Computing the Ground-State Energy of the Ising Model in a Transverse Field with Matrix Product States

by

Pim Vree

to obtain the double degree in Bachelor of Physics and Mathematics
at the Delft University of Technology

Student number:	4551125	
Project duration:	September 1, 2018 – July 19, 2019	
Thesis committee:	Dr. J. Thijssen,	TU Delft, Physics supervisor
	Prof. dr. ir. C. Vuik,	TU Delft, Mathematics supervisor
	Dr. M. Blaauboer,	TU Delft
	Dr. M. P. T. Caspers,	TU Delft

TU Delft

Acknowledgements

I would like to express my very great appreciation to Jos Thijssen. Our weekly meetings have not only provided me with great insight and help on the subject of this thesis, but it has given me a whole new perspective of physics in general. Furthermore the aid he provided in writing the eventual code for this thesis can not be understated and it is safe to say I could not have finished the project without Jos.

My special thanks also extends to Kees Vuik for co-supervising this thesis on the mathematical side and for all the help he provided during the year. I would also like to thank Miriam Blaauboer and Martijn Caspers for taking their time to read and evaluate this thesis.

Finally my friends and family who have stood by me, not only for the duration of this thesis but my entire life, I want say how grateful I am that I have you all in my life. To all of you, thank you.

Pim Vree
Delft, July 2019

Contents

1 Introduction

Physics and mathematics are both studies in which we search for answers. The scientist in these fields are driven to find an explanation or proof for every problem they encounter. With the goal of gaining an exact understanding on the problem at hand. Especially in mathematics a lot of time and energy is spend on rigorously proving everything that falls within its domain. Unfortunately not all the problems have a definite answer or solution. Sometimes we can show that these answers do not exists, but more often we are just completely left in the dark.

However there is hope. With the invention of the computer we are now able to answer most questions numerically. Even tough an exact answer is not known we can still understand a lot about a system with the numerical result. The effective implementation of these problems in the computer has become a whole subfield of its own in physics and mathematics. Since, generally the computer needs a lot of help to find the correct solutions.

With the discovery of quantum mechanics,at the turn of the last century, scientist immediately put all their effort in solving the problems that arose in this field. The study of the quantum world has been an on-going effort of millions of scientist ever since. Although a lot of progress has been made on this subject we are still far away of a complete understanding of this topic. The difficulty in quantum mechanics is that even numerical results are hard to obtain. This is due to entanglement whereby the complexity of the system grows exponential with the amount of particles in that system. Our normal computers are incapable of handling these problems effectively and so other approaches need to be sought. Currently a lot of time and resources are being spend on developing the quantum computer which is said to solve this problem. But for the moment we need to use algorithms that can work with the classical computer.

The algorithms that are used at the present are able to numerically solve one dimensional Hamiltonians if the system is low-entangled. The Density Matrix Renormalization group (DMRG) first developed by Steven R. White [19] is one of the most efficient algorithms that is used over the past 20 years. However, following the paper of S.Rommer and S.Ostlond [11], the new method of Matrix Product States is one of the most promising algorithms for solving one dimensional quantum system numerically.
In this thesis we will use Matrix Product States (MPS) to solve systems involving many quantum particles. To this extend we use a version of the TEBD algorithm (Time-Evolving Block Decimation) that exploits the form of Matrix Product States. This method was first developed by G.Vidal [17] in order to compute the quantum properties for an infinite chain in one dimension. In the thesis the idea of MPS is explained and used in the TEBD algorithm, with the goal that we can stimulate the (imaginary) time evolution of a chain consisting of spin particles. To validate that the results of this algorithm are indeed correct the quantum Ising chain has been chosen as a test model.

The Ising model describes spin systems that only interact with other spin systems that are in its vicinity. In this thesis the restriction is put that the particles only interact with its nearest-neighbours. Furthermore, a magnetic field perpendicular to the interaction is applied to the spin particles as well, this is done to make sure that the Hamiltonian is gapped-the reason that this is necessary will be explained in the thesis. The Ising chain has been chosen as the trial system due to the fact that the Hamiltonian can be solved analytically [9]. Furthermore, the model falls in the class of Hamiltonians that can be analysed with MPS.

The thesis has the following structure: In chapter 1 the introduction to problem of the thesis is given. Chapter 2 is about solving the test system of the Ising model in a Transverse field analytically. The representation of Matrix Product States is introduced in chapter 3, together with the explanation of why this representation is so successful for using in the TEBD algorithm. In chapter 4 the TEBD algorithm is described and in chapter 5 the results of the algorithm are presented and discussed. Finally, in chapter 6 we give a short conclusion and a few proposals for further research.

2

Exact Solution of the Ising Model in a Transverse Field

2.1. The Model

In order to validate algorithms based on matrix Product States, we use the model of the quantum Ising chain in a transverse field, of which the ground state can be found analytically [9]. This model was first used by Ernst Ising to analyze phase transitions in 1924 [6]. The Ising model describes spin-$\frac{1}{2}$ systems, where the particles can be either in a spin up state or in a spin down state. It is possible for these spins to interact with each other. In our case we restrict ourselves to only nearest-neighbours interactions. This local character of the interactions is vital to the success of matrix product states as will be explained in the next chapter.
We use the formulation of Pfeuty [9], where the magnetic field is applied in the z-direction and the nearest-neighbour interaction of the spin operator is in the x-direction,

$$H = -J\sum_i S_i^x S_{i+1}^x - \Gamma \sum_i S_i^z. \tag{2.1}$$

The variable J represents the interaction strength between neighbouring spins and the variable Γ is the magnetic field strength. A uniform magnetic field is assumed and the interaction between each spin site has the same strength. In the Hamiltonian the spin angular momentum operators are used, for these operators the fundamental commutation relation hold: $[S_i^x, S_i^y] = i\hbar S_i^z$. To make further calculations easier, $\hbar$ is set to be one from of now on.

Hamiltonian (2.1) describes a finite chain of length N. There are two different configurations for this chain, either open boundary conditions with a starting point and an end point at which the spin-$\frac{1}{2}$ particles have only one neighbour, or a periodic chain which has the topology of a circle. The boundary conditions influence the summation of the S^x operators; for the open chain the sum runs from 1 to $N-1$ and for the periodic chain from 1 to N, where we put $S_{N+1}^x = S_1^x$. We will first work out the problem for periodic boundary conditions.

The goal for this chapter is to obtain the eigenstates and the corresponding eigenenergies of Hamiltonian (2.1) analytically. This is done by getting equation the Hamiltonian in a diagonal form, to achieve this we first have to perform a couple of transformations on the set of operators. The approach we take here is from Schultz et al. [12].

2.2. The Raising and Lowering Operators

The first transformation is from the Cartesian spin operators S^x, S^y and S^z, to the raising and lowering spin operators. This pair of operators is defined at each site i as:

$$a_i^+ = S_i^x + \mathrm{i} S_i^y; \tag{2.2a}$$

$$a_i = S_i^x - \mathrm{i} S_i^y. \tag{2.2b}$$

To transform equation (2.1), the inverse relations of this transformation need to be known, so we will first determine these equations. Clearly we have that $S_i^x = \frac{a_i^+ + a_i}{2}$. However for the inverse relation of the operator S_i^z, one needs to perform a little more work. We use the well-known eigenvalues of the spin operators (see Griffiths, page 171) [4]: $S^2|sm\rangle = \hbar^2 s(s+1)|sm\rangle$ and $S^z|sm\rangle = \hbar m|sm\rangle$. In this case $s = \frac{1}{2}$ and $\hbar = 1$, thus $S^2 = \frac{1}{2}(\frac{1}{2}+1) = \frac{3}{4}$. For the spin operator in the z direction, we have $m = \frac{1}{2}$ or $-\frac{1}{2}$, which are commonly identified as the spin-up and spin-down state respectively. However if the S^z operator is applied twice, this sign difference disappears and one just gets $S^z S^z = \frac{1}{4}$. This can be used to derive the inverse relation of S^z if one uses the fact that $S^2 = (S^x)^2 + (S^y)^2 + (S^z)^2$:

$$\begin{aligned} a^+ a &= (S^x + \mathrm{i}S^y)(S^x - \mathrm{i}S^y) = S^x S^x + S^y S^y - \mathrm{i}S^x S^y + \mathrm{i}S^y S^x \\ &= S^2 - S^z S^z - i[S^x, S^y] = \frac{3}{4} - \frac{1}{4} + S^z = S^z + \frac{1}{2}. \end{aligned}$$

Where in the last line the fundamental commutation relation is used. With the inverse transformation known, we can express Hamiltonian (2.1) in terms of the raising and lowering operator,

$$H = -\frac{J}{4}\sum_i [a_i^+ a_{i+1}^+ + a_i^+ a_{i+1} + a_i a_{i+1}^+ + a_i a_{i+1}] - \Gamma \sum_i [a_i^+ a_i - \frac{1}{2}]. \tag{2.3}$$

The new formulation is not in the diagonal form, to still make progress we need to perform another transformation. The idea of the following transformation is to get the Hamiltonian in the second quantization formalism, meaning that we need to obtain fermionic operators. To understand where we are going it is useful to know what kind of commutation relations hold for the raising and lowering operators. It is quite easy to compute these and we will just state them here without proof. The raising and lowering operators are a mix of Fermi and Boson operators:

$$i \neq j \qquad [a_i^+, a_j] = [a_i^+, a_j^+] = [a_i, a_j] = 0; \tag{2.4a}$$

$$\{a_i, a_i^+\} = 1; \tag{2.4b}$$

$$(a_i^+)^2 = (a_i)^2 = 0. \tag{2.4c}$$

2.3. The Jordan-Wigner Transformation

As one can see from equations (2.4a)-(2.4c) the raising and lowering operators are not Fermion operators, we would like to work with Fermion operators instead because they allow for finding the eigenvalues and -states. The transformation that maps the spin raising and lowering operators to a set of Fermion operators was first proposed by Jordan and Wigner. The transformation is just mapping the old raising and lowering operators to the new operators by the identity transformation and in some cases adding a sign shift. This sign shift depends on the amount of occupied states to the left of the chain:

$$c_i = e^{\pi \mathrm{i} \sum_{j=1}^{i-1} a_j^+ a_j} a_i; \tag{2.5a}$$

$$c_i^+ = a_i^+ e^{-\pi \mathrm{i} \sum_{j=1}^{i-1} a_j^+ a_j}. \tag{2.5b}$$

These operators are referred to as the annihilation and creation operators respectively and are again defined per site i. In the exponent we have sum of operators, it is vital to have a sufficient understanding on how the handle an operator of this form, since such operators will be used a lot throughout this thesis. For that purpose we introduce the following theorem

Theorem 2.1: *If two operators A and B commute the following decomposition can be done*

$$e^{t(A+B)} = e^{tA} e^{tB}. \tag{2.6}$$

Proof: To see why this is the case, we look at the following function $f(t) = e^{t(A+B)} e^{-tA} e^{-tB}$ and differentiate this function with respect to t:

$$\begin{aligned} \frac{\mathrm{d}f(t)}{\mathrm{d}t} = (A+B)e^{t(A+B)}e^{-tA}e^{-tB} &+ e^{t(A+B)}(-A)e^{-tA}e^{-tB} \\ &+ e^{t(A+B)}e^{-tA}(-B)e^{-tB}. \end{aligned} \tag{2.7}$$

If the exponent $e^{t(A+B)}$ commutes with the operators A and B, we have that the last two terms cancel the first term and the derivative vanishes for all t. For this purpose we will show that the these operators commute, using the definition of an exponential operator. Now, since $[A,B]=0$ and $[A,A]=0$ we have that $(A+B)^n A = A(A+B)^n$ for all $n \in \mathbb{N}$. With this result we have that $e^{(A+B)}A = \sum_n \frac{(A+B)^n}{n!} A = \sum_n \frac{(A+B)^n A}{n!} = \sum_n \frac{A(A+B)^n}{n!} = Ae^{(A+B)}$. Thus the operators A and $e^{t(A+B)}$ commute with each other. Analogously it can be shown that the operator B commutes with $e^{t(A+B)}$ and e^{-tA}. We can conclude that equation (2.7) is always zero. The function $f(t)$ is thus a constant function, calculating the value for this constant function can be done straightforwardly for $t=0$ and we find that $f(t=0)=I$.

■

With the help of theorem 2.1 it is quite simple to see that $c_i^+ c_i = a_i^+ e^{-\pi \mathrm{i} \sum_{j=1}^{i-1} a_j^+ a_j} e^{\pi \mathrm{i} \sum_{j=1}^{i-1} a_j^+ a_j} a_i = a_i^+ a_i$, since obviously an operator commutes with itself. This expression can be used to find the inverse Jordan-Wigner transformation:

$$a_i = e^{-\pi \mathrm{i} \sum_{j=1}^{i-1} c_j^+ c_j} c_i; \tag{2.8a}$$

$$a_i^+ = c_i^+ e^{\pi \mathrm{i} \sum_{j=1}^{i-1} c_j^+ c_j}. \tag{2.8b}$$

Before we continue with expressing Hamiltonian (2.3) in the creation and annihilation operators, we are first going to prove that these operators are indeed Fermi operators, otherwise all the hard work will be done for nothing. The prove itself will use the following Lemma.

Lemma 2.2: *The following equations hold for the raising and lowering operators*:

$$e^{\pi \mathrm{i} a_k^+ a_k} a_k = a_k; \tag{2.9a}$$

$$a_k^+ e^{\pi \mathrm{i} a_k^+ a_k} = a_k^+. \tag{2.9b}$$

Proof: The Lemma can easily be proven, when one considers the definition of an operator in the exponent:

$$e^{\pi \mathrm{i} a_k^+ a_k} a_k = \sum_{n=0}^{\infty} \frac{(\pi \mathrm{i} a_k^+ a_k)^n}{n!} a_k = a_k.$$

For every $n \neq 0$ the term becomes zero due to equation (2.4c). The second part of the lemma can be proven in exactly the same way.

