Universiteit Utrecht

Faculteit Bètawetenschappen

Density Matrix Renormalization Group calculations for the Ising Model with a Transverse Field

Bachelor Thesis

A.H. Kole

Natuur- en Sterrenkunde & Scheikunde

Supervisor:

Dr. L. Fritz
Institute for Theoretical Physics

June, 2018

Abstract

In this thesis, the Density Matrix Renormalization Group algorithm is introduced. Some key concepts from quantum mechanics are discussed to aid in understanding the method and it is put into the context of the Renormalization Group. The algorithm is discussed from a mostly practical perspective. As a model to test the algorithm, the Ising Model with a Transverse Field was used. This model is interesting because it is one of the simplest one-dimensional models that displays a phase transition at a finite external field. It can also be solved analytically, which makes it possible to easily check the results of DMRG calculations. The DMRG algorithm was implemented from scratch in C++ for the Transverse Field Ising Model. Using this implementation, the errors in DMRG calculations were explored, as well as the phase transition in the Transverse Field Ising Chain.

I would like to thank Lars Fritz, Sonja Fischer and Benedikt Schönauer

Contents

1 Introduction

In quantum mechanics, the central problem is solving the Schrödinger equation for the eigenstates and eigenenergies of a system. For systems with a finite-dimensional Hilbert Space, this often comes down to the diagonalization of a matrix. The problem, however, is that for many systems of interest, even seemingly simple ones, the dimensions of the matrix scale exponentially with the size of the systems. This quickly makes regular diagonalization of the matrix unfeasible. The question arises how we can extract information from these systems.

Many algorithms have been developed in the past that use clever iterative techniques to find some eigenvectors and eigenvalues without having to compute them all. This often saves a lot of computation time. A famous example is the Lanczos method. But even these algorithms have their limit.

In 1992 White proposed the Density Matrix Renormalization Group as a method to solve large one-dimensional quantum lattice problems. His algorithm builds on the principles of the Renormalization Group by iteratively thinning out degrees of freedom of a system. It achieves this by a procedure of truncated diagonalisation, which makes good use of the properties of density matrices. With this method, properties of large one-dimensional quantum lattices can be obtained for the ground state and a few excited states.

In this thesis, the theory of the Density Matrix Renormalization Group is discussed from a practical perspective. To this end, we will first discuss the properties of the product basis and of the density operator in section 2. In section 3, we will also put DMRG in the context of the Renormalization Group. The theory of DMRG itself is extensively discussed in section 4.

The Transverse Field Ising Model is used as a test model for an implementation of the algorithm. This model is discussed in section 5. This model is interesting because it displays a phase transition and is also analytically solvable. Using this model, the errors of DMRG calculations will be explored and the phase transition is investigated.

2 The product basis and the density operator

In this section, we will review two topics in quantum mechanics which will be important for the subsequent discussion on the Density Matrix Renormalization Group (DMRG). The first of these topics is the product basis, which plays an important role in describing the states used in DMRG. The second is density operators, which are central to the truncation procedure used in DMRG.

2.1 The product basis

In quantum mechanics, the full state of a system is described by its state vector living in a certain space of states (a Hilbert space). This space of states can be described by a set of vectors $\{|i\rangle\}$ which form a complete and orthonormal basis: $\sum_i |i\rangle \langle i| = \mathbb{1}$ and $\langle i|j\rangle = \delta_{ij}$. Here $\mathbb{1}$ denotes the identity operator in that space of states. When we combine two physical systems, both having their own space of states, then the most natural basis to describe this combined system is often the product basis. [1] This basis is formed by taking the tensor product between the two bases. Let $\mathcal{H}_1^N$ be the Hilbert space of dimension N for system 1, described by the basis $\{|i\rangle\}$. And let $\mathcal{H}_2^M$ be the Hilbert space of dimension M for system 2, described by the basis $\{|j\rangle\}$. Then the tensor product of these two spaces $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$ is the NM dimensional Hilbert space for the joint system and the product basis $\{|i\rangle \otimes |j\rangle\}$ can be used to describe this Hilbert space. [1] So $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$ is a space spanned by pairs of the basis vectors which span $\mathcal{H}_1^N$ and $\mathcal{H}_2^M$. A state $|\phi\rangle \otimes |\chi\rangle$, where $|\phi\rangle \in \mathcal{H}_1^N$ and $|\chi\rangle \in \mathcal{H}_2^M$, can be thought of as a state where system 1 is in state $|\phi\rangle$ and system 2 is in state $|\chi\rangle$. [1] If we can decompose $|\phi\rangle$ and $|\chi\rangle$ onto their respective bases, like $|\phi\rangle = \sum_i c_i |i\rangle$ and $|\chi\rangle = \sum_j d_j |j\rangle$, then the combined state can be written in terms of the product basis as $|\phi\rangle \otimes |\chi\rangle = \sum_{i,j} c_i d_j |i\rangle \otimes |j\rangle$. [1] From this, the linearity of the tensor product can be verified directly:

$$\begin{aligned}
|\phi\rangle \otimes (|\chi_1\rangle + |\chi_2\rangle) &= \sum_{i,j} c_i(d_j + d'_j) |i\rangle \otimes |j\rangle \\
&= \sum_{i,j} c_i d_j |i\rangle \otimes |j\rangle + \sum_{i,j} c_i d'_j |i\rangle \otimes |j\rangle \\
&= |\phi\rangle \otimes |\chi_1\rangle + |\phi\rangle \otimes |\chi_2\rangle
\end{aligned}$$

