian 2.1 can be rewritten in a more convenient form:

$$H = \sum_{\langle i,j\rangle} J\sigma_i^z\sigma_j^z + \frac{J}{2}\left(\sigma_i^+\sigma_j^- + \sigma_i^-\sigma_j^+\right) + h\sum_i \sigma_i^z, \tag{2.2}$$

It is not hard to see that the Hamiltonian 2.2 conserves the total spin in z-direction i.e.: it commutes with the operator $S^z_{tot} = \sum_i S_i^z$. This means

[1] Consider $JS_i^zS_j^z\,|11\rangle = J\,|11\rangle$ and $JS_i^zS_j^z\,|10\rangle = -J\,|10\rangle$. Clearly if $J > 0$ the anti-ferromagnetic state has lower energy than the ferromagnetic one.

that the action of 2.2 on a basis state results in a new state with the same number of up-spins in it. Of course this is also true for any function of the Hamiltonian - a property which will be needed later.

Chapter 3

Reduced Density Matrices

Most of the time one is not able to work with quantum systems not affected by the environment of the system. Even, if a state is initially separated from the surrounding world [1], interactions will destroy this independence and one has to work with a composite system, and therefore, with mixed states. To extract information of the system one needs to somehow remove the environment, while still keeping in mind that it affects the system.

If some measurement described by the operator A can only be performed on the system, the expectation value of A in state ρ is:

$$\langle A \rangle = tr\big(\rho A\big) = tr_S \underbrace{\big(tr_E \rho\big)}_{\rho_S} A = tr_S\big(\rho_S A\big). \tag{3.1}$$

tr means the total trace over all degrees of freedom of the system and its environment, while tr_E (tr_S) means that the trace is only over the degrees of freedom of the environment (system) respectively. Equation 3.1 suggests that ρ_S is the density matrix of the system, if the system and environment together are in state ρ. Since all degrees of freedom of the environment are removed ρ_S is called reduced density matrix of the system (RDM).

Note that equivalently one could trace out all the degrees of freedom of the

[1] Its density matrix has the form $\rho = \rho_E \otimes \rho_S$ with ρ_E being the state of the environment and ρ_S the state of the system one measures in. Note that this form means that there is no entanglement between system and environment

system to obtain a description of the environment. But usually one thinks about the environment as being very large with ideally continuous spectrum, while the system is rather small.

It is obvious that the concept of a reduced density matrix also applies to cases, where there is no fundamental difference between system and environment. Consider a composite system of 2 qubits A and B with basis states $|0\rangle$ and $|1\rangle$. Let the 2 qubits be in the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}\big(|11\rangle + |00\rangle \big) = \frac{1}{\sqrt{2}}\big(|1_A\rangle \otimes |1_B\rangle + |0_A\rangle \otimes |0_B\rangle \big).$$

Two observer we call them Alice and Bob share those two qubits. Alice gets the one labeled A while Bob gets the other named B. In passing we note that this is a pure, but maximally entangled state. Under the assumption that the state of the 2 qubits does not change during the transport i.e.: they are completely shielded from interactions with the environment - How can we describe the state either of the two has in their laboratory? It is the reduced density matrix obtained by tracing out the qubit in the other persons laboratory. Since the total state is:

$$\rho = |\psi\rangle \langle\psi| = \frac{1}{2}\big(|11\rangle \langle 11| + |00\rangle \langle 00| + |11\rangle \langle 00| + |00\rangle \langle 11| \big),$$

tracing out, say, particle 2 with index B yields the reduced density matrix ρ_A:

$$\rho_A = tr_B(\rho) = \sum_{i=0}^{1} \langle i_B| \rho |i_B\rangle = \frac{1}{2}\big(|1\rangle \langle 1| + |0\rangle \langle 0| \big)$$

which is a totally mixed state. If Alice measures her qubit, she will see, with equal probability, the values 1 and 0, so she is maximally unsure about her outcome, although she knows the total state $|\psi\rangle$ with certainty.

The interest in reduced density matrices lies on the one hand in being able to describe a system interacting with its environment. On the other hand one is able to split a system into parts and figure out what one would measure in this subsystem.

The latter view of RDMs will be used in this work. Starting from an isotropic,

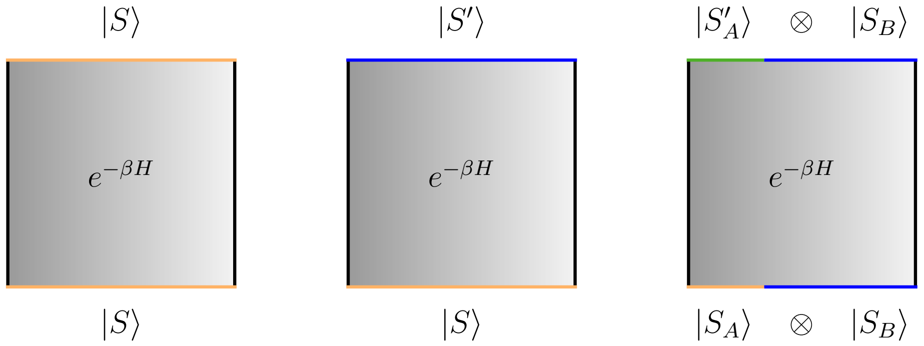

Figure 3.1: Graphical representation of the partition function (left), density matrix (middle) and reduced density matrix (right). In all 3 cases the shaded area corresponds to the operator $e^{-\beta H}$ written as Path-Integral. The axis of imaginary time is the vertical axis.
(Left) Partition Function $Z = tre^{-\beta H} = \sum_{\{|S\rangle\}} \langle S| e^{-\beta H} |S\rangle$. Reading this equation from the right to the left, one starts with some state $|S\rangle$ which gets changed according to $e^{-\beta H}$, but at the end the state has to be $|S\rangle$ again. With other words one can impose periodic boundary conditions in imaginary time.
(Middle) One matrix element of the unnormalized density matrix $\langle S'| \rho |S\rangle = \langle S'| e^{-\beta H} |S\rangle$. Again reading from the right to the left we start with $|S\rangle$, let the operator change this state and at the end we project on $|S'\rangle$.
(Right) The reduced density matrix ρ_A is a mixture of the above mentioned quantities. A matrix element of the RDM describing region A $\langle S'_A| \rho_A |S_A\rangle = \sum_{\{|S_B\rangle\}} \langle S'_A| \langle S_B| e^{-\beta H} |S_A\rangle |S_B\rangle$. Has periodic boundaries in imaginary time at all sites belonging to B, while it allows for different spin states $\langle S'_A|$ and $|S_A\rangle$ at all sites from A.

anti-ferromagnetic Heisenberg Model, the reduced density matrix for some sites in the lattice will be calculated. We will see that this can be done for the ground state as well as for thermal states. Actually the latter will be easier to calculate since the computational effort of the Monte Carlo method I use is proportional to the inverse temperature β. But let's not jump ahead.

Consider the density matrix of some system with respect to some Hamiltonian H in thermal equilibrium:

$$\rho := Z^{-1} e^{-\beta H}$$

with $\beta = \frac{1}{T}$ the inverse temperature and the partition function Z, ensuring the normalization, defined as:

$$Z \coloneqq tr\big(e^{-\beta H}\big). \tag{3.2}$$

Now take 2 sets of sites A and B, such that $A \cap B = \emptyset$ (no point from A is in B) and together they form the lattice. The reduced density matrix describing region A is, as before, obtained by tracing out all sites belonging to B from the total density matrix:

$$\rho_A = tr_B(\rho). \tag{3.3}$$

Using path integrals (for more explanation regarding path integrals see chapter 5) for the term $e^{-\beta H}$ one can write the density matrix, the reduced density matrix as well as the partition function in this language. Figure 3.1 explains the similarities and differences between those 3 quantities. For example a matrix element of the density matrix with respect to some basis elements $|S_1, \cdots, S_N\rangle = |S\rangle$ and $|S'_1, \cdots, S'_N\rangle = |S'\rangle$ is [7][2]:

$$\langle S|\, \rho\, |S'\rangle = Z^{-1} \int \big[d\mathcal{S}(j,\tau)\big] \prod_{i=1}^{N} \delta\big(\mathcal{S}(j,0) - S_i\big) \prod_{i'=1}^{N} \delta\big(\mathcal{S}(j,\beta) - S'_{i'}\big) e^{-S_E}. \tag{3.4}$$

The path integral is over all possible configurations of spins $\mathcal{S}(j,\tau)$ with $j \in \{1, \cdots, N\}$ a lattice point and $0 \leq \tau \leq \beta$ in a (d+1)-dimensional lattice. The extra dimension is the inverse temperature also called imaginary time, since $e^{-\beta H} = e^{itH}$ is equivalent to a time evolution in imaginary time $t = i\beta$. The delta-distributions ensure that at inverse temperature 0 the spin state is $|S\rangle$ and at β it is $|S'\rangle$.

Demanding the spins to be the same at $\tau = 0$ and $\tau = \beta$ i.e.: $|S\rangle = |S'\rangle$, and summing over all configurations with this restriction yields the partition function.

So we found that up to the normalization the distribution function and the density matrix can be expressed in very similar ways. Start with some spins at time $\tau = 0$, change them according to the current configuration in the

[2] Note that in 7 a different notation was used. I chose to write it this way, because here only spin-systems are of interest and introducing their notation seemed to confuse more than it helps.

path integral resulting in a final spin state at $\tau = \beta$. Finally sum over all these configurations. The difference is only in the behavior at the borders $\tau = 0$ and $\tau = \beta$. While the density matrix allows for different spin states, the distribution function demands periodic boundary conditions expressed in the trace in equation 3.2.

From this point it should not be to hard to figure out how to get a reduced density matrix. Because equation 3.3 has a partial trace over all sites belonging to B, one has to impose periodic conditions at these sites, while leaving all sites belonging to A open. The open sites in A can also be viewed as cutting open the path integral of the distribution function. Therefore, these open regions of the space-time lattice are referred to as cuts in this work.

To summarize the result of this section we see that the density matrix, the reduced density matrix, as well as the distribution function are described by very similar path integrals. Quantum Monte Carlo (QMC) procedures sample from the partition function (that is their purpose). Most of them actually have the path integrals built into them even if it is not obvious at first sight. Hence it should be possible to alter existing ways to do QMC, to be able to sample reduced density matrices. In this work I use the Loop Algorithm [1] and verify that one is able to sample RDMs. Other Monte Carlo procedures which should[3] work are the stochastic series expansion (SSE) [8] or the projector QMC in valence bond basis [9] [10]. A different approach to sample density matrices has been made by Blunt et al. [11]. In their work they did not use a path integral formulation of the partition function with some loop update like the so far introduced QMC-Algorithms. Instead they worked in a discrete basis and determined coefficients of the density matrix as an average over so called walker occupation of these states. With this method it is possible to calculate either the ground state or thermal density matrix.

[3] understand the word should as I see no reason why it shouldn't work, but I did not test it

Chapter 4

Entanglement and Reduced Density Matrices

The previous chapter showed that we should be able to calculate reduced density matrices (or even full density matrices if the system is very small) using QMC. This chapter explains why it is interesting to be able to do so. Firstly it can be used to detect first or second order quantum phase transitions [12] [13]. Discontinuities in the derivatives of the Energy translate in discontinuities into some of the matrix elements of the RDM.
The discontinuities in the matrix elements then in turn yield to a similar behavior in the entanglement measures (see chapter 4.2 for the definition of entanglement). Furthermore, with the RDM the von Neumann entropy can be calculated:

$$S^{vN}(\rho_A) \equiv S(\rho_A) := -tr\Bigg(\rho_A \cdot \ln \rho_A\Bigg) \tag{4.1}$$

or its generalized version, the Renyi entropies

$$S_n(\rho_A) := \frac{1}{n-1} \ln \Bigg[tr\rho_A^n\Bigg]. \tag{4.2}$$

Note that the Renyi entropy is equal to the von Neumann entropy in the limit $n \to 1$. Both measures are entanglement measures when the total density matrix is in a pure state. Renyi entropies are accessible for very big subsystems through Monte Carlo simulations using the replica trick [14] for finite temperatures or by introducing the so called swap operator for ground state entropies[15]. In both cases, n versions of the same system are present, which lets one calculate the n-th Renyi entropy.

From the viewpoint of information theory, entanglement is a resource (see chapter 4.1), hence knowing the entanglement of a state is very much desired. In condensed matter physics the interest is not so much in the actual value of entanglement, but more in its scaling behavior if the subsystem size becomes large. The scaling of the entropy tells us about the underlying conformal field theory (CFT) describing the model at the critical point [7] [16]. For example the von Neumann entropy at a quantum phase transition in d=1 scales as:

$$S = \frac{c}{3} \ln \left[\frac{L}{a\pi} \sin \left(\frac{\pi l}{L} \right) \right] + c_1'. \tag{4.3}$$

c is the central charge, L is the size of the total system, a the lattice spacing and l the size of the subsystem. There are similar relations for CFTs at finite temperatures and for the Renyi entropies [7]. It turns out that this is one of the most efficient ways to calculate the central charge c, which increases the interest in this quantities even more. For the Heisenberg model, the pre-factor of the logarithmic divergence is expected to be $\frac{1}{3}$ which makes the central charge $c = 1$ [2]. In non-critical regions the entropy obeys an area law and scales with the size of the boundary between the 2 regions [17]. The scaling of entropies is not restricted to condensed matter physics, but influences topics as unrelated as black holes.

The scaling behavior of entropies and entropy-like measures to observe phase transitions has been investigated intensively [2] [18] [19] [3].

4.1 Introduction to Entanglement

Now we want to turn away from phase transitions and take a look at the RDM from information theoretic aspects, namely its connection with entanglement. For the field of Quantum Information entanglement became a resource with which exciting things like sharing cryptographic keys, quan-

tum dense coding or quantum teleportation can be performed. This chapter follows [20] until section 4.4.
We will see later that the von Neumann entropy is only a good measure of entanglement when the complete density matrix ρ is a pure state i.e. $\rho = |\psi\rangle \langle\psi|$. From the point of view of the Monte Carlo simulation (thermal equilibrium) this is only true for zero temperature. This would mean a great loss of potential, since it is even easier to sample the RDM for higher temperatures. For mixed states the definition and evaluation of entanglement becomes much more complex. This and the fact that entanglement seems to be one of the most non-classical features in quantum mechanics - therefore also one of the most interesting - made me invest some time into research what entanglement actually is and how to measure it. In this work I only consider bipartite entanglement i.e. entanglement shared between 2 subsystems. For a work on multi-partite entanglement see [20].

Already at the beginning of quantum mechanics people noticed that strange physical states are possible which seemed to contradict common sense. One such state can be seen in equation 4.8, and physicists are so used to seeing this kind of states that at first sight it doesn't even seem to be odd - but, as we will see, they definitly are!
Because of this Einstein, Podolski and Rosen (EPR) assumed that the description of reality by quantum mechanics is not complete [21]. Their idea was that quantum mechanics contains so called *hidden variables* which restore locality and causality. In 1964 Bell showed that there is a measurable difference between a theory containing hidden variables and quantum mechanics with regard to correlation functions [22]. This led to the famous Bell inequalities which were experimentally tested in favor of quantum mechanics [23].
Consider a system of 2 qubits. A generalization of the Bell inequalities, which is more useful than the original one, is the so called CHSH inequality [24] which imposes an upper bound to the expectation value of the so called CHSH operator B_{CHSH}:

$$B_{CHSH} = A_1 \otimes (B_1 + B_2) + A_2 \otimes (B_1 - B_2), \tag{4.4}$$

where $A_1 = \vec{a}_1\vec{\sigma}$, $A_2 = \vec{a}_2\vec{\sigma}$, $B_1 = \vec{b}_1\vec{\sigma}$ and $B_2 = \vec{b}_2\vec{\sigma}$. a_i and b_i are unity vectors defining the direction of measurement A_i and B_i, while $\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ is the vector of pauli matrices. If the system is in state ρ obeying a local hidden variable theory the CHSH inequality states:

$$|\langle B_{CHSH} \rangle| = |tr\big(B_{CHSH} \cdot \rho\big)| \leq 2. \tag{4.5}$$

Quantum mechanics on the other hand gives a different inequality:

$$|\langle B_{CHSH} \rangle_{QM}| = |tr\big(B_{CHSH} \cdot \rho_{QM}\big)| \leq 2\sqrt{2}, \tag{4.6}$$

if the system is in the quantum mechanical state ρ_{QM}.
Thus quantum mechanics allows for more correlations being present in a state than local hidden variables do.

4.2 What is Entanglement?

Consider a bipartite system with the spatial partitions A and B. We call a state $|\psi\rangle$ entangled (not entangled or separable) if it cannot (can) be written as $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$, with $|\psi_A\rangle$ ($|\psi_B\rangle$) being a state vector of system A (B) respectively. For example, the state $|11\rangle = |1\rangle \otimes |1\rangle$ has no entanglement, while $\frac{1}{\sqrt{2}}\big(\,|10\rangle - |01\rangle\,\big)$ is entangled. The entanglement of the latter is actually the maximal value possible for a 2-qubit system. It's important to notice that entanglement makes only sense in the context of the partitions of the total system. For example, if we add a third system C and take look at the state:

$$|\psi_{ABC}\rangle = \frac{1}{\sqrt{2}}\big(\,|10\rangle_{AB} - |01\rangle_{AB}\,\big) \otimes |1\rangle_C \tag{4.7}$$

it is immediately obvious that it is not entangled with regard to a bi partition (AB,C) but it is when a bi partition (A,CB) is used.
Furthermore it is very important to use local degrees of freedom to define separability. If we allow the definition of separability to be used in any basis a state which actually is entangled could become not entangled in a different basis.

Assume the 2 qubits prepared in the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}\big(\,|10\rangle_{AB} - |01\rangle_{AB}\,\big) \tag{4.8}$$

are shared between Alice and Bob. If Alice performs a measurement on her qubit and finds it in the state $|1\rangle$ she knows [1] that Bob now has the state $|0\rangle$ on his qubit. This means that a measurement on one qubit can have an impact on the other at a macroscopic distance away. This non-locality was the main reason why there were doubts about quantum mechanics by EPR.

For mixed states the situation becomes a bit more complex. One says a bipartite mixed state is separable (not entangled)[2] if it can be written as [20]:

$$\rho = \sum_i p_i \cdot \rho_A^{(i)} \otimes \rho_B^{(i)} \tag{4.9}$$

, with ρ_A and ρ_B being density matrices of the corresponding subsystems and $\sum_i p_i = 1$. If ρ cannot be written in the form 4.9 one says that it is entangled. This definition of mixed state entanglement is a generalization of the definition of pure state entanglement, since for every separable pure state $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$:

$$\rho = |\psi\rangle\,\langle\psi| = |\psi_A\rangle\,\langle\psi_A| \otimes |\psi_B\rangle\,\langle\psi_B|\,,$$

which means that this state is also separable in terms of mixed state entanglement.

The reason why we use 4.9 as the definition of entanglement is that these kind of states can be produced locally in two separate laboratories A and B only by using classical communication (see also the LOCC paradigm presented in chapter 4.4)

[1] It is important to note that Bob does not know!

[2] There can still be non classical effects in separable states, see chapter 4.5.

4.3 Entanglement Measure for Pure States

It was shown [25] that the the von Neumann entropy of a reduced state is greater than that of the original (not-reduced) one only when this state is entangled.

This is true for pure and mixed states, but only in the former case this results in a convenient way to measure entanglement, since the von Neumann entropy of a pure state is zero. This means:

$$S(\rho_A) \geq S(\rho) = 0 \tag{4.10}$$

if and only if the state is entangled.

This property makes the von Neumann entropy of the reduced density matrix a good measure of entanglement when the state is pure.

Equation 4.10 is actually a very surprising result when one thinks about the classical counterpart of the von Neumann entropy, the Shannon entropy:

$$H(X) = -\sum_{i=1}^{N} p_i \log p_i \tag{4.11}$$

of some random variable X. It has the meaning of the number of bits needed to communicate a message taken from an alphabet $X = \{x_1, \cdots, x_N\}$ with probabilities $\{p_1, \cdots, p_N\}$. In other words, the Shannon entropy is the information stored in a message composed of letters x_i occurring with probability p_i. For the Shannon entropy the inequalities:

$$H(X,Y) \geq H(X), \quad H(X,Y) \geq H(Y) \tag{4.12}$$

always hold. They mean that the information in a total system (X,Y) is always greater than the information in (X) or (Y) alone. This is not surprising since it obeys common sense. But what if these inequalities are violated? When we think about information as being complexity this means that a subsystem is more complex than the total system. Taking a look at state 4.8

and tracing out the degrees of freedom of subsystem B we arrive at:

$$\rho_A = tr_B|\psi\rangle\langle\psi| = \frac{1}{2}\Big(|1\rangle\langle 1| + |0\rangle\langle 0|\Big) \tag{4.13}$$

which is a mixed state. Subsystem A is in the state $|1\rangle$ and in the state $|0\rangle$ with the same probability whereas the total system is with probability 1 in the state $|\psi\rangle$. So roughly speaking the complexity of the subsystem is somewhat greater than that of the combined system. This is was Schrödinger meant by: *Best possible knowledge of the total does not imply best possible knowledge of its parts and that is at the root of the whole problem.*[26]

4.4 Entanglement Measures for Mixed States

When dealing with mixed states, the von Neumann entropy of the reduced system is no longer a good measure of entanglement. This is because it becomes the thermodynamic entropy which has nothing to do with entanglement. This becomes clear if one takes a look at the state $\rho = \frac{1}{2}\big(|00\rangle\langle 00| + |11\rangle\langle 11|\big)$. According to definition 4.9 this state is not entangled, but the von Neumann entropy is $\log 2 \neq 0$[3]. So S^{VN} is no useful entanglement measure anymore and we have to find something else.

Before we talk about the actual value of entanglement of mixed states we need to introduce one more concept, the so called LOCC paradigm. It means Local Operations and Classical Communications. Again think about a qubit shared between Alice and Bob. Under LOCC, both can do any kind of measurements in their laboratory but whatever they do they can only use operations which act local in their laboratories. [4]. In addition to this they are allowed to communicate the result of these local measurements to each other and talk about which measurements to take next (classical communications). This allows them to produce classical correlations between the states. If we accept the non-local nature of entanglement it is clear that no

[3] The reduced state $\rho_A = \frac{1}{2}\big(|0\rangle\langle 0| + |1\rangle\langle 1|\big)$, what makes the von Neumann entropy $S(\rho_A) = -0.5 \cdot \ln 0.5 - 0.5 \cdot \ln 0.5 = \ln 2$

[4] This includes appending a new system from their laboratory, say some more particles, to their qubit.

entanglement should be produced under LOCC. Now we can understand the definition of mixed state entanglement equation 4.9. States that can be written like that, can be produced locally in Alice's and Bobs laboratory under the LOCC paradigm, without the need to exchange any entanglement. More about LOCC can be found in 27.

But how does one quantify the amount of entanglement? One approach would be to define some measure $E(\rho)$ which obeys certain axioms following 27:

1. A bipartite entanglement measure E is a mapping from density matrices into positive real numbers i.e. $\rho \to E(\rho) \in \mathbb{R}^+$
2. $E(\rho) = 0$ if the state is separable
3. If the state gets transformed according to LOCC, $E(\rho)$ cannot increase on average (no entanglement can be produced reliably although some outcomes of LOCC can have more entanglement than before)
4. For pure states, $E(\rho)$ reduces to the von Neumann entropy

4.4.1 Common Ways of Measuring Mixed State Entanglement

There are many entanglement measures, and I will only introduce the most important ones. The problem with nearly all mixed state entanglement measures is that they require a complicated minimization, which is called convex roof expansion in literature [20]. An example of such a convex roof measure can be seen in equation 4.14. This makes them very hard to calculate even if one knows the total state. Furthermore they do not depend on the RDM anymore, but are a property of the whole state, so one needs to know the complete density matrix.

Entanglement of Formation

This measure was first introduced by Wootters [28] in 2001. Consider an arbitrary state ρ and a single decomposition of it in a linear combination of pure states: $\rho = \sum_i p_i \left|\phi_i\right\rangle \left\langle\phi_i\right|$. We define the set of all possible decompositions of the density matrix as $\mathcal{C}_\rho = \{p_n,\ \left|\phi_n\right\rangle \mid \rho = \sum_i p_i \left|\phi_i\right\rangle \left\langle\phi_i\right|\}$. The entanglement of Formation is then defined as:

$$E_F = \min_{\mathcal{C}_\rho} \left(\sum_i p_i \cdot S^{VN}\big(tr_B \left|\phi_i\right\rangle \left\langle\phi_i\right| \big) \right) \tag{4.14}$$

Roughly speaking this means one has to find the decomposition of ρ which has the minimal average von Neumann entropy. The minimization is required since different decompositions yield different results. For example the, according to equation 4.9, non entangled state $\frac{1}{2}\big(\left|00\right\rangle \left\langle 00\right| + \left|11\right\rangle \left\langle 11\right| \big)$ can either be produced from an equal mixture ($p_1 = p_2 = 0.5$) of the states $\left|00\right\rangle$ and $\left|11\right\rangle$ or from the states $\frac{1}{\sqrt{2}}\big(\left|00\right\rangle + \left|11\right\rangle \big)$ and $\frac{1}{\sqrt{2}}\big(\left|00\right\rangle - \left|11\right\rangle \big)$. In the former case ρ has no entanglement since $\left|00\right\rangle$ and $\left|11\right\rangle$ are both separable, while in the latter equation 4.14 would yield:

$$\frac{1}{2} S^{vN}\left(\frac{1}{2}\big(\left|1\right\rangle \left\langle 1\right| + \left|0\right\rangle \left\langle 0\right| \big)\right) + \frac{1}{2} S^{vN}\left(\frac{1}{2}\big(\left|1\right\rangle \left\langle 1\right| + \left|0\right\rangle \left\langle 0\right| \big)\right) = \log 2 \tag{4.15}$$

since both states have the same reduced state $\frac{1}{2}\big(\left|1\right\rangle \left\langle 1\right| + \left|0\right\rangle \left\langle 0\right| \big)$

So which one do we choose?
When thinking about LOCC it is clear that we have to take the minimal value, since we are able to prepare the complete state ρ by sharing only this minimal value of entanglement.
At first calculating the entanglement of formation looks not really doable since there is an infinite number of possible decompositions of any state. Luckily there exists a bound on the number of states needed in the decomposition. Uhlmann [29] showed that the maximal number of states one has to include to get the minimal value of 4.14 is the square of the rank of ρ. This is still a pretty challenging task, but for smaller systems it is possible

to calculate. A way of doing this numerically by using a modified conjugate gradient method is given in 30.

In addition to this there are closed formulas in case of a 2 qubit-system. Wootters showed [31] that in this case the entanglement of formation is given by:

$$\begin{aligned} E_F &= h\left(\frac{1+\sqrt{1-C^2}}{2}\right) \\ h(x) &= -x\log x - (1-x)\log(1-x), \end{aligned} \tag{4.16}$$

with the concurrence C which is a entanglement measure by itself.