■

Another equation that is of importance in the proof is: $[e^{\pi \mathrm{i} a_k^+ a_k}, a_j] = 0$ if $k \neq j$, this result follows from the fact that the raising and lowering operators commute with each other if they are not working on the same site. We are now ready to prove that the creation and annihilation operators are Fermion operators:

Theorem 2.3: *The creation and annihilation operators are Fermion operators where the following anti-commutation relations for hold*:

$$\left\{c_i, c_j^+\right\} = \delta_{ij}; \qquad \left\{c_i^+, c_j^+\right\} = 0; \qquad \left\{c_i, c_j\right\} = 0. \tag{2.10}$$

Proof: The first of these anti-commutation relations will be proven with the help of the following identity: $a_k^+ a_k a_k^+ = a_k^+(1 - a_k^+ a_k) = a_k^+$ here equations (2.4b) and (2.4c) have been used. Now suppose $i > j$ with the proof for $i < j$ being similar. In this case we have:

$$\left\{c_i, c_j^+\right\} = e^{\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} a_i a_j^+ e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} + a_j^+ e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} e^{\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} a_i.$$

We can interchanging the exponents with the raising and lowering operator in the first term, since all operators act on different sites. We bring the exponents together as well and we note that a lot of terms in the sum cancel each other,this all results in

$$\left\{c_i, c_j^+\right\} = a_i e^{\pi \mathrm{i} \sum_{k=j}^{i-1} a_k^+ a_k} a_j^+ + a_j^+ e^{\pi \mathrm{i} \sum_{k=j}^{i-1} a_k^+ a_k} a_i = e^{\pi \mathrm{i} \sum_{k=j}^{i-1} a_k^+ a_k} a_i a_j^+ + (a_j^+ e^{\pi \mathrm{i} a_j^+ a_j}) e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} a_i.$$

In the last term we separated the only term in the exponent that does not commute with a_j^+. For this term we apply lemma 2.2:

$$\left\{c_i, c_j^+\right\} = e^{\pi \mathrm{i} \sum_{k=j}^{i-1} a_k^+ a_k} a_j^+ a_i + a_j^+ e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} a_i = e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} (e^{\pi \mathrm{i} a_j^+ a_j} a_j^+) a_i + e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} a_j^+ a_i.$$

In the first term on the right we separated the exponent that was just canceled. Using the definition of the exponent we can further simplify the expression

$$\left\{c_i, c_j^+\right\} = e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} [\sum_{n=0}^{\infty} \frac{(\pi \mathrm{i} a_j^+ a_j)^n}{n!} a_j^+ a_i + a_j^+ a_i],$$

as stated before $a_j^+ a_j a_j^+ = a_j^+$, with this we obtain

$$\begin{aligned}\left\{c_i, c_j^+\right\} &= e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} [\sum_{n=0}^{\infty} \frac{(\pi \mathrm{i})^n}{n!} a_j^+ a_i + a_j^+ a_i] = e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} [e^{\pi \mathrm{i}} a_j^+ a_i + a_j^+ a_i] \\ &= e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} [-a_j^+ a_i + a_j^+ a_i] = e^{\pi \mathrm{i} \sum_{k=j+1}^{i-1} a_k^+ a_k} [a_j^+, a_i] = 0.\end{aligned}$$

Where the last step follows from equation (2.4a). Now the only thing that rest us to do, is to show that $\left\{c_j, c_j^+\right\} = 1$. Now if $i = j$ the sum in the exponents will just cancel each other and we are left with $\{c_i, c_i^+\} = \{a_i, a_i^+\} = 1$. Thus the first anti-commutation relation has been proven.
Now we will show that $\left\{c_i^+, c_j^+\right\} = 0$, here we again assume that $i > j$ with the proof of the reverse inequality being extremely similar and in the case that $i = j$, the proof is trivial. Calculating the anti-commutation gives

$$\left\{c_i^+, c_j^+\right\} = a_i^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} a_j^+ e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} + a_j^+ e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} a_i^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k}.$$

It is again possible to interchange the exponents and the operators that are acting on different sites;

$$\begin{aligned}\left\{c_i^+, c_j^+\right\} &= e^{-\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} a_i^+ a_j^+ + e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} a_j^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} a_i^+ \\ &= e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1, k\neq j}^{i-1} a_k^+ a_k} e^{-\pi \mathrm{i} a_j^+ a_j} a_j^+ a_i^+ + e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1, k\neq j}^{i-1} a_k^+ a_k} a_j^+ e^{-\pi \mathrm{i} a_j^+ a_j} a_i^+.\end{aligned}$$

Using lemma 2.2 and the definition of the exponent we get:

$$\begin{aligned}\left\{c_i^+, c_j^+\right\} &= e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1, k\neq j}^{i-1} a_k^+ a_k} [\sum_{n=0}^{\infty} \frac{(-\pi \mathrm{i} a_j^+ a_j)^n}{n!} a_j^+ a_i^+ + a_j^+ a_i^+] \\ &= e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1, k\neq j}^{i-1} a_k^+ a_k} [\sum_{n=0}^{\infty} \frac{(-\pi \mathrm{i})^n}{n!} a_j^+ a_i^+ + a_j^+ a_i^+] \\ &= e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1, k\neq j}^{i-1} a_k^+ a_k} [-a_j^+ a_i^+ + a_j^+ a_i^+] = e^{-\pi \mathrm{i} \sum_{k=1}^{j-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1, k\neq j}^{i-1} a_k^+ a_k} [a_j^+, a_i^+] = 0.\end{aligned}$$

Where the last step follows from (2.4a). In the second line above we again made use of the fact that $a_j^+ a_j a_j^+ = a_j^+$. The proof for $\{c_i, c_j\} = 0$ is similar to the prove above of $\left\{c_i^+, c_j^+\right\} = 0$, so we will not discuss it here in detail.

■

We have shown that the creation and annihilation operators of the Jordan-Wigner transformation are indeed Fermion operators. In order to facilitate the transformations of Hamiltonian (2.3) into the form of the creation and annihilation operators, we will first need a few more statements relating these operators with the raising and lowering operators. These formulas are shown and proven in the following theorem.

Theorem 2.4: For $i = 1, 2, ..., N-1$ the following expressions hold:

$$a_i^+ a_i = c_i^+ c_i; \tag{2.11a}$$
$$a_i^+ a_{i+1}^+ = c_i^+ c_{i+1}^+; \tag{2.11b}$$
$$a_i^+ a_{i+1} = c_i^+ c_{i+1}; \tag{2.11c}$$
$$a_i a_{i+1}^+ = -c_i c_{i+1}^+; \tag{2.11d}$$
$$a_i a_{i+1} = -c_i c_{i+1}. \tag{2.11e}$$

Proof: Equation (2.11a) was already shown to be true and is just repeated for completeness, note that in this case it does hold for $i = N$ as well. Now the sequence of the operators $a_i^+ a_i$ can only have the value one or zero (and from (2.11a) $c_i^+ c_i$ has the same property!). Therefore we have that $e^{2\pi \mathrm{i} a_k^+ a_k} = 1$, this property will assist us in proving the other relations. We will first show that equation (2.11b) is valid:

$$c_i^+ c_{i+1}^+ = a_i^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} a_{i+1}^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i} a_k^+ a_k},$$

now we move a_{i+1}^+ to the left yielding,

$$c_i^+ c_{i+1}^+ = a_i^+ a_{i+1}^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} e^{-\pi \mathrm{i} \sum_{k=1}^{i} a_k^+ a_k} = a_i^+ a_{i+1}^+ e^{-\pi \mathrm{i} a_i^+ a_i} e^{-2\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k},$$

here we can apply lemma 2.2 to obtain:

$$c_i^+ c_{i+1}^+ = a_{i+1}^+ (a_i^+ e^{-\pi \mathrm{i} a_i^+ a_i}) = a_i^+ a_{i+1}^+.$$

What we have just done is certainly not true for $i = N$, since in that case $a_{i+1}^+ = a_1^+$ due to the periodic boundary conditions. This means that the operator a_{i+1}^+ does not commute with one of the terms in the exponent and as a result can not be interchanged. This makes the first step invalid. This is also the reason why the relations (2.11c)-(2.11e) are incorrect for $i = N$.
Formula (2.11c) can be shown to be correc with the use of lemma 2.2

$$c_i^+ c_{i+1} = a_i^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} e^{\pi \mathrm{i} \sum_{k=1}^{i} a_k^+ a_k} a_{i+1} = a_i^+ e^{\pi \mathrm{i} a_i^+ a_i} a_{i+1} = a_i^+ a_{i+1}.$$

To derive formula (2.11d) we again use the definition of the exponent and the fact that $a_i a_i^+ a_i = a_i$:

$$c_i c_{i+1}^+ = e^{\pi \mathrm{i} \sum_{k=1}^{i-1} a_k^+ a_k} a_i a_{i+1}^+ e^{-\pi \mathrm{i} \sum_{k=1}^{i} a_k^+ a_k} = a_i a_{i+1}^+ e^{-\pi \mathrm{i} a_i^+ a_i} = a_{i+1}^+ a_i \sum_{n=0}^{\infty} \frac{(-\pi \mathrm{i} a_i^+ a_i)^n}{n!}$$
$$c_i c_{i+1}^+ = a_{i+1}^+ a_i \sum_{n=0}^{\infty} \frac{(-\pi i)^n}{n!} = a_{i+1}^+ a_i e^{-\pi \mathrm{i}} = -a_i a_{i+1}^+.$$

In the same way that we have just shown equation (2.11d) to be true, we can also show that equation (2.11e) is correct.

■

With the use of theorem 2.4 one can express the Hamiltonian in the Jordan-Wigner operators. The only problem that still remains is that the previous results are not valid for $i = N$ and the terms containing an operator that acts on this site should be studied separately. However, this term is a result of the boundary alone and since we are mainly interested in a chain which has a large length we can ignore the influence of this correction term, furthermore this term is of order 1, while the other terms in the Hamiltonian will grow with linearly with order N and as a result can be safely neglected for large N. The final result of the Jordan-Wigner transformation is the Hamiltonian in the second quantization formalism

$$H = \frac{\Gamma N}{2} - \Gamma \sum_i c_i^+ c_i - \frac{J}{4} \sum_i [c_i^+ c_{i+1}^+ + c_i^+ c_{i+1} - c_i c_{i+1}^+ - c_i c_{i+1}]. \tag{2.12}$$

2.4. The Fourier Transformation

The objective is now to get Hamiltonian (2.12) in a diagonal form. The approach we take here is to transform the creation and annihilation operators into the momentum space by computing the discrete Fourier transformation. This can be done efficiently since we are working with periodic boundary conditions. Since, the inverse Fourier transformation is going to be used to rewrite Hamiltonian (2.12), we will state them here as well.

$$c_q = \frac{1}{\sqrt{N}}\sum_j c_j e^{\mathrm{i}qj}; \qquad (2.13a) \qquad\qquad c_j = \frac{1}{\sqrt{N}}\sum_q e^{-\mathrm{i}qj} c_q; \qquad (2.13c)$$

$$c_q = \frac{1}{\sqrt{N}}\sum_j c_j^+ e^{-\mathrm{i}qj}. \qquad (2.13b) \qquad\qquad c_j^+ = \frac{1}{\sqrt{N}}\sum_q e^{\mathrm{i}qj} c_q^+. \qquad (2.13d)$$

In this case the index q can achieve the following values:$q = \frac{2\pi n}{N}$ where $n = \frac{-(N-1)}{2}, \frac{-(N-1)}{2}, ..., \frac{N-1}{2}$. Observe that this defines a complete set of the new operators. It is also important to state that these new operators still follow the anti-commutation rules from theorem 2.3. This result can easily be proven from the definition of the Fourier transform and the fact that the creation and annihilation operators are Fermion operators. To recast the Hamiltonian in these new operators is pretty straightforward, since the inverse transformation is already known. We state here a result from complex analysis which simplifies the transformation: $\sum_{j=1}^N e^{\mathrm{i}(q_n-q_{n'})j} = N\delta_{nn'}$. We also define a new parameter λ, which helps to keep the notation clean in the remaining calculations. λ is defined as the ratio between the interaction strength of the spin particles and the magnetic field strength, i.e $\lambda = \frac{J}{2\Gamma}$. With all of this the new Hamiltonian becomes

$$\begin{aligned}\frac{H}{\Gamma} &= \frac{N}{2} - \sum_q c_q^+ c_q - \frac{\lambda}{2}\sum_q [e^{-\mathrm{i}q} c_q^+ c_{-q}^+ + e^{-\mathrm{i}q} c_q^+ c_q - e^{\mathrm{i}q} c_q c_q^+ - e^{\mathrm{i}q} c_q c_{-q}] \\ &= \frac{N}{2} - \sum_q (1+\lambda\cos q) c_q^+ c_q - \frac{\lambda}{2}\sum_q [e^{-\mathrm{i}q} c_q^+ c_{-q}^+ - e^{\mathrm{i}q} c_q c_{-q}].\end{aligned} \qquad (2.14)$$

The cosine in equation (2.14) can be derived by using the fact the c_q and c_q^+ from (2.13a) and (2.13b) are fermionic operators for which the anti-commutation relations hold. We also see immediately that the Hamiltonian scales with the interaction strength Γ. Now we are going to perform a trick that is also used in [14]. The idea is to split our Fourier operators into two sets, the ones with a negative index q into a set and a set containing the positive indices. The two sums in equation (2.14) will only be carried out over the positive set q. The q's with a negative sign will be incorporated by just writing them down. Since the index q is symmetric around the value zero this can be done quite nicely. Still careful consideration has to be given if the amount of spin particles is odd, cause then we have an operator $c_{q=0}$ which is not contained in either set. Here we will just put that term in an another operator H_0 to keep the notation clean. Now, doing all this we obtain

$$\begin{aligned}\frac{H}{\Gamma} &= \frac{N}{2} + H_0 - \sum_{q>0}(1+\lambda\cos q)[c_q^+ c_q + c_{-q}^+ c_{-q}] \\ &\quad - \frac{\lambda}{2}\sum_{q>0}[e^{-\mathrm{i}q} c_q^+ c_{-q}^+ + e^{\mathrm{i}q} c_{-q}^+ c_q^+ - e^{\mathrm{i}q} c_q c_{-q} - e^{-\mathrm{i}q} c_{-q} c_q] \\ &= \frac{N}{2} + H_0 - \sum_{q>0}(1+\lambda\cos q)[c_q^+ c_q + 1 - c_{-q} c_{-q}^+] \\ &\quad - \frac{\lambda}{2}\sum_{q>0}[e^{-\mathrm{i}q} c_q^+ c_{-q}^+ - e^{\mathrm{i}q} c_q^+ c_{-q}^+ + e^{\mathrm{i}q} c_{-q} c_q - e^{-\mathrm{i}q} c_{-q} c_q] \\ &= +H_0 - \sum_{q>0}(1+\lambda\cos q)[c_q^+ c_q - c_{-q} c_{-q}^+] \\ &\quad + \mathrm{i}\lambda\sum_{q>0}\sin q[c_q^+ c_{-q}^+ - c_{-q} c_q],\end{aligned} \qquad (2.15a)$$

$$H_0 = \begin{cases} 0 & \text{, If N is even} \\ -(1+\lambda)c_0^+ c_0 & \text{, If N is odd}\end{cases} \qquad (2.15b)$$

The constant $\frac{N}{2}$ is cancelled by the sum $\sum_{q>0} 1$, and the term$\sum_{q>0}\lambda\cos q$ vanishes the sum ranges symmetrically from 0 to π and the cosine is anti-symmetric. We can rewrite equation (2.15a) in a more insightful way, if we set all the terms inside the sum equal to an operator H_q, doing this we can write the Hamiltonian (2.15a) as $\frac{H}{\Gamma} = H_0 + \sum_{q>0} H_q$. With the new operator H_q defined as

$$H_q = -(1+\lambda\cos q)[c_q^+ c_q - c_{-q}c_{-q}^+] + \mathrm{i}\lambda\sin q[c_q^+ c_{-q}^+ - c_{-q}c_q] \tag{2.16a}$$

$$= \begin{pmatrix} c_q^+ & c_{-q} \end{pmatrix} \begin{pmatrix} -(1+\lambda\cos q) & \mathrm{i}\lambda\sin q \\ -\mathrm{i}\lambda\sin q & 1+\lambda\cos q \end{pmatrix} \begin{pmatrix} c_q \\ c_{-q}^+ \end{pmatrix}. \tag{2.16b}$$

The last line is the operator in matrix notation, here we immediately see that the operator is not yet in diagonal form, in the subsequent section we are going to explain the last transformation such that these off-diagonal terms vanishes.