In general, it will not be possible to write a vector from the joint space as a tensor product of two states, one from each subspace. [1] And the most general state vector from $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$ will be:

$$|\Phi\rangle = \sum_{i,j} b_{ij} |i\rangle \otimes |j\rangle \tag{2.1}$$

For it to be possible to write a state as a tensor product, b_{ij} needs to be factorisable in the form $c_i d_j$. And this is only possible for two independent systems. [1]

Just as it isn't generally possible to write a state as a tensor product of two states, this is also often the case for a general operator acting in $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$. [1] The matrix elements of such a general operator are: $\langle i'| \otimes \langle j'| \hat{C} |i\rangle \otimes |j\rangle = C_{i'j'ij}$. Analogous to the coefficients of state vectors, in the special case where $C_{i'j'ij}$ can be factorised as $A_{i'i}B_{j'j}$, the operator can be written as a tensor product: $\hat{C} = \hat{A} \otimes \hat{B}$. These product operators can be defined by their action on a product state: [1]

$$(\hat{A} \otimes \hat{B}) |\phi\rangle \otimes |\chi\rangle = \hat{A} |\phi\rangle \otimes \hat{B} |\chi\rangle \tag{2.2}$$

An interesting case is when either $\hat{A}$ or $\hat{B}$ is the identity operator. You then get an operator which only acts in one of the two subspaces: $\left(\mathbb{1}_1 \otimes \hat{B}\right) |\phi\rangle \otimes |\chi\rangle = |\phi\rangle \otimes \hat{B} |\chi\rangle$.

As an example of the previous, we can look at a system of two spins 1/2. A single spin 1/2 can be described in terms of the basis formed from the eigenstates of the $\hat{S}_z$ operator for the z-component of the spin: $\{|+\rangle, |-\rangle\}$, where $\hat{S}_z |+\rangle = \frac{\hbar}{2} |+\rangle$ and $\hat{S}_z |-\rangle = -\frac{\hbar}{2} |-\rangle$. So both spins live in a two-dimensional Hilbert space. When the two spins are combined, the new system lives in a four-dimensional space of states, which can be described with the product basis: $\{|+\rangle \otimes |+\rangle, |+\rangle \otimes |-\rangle, |-\rangle \otimes |+\rangle, |-\rangle \otimes |-\rangle\}$. An example of a state in this basis would be:

$$|\Phi\rangle = \frac{1}{\sqrt{2}} (|+\rangle \otimes |+\rangle + |-\rangle \otimes |-\rangle)$$

Clearly, this state cannot be written as a simple tensor product of two states, but it can still naturally be described in the product basis. The spin-spin correlator $S_{1z} \otimes S_{2z}$ is an example of an operator in this space of states that can be written as a tensor product. Its action on our example state can easily be determined:

$$\begin{aligned}
(S_{1z} \otimes S_{2z}) |\Phi\rangle &= \frac{1}{\sqrt{2}} (S_{1z} |+\rangle \otimes S_{2z} |+\rangle + S_{1z} |-\rangle \otimes S_{2z} |-\rangle) \\
&= \frac{1}{\sqrt{2}} \left(\frac{\hbar}{2} |+\rangle \otimes \frac{\hbar}{2} |+\rangle + \left(-\frac{\hbar}{2}\right) |-\rangle \otimes \left(-\frac{\hbar}{2}\right) |-\rangle\right) \\
&= \frac{1}{\sqrt{2}} \frac{\hbar^2}{4} (|+\rangle \otimes |+\rangle + |-\rangle \otimes |-\rangle) = \frac{\hbar^2}{4} |\Phi\rangle
\end{aligned}$$

When the operators can be represented with matrices, which is often the case when you have a finite-dimensional Hilbert space, then the matrix representation of such product operators can be constructed via a simple procedure. In this case, the tensor product of two operators is equivalent to the Kronecker product of two matrices. The procedure is as follows: [2]

$$\hat{A} = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix} \qquad \hat{B} = \begin{pmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{M1} & \cdots & b_{MM} \end{pmatrix}$$

$$\hat{A} \otimes \hat{B} = \begin{pmatrix} a_{11} \begin{pmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{M1} & \cdots & b_{MM} \end{pmatrix} & \cdots & a_{1N} \begin{pmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{M1} & \cdots & b_{MM} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ a_{N1} \begin{pmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{M1} & \cdots & b_{MM} \end{pmatrix} & \cdots & a_{NN} \begin{pmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{M1} & \cdots & b_{MM} \end{pmatrix} \end{pmatrix} \tag{2.3}$$

As can be seen from this procedure, if $\hat{A}$ is an $N \times N$ matrix and $\hat{B}$ is an $M \times M$ matrix, then their Kronecker product is an $NM \times NM$ matrix as is expected for an operator acting in an NM-dimensional Hilbert space. This procedure will be useful when we want to implement the DMRG-algorithm on a computer.