Concurrence

The concurrence is defined as:

$$C(\rho) := max\big(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4\big), \tag{4.17}$$

where λ_i are the square roots of the eigenvalues of the matrix:

$$\sqrt{\rho} \cdot (\sigma_y \otimes \sigma_y) \cdot \rho^* \cdot (\sigma_y \otimes \sigma_y) \cdot \sqrt{\rho}$$

with ρ^* as the complex conjugate of the state ρ in the S^z-basis and $\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ the Pauli matrix.

Since the entanglement of formation is a monotonous function of the concurrence, the latter itself can be used as an entanglement measure. The 2 site density matrix of the Heisenberg anti-ferromagnet takes the form (see chapter 6.1):

$$\rho_A = \begin{pmatrix} X & 0 & 0 & 0 \\ 0 & Z & Y & 0 \\ 0 & Y & Z & 0 \\ 0 & 0 & 0 & X \end{pmatrix} \tag{4.18}$$

with $X, Y, Z \in \mathbb{R}$ and $2X + 2Z = 1$. In this state the concurrence is simply [32]:

$$C = \max\Big(0,\ 2 \cdot (|Y| - X)\Big) \tag{4.19}$$

In chapter 6.1 we will see that the entries in the RDM correspond to a sum of correlation functions. For the case of 2 spins they simply are: $X = 0.25 + \langle S_i^z S_j^z \rangle$, $Z = 0.25 - \langle S_i^z S_j^z \rangle$ and $Y = \langle S_i^+ S_j^- \rangle$. Plugging in, doing a little algebra and using $\langle S_i^+ S_j^- \rangle = \langle S_i^z S_j^z \rangle$, which is true for the isotropic Heisenberg model, yields:

$$C = \begin{cases} 0, & \text{if } \langle S_i^z S_j^z \rangle \geq -\frac{1}{8} \\ -2 \cdot \big(2\langle S_i^z S_j^z \rangle + 0.25\big)\,, & \text{if } \langle S_i^z S_j^z \rangle < -\frac{1}{8} \end{cases}$$

Negativity

Another measure for entanglement is the so called Negativity introduced by Vidal and Werner [33]. Its major advantage is that it is easily calculated once one knows the density matrix. It is defined as:

$$N(\rho) := \frac{||\rho^{T_A}|| - 1}{2} \tag{4.20}$$

ρ^{T_A} is the matrix obtained by partial transposition of the degrees of freedom belonging to A and $||X|| = tr\sqrt{X^\dagger X}$ denotes the trace-norm. Partial transposition of a matrix $\rho = \sum_{i,j,m,n} c_{i,j,m,n} \left|i\right\rangle \left\langle j\right| \otimes \left|m\right\rangle \left\langle n\right|$ is defined by: $\rho^{T_A} := \sum_{i,j,m,n} c_{i,j,m,n} \left|i\right\rangle \left\langle j\right| \otimes \left|n\right\rangle \left\langle m\right|$.

From equation 4.9 we can easily see that in case of separable states, partial transposition results in another valid state. The trace-norm of such a state is 1, so N=0. This means that the negativity measures *how much the partially transposed state is not a state itself.* Unfortunately there is a drawback to it. There are certain states, so called bound-entangled states [34], which have zero negativity but are not separable and contain entanglement which

is called bound entanglement. So positivity[5] of the partially transposed state is generally not a sufficient condition for separability[6]. On the other hand, bound entanglement is not as useful as its "normal" counterpart since it is not possible to produce (distillation or purification are the terms normally used for that) any maximal entangled pairs from bound entangled states [20]. See [35] for the pioneering work on purification.
Furthermore the Negativity is not additive in the sense that $N(\rho_1 \otimes \rho_2) \neq N(\rho_1) + N(\rho_2)$. This can be cured by using of the Logarithmic Negativity:

$$E_N(\rho) := \log_2 ||\rho^{T_A}||. \tag{4.21}$$

But it has also has some drawbacks to it [33].
For the 2-site RDM shown in equation 4.18, the negativity can be calculated exactly. Partial transposition yields:

$$\rho_A^{T_A} = \begin{pmatrix} X & 0 & 0 & Y \\ 0 & Z & 0 & 0 \\ 0 & 0 & Z & 0 \\ Y & 0 & 0 & X \end{pmatrix} \tag{4.22}$$

This matrix has the eigenvalues $\lambda_1 = \lambda_2 = Z$, $\lambda_3 = X + Y$ and $\lambda_4 = X - Y$. The easiest way to calculate the trace norm is to sum the absolute value of the eigenvalues of the partially transposed state. Since $X \geq 0$ and $Z \geq 0$, doing this yields:

$$||\rho_A^{T_A}|| = -0.75 - 2\langle S_i^z S_j^z \rangle + |2\langle S_i^z S_j^z \rangle + 0.25|.$$

This makes the Negativity:

$$N = \begin{cases} 0, & \text{if } \langle S_i^z S_j^z \rangle \geq -\frac{1}{8} \\ -\Big(2\langle S_i^z S_j^z \rangle + 0.25\Big), & \text{if } \langle S_i^z S_j^z \rangle < -\frac{1}{8}, \end{cases} \tag{4.23}$$

[5] Positive spectrum.

[6] Only in the cases $\dim(H_1) \otimes \dim(H_2) = 2 \otimes 2$ or $2 \otimes 3$, positivity of the partial transposition is necessary and sufficient for separability [20].

showing that for RDMs of the form 4.18, the Concurrence is two times the negativity. This means that both measures are equivalent in this case.

The Negativity and the Logarithmic Negativity are no perfect measures of entanglement, but because of their simplicity they can be very useful tools for example in detecting phase transitions[36]. For example, the Logarithmic Negativity has been shown to be an universal property at the quantum phase transition of the Lipkin-Meshkov-Glick model [37]. Furthermore, the Negativity is a bound to how well a single copy of a state can be used for quantum teleportation [33] and the Logarithmic Negativity is a bound to the so called Distillable entanglement [33] [7]. Like with the Renyi entropies, the moments of the partially transposed state can be simulated with Monte Carlo (i.e.: $tr(\rho^{T_A})^n$, with $n \in \mathbb{N}$) using the replica trick [38].

4.5 Fundamental Difference between Quantum and Classical Correlations

The CHSH-Bell inequalities, equations 4.6 and 4.5 suggest that there are more correlations possible in a quantum state than in a state stemming from classical world with local hidden hidden variables. The excess correlations are (mostly) attributed to entanglement. One might ask whether there is a difference between correlations stemming from entanglement and classical ones, and is there a way separate correlations? The first question is answered easily, since with entanglement one is able to do things which wouldn't be possible in the classical world, like for example quantum teleportation [39]. Before we can address the second question we first need to understand what a classical state actually is, and how it is different from a quantum mechanical one. This topic is very well explained in 40 and their notation will be used here.

Take a system consisting of N sites. We define an element of the set of

[7] The Distillable entanglement is another way to measure entanglement. It is not presented in this work, for its definition see [20].

product states $\mathcal{P}$ as:

$$\pi := \pi_1 \otimes \cdots \otimes \pi_N, \tag{4.24}$$

with π_j being the local state (reduced density matrix) of site j. These states have no correlations at all.
Next, the set of classical states $\mathcal{C}$ is defined as all states that can be written as:

$$\chi := \sum_{\vec{k}} p_{\vec{k}} \left| k_1, \cdots, k_N \right\rangle \left\langle k_1, \cdots, k_N \right|, \tag{4.25}$$

with $p_{\vec{k}}$ the joint probability distribution of the *local* state $|k_1, \cdots, k_N\rangle$. Which means that the k_n are some local quantum numbers.
One might ask: Cannot every state be written like that, since every density matrix is diagonalizable? The answer to this lies in the word *local.* In the Heisenberg anti-ferromagnet the local variables would be the S^z-component of the spin on a certain site, so $k_i \widehat{=} S_i, \forall i$. Since the Heisenberg Hamiltonian is not diagonal in the S^z-basis most states are not either. Hence a typical state of the Heisenberg Model is not part of the set of classical states $\mathcal{C}$.
Correlations of the probability distribution $p_{\vec{k}}$ in states like 4.25 are completely classical.
Totally Separable states on the other hand can be written as (note that in the definition of mixed state entanglement 4.9 we looked only at one bi partition, whereas here we think about total separability, meaning separability in any bi partition):

$$\sigma := \sum_i p_i \cdot \pi_1^i \otimes \cdots \otimes \pi_N^i \tag{4.26}$$

and belong to the set of separable states called $\mathcal{S}$. Very often these states are said to be classical, but this is not true. The so called quantum discord is, alongside entanglement, another non-classical feature of quantum systems [41]. As a example of quantum discord take a look at the so called Werner state:

$$\rho_W(z) = \frac{1-z}{4}\mathbb{1} + z \left|\psi\right\rangle \left\langle\psi\right|, \tag{4.27}$$

with $|\psi\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big)$ and $z \in \mathbb{R}$. It is separable for $z < \frac{1}{3}$ (see 41) while it still contains non classical features for any $z > 0$. These features are

quantified in the quantum discord which we will not define nor discuss here (see 41 for its definition).
Finally all states which cannot be written like 4.26 are called entangled and all entangled states form the set $\mathcal{E}$.
To distinguish different correlations, some kind of measure of distance is needed: the relative entropy between 2 states x and y is defined as:

$$S(x||y) := -tr(x \log y) - S^{vN}(x). \tag{4.28}$$

It is a measure of *distance* between 2 probability distributions x and y [42], although it is not a metric, since it is not symmetric between x and y. The idea is now to define for example entanglement of a state ρ as the minimal distance to the set of separable states:

$$E := \min_{\sigma \in \mathcal{S}} S(\rho||\sigma). \tag{4.29}$$

The same way one can define the total correlations C of a state ρ as:

$$C = \min_{\pi \in \mathcal{P}} S(\rho||\pi). \tag{4.30}$$

All these definitions still need a very complex minimization, so it is very hard to calculate them for typical states. It is not yet clear if the different forms of correlations add up the right way, as pointed out in [40], but it is still an interesting way of thinking about them and answers at least a little the second question asked in the beginning of this section.

Mutual Information

The total correlations C in equation 4.30 turns out to be equivalent to the mutual information $I(\rho)$ another important quantity in quantum information [40]:

$$I(\rho) := S^{VN}(\rho_A) + S^{VN}(\rho_B) - S^{VN}(\rho), \tag{4.31}$$

with ρ_A and ρ_B the states obtained by tracing out the degrees of freedom of bi partition A and B respectively.

Chapter 5

The Loop Algorithm for the Heisenberg Model

As it was explained in chapter 3 the partition function, and the (reduced) density matrix appear in the path integral. The difference in those 3 quantities lies just in the boundary conditions in imaginary time. Mathematically, these boundary conditions show in how many sites of the lattice get traced out from the term $e^{-\beta H}$. To sample RDMs it is necessary to choose a Monte Carlo method, which can sample from this ensemble. In this work I used the Loop Algorithm, which is a way of performing non-local updates in the *path integral*. In this chapter I will first present the worldline configuration of the partition function, because worldlines are an important concept in understanding the Loop Algorithm. How to sample RDMs by altering the Loop Algorithm will be shown in the next chapter.

5.1 Worldine representation of the partition function

Consider, for simplicity, a 1 dimensional[1] isotropic Heisenberg Anti-ferromagnetic model with N sites. The hamiltonian governing this system is (see equation 2.2):

$$H = \sum_{\langle i,j \rangle} J\hat{S_i^z}\hat{S_j^z} + \frac{J}{2}\left(\hat{S_i^+}\hat{S_j^-} + \hat{S_i^-}\hat{S_j^+}\right),$$

where $J > 0$, S_i^z, S_j^z are spin operators, and S_i^+ (S_i^-) are the corresponding raising (lowering) operators respectively. $\langle i, j \rangle$ means the sum is restricted to nearest neighbors only. Note that the Loop Algorithm is not restricted to the isotropic Heisenberg Model and magnetic fields can be included as well. Performing an even-odd breakup of the Hamiltonian yields:

$$\hat{H} = H_{even} + H_{odd} = \sum_{i:even} H_{i,i+1} + \sum_{i:odd} H_{i,i+1}.$$

Both terms consist of commutating operators, since every bond operator only acts on two neighboring sites. The partition function can be written:

$$Z := tr(e^{-\beta \hat{H}}) = tr(e^{-\beta(H_{even}+H_{odd})}) = tr\left(\lim_{L\to\infty}(e^{-\frac{\beta}{L}H_{even}}e^{-\frac{\beta}{L}H_{odd}})^L\right) \quad (5.1)$$

, with $\beta = \frac{1}{k_b T}$ the inverse temperature, and k_b Boltzmann's Constant. The last step in this equation is the so called Suzuki-Trotter breakup [43]. Since H_{even} and H_{odd} are sums of commuting operators, the exponential factorizes resulting in a product of exponentials of operators acting only on two

[1] The extension to higher dimensions is straightforward.

neighboring sites. Introducing $\Delta\tau = \frac{\beta}{L}$ yields:

$$\begin{aligned} Z &= tr\Bigg(\lim_{N\to\infty} \big(\prod_{i:even} e^{-\Delta\tau H_{i,i+1}} \prod_{i:odd} e^{-\Delta\tau H_{i,i+1}} \big)^{L} \Bigg) \\ &= \sum_{S_1=\pm1,\ldots,S_N=\pm1} \langle S_1, \ldots S_N| \big(\prod_{i:even} e^{-\Delta\tau H_{i,i+1}} \prod_{i:odd} e^{-\Delta\tau H_{i,i+1}} \big)^{L} |S_1, \ldots S_N\rangle \,. \end{aligned} \tag{5.2}$$

By inserting a complete set of eigenstates between every exponential, one arrives at the worldline representation of the partition function:

$$\begin{aligned} Z = &\sum_{\{S_{il}\}} \langle S_{1,2L}, \ldots S_{N,2L}| e^{-\Delta\tau H_{2,3}} |S_{1,2L-1}, \ldots S_{N,2L-1}\rangle \langle S_{1,2L-1}, \ldots S_{N,2L-1}| \\ &\cdots e^{-\Delta\tau H_{4,5}} |S_{1,2L-2}, \ldots S_{N,2L-2}\rangle \langle S_{1,2L-2}, \ldots S_{N,2L-2}| e^{-\Delta\tau H_{6,7}} \\ &\cdots \langle S_{1,2}, \ldots S_{N,2}| e^{-\Delta\tau H_{N-1,N}} |S_{1,1}, \ldots S_{N,1}\rangle \,, \\ = &\sum_{\{S_{il}\}} \prod_{p} W_p\{S_p\} \end{aligned} \tag{5.3}$$

with $S_{1,2L} = S_{1,1}$, $S_{2,2L} = S_{2,1}$ ... $S_{N,2L} = S_{N,1}$.
The limit was dropped at this point, since the error one makes is of order $(\Delta\tau)$ and we assume L to be sufficiently large [2]. The sum is over all spin-configurations $\{S_{il}\}$ on a (1+1) dimensional lattice with indices $i = 1...N$ and the imaginary time index $l = 1...2L$. The product in the last line of equation 5.3 on the other hand, results from the individual exponentials, with the index p denoting the 4 spins with indices (i,l), (i+1,l), (i,l+1), (i+1,l+1) contributing to this term.

The weight W_p of one configuration is given by:

$$\begin{aligned} W_p = &\langle S_{1,l+1}, \ldots., S_{i,l+1}S_{i+1,l+1}, \ldots, S_{N,l+1}| e^{-\Delta\tau \hat{H}_{i,i+1}} |S_{1,l}, \ldots., S_{i,l}S_{i+1,l}, \ldots, S_{N,l}\rangle \\ = &\delta_{S_{1,l+1},S_{1,l}} \cdots \delta_{S_{i-1,l+1},S_{i-1,l}} \delta_{S_{i+2,l+1},S_{i+2,l}} \cdots \delta_{S_{N,l+1},S_{N,l}} \\ &\langle S_{i,l+1}S_{i+1,l+1}| e^{-\Delta\tau \hat{H}_{i,i+1}} |S_{i,l+1}S_{i+1,l+1}\rangle \end{aligned} \tag{5.4}$$

[2] Later the limit $\Delta\tau \to 0$ will be made anyway, so I neglect the error in the discrete case.

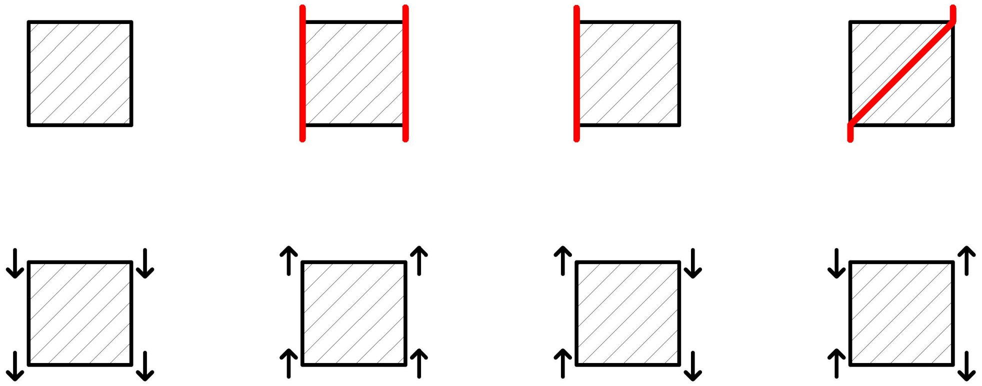

Figure 5.1: Possible plaquette configurations. The 2 configurations missing (there are 6 in total) are obtained by mirroring the third and fourth plaquette at the vertical axis. The bottom row shows the spin states, while in the top row the corresponding worldline-configurations are shown. Worldlines are obtained by connection the up-spin along plaquettes. From the left to the right the plaquettes correspond to the first, second, third and fourth line in equation 5.5

With other words, a single plaquette can only make local spin changes on 4 spins ($S_{i,l}$, $S_{i+1,l}$ $S_{i,l+1}$ and $S_{i+1,l+1}$), which are called the active spins of the plaquette. Therefore, there are 16 different configurations for each plaquette. From these only 6 have a non-zero contribution to the partition function, since every bond-Hamiltonian $H_{i,i+1}$ conserves total spin. The 6 non-zero weights are:

$$\begin{aligned}
\langle --| \, e^{-\Delta\tau \hat{H}_{i,i+1}} \, |--\rangle &= e^{-\frac{\Delta\tau}{4}J} \\
\langle ++| \, e^{-\Delta\tau \hat{H}_{i,i+1}} \, |++\rangle &= e^{-\frac{\Delta\tau}{4}J} \\
\langle +-| \, e^{-\Delta\tau \hat{H}_{i,i+1}} \, |+-\rangle = \langle -+| \, e^{-\Delta\tau \hat{H}_{i,i+1}} \, |-+\rangle &= e^{\frac{\Delta\tau}{4}J} \cosh \frac{\Delta\tau J}{2} \\
\langle +-| \, e^{-\Delta\tau \hat{H}_{i,i+1}} \, |-+\rangle = \langle -+| \, e^{-\Delta\tau \hat{H}_{i,i+1}} \, |+-\rangle &= -e^{\frac{\Delta\tau}{4}J} \sinh \frac{\Delta\tau J}{2}
\end{aligned} \tag{5.5}$$

, with $|+\rangle$ meaning a spin-up state and $|-\rangle$ spin down. The graphical representations of these configurations are shown in figure 5.1.

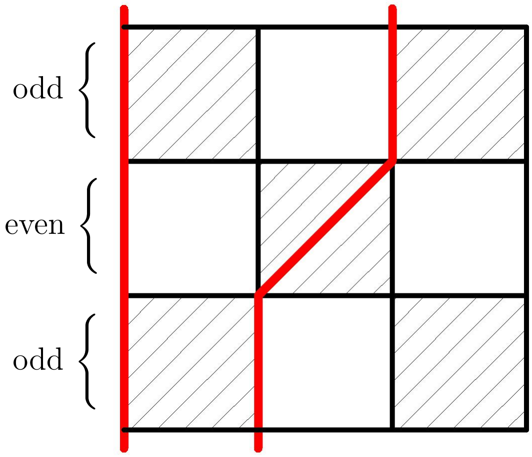

Figure 5.2: A segment of the checkerboard lattice with 2 worldlines (red lines). The shaded areas are active plaquettes representing one bond-operator $e^{-\Delta\tau\hat{H}_{i,i+1}}$. Starting with a spin-state $|++--\rangle$ at the bottom a set of odd bond-operators acts and the spins stay the same. Next the even bond-operators act changing the state into $|+-+-\rangle$. The last row of odd bond-operators again does nothing to the state.

So what is the total effect of all the plaquettes? Let us read equation 5.3 from the right to the left. First, one starts with a spin-state $|S_{1,1},...S_{N,1}\rangle$ onto which a set of odd bond-operators $e^{-\Delta\tau H_{i,i+1}}$ acts. The resulting spin state then acts on a set of even bond operators, and its outcome again on odd bond-operators and so on. Finally one arrives at $\langle S_{1,2L},...S_{N,2L}|$ which must be the same state as the start-vector $|S_{1,1},...S_{N,1}\rangle$.

From a graphical viewpoint this can be modeled by a checkerboard lattice in one space ,and one imaginary time dimensions, where each exponential corresponds to a plaquette (shaded field of the checkerboard). A segment of one such checkerboard can be seen in figure 5.2. The offset of the plaquettes is a result of the first $\frac{N}{2}$ plaquettes at time l only acting on sites $(i, i+1)$ with i odd, while the next $\frac{N}{2}$ on sites $(i, i+1)$ with an even i. The trace of the partition function is responsible for the spins being the same for $l = 1$ and $l = 2L$ for all sites. So the checkerboard can also be viewed as a checkerboard-cylinder for open boundaries, or a donut for periodic boundary conditions.

So what changes can the individual bond operators i.e.: a single plaquette cause? Starting with an up-spin on some site i, a plaquette can only shift it to $i + 1$ or $i - 1$ or do nothing so it stays at i, but the up spin can never vanish (since every bond operator conserves total spin). This, with the periodic boundaries in imaginary time, leads to closed lines of spin-up which are called worldlines. The same, of course, is true for spin-down, and one could identify them as the worldlines, but one usually thinks about worldlines as regions with an up spin. Knowing all the worldlines is equivalent to knowing the whole spin configuration of the space-time lattice, since on every point without a worldline the spin has to point downwards.

The requirement of the worldlines to be closed can only be met if the number of jumps every worldline does is an even number (one needs to be careful with pbc and an odd N). This means, that the minus sign in the last row[3] of equation 5.5 appears an even number of times. So this sign has no effect on the total weight of the configuration and can be neglected. Later, when introducing the algorithm which can sample RDMs, this minus sign will become important again.

5.2 Continuous Imaginary Time and Loop Algorithm

With this understanding of what a single configuration looks like, we can start thinking about a way to do Monte Carlo. We need some procedure that produces new configurations. It has to be ergodic, and should fulfill detailed balance with respect to two consecutive spin configurations. The first idea would be to use local deformations of world lines, which is quite ineffective. A much better way to produce a new configuration is the Loop Algorithm. It samples new world line configurations by constructing and

[3] Note that the last row of equation 5.5 is responsible for jumping worldlines, i.e.: it moves the spin-up.

flipping all spins along a whole region of the space-time lattice, a so called loop. The decision making which direction a loop moves is still local tough. These kind of updates reduce autocorrelations between successive configurations, and it is possible to sample the great canonical ensemble, (different magnetization) as opposed to the local updates [1].

Practically, one usually would work with a finite, but small, $\Delta\tau$ resulting in methodical errors. It is possible tough to use continuous imaginary time, removing these errors. Generally, when using continuous imaginary time, one doesn't think in terms of plaquettes anymore, but rather defines configurations by times when worldlines jump. See figure 5.3 for an example of a configuration in imaginary time.
Beard and Wiese showed that it is possible to take the limit $\Delta\tau \to \infty$ [44]. Equivalently, one can directly employ a poisson-process representation of $e^{-\beta H}$ in continuous imaginary time [45].

To get a rough idea about the Loop Algorithm lets go down the second road (following [1]). Lets start with a shifted bond operator, appearing in equation 2.2:

$$\begin{aligned} -\frac{1}{J}H_{ij} + \frac{1}{4} &= -\big(S_i^z S_j^z - \frac{1}{4}\big) - \frac{1}{2}\big(S_i + S_j^- + S_i^- S_j^+\big) \\ &= \frac{1}{\sqrt{2}}\big(\,|{+}{-}\rangle - |{-}{+}\rangle\,\big) \cdot \frac{1}{\sqrt{2}}\big(\,\langle{+}{-}| - \langle{-}{+}|\,\big) \\ &= \frac{1}{2}S_{ij}. \end{aligned} \tag{5.6}$$

Equation 5.6 means, that a single bond hamiltonian can be written as a singlet projection operator S_{ij}. The total hamiltonian shifted by a constant is then

$$H + const = -\frac{J}{2}\sum_{\langle i,j\rangle} S_{ij} = -\sum_b J_b S_b, \tag{5.7}$$

where in the last term the summation of nearest neighbor pairs $\langle i,j\rangle$ was replaced by a summation over bonds b and $J_b = \frac{J}{2}$. For Hamiltonians of this form $e^{-\beta H}$ can be written as a poisson integral, as proved by [45]:

$$\begin{aligned} e^{-\beta H} &= e^{\beta \sum_b J_b} \lim_{\Delta\tau \to 0} \left(\prod_b e^{-(J_b + J_b S_b)\Delta\tau} \right)^{\beta/\Delta\tau} \\ &= e^{\beta \sum_b J_b} \lim_{\Delta\tau \to 0} \left(\prod_b \{(1 - J_b \Delta\tau) + J_b S_b \Delta\tau\} \right)^{\beta/\Delta\tau} \\ &= e^{\beta \sum_b J_b} \int \rho(d\omega) \prod^* S_b, \end{aligned} \tag{5.8}$$

where $\prod^* S_b$ is a time ordered product of bond operators S_b, ω is the bond configuration, and $\rho(d\omega)$ is a poissonian probability measure with density $\prod_b J_b dt$. Thus, we get a random countable collection of time-indexed bonds, which occur independently in disjoint regions of spacetime.

For our purposes this means that $e^{-\beta H}$ can be written as a integral over configurations of singlet projection operators. On every bond b there is a constant probability density $\frac{J}{2}$ that an operator appears. At all times, at which there is no singlet-operator an identity operator can be introduced. Quantizing this in S^z-basis and performing the trace yields lines between the operators. Following those lines, and changing direction and position when hitting a bond operator produces closed loops. Because the Bond operators are singlet projection operators, along every loop there are two possible, equally likely, staggered spin configurations.