2.5. The Bogoliubov-Valatin transformation

The purpose of the next transformation is to have the Hamiltonian in the following form

$$H = \Gamma \sum_k \Lambda_q \eta_q^+ \eta_q + C. \tag{2.17}$$

The transformation that accomplishes this feat was first formulated in 1958 by Bogoliubov and Valatin independently from each other [1] [16]. Hence, the Bogoliubov-Valatin transformation is named after both of them. The transformation itself is used a lot in the BCS derivation and in the theory of superconductivity. In our case we have that the old Fourier operators are mapped to the new operators by

$$\begin{pmatrix} \eta_q \\ \eta_q^+ \end{pmatrix} = \begin{pmatrix} u_q & \mathrm{i}v_q \\ \mathrm{i}v_q & u_q \end{pmatrix} \begin{pmatrix} c_q \\ c_{-q}^+ \end{pmatrix} \tag{2.18a}$$

$$\begin{pmatrix} c_q \\ c_q^+ \end{pmatrix} = \begin{pmatrix} u_q & -\mathrm{i}v_q \\ -\mathrm{i}v_q & u_q \end{pmatrix} \begin{pmatrix} \eta_q \\ \eta_{-q}^+ \end{pmatrix} \tag{2.18b}$$

The restriction is put that the variables u_q and v_q have to be real numbers. Furthermore, the operators η_q and η_q^+ are again Fermion operators with the corresponding anti-commutation relations:

$$\{\eta_k, \eta_l\} = 0, \qquad \{\eta_k^+, \eta_l^+\} = 0, \qquad \{\eta_k, \eta_l^+\} = \delta_{kl}. \tag{2.19}$$

The last equation of (2.19) can be used to gain extra information about the numbers u_q and v_q

$$\begin{aligned} 1 &= \left\{\eta_q, \eta_q^+\right\} = \eta_q\eta_q^+ + \eta_q^+\eta_q \\ &= (u_q c_q + \mathrm{i}v_q c_{-q}^+)(\mathrm{i}v_q c_q + u_q c_{-q}^+) + (\mathrm{i}v_q c_q + u_q c_{-q}^+)(u_q c_q + \mathrm{i}v_q c_{-q}^+) \\ &= u_q^2(c_q c_q^+ + c_q^+ c_q) + v_q^2(c_{-q}^+ c_{-q} + c_{-q}c_{-q}^+) + \mathrm{i}u_q v_q(c_{-q}^+ c_q^+ + c_q^+ c_{-q}^+ - c_q c_{-q} - c_{-q}c_q) \\ &= u_q^2\{c_q, c_q^+\} + v_q^2\{c_{-q}, c_{-q}^+\} + \mathrm{i}(\{c_{-q}^+, c_q^+\} - \{c_{-q}, c_q\}) = u_q^2 + v_q^2. \end{aligned}$$

This result can be used to calculate the inverse Bogoliubov-Valatin transformation, because this informs us that the determinant of the matrix form of (2.18a) is equal to 1 and with this the inverse relation can be easily computed. The equations for this inverse relation are given in matrix form in (2.18b), with these established we can cast equation (2.16b) in terms of the operators η_q and η_q^+

$$\begin{aligned} H_q = &\begin{pmatrix} \eta_q^+ & \eta_{-q} \end{pmatrix} \begin{pmatrix} u_q & \mathrm{i}v_q \\ \mathrm{i}v_q & u_q \end{pmatrix} \\ &\times \begin{pmatrix} -(1+\lambda\cos q) & \mathrm{i}\lambda\sin q \\ -\mathrm{i}\lambda\sin q & 1+\lambda\cos q \end{pmatrix} \begin{pmatrix} u_q & -\mathrm{i}v_q \\ -\mathrm{i}v_q & u_q \end{pmatrix} \begin{pmatrix} \eta_q \\ \eta_{-q}^+ \end{pmatrix}. \end{aligned} \tag{2.20}$$

It is possible to diagonalize equation (2.20), to do this we find the eigenvalues of the central matrix in (2.20), with the eigenvectors given by $\begin{pmatrix} u_q & -\mathrm{i}v_q \\ -\mathrm{i}v_q & u_q \end{pmatrix}$. Since it is a 2×2 matrix this is quite an easy task and we obtai

$$\Lambda_q^\pm = \pm\sqrt{1+\lambda^2+2\lambda\cos q}. \tag{2.21}$$

Where the Λ_q^+ corresponds with the first eigenvector and Λ_q^- with the second eigenvector. Now there is still some freedom in choosing the u_q and v_q, here we choose them in such a way that $u_q > 0$ for $0 < q < \pi$.

Plugging in these results in the matrix notation of equation (2.20) we obtain

$$\begin{aligned}
H_q &= \begin{pmatrix}\eta_q^+ & \eta_{-q}\end{pmatrix}\begin{pmatrix}u_q & \mathrm{i}v_q\\ \mathrm{i}v_q & u_q\end{pmatrix}\begin{pmatrix}\Lambda_q^+ u_q & \Lambda_q^-(-\mathrm{i}v_q)\\ \Lambda_q^+(-\mathrm{i}v_q) & \Lambda_q^- u_q\end{pmatrix}\begin{pmatrix}\eta_q\\ \eta_{-q}^+\end{pmatrix}\\
&= \begin{pmatrix}\eta_q^+ & \eta_{-q}\end{pmatrix}\begin{pmatrix}\Lambda_q^+(u_q^2+v_q^2) & 0\\ 0 & \Lambda_q^-(u_q^2+v_q^2)\end{pmatrix}\begin{pmatrix}\eta_q\\ \eta_{-q}^+\end{pmatrix}\\
&= \begin{pmatrix}\eta_q^+ & \eta_{-q}\end{pmatrix}\begin{pmatrix}\Lambda_q & 0\\ 0 & -\Lambda_q\end{pmatrix}\begin{pmatrix}\eta_q\\ \eta_{-q}^+\end{pmatrix}\\
&= \Lambda_q\eta_q^+\eta_q - \Lambda_q\eta_{-q}\eta_{-q}^+ = \Lambda_q\eta_q^+\eta_q - \Lambda_q(1-\eta_{-q}^+\eta_{-q})\\
&= \Lambda_q(\eta_q^+\eta_q + \eta_{-q}^+\eta_{-q} - 1).
\end{aligned} \tag{2.22}$$

Thus with the Bogoliubov-Valatin transformation we have successfully diagonalized the Hamiltonian. Now if we want to write out the total Hamiltonian it is useful to sum over all the q's. Careful consideration must be again given to $q = 0$. Fortunately for us we have that $\Lambda_0 = (1+\lambda)$, so it fits perfectly in the used notation of H_0. In the end we find the following diagonal Hamiltonian

$$\begin{aligned}
H &= \Gamma H_0 + \Gamma\sum_{q>0} H_q\\
&= \Gamma H_0 + \Gamma\sum_{q>0}\Lambda_q(\eta_q^+\eta_q + \eta_{-q}^+\eta_{-q} - 1)\\
&= \Gamma H_0 - \Gamma\sum_{q>0}\Lambda_q + \Gamma\sum_{q,q\neq 0}\Lambda_q\eta_q^+\eta_q\\
&= -\frac{\Gamma}{2}\sum_q\Lambda_q + \Gamma\sum_q\Lambda_q\eta_q^+\eta_q
\end{aligned} \tag{2.23}$$

and with this we have finished what we have set out to accomplish. We have transferred the Hamiltonian (2.1) of the Ising model in a transverse field to a diagonal form. For the attentive reader we will repeat the four different operator transformations we have done. We have gone from the Cartesian Spin operators to the raising and lowering operators $S_i^x, S_i^z \mapsto a_i, a_i+$. From these operator we went to the creation and annihilation operators with the Jordan-Wigner transformation $a_i, a_i^+ \mapsto c_i, c_i^+$. Because of the periodic boundary conditions it was fruitful to execute a Fourier Transform with which the site indices changed from i to q with $q = \frac{2\pi n}{N}$ and $n = \frac{-(N-1)}{2}, \frac{-(N-1)}{2}, \ldots, \frac{N-1}{2}$. The last transformation that was needed was to diagonalize the Hamiltonian was the Bogoliubov-Valatin transformation $c_q, c_q^+ \mapsto \eta_q, \eta_q^+$. Finally it was possible to diagonalize the found matrix with an easy eigenvalue problem. In the end the solution for the ground state energy for the Ising model in a transverse field with periodic boundary conditions is:

$$E_0 = -\frac{\Gamma}{2}\sum_{n=1}^{N}\sqrt{1+\lambda^2+2\lambda\cos\left(\frac{2\pi}{N}\left[\frac{-(N-1)}{2}+n\right]\right)} \tag{2.24}$$

with $\lambda = \frac{J}{2\Gamma}$. Note that this expression is only valid if N is large enough, because a cyclic term was neglected in the Jordan-Wigner transformation that has an order of $\mathcal{O}(\frac{1}{N})$.

2.6. Open Boundary Conditions

The result of equation (2.24) is valid when we have periodic boundary conditions for Hamiltonian (2.1). Now in the next chapter where Matrix Product States is explained, we shall see that these can only be developed for an open chain. Even more the periodicity of a circle introduces a non-locality in the Hamiltonian which reduces the efficiency of the eventual TEBD algorithm. For these reasons we will state here also the solution of Hamiltonian (2.1) with open boundary conditions. Finding the ground state energy for this problem goes along the same process as described here for the periodic boundary conditions. Only the Fourier transform does not work anymore and a different tactic needs to be employed. This case has been worked out in full by Schultz et al. Schultz et al. [12], we will just copy the results here. The ground state energy for OBC has the same excitation energy (2.21) as the case of PBC. The only difference is that the index q is now given by the roots of the following equation

$$\frac{\sin q(N+1)}{\sin qN} + \lambda = 0 \tag{2.25}$$

where $0 < q < \pi$. For $\lambda \leq 1$ there are N distinct roots in this interval. If $\lambda > 1$ then there are $N-1$ distinct roots and the last root that is needed is complex solution with a real part of π. The imaginary part $\Im(q) = a$ of this root is given by the solution of the following equation $\frac{\sinh a(N+1)}{\sinh a} = \lambda$. All of these roots have to be determined numerically. For large N it is quite a difficult task task to find N distinct roots and for that reason we will only use this solution for validating the algorithm in the region where N is small. For large N the solution of the periodic boundary condition is sufficient, since the boundary can be ignored.

3 Matrix Product States

3.1. Entanglement

For a general 1D quantum system with a local orthonormal basis denoted by $\{|\sigma_i\rangle\}$, where the basis has dimension d, we can write a random wave function in the following way:

$$|\psi\rangle = \sum_{\sigma_1,...,\sigma_N} c_{\sigma_1,...\sigma_N} |\sigma_1 ... \sigma_N\rangle, \tag{3.1}$$

here, we assume that we have a chain of length N. Now $c_{\sigma_1,...\sigma_N}$ is a tensor with dimensions d^N, thus the amount of information needed to store a random quantum state $|\psi\rangle$ grows exponentially with the length of the chain. Hence it is impossible to perform any computations with (3.1) for all but the smallest chains. Fortunately there is a way to rewrite $|\psi\rangle$ in a different way, called Matrix Product States (MPS) with which it is possible to perform some computations. Before we derive this MPS form, we will first explain to what kind of systems one can apply MPS. A key property in this context is the Von Neumann entropy, which is a measure of the entanglement of an arbitrary quantum state. This is defined as

$$S(\rho) = -\operatorname{Tr}(\rho \log_2 \rho). \tag{3.2}$$

The quantity ρ is the density matrix and for a pure state can be written down as $\rho = |\psi\rangle\langle\psi|$. The Von Neumann entropy for a random quantum state $|\psi\rangle$ scales linearly with the volume (N) [8], which is what we expect from the theory of statistical mechanics [15]. However, for gapped Hamiltonians with only a local character this scaling property is not always true. The quantum states near the ground-state of such Hamiltonians follow what we call an area law, meaning that the entropy scales linearly with the boundary of the system [3]. With the boundary we mean the cross section of the system. In one dimensions this boundary is off course constant, thus as a result we have that the entropy of low-excited states does not scale with the system size, but remains constant. Thus, when we search for the ground-states of low entangled Hamiltonians we can restrict ourselves to only the states that follow this area law. Therefore, we do not have to investigate the total Hilbert space of dimensions d^N, but only a sub domain of the Hilbert space. For large N, this sub domain is much smaller than the total Hilbert space. Matrix Product states provides a representation in which we can naturally restrict ourselves to only the quantum states that fall within this area law.