2.2 The density operator

As of now, we have only regarded quantum systems which are in a pure state. That is systems whose full state can be described by a state vector in their respective Hilbert space. But this is not the only possibility. As it turns out, when you have a system composed of two subsystems, and the state for the whole system cannot be written as a tensor product, then the subsystems cannot be regarded as being in a pure state. [1] Instead the subsystems are said to be in an incoherent mixture. In this mixture, a subsystem has a statistical probability p_i to be in state $|i\rangle$. [1] These probabilities obey: $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$. Here $\{|i\rangle\}$ is the set of all possible states that the subsystem can be in. You can often just use a basis of the Hilbert space as this set of states, but some of the p_i might then be 0. A convenient way to describe such an incoherent mixture is with the density operator. Using the mixture described above, this density operator is defined as: [1]

$$\hat{\rho} = \sum_i p_i \left|i\right\rangle \left\langle i\right| \tag{2.4}$$

One of the useful features of this density operator follows from the expectation value for a certain operator $\hat{A}$. For an incoherent mixture, this expectation value is defined as: [1]

$$\begin{aligned} \langle \hat{A} \rangle &= \sum_i p_i \langle i|\hat{A}|i\rangle = \sum_{i,j,k} p_i \langle i|j\rangle \langle j|\hat{A}|k\rangle \langle k|i\rangle \\ &= \sum_{i,j,k} p_i \langle k|i\rangle \langle i|j\rangle \langle j|\hat{A}|k\rangle = \sum_{i,j,k} \langle k|p_i|i\rangle \langle i|j\rangle \langle j|\hat{A}|k\rangle \\ &= \sum_k \langle k|\hat{\rho}\hat{A}|k\rangle = \mathrm{Tr}\left(\hat{\rho}\hat{A}\right) = \end{aligned} \tag{2.5}$$

Where the last operation represents the trace of a matrix. So then the expectation value of an operator can easily be calculated by taking a trace. Another fundamental property of the density operator, which will be useful later, is that it has unit trace:

$$\mathrm{Tr}\left(\hat{\rho}\right) = \sum_{i,j} \langle j|p_i \left|i\right\rangle \left\langle i\right| \left|j\right\rangle = \sum_{i,j} p_i \langle j|i\rangle \langle i|j\rangle = \sum_i p_i = 1 \tag{2.6}$$

The density operator can also be constructed for a pure state. For in a pure state, you can just choose a set of states only containing that pure state and set the probability of being in that state to 1. Therefore, the density operator for a system in the pure state $|\phi\rangle$ follows readily from the previous definition (2.4):

$$\hat{\rho} = |\phi\rangle \langle\phi| \tag{2.7}$$

But how do we get these statistical probabilities p_i when we know that our system is in an incoherent mixture? That usually depends on the physical problem under consideration. Two important examples are: [1]

- The system is a subsystem of a larger system.
- The system is described by equilibrium statistical mechanics.

We will be focusing on the first of these two examples. Let's go back to the system under consideration from the previous section. Here we had two subsystems with Hilbert spaces $\mathcal{H}_1^N$ and $\mathcal{H}_2^M$ respectively, which were combined to form a larger system with Hilbert space $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$. If the joint system is in a pure state and we know its state vector, then we can easily determine its density operator from (2.7). How do we now determine the density operator for one of the subsystems, say system 1? For this purpose, it is insightful to look at the expectation value of an operator which only acts in $\mathcal{H}_1^N$. [1] This operator has the form $\hat{A} \otimes \mathbb{1}_2$. And in analogy to (2.5), we want the expectation value for this operator to obey:

$$\langle \hat{A} \otimes \mathbb{1}_2 \rangle = \langle \hat{A} \rangle = \mathrm{Tr}\left(\hat{\rho}_1 \hat{A}\right) \tag{2.8}$$

Where $\hat{\rho}_1$ is the density operator for system 1. Starting from this expectation value, we find:

$$\begin{aligned}
\langle \hat{A} \rangle = \mathrm{Tr}\left(\hat{\rho}\left(\hat{A} \otimes \mathbb{1}_2\right)\right) &= \sum_{i,j} \left(\langle i| \otimes \langle j|\right) \hat{\rho} \left(\hat{A} \otimes \mathbb{1}_2\right) \left(|i\rangle \otimes |j\rangle\right) \\
&= \sum_{i,j} \left(\langle i| \otimes \langle j|\right) \hat{\rho} \left(\hat{A}\,|i\rangle \otimes |j\rangle\right) = \sum_{i,j} \langle i| \left(\mathbb{1}_1 \otimes \langle j|\right) \hat{\rho} \left(\mathbb{1}_1 \otimes |j\rangle\right) \hat{A}\,|i\rangle \\
&= \sum_{i} \langle i| \,\hat{\rho}_1 \hat{A}\, |i\rangle = \mathrm{Tr}\left(\hat{\rho}_1 \hat{A}\right)
\end{aligned}$$

So we recover (2.8) if we define the density operator for system 1 to be:

$$\hat{\rho}_1 = \sum_j \left(\mathbb{1}_1 \otimes \langle j|\right) \hat{\rho} \left(\mathbb{1}_1 \otimes |j\rangle\right) = \mathrm{Tr}_2\left(\hat{\rho}\right) \tag{2.9}$$

This also turns out to be the definition of the density operator for system 1, also called the reduced density operator, and the operation Tr_2 is called the partial trace over $\mathcal{H}_2^M$. [1, 3]