With this the Multi Loop Variant of the Loop Algorithm is easily obtained:

1. Starting with some spin configuration, choose a compatible operator configuration. To do this introduce with constant probability density $\frac{J}{2}$ at every time interval, singlet projection operators, if the spins are different. At points at which the worldlines jump, a singlet operator appears with certainty.

2. Forget the current spin configuration and choose a new one, compatible with this operator configuration.

At first the algorithm I used was the single loop variant tough. Instead of constructing a complete new operator configuration, it produces only one of these loops and changes the spin in the region of the space time lattice it transverses with certainty. A recipe for it is as follows [1]:

- Choose an arbitrary site j, and start the loop at an arbitrary time t_1. Move upwards (downwards) in time if the spin points upwards (downwards) respectively
- Determine the minimal time interval Δt, in which the worldline-configuration of the neighbors of j and of j itself doesn't change (note that at this point we moved away from the 1d - model; this recipe is formulated for arbitrary dimensions and also applies to not only nearest neighbor couplings)
- The loop is allowed to jump to all neighbors with different spin orientation than j at time t_1, if they weren't already visited by the loop
- For every neighbor to which the loop is allowed to jump draw a random number from $\lambda e^{-\lambda t}$ with $\lambda = \frac{J}{2}$, take t_{jump} the minimum of these numbers
- If $t_{jump} < \Delta t$ let the loop jump to this neighbor, therefore introducing a new worldline jump. In doing so, loop changes the direction it moves. Otherwise move the loop to $t_1 + \Delta t$ at site j
- Change the spin orientation along the produced loop segment (in the implementation this is only necessary when the loop jumps or closes)
- Reiterate, until the loop closes

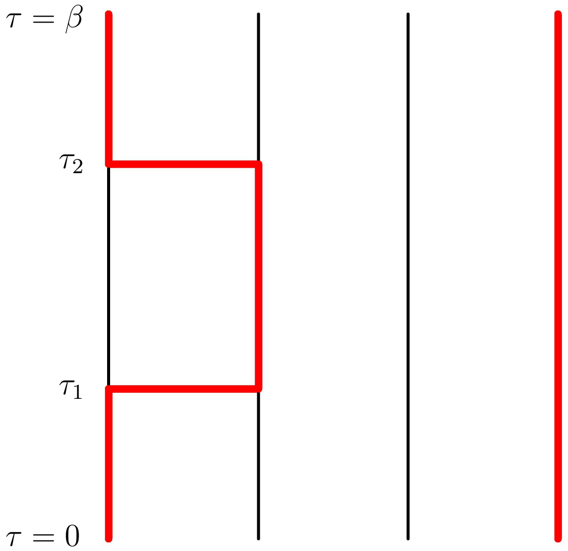

Figure 5.3: Typical configuration of a worldline Monte Carlo algorithm in continuous time. The lattice consist of 4 sites (horizontal dimension), and the imaginary time starts at the bottom, at $\tau = 0$, and ends at the top, at $\tau = \beta$. Because of the trace in equation 5.1 the spins at t=0 and $t = \beta$ have to be the same. The red lines correspond to worldlines which are regions of spin-up in the space-time-lattice, while the black lines are regions of spin-down. At time τ_1 the worldline jumps from the first site to the second and jumps back to the first at τ_2. The number of worldline-jumps in this configuration is 2. As one can see, for the worldlines to be closed, this number always needs to be even. Also note that the jumps happen *instantaneous*.

To produce this spin configuration from a singlet projection operator configuration, we note that between site 1 and 2 there are at least 2 bond operators at times τ_1 and τ_2. Straight worldlines are not necessarily a result of no bond operator tough. For example two more bond operators between sites 3 and 4 could result in the same worldline configuration, since every loop can choose from 2 different spin configurations.

If one wants to do measurements, one can average the observable over all imaginary times. If $\{S(\tau)\}$ is the spin configuration at time t, and $O(\{S\})$ is the observable depending on the spins, one can calculate: $\frac{1}{\beta}\int_0^\beta O(\{S(\tau)\})d\tau$.

Chapter 6

Density Matrix Loop Algorithm (DMLA)

The Density Matrix Loop Algorithm (DMLA) is a quantum Monte Carlo Method, based on the Loop Algorithm. It can stochastically calculate reduced density matrices (RDM) of a Heisenberg-Anti-ferromagnet. Consider the lattice being split in two disjoint regions A and B. The counting of sites is as follows: the sites $1 \cdots |A|$ belong to A (with $|A|$ the size of region A) and sites $|A| + 1 \cdots N$ belong to B. $\mathcal{A}$ is the set of spin-states belonging to A, with $\dim \mathcal{A} = 2^{|A|}$, while $\mathcal{B}$ is the set of spin-states belonging to B, with $\dim \mathcal{B} = 2^{N-|A|}$.
A specific matrix element of the RDM is according to equation 3.1:

$$c_{ij} = \langle i| \, \rho_A \, |j\rangle = Z^{-1} \sum_{|k\rangle \in \mathcal{B}} \langle i| \otimes \langle k| \left(e^{-\beta \hat{H}} \right) |j\rangle \otimes |k\rangle \tag{6.1}$$

Already without the knowledge of path integrals, equation 6.1 shows that an element of the RDM can be simulated by a worldline-configuration where sites, belonging to region A, have some kind of cut, which allows the spins to be $\langle i|$ on one side and $|j\rangle$ on the other. All sites in B on the other hand, have periodic boundary conditions in imaginary time i.e. they have no cut and therefore the same spin. This is signaled by the partial trace over all spins of region B. In case $S_i = S_i' \; \forall \; i \in A$ (the diagonal elements of ρ_A) we get a valid configuration of the normal Loop Algorithm. If the spins are different on the other hand, one has a state that isn't possible in traditional worldline Monte Carlo. Therefore, one samples from a bigger space of configurations, which also includes the normal ones i.e.: an expansion of the phase space. So it is possible (tough not efficient) to do a normal worldline Monte Carlo by including cuts, but only counting valid configurations.

RDMs can be sampled with only minor changes to the single loop variant of the traditional Loop Algorithm:

- Introduce cuts at all lattice sites belonging to region A at any (but for all cuts the same) imaginary time. Spins can be different above and below the cuts, contrary to the regular Loop Algorithm.
- When a loop moves to one of these cuts move it back to the start, and continue in the opposite direction it started. When one moved upwards on up-spins, and downwards on down-spins, one now moves in the opposite directions. Continue construction of the loop, until it reaches a cut again. Only then the loop *closes.*
- If the loop doesn't reach a cut at all, no changes to the traditional algorithm have to be made

A few remarks to this algorithm:

Because the Hamiltonian (and also $e^{-\beta \hat{H}}$) still conserves total spin, so must the RDM. This means that only those matrix elements are non-zero, that have the same number of up-spins in the bra- and ket-vector [1]. The above algorithm satisfies this constraint.

To prove this, assume that before reaching a cut the loop moves upwards on up spins, and downwards on downwards pointing spins. Furthermore assume that the loop reaches a cut coming from below. This means that the spin below the cut was up, and now points downwards, i.e.: there is one up-spin less in the ket vector. Now there are 2 possibilities: either the loop ends below a cut, or it ends coming from above. In the former case, the spin must have

[1] Consider a 2 site RDM the matrix element $|{+}{+}\rangle\langle{+}{-}|$ would always be zero while $|{+}{-}\rangle\langle{-}{+}|$ can be non-zero and $|{+}{-}\rangle\langle{+}{-}|$ can also be non-zero and is a valid configuration of the traditional Loop Algorithm.

pointed downwards (directions get changed), and an up spin is introduced. The latter situation means that an up spin gets removed in the bra vector. In both situations the number of up spins in the bra vector, and in the ket vector stays the same.

The other case that the loop reaches a cut the first time coming from above has the same outcome. Only the bra- and ket-vector have switched roles. So in all possible cases the total spin in the bra vector stays the same as in the ket vector, proving the claim that this Monte Carlo method doesn't produce any forbidden configurations.

The time at which the cuts are introduced doesn't matter. Equation 6.1 suggests that this time is $t = 0$ or equivalently $t = \beta$. But since a translation in imaginary time would not change any relative position of worldline-jumps [2] to each other, it is not a problem to move the cuts to any time. Mathematically, this shows in the invariance of the reduced trace under cyclic permutation, if the operators are real and hermitian (the complete trace is invariant under all cyclic permutation).

This algorithm samples states of the RDM $|i\rangle \langle j|$ with probability $|c_{ij}|$. The absolute value is important, because c_{ij} can be negative. Since worldlines now have a beginning and an end, the sign of the last configuration in equation 5.4 becomes important. Consider a single worldline ending in a cut. Because of conservation of total spin there needs to be a cut where a worldline begins[3]. If the distance between these two cuts is an odd number, there also must be an odd number of worldline-jumps to get there. This means that the sign of this particular worldline is negative. The product of all signs off all worldlines gives the sign of the configuration i.e.: the sign of the matrix element. To cure this one has to give the right sign to every configuration, found by looking at all sites of one bipartition of region A. If the number of sites with a different spin below and above the cut is even (odd), the configuration has a positive (negative) sign respectively. For more explanation see figure 6.1.

[2] Worldline jumps can be seen as the *physical* important quantities, because they uniquely determine a configuration

[3] A worldline ends if it moves to a cut, coming from below, and one starts if it moves from above a cut. Note that one could argue that the worldline doesn't really end, since one coming to a cut, must reappear at some site again.

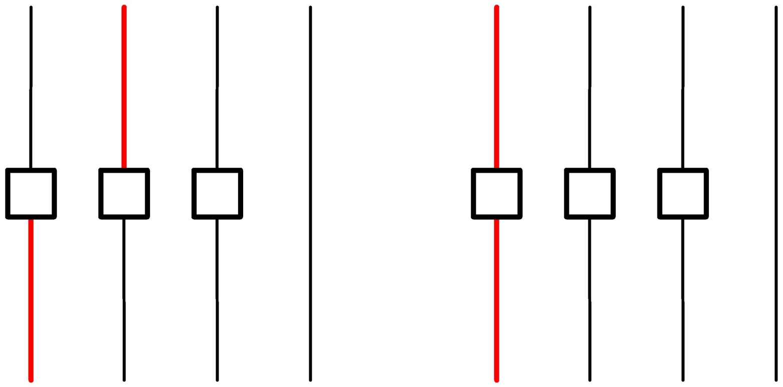

Figure 6.1: Two configurations of a 1d N=4 lattice with 3 sites having a cut indicated by the squares. Red lines again are worldlines. This is just a segment of the imaginary time dimension, so worldlines don't close in this picture. To determine the sign of a configuration we need the 2 possible bi partitions of region A. They are sites $\{1, 3\}$ and $\{2\}$ when starting to count at 1 at the leftmost site.

Left picture: This configuration contributes to the matrix element $|+--\rangle\langle -+-|$ and/or $|-+-\rangle\langle +--|$. When counting the number of different spins on one bi partition of A below and above the cuts one gets in both cases 1, so this configuration has a negative sign. Another way to look at the sign: since the worldline must do an odd number of jumps to get from the second site to the first (1 in the easiest case) this matrix element has a negative sign according to equation 5.5.

Right picture: This configuration on the other hand contributes to the matrix element $|+--\rangle\langle +--|$. Since this worldline must do an even number of jumps this matrix element has a positive sign. In this case counting the number of spins different below and above the cut gives 0 for every bi partition. Furthermore this is a valid configuration in the traditional Loop Algorithm.

6.1 Correlation Functions

The above algorithm can be used to calculate Correlation Functions. This is not very surprising, since a matrix element of the RDM can be written as:

$$\langle \tilde{S}_i, \tilde{S}_j, \cdots | \, \rho_A \, | S_i, S_j, \cdots \rangle = tr_A \Big(\, |\tilde{S}_i, \tilde{S}_j, \cdots \rangle \, \langle S_i, S_j, \cdots | \, \rho_A \Big) \tag{6.2}$$

$$= tr_A \Big(\, |\tilde{S}_i, \tilde{S}_j, \cdots \rangle \, \langle S_i, S_j, \cdots | \, tr_B \rho \Big) \; = \; \underbrace{tr_A tr_B}_{=tr_{A\cup B}=tr} \; |\tilde{S}_i, \tilde{S}_j, \cdots \rangle \, \langle S_i, S_j, \cdots | \, \rho \tag{6.3}$$

$$= tr \Big(\, |\tilde{S}_i, \tilde{S}_j, \cdots \rangle \, \langle S_i, S_j, \cdots | \, \rho \Big) \; = \Big\langle \, |\tilde{S}_i, \tilde{S}_j, \cdots \rangle \, \langle S_i, S_j, \cdots | \, \Big\rangle . \tag{6.4}$$

$$\tag{6.5}$$

and the operator $|\tilde{S}_i, \tilde{S}_j, \cdots \rangle \, \langle S_i, S_j, \cdots |$ can be constructed from $\mathbb{1}$, S^z and $S^{\pm}$ operators, as shown below. Note that in this case spin-conservations requires that $\sum_k \tilde{S}_k = \sum_k S_k$.

Correlation Functions for $|A| = 2$

For the simplest[4] case $|A| = 2$ the correlation functions $\langle S_i^z S_j^z \rangle$, $\langle S_i^+ S_j^- \rangle$ and $\langle S_i^- S_j^+ \rangle$ can be extracted from the RDM. Because of the symmetry of the Hamiltonian $\langle S_i^z S_j^z \rangle = \langle S_i^x S_j^x \rangle = \langle S_i^y S_j^y \rangle$. Furthermore, correlations of the type $\langle S_i^z S_j^+ \rangle$ or $\langle S_i^z S_j^- \rangle$ are zero because of spin conservation. This means that all 2-site correlation functions can be calculated once one knows the RDM. The 6 non-zero matrix elements are:

- $|{+}{+}\rangle \langle{+}{+}|$, $|{+}{-}\rangle \langle{+}{-}|$, $|{-}{+}\rangle \langle{-}{+}|$, $|{-}{-}\rangle \langle{-}{-}|$ $\cdots$ 4 diagonal elements
- $|{+}{-}\rangle \langle{-}{+}|$, $|{-}{+}\rangle \langle{+}{-}|$ $\cdots$ 2 off-diagonal elements

[4] Actually the simplest case is $|A| = 1$, but then one does not talk about correlation functions. Furthermore in the isotropic case all of these expectation values are zero.

The diagonal operators can be written as (remember all single site expectation values are zero, therefore these operators are omitted):

$$
\begin{aligned}
\left|{+}{+}\right\rangle\left\langle{+}{+}\right| &= (S^z_i + 0.5\cdot\mathbb{1})\otimes(S^z_j + 0.5\cdot\mathbb{1}) = S^z_i S^z_j + 0.25\\
\left|{+}{-}\right\rangle\left\langle{+}{-}\right| &= -(S^z_i + 0.5\cdot\mathbb{1})\otimes(S^z_j - 0.5\cdot\mathbb{1}) = -S^z_i S^z_j + 0.25\\
\left|{-}{+}\right\rangle\left\langle{-}{+}\right| &= -(S^z_i - 0.5\cdot\mathbb{1})\otimes(S^z_j + 0.5\cdot\mathbb{1}) = -S^z_i S^z_j + 0.25\\
\left|{-}{-}\right\rangle\left\langle{-}{-}\right| &= (S^z_i - 0.5\cdot\mathbb{1})\otimes(S^z_j - 0.5\cdot\mathbb{1}) = S^z_i S^z_j + 0.25
\end{aligned}
$$

The off diagonal operators on the other hand are simply:

$$
\begin{aligned}
\left|{-}{+}\right\rangle\left\langle{+}{-}\right| &= S^-_i S^+_j\\
\left|{+}{-}\right\rangle\left\langle{-}{+}\right| &= S^+_i S^-_j
\end{aligned}
\tag{6.6}
$$

General Case

Using the results of the previous section, it is not very hard to generalize this result:

- Take one matrix-element of the RDM and compare the spins in the bra- and ket-vector
- If the spin stays the same on site i take $S^z_i + 0.5$ if this spin is up and $-(S^z_i - 0.5)$ if it points downwards
- If the spin changes take S^+_i (S^-_i), if it changes from up to down (down to up) respectively (see equation 6.6)
- Multiply all of these operators. The coefficient of the matrix element is the expectation value of this product

For $|A| > 2$ the product results in a sum of the $|A|$-point correlation function with many lower correlation functions (if $|A| = 5$ one would get a 4-point

function plus 3- and 2-point functions). These lower correlation functions appear, because of the 0.5 term in the diagonal operators $(0.5 \pm S^z)$.
For example, take the matrix element $|+--\rangle\langle -+-|$ of a RDM describing 3 sites labeled 1 2 and 3. According to the rules, this entry corresponds to

$$\langle S_1^+ S_2^- (0.5 - S_3^z)\rangle = -\langle S_1^+ S_2^- S_3^z\rangle + 0.5 \cdot \langle S_1^+ S_2^-\rangle$$

So indeed, we end up with the 3-point correlation function $\langle S_1^+ S_2^- S_3^z\rangle$, plus the 2-point function $0.5 \cdot \langle S_1^+ S_2^-\rangle$.
This means that if one wants to calculate them, one needs to do simulations for all sizes of $|A|$ up to the highest one wants to know, and construct them by adding or subtracting the right matrix-elements.
As already mentioned in this chapter, the DMLA samples in an extended phase space with regard to the traditional Loop Algorithm. The idea, by expanding the phase space one is able to measure new observables not accessible in the not expanded case, is not new. In fact the single loop variant of the Loop Algorithm [1] is such an expansion as long as the loop doesn't close. Open loops are used to measure $\langle S_i^+ S_j^-\rangle$ very similarly to the DMLA [46]. Closely related to this is the so called Worm-Algorithm [47].

Chapter 7

Measures Related to Entanglement

As we have seen in chapter 4, there is a wide interest in the von Neumann entropy. For spin systems, this desire led to the definition of some quantities that approximate it, or has at least a similar scaling as the von Neumann entropy. In the first section of this chapter I will give an overview of the already existing ways to approximate the von Neumann entropy. The most important concept of these measures is the Valence Bond Basis, which is closely related to the loops of the Loop Algorithm. The second section of this chapter introduces some new ways to approximate the von Neumann entropy based on this relation.

7.1 Valence Bond and Loop Entropies

All of the existing measures are formulated in the Valence Bond basis, which is a massively over complete basis in the singlet space (total spin $S = 0$). One of its major advantages is that the outcome of a bond operator from the Heisenberg Model (equation 2.2) is easily calculated. This results in an efficient way to let high powers of the Hamiltonian act on a valence bond state. Therefore, one has an easy way of ground state projection.

But lets start with the actual Valence Bond basis (VB-basis), the ground state projection will be explained in more detail later. Since the ground-state projection using the valence bond basis was introduced by Sandvik [9], I use the notation of this paper.

Denote a singlet state of two spins on sites a and b:

$$(a, b) := \frac{1}{\sqrt{2}} \left(|1\rangle_a \otimes |0\rangle_b - |0\rangle_a \otimes |1\rangle_b \right) \tag{7.1}$$

The valence bond basis is a basis formed from all possible singlets between all the sites in a lattice. A single basis state in the Valence Bond basis of a lattice with N sites is:

$$(a_k^1, b_k^1) \otimes (a_k^2, b_k^2) \cdots \otimes (a_k^{N/2}, b_k^{N/2}) \equiv |S_k\rangle \tag{7.2}$$

with (a_k^i, b_k^i) being site indices which only appear once in every basis state. k labels one of the $(N-1) \cdot (N-3) \cdots (N-N+1)$ possibilities to distribute the singlets. In case of non-frustrated lattices one can show [9] that only those singlet distributions k are necessary, in which a_k^i and b_k^i are on different sub-lattices for every i. This leads to a huge reduction in the size of the basis, since this way only $(N/2)!$ states contribute. But also in the bipartite case the basis is over-complete. So every state can be expanded in it, but the coefficients are not unique. This will lead to some trouble in interpreting measures related to Valence Bonds.

When picturing Valence Bonds, we think of them as arks connecting the 2 sites forming the singlet, as one can see in figure 7.1.

To implement an efficient algorithm the Hamiltonian of every bond in equation 2.1 gets shifted by a constant value of $\frac{1}{4}$:

$$H = \sum_{\langle i,j \rangle} J \left(\vec{S}_i \vec{S}_j - \frac{1}{4} \right), \tag{7.3}$$

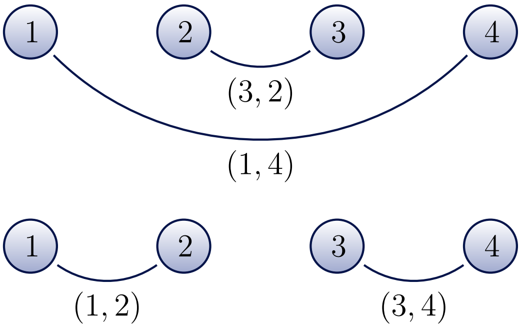

Figure 7.1: The 2 possible Valence Bond states of a 1 dimensional bipartite lattice with 4 sites. The sites are indicated by circles, and the valence bonds are drawn as arks connecting the 2 sites, which are in the singlet state. This means that the upper state is $|(1,4),(3,2)\rangle$, and the lower is in state $|(1,2),(3,4)\rangle$. The contribution to Valence Bond entanglement entropy between the regions A=(1,2) and B=(3,4) would be $S^{VB}=2$ in the upper and $S^{VB}=0$ in the lower state.

with $\vec{S}_i$ the vector of Pauli matrices. This leads to an easy action of bond-Hamiltonians onto a valence bond-state:

$$\begin{aligned} H_{ab}\,|\cdots(a,b)\cdots\rangle &= |\cdots(a,b)\cdots\rangle \\ H_{bc}\,|\cdots(a,b)\cdots(c,d)\cdots\rangle &= \frac{1}{2}\,|\cdots(a,d)\cdots(c,b)\cdots\rangle\,. \end{aligned} \tag{7.4}$$

Starting from a trial state in valence bond basis $|S_0\rangle$, one projects out the ground state by acting with high powers n of the Hamiltonian on it. To do this write H^n as $\sum_{P_n}\prod_{p=1}^{n} H_{a_p b_p}$ with $P_n = \Big((a_1,b_1),(a_2,b_2)\cdots(a_n,b_n)\Big)$ a bond index sequence. In the easiest way the Monte Carlo algorithm starts with any sequence of bond operators and attempts to change some (up to 4 in [9]). The new sequence gets accepted with metropolis probability. The weights for a sequence of bond operators can be found by using equation 7.4, and are just a power of $\frac{1}{2}$.

A few years later, a more efficient way of sampling the ground state of valence bond states was proposed by Sandvik and Evertz [10]. This algorithm is essentially the Loop Algorithm, with the Valence Bonds as boundaries instead of periodic boundary conditions. Hence it uses a combined spin- and Valence

Bond representation to make non local updates in the operator string.

7.1.1 Entanglement using the Valence Bond Basis

Now that we are able to sample valence bond configurations from the ground state, we can start thinking of entanglement measures. A singlet (a,b) (equation 7.1) is a maximal entangled state in a system of 2 qubits, and therefore can be regarded as a unit of entanglement. It is reasonable to suspect that the number of valence bonds crossing two regions have something to do with entanglement.

Let's consider a system consisting of 4 sites, and let us assume the ground state is a single valence bond state. We want to calculate the entanglement between region A with sites 1 and 2 and region B with sites 3 and 4. The 2 possible states in VB-basis are shown in figure 7.1.

If the ground state is $(1,2) \otimes (3,4)$, we see from the definition of separability that this state is not entangled. If the state is $(1,4) \otimes (3,2)$ [1] on the other hand, 2 singlets are shared between the 2 regions, and the entanglement is 2 qubits or $2 \cdot \log 2$.

Valence Bond Entanglement Entropy

We saw that if the ground state consists of a single Valence Bond-state, the number of them crossing the border between 2 regions is the same as the von Neumann entropy of this state. This led to the definition of the so called Valence Bond entanglement Entropy (SVB) by Alet et al. [2] and by Chhajlany et al. [48]. It is defined as:

$$S^{VB} := \log 2 \sum_i a_i n^c_\Omega \left(|S_i\rangle\right) \frac{1}{\sum_i a_i}, \tag{7.5}$$

[1] The order in which the sites appear in this notation is not random. To get a positive weight from equation 7.4 one needs to stick to the rule that the first index of a valence bond is from one sub lattice and the second from the other [9].

if the ground state is decomposed in the Valence Bond Basis $|GS\rangle = \sum_i a_i |S_i\rangle$. $n^c_\Omega (|S_i\rangle)$ is the number of Valence Bonds crossing the border of region Ω in state $|S_i\rangle$. This entropy may seem ill defined, in the sense of not looking like a quantum mechanical expectation value. Since the basis is over-complete, the coefficients a_i are not unique. But as it turns out, for any state $|\psi\rangle$, the number of valence bonds $n_{ij}(|\psi\rangle)$ connecting site i and site j can be written as [49]:

$$n_{ij}(|\psi\rangle) = \frac{\langle R_s| S_i^+ S_j^- |\psi\rangle}{\langle R_s|\psi\rangle}, \tag{7.6}$$

with the reference state $|R_s\rangle = |1, 0, 1, 0 \cdots\rangle$ in the S^z-basis. Note that $n_{ij}(|\psi\rangle)$ is not the same quantity as n^c_Ω in equation 7.5. By summing over all i from Ω, and over all j from its complement $\bar{\Omega}$, one obtains the Valence Bond entanglement entropy [49]:

$$S^{VB} = \log 2 \sum_{(i,j) \text{ such that } i\in\Omega, j\in\bar{\Omega}} n_{ij}(|\psi\rangle). \tag{7.7}$$

This formula shows that SVB only depends on the choice of bipartition in the reference state. Furthermore, it enables one to calculate SVB not only in Monte Carlo algorithms with a bipartite lattice, but also for frustrated lattices [49], or for any method yielding the ground state.
At first it was believed that the Valence Bond entropy scales like the von Neumann entropy, as predicted by CFT (equation 4.3). As it turned out, the pre-factor of the logarithmic divergence of S^{VB} is not $\frac{1}{3}$, but $\frac{4 \log 2}{\pi^2} = 0.281$ [50]. Furthermore S^{VB} can either be greater or smaller than the von Neumann entropy, so it also cannot be an upper or lower bound to the latter [18].

Second Valence Bond Entropy and Loop Entropy

Later, two different ways of measuring entanglement using Valence Bond basis states were introduced by Lin and Sandvik [3]. They are called S_2^{VB} and S_{loop} and rely on the overlap of 2 Valence Bond states. Hence, both measures need a double projection to be able to calculate this overlap. In a double projection not one trial state gets projected onto the ground state, but two. Therefore, one needs two operator sequences and update them independently, so that in every step one ends up with 2 independent representations of the ground state.