3.2. The Schmidt Decomposition

An efficient method to calculate the entanglement of a quantum state $|\psi\rangle$ with equation (3.2) is to perform a Schmidt decomposition. This decomposition is also used to transform equation (3.1) into the form of Matrix Product States. To describe the Schmidt decomposition, we start with a bipartite system for which the wave function can be written as

$$|\psi\rangle_{AB} = \sum_{ij} C_{ij} |i\rangle_A |j\rangle_B, \tag{3.3}$$

where $|i\rangle_A$ and $|j\rangle_B$ are orthonormal basis sets of system A and B respectively. Now the technique is to transform this bipartite system (3.3) to another basis with a singular value decomposition:

$$C = (C_{ij}) = U\Lambda V^\dagger = \sum_k \lambda_k |u_k \times v_k|, \tag{3.4}$$

where the λ_k are the singular values which are always positive and decreasing. The vectors u_k and v_k are orthonormal and are form again a basis for system A and B. The matrix elements of C can now be written as: $C_{ij} = \langle i|C|j\rangle = \sum_k \lambda_k \langle i|u_k\rangle\langle v_k|j\rangle$. If we plug this expression in to (3.3), we obtain

$$|\psi\rangle_{AB} = \sum_k \lambda_k (\sum_i \langle i|u_k\rangle |i\rangle_A)(\sum_j \langle v_k|j\rangle |j\rangle_B) = \sum_k \lambda_k |u_k\rangle_A |v_k^*\rangle_B. \tag{3.5}$$

The strength of the Schmidt decomposition is that quantum state $|\psi\rangle_{AB}$, which first contained a double sum over the indices i and j now only contains a single sum. Furthermore, it is possible to calculate the Von Neumman entropy (3.2) with equation (3.5) [7] directly

$$S(\rho) = -\sum_k \lambda_k^2 \log_2 \lambda_k^2. \tag{3.6}$$

Thus the singular values λ_k's are a measure for the entanglement of a bipartite system. Now, the low singular values of $|\psi\rangle$ contain almost no information about the quantum state $|\psi\rangle$ and can be ignored without resulting in a large error. In figure 3.1 we have plotted the singular values of two different bipartite states $|\psi\rangle_{AB}$. The first state is the ground state of the quantum Ising model in a transverse field where the analytic solution has been discussed at length in chapter 2. The Hamiltonian of the Ising model has only a local character. Therefore the ground state falls within the area law. From the figure we see that the singular values of this ground-state decreases exponentially to zero. As stated the vectors corresponding with the low singular values do not carry any important information and can therefore be ignored. This is the power of the MPS formalism, in essence we can introduce a cut-off index χ from which on all the singular values can be ignored. Vidal even proved that for all the quantum states that fall withing the area law, the corresponding singular values decrease exponential [18].It is important to stretch that this is only true for quantum states that fall within this area law. To see this we have also plotted the singular values for a random quantum state in figure 3.1. From the graph it is obvious that even for a large index these singular values still play a crucial role.

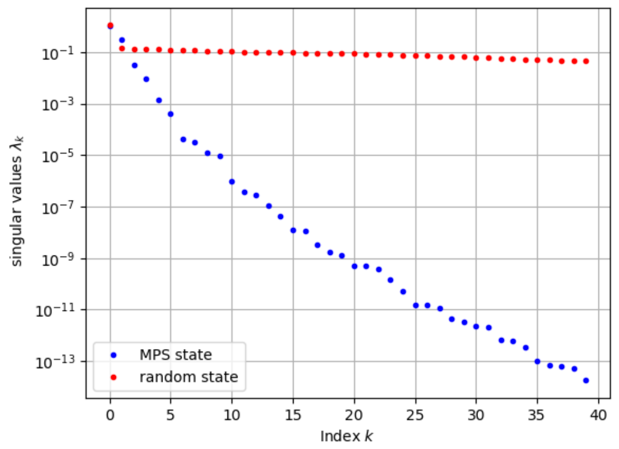


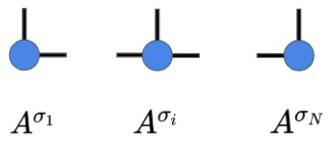
Figure 3.1: Evidence for the exponential decreasing of the singular values for a bipartite quantum state within the area law. Blue dots are the singular values of the ground state of the Ising model with $L = 14$ and $\lambda = 1$. Red dots are the singular values of a random quantum state with $L = 14$ sites.

3.3. Matrix Product States formalism

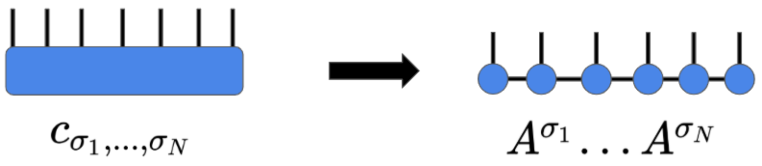
The MPS formalism is another way of writing down the tensor c in equation (3.1), now as a product of a set of matrices, in order to reduce the amount of information needed to store the quantum state $|\psi\rangle$. A general matrix product state can be written down as

$$|\psi\rangle = \sum_{\sigma_1,\dots,\sigma_N} A^{\sigma_1}\dots A^{\sigma_N} |\sigma_1 \dots \sigma_N\rangle. \tag{3.7}$$

The A^{σ_i}'s are matrices and we have that A^{σ_1} and A^{σ_N} are row and column vectors respectively. The matrices have the size of $\chi_i \times \chi_{i+1}$. These numbers are referred to as the bond dimensions. The equations used from now on are quite complex and the ideas behind the method we use can easily be lost. To make the steps easier to follow we introduce a schematic representation for the tensor $c_{\sigma_1,\ldots\sigma_N}$ and the matrices A^{σ_i}. These are shown in figure 3.2. The schematic representation used here is the same as used by Garnet Chan [2].



(a) Schematic representation for the row matrix A^{σ_1}, the matrix A^{σ_i} and the column matrix A^{σ_N} in order.


$c_{\sigma_1,\ldots,\sigma_N}$
$A^{\sigma_1}\ldots A^{\sigma_N}$

(b) Schematic representation for what happens to the tensor $c_{\sigma_1,\ldots\sigma_N}$ (left), when it is written down in the MPS formalism (right).

Figure 3.2: A schematic representation of the mathematical objects that are used in Matrix Product States. The amount of back legs refer to the number of dimensions. The vertical legs of each site represent the physical indices and the horizontal legs represent the bond indices.

In this diagram the blue circles are the tensors and the black legs represents a single index of the tensor. The vertical legs have the size of the local basis d and the horizontal legs are the bond dimensions of the corresponding matrix. In this representation it is also possible to visualize a tensor contraction, this occurs if two tensors are connected to each other by a black leg and the index we sum over in the contraction is the index of the leg. In 3.2b the schematic representation of equation (3.7) is visualized.
The Schmidt decomposition can be used to transform a random quantum state $|\psi\rangle$ in the form of MPS. The discussion of the previous section indicates that we start with a singular value decomposition. We are going to build the MPS form from left to right. To this end we start with the tensor $c_{\sigma_1,\ldots\sigma_N}$ and reshape it into a matrix with dimensions $d \times d^{N-1}$. On this new matrix we perform a SVD:

$$\begin{aligned} c_{\sigma_1,\ldots\sigma_N} = c_{\sigma_1,(\sigma_2,\ldots,\sigma_N)} &= \sum_{a_1} U_{a_1\sigma_1}\Lambda_{a_1a_1}(V^\dagger)_{a_1(\sigma_2,\ldots,\sigma_N)} \\ &= \sum_{a_1} A^{\sigma_1}_{a_1}\Lambda_{a_1a_1}(V^\dagger)_{a_1(\sigma_2,\ldots,\sigma_N)}. \end{aligned}$$

The matrix $U_{a_1\sigma_1}$ was reshaped into a set of d vectors with dimension a_1, so to give the correct dimensions for the MPS formalism at the left boundary site. The index a_1 can run from 1 to at most d. Now we can repeat this process by multiplying the matrix Λ with the row-orthonormal matrix $V^\dagger$ and reshaping the resulting tensor into another matrix on which another SVD can be calculated. The matrix we obtain from that computation is again reshaped and we get

$$c_{\sigma_1,\ldots\sigma_N} = \sum_{a_1,a_2} A^{\sigma_1}_{a_1}A^{\sigma_2}_{a_1a_2}\Lambda_{a_2a_2}(V^\dagger)_{a_2(\sigma_3,\ldots,\sigma_N)},$$

with the index a_2 running from 1 to at most d^2. Continuing the process in the same way we arrive at the form of equation (3.7). The bond dimensions χ_i of the matrices A^{σ_i} grow with at worst $d \times d$, $d \times d^2$, $d \times d^3$, ... thus the growth of information is still exponential and the MPS form is not really more efficient than (3.1). The technique to resolve this will be explained shortly. However, we first have to answer another complication that equation (3.7) gives. The matrices A^{σ_i} in the equation are not unique, to view why we insert a matrix B and its inverse B^{-1} between sites i and $i+1$

$$\begin{aligned}|\psi\rangle &= \sum_{\sigma_1,\ldots,\sigma_N} A^{\sigma_1}\ldots A^{\sigma_i}BB^{-1}A^{\sigma_{i+1}}\ldots A^{\sigma_N}|\sigma_1\ldots\sigma_N\rangle\\ &= \sum_{\sigma_1,\ldots,\sigma_N} A^{\sigma_1}\ldots \widetilde{A}^{\sigma_i}\widetilde{A}^{\sigma_{i+1}}\ldots A^{\sigma_N}|\sigma_1\ldots\sigma_N\rangle.\end{aligned}$$

The ambiguity of the MPS formalism can be eliminated by putting the matrices Λ_i from the singular value decomposition between the corresponding sites. We therefore put $\Lambda_i\Lambda_i^{-1}$ between site i and $i+1$ and rename the matrix product $\Lambda_i^{-1}A^{\sigma_{i+1}} = \Gamma_{i+1}$. The resulting quantum state we obtain is

$$\begin{aligned}|\psi\rangle &= \sum_{\sigma_1,\ldots,\sigma_N} A^{\sigma_1}\Lambda_1\Lambda_1^{-1}A^{\sigma_2}\ldots\Lambda_{N-1}\Lambda_{N-1}^{-1}A^{\sigma_N}|\sigma_1\ldots\sigma_N\rangle\\ &= \sum_{\sigma_1,\ldots,\sigma_N} \Gamma^{\sigma_1}\Lambda_1\Gamma^{\sigma_2}\ldots\Lambda_{N-1}\Gamma^{\sigma_N}|\sigma_1\ldots\sigma_N\rangle.\end{aligned}\tag{3.8}$$

The structure of equation (3.8) is known as the Canonical form of Matrix Product States and this representation is unique [2]. In figure 3.3 the schematic representation of the Canonical form is shown in the notation introduced before.

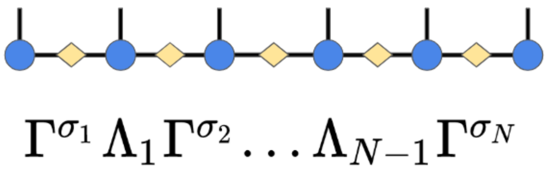

$\Gamma^{\sigma_1}\Lambda_1\Gamma^{\sigma_2}\ldots\Lambda_{N-1}\Gamma^{\sigma_N}$

Figure 3.3: The schematic representation for the canonical form of the Matrix Product States formalism.

So far, we have not yet reduced the information needed to store a quantum state $|\psi\rangle$. It is important to realize that the matrix dimensions grow from a size $1\times d$ at the left by a multiple of d in size towards the middle of the chain, after which they decrease towards the right, i.e the sites to the right of the first site are respectively $d\times d^2$,$d^2\times d^3$... in size.
All we did until this point is just a series of Schmidt decompositions. This can also be seen if we compute the product of all the matrices to the left of Λ_i in formula (3.8) and also all the products of the matrices to its right. Calculating the result gives us directly the Schmidt decomposition of a bipartite system $|\psi\rangle_{AB}$. Where system A contains all the sites until i and with system B containing all the sites from $i+1$ on-wards.

$$|\psi\rangle = \sum_{a_i}\Lambda_{a_ia_i}|\mu_{a_i}\rangle_A|\mu_{a_i}\rangle_B;\tag{3.9a}$$

$$|\mu_{a_i}\rangle_A = \sum_{\sigma_1,\ldots,\sigma_i}\sum_{a_1,\ldots,a_{i-1}}\Gamma^{\sigma_1}_{a_1}\Lambda_{a_1a_1}\Gamma^{\sigma_2}_{a_1a_2}\ldots\Lambda_{a_{i-1}a_{i-1}}\Gamma^{\sigma_i}_{a_{i-1}a_i}|\sigma_1\ldots\sigma_i\rangle;\tag{3.9b}$$

$$\begin{aligned}|\mu_{a_B}\rangle_B = \sum_{\sigma_{i+1},\ldots,\sigma_N}\sum_{a_{i+1},\ldots,a_N}&\Gamma^{\sigma_{i+1}}_{a_ia_{i+1}}\Lambda_{a_{i+1}a_{i+1}}\Gamma^{\sigma_{i+2}}_{a_{i+1}a_{i+2}}\ldots\Lambda_{a_{N-1}a_{N-1}}\Gamma^{\sigma_N}_{a_{N-1}}\\ &\times|\sigma_{i+1}\ldots\sigma_N\rangle.\end{aligned}\tag{3.9c}$$

Here $\Lambda_{a_ia_i}$ are the singular values at site i. Now in the previous section we found, that these singular values have a strong decreasing character and most of these values carry almost no information, if the state falls in the area law. What we do now to make equation (3.8) more efficient is to keep only the largest χ singular values and ignore the other values together with the corresponding Schmidt vectors. This reduction to a constant bond dimension for the matrices is the strength of MPS. It is now possible to work with a random quantum state $|\psi\rangle$, so long it is within the area law. The exponential growth of storing the chain is now gone and replaced by a polynomial growth of $\mathcal{O}(dN\chi^2)$.

3.4. Efficient Computations on Matrix Product States

In chapter 4 the TEBD algorithm will be explained. The TEBD algorithm is a method that uses the MPS form of equation (3.8). In the previous section we have concluded that it is possible to store a quantum state that follows the area law in a polynomial size. Now it is desired that all computations that we are going to perform with the TEBD algorithm are also polynomial in size, otherwise all the hard work we have just done has been for nothing. Therefore we will first explain how to compute the known operations from quantum mechanics on a state that is in the Canonical form om MPS in polynomial time.