While (2.9) is a correct way to calculate the partial trace, in practice it is often not very useful due to the large number of multiplications required. When we use matrices to represent the operators, there is a faster way to determine the partial trace. Once again, this method follows easiest from an example. Let's say we have $N = 2$ and $M = 3$ so that the density matrix will have dimensions 6 × 6. We choose our $\{|j\rangle\}$ to be the basis states that were also used in the representation of the density matrix, so that:

$$|j\rangle = \begin{pmatrix} \delta_{1j} \\ \delta_{2j} \\ \delta_{3j} \end{pmatrix}$$

Then we find:

$$
\begin{aligned}
\mathrm{Tr}_2\left(\hat{\rho}\right) &= \sum_j \left(\mathbb{1}_1 \otimes \langle j|\right) \hat{\rho} \left(\mathbb{1}_1 \otimes |j\rangle\right) \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \\
&+ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \\
&+ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} a_{11} + a_{22} + a_{33} & a_{14} + a_{25} + a_{36} \\ a_{41} + a_{52} + a_{63} & a_{44} + a_{55} + a_{66} \end{pmatrix}
\end{aligned} \tag{2.10}
$$

So we can calculate the partial trace by subdividing the density matrix in blocks of 3 × 3 and taking the trace over these. This approach is correct for all (finite) Hilbert space dimensions as long as you are taking the partial trace over the second space of states (to the right of the tensor product), but then, of course, you have to subdivide the density matrix into blocks of $M \times M$. [3]

Now we have a way of finding the density operator for system 1. The easiest way then to determine the probabilities p_i that describe the incoherent mixture of system 1, is by determining the diagonal form of its density operator. For in the diagonal form we have:

$$
\begin{pmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & p_N \end{pmatrix} = \sum_i p_i \left|i\right\rangle \left\langle i\right|, \qquad \left|i\right\rangle = \begin{pmatrix} \delta_{1i} \\ \delta_{2i} \\ \vdots \\ \delta_{Ni} \end{pmatrix}
$$

So if we choose our set of states to be the eigenstates of the density operator and diagonalize the density matrix, then the probabilities of system 1 being in one of those states can be read off from the diagonal of the matrix. This can be done for any density operator, so any system can be described as an incoherent mixture of the eigenstates of its density operator. [4] But what then if we have a system in a pure state, how can such a system also be described by an incoherent mixture? The answer is that if a system is in a pure state, then its density operator will have one eigenvalue equal to 1 and all others equal to 0. [4] So one of the eigenstates of the density operator will then simply be that pure state and the system will have a probability of 1 of being in that state.

As an illustration of the above, let's try to determine the density matrix for the left spin of the two spins 1/2 from our example in section 2.1. This system was in the following state:

$$
\left|\Phi\right\rangle = \frac{1}{\sqrt{2}} \left(\left|+\right\rangle \otimes \left|+\right\rangle + \left|-\right\rangle \otimes \left|-\right\rangle\right), \qquad \left|+\right\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \left|-\right\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}
$$

Using this representation for the basis, we find:

$$
\begin{aligned}
\left|\Phi\right\rangle &= \frac{1}{\sqrt{2}}\left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix}+\begin{pmatrix}0\\0\\0\\1\end{pmatrix}\right) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}\\
\hat{\rho} = \left|\Phi\right\rangle\left\langle\Phi\right| &= \frac{1}{2}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 0 & 0 & 1\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\1 & 0 & 0 & 1\end{pmatrix}\\
\hat{\rho}_1 = \mathrm{Tr}_2\left(\hat{\rho}\right) &= \frac{1}{2}\begin{pmatrix}1+0 & 0+0\\0+0 & 0+1\end{pmatrix} = \begin{pmatrix}\frac{1}{2} & 0\\0 & \frac{1}{2}\end{pmatrix}
\end{aligned}
$$

Fortunately, the density matrix for system 1 is already in the diagonal form in this basis and we can immediately see that system 1 is in an incoherent mixture with equal probabilities of being in $\left|+\right\rangle$ and $\left|-\right\rangle$.

3 Introduction to the Renormalization Group

Many interesting problems in physics are characterised by the fact that they involve a large number of degrees of freedom. Think of solids, liquids or gases which in macroscopic quantities contain on the order of 10^{23} electrons. Advancements in computers and diagonalization algorithms have made it possible for physicists to calculate the properties of larger and larger systems, but even with the best supercomputer available today one cannot hope to solve the Schrödinger equation for systems containing that many electrons. You can usually reduce these degrees of freedom enormously by making use of the intensive and extensive character of observables, which allow you to reconstruct the properties of a macroscopic system from a microscopic sample. [5] But how far can you reduce the size of your sample before it will no longer accurately describe the properties of the macroscopic system? This will depend on the state of the system and we call the minimum size one can reach the correlation length ξ. [5] If you are lucky, then the correlation length will only be on the order of a few atomic spacings and then there is a large variety of methods to choose from, perturbation expansions and Hartree-Fock methods for example. [5] In some cases, however, the correlation length will far exceed a few atomic spacings and these systems are characterised by having very many degrees of freedom in a single correlation length, possibly even an infinite amount. The Renormalization Group was developed to study these kinds of problems. While an introduction to the full theory behind the Renormalization Group is beyond the scope of this thesis, discussing some of the key ideas behind it might help put the Density Matrix Renormalization Group into context.