The overlap of 2 Valence Bond states is given by:

$$\langle S_1|S_2\rangle = 2^{N_0-N/2}, \tag{7.8}$$

where N_0 is the number of loops in the so called transition graph of the two Valence Bond states $|S_1\rangle$ and $|S_2\rangle$. To obtain this graph, again think of valence bonds as arks connecting 2 sites. Starting from site i follow it to the site of the valence bond in $|S_1\rangle$, call this site j. Now go from j to its VB-neighbor in the state $|S_2\rangle$ which we name k. From k go to l which is the Valence Bond in state $|S_1\rangle$ and so on, until one arrives at i again. This leads to closed loops and the set of all these loops is called the transition graph of the two basis states $|S_1\rangle$ and $|S_2\rangle$. For examples of transition graphs take a look at figure 7.2.

The first of the new measures, called S_2^{VB}, is just a different weighting of the single valence bond states obtained in the projection algorithm. Every VB state gets weighted with the inverse of the overlap $2^{-N_0+N/2}$ from equation 7.8. Hence S_2^{VB} is similarly defined as S^{VB}:

$$S_2^{VB} := \log 2 \sum_i a_i \left[\frac{n_\Omega^c\left(|S_i\rangle\right)}{\langle S_i|S_i'\rangle}\right] \frac{1}{\sum_i a_i}. \tag{7.9}$$

It scales the same as S^{VB}, but seems to converge faster to this scaling behavior [3].

The second measure, presented by Lin and Sandvik, is the so called loop entropy. Instead of counting the shared valence bonds in either state $|S_i\rangle$ or $|S_i'\rangle$, one counts the number of loops in the transition graph:

$$S_{loop} := \langle \Lambda_{AB} \rangle. \tag{7.10}$$

with Λ_{AB} counting the number of loops in the transition graph. An example

of 2 transition graphs can be found in figure 7.2. As to the scaling of the Loop entropy: in 1 dimensional system the logarithmic pre-factor is $\frac{\log 2}{3}$, so even worse as the Valence Bond entropies, but for 2 dimensions it obeys an area law, and mimics the scaling of the von Neumann entropy very well.
As already mentioned the Valence Bond entropy can be written as a quantum mechanical expectation value, depending only on the bi partition of the lattice. For the Loop entropy this is no longer true. As it was shown by Capponi et al. for a simple 6 site lattice the same state can have 2 different values of S_{loop} due to the over-completeness of the VB-basis [19]. This shows that its physical meaning is questionable, but it can still be a useful tool in investigating scaling behavior of the von Neumann entropy

To close this section I give a short overview. First the Valence Bond basis was introduced and 3 entanglement measures derived from this basis were discussed, see equations 7.5, 7.9 and 7.10. The reason to introduce them was not only to summarize the work done so far in this field, but as it turns out the Valence Bonds basis is a natural basis for the Loop Algorithm, and therefore also for the one sampling the RDM. As we will see in the next chapter (chapter 8) one is able to sample Valence Bond states from the ground state with the DMLA. Not only that, we will also see that the modified DMLA will produce 2 separate representations, giving access to the overlap of two VB-states. Hence it is possible to calculate all 3 entropies introduced in this section with only minor adjustments to the algorithm.

7.2 Cluster Entropies

In the previous chapter we saw, how to sample RDMs using a modified version of the Loop Algorithm. Since that way the whole matrix of the RDM needs to be stored, the size of the system described by the RDM is very limited. As mentioned earlier, one is very often interested in the scaling of the entropy with the size of the subsystem. This means, one would like to calculate it for very big subsystems. With the algorithm presented so far,

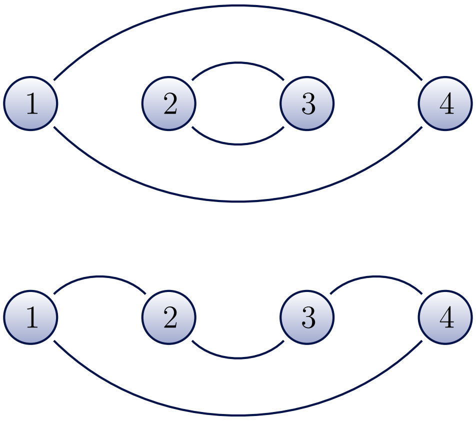

Figure 7.2: Two of the 4 possible transition graphs of a 1 dimensional bipartite lattice with 4 sites. The states in the upper picture have two closed loops, so their overlap is $2^{2-2} = 1$. The states in the lower picture on the other hand have one loop in the transition graph and therefore an overlap of $2^{1-2} = \frac{1}{2}$. For the loop-entanglement between the 2 regions $A = (1, 2)$ and $B = (3, 4)$ this means that the upper configuration contributes $\Lambda_{AB} = \frac{2}{1} = 2$ and the lower one $\Lambda_{AB} = \frac{1}{0.5} = 2$.

this simply is not possible.

Inspired by the Valence Bond entropies, some new ways of counting properties similar to Valence Bonds are presented in this section. In chapter 4 we saw that the entanglement in the ground state, between two regions A and B, is a property of the Reduced Density Matrix. This means that it must also be a property of the Path Integral, from which the DMLA samples RDM states. Therefore, the entanglement must somehow be encoded in the loop structure obtained in the DMLA.

In the next chapter we will see that the loops from the DMLA become the Valence Bonds, when introducing cuts at all sites. This means that loops are somehow related to the Valence Bonds, and if not all sites are cut open, the loops of the DMLA can be viewed as a generalization of Valence Bonds. This led to the idea that when sampling the RDM (i.e.: cuts only appear on those sites belonging to region A), the loop structure can, or should, be related to entanglement like the Valence Bond entropies should.

To clarify the idea take a look at figure 7.3. In this picture one can see a single cut of the Path Integral, and a single loop starting from below this cut. In this arrangement the RDM (region A) consists of only one site, while sites 2-5 are in region B. The dashed line signals the time at which the cuts are introduced, and red lines mean spin up while black lines spin down. Note that every time the loop changes its direction also the spin changes. Since loops have 2 possible spin orientations, figure 7.3 means a superposition of the 2 possible states: $|10101\rangle + |01010\rangle$. If this would be the exact ground state, there would be one unit of entanglement between A and B.

So the idea is, that similar to the Valence bonds, one counts the number of loops connecting regions A and B to get information about the entanglement. There are a certain number of ways one can do the counting, and I will present the most reasonable ones in this work. Before we even start to look at any results, we can already make out some problems.

First: since cutting open the path integral leads to a different treatment of regions A and B, all the measures will, most likely, not be symmetric anymore, whereas the von Neumann entropy of the ground state is!

Second: By cutting open imaginary time at some sites, we effectively sample from the RDM. This means that also the loop structure obtained by doing so depends on the RDM not on the total state. But since thermal entanglement is a property of the total state, the connection of the Cluster entropies to thermal entanglement is questionable.

Third: The von Neumann entropy is a proper quantum mechanical expectation value, but it is not clear if these measures are. It is the same problem that S_2^{VB} (equation 7.9) faces. We can only hope that it behaves similarly as the von Neumann entropy.

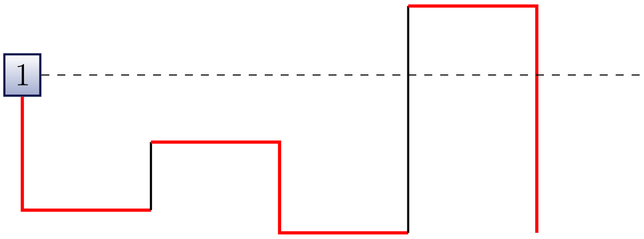

Figure 7.3: Explanation of the relation between the loops in the Loop Algorithm and entanglement. Red lines are worldlines and black lines signal spin down.

The general procedure to measure this loop properties is: Introduce cuts at all sites belonging to region A. This is the same step as one would take when sampling the RDM of A. Follow all loops starting from below and/or above the cut, and look at which sites on region B they appear. Depending on the way of counting, add some number (typically 1 or log(2)) to the proposed measure. Here, I introduce 4 new measures called Cluster entanglement 0-3. In figures 7.4 and 7.5 two examples of loop configurations are shown, and their contribution to every Cluster entanglement measure is given. The 4 measures are:

- **Cluster Entanglement 0** (CE_0) :
  Counts the number of sites with a cut (therefore belonging to sub lattice A), which are connected by a loop to a site from sub-lattice B at imaginary time $\frac{\beta}{2}$. One counts only below the cuts.

- **Cluster Entanglement 1** (CE_1):
  Counts the number of loops starting at a cut (below and above!) and moving through $\frac{\beta}{2}$ on sub-lattice B. Cluster entanglement 1 is this number divided by 2, and it turns out that it coincides (on average) with Cluster entanglement 0, as will be explained later.

- **Cluster Entanglement 2** (CE_2):
  Counts the number of loops starting at a cut and arriving at a place on sub-lattice B. Contrary to CE0 and CE1 it doesn't matter at which imaginary time the loop moves to B. Cluster entanglement 2 is this number divided by 2.

- **Cluster Entanglement 3** (CE_3):
  Counts the number of loops connecting a cut to a point at sub lattice B at imaginary time $\frac{\beta}{2}$. Cluster entanglement 3 is this number divided by 2.

Note that in all measures except CE_0 count below and above the cut.

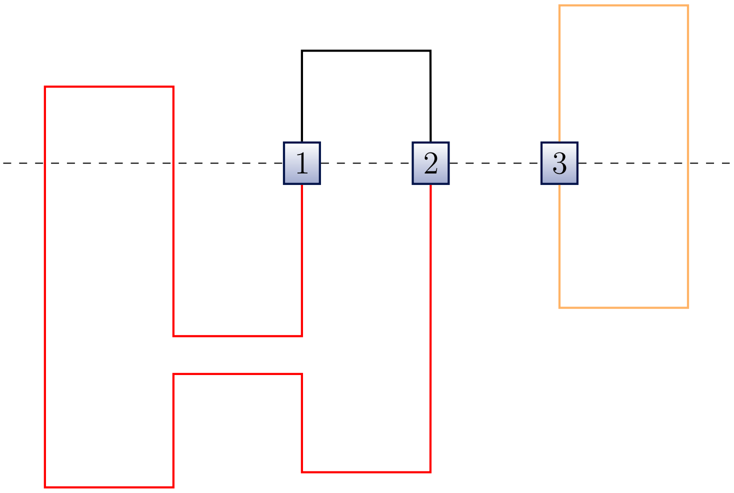

Figure 7.4: A loop configuration with 3 loops and 3 sites having a cut indicated by the numbered square boxes. The dashed line shows the imaginary time at which the cuts are introduced.
The red loop has a contribution of 2 to CE_0, of 1 to CE_1, of 1 to CE_2, and of 1 to CE_3.
The black loop on the other hand has no contribution to any of the 4 measures, because it moves only through sites with a cut.
Finally the orange loop has a contribution of 1 to CE_0, of 1 to CE_1, of 1 to CE_2, and of 1 to CE_3

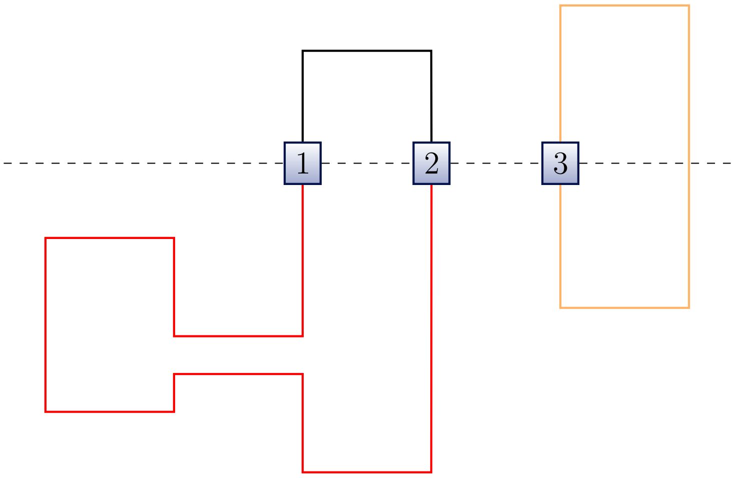

Figure 7.5: A loop configuration with 3 loops and 3 sites having a cut indicated by the numbered square boxes. The dashed line shows the imaginary time at which the cuts are introduced.
The red loop has a contribution of 0 to CE_0, of 0 to CE_1, of 1 to CE_2, and of 0 to CE_3.
The black loop on the other hand has no contribution to any of the 4 measures, because it moves only through sites with a cut.
Finally the orange loop has a contribution of 1 to CE_0, of 1 to CE_1, of 1 to CE_2, and of 1 to CE_3

Chapter 8

Sampling of Valence Bond States

In chapter 7, three ways of approximating the von Neumann entropy were presented, which all use the Valence Bond basis. With the DMLA shown in chapter 6 one can easily produce Valence Bond states sampled from the ground state, as in the usual Projection Algorithms. To do this, cuts need to be introduced at all sites of the lattice. Again the cuts are introduced at $\frac{\beta}{2}$ so they are in the middle of the imaginary time axis.

Doing this yields spin states sampled from the full density matrix ρ. If β is high enough, the density matrix approaches the ground state i.e.: $\rho = |GS\rangle \langle GS|$. In that case, the obtained states below and above the cuts are separate representations of the ground state. If we are able to somehow construct Valence Bond from the these states, we get essentially the double projection method. We then would be able to calculate the overlap of the two representation of the groundstate, and hence, have access to all 3 Valence Bond entanglement measures.

But how does one get Valence Bonds, when there are only spin states and no singlets? In chapter 5.2 the Multi Loop Algorithm was shown to consist of switching between 2 representations, the normal spin representation and the bond operator representation. It is true that a single spin configuration is no singlet, but in the intermediate step, when the configuration consists only of the bond operators, we do have singlets. This is, because the the bond operator is a singlet projection operator, what means that it projects onto the singlet subspace and the resulting state is in a singlet.

What does this mean for the DMLA? If one looks at the loops that can be found there, one sees some consisting of closed loops, and others starting and ending in a cut. The former are the normal loops appearing in the traditional Loop Algorithm, while the latter can only be found in the DMLA. If β is high enough, there are no loops connecting spins below the cut to spins above the cut (one needs to make sure of that). With other words, all loops starting below (above) a cut also end in a cut coming from below (above) respectively [1].

Putting all those pieces together one can say that if a loop connects site i and j below the cut, they are in a singlet state i.e.:

$$(i, j) = \frac{1}{\sqrt{2}} \left(|10\rangle - |01\rangle \right) .$$

The set of all these loops result in nothing but one of the Valence Bond basis states. Figure 8.1 shows an example how to get from a loop configuration to Valence Bond states. Since the same is true for all sites above the cut, one gets 2 Valence Bond representations of the ground state and effectively has a double projection method.

In case one wonders about the periodic boundary conditions in imaginary time: Since high powers of the Hamiltonian project *every* state onto the ground state (as long as its overlap with the ground state is not zero), it does not matter with which state we start the projection. Hence, we are allowed to start both projections with the same state. This is equivalent to periodic boundary conditions in imaginary time.

[1] Note that in that case loops only connect cuts belonging to different sub-lattices since they need to make an odd number of direction-changes, which implies that the cuts are an odd number of sites apart. This is another formulation of the argument that only those Valence Bonds contribute, which connect sites on different sub-lattices.

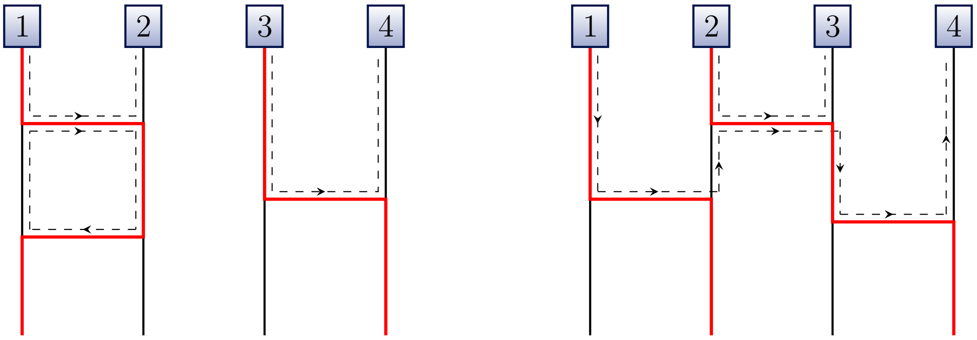

Figure 8.1: Two examples of the worldline structure of a 1 dimensional lattice with 4 sites. The rectangles signal the cuts. In this picture the red lines are not necessarily worldlines anymore, but only show a point where the loop changes direction. They need not be points where the spin changes. The dashed lines show, which regions of the lattice belong to which loop. Note that there are loops consisting of closed *loops* and others that start at a cut and end at one. As mentioned in the text, they are constructed in a way that one can flip all spins belonging to a loop. Note that in this figure only the worldlines below the cuts are shown. Above the cuts one gets different loops and therefore different Valence Bonds and a VB overlap. Hence, one can calculate the transition graph i.e.: the overlap of 2 Valence Bond states.
In the left picture a loop connects sites 1 and 2, and another loop sites 3 and 4. In the spin representation the state of site 1 and 2 is $|10\rangle_{12}$. The state when one flips this loop would be $|01\rangle_{12}$. This means that sites 1 and 2 are in a singlet. The corresponding Valence Bond state therefore is $|(1,2),(3,4)\rangle$. This is the same Valence Bond configuration as the bottom configuration in figure 7.1.
In the right picture one loop connects sites 1 and 4 and the other loop connects sites 3 and 2. The corresponding Valence Bond state therefore is $|(1,4),(3,2)\rangle$. This is the same Valence Bond configuration as the top configuration in figure 7.1.

Chapter 9

Emulation of the Monte Carlo Simulation

Before I started writing the actual algorithm, I wanted to get an idea about the potential of this method. Therefore, I decided to *emulate* the Monte Carlo Method. At temperature $T = 0 \Leftrightarrow \beta \to \infty$ the system is in a single state (the ground-state), expanded in the S^z-basis: $|\psi_{GS}\rangle = \sum_{|i\rangle \in \mathcal{A},} \sum_{|j\rangle \in \mathcal{B}} c_{ij} |i\rangle \otimes |j\rangle$ with the density matrix:

$$\rho_{GS} = |\psi_{GS}\rangle \langle\psi_{GS}| = \sum_{|i\rangle,|n\rangle \in \mathcal{A},} \sum_{|j\rangle,|m\rangle \in \mathcal{B}} c_{ij} c^*_{nm} |i\rangle \otimes |j\rangle \langle n| \otimes \langle m| \,. \tag{9.1}$$

In equation 9.1 the lattice is already split into 2 regions A and B, with $\mathcal{A}$ ($\mathcal{B}$) the set of states in the S^z-basis belonging to A (B) respectively. The number of sites in A is $|A|$, and in B are $N - |A|$ sites, so $dim(\mathcal{A}) = 2^{|A|}$ and $dim(\mathcal{B}) = 2^{N-|A|}$. Again the reduced density matrix (RDM) is obtained by tracing out the degrees of freedom belonging to region B:

$$\rho_{A,GS} = tr_B \Big(\rho_{GS} \Big) = \sum_{|l\rangle \in \mathcal{B}} \langle l| \rho_{GS} |l\rangle = \sum_{|i\rangle,|n\rangle \in \mathcal{A},} \underbrace{\left(\sum_{l=0}^{dim(B)} c_{il} c^*_{nl} \right)}_{(\rho_A)_{in}} |i\rangle \langle n| \tag{9.2}$$

Since the ground state of the Heisenberg Anti-Ferromagnet is real, one can drop the complex conjugation. The DMLA-Method produces matrix elements $|i\rangle \langle n|$ of the RDM according to its weight $|(\rho_A)_{in}| = |\sum_l c_{il}c_{nl}|$. If one knows the exact ground state, one can choose randomly with unnormalized probability $(\rho_A)_{in}$ states, hence sampling from the RDM. This emulates the Monte Carlo method, and the effort to write a program doing this is much less than to write a Loop Algorithm. With the RDM we can take a look at some measure derived from it, which will be the von Neumann entropy.
In our case we used the Lanzcos method to calculate the ground state of a $L = 20$, 1d lattice. The coupling constant J was set to 1. We then sampled different numbers of states according their weight in the RDM. This was essentially done by producing a vector $|v\rangle$:

$$|v\rangle = |(\rho_A)_{in}| \cdot |S_1, ..., S_L\rangle \otimes |S_1, ...S_{|A|}\rangle . \tag{9.3}$$

Note that $|v\rangle$ is not a state vector in the usual sense. By searching a random number, using bisection, in the cumulated sum of $|v\rangle$ we obtained one element of the sample of the RDM. Repeating this a certain number of times one gets the whole sample, in which a certain state of the S^z-basis can be in it more than once. In a next step we calculated the von Neumann entropy of the sampled RDM:

$$S(\rho_A) = tr\bigg(\rho_A \log \rho_A\bigg). \tag{9.4}$$

For every sample size and number of sites in the RDM this was done 10 times to get an average von Neumann entropy. No errors were calculated, and there were no checks whether the sample is converged, since in this part of the work I was only interested in getting a rough idea about how well this method will work. Finally the entropy of the sampled RDM was compared with the exact one. The latter is easily calculated once one knows the ground state. The result of this can be seen in figure 9.1.

Since every state sampled is equivalent to an independent Monte Carlo configuration, the result looks promising. The error in the entropy is mostly

[2] Although not all elements are non-zero.

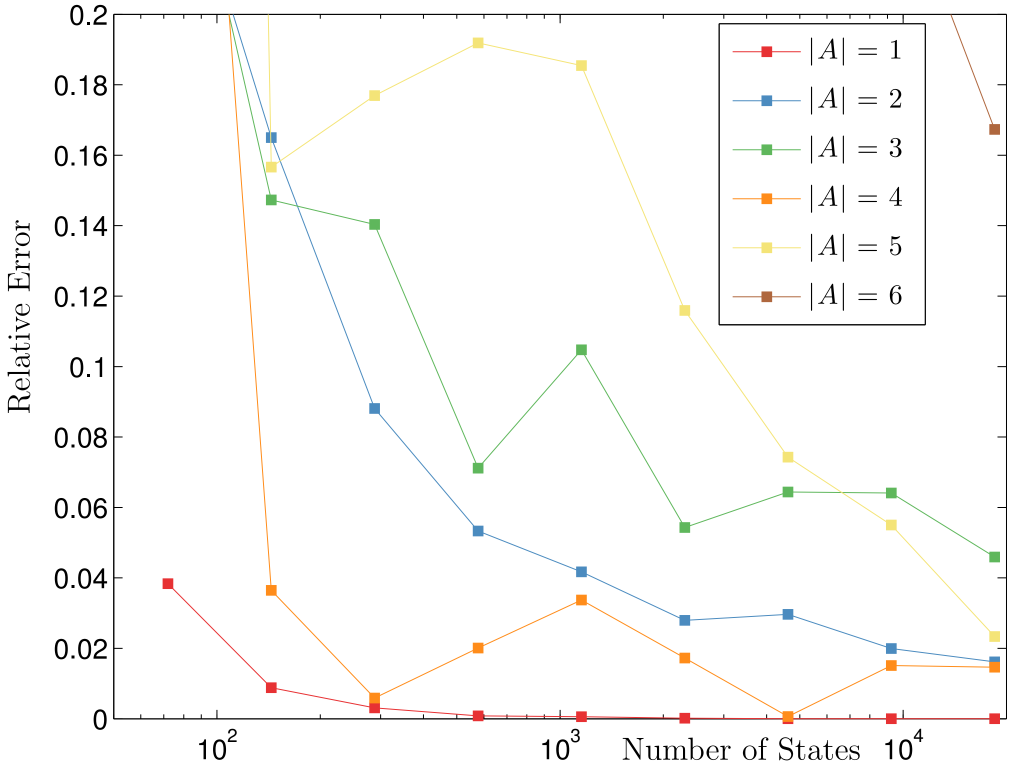

Figure 9.1: Number of states in one sample versus the relative error $\frac{|S_{A,ex}-S_{A,sample}|}{|S_{A,ex}|}$ of the sampled RDM. $S_{A,ex}$ is the exact entropy while $S_{A,sample}$ is the entropy obtained from the sampled RDM. The number of basis-states in each sample was 72, 144, 288, 577, 1154, 2309, 4618, 9237 and 18475. Compared to that, the number of basis-states is $2^{20} \sim 10^6$. Generally the more basis states one takes, the better the convergence, as it is expected. Note that on the top-right the graph from $|A| = 6$ enters the graphic. For bigger sample sizes all errors except $|A| = 6$ are smaller than 10% and seem to converge. This is encouraging, since the number of elements in the RDM of say $|A| = 5$ is $32 \times 32 = 1024^2$and compared to that 18000 is not very much. Moreover every sample element corresponds to a uncorrelated Monte Carlo measurement, and 18000 measurements taken in a MC-run is very little. I didn't find an explanation to why the errors for $|A| = 4$ are smaller than those of $|A| = 2$, but running the simulation several times resulted always in essentially the same behavior. Overall it looks promising that the DMLA will work.

under 10%, if 10^4 S^z-basis states are in one sample.

Chapter 10

Sampling Reduced Density Matrices - Results

In this section I will show and discuss the results obtained by sampling reduced density matrices via the DMLA, as explained in section 6. To compare reduced density matrices from the DMLA with exact ones, I will take the von Neumann entropy as a measure of choice. Since the Loop Algorithm also works for finite temperature, I will show von Neumann entropies not only for the ground state, but also for lower values of β. Note that in those cases the entropy is not a measure of entanglement, which does not mean it cannot be used to compare RDMs.

When sampling RDMs with Monte Carlo, the result is usually not symmetric, so one has to symmetrize it manually. Furthermore, it turns out that when diagonalizing RDMs of big subsystems one often gets small negative eigenvalues. Because of this one has to introduce a cut-off for the eigenvalues at some small, positive value.

10.1 Von Neumann Entropies and Biases

First take a look at figure 10.1, which shows the exact and Monte Carlo von Neumann entropy for various temperatures and different number of sites in the RDM. Since this figure includes error-bars calculated with jackknife, it is obvious that something is very wrong. The deviation from the exact value is far bigger than the statistical errors, for some data points. For small reduced systems up to 3 sites, the values look very good for all temperature ranges. At 4 sites in the RDM something starts to happen that yields too big entropies. One can also see that the deviation gets bigger, as the temperature approaches zero.