Two-site operator

For the TEBD program we need to be able to compute the effect of a two-site operator on (3.8). To do this efficient we use the method described by Pollmann [10]. Let the operator be acting on site i and on site $i+1$. To calculate the effect of the operator $O_{i,i+1}$ on the quantum state we first compute all the matrix products to the left of Γ^{σ_i} and all the matrix products to the right of $\Gamma^{\sigma_{i+1}}$. This is the same process as in equation (3.9b) only now the two middle sites are not included in the calculation

$$\begin{aligned}|\psi\rangle &= \sum_{\sigma_i,\sigma_{i+1}} \sum_{a_{i-1},a_i,a_{i+1}} \Gamma^{\sigma_i}_{a_{i-1},a_i} \Lambda_{a_i,a_i} \Gamma^{\sigma_{i+1}}_{a_i,a_{i+1}} |\mu_{a_{i-1}}\rangle_L |\sigma_i \sigma_{i+1}\rangle |\mu_{a_{i+1}}\rangle_R \\ &= \sum_{\sigma_i,\sigma_{i+1}} \sum_{a_{i-1},a_{i+1}} \Theta^{\sigma_i,\sigma_{i+1}}_{a_{i-1},a_{i+1}} |\mu_{a_{i-1}}\rangle_L |\sigma_i \sigma_{i+1}\rangle |\mu_{a_{i+1}}\rangle_R.\end{aligned} \tag{3.10}$$

The tensor Θ contains the information of site i and $i+1$. Now we update this information with the operator $O_{i,i+1}$. The effect of this operator can be calulated efficiently with the help of the unit operator $\mathbb{1} = \sum_{\sigma'_i,\sigma'_{i+1}} |\sigma'_i\sigma'_{i+1}\rangle\langle\sigma'_i\sigma'_{i+1}|$. Inserting this operator in equation (3.10) and letting the operator $O_{i,i+1}$ act on it, we obtain

$$\begin{aligned}O_{i,i+1}|\psi\rangle &= \sum_{\sigma_i,\sigma_{i+1}} \sum_{a_{i-1},a_{i+1}} \sum_{\sigma'_i\sigma'_{i+1}} \Theta^{\sigma_i,\sigma_{i+1}}_{a_{i-1},a_{i+1}} \langle\sigma'_i\sigma'_{i+1}|O_{i,i+1}|\sigma_i\sigma_{i+1}\rangle |\mu_{a_{i-1}}\rangle_L |\sigma'_i\sigma'_{i+1}\rangle |\mu_{a_{i+1}}\rangle_R \\ &= \sum_{a_{i-1},a_{i+1}} \sum_{\sigma'_i\sigma'_{i+1}} \widetilde{\Theta}^{\sigma'_i,\sigma'_{i+1}}_{a_{i-1},a_{i+1}} |\mu_{a_{i-1}}\rangle_L |\sigma'_i\sigma'_{i+1}\rangle |\mu_{a_{i+1}}\rangle_R,\end{aligned} \tag{3.11}$$

as the updated quantum state. The new tensor is $\widetilde{\Theta}^{\sigma'_i,\sigma'_{i+1}}_{a_{i-1},a_{i+1}} = \sum_{\sigma_i,\sigma_{i+1}} \Theta^{\sigma_i,\sigma_{i+1}}_{a_{i-1},a_{i+1}} \langle\sigma'_i\sigma'_{i+1}|O_{i,i+1}|\sigma_i\sigma_{i+1}\rangle$, this computation takes a polynomial time of $\mathcal{O}(d^4\chi^2)$, because we have four spin bases σ of dimension d and the site matrices have a maximum dimension of $\chi \times \chi$ due to the performed truncations. Now we would like to express the quantum state (3.11) in the canonical form of equation (3.8). This is computationally the most time demanding step. To achieve the original form we first have to reshape the tensor $\widetilde{\Theta}$ into a matrix. On this matrix we compute a SVD and we reshape the matrices of the SVD to the correct shape

$$\widetilde{\Theta}^{\sigma'_i,\sigma'_{i+1}}_{a_{i-1},a_{i+1}} = \widetilde{\Theta}_{(\sigma'_i a_{i-1}),(\sigma'_{i+1} a_{i+1})} = U^{\sigma'_i}_{a_{i-1},a'_i} \widetilde{\Lambda}_{a'_i a'_i} V^{\sigma'_{i+1}}_{a'_i,a_{i+1}}. \tag{3.12}$$

Now the new index a'_i can be larger than the bond dimension χ so as to keep the information that we need to store low enough, we also truncate all the Schmidt coefficients after χ. The truncated versions of the matrices U and V are the new site matrices $\widetilde{\Gamma^{\sigma}_i}$ and $\widetilde{\Gamma^{\sigma}_{i+1}}$ and we are back in the form of equation (3.8). The process just described is also shown in figure 3.4 with the schematic representation defined in the previous section.

Dot product

To calculate the ground state energy in the TEBD algorithm we need to be able to compute a dot product. There are two methods to calculate a dot product between two quantum states $\langle\psi|$ and $|\phi\rangle$ when both of these states are in the canonical form of MPS. The first approach, although quite easy to grasp is incredibly inefficient. It consist of just calculating all the contractions of the physical legs with each other at the same time. The reason this method is not efficient is that this computation is similar to writing the states $\langle\psi|$ and $|\phi\rangle$ down in the form of equation (3.1) and then computing the tensor product. But the total tensors have a dimension of d^N. So the dot product cannot be calculated in polynomial time. Fortunately, there is a second algorithm to compute the dot product which does grow with polynomial size. We start by contracting the matrices of the most left sites with each other over the physical index

$$\Omega_0 = \sum_{\sigma_1} (\Gamma^{\sigma_1}_{a'_1})^* \Gamma^{\sigma_1}_{a_1} (\Lambda'_1)^* \Lambda_1, \tag{3.13}$$

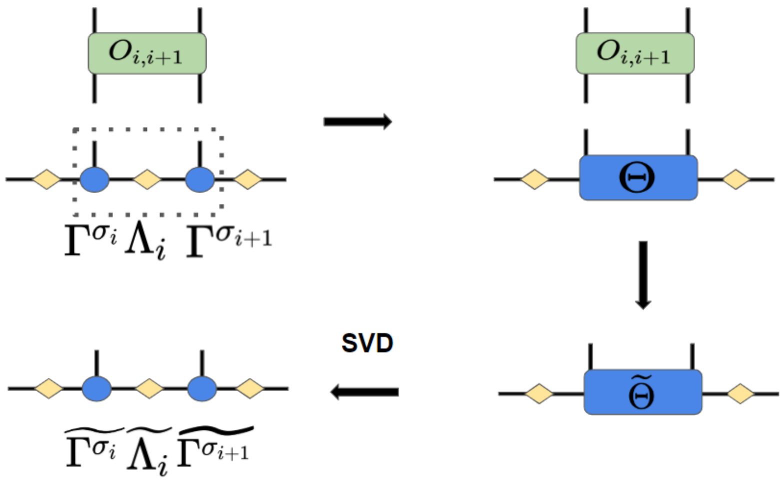

$O_{i,i+1}$
$\Gamma^{\sigma_i}\ \Lambda_i\ \Gamma^{\sigma_{i+1}}$
Θ
$\widetilde{\Theta}$
SVD
$\widetilde{\Gamma^{\sigma_i}}\ \widetilde{\Lambda_i}\ \widetilde{\Gamma^{\sigma_{i+1}}}$

Figure 3.4: The schematic representation of a two-site operator acting on a site i and $i+1$ and what steps need to be taken to efficiently calculate the effect of the operator.

here the singular values are also taken into account. We will now move through the chains from left to right with a kind of 'zipping' method. At each site we compute the tensor product of the site matrices of $\langle\psi|$ and $|\phi\rangle$. With this we update the tensor Ω_i

$$\Omega_{i+1} = \sum_{\sigma_{i+1},a_{i+1},a'_{i+1}} (\Omega_i)_{a_{i+1},a'_{i+1}} (\Gamma^{\sigma_i}_{a'_{i+1},a'_{i+2}})^* \Gamma^{\sigma_1}_{a_{i+1},a_{i+2}}. \tag{3.14}$$

Before this the singular values should be added to the site matrices as well. Since we have a column matrix at the most right site this process will converge to a single number $\Omega_N = \langle\psi|\phi\rangle$. Computing an update of Ω takes $\mathcal{O}(d\chi^4)$ polynomial time and since this calculation is repeated N times we find that the calculation of the dot product grows with $\mathcal{O}(d\chi^4 N)$. In figure 3.5 we show the two different methods to calculate the dot product.

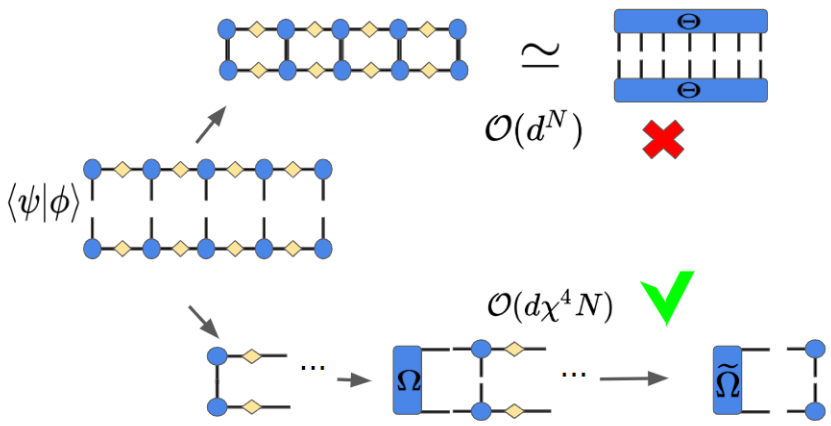


Figure 3.5: The schematic representation for the two algorithms to calculate the dot product between two quantum states in the canonical form of MPS. Here the 'zipping' method below is preferred due to its polynomial growth in N.

Hastings

We have just discussed how one can compute the effect of a two-site operator when the quantum state is in the Canonical form. This operation can sometimes fail in the code. If the calculation is unstable, the function to approximate the singular value decomposition is unable to converge. Fortunately, there is a technique to make the matrices on which we perform the SVD more stable. The technique was first described by Hastings [5]. The idea is to include in the construction of the tensor Θ in equation (3.10) the singular values of the site $i-1$ and the site $i+1$, thus adding the diagonal matrices $\Lambda_{a_{i-1},a_{i-1}}$ and $\Lambda_{a_{i+1},a_{i+1}}$ to equation (3.10). The rest of the algorithm is completely the same only till in the end where the original canonical form needs to be restored. Since we added the singular values to the calculation these need to be brought back, this can be done by multiplying the site matrices with their inverse:

$$\widehat{\Gamma}^{\sigma_i}_{a_{i-1},a_i} = \sum_{a'_{i-1}} (\Lambda_{a_{i-1},a'_{i-1}})^{-1} \widetilde{\Gamma}^{\sigma_i}_{a_{i-1},a'_i}; \tag{3.15a}$$

$$\widehat{\Gamma}^{\sigma_{i+1}}_{a_i,a_{i+1}} = \sum_{a'_{i+1}} \widetilde{\Gamma}^{\sigma_{i+1}}_{a_i,a'_{i+1}} (\Lambda_{a'_{i+1},a_{i+1}})^{-1}. \tag{3.15b}$$

By adding the singular value to equation (3.10) we amplify the most crucial information in the tensor, while diminishing the inessential information ans as a result the SVD becomes more stable.

4

Time-Evolving Block Decimation

In the previous chapter the idea behind Matrix Product States was explained. It was shown how one can construct the Canonical form of MPS and that if a quantum state $|\psi\rangle$ is in the the regime of the area law, truncations can be performed on the singular values with minimal loss of information. This all resulted in an efficient way to represent a quantum state. In this chapter an algorithm is explained which uses this representation to calculate the ground state energy of a local Hamiltonian. With this algorithm normal time evolution can also be computed as long as the state falls within the area law. The algorithm is called Time-Evolving Block Decimation (TEBD) and the reason it is so powerful is that all the computations in this algorithm can be done in polynomial time. The algorithm and eventual code are based on an iTEBD code developed by Pollmann [10].

4.1. Time-Evolution

With the TEBD algorithm one can not only compute the time evolution of a quantum state but the low-lying eiegenstates of the Hamiltonian H as well. To see why this is, we introduce the well-known Schrodinger equation

$$\mathrm{i}\hbar\frac{\mathrm{d}|\Psi\rangle}{\mathrm{d}t} = \widehat{H}|\Psi\rangle. \tag{4.1}$$

The general solution of the Schrodinger equation is $|\Psi(t)\rangle = e^{-\mathrm{i}\widehat{H}t/\hbar}|\Psi(t=0)\rangle$, here the exponent $e^{-\mathrm{i}\widehat{H}t}$ is a complex operator ($\hbar$ is again set to be one). This solution can also be expressed in the basis of the eigenstates of the Hamiltonian H. If we then let the time evolution operator act on it, we obtain

$$e^{-\mathrm{i}\widehat{H}t}|\Psi(t)\rangle = \sum_n e^{-\mathrm{i}\widehat{H}t}c_n|\phi_n\rangle = \sum_n c_n e^{-\mathrm{i}E_n t}|\phi_n\rangle. \tag{4.2}$$

Generally the eigenstates of a system are unknown and this form does not give us anything new. However, if the time evolution is not done in real time but in imaginary time, i.e. $t \to -\mathrm{i}\tau$ expression (4.2) can be used to compute the ground state. To do this we take the limit with respect to τ of equation (4.2)

$$\lim_{\tau\to\infty}|\Psi(\tau)\rangle = \lim_{\tau\to\infty}\sum_n c_n e^{-E_n\tau}|\phi_n\rangle. \tag{4.3}$$

Now we assume that the Hamiltonian is gapped, i.e. $E_0 < E_i$ for all $i \in \mathbb{N}$. In that case the limit of (4.3) decreases exponentially faster for all the excited energies compared to the ground energy and as a result

$$\lim_{\tau\to\infty}|\Psi(\tau)\rangle \propto |\phi_0\rangle. \tag{4.4}$$

If the ground state is degenerate than the limit is a superposition of these degenerate states. With this method it is also possible to find the other excited states. If the quantum state $|\Psi(0)\rangle$ has no overlap with the ground state, then limit (4.3) will converge to the first excited state and not to the ground state. With this method all the eigenstates can be found, if one is able compute the effect of the time evolution operator.