3.1 Central goals and ideas

It can be said that the Renormalization Group has two main goals: [5]

1. Simplify the task of solving systems with many degrees of freedom within a single correlation length.
2. Explain how the qualitative features of cooperative behaviour arise.

The first of these objectives is achieved by introducing a new and smaller set of effective degrees of freedom to replace the original microscopic ones. [5] This is similar to what is done in electrostatics, for example, where you describe a charged piece of material by a smooth charge density $\rho\left(\vec{r}\right)$, instead of specifying the charge and position of all microscopic particles.

There are several ways of obtaining these effective degrees of freedom. One often used approach is to determine them in an iterative manner, where each step some of the original degrees of freedom are thinned out. [5] Say, for example, that our original microscopic system is a chain of spins with spacing L_0 between each spin. And our correlation length is much larger than this spacing. Each spin here represents a degree of freedom, so the original spacing between degrees of freedom is L_0. And say that we want to reduce the number of degrees of freedom by a factor of 2 each step. Then the spacing between degrees of freedom will be $2L_0$ after a single renormalization step, $4L_0$ after two steps and 2^nL_0 after n steps. These steps are carried out until the spacing between degrees of freedom is on the order of the correlation length. [5] At each step we will have to determine the effective interactions between our new effective degrees of freedom. Say that the Hamiltonian $\hat{H}_0$ describes the interactions between our original spins spaced by L_0, then after a single step, we construct an interaction $\hat{H}_1$ which describes the effective interaction between pairs of spins spaced by $2L_0$. And after two steps we find an interaction $\hat{H}_2$ which describes the effective interaction between pairs of pairs of spins spaced by $4L_0$ and so on and so forth. We, therefore, end up with a sequence of interactions $\hat{H}_0$,$\hat{H}_1$,$\hat{H}_2$,..., where the last Hamiltonian in the sequence describes the effective interactions between effective degrees of freedom with spacing on the order of the correlation length. One hopes that these interactions are all local interactions. [5] In other words, the range of the interactions described by $\hat{H}_n$ are on the order of 2^nL_0. So, while the correlation length might be much larger than L_0, to determine the interaction $\hat{H}_1$ you only have to consider a region of size $2L_0$, which limits the number of degrees of freedom you have to deal with. And while for $\hat{H}_2$ you have to consider a region of size $4L_0$, the idea is that you determine it from $\hat{H}_1$, whose degrees of freedom are spaced by $2L_0$, so that again the number of degrees of freedom remains limited. The same argument applies for any $\hat{H}_n$, which can be constructed from $\hat{H}_{n-1}$. Once an interaction $\hat{H}_N$ is reached for which 2^NL_0 is on the order of the correlation length, one should have a system with only a few degrees of freedom per correlation length. At this point, the problem is hopefully tractable enough to be solved by other methods, numerical diagonalization for example.

The construction of effective interactions is the same whether you are constructing $\hat{H}_1$ from $\hat{H}_0$ or $\hat{H}_5$ from $\hat{H}_4$. [5] In the Renormalization Group formalism this construction is often called a transformation. So there is a transformation τ for which we have:

$$\tau\left(\hat{H}_0\right) = \hat{H}_1, \quad \tau\left(\hat{H}_1\right) = \hat{H}_2, \quad \tau\left(\hat{H}_2\right) = \hat{H}_3, \quad \text{etc}$$

The second goal of the Renormalization Group, examining the qualitative features of cooperative behaviour, is obtained by studying the properties of this transformation. [5] One important property, for example, is the existence of so-called fixed points, which are interactions for which we have:

$$\tau\left(\hat{H}^*\right) = \hat{H}^*$$

These fixed points are often the limit of the sequence of interactions we get by repeatedly applying the transformation τ. The existence of these fixed points can give important information about different phases in your system and the location of phase transitions. A more concrete example of what these transformations might look like will be given next.

3.2 Truncated Diagonalization

Many different applications of the Renormalization Group and their corresponding transformations have been developed over the years. The one most relevant in the context of the Density Matrix Renormalization Group is known as the Real-Space Numerical Renormalization Group and was developed by Kenneth G. Wilson for solving the Kondo problem. [6] The specific details of the Kondo problem or the transformation used will not be relevant here, but the Density Matrix Renormalization Group will follow naturally from the general ideas behind Wilson's procedure.

The Kondo problem describes a system of a magnetic impurity embedded in a non-magnetic metal. [6] Wilson's approach was to build up the system iteratively, starting small with just the impurity state and adding successive layers of the metal's conduction band states in each step. [6] For this, a recursion relation was set up, relating the Hamiltonian $\hat{H}_n$ to the Hamiltonian of the previous step $\hat{H}_{n-1}$. This recursion relation was in principle exact, but if an exact treatment would be used in each step, then the problem would quickly become intractable as the dimensions of the Hamiltonian grow too quickly. In order to solve this problem, Wilson proposed a method of truncated diagonalization. Instead of using the exact representation of the interaction $\hat{H}_{n-1}$ from the previous step, the Hamiltonian was to be projected onto a truncated basis, thereby thinning out some of the degrees of freedom. [6] This raises the question of which states to use for this truncation. If we want to capture the low energy behaviour of the system, it may seem a logical choice to find the eigenstates of the interaction $\hat{H}_{n-1}$ and use the lowest lying eigenstates as a basis for the truncation, since these states should be dominant at low energies. This was also exactly the truncation that Wilson suggested in his treatment of the Kondo problem. [6] The truncated representation of $\hat{H}_{n-1}$ would then be used in the recursion relation to construct the next effective interaction $\hat{H}_n$. In summary, a single renormalization step in the Numerical Renormalization Group consists of the following operations:

1. Determine $\hat{H}_n$ from the truncated representation of $\hat{H}_{n-1}$ via the recursion relation.
2. Diagonalize $\hat{H}_n$ to find its eigenstates and eigenenergies.
3. Construct a truncation operator $\hat{O}$ whose columns are a few of the lowest lying eigenstates of $\hat{H}_n$.
4. Determine the truncated representation of $\hat{H}_n$ by applying the truncation operator: $\hat{O}^\dagger \hat{H}_n \hat{O}$. This truncated representation will become the starting point of the next step.

The number of states to use in the truncation is usually determined at the start and remains constant throughout the calculation. [6] This single renormalization step defines the transformation τ in the Numerical Renormalization Group. [6] By repeatedly applying this transformation while building up the system, an effective Hamiltonian can be constructed for the full system with a manageable amount of effective degrees of freedom.

3.3 Numerical Renormalization Group for quantum lattices

While the Numerical Renormalization Group was very successful at solving the Kondo problem, it quickly became apparent that it performed very poorly for other, seemingly similar problems, especially many quantum lattice problems. [7] It turns out that there are two main difficulties which cause the Numerical Renormalization Group to yield unsatisfactory results for quantum lattices. The first difficulty has to do with using the lowest lying eigenstates for the truncation. While it is true that the higher lying eigenstates contribute little to the low energy behaviour of your full system, they are still responsible for a lot of interesting physics. All of this is lost when these states are discarded. But the main culprit is that the Numerical Renormalization Group approach introduces very large boundary errors. This is easy to see in an example given by Steven R. White in which a 1D tight binding lattice is studied in the framework of the Numerical Renormalization Group.

In this model, the Hamiltonian contains an on-site energy of 2 and a hopping term of -1: [7]

$$\hat{H}_{ij} = \begin{cases} 2 & \text{if } i = j \\ -1 & \text{if } |i-j| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

If two blocks with an equal number of sites are combined to form a new larger block of the chain in the renormalization step, then the recursion relation can be written as:

$$\hat{H}_n = \begin{pmatrix} \hat{H}_{n-1} & \hat{T}_{n-1} \\ \hat{T}_{n-1}^\dagger & \hat{H}_{n-1} \end{pmatrix} \tag{3.2}$$

$$\hat{T}_n = \begin{pmatrix} 0 & 0 \\ \hat{T}_{n-1} & 0 \end{pmatrix} \tag{3.3}$$

Here the initial interactions are for a block containing a single site: $\hat{H}_0 = \begin{pmatrix} 2 \end{pmatrix}$ and $\hat{T}_0 = \begin{pmatrix} -1 \end{pmatrix}$. In the Numerical Renormalization Group, this model would be solved by iteratively constructing the next interaction via (3.2) and (3.3) and then using the eigenstates of this Hamiltonian to truncate it before proceeding with the next step. However, this method will perform poorly here and it is fairly easy to see why by looking at the low lying eigenstates of a block. For this model describes a finite discretization of the 1D particle-in-a-box and for very large blocks its eigenstates will be particle-in-a-box states. At the boundaries, the connections to surrounding blocks are ignored and this translates to requiring the wave function to be 0 at sites just outside a block. The problems that this causes for describing the low-energy behaviour of larger blocks can be seen in figure 3.1. As can be seen in this figure, every state for the larger block that is composed of low lying eigenstates of the smaller block will have a kink at the boundary of the two blocks, while the true ground state of the larger block actually has its maximum at the boundary. The use of periodic boundary conditions will not fix this problem, because this results in the requirement that the eigenstates are equal at the boundaries and only the ground state of large blocks can be accurately represented by this, which is constant everywhere for periodic boundary conditions. [7] Clearly, the Real-Space Numerical Renormalization Group is unsuitable for dealing with these types of models and some other method must be devised.

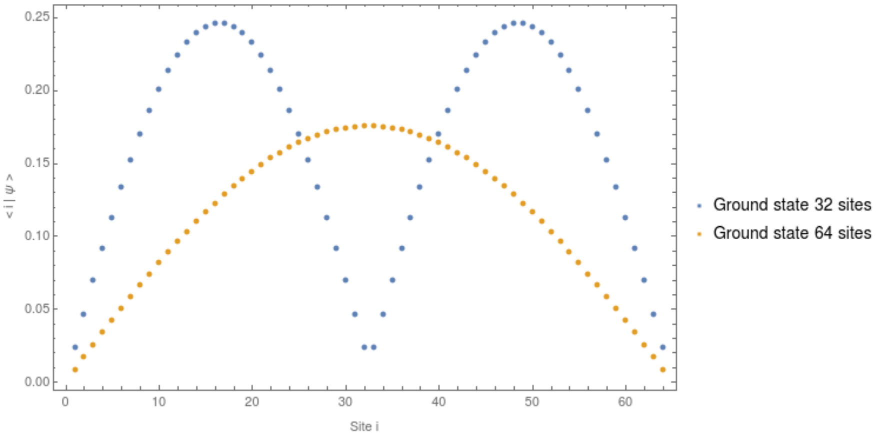


Figure 3.1: *Ground states of the 1D tight binding chain for two adjacent blocks of 32 sites and one block of 64 sites. All operators and states were represented by the position basis, where a state $|i\rangle$ represents a state on site i.*