This result opens a few questions:
Why is it harder to sample RDMs from the ground state, as it is to get them for higher temperatures?
Why is the Monte Carlo entropy always bigger than the exact one - naively one would assume that one ends up with smaller and bigger values, like typical sample results?
These two questions will be answered in the following pages of this section. Since it is clear that this algorithm converges to the exact RDM, the problem has to be that we simply didn't use enough data. Figure 10.1 was produced using 10^7 sweeps. The RDM for say 5 sites has 1024 elements. This means that every element would occur 10^4 times, assuming that they all have the same probability of occurring. This is not very much, but it normally should not lead to deviations that big.
When dealing with statistical outliers or biased data, a useful measure is the jackknife bias b, defined by:

$$b := (N_b - 1) \cdot (S_{jack} - S); \tag{10.1}$$

with, N_b the number of bins, S_{jack} the average of the entropy by leaving out data from one of the bins, and S the entropy using all the data. If this bias is bigger than the statistical error, there is a problem with the data, or it has

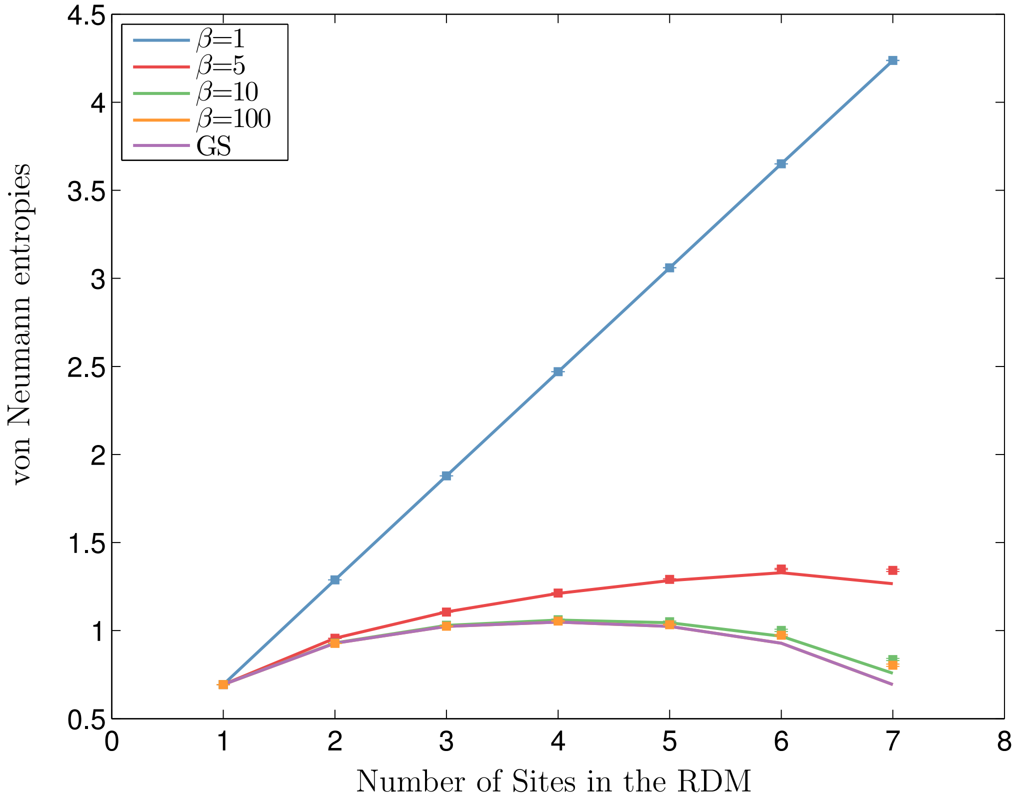

Figure 10.1: Von Neumann entropies of a L=8 lattice, for different number of sites in the RDM, and for various inverse temperatures β. The line shows the results obtained by full diagonalization, while the squares are Monte Carlo values. The coupling constant J was set to 1 in this, and for all following calculations. The eigenvalue cutoff was 10^{-4}. To produce this result, 10^7 sweeps were made. One sweep consists of as many loops as it takes to flip an area as big as the whole space-time lattice (see section A.2). The purple line, labeled GS, shows the von Neumann entropy of the ground state by using $\rho_{A,GS} = tr_B \left|\psi_{GS}\right\rangle \left\langle\psi_{GS}\right|$. These entropies completely overlap with those from $\beta = 100$, what explains, why the yellow line cannot be seen. Note that this figure includes error-bars, but they are very small - much smaller than the deviations from the exact result. The errors were calculated using jackknife.

some inherent bias. So take a look at figure 10.2, which compares the statistical error to the bias. Like we expect from figure 10.1 starting at 4 sites in the RDM the bias gets bigger than the statistical error and, it stays that way.

So do we have statistical outliers ruining the entropies? The answer is: No! When looking at figure 10.3, one can see a rather surprising result. It shows the individual jackknife bins, the jackknife entropy and the entropy when including all data. There is no bin, which has any odd looking deviation from all the other bins. It rather seems that removing data yields, on average, higher entropy values, with other words a real bias. So we conclude that increasing data will lead to smaller entropies, which are very much desired when looking at figure 10.1.

Spectrum

To further understand why more measurements yield lower entropies, we need to go into more detail of the data. Since the von Neumann entropy is calculated from the eigenvalues of the RDM, it is obvious to take a look at them. Figure 10.4 compares the exact eigenvalues with those stemming from the Monte Carlo data. It shows the case of a L=8 lattice with 6 sites in the RDM. In this case the RDM can only have 4 non-zero eigenvalues. This is a implication of the Schmidt decomposition of the ground state. It states that the N-site ground state RDM of a lattice with L sites has the same spectrum as the (L-N)-site RDM (neglecting all the zeros in the bigger RDM). At first glance the spectrum actually looks pretty good, but the small eigenvalues are to big, which makes the bigger eigenvalues to small (since the trace of the density matrix must be one).

This result helps us understand what happens: In order to produce many eigenvalues that are (very close to) zero, the individual matrix elements must be very fine tuned to each other. Small statistical errors in them are enough

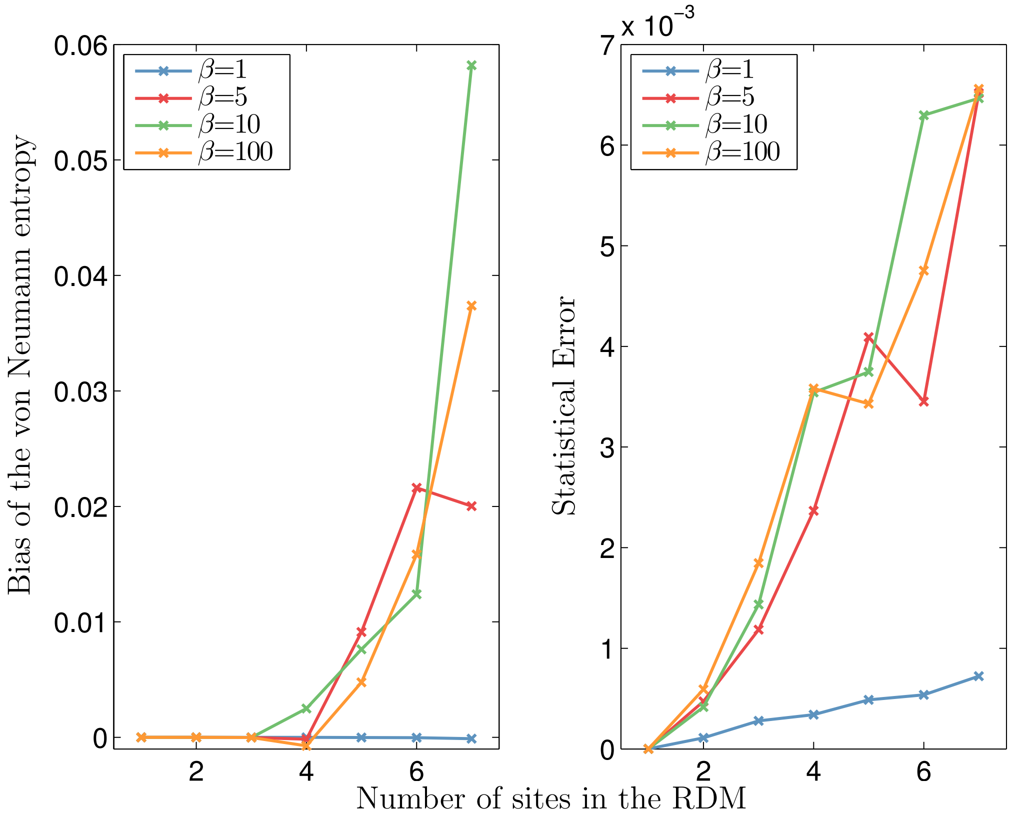

Figure 10.2: *Left:* bias of the von Neumann entropy calculated with equation 10.1. *Right:* The statistical error of the von Neumann entropy calculated using jackknife.

These plots are from the same data that was used to calculate the entropies in figure 10.1. Starting at about 4 sites in the RDM (the x-axis), the bias becomes about the size of the error. Again one can observe that for $\beta = 1$ the bias stays essentially zero. This means that these entropies are trustworthy, and shows that the problem becomes more severe as the temperature approaches zero. Note that the values for $\beta = 100$ are converged to the ground state.

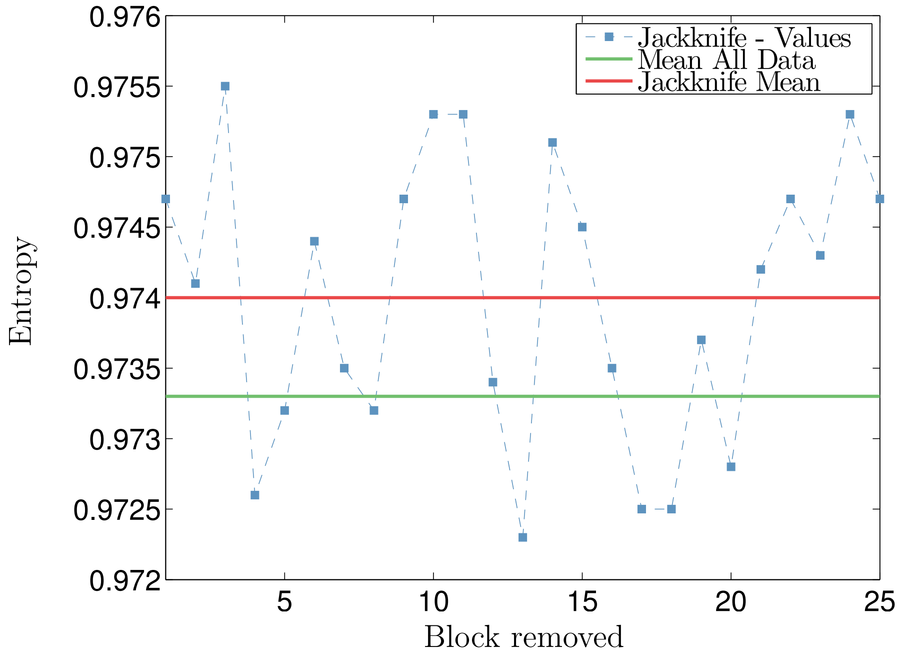

Figure 10.3: The blue squares show 25 entropies from leaving out one of the 25 blocks of data in jackknife. The red line is the mean of these 25 entropies, and the green line is the value of the entropy from using all data. These values are produced from the data of the 6-site RDM at the $\beta = 100$ in figures 10.1 and 10.2. This picture shows that there are no statistical outliers in the data, but it seems that less statistics leads to an overestimation of the entropy. Also from this picture one can understand the size of the jackknife bias, which is proportional to the distance between the red and the green line.

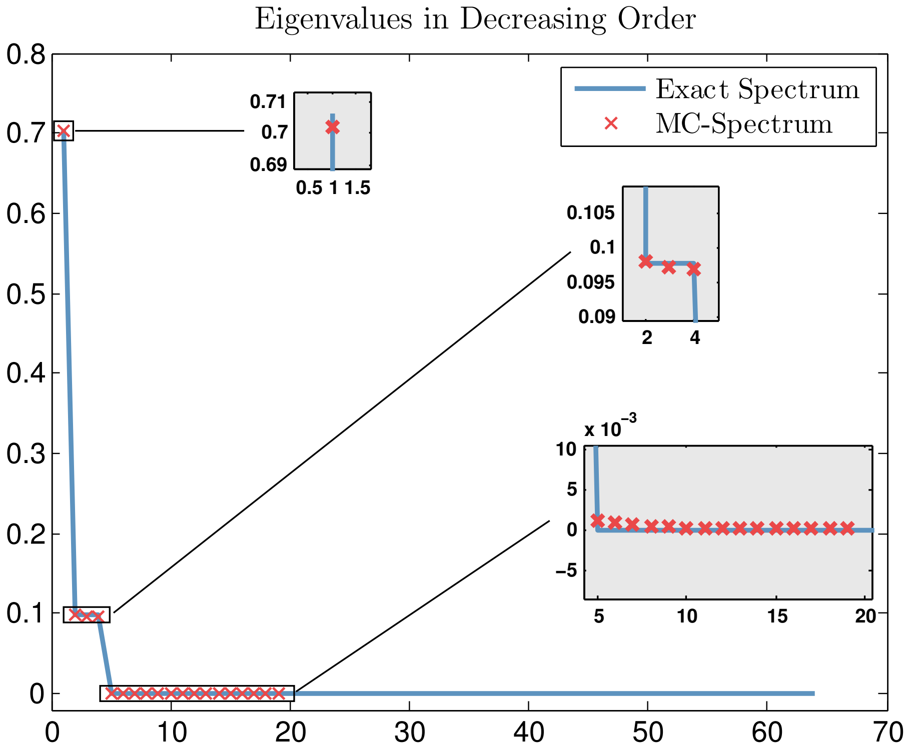

Figure 10.4: Eigenvalues of the 6 site RDM of the L=8 lattice at temperature $\beta = 100$. The blue line shows the eigenvalues from full diagonalization, while the red crosses are the Monte Carlo eigenvalues. Clearly the overall structure of the eigenvalues is pretty well captured, but when zooming in one can see that the big eigenvalues are too small, while the small eigenvalues (they should be zero in this case) are too big. When analyzing the error of the small eigenvalues one sees essentially the same picture as for the entropies, namely the more data is used, the smaller they get and they have a rather big bias compared to their error. The size of the statistical errors in this plot is about the height of the red crosses. Again this plot stems from the same data that was shown in figure 10.1.

to make some of the small eigenvalues too big, which in turn makes bigger eigenvalues too small. This behavior is not only the case for the shown set of parameters (L=8, 6 sites in RDM and $\beta = 80$), but can be found in all other system sizes and different number of sites in the RDM.
The root of the problem lies in the function $f(x) = -x \log x$. For $x < 0.368$, $f(x) > x$, what makes contributions of the small eigenvalues still quite big. In fact the deviations of all eigenvalues (also from the small ones) have about the same contribution to the deviation of the entropy.
This can also explain, why for higher temperatures the problem becomes smaller. When β approaches zero every state is occupied with the same probability, so in its eigenbasis the RDM is the identity operator (of course it has to be normalized). Then it is possible that statistical errors in the matrix elements do not translate as badly in errors of the entropy, as when there are many zeros in the diagonal.

At this point on might ask, if the problem with the big number of nearly zero eigenvalues vanishes, if the total system becomes larger. To answer this let us move away from the L=8 lattice, and take a look at the L=12 lattice. The number of sites in the RDM stays at 6 sites. This means that $L - |A|$ is also 6, so it is not possible to say, before doing any calculations, that there will be a certain number of zero eigenvalues. We already saw that the computational effort is quite high, and that we will need much more than 10^7 sweeps to see convergence of the entropy. Therefore I decided to reduce the inverse temperature to $\beta = 80$. The convergence to the ground state at this temperature was checked, by looking at the the sum over all elements of $|\rho_{\beta\to\infty} - \rho_\beta|$, by looking at the ground state energy and by looking at the spectrum of the 6 site RDM. Since $L = 12$ full diagonalization is an option, yielding ground state and thermal density matrices. All cases showed convergence in β.

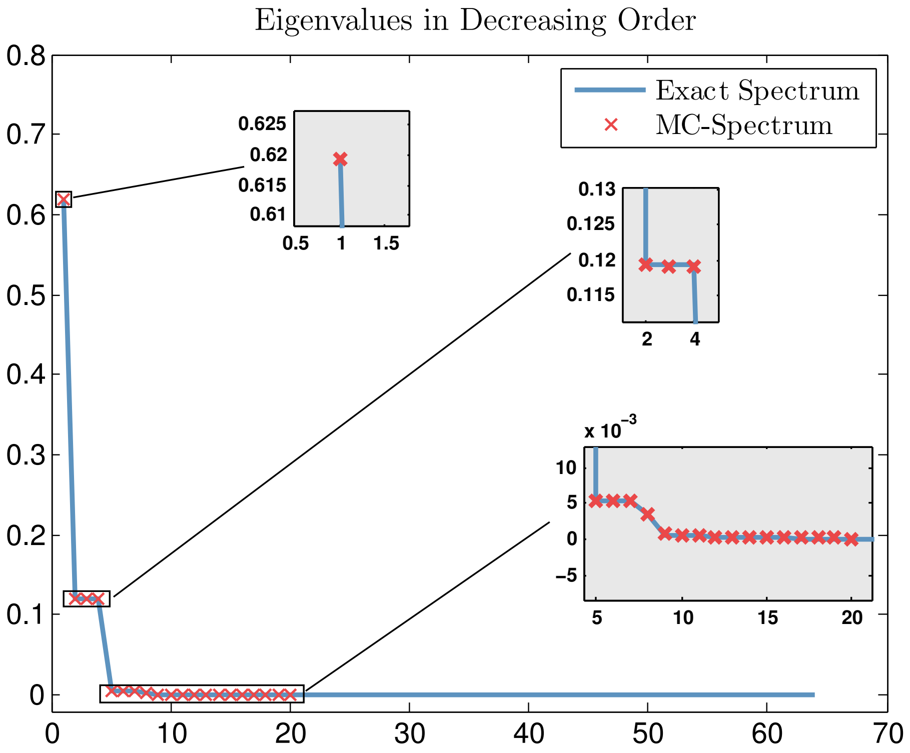

Figure 10.5: Eigenvalues of the 6 site RDM of the L=12 lattice at temperature $\beta = 80$. Compared to all the previous data, the number of sweeps was increased to 10^8. Again the blue line shows the eigenvalues from full diagonalization, while the red crosses are the Monte Carlo eigenvalues. Its a bit of a surprise to see that this spectrum of the RDM looks very much like the one of the L=8 lattice, except that there are some eigenvalues that are small but non-zero. Increasing the data by a factor of 10 leads to a visible improvement of the eigenvalues. But even here the difference of the exact- to the Monte Carlo entropy is $6.5 \cdot 10^{-3}$, while the statistical error is $2.6 \cdot 10^{-3}$ and the bias is $11.4 \cdot 10^{-3}$. So although the spectrum already looks good and the difference is very small, the data is still not converged, and the entropy is still too big.

Figure 10.5 shows the spectrum a L=12 lattice, with 6 sites in the RDM. In this plot the data was increased by a factor of 10, resulting in a better spectrum. But still it is NOT converged, as is explained in the caption of the figure.

Increasing Data and Artificial Noise

So how much data is needed in order to get reliable results? Partly, this question gets answered in figure 10.6, which compares the MC-entropy to the exact one, for an increasing number of sweeps. It shows that the entropy approaches the exact value, but seems to converge to a bigger value than the exact result. This is not very troubling, since the bias is still much bigger than the statistical error, what means that the result is not yet converged. Nevertheless we can already see that correcting for the bias would result in acceptable results.

To make sure that the behavior we have seen so far really comes from statistical fluctuations of the RDM matrix elements, let us take a look at a RDM with artificial noise. For this purpose a normally distributed random number with mean 0 and noise σ is added to the 6 site RDM with L=12 and $\beta = 80$[1]. The statistical error of a sample of size N scales as $\frac{1}{\sqrt{N}}$, what means that given some number of sweeps (data points) a rough estimators for the statistical error i.e.: the noise σ can be made. When doing this, we expect a similar behavior of the entropy of the noisy RDM, as the Monte Carlo entropies show in figure 10.6. Especially, the exponential convergence to the exact value should roughly be the same. The result of this can be seen in figure 10.7. Indeed it shows the same overall behavior as the Monte Carlo data does, namely noisy entropies are bigger than the exact ones and they always have a positive bias. This shows that we really have a literal bias and that we can correct for it to get better estimators for the von Neumann entropy (see also figure 10.13).

[1] This is the same set of parameters used in figure 10.5 and in figure 10.6.

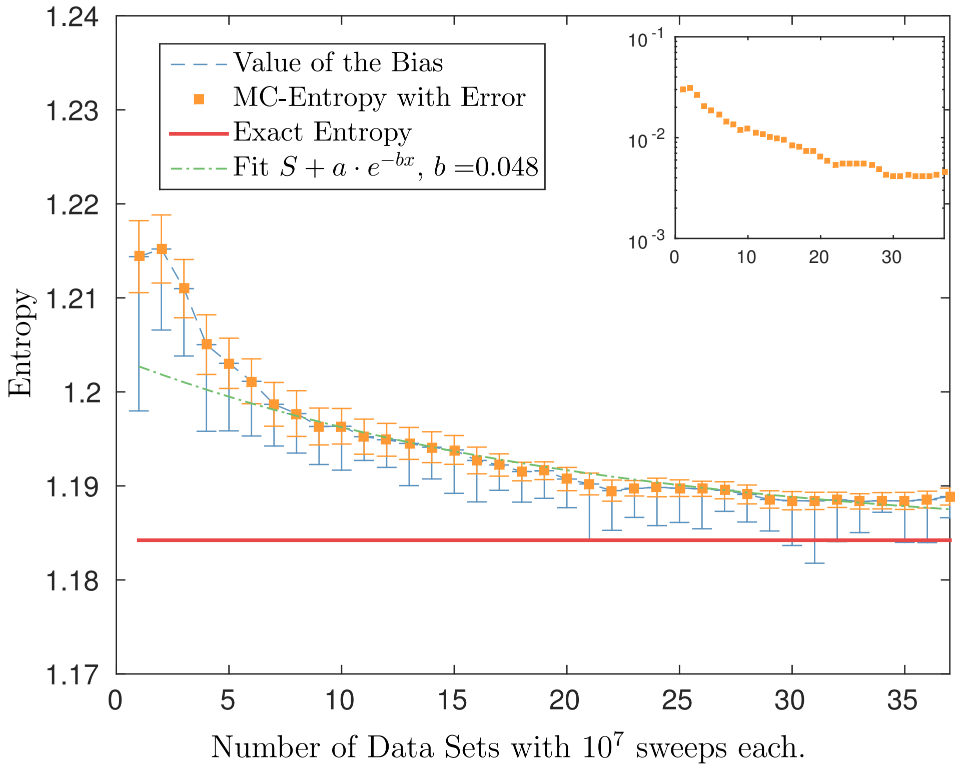

Figure 10.6: Von Neumann entropy of the Monte Carlo data compared to the exact result for an increasing number of data sets, containing 10^7 sweeps each. For the statistical data, both the error (orange) and the bias (blue) are plotted as error-bars. Opposed to statistical errors biases are one sided, and, as expected, they always are positive. This shows again that the entropy from the monte carlo data is too big. Convergence to the exact entropy (red line) can not really be seen on this scale. But the bias is still bigger than the statistical error, which means that the entropy is not converged. This result also depends heavily on the small eigenvalue cutoff (which was 10^{-5}). In addition to this an exponential fit is plotted, to get an idea of how much data one needs to reduce the deviation. In this case about 20 data sets i.e.: $20 \cdot 10^7$ sweeps are needed to reduce the deviation to a factor of e.
The inset shows the function $S_{MC} - S_{ex}$ on a semi-logarithmic scale, showing that the convergence is not exponential.

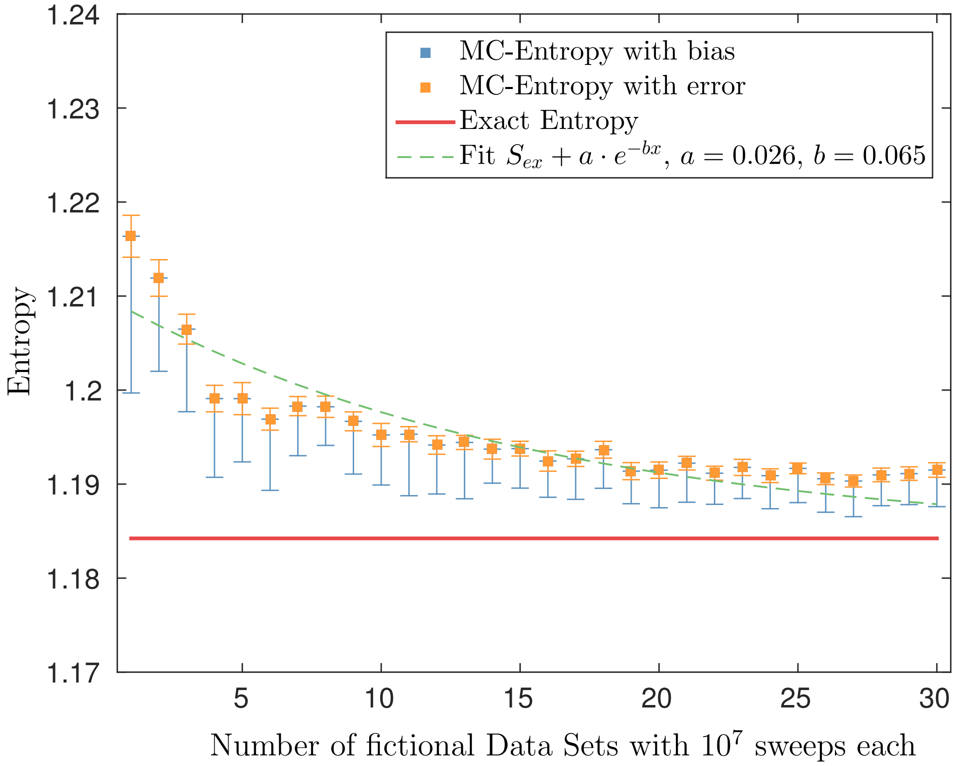

Figure 10.7: Von Neumann entropies of a RDM with artificial noise, compared to the exact von Neumann entropy for a L=12 lattice with 6 sites in the RDM and for $\beta = 80$. The noise added to the exact RDM is normally distributed with mean 0 and standard deviation $\sigma = \frac{1}{\sqrt{N}}$, with N the number of sweeps on the x-axis (don't forget the factor 10^7). Of course, this can only be a rough estimator of the Monte Carlo behavior, because to produce this figure I assumed that every matrix element has the same error of σ. Furthermore the number of sweeps splits between all non-zero matrix elements, making the effective number of data points per matrix element much smaller. To calculate errors and biases 25 noisy RDMs were produced for the same noise σ, which served as bins for jackknife. Putting all 25 RDMs together yielded the shown data point. Nevertheless one can see the same overall behavior as in figure 10.6. The entropy of the noisy RDM is always bigger than the exact one and there is a very slow convergence to the exact value. The exponential decay is about half the value it is from the Monte Carlo data. Regarding how many assumptions were made this is surprisingly good.

10.2 Reducing the Computational Effort

The last section showed that we are in dire need of some ways to improve the performance of the program. The slow convergence of the entropies make Monte Carlo Simulations for larger lattices very inefficient, if the region described by the RDM has more than 6 sites. This chapter shows solutions to this problem.