Ground-Energy

Equation (4.3) can also be used to calculate the ground state energy. Suppose that we have taken already a large time step τ and that we have found a quantum state $|\phi\rangle$ which is proportional with the ground state. Now we let the operator $e^{-\widehat{H}\delta}$ work on it one more time, this gives $|\psi_{n+1}\rangle = e^{-\widehat{H}\delta}|\psi_n\rangle = e^{-E_0\delta}|\psi_n\rangle$. Now we compute the inner-product of these two states with them self. The ratio of these dot products is a measure for the ground state energy

$$E_{ground} = \frac{\ln \frac{\langle \psi_{n+1}|\psi_{n+1}\rangle}{\langle \psi_n|\psi_n\rangle}}{2\delta}. \tag{4.5}$$

The energy of the excited states can be found in a similar manner as long as the quantum state is proportional to the excited state and the state falls within the area law.

4.2. Block Decimation

If one wants to perform real or imaginary time evolution, one needs to be able to compute the effect of the operator $e^{-\mathrm{i}\widehat{H}t}$ on the quantum state $|\phi\rangle$. In the current form the operator $e^{-\mathrm{i}\widehat{H}t}$ grows exponentially with the system size d^N, so it is generally not possible to calculate the numerical result of (4.2). With the TEBD algorithm we decompose this complex tensor into different sub blocks, the effect of these sub blocks can be computed individually. Putting these effects in a sequence we recreate the total time evolution. To be able to conveniently describe the process of decomposing the Hamiltonian we are going to assume all the Hamiltonians we are going to work with have the following form

$$H = \sum_{i=1}^{N} O^{[i]} + \sum_{i=1}^{N-1} O^{[i,i+1]}. \tag{4.6}$$

That is to say, the Hamiltonian only consist of operators which either act on an individual site or on two sites right next to each other. Note that such Hamiltonians have the local character that is needed for the MPS formalism to work. To decompose the operator $e^{-\mathrm{i}\widehat{H}t}$ into different blocks, the summation in the exponent needs to be replaced with a product op exponents. Theorem 2.1 shows that this can be done without error if the operators in the exponent commute with each other. The idea is now to decompose Hamiltonian (4.6) into a group of sub Hamiltonians, such that each sub Hamiltonian only consists of operators that commute with all the other operators that are in the same group. With theorem 2.1 we are able to decompose the sub Hamiltonian in a series of exponential operators without creating any errors.

Theorem 2.1 says nothing about operators that do not commute with each other. Still, there are operators in Hamiltonian (4.6) that do not commute with each other. Now there exists still a method to decompose the sum in the exponent to a product of exponents if the operators do not commute, this process does introduce a error in the computation. The exponent $e^{-\mathrm{i}\widehat{H}t}$ can be divided into blocks with the Trotter-Suzuki decomposition [13]

$$e^{t(A+B)} = e^{tA}e^{tB} + \mathcal{O}(t^2). \tag{4.7}$$

Since the error the Trotter-Suzuki decomposition introduces is quadratic in time, it is recommend to make the size of the time steps as small as possible. On the other hand if the time step is too small the limit of equation (4.4) will never converge. These two considerations need to be carefully studied in the code to have an optimal balance in convergence speed and minimal error.

To further reduce the error equation (4.7) bestows, we make sure that the decomposition has to be carried out only once. The decomposition we do here is taken from [7]. Hamiltonian (4.6) is split into a set of operators that work on the even bonds (the bond indices are the same as the site indices ony they run from 1 to $N-1$) and into a set of operators that work on the odd bonds. The operators that act on a single site are halved and incorporated in the bond operators to the left and to the right. This is done to make the operators symmetric. In the system contains an odd amount of spin particles the decomposition is

$$H = H_{odd} + H_{even} \tag{4.8a}$$

$$H_{odd} = O^{[1]} + O^{[N]} + \frac{1}{2}\sum_{i=1}^{N} O^{[i]} + \sum_{\text{odd i}}^{N} O^{[i,i+1]} \tag{4.8b}$$

$$H_{even} = \frac{1}{2}\sum_{i=1}^{N} O^{[i]} + \sum_{\text{even i}}^{N} O^{[i,i+1]}. \tag{4.8c}$$

If N is even the term $O^{[N]}$ moves from (4.8b) to (4.8c). With these two sub Hamiltonian we only need to perform the Trotter-Suzuki decomposition (4.7) once. The single-site and two-site operators that act on the same sites are brought together into one larger operator, i.e. $\frac{1}{2}(O^{[i]} + O^{[i+1]}) + O^{[i,i+1]} = \widetilde{O}^{[i]}$. Observe that we have that $[\widetilde{O}^{[i]}, \widetilde{O}^{[j]}] = 0$ if there is another bond between i and j. Thus all the operators in the Hamiltonian H_{even} commute with each other and can be computed independently without yielding an extra error. The same can be done for all the operators that act on a odd bonds. The decomposition we have just described is also shown visually in figure 4.1, where the schematic representation as introduced before is used. In chapter

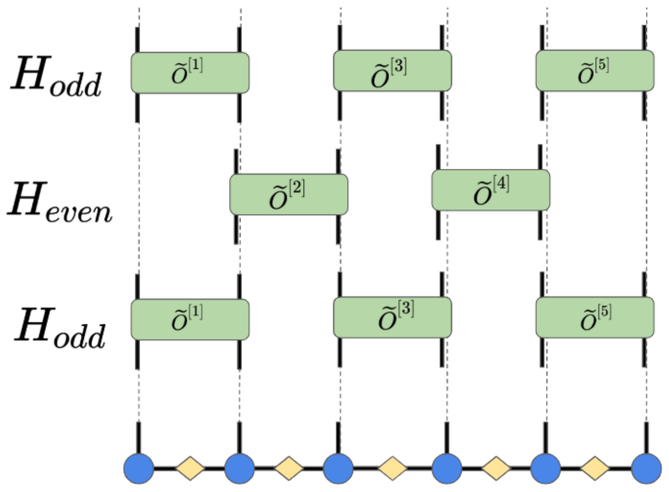


Figure 4.1: The schematic representation of the decomposition of Hamiltonian (4.6) in the TEBD algorithm.

3 we explained how to efficiently compute the effect of the two-site operators $\widetilde{O}^{[i]}$. So, we are now able to implement almost able to implement the TEBD algorithm into a functional code.

4.3. Bond Dimensions

In this chapter we explain the idea behind TEBD algorithm. In the literature an easy implementation of this algorithm is often used. This implementation is called iTEBD (i for infinite) and can be used if boundary of the system can be ignored. Although both algorithms employ the same idea of MPS, time-evolution and block decimation there is a difference between the two algorithms and non-surprisingly it has to do with the boundary. Here we will explain how to implement the normal TEBD algorithm.
In chapter 3 we showed how to obtain the Canonical form (3.8) of a quantum state $|\phi\rangle$. There it was demonstrated that the dimensions of the bond indices grow from left to right with a multiplication of d each time. The same process is happening from right to left and the real bond size is the minimum of the two, all in all the bond dimensions grow like this

$$1 \times d, d \times d^2, d^2 \times d^3, ..., d^{i-1} \times d^i, d^i \times d^{i-1}, ..., d^2 \times d, d \times 1. \tag{4.9}$$

Now, since it is possible to truncate many of the singular values, we found that we only had to keep the χ Schmidt vectors corresponding to the the largest SVD eigenvalues. However, clearly this result is not correct at the boundary sites, since in most cases χ is larger than d or even higher powers of d. Fortunately, this only means that at the boundaries we do not need to throw away any information and the information at the sites is only more accurate. The matrix dimensions have to be in the form that the product of the matrices results in a single number. In the algorithm we start with a random quantum state that falls within the area law and is saved in the Canonical form of MPS. The matrices of the sites that are in the middle of the chain are already truncated, to that end we have figure **??** where the dimensions of each leg are written down in the schematic notation introduced before. Here the minimum of χ and the local bond dimension have been taken.

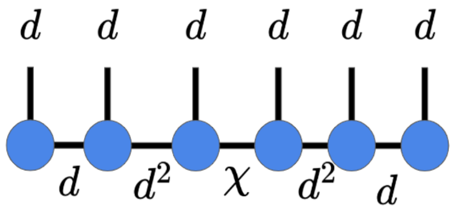


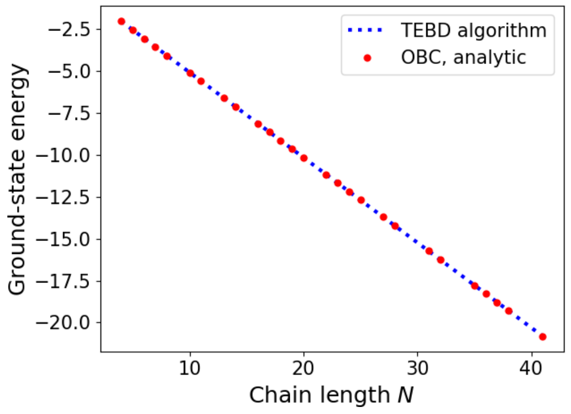
Figure 4.2: A quantum state $|\phi\rangle$ in the Canonical form of MPS. The black legs represent the indices of the mathematical object. The number at each leg is the dimension of that index.

5 Numerical results of the TEBD algorithm and Error Analysis

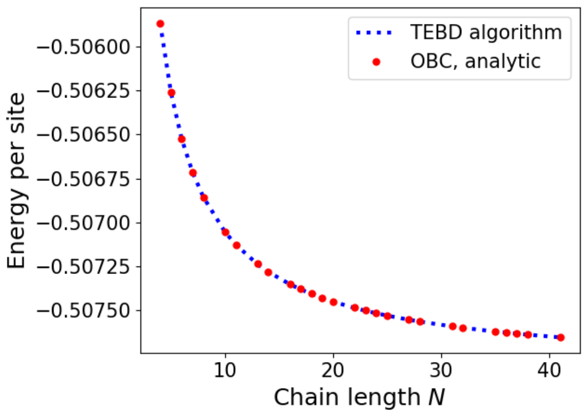
In the previous chapter it was explained how one can implement the TEBD algorithm. It was found that the ground state energy of a Hamiltonian in the form of equation (4.6) can be computed using the TEBD algorithm. The algorithm was implemented in Python for the imaginary time evolution. In order to confirm that the ground state energy found by the python code is indeed correct the results are compared with the exact solution of the Ising model in a transverse field which were determined in chapter 2.

5.1. Ground State Energy

First we will verify that the ground state energy of both systems are the same. Since the TEBD code only works for open boundary conditions we will use the solutions of Hamiltonian (2.1) with the same boundary conditions. For the exact solution we have to find N distinct roots of equation (2.25). Sometimes python was unable to find N distinct roots due to its numerical precision, therefore only the values are shown for which the solution was found. The exact solution is plotted together with the solution of the TEBD algorithm in figure 5.1a as a function of the total number of sites. Furthermore the same results are also plotted in 5.1b, but now as energy per site. From figure 5.1 we can confirm that both solutions are in excellent agreement



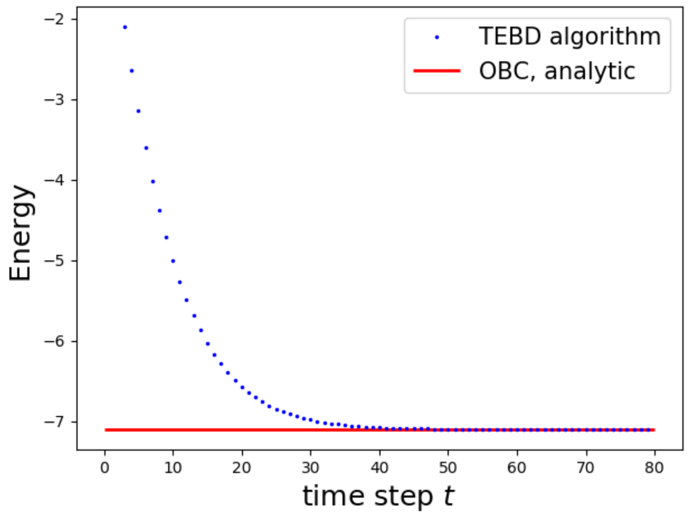
(a)



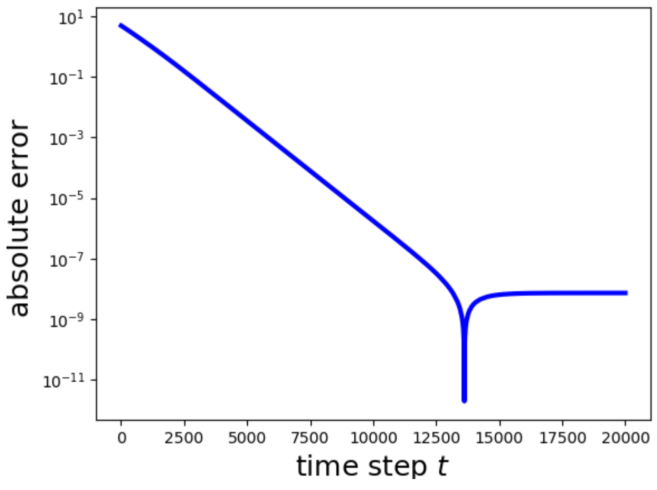
(b)

Figure 5.1: Plot of the ground state energy computed with the TEBD code as a function of the chain length. The exact solutions for OBC are also plotted in the same graph. Here $\Gamma = 1.0$ and $J = 0.5$. The size of the time step is $\delta = 0.05$ and we have a maximum bond size of $\chi = 5$.

with each other. To further research how accurate the TEBD algorithm is we compute the ground state energy after every time step and plot this against the time. From equation (4.3) we expect that the code converges exponential to the ground state limit. The graph of this convergence can be seen in 5.2a. From the plot it can be deduced that the TEBD code can compute an estimate of the ground state energy fairly quickly. To obtain a better understanding of how fast the code converges to the ground state energy we made a log-plot of the absolute error in the energy as a function of time. In figure 5.2b the graph of this result can be seen. From the graph we observe that the code converges exponentially to the ground state energy just as is expected. However, at $t = 13000$ something strange is happening we would expect that the successive approximation of the energy keeps on converging to the exact limit. What happens is that the decomposition of the Hamiltonian introduces a small absolute error to which the plot converges. The computed ground state energy is eventually a little lower than the exact result. However at the start the random quantum state is a sum of all the different excited states. Therefore the computed energy at the start of the time evolution will always be higher than the ground state energy. Since the successive energy approximation converges to a limit lower than the exact result we have that the graphs of the energies have to cross. The spike in figure 5.2b is the product of this crossing. Notice that quickly after the crossing the energy becomes constant in time showing that the solution has converged. Figures 5.2 are further proof that the TEBD code does not only produce correct results, the predictions of the TEBD algorithm on convergence are correct as well.