4 The Density Matrix Renormalization Group

The Density Matrix Renormalization Group (DMRG) was developed by Steven R. White in 1992 as an improvement over the Real-Space Numerical Renormalization Group for solving quantum lattice models. [8] It improved upon the Numerical Renormalization Group approach in two ways. Firstly, it solved the boundary problems by not studying isolated blocks, instead blocks are embedded in an environment by connecting them to another block. This embedding accounts for the fact that in a larger system subsystems are also not isolated but connected to their environment. Secondly, the eigenstates of the Hamiltonian of a block are no longer used as the basis for truncation. White proposed that it would be better to use the eigenstates of the density operator. [8] Hence, the name Density Matrix Renormalization Group. It will turn out that this choice indeed leads to a truncated representation which is optimal in some sense.

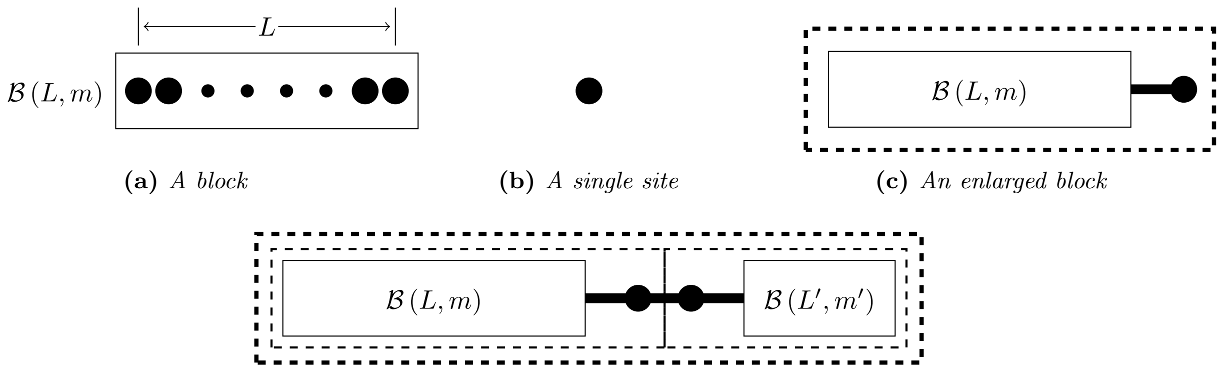
In this section, the Density Matrix Renormalization Group method is discussed for one-dimensional lattices. We will take a practical approach to this and the emphasis will be on the implementation of the method. Nevertheless, we will also briefly touch on the rich theory of quantum information which has been developed to understand how, why and when the Density Matrix Renormalization Group method works. We will now introduce some important concepts and notation which form the building blocks of the Density Matrix Renormalization Group.

4.1 The building blocks

There are two elementary units which are used to describe the system under consideration. These are the block and the single site. A single site represents an isolated site of the lattice. [9] It could be, for example, a single spin, when you have a chain of spins. Depending on the details of the system, the single site lives in a D-dimensional Hilbert space $\mathcal{H}^D_{\text{site}}$. The single site is always described by an exact basis. A block is a contiguous segment of the lattice. [9] It is a chain of several connected sites. We write $\mathcal{B}\left(L, m\right)$ to denote a block containing L sites, which lives in an m-dimensional Hilbert space $\mathcal{H}^{m}_{\text{block}}$. A block could contain just a single site, but it cannot be empty. The dimension m depends on the details of the system, but also on whether the block has been projected onto a truncated basis. Besides these two elementary units, there are also two important compound objects. An enlarged block is formed by connecting a block to single additional site. [9] If the single site is connected to the rightmost site of the block $\mathcal{B}\left(L, m\right)$, then the enlarged block lives in an mD-dimensional Hilbert space $\mathcal{H}^{mD}_{\text{left}} = \mathcal{H}^{m}_{\text{block}} \otimes \mathcal{H}^{D}_{\text{site}}$. This configuration is also referred to as the left enlarged block, for reasons which will become apparent later. The opposite configuration, with the single site connected to the leftmost site of the block, is also possible. Then the enlarged block lives in the Hilbert space $\mathcal{H}^{Dm}_{\text{right}} = \mathcal{H}^{D}_{\text{site}} \otimes \mathcal{H}^{m}_{\text{block}}$. If the block and the single site can be described by the bases $\{|b_i\rangle\}$ and $\{|j\rangle\}$ respectively, then the enlarged block can be described by the product basis $\{|b^e_k\rangle\}$:

$$|b^e_k\rangle = |b_i\rangle \otimes |j\rangle \tag{4.1}$$

Here, all the unique combinations of the indices i and j are mapped onto a different k. One example of such a mapping would be $k = i + m(j-1)$. This mapping is just a technicality to make the notation a bit less cumbersome. Lastly, there is the superblock, which is constructed by connecting two enlarged blocks to each other. [9] The configuration most often used is the one where the single site of a left enlarged block is connected to the single site of a right enlarged block. For this configuration, the superblock lives in the mD^2m'-dimensional Hilbert space $\mathcal{H}^{mD}_{\text{left}} \otimes \mathcal{H}^{Dm'}_{\text{right}}$. This is why one of the enlarged blocks is referred to as "left" and the other as "right".