Tracing out Sites Manually

So far we have seen that huge computational effort has to be made in order to get reliable results for the entropy. Although in many cases the spectrum of the RDM is captured pretty good. We can suspect that the number of sweeps one needs is mostly determined by the size of the RDM, not so much by the size of the total system. This is reassuring, because then the computational effort for enlarging the total system scales with this size and the number of sweeps needed to get reliable entropies stays roughly the same.

Up to now, I did a separate calculation for every number of sites in the RDM. Of course it is also possible to only sample the big RDM, and do the trace over the degrees freedom of the not wanted sites manually. The reason this was not done so far is that, by doing this, some of the configurations produced for the big RDM are not valid Monte Carlo states for the smaller ones. When looking at 6 site RDM, and I want to get the RDM describing sites 1 to 5 I am only allowed to take those Monte Carlo data points, which have the same spin on the sixth site below and above the cut. In short, some data is produced that doesn't contribute to the smaller RDMs. Normally this would lead to an increase in the statistical error, but as we have seen so far smaller matrices need less data to produce reliable results. What happens, if one does the trace manually can be seen in figure 10.8 for the entropies and in figure 10.9 for the corresponding errors and biases. The result looks promising, since the entropies seem to get better the more sites are traced out. Also the errors and biases become smaller, and for 4 sites in the RDM

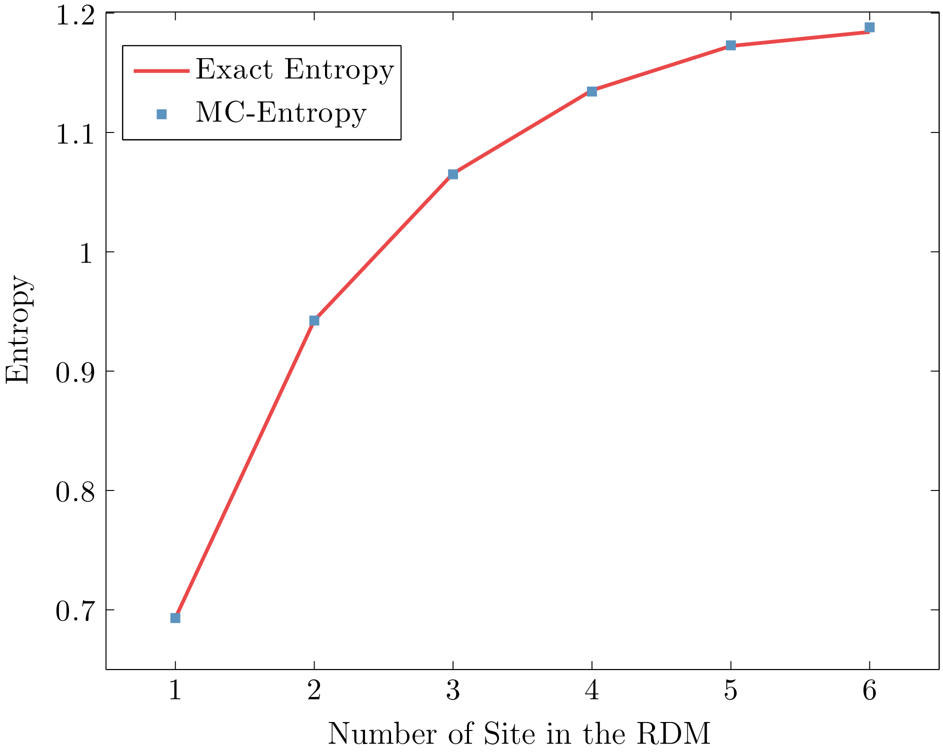

Figure 10.8: Monte Carlo - and exact von Neumann entropy for different number of sites in the RDM. The calculations were made for a L=12 lattice and the maximal number of sites in the RDM was 6. To produce this data $42 \cdot 10^7$ sweeps were made. The entropies for smaller subsystems were calculated by taking the 6 site RDM, and doing the trace manually. As one can see the values seem to get better the smaller the system gets, even if data is lost in doing the trace.

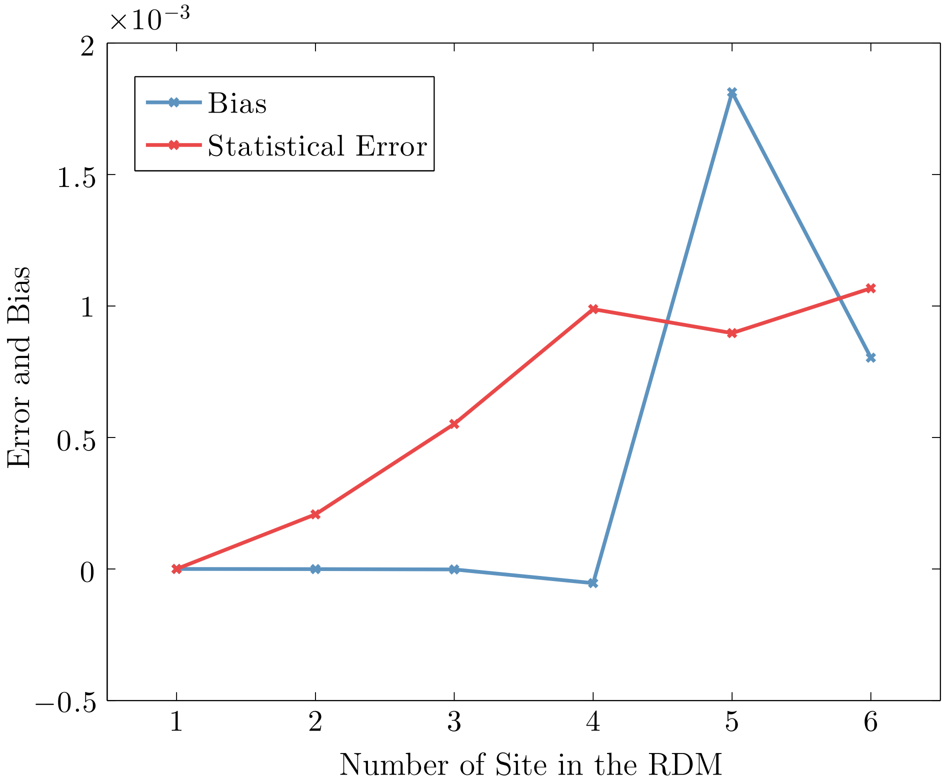

Figure 10.9: Error and bias of the Monte Carlo entropies from figure 10.8.

the bias already is negligible compared to the error.

So we see that only one (very long) Monte Carlo run, for the biggest subsystem-size one is interested in, has to be made, and that the trace can be done manually. If the big system is converged, so are the smaller ones. The data point for 5 sites in the RDM in figures 10.8 and 10.9 is a bit troubling bias-wise, but the value for the entropy shows a far better behavior.

Using Improved Estimators

Until now every new configuration, produced by the DMLA, yielded one entry in the RDM. Therefore, the spin states below and above the cuts specified the matrix element this configuration belongs to. We also saw that it is hard to produce enough data, to be able to sample RDMs of subsystems bigger than about 6 sites.

In section 5.2 we introduced and discussed the loops appearing in the Loop Algorithm and in the DMLA. They are produced in a way that one has the freedom to flip all spins along everyone of these loops. This allows one to make use of improved estimators [1], which use the fact that for some loop structure more than one spin configuration is possible. If those spin configurations have a different contribution to some measurement, one can take of all of them, weight them properly, and add them to the measurement. Essentially the same can be done here.

Imagine a RDM with N sites. We already know that every loop starting at a cut, also has to end in one. This means that there are N loops connecting the $2N$ spins below and above the cut. Flipping all the spins in one loop yields a new configuration (entry in the RDM). So by flipping all spins in one loop 2 configurations are produced. Doing this for every loop that starts and ends in a cut, gives a total of 2^N entries in the RDM.

Doing this has two advantages:

First, it simply increases the data produced[2]. This increase is by no means linear, but scales exponentially in the number of sites in the RDM. In figure 10.1 we saw that the bigger the RDM, the bigger the convergence problem gets. This new way of producing entries in the RDM counters that problem, because the bigger the RDM gets the more data is produced from a single spin-configuration. A small disadvantage of this is that the computational effort of the measurement itself now grows exponentialy, but it turns out that the gain by far outweighs the loss.

Second (and most important in my opinion), making use of all possible configurations respects more of the symmetries of the RDM. The easiest way to see this is that improved estimators make the RDM symmetric by construction. But it does not stop there. For example, the diagonal entry of the RDM with all spins pointing upwards has to be the same as the one with all spins pointing downwards. This is true, because the Hamiltonian has no preferential spin direction, and this property translates to the RDM. To clarify this further, lets take a look at 2 simple examples:

[2] This is the usual argument why improved estimator are called *improved*.

1-Site RDM:

Because of spin symmetry and spin conservation the reduced density matrix for 1 site is:

$$\rho_A = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

Counting matrix element entries, as done so far, yields states $|0\rangle \langle 0|$ and $|1\rangle \langle 1|$ on average with probability 0.5. In most runs, of course, they do not appear the same number of times, so that the estimator for these probabilities is not exactly 0.5, but very close to it. This means that we end up with one number having a statistical error (when utilizing that the trace of the RDM must be equal to 1).

When using improved estimators this changes. In every spin configuration there is a single loop connecting the RDM site below and above the cut. Every configuration that yielded an entry in $|0\rangle \langle 0|$ now also produces one in $|1\rangle \langle 1|$ and vice versa. This means that every configuration produces 2 entries in the RDM with the same value i.e.: 1 or 0.5. Even when doing only one measurement, this way of counting RDM entries would yield the correct result.

So we see that for the problem of having one site in the RDM, the symmetries of the Hamiltonian allowed us to determine the RDM with certainty. Using improved Estimators completely respects this.

2-Site RDM:

Without making use of all symmetries a general form of the 2-site RDM is:

$$\rho_A = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & d & 0 \\ 0 & d & c & 0 \\ 0 & 0 & 0 & 1-x \end{pmatrix},$$

with 4 free parameters a, b, c and d, where x ensures that the trace is equal to one i.e.: $x = a + b + c$. The old way of counting produced one of 6 possible configurations, from which 2 are dependent on the others[3]. Or with other

[3] The RDM must be symmetric and the trace must be one.

words, 4 variables that have a statistical error.

Using the relation of the matrix elements of the RDM with correlation functions (see chapter 6.1) and making use of all symmetries, the possible form of the RDM reduces to

$$\rho_A = \begin{pmatrix} a+0.25 & 0 & 0 & 0 \\ 0 & -a+0.25 & b & 0 \\ 0 & b & -a+0.25 & 0 \\ 0 & 0 & 0 & a+0.25 \end{pmatrix}.$$

In fact there is only 1 independent parameter in a 2-site RDM [4]. Now let us take a look at the 2 possible loop structures shown in figure 10.10. When counting every possible state from the loops, the left loop configuration contributes

$$\rho_A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

to the RDM. The right configuration on the other hand contributes

$$\rho_A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

So again we see that making use improved estimators not only yields more data, but also respects more of the inherent symmetries of the RDM.
From these two simple examples I conjecture that the number of possible loop configurations is linked to the number of independent parameters in the RDM, when taking in account all of the symmetries. The fact that when counting spin states one ends up with such a big bias, compared to the statistical error, could even partially be the fault of breaking those. If this is true, improved estimators could mean a huge improvement to the performance.

[4] $a = b$ since $\langle S_i^z S_j^z \rangle = \langle S_i^+ S_j^- \rangle$

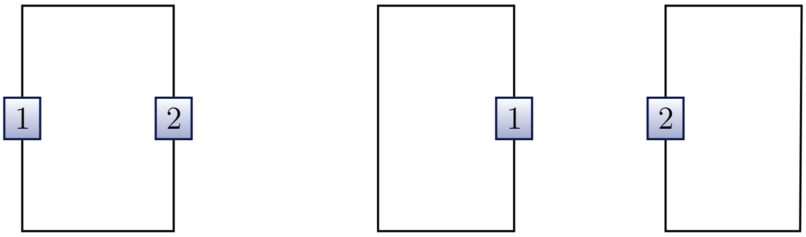

Figure 10.10: The two possible loop structures of a 2 site RDM in the isotropic Heisenberg Model. The squares labeled 1 and 2 symbolize the cuts of the 2 sites described by the RDM. Note that the lines drawn are only symbolic for a (maybe much more complex) loop. Since one flips the spin along a whole loop the complexity of it doesn't matter, because only the spins below and above the cuts are of relevance. What counts is, which cuts are connected to each other at which altitude (below or above the cut). A loop starting at site 1 below the cut can either connect it to site 1 above the cut or to site 2 below the cut. In both cases the 2 remaining spins only have 1 possibility how they connect. This means that there are only 2 possible loop configurations for the 2-site RDM.

As already mentioned, when using improved estimators, one needs to weight the different outcomes of a measurement. Here this is not needed, since the weighting factor is always the same, and only depends on the number of sites in the RDM.
Until now this is only theory, so lets take a look at some results. To be able to compare it to counting only the single spin states, I did the same calculation as shown in figure 10.1, only with 2 differences. First, the number of sweeps was reduced to 10^6 i.e.: a factor 10 less configurations. Second, only one run for the biggest system with 7 sites in the RDM was made. The RDMs for the smaller subsystems were calculated by tracing out one site after the other, as explained in chapter 10.2. The result of this can be seen in figure 10.11 and looks great compared to the previous result.

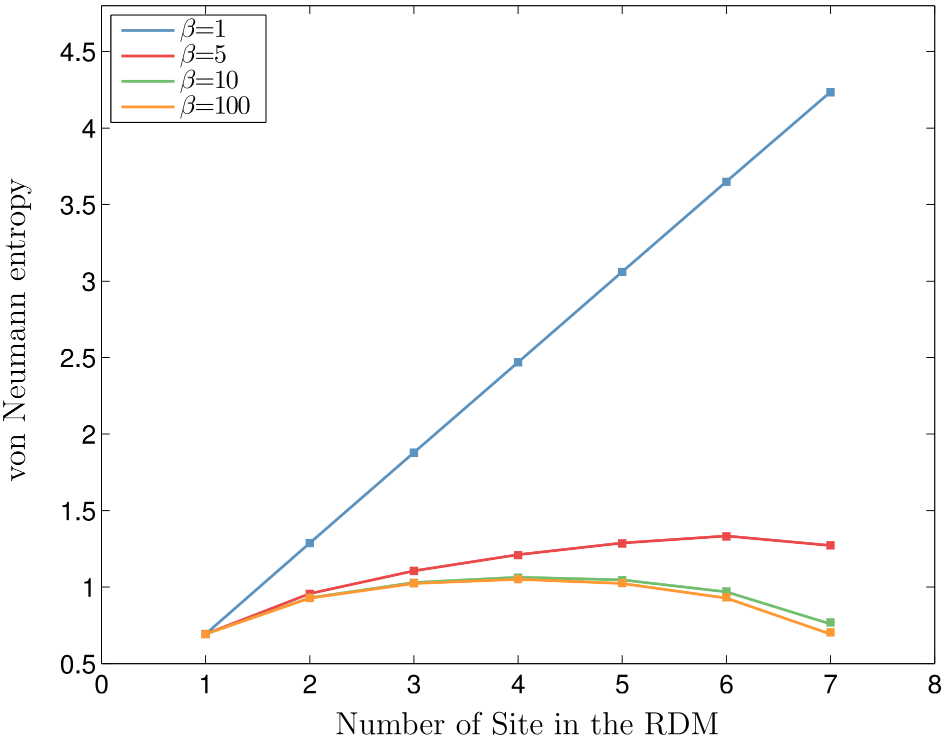

Figure 10.11: Von Neumann entropy versus number of sites in the RDM for different values of β, produced by making use of improved estimators. This result looks very good, since the number of sweeps to produce this was only a tenth of the sweeps needed to produce figure 10.1. And by doing the simulation only for 7 sites in the RDM, the reduction in computational effort is about a factor 50 (this is only a rough estimate - it is not 70, since measurements are more expensive now).

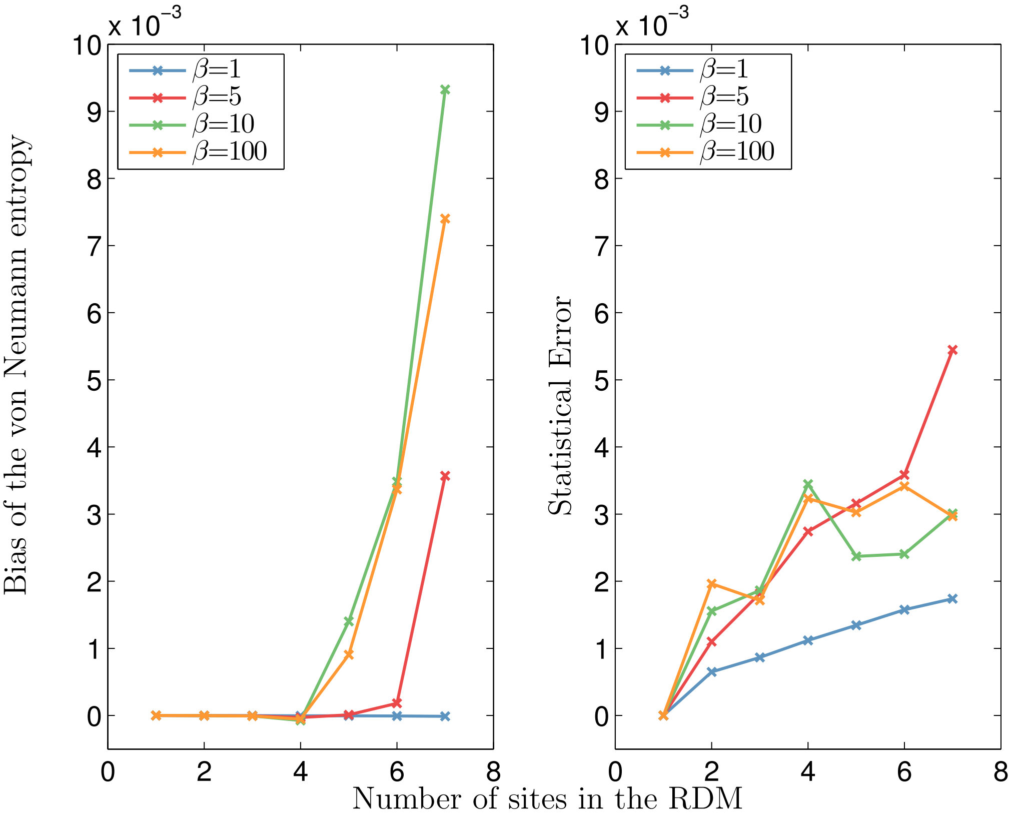

Figure 10.12: *Left:* bias of the entropy calculated with equation 10.1. *Right:* The statistical error calculated using jackknife. Again the bias becomes bigger than the statistical error, but in this case only at about 6 sites in the RDM. It shows that bigger RDMs are still more difficult to calculate. This is not surprising, since the growth in the number of measurements made is smaller than the growth in the number of total elements in the RDM. Having a bias bigger than the statistical error means that the result is not converged. Nevertheless, the entropies seem to be much more reliable.

In this picture no visible deviation from the exact value can be seen, and the result doesn't depend on the small eigenvalue cutoff anymore. To answer the question about convergence take a look at figure 10.12. It shows that the problem with convergence, we were facing before, is still present, but is much smaller. If 10^7 sweeps are done (the same number used for producing figure 10.1) the bias becomes smaller and is about as big as the statistical error.

Finally, I want to emphasize the order of magnitude that using improved estimator enhanced the data. In order to get results, as bad as figure 10.1, one needs to make about $5 \cdot 10^4$ sweeps. This means that this way one can save a factor of 200 in computational effort. Note that comparing the data was only done visually. Hence this is only a rough estimate of the improvement, not some hard fact. Overall, it turns out that the exact convergence of the entropy is still rather slow, but it is much faster than it was before, especially when we additionally correct for the bias. Therefore, we conclude that making a simulation with 10^6 sweeps should be enough for reliable results for a 7-site RDM.

Bias Correction

With improved estimators we already got a huge boost to the convergence of the entropy, but there is one more thing we can do. In the present case the bias is no product of some outlier in any of the bins (as was shown in figure 10.3), but a result of the fact that taking less data yields a too big von Neumann entropy. Because of this, the jackknife bias is a literal bias, with the meaning that we should be able to get better results, if we correct the Monte Carlo data for its value. To take the bias into account, one needs to subtract it from the Monte Carlo value to get the corrected result. To show that using the bias enhances the result, take a look at figure 10.13. It shows the deviation of the Monte Carlo entropies for $\beta = 100$ from the exact value for a different number of sites described by the RDM. The data is the same that was used to produce the previous 2 plots, figures 10.11 and 10.12. Indeed the deviations of the entropies become smaller, and they are, in all cases, of the size of the statistical error (with some finger-crossing for the last data point).

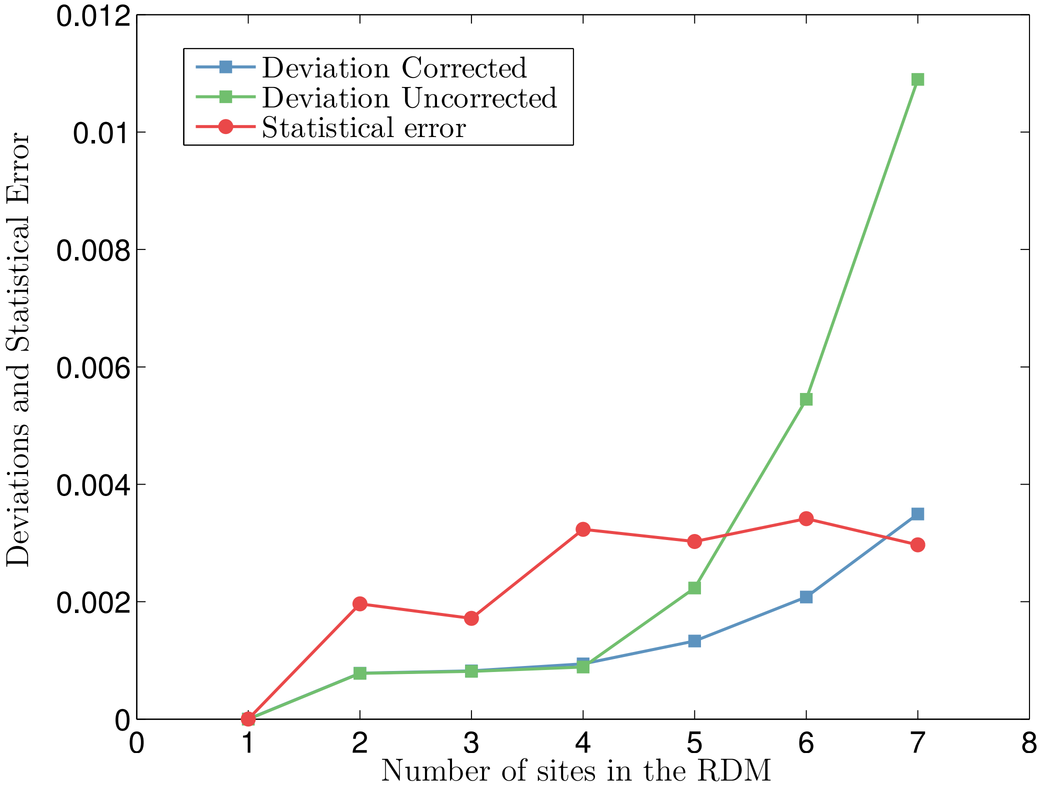

Figure 10.13: The deviation of the Corrected and Uncorrected entropies for the data used in figures 10.11 and 10.12 for $\beta = 100$. As a reference the statistical error is also shown. To correct the entropies, the bias was subtracted from the monte carlo data, and in calculating the deviation no absolute values were taken i.e: it is the MC-entropy minus the exact one. As already expected the result becomes better and the deviation is for all sizes of the RDM of reasonable size (in total as well as when compared to the statistical error).

10.3 New Measures of entanglement and Valence Bond Entropies

As mentioned earlier, there is great interest in entanglement measures for big subsystems, and in the scaling of entropy-like quantities. In section 7.2 4 new proposals, inspired by the Valence Bond entropies, were introduced and called Cluster entropies. Figure 10.14 shows these 4 Cluster entropies for a 1d L=10 lattice at inverse temperature $\beta = 60$ for different number of sites in

the RDM. None of these measures seems to capture the behavior of the von Neumann entropy very well. As already explained, they are not symmetric regarding having N and $L-N$ number of sites in the RDM, and they have a strong even-odd effect. The latter can be explained easily. For an even number of cuts it is possible that loops only connect spins with at the same altitude, like the black loop in figure 7.4. These states do not contribute to any of the measures. If the number of cuts is odd on the other hand, there must be at least one loop starting from below the cuts and ending above. This results in always having at least on loop contributing to these measures. We can also see that CE_0 is on average the same as CE_1. This can be understood by looking at figure 7.4. The green loop has a contribution of 1 to both measures (CE_1 gets divided by 2), so for loops starting below and ending above the cut, these 2 ways of counting are the same. The red loop, on the other hand, has a contribution of 2 to CE_0 and 1 to CE_1, so all loops starting and ending below the cuts are counted differently. The reason why, on average, they are the same is that a configuration where the black and red loop exchange their start and end points i.e. the red loop starting and ending above the cut, is as likely as the drawn configuration. With above and below switched, the contribution to CE0 is 0 and to CE1 still 1. So, on average, CE_0 and CE_1 are the same
CE_2 which counts loops moving through regions from sub-lattice B seems to grow the bigger region A gets.

Figure 10.15 shows the Valence Bond Entropies for the same set of parameters and also compares them to the von Neumann entropy. We find that the existing Valence Bond entropies mimic the von Neumann entropy better than all of the Cluster entropies for the L=10 Heisenberg anti-ferromagnet. Since one is mostly interested in the scaling of the measures, not so much in the exact value, there still is hope for them. Furthermore, the Cluster entropies make sense for thermal states, not only at zero temperature. It could be interesting to find out, what's the relation (if there is one) to thermal entanglement.