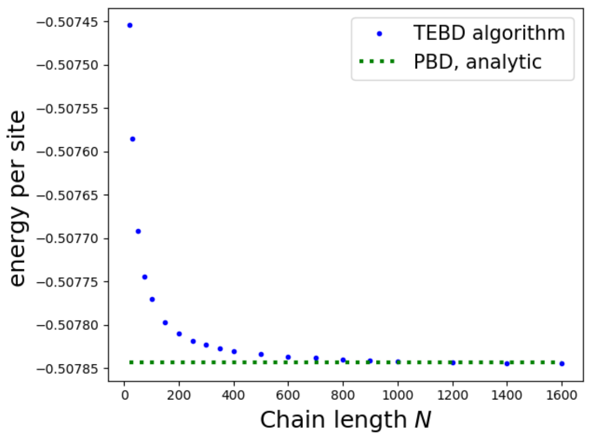
(a) The ground state energy is plotted with the blue dots, the red line is the exact solution of the problem, here $\delta = 0.1$.



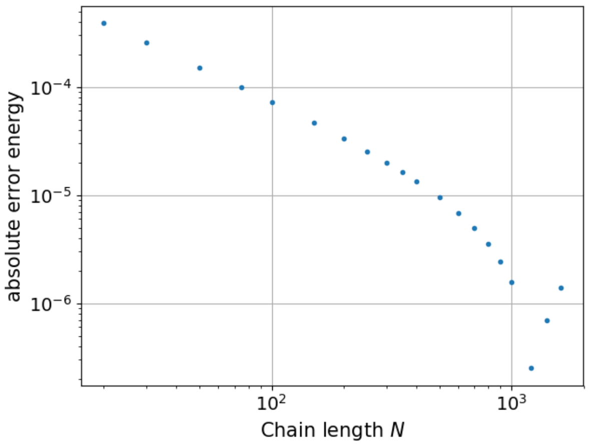
(b) A log plot of the absolute error as a function of time, here $\delta = 0.001$.

Figure 5.2: Plots of the successive approximation of the ground state energy versus the time t. The code was run for a chain with $L = 14$. Here $\Gamma = 1.0$ and $J = 0.5$ and we have a maximum bond size of $\chi = 5$

We will now present the results of the TEBD algorithm for long chains. In this case we expect that the energy per site will converge to the known limit of the energy per site of Hamiltonian (2.1) with periodic boundary conditions. The reason periodic boundary conditions are considered is that for large N the boundary effects can be neglected. In 5.3a the energy per site, computed by the TEBD code is plotted versus the chain length N. The PBC solution is also plotted with equation (2.24). Clearly, for small chains the energies do not coincide, since at this time we are ignoring boundary terms that still have a large influence. However, for larger N the energy per site converges to the correct limit. To see how accurate these results are we have also plotted the absolute error in figure 5.3b as a function of N in a log-log plot. Here we can see that for the relative high N values up to $N = 300$ the error in the energy per site decreases as $\mathcal{O}(\frac{1}{N})$., while for larger chains the convergence suddenly increaes after which it stops. In order to obtain the results for large N a few optimizations had to be done, as the numbers would become so huge that the SVD could not converge or the norm of the quantum state would become infinite. At this moment the code is optimized to work for values up to $N = 1600$. We can conclude that the code is working for considerable large N.



(a) Energy per site for both the TEBD code and the solution of the periodic boundary condition.



(b) Difference of the energy per site of the TEBD code and the periodic boundary condition.

Figure 5.3: Plot of the ground state energy from the TEBD code versus the number of sites N. Here $\Gamma = 1.0$ and $J = 0.5$. The parameters of the code are $\delta = 0.1$ and we have a maximum bond size of $\chi = 5$.

5.2. Error Analysis

To obtain the an efficient TEBD code a few assumptions and truncations have been made in the algorithm. In this section we will investigate what the errors are of these simplifications on the ground state energy.

Truncation

In chapter 3 the idea of Matrix Product States was explained, there we saw that the reason MPS is so successful is that it is possible to truncate a lot of useless information if a quantum state falls within the area law. In figure 3.1 we already saw that the singular values belonging to a bond in the middle of the chain, decreases exponential for the Ising model just as was proven by Vidal [18]. To gain further insight on what the resulting error is from this truncation on the obtained ground state we show figure 5.4. Here the relative error of the TEBD code is plotted as a function of the maximum bond sizes χ. In the graph we can not see any relation between the two quantities. One would expect that the relative error decreases if more singular values are taken into account. However, it turns that the singular values decrease so fast that for small N that it is already sufficient to only keep the two Schmidt vectors corresponding with the two largest singular values. This result is advantageous to us, not only are we fully inside the region of states where the area law holds, we can even choose the bond size as low as 2. In chapter 3 we found that the computation of the two-site operator has a polynomial growth of $\mathcal{O}(d\chi^4)$, thus being able to use such a low bond size is extremely useful.

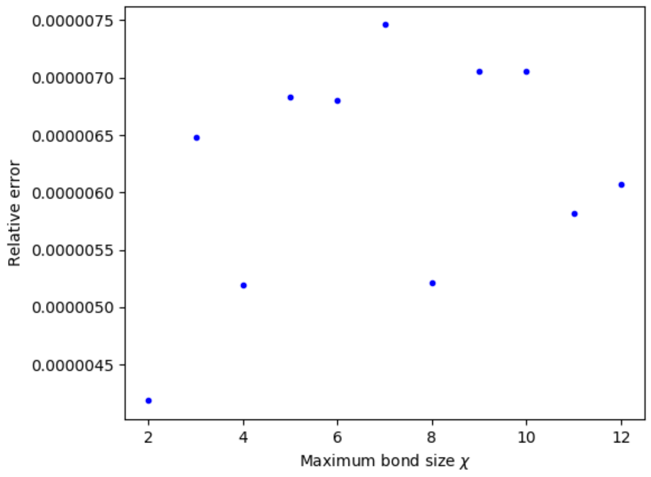
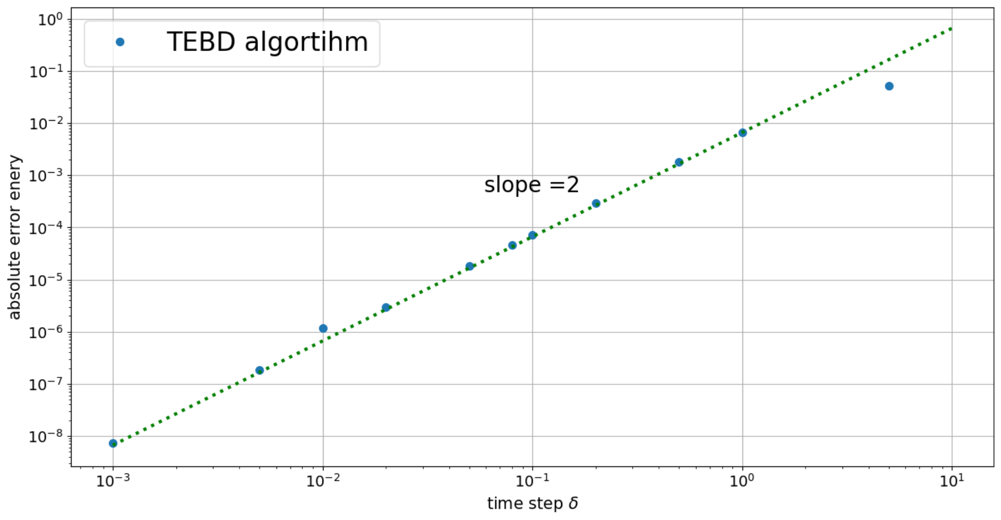


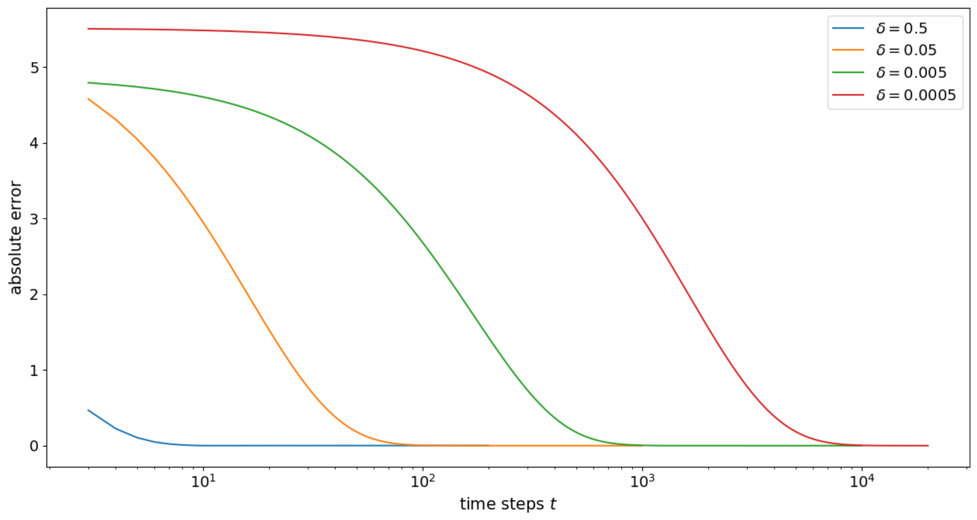
Figure 5.4: Relative error of the ground state energy computed with different maximum bond sizes χ for a of chain length $N = 14$. Here $\Gamma = 1.0$ and $J = 0.5$ and $\delta = 0.01$.

Decomposition

In order to compute the effect of the operator e^{-Ht} on a quantum state $|\psi\rangle$, we had to decompose the Hamiltonian into different blocks. If the operators that we separated into blocks commuted with each other this did not pose any problems. However, if they did not commute we had to perform a Trotter-Suzuki decomposition (4.7). This resulted into an extra error which grew with $\mathcal{O}(\delta^2)$, where δ is the size of a single time step. In figure 5.5a this time step is plotted versus the absolute error of the TEBD code on a log-log plot. The green line drawn in the graph is the function $y = cx^2$, which presents the error of the Trotter-Suzuki decomposition. The constant c is the value of the absolute error at $\delta = 1$. Since all the blue points are close to the green line we can conclude that the TEBD code has the same quadratic relation in its error of the time step, i.e $\mathcal{O}(\delta^2)$. One does need to keep in mind that even tough, a twice as small τ results in error reduction of 1/4, that it will take substantial more time for the code to converge. The code will reach its limit, but the total time steps taken have to be increased. To show this effect figure 5.5b has been created. There the absolute error of the successive approximation of the ground state energy has been plotted for various δ's. We find that the total time the TEBD code needs to convergence depend on the size of the time step with $\mathcal{O}\frac{1}{\delta}$. This is also what we expect from the theory on TEBD with equation (4.3).



(a) The blue dots are a log-log plot of the absolute error versus the size of the time step δ as computed with the TEBD code. The green line is the function $y = cx^2$.



(b) A graph on the convergence speed of the TEBD code for various time step sizes, the total amount of time steps axis is log-scaled.

Figure 5.5: Graphs on the error dependence of the time step δ and the corresponding amount of time steps taken before the solution converges to the ground state energy. Here $\Gamma = 1.0$ and $J = 0.5$,the chain has length $L = 14$ and we have a maximum bond size of $\chi = 5$.

6 Conclusion and Suggestions for Further Research

The excellent agreement of the numerical results with the exact solution combined with the manageable computation times proves that the TEBD algorithm is an ideal method to compute the ground state energy of Hamiltonians with a local character. We have seen that for small or large spin systems the TEBD code converges exponentially to its result. Furthermore the resulting error decreases quadratically with the size of the time step. Thus any precision can be achieved, albeit at the expense of longer run times. We have also seen that the maximum bond size has little or negligible influence on the error of the TEBD code and as a result for the Ising model this bond size could be chosen as low as $\chi = 2$. We can conclude that the theory of Matrix Product States can be implemented in the TEBD algorithm resulting in an efficient code for computing the ground state energy.

Now that we have validated that the algorithm returns accurate results, we can use it to solve more difficult problems of which the exact solution of the Hamiltonian is unknown. Sadly, due to time constrains we were not able to further develop the algorithm. Here we will present a few interesting upgrades that can be done on the algorithm to increase the class of systems it is able to solve. One of these upgrades is to create a function that calculates the addition or subtraction of two quantum states in the Matrix Product States form. This function can be used to compute the excited states and energies. Another powerful upgrade that can be implemented is to include real time evolution in the algorithm, with which it is possible to compute the wave function for all times, if the quantum state falls within the area law.

To broaden the class of Hamiltonians that the TEBD algorithm is able to solve a swap operator can be defined as in [7]. Hamiltonians consisting of two-site operators that are not acting on nearest neighbours sites can be computed with the use of this swap operator. If implemented this swap operator could be able to solve Hamiltonians which have periodic boundary conditions. Furthermore, it is possible to implement a function that computes the effect of an operator that acts on three of four sites at once this would take a considerable longer run time, but it can help solve more difficult systems. For all of these implementations the non-locality of the problem is enlarged and as a result the maximum bond dimension should be increased. Finally, with relative few changes the code could be used for higher spin values, as the only parameter in the code that changes is d, the size of the local basis.

The Ising model in a transverse field could also be further explored to see what happens with the correlation functions for different chain lengths. This can be done both analyticly and numerically with the TEBD code. The Ising model was also a good candidate to confirm the results of the TEBD algorithm. Not only did it fit the conditions necessary for the Matrix Product States representation to be applicable, the ability to calculate exact results were useful for checking the accuracy of the code. The process described to analytically solve the Hamiltonian can be copied for similar systems. In the end we can conclude that with Matrix Product States and the Time-Evolving Block Decimation algorithm we can numerically solve low-entangled systems in one dimension within a reasonable run time.