L
$\mathcal{B}\left(L, m\right)$
$\mathcal{B}\left(L, m\right)$
$\mathcal{B}\left(L, m\right)$
$\mathcal{B}\left(L', m'\right)$

(a) *A block*

(b) *A single site*

(c) *An enlarged block*

(d) *A superblock*

Figure 4.1: *The building blocks of the Density Matrix Renormalization Group. The notation $\mathcal{B}\left(L, m\right)$ denotes a block with L sites represented by an m-dimensional basis. An enlarged block is composed of a block connected to a single extra site, as indicated by the dashed box in (c). Meanwhile, a superblock is composed of two connected enlarged blocks, as indicated by the dashed boxes in (d).*

Given the bases $\{|b_k^e\rangle\}$ and $\{|b_l^e\rangle\}$ of the left and right enlarged block, a state of the superblock can be written as:

$$|\psi\rangle = \sum_{k=1}^{mD} \sum_{l=1}^{Dm'} c_{kl}\, |b_k^e\rangle \otimes |b_l^e\rangle \tag{4.2}$$

These are the building blocks of the Density Matrix Renormalization Group. They are summarised in figure 4.1. Next, we will discuss the elementary renormalization step, which is at the heart of the algorithm.

4.2 The renormalization step

A single renormalization step in the DMRG algorithm consists of the growth of a block, followed by a projection onto a truncated basis to thin out some degrees of freedom. This is not unlike the procedure followed in the Numerical Renormalization Group. But, while in the NRG method a block is usually grown by connecting it to an identical block, leading to exponential growth, in the DMRG approach blocks are grown linearly by adding just a single extra site. Forming an enlarged block from the block of the previous iteration constitutes the first step in the renormalization. [8] In most cases, a block and a site will be represented by their Hamiltonians $\hat{H}_{\mathcal{B}(L,m)}$ and $\hat{H}_{\text{site}}$. Forming an enlarged block then corresponds to constructing its Hamiltonian. How to do this will depend on the system, but the procedure usually consists of adding the contributions of the block, the site and the interactions between them:

$$\hat{H}_e = \hat{H}_{\mathcal{B}(L,m)} \otimes \mathbb{1}_{\text{site}} + \mathbb{1}_{\mathcal{B}(L,m)} \otimes \hat{H}_{\text{site}} + \hat{H}_{\text{interactions}} \tag{4.3}$$

Here $\mathbb{1}_{\text{site}}$ and $\mathbb{1}_{\mathcal{B}(L,m)}$ represent the identity operators in the Hilbert spaces of the site and the block respectively. Constructing the interaction term often requires other, system specific operators, like spin operators for spin chains. If this is the case, then the equivalent operators for the enlarged block also have to be constructed in this step. The expression that relates the Hamiltonian of the enlarged block to the Hamiltonian of the block from the previous iteration can be seen as the recursion relation that we encounter more often in Renormalization Group methods.

If we were doing a Numerical Renormalization Group calculation, then we now would diagonalize the enlarged block to find its lowest lying eigenstates and project all the operators onto that truncated basis. But as mentioned in section 3.3, this method performs poorly for most quantum lattice models. So, instead, we want to account for the fact that our enlarged block isn't isolated in the final system. We do this by embedding the system in an environment. This is accomplished by connecting the enlarged block to another enlarged block. [8] So we embed our system in an environment by forming a superblock. Once again, the details of the system will dictate how to form a superblock, but it also usually involves adding the Hamiltonians of both enlarged blocks and the interactions between them:

$$\hat{H}_s = \hat{H}_e^{\text{left}} \otimes \mathbb{1}_e^{\text{right}} + \mathbb{1}_e^{\text{left}} \otimes \hat{H}_e^{\text{right}} + \hat{H}_{\text{interactions}} \tag{4.4}$$

$\mathbb{1}_e^{\text{right}}$ and $\mathbb{1}_e^{\text{left}}$ are the identity operators of the right and left enlarged block respectively. It is customary to regard the left block as the system and the right one as the environment. [10] The enlargement step then consists of adding a site to the right of a block to form a left enlarged block. In principle, the right block also has to be constructed from scratch. But, many lattice models are symmetric when spatially reflected through their centre, including the models that will be studied in this thesis. If this is the case, then it suffices to reflect a left block of the appropriate size to get the right block. Otherwise, both blocks will have to be constructed independently, which will roughly double the amount of work required.

We now want to find the optimal way to truncate the left enlarged block, while it is embedded in a larger system. The DMRG algorithm is usually used to extract information about a specific eigenstate of the system, which is referred to as a target state. [8] This means that we want to truncate the left enlarged block in such a way that it still accurately represents the target state of the full system. One, therefore, first needs to diagonalize