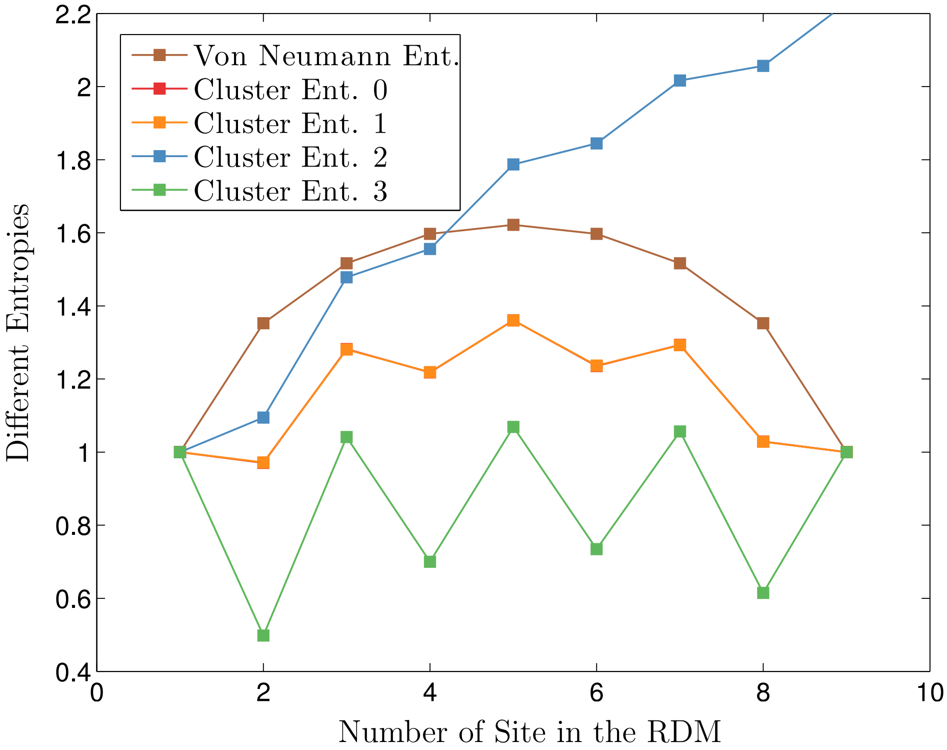

Figure 10.14: Cluster entropies compared to the von Neumann entropies for a L=10 lattice at inverse Temperature $\beta = 60$. The number of sweeps to produce these results was 10^7 and all results are well converged (errors of the order of 10^{-4}) and they have no jackknife bias. Furthermore, this temperature is low enough so that the results are converged to the ground state. This is checked by looking at the value of the Cluster entropies at $\beta = 100$, with the result that the difference in the value was smaller than the statistical error. Moreover the difference between the thermal density matrix and the ground state was calculated, and also showed convergence . As explained in the text CE_0 and CE_1 have the same value, so one sees only one curve. Overall, all measures have a strong even-odd effect, and don't really capture the behavior of the von Neumann entropy. The most surprising result is the graph of CE_2. It simply continues growing and doesn't show any sign of symmetry.

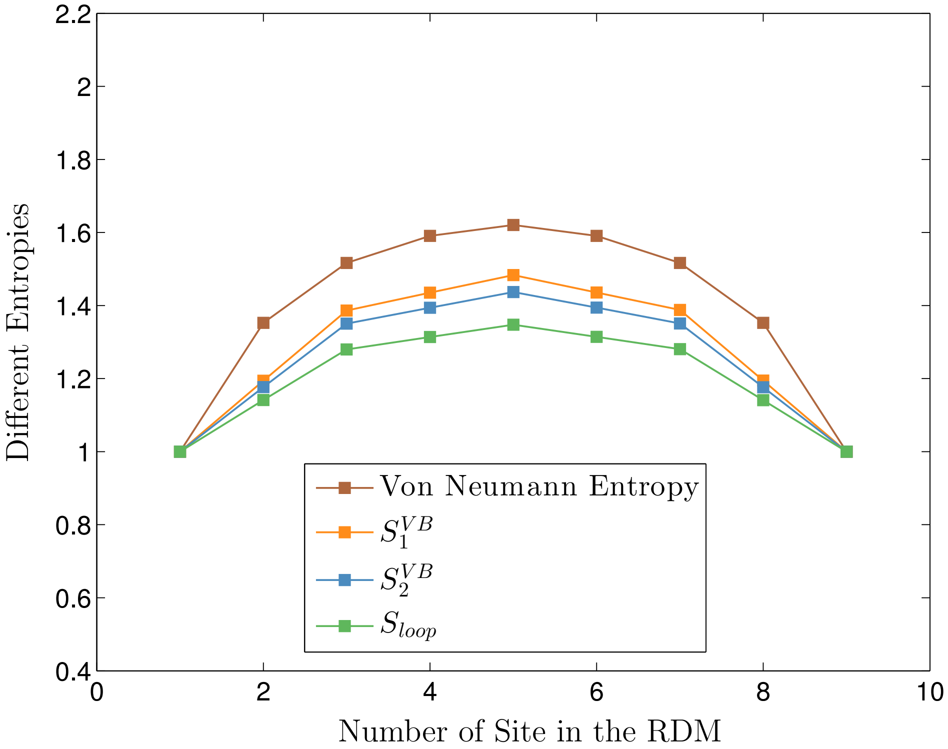

Figure 10.15: Different Valence Bond entropies introduced in section 7 for a L=10 lattice at inverse temperature $\beta = 100$. To obtain these results 10^7 sweeps were made. All results are well converged (errors of the order of 10^{-4}) and have no jackknife bias. The reason for taking a different value of β, as for the Cluster entanglement is that when there is one loop moving from below the cut to above another, it is not possible to interpret this in terms of valence bonds. At a temperature $\beta = 60$ this kind of configuration appears about every millionth sweep (in a run with 10 million sweeps it happened exactly 10 times). So it is rare enough to not be worried about convergence, but I still increased the value of β. To be able to compare the Valence Bond entropies to the Cluster entropies, the scaling of the y-axis of this plot was chosen to match the one of figure 10.14. Clearly, for small systems those measures are better approximations than the Cluster entropies. Especially, since they do not show any even-odd effects, and they are symmetric around $N = 5$ sites in the RDM.

Chapter 11

Thermal Entanglement

In chapter 4 we saw that the von Neumann entropy is a measure of entanglement, only when the system is in a pure state. In thermal equilibrium this is only true if the system is in the ground state. So most of the von Neumann entropies shown in the previous chapter do not quantify entanglement, but were used for comparing the Monte Carlo RDM to the exact one. Since the DMLA is formulated at finite temperature, it would be a waste of potential not to look at thermal entanglement.

In chapter 4 we also noticed that one has to use much more complicated measures to quantify thermal entanglement. Furthermore, all of them depend on the total density matrix not on the reduced state. This means that, with the DMLA, we can not calculate the entanglement between some big region and a small one, as we did when the system was in the ground state. Now we can only look at the entanglement between 2 regions, both described by the RDM. This means means that both of them can only consist of a few sites. Still, there are advantages to full diagonalization, namely, these regions can be embedded in a big *environment*[1]. For example, we can take a look at how, and if two blocks of spins are entangled, when a certain number of sites

[1] Note that at this point we changed perspective. Before, the total lattice was the system and we calculated the entanglement between two regions of it. Now we treat the rest of the lattice as some environment for our (smaller) system.

is between them, and how this entanglement changes with temperature. The easiest measures presented in this work were the so called Negativity and the Logarithmic Negativity (see equations 4.20 and 4.21). They both rely on the fact that the partial transposition of a separable state is again a valid quantum mechanical state. Since states have a trace norm equal to 1, both measures show how much the partially transposed density matrix is not a state itself. The computational effort to calculate both of them is mostly the diagonalization of the RDM, which is, for small RDMs, next to nothing compared to the effort needed to sample the RDMs in the first place. Like for the von Neumann entropy there are scaling laws for the Logarithmic Negativity [38] for systems described by a Conformal Field Theory. Again, with the DMLA scaling properties can not be observed, since the subsystems are too small.

Figure 11.1 shows the Logarithmic Negativity of neighboring spins as a function of β. The behavior shown there is not very surprising, since the negativity (and also the logarithmic negativity) is closely related to a 2 point correlation function. Both measures look very similar when plotted. To make this easier let us write equation 4.23 again:

$$N = \begin{cases} 0, & \text{if } \langle S_i^z S_j^z \rangle \geq -\frac{1}{8} \\ -\Big(2\langle S_i^z S_j^z \rangle + 0.25\Big), & \text{if } \langle S_i^z S_j^z \rangle < -\frac{1}{8}. \end{cases}$$

If the correlation function $\langle S_i^z S_j^z \rangle$ becomes bigger than $-\frac{1}{8}$ the negativity is always zero.

Figure 11.2 shows the Logarithmic Negativity for various temperatures as a function of the distance of the two concerning spins. We note that if the spins are not nearest neighbors, the entanglement seems to vanishes. This could be a consequence, of the correlation function $\langle S_i^z S_j^z \rangle$ being bigger than $\frac{1}{8}$. A similar result was obtained by Osterloh et al. [51]. They looked at the concurrence as a function of distance between the two spins in the quantum Ising Model and found that it vanishes if the 2 spins are further apart than next nearest neighbors. This happened even at quantum phase transitions, which is somewhat curious, since at phase transitions the correlation length diverges. On the other hand for the XY-Model they found that this range can diverge.

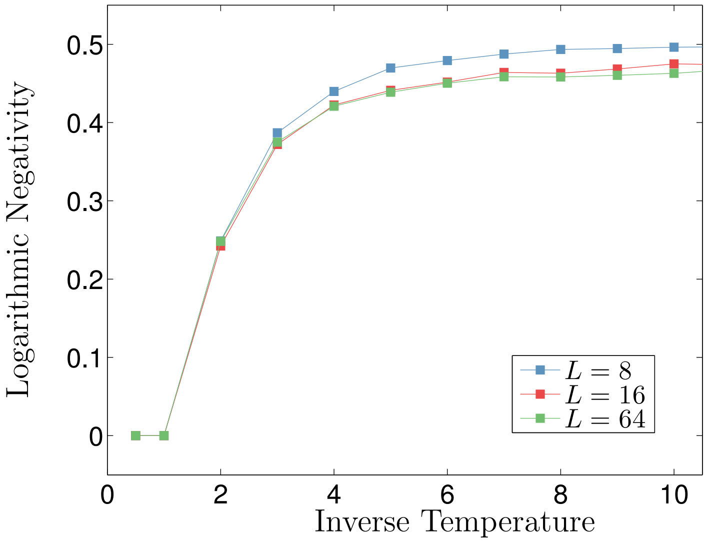

Figure 11.1: Nearest neighbor Logarithmic Negativity of 2 spins as a function of β for various lattice sizes. Data was also available for $L = 32$, but it is not shown, since it would be hard to keep the lines apart. To produce this plot 10^6 sweeps[3]were made. The statistical errors are smaller than the size of the data points. What is interesting is that entanglement seems to be very stable over a wide range of β-values (see figure 11.2 for the value of $\beta = 200$). Somewhere around $\beta = 4$ the entanglement starts to decrease rapidly, and stays zero for inverse temperatures lower than 1.

[3] In the Multi-Loop Algorithm a sweeps consist of producing all loops, and flipping each one of them with probability 0.5.

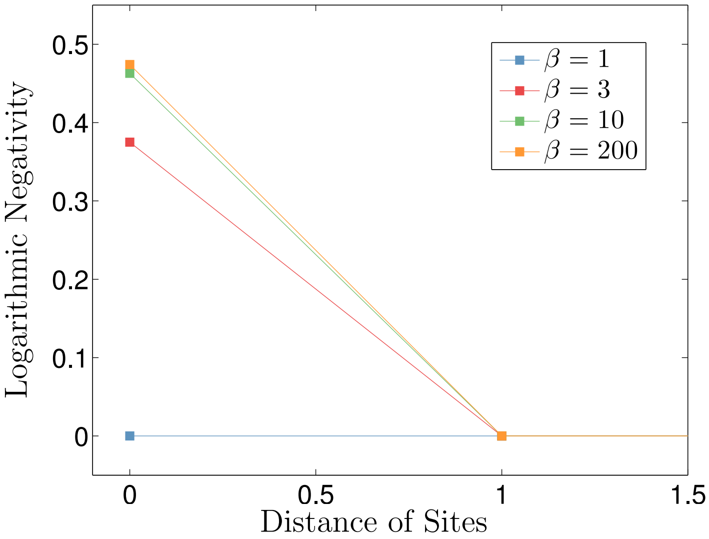

Figure 11.2: Logarithmic Negativity for a $L = 64$ chain as a function of the distance between two spins, for various values of β. Note that distance 0 means that the spins are nearest neighbors. This figure shows that 2-spin entanglement vanishes, if they are not nearest neighbors. As figure 11.1 already suggests the Logarithmic Negativity is very stable for a big range of β values. Note that the lattice consists of 64 sites, which means that $\beta = 10$ is far from the ground state. Still the Logarithmic Negativity doesn't change much when decreasing β.

I also looked at logarithmic negativities with more spins involved, but the overall picture is always the same. As a function of temperature the entanglement starts to decrease quite fast at a certain value of β (somewhere around $\beta = 2$ in all cases), and as a function of distance it vanishes, if the spins are further than one site apart from each other. Showing more results would not yield any further insight.

Chapter 12

Results for 2 Dimensional Systems

Up to this point every result presented was calculated for 1 dimensional chains of various lengths. One of the biggest advantages of Monte Carlo methods is that for spins and bosons they also work quite well for systems with higher dimensionality. So it is only natural to present some results obtained for 2 dimensional systems.

The Algorithm itself doesn't really change, the only difference is that now there are 4 possible neighbors a loop can jump to. To be sure that the DMLA still works in 2 dimensions (and that the program works correctly) take a look at figure 12.1. It shows for a 2×2 lattice the von Neumann entropy for different number of sites in the RDM, at various temperatures. From this result we conclude that the Algorithm is correct.

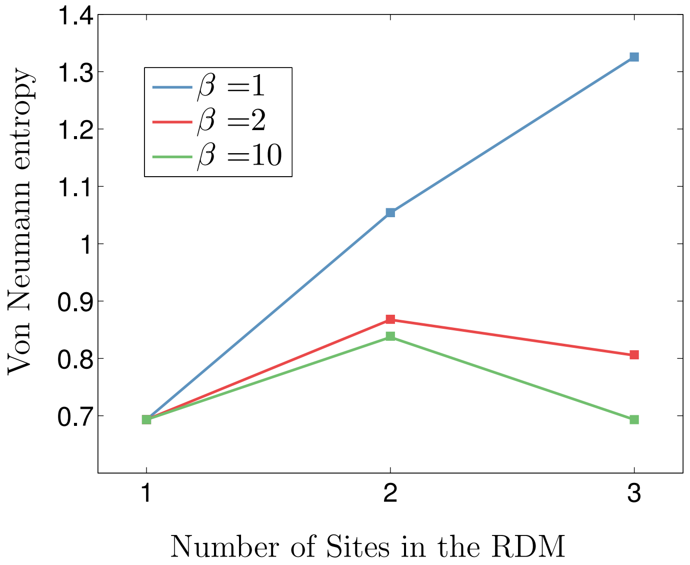

Figure 12.1: Exact, and Monte Carlo von Neumann entropies for various values of β for a 2 dimensional 2×2 lattice. As before, the solid lines show the exact values obtained by full diagonalization, while the squares are the entropies obtained by diagonalizing the stochastic RDM. 2 sites in the RDM means, that it consists of two spins which are nearest neighbors (opposed to two spins being diagonal to each other). This ambiguity is not present when having 1 or 3 sites in the RDM. To produce this plot 10^6 sweeps were made, and improved estimators were used. Furthermore the result was corrected with its bias, as explained in the previous chapter. The errors are much smaller than the markers and the problem with having a huge bias is not present (since there are only 3 sites in the RDM). The purpose of this figure is to show that the algorithm (by that, I mean the actual code) for the 2 dimensional system works.

Entanglement

As we did in the one dimensional case, let us take a look at the thermal entanglement between 2 regions both described by the RDM. Again, the Logarithmic Negativity is our measure of choice. In two dimensions there are many arrangements of the regions one can think of. I choose to let the RDM consist of 2 blocks, consisting of 2 spins each, moving away from each other as explained in figure 12.2.

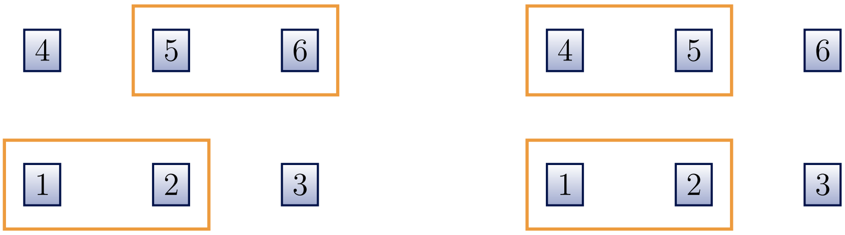

Figure 12.2: This figure shows 2 sections of a 2 dimensional lattice. The orange squares define, which sites belong to the RDM. In the left picture the RDM consists of sites $(1, 2, 5, 6)$. This arrangement will be referred to as having a distance d of 1 (since the leftmost spins in the upper block is 1 site shifted to the right). The right picture on the other hand shows a situation where the sites $(1, 2, 4, 5)$ are in the RDM, and it has a distance d of zero.

The results of this can be seen in figures 12.3 and 12.4. The former shows the Logarithmic Negativity as a function of β for various distances, while in the latter one can see the same measure as a function of distance for various temperatures. The result is very similar to those shown in the one dimensional case (figures 11.1 and 11.2).

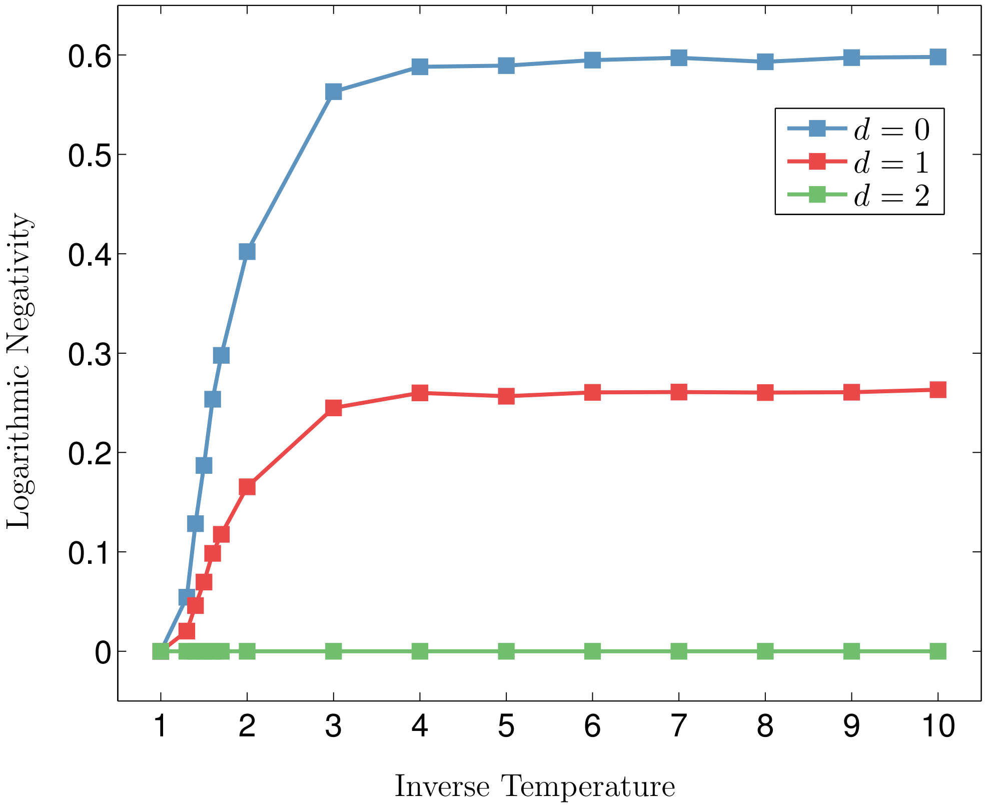

Figure 12.3: Logarithmic Negativity between 2 blocks consisting of 2 spins each as a function of β, for various block-distances on a 2 dimensional 16×16 lattice. The meaning of the different values of d is explained in figure 12.2. To produce this plot, 10^6 sweeps were made. Statistical errors are much smaller than the size of the markers and no relevant bias occurred.

As in the 1 dimensional case, the entanglement is quite stable over a wide range of β-values, and decreases quickly somewhere around $\beta = 1.5$. In cases of $d = 0$ and $d = 1$ there are at least two spins of the two blocks nearest neighbors. As soon as this is not true anymore the Logarithmic Negativity becomes zero. Nevertheless the value for $d = 0$ and $\beta = 10$ (Logarithmic Negativity $E_N = 0.263(3)$) is not twice the value of $d = 1$ and $\beta = 10$ ($E_N = 0.598(2)$) , so there seems to be some entanglement, which is not explained by contributions from nearest neighbors only.

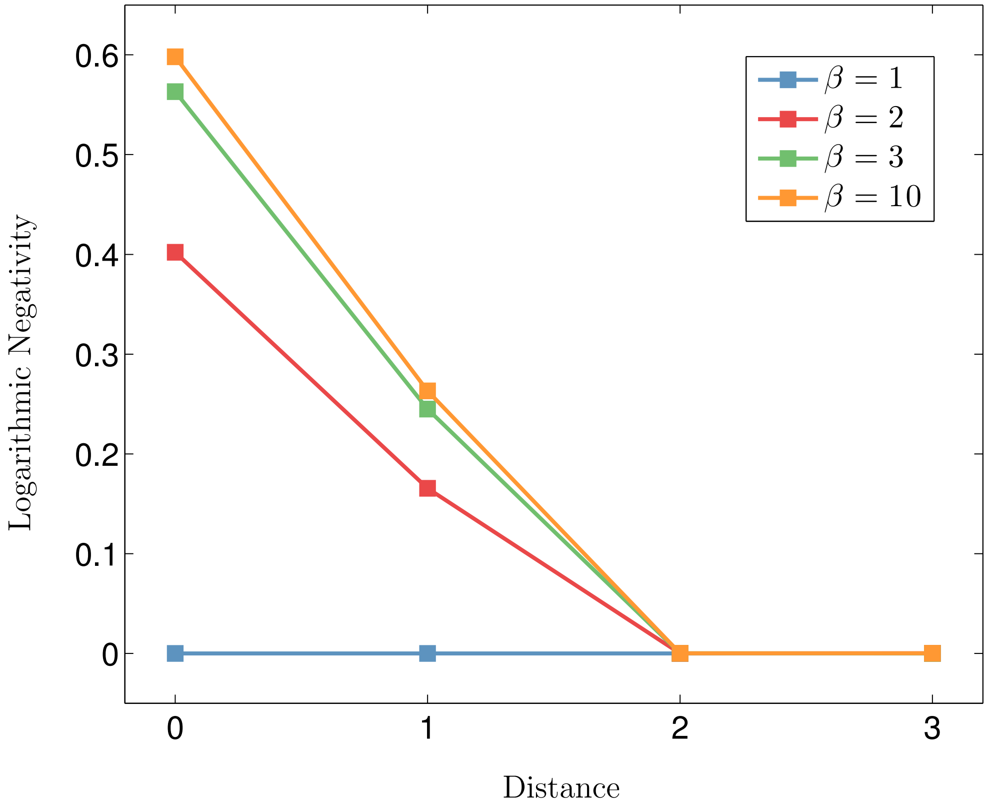

Figure 12.4: Logarithmic Negativity between 2 blocks consisting of 2 spins each, as a function of distance between the blocks. These results are obtained from the same data that was shown in figure 12.3.

Again we see that entanglement vanishes, as soon as there are no nearest neighbors in the two blocks, which happens at distance $d = 2$. This result is not very surprising, since in non-critical systems the correlation length is finite which also means that the length scale at which entanglement occurs must be finite. It is interesting tough that it decreases that quickly. This again could be a consequence of a similar maximum behavior, which we obtained for 2 spins (see equation 4.23).

Chapter 13

Conclusion

In this work we successfully introduced and tested an algorithm that stochastically determines reduced density matrices, called density matrix loop algorithm (DMLA). This was done by imposing open boundary conditions in the path integral of $e^{-\beta H}$ on all sites described by the RDM. The actual updates in the worldline configuration were done with a variant of the Loop Algorithm [1], which respected these boundary conditions. This new way of doing monte carlo was tested for the isotropic Heisenberg antiferromagnet introduced in chapter 2. Descriptions of the Loop Algorithm and the DMLA can be found in the chapters 5 and 6, respectively.

With a working algorithm we did some first tests, and compared reduced density matrices by means of the von Neumann Entropy 4.1. At first, we encountered problems with the convergence (see figure 10.1) of the measurements, since statistical errors in the matrix elements translate into a positive bias of the von Neumann entropy. This was extensively discussed in chapter 10. We were able to cure this problem by using improved estimators, which on the one hand respect symmetries of the RDM, and also simply (exponentially) increases the amount of data produced from a single configuration. The result can be improved even further, if it is corrected for its bias, see chapter 10.2.

The von Neumann entropy is not only useful as a tool to compare reduced density matrices, it is also a measure of entanglement, if the total state of the system is pure. Moreover it has an interesting scaling behaviour under increasing the size of the bipartition, if the system can be described by a conformal field theory (see equation 4.3). In chapter 4 we discuss entanglement and how to quantify it when dealing with pure, as well as with thermal states. Useful ways to measure thermal entanglement turned out to be the negativity 4.20 and the logarithmic negativity 4.21. Results of the latter are discussed in chapter 11 for a 1 dimensional system, and in chapter 12 for a 2 dimensional lattice. The Logarithmic Negativity has the property of being zero if no two spins the the two regions are nearest neighbors. It shows to be very stable over a wide range of temperatures, and decreases very quickly somewhere around $\beta = 2$. This behavior was observed for 1- and 2 dimensional systems.

In chapter 7 and 8, we found that the DMLA can also calculate the so called Valence Bond entropies, equations 7.5, 7.5 and 7.10. All 3 measures are supposed to approximate the von Neumann entropy. Inspired by the Valence Bond Entropies we thought of some new ways to do this, which led to the definition of the Cluster Entropies. Chapter 10.3 compares all those different entropies with each other. Unfortunately it turns out, that the Cluster entropies exhibit strong even odd effects, and are not symmetric with respect to $|A|$ and $L - |A|$ sites in the RDM, if L is the total number of sites in the lattice.

Acknowledgements

Here, at the end of this thesis, I would like to thank all the people that made this thesis possible.

My deepest gratitude goes to my supervisor Prof. Hans Gerd Evertz, for his great support, and for answering my many questions very patiently. Without his guidance this work would not have been possible.

My deepest thanks goes to Claudia Mair, my life partner. Without You, all the things I achieved would only be worth half as much.

I would like to thank all my colleagues and friends, for their help and the fruitful discussions we had about whatever our mind was up to. The importance of this should not be neglected, because it puts the work we are doing into perspective, and it enables everyone of us to look outside our bubbles.

Last, but by no means least, my thank goes to my parents and the rest of my family. Without their emotional and financial support my studies would not have been possible in the first place. I would not want to close this thesis without mentioning my nieces and my nephew, Eleonora, Johanna, Josephine, Juliane and Alois - greetings from uncle Daia.