A The Python code

```python
"""
File for the TEBD algorithm for the Bachelor thesis project of Pim Vree,
MPS with boundary conditions and the Hamiltonian of the Ising Model.
"""

import matplotlib.pyplot as plt
import numpy as np
import scipy.special as sc
import copy
import math
from scipy import optimize
"""
Functions for calculating the analytics solutions of the Ground state energies, depending on
"""
def Ground_Energy_Pfeuty(g,Lambda,N):
    """The ground energy of the Ising model with periodic boundary conditons, works only for
    theta = np.sqrt(4*Lambda/(1+Lambda)**2)                    # Lambda = J/(2*g) as defined in
    return -g*2/np.pi * (1+Lambda)*sc.ellipe(theta)       # This function seems to be incorre

def Ground_energy_Pim(Gamma,J,N):
    """The ground energy of the Ising model with periodic condtiotns, calculated by Pim Vree
    m = np.linspace(-(N-1)/2, (N-1)/2,N)
    Lambda = J/(2*g)
    k = 2*np.pi *m /N
    res = np.sum(np.sqrt(1+ Lambda**2 + 2*Lambda*np.cos(k)))        # Lambda = J/(2*g) as def
    return -Gamma * res/2

def Ground_energy_OBC(g,Lambda,N):
    """ Returns the ground state energy for the Ising model with open boundary conditions, the
difficult part is that we have to find N distinct roots of the function ff and for large N thi
    def ff(k):          # Note this function is only valid when Lambda <= 1 !!!
        return np.sin(k*(N+1))/np.sin(k*N) + Lambda
    sol = optimize.root(ff,np.arange(0.001,np.pi,0.05),tol = 10**-16).x     # Find all roots
    diff = []
    test = []
    sol2 = np.round(sol,12)
    for k in range(len(sol)):
        if sol[k]<np.pi and sol[k]>0:           # keep only the distinct roots
            if sol2[k] not in diff:
                diff.append(sol2[k])
                test.append(sol[k])
    diff.sort()
    if len(diff) == N:                          # if a different number of distinct roots is
```

```
        Lambda2 = 1+ Lambda**2 + 2*Lambda*np.cos(test)
        res = sum(np.sqrt(Lambda2))
        return -g*res/2
    print('Function failed')

def Hamiltonian_Ising_model(g,J):
    """
    Here the Hamitlonian we use is constructed, note that we have three different forms
    After this an array is filled with the e^H form per two site operator used for in th
    """
    H = np.array([[-g/2,0,0,-J/4],[0,0,-J/4,0],[0,-J/4,0,0],[-J/4,0,0,g/2]])
# Hamiltonian for a bond that is located in the middle
    H_left = np.array([[-3*g/4,0,0,-J/4],[0,-g/4,-J/4,0],[0,-J/4,g/4,0],[-J/4,0,0,3*g/4]
# Hamiltonian for the most left bond
    H_right = np.array([[-3*g/4,0,0,-J/4],[0,g/4,-J/4,0],[0,-J/4,-g/4,0],[-J/4,0,0,3*g/4
    return H,H_left,H_right

def two_site_Hamiltonian(H,delta):
    """  Returns the two site operator version of Hamiltonian e^{-H*delta}, as (2,2,2,2)
    w,v = np.linalg.eig(H)
    return np.reshape(np.dot(np.dot(v,np.diag(np.exp(-delta*(w)))),np.transpose(v)),(2,2

def Initializing_State(L,chi,d):
    """For the OBC, the maxium bond size is smaller than chi at the boundaries and here
    arr = np.arange(0,L+1)
    arr = np.minimum(arr, L-arr)
    arr = np.minimum(arr,chi)                # For large L, d**arr returns negative value
    loc_size = np.minimum(d**arr, chi)
    """
    Here  a random state Phi is constructed which is in the canoncial form from the bach
    """
    lambdas = np.zeros((L+1,chi))
    gammas  = np.zeros((L,chi,chi,d))
    for j in np.arange(0,L):
        lef_size = loc_size[j]
        rig_size = loc_size[j+1]
        lambdas[j,:lef_size] = np.random.rand(lef_size)
# last lambda is neglected
        gammas[j,:lef_size,:rig_size,:d]  = np.random.rand(lef_size, rig_size,d)
    lef_size = loc_size[L]
    lambdas[L,:lef_size] = np.random.rand(lef_size)
# last lambda is neglected
    lambdas[0,0] = 1
    lambdas[L,0] = 1
    return lambdas,gammas,loc_size

def calc_norm(gammas, lambdas, L):
    """ function for calculating the norm of a state Phi with the 'zipping' method descr
    m_total = np.eye(chi)
    for j in range(0, L):
        sub_tensor = np.tensordot(gammas[j,:,:,:],np.diag(lambdas[j+1,:]), axes=(1,0))
        mp = np.tensordot(sub_tensor,sub_tensor,axes = (1,1))
        m_total = np.tensordot(m_total,mp,axes=([0,1],[0,2]))
    if math.isnan(m_total[0,0]):
# For large N value can be too large
        lambdas[1:L] = lambdas[1:L]/1.5
# These lines make sure that the function returns a value
        m_total[0,0] = calc_norm(gammas,lambdas,L)
    return m_total[0,0]
# Only the first element is non-zero
```

```python
def dotproduct(gammas1,lambdas1,gammas2,lambdas2,L):
    """ This function can be used to calculate the dotproduct between two states, currently o
    m_total = np.eye(chi)
    for j in range(0,L):
        sub_tensor1 = np.tensordot(gammas1[j,:,:,:],np.diag(lambdas1[j+1,:]), axes=(1,0))
        sub_tensor2 = np.tensordot(gammas2[j,:,:,:],np.diag(lambdas2[j+1,:]), axes=(1,0))
        mp = np.tensordot(sub_tensor1,sub_tensor2,axes = (1,1))
        m_total = np.tensordot(m_total,mp,axes=([0,1],[0,2]))
    return m_total[0,0]
# Only the first element is non-zero

def Two_site_Operator(i,gammas,lambdas,T,O_arr,L,d,chi,loc_size):
    """The result of a two-site operator is computed with the process explained in the Bachel
    theta = np.tensordot(np.diag(lambdas[i,:]), gammas[i,:,:,:], axes=(1,0))
    theta = np.tensordot(theta,np.diag(lambdas[i+1,:]),axes=(1,0))
    theta = np.tensordot(theta, gammas[i+1,:,:,:],axes=(2,0))
    theta = np.tensordot(theta,np.diag(lambdas[i+2,:]), axes=(2,0))
    theta_prime = np.tensordot(theta,O_arr[i,:,:,:,:],axes=([1,2],[0,1]))
# Two-site operator
    theta_prime = np.reshape(np.transpose(theta_prime, (2,0,3,1)),(d*chi,d*chi)) # danger!
    #Singular value decomposition
    X, Y, Z = np.linalg.svd(theta_prime); Z = Z.T
    #truncation
    lambdas[i+1,:] = Y[:chi]
    X = np.reshape(X[:d*chi,:chi], (d, chi,chi))  # danger!
    tmp_gamma = np.tensordot(np.diag(lambdas[i,:loc_size[i]]**(-1)),X[:,:loc_size[i],:loc_siz
    gammas[i,:loc_size[i],:loc_size[i+1],:] = np.transpose(tmp_gamma,(0,2,1))
    Z = np.reshape(Z[0:d*chi,:chi],(d,chi,chi))
    Z = np.transpose(Z,(0,2,1))
    tmp_gamma = np.tensordot(Z[:,:loc_size[i+1],:loc_size[i+2]], np.diag(lambdas[i+2,:loc_siz
    gammas[i+1,:loc_size[i+1],:loc_size[i+2],:] = np.transpose(tmp_gamma,(1, 2, 0))
    return gammas,lambdas

def Time_evolution(gammas,lambdas,T,O_arr,L,d,chi,loc_size,delta):
    """The Odd-Even time evoltuon is done here, after each time step the Ground state energy 
    time = [];Ground_Energy = []
    for t in range(T):
        for i in range(1,L-1,2):            # Odd bonds
            gammas, lambdas = Two_site_Operator(i,gammas,lambdas,T,O_arr,L,d,chi,loc_size)
        for i in range(0,L-1,2):            # Even bonds
            gammas, lambdas = Two_site_Operator(i,gammas,lambdas,T,O_arr,L,d,chi,loc_size)
        norm = calc_norm(gammas, lambdas, L)
        time.append(t)
        Ground_Energy.append(-np.log(norm)/(2*delta))
        lambdas[1:,:] = lambdas[1:,:]/norm**(0.5/(L))       # Normalizing for stability and e
        if math.isnan(Ground_Energy[-1]):                   # Break before the code crashes
            return time,Ground_Energy
        print(t,Ground_Energy[-1])
    return np.array(time),np.array(Ground_Energy)

"""
The different values for the parameters and variabels are created here
"""
d = 2               # Dimension of the spins
chi = 5             # bond dimension
L = 14              # length of the Chain
J = 0.5             # coupling contant of the Nearest_Neigbour
g = 1.0             # The strength of the magnetic field
delta = 0.05         # The time step of the time_evolution
T = 200             # The total time

```

```python

def Converging_Plot(g,J,L,chi,delta,T):
    H,H_left,H_right = Hamiltonian_Ising_model(g,J)
    O_arr = np.zeros((L-1,d,d,d,d))
    for i in range(1,L-2):
        O_arr[i,:,:,:,:] = two_site_Hamiltonian(H,delta)
    O_arr[0,:,:,:,:] = two_site_Hamiltonian(H_left,delta)
    O_arr[L-2,:,:,:,:] = two_site_Hamiltonian(H_right,delta)

    lambdas,gammas,loc_size = Initializing_State(L,chi,d)
    time,Ground_Energy = Time_evolution(gammas,lambdas,T,O_arr,L,d,chi,loc_size,delta)

    plt.semilogx(time[2:],Ground_Energy[2:])
    plt.hlines(Ground_energy_OBC(g,J/(2*g),L),0,T)
    plt.xlabel('Time step t')
    plt.ylabel('Energy')
    plt.show()


Converging_Plot(g,J,L,chi,delta,T)
```

Bibliography

[1] N. N. Bogoljubov. On a new method in the theory of superconductivity. *Il Nuovo Cimento (1955-1965)*, 7 (6):794–805, Mar 1958. ISSN 1827-6121. doi: 10.1007/BF02745585. URL `https://doi.org/10.1007/BF02745585`.

[2] Garnet Kin-Lic Chan. Matrix product states for the absolute beginner [powerpoint slides], 2014. URL `http://theory.iphy.ac.cn/TNSAA2014/ppt/Garnet%20Chan%20.pptx`.

[3] J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82:277–306, Feb 2010. doi: 10.1103/RevModPhys.82.277. URL `https://link.aps.org/doi/10.1103/RevModPhys.82.277`.

[4] D.J. Griffiths. *Introduction to Quantum Mechanics*. Pearson international edition. Pearson Prentice Hall, 2005. ISBN 9780131118928. URL `https://books.google.nl/books?id=z4fwAAAAMAAJ`.

[5] M. B. Hastings. Light-cone matrix product. *Journal of Mathematical Physics*, 50(9):095207, 2009. doi: 10.1063/1.3149556. URL `https://doi.org/10.1063/1.3149556`.

[6] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, Feb 1925. ISSN 0044-3328. doi: 10.1007/BF02980577. URL `https://doi.org/10.1007/BF02980577`.

[7] André Melo. Numerical study of a superconducting qubit for the realization of quantum ising chains using matrix product state techniques. September 2017.

[8] Don N. Page. Average entropy of a subsystem. *Phys. Rev. Lett.*, 71:1291–1294, Aug 1993. doi: 10.1103/PhysRevLett.71.1291. URL `https://link.aps.org/doi/10.1103/PhysRevLett.71.1291`.

[9] Pierre Pfeuty. The one-dimensional ising model with a transverse field. *Annals of Physics*, 57(1):79–90, March 1970. ISSN 00034916. doi: 10.1016/0003-4916(70)90270-8. URL `http://dx.doi.org/10.1016/0003-4916(70)90270-8`.

[10] Frank Pollmann. Efficient numerical simulations using matrix-product states, September 2015. URL `http://indico.ictp.it/event/a14246/session/13/contribution/20/material/0/0.pdf`.

[11] Stefan Rommer and Stellan Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B*, 55:2164–2181, Jan 1997. doi: 10.1103/PhysRevB.55.2164. URL `https://link.aps.org/doi/10.1103/PhysRevB.55.2164`.

[12] T. D. Schultz, D. C. Mattis, and E. H. Lieb. Two-dimensional ising model as a soluble problem of many fermions. *Rev. Mod. Phys.*, 36:856–871, Jul 1964. doi: 10.1103/RevModPhys.36.856. URL `https://link.aps.org/doi/10.1103/RevModPhys.36.856`.

[13] Masuo Suzuki. Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Comm. Math. Phys.*, 51(2):183–190, 1976. URL `https://projecteuclid.org:443/euclid.cmp/1103900351`.

[14] Sei Suzuki, Jun-ichi Inoue, and Bikas K Chakrabarti. *Quantum Ising phases and transitions in transverse Ising models; 2nd ed.* Lecture notes in physics. Springer, Berlin, 2013. doi: 10.1007/978-3-642-33039-1. URL `http://cds.cern.ch/record/1513030`.

[15] Jos Thijssen. Lecture notes statistical physics, Spring 2018.

[16] J. G. Valatin. Comments on the theory of superconductivity. *Il Nuovo Cimento*, 7:843–857, March 1958. doi: 10.1007/BF02745589.

[17] G. Vidal. Classical simulation of infinite-size quantum lattice systems in one spatial dimension. *Phys. Rev. Lett.*, 98:070201, Feb 2007. doi: 10.1103/PhysRevLett.98.070201. URL `https://link.aps.org/doi/10.1103/PhysRevLett.98.070201`.

[18] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93: 040502, Jul 2004. doi: 10.1103/PhysRevLett.93.040502. URL `https://link.aps.org/doi/10.1103/PhysRevLett.93.040502`.

[19] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69: 2863–2866, Nov 1992. doi: 10.1103/PhysRevLett.69.2863. URL `https://link.aps.org/doi/10.1103/PhysRevLett.69.2863`.