Appendix A

Testing the Continuous Time Loop Algorithm

In chapter 5 the basics of the Loop Algorithm for the Heisenberg model were introduced, and the limit to continuous time was presented. Before we are able to sample reduced density matrices, we first need a program for the Loop Algorithm. It took more time than expected to get a reliably running code.
At the beginning of this section I give a short overview of how I structured the program. Then some results will be presented which we will compare to exact calculations and results from the ALPS-library Loop Algorithm [52].

A.1 Implementation of the Loop Algorithm

Here I will give an overview about how the program I used works. If one is interested in a more detailed explanation see appendix B. All the data needed to construct and flip the loops are stored in a class called *lattice_1d* in the 1 dimensional case and *lattice_2d* if the dimension is 2.

Since in the continuous time limit the plaquette-concept gets useless, one

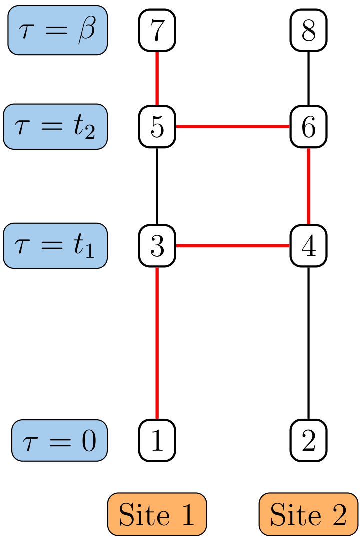

Figure A.1: A worldline configuration of a 1d lattice with 2 sites and 8 events indicated by the numbered boxes. Normally events define a worldline jump, but on every site there are 2 additional events at $\tau = 0$ and $\tau = \beta$ (events with the numbers 1, 2, 7 and 8). As explained in the text, the events store some variables needed to construct the loop.
Event 3 for example stores: Spin points down, time is t_1, its place is 1, and its partner event is event 4, stored as an iterator (pointer).
Event 5 on the other hand has: Spin up, time t_2, place 1 and partner event is event number 6.

only stores times at which worldlines jump. One such jump is modeled by two variables of the *struct* event which stores the time of the jump, the spin after [1] the jump, the site at which it happens and an iterator (closely related to a pointer) to the event the worldline jumps to. Conceptionally (nearly) always 2 events are bound together this way, and henceforth we call the one event partner-event of the other.

A single site of the lattice is modeled by the class *site* which essentially consists of an index numbering the site in the lattice and a *list* [2] of *events*, which has at least 2 entries at inverse temperature 0 and β. These 2 *events* are not points at which a worldline jumps, but have the purpose of having

[1] After means in direction of higher imaginary time.

[2] By list I mean the c++ data type list

fixed starting and ending points of the list. The container class list is used, because very often elements in the middle of the list will get created or deleted and in these cases lists perform better than vectors. The drawback of using lists is that one has to work with iterators, which need to be incremented to get to the next element.

Finally, the whole lattice is modeled by the class *lattice_1d* and *lattice_2d* consisting of a vector of site class elements, inverse temperature, and some variables needed to construct loops and generate random numbers. With this structure it is pretty easy to move around the space-time-lattice. Simply pick the iterator of the start point, increment (decrement) it if one wants to move to higher (lower) temperatures respectively. When arriving at temperature β move to the start of the list to continue further (or to the end when arriving at 0). In case one wants follow the worldline one only needs to access the iterator of the partner event.

Constructing a loop becomes picking a random start-time and start-position and inserting an event at this point [3]. Then reiterate the steps described in chapter 5, until the loop closes. Every time the loop jumps without arriving at a worldline-jump, new events get introduced into the lists of the 2 concerning sites. Because the loop flips all the spins along its way, it is possible (and likely) to end with a configuration with events not needed anymore. This has to be cured by looking at all events the loop passed and deleting the ones which are redundant.

A.2 Testing of the Loop Algorithm

The continuous imaginary time was essentially responsible for some not expected bugs, which were not that easy to find. To make sure that the loop algorithm itself works I present in this section some measurements made with my program, and compare it to exact[4] results. To reduce autocorrelations measurements were not take after every loop. During thermalization the av-

[3] Note that this event also has no partner event and will get deleted once the loop is closed. It is needed so the loop has a fixed point at which it ends.

[4] Produced by completely diagonalizing the Hamiltonian

erage loop length (L_L) was determined. Measurements were taken, after the number of loops was $N_{loops} = \frac{N_{sites}\beta}{L_L} + 1$ i.e.: after a region as big as the space-time lattice was flipped.

In this section I will show values of the energy (figure A.2) and susceptibility (figure A.3) for several L and β. To calculate the former the expectation value of $\sum_{\langle i,j\rangle} S_i^z S_j^z$ was calculated by averaging over all β values. Because of spin symmetry the total energy is 3 times this value multiplied by the coupling constant J. To calculate the susceptibility χ on the other hand, one needs to use improved estimators [1]:

$$\chi = \frac{\beta^2}{4} \left\langle \frac{w_c^2}{L_L} \right\rangle ,$$

with w_c the temporal winding number of the single loop.

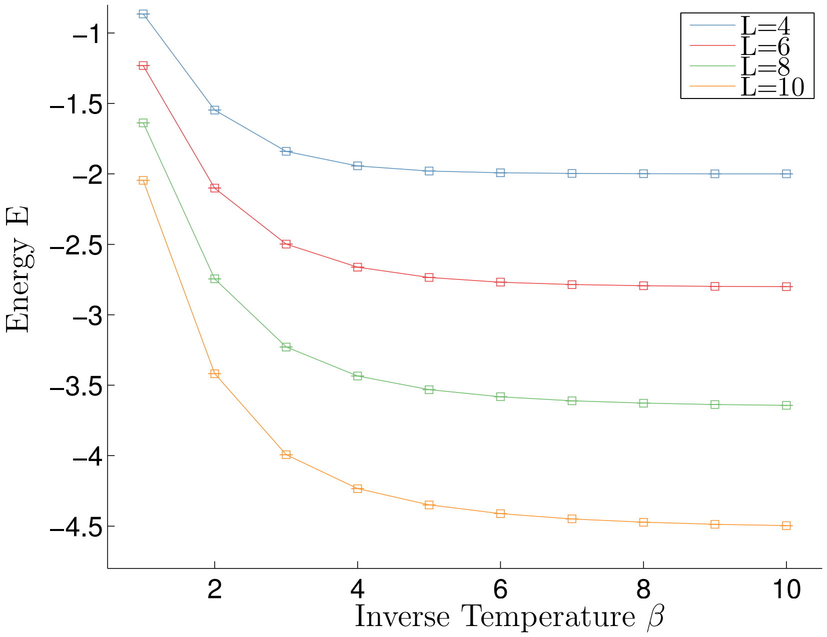

Figure A.2: Energy versus inverse Temperature β for a 1-dimensional Heisenberg chain with length L. The coupling constant was J=1. The solid lines were obtained by calculating the full density matrix $\rho = \frac{1}{Z}e^{-\beta H}$, and using the well known relation for the expectation value of the energy: $\langle E \rangle = tr\rho H$. The squares are values calculated by the loop algorithm with 10^7-sweeps. Note that the errors of the Monte Carlo results are also shown, but they are very small. They were calculated using jackknife, and their convergence was checked. Since it is not possible to detect visibly if the Monte Carlo values converge to the exact result, the proportion of values with a greater distance from the exact value than the error was counted. It turned out to be 27%, which is about the proportion expected using elementary statistics (32%). Overall, this result is a strong indicator that the algorithm works correctly.

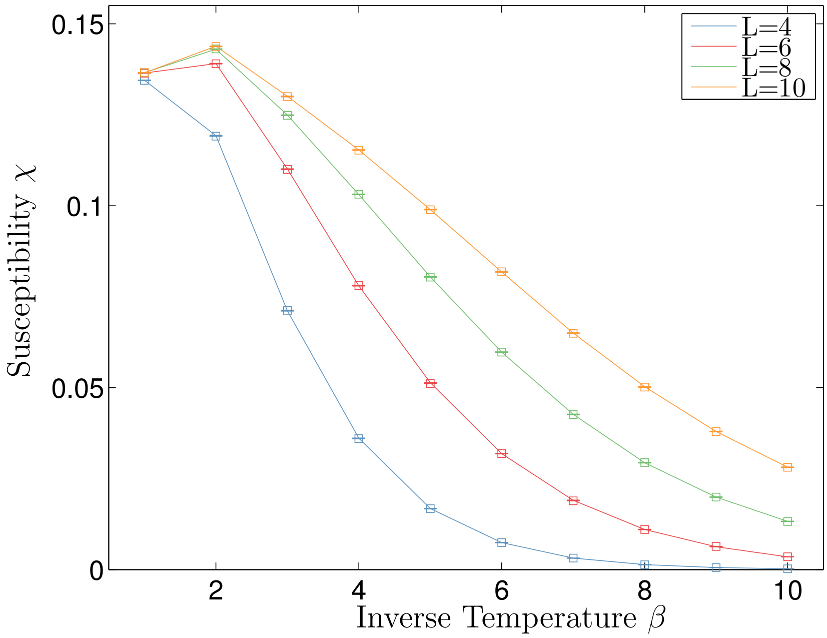

Figure A.3: Susceptibility χ versus inverse temperature β for a 1 dimensional Heisenberg chain with length L. The coupling constant was J=1. The solid lines were obtained by using the *Looper Algorithm* [53] from the ALPS project [52], whereas the squares are measurements of the algorithm I wrote. Both Monte Carlo Algorithms made 10^7-sweeps. Note that in both cases error-bars are printed, but they are very small. The errors for my algorithm were calculated using jackknife. Both errors are of the same size and only 5% of the values are further apart than the sum of the errors from both Monte Carlo procedures.

Appendix B

Pseudo Code of the DMLA

In this section I will give a more detailed description of the actual code I used, than I gave in section A.1. All the functions are presented with no arguments, because of readability. In addition to this all variables can be stored as members of the class, in which case the functions do not even need any.

I wrote the pseudo code quite abstract, because the intention of this section is to give the reader a fundamental understanding of the algorithm itself. So although I used iterators of lists (the language I programmed in was C++) quite heavily, they only rarely appear in the pseudo-codes. If one is interested in the structure and data-types used see chapter 5. The relevant function one needs to call is *create_loop(double J)*. J is the coupling strength of the Heisenberg Model (see equation 2.2). Physically, the relevant parameter is the product βJ, but I chose β to be a member variable of the class lattice, while J is provided when the loop is constructed. In algorithm 1 one finds the pseudo code for this function.

This algorithm needs some remarks:

- Mathematically, it is not possible to have 2 events at the exact same time. If the variables in the program would be really continuous, this wouldn't cause any troubles. Since they are not, one has to make sure that there are no two events at the same time and site. This is the reason for the first while-loop. Of course, one has to make sure of this every time a new event gets introduced, not only here. For the sake of readability I will not include this in the other functions.

- For performance reasons every event not only has an iterator to its partner-event (i.e.: the event to which the worldline jumps), but also includes iterators to the event previous in imaginary time of all neighbors. They are needed, when one has to determine the minimal time interval in which the worldline configuration of the neighbors doesn't change (see chapter 5). So when a new event gets introduced, one has to change the iterators of all neighbors accordingly.
- As already mentioned in chapter A.1 after the loop closes, one is likely to end up with a configuration with unnecessary events in it. To be able to delete them, all events a loop moves through are stored in the variable graph. The function *clean_up_events()* looks at all these events, and decides if they are still needed, or not. All excess ones get deleted.

The function *create_loop* calls *make_fragment()*, which moves the loop by one step. Its pseudo code can be found in algorithm 2. The first line needs some more explanation, since I didn't find a way to formulate it as correct and short, as it is required in pseudo code. Assume the loop is at time T at position P. This point does not always have an event. What one needs to do, is to find the minimal time-difference Δt in which the spin state of P and its neighbors doesn't change. Of course it makes a difference if the loop moves upwards or downwards, since different events are the closest in either case.
To determine to which neighbors the loop is allowed to jump 2 simple rules have to be followed:

- The spin of the neighbor must be different than the spin of the current position.
- The loop must not have visited this part of the space-time lattice before.

If the loop produces new events the function *introduce_jumps()* (see algorithm 3) is called. Otherwise it calls the function *move_in_time()* (see algorithm

4). The latter moves the loop forward to the time determined in the first step. There it either stays at the same position or follows the worldline if the loop reaches an event. The pseudo code shown here is rather long, but it is essentially two times the same. Once, if the loop moves upwards in time, and once, if it moves downwards. Also note that in my actual program I didn't write these 2 functions separated, but everything was included in *make_fragment()*. In this work however *make_fragment()* would have become too long, so I decided to split it up.

Algorithm 1 function create_loop(J)

```
get random start time t_start
get random start site p_start
while t_start is no valid start time do
    get random start time t_start
insert start event on site p_start at time t_start
change all neighbor iterators so they point to the right events
add iterator of start event to variable graph
loop_closed=false
loop direction is upwards(downwards) if spin at start-point is up (down)
while loop_closed==false do
    call function make_fragment()
    if loop_closed==true then
        call function clean_up_events()
return
```

It is important to note that the algorithms presented so far don't include the cuts necessary for calculating the RDM. Luckily they can easily be introduced as events. To tell them apart from actual worldline jumps I added a boolean variable *is_cut* to every event, defining if this event is a cut (true) or if it is not (false). Besides that, only a few changes have to be made to allow cuts:

- One has to make sure that cuts never get deleted. This only changes the function *clean_up_events()*.
- Once a loop arrives at a cut, it moves back to the start, and continues in

Algorithm 2 function make_fragment()

```
Find minimal time interval t_min in which worldline configuration of neighbors does not change    ▷ This line is just a filler - what one needs to do exactly is explained in the text
Find the neighbors to which loop is allowed to jump
N_jump = Number of these neighbors
Draw N_jump random numbers r from exponential distribution with parameter $\frac{J}{2}$
t_exp = min(r)
N_exp = neighbor to which min(r) belongs
if t_exp <t_min and at least one jump is allowed then
    call function introduce_jumps()
else
    call function move_in_time()
return
```

Algorithm 3 function introduce_jumps()

```
if Loop Moves up in time then
    new_time = current_time+t_exp
    New direction is downwards
else
    new_time = current_time-t_exp
    New direction is upwards
introduce event at new_time at current_point and N_exp
change all neighbor iterators so they point to the right events
add new events to graph
new_pos=N_exp
change spin of fragment
```

Algorithm 4 function move_in_time()

```
if Loop Moves up in time then
    new_time=current_time+t_min
    if Loop closed then
        change spin of fragment
        closed=true
    else if Arriving at time $\beta$ then
        new_time=0
        new_pos=current_pos
        change spin of fragment
    else if Arriving at a Worldline jump then
        change spin of fragment
        new_pos = position of partner event
        add event and partner event to graph
        New direction is downwards
    else
        new_pos=current_pos
else
    new_time=current_time-t_min
    if Loop closed then
        change spin of fragment
        closed=true
    else if Arriving at time 0 then
        new_time=$\beta$
        new_pos=current_pos
        change spin of fragment
    else if Arriving at a Worldline jump then
        change spin of fragment
        new_pos = position of partner event
        add event and partner event to graph
        New direction is upwards
    else
        new_pos=current_pos
return
```

the opposite direction it started. So if the loop started to move upwards on spin-up regions and downwards on spin-down regions, it now moves in the opposite direction (down if spin is up and up if spin is down). This has to be taken into account when the spins of a fragment are changed in the functions *move_in_time()* and *introduce_jumps()*

- The last change one has to do is to define what happens exactly when the loop arrives at a cut. Therefore one only needs to change the function *move_in_time()*. For every direction another Else-if-statement has to be included, see algorithm 5. Note that if a loop arrives at a cut a second time it closes.

Algorithm 5 modifications of the function move_in_time() to also include cuts

```
if ··· then
    ...
else if Arriving at cut then
    change spin of fragment
    new_pos =start_pos
    new_time =start_time
    new direction is opposite to start direction
    if Arriving at cut a second time then
        closed =true return
    else
        return
else if ··· then
    ...
```

Multi-Loop Variant

I made use of the fact that the loops in the Multi-Loop Algorithm are closely related to the Valence Bonds and that they can be used for improved estimators. Changing the program to the Multi-Loop Variant is not very hard, only relatively few lines of code have to be changed. Conceptionally one thing changes: Events are not times, where worldlines jump anymore. Instead they define the loop structure of the configuration. At these points there still can be a worldline jump, but depending on which spin orientation one chooses, the worldline can as well move straight. I will not present a pseudo code for this, but give an overview about the most important things one needs to change. Note that this is by far not the most efficient way to do this, but when one already has a single loop Algorithm it is still useful, and the changes can be made very quickly.

- All loops need to be constructed, so one produces as many single loops as it takes to cover the whole space-time lattice. To do this one needs to keep track of all events already visited, and make sure that another loop doesn't move through those regions.
- Loops now should start directly at the events, not at some random point anymore. Since the whole lattice needs to be covered, it is convenient that they always start to move upwards in imaginary time. If the spin at this event points downwards the direction the loop moves is opposite to the spin. This has the same effect, as when one reached a cut in the Single Loop Variant.
- All the loops are flipped with probability 0.5, not with certainty anymore. This means, one needs to include that at all points in the program, at which spins get flipped.
- At the end of each loop DO NOT call the function *clean_up_events* (which deletes all events where 2 consecutive events had the same spin)! The loop structure has the physical meaning of correlated regions in the space-time lattice, even if the event does not cause the worldline to jump. Instead, leave the cluster structure the way it is, construct all of the loops, perform measurements, and before starting another sweep take a look at all events, and delete them if two consecutive events have the same spin. Doing this corresponds to: Given a spin configuration choose the loops with suitable probability. Forget about the spins and choose for every loop one of the two allowed spin orientations. This results in a new spin configuration and one can forget about the loops again (depending on the observable forgetting can be done before or after the measurement).

Bibliography

[1] H. G. Evertz. The loop algorithm. *Advances in Physics*, 52(1):1–66, 2003. .

[2] Fabien Alet, Sylvain Capponi, Nicolas Laflorencie, and Matthieu Mambrini. Valence Bond Entanglement Entropy. *Physical Review Letters*, 99(11):117204, 2007. .

[3] Yu-Cheng Lin and Anders W. Sandvik. Definitions of entanglement entropy of spin systems in the valence-bond basis. *Phys. Rev. B*, 82:224414, 2010.

[4] D. Baeriswyl, D.K Campbell, J.M.P. Carmelo, F. Guinea, and E. Louis. *The Hubbard Model.* Nato Science Series B:, Vol. 343. Springer, 1995.

[5] Bill Sutherland. *Beautiful Models.* World Scientific Publishing Co. Pte. Ltd., 2004.

[6] Luigi Amico and Rosario Fazio. Entanglement and magnetic order. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504001, 2009. .

[7] P. Calabrese and J. Cardy. Entanglement entropy and conformal field theory. *Journal of Physics A Mathematical General*, 42:4005, 2009. .

[8] Anders W. Sandvik. Stochastic series expansion method with operator-loop update. *Phys. Rev. B*, 59:R14157–R14160, 1999. .

[9] Anders W. Sandvik. Ground State Projection of Quantum Spin Systems in the Valence-Bond Basis. *Phys. Rev. Lett.*, 95:207203, 2005. .

[10] Anders W. Sandvik and Hans Gerd Evertz. Loop updates for variational and projector quantum Monte Carlo simulations in the valence-bond basis. *Phys. Rev. B*, 82:024407, 2010. .

[11] N. S. Blunt, T. W. Rogers, J. S. Spencer, and W. M. C. Foulkes. Density-matrix quantum monte carlo method. *Phys. Rev. B*, 89:245124, 2014.

[12] L.-A. Wu, M. Sarandy, and D. Lidar. Quantum phase transitions and bipartite entanglement. *Phys. Rev. Lett.*, 93:250404, 2004. .

[13] A. Anfossi, P. Giorda, A. Montorsi, and F. Traversa. Two-Point Versus Multipartite Entanglement in Quantum Phase Transitions. *Physical Review Letters*, 95(5):056402, 2005.

[14] Roger G. Melko, Ann B. Kallin, and Matthew B. Hastings. Finite-size scaling of mutual information in Monte Carlo simulations: Application to the spin-$\frac{1}{2}$ XXZ model. *Phys. Rev. B*, 82:100409, 2010. .

[15] Matthew B. Hastings, Ivan Gonzalez, Ann B. Kallin, and Roger G. Melko. Measuring Renyi Entanglement Entropy in Quantum Monte Carlo Simulations. *Physical Review Letters*, 104(15):157201, 2010. .

[16] C. Holzhey, F. Larsen, and F. Wilczek. Geometric and renormalized entropy in conformal field theory. *Nuclear Physics B*, 424:443–467, 1994. .

[17] J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277, 2010. .

[18] Ann B. Kallin, Ivan Gonzalez, Matthew B. Hastings, and Roger G. Melko. Valence Bond and von Neumann Entanglement Entropy in Heisenberg Ladders. *Physical Review Letters*, 103(11):117203, 2009. .

[19] SYLVAIN CAPPONI, FABIEN ALET, and MATTHIEU MAMBRINI. Entanglement of quantum spin systems: A valence-bond approach. *Modern Physics Letters B*, 25(12n13):917–928, 2011. .

[20] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Reviews of Modern Physics*, 81:865–942, 2009. .

[21] A. Einstein, B. Podolsky, and N. Rosen. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Phys. Rev.*, 47:777–780, 1935. .

[22] J. S. Bell. On the Einstein Podolsky Rosen Paradox. *Physics*, 1(3):195–200, 1964.

[23] Alain Aspect, Jean Dalibard, and Gérard Roger. Experimental test of bell's inequalities using time- varying analyzers. *Phys. Rev. Lett.*, 49:1804–1807, 1982. .

[24] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, 1969.

[25] R. Horodecki and P. Horodecki. Quantum redundancies and local realism. *Physics Letters A*, 194(3):147–152, 1994. .

[26] E. Schroedinger. The current situation in quantum mechanics. *NATURWISSENSCHAFTEN*, 23:807–812, 1935.

[27] Martin B. Plbnio and Shashank Virmani. An introduction to entanglement measures. *Quantum Info. Comput.*, 7(1):1–51, 2007. .

[28] William K. Wootters. Entanglement of Formation and Concurrence. *Quantum Information and Computation*, 1(1):27–44, 2001.

[29] Armin Uhlmann. Entropy and optimal decompositions of states relative to a maximal commutative subalgebra. *Open Systems & Information Dynamics*, 5(3), 1998. .

[30] Koenraad Audenaert, Frank Verstraete, and Bart De Moor. Variational characterizations of separability and entanglement of formation. *Phys. Rev. A*, 64:052304, 2001.

[31] William K. Wootters. Entanglement of formation of an arbitrary state of two qubits. *Phys. Rev. Lett.*, 80:2245–2248, 1998. .

[32] E. Faizi and H. Eftekhari. Investigation of quantum correlations for a s = 1/2 ising–heisenberg model on a symmetrical diamond chain. *Reports on Mathematical Physics*, 74(2):251 – 264, 2014. .

[33] G. Vidal and R. F. Werner. A computable measure of entanglement. *Phys. Rev. A*, 65:032314, 2002. .

[34] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Mixed-State Entanglement and Distillation: Is there a "Bound" Entanglement in Nature? *Phys. Rev. Lett.*, 80:5239–5242, 1998. .

[35] Charles H. Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.*, 76:722–725, 1996. .

[36] Chia-Min Chung, Vincenzo Alba, Lars Bonnes, Pochung Chen, and Andreas M. Läuchli. Entanglement negativity via the replica trick: A quantum monte carlo approach. *Phys. Rev. B*, 90:064401, 2014. .

[37] Hannu Wichterich, Julien Vidal, and Sougato Bose. Universality of the negativity in the lipkin-meshkov-glick model. *Phys. Rev. A*, 81:032311, 2010. .

[38] Vincenzo Alba. Entanglement negativity and conformal field theory: a monte carlo study. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(05):P05013, 2013. .

[39] Dik Bouwmeester, Jian-Wei Pan, Klaus Mattle, Manfred Eibl, Harald Weinfurter, and Anton Zeilinger. Experimental quantum teleportation. *Nature*, 390(6660):575–579, December 1997. .

[40] Kavan Modi, Tomasz Paterek, Wonmin Son, Vlatko Vedral, and Mark Williamson. Unified View of Quantum and Classical Correlations. *Physical Review Letters*, 104(8), 2010. .

[41] Harold Ollivier and Wojciech H. Zurek. Quantum discord: A measure of the quantumness of correlations. *Phys. Rev. Lett.*, 88:017901, 2001.

[42] V. Vedral. The role of relative entropy in quantum information theory. *Rev. Mod. Phys.*, 74:197–234, 2002. .

[43] M. Suzuki. Relationship between d-Dimensional Quantal Spin Systems and (d+1)-Dimensional Ising Systems: Equivalence, Critical Exponents and Systematic Approximants of the Partition Function and Spin Correlations. *Prog. Theor. Phys.*, 56:1454, 1976.

[44] B. B. Beard and U.-J. Wiese. Simulations of Discrete Quantum Systems in Continuous Euclidean Time. *Phys. Rev. Lett.*, 77:5130–5133, 1996. .

[45] M. Aizenman and B. Nachtergaele. Geometric aspects of quantum spin states. *Communications in Mathematical Physics*, 164, 1994.

[46] R. Brower, S. Chandrasekharan, and U.-J. Wiese. Green's functions from quantum cluster algorithms. *Physica A Statistical Mechanics and its Applications*, 261:520–533, 1998. .

[47] N. V. Prokof'Ev, B. V. Svistunov, and I. S. Tupitsyn. Exact, complete, and universal continuous-time worldline Monte Carlo approach to the statistics of discrete quantum systems. *Soviet Journal of Experimental and Theoretical Physics*, 87:310–321, 1998.

[48] Ravindra W. Chhajlany, Piotr Tomczak, and Antoni Wojcik. Topological Estimator of Block Entanglement for Heisenberg Antiferromagnets. *Physical Review Letters*, 99(16), 2007. .

[49] Fabien Alet, Ian P. McCulloch, Sylvain Capponi, and Matthieu Mambrini. Valence-bond entanglement entropy of frustrated spin chains. *Phys. Rev. B*, 82, 2010. .

[50] J. L. Jacobsen and H. Saleur. Exact Valence Bond Entanglement Entropy and Probability Distribution in the XXX Spin Chain and the Potts Model. *Phys. Rev. Lett.*, 100:087205, 2008. .

[51] A. Osterloh, Luigi Amico, G. Falci, and Rosario Fazio. Scaling of entanglement close to a quantum phase transition. *Nature*, 416(6881):608–610, 2002. .

[52] B. Bauer, L. D. Carr, H. G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, and S. Wessel. The ALPS project release 2.0: open source software for strongly correlated systems. *Journal of Statistical Mechanics: Theory and Experiment*, (05):P05001, 2011. .

[53] Synge Todo and Kiyoshi Kato. Cluster Algorithms for General- S Quantum Spin Systems. *Phys. Rev. Lett.*, 87:047203, 2001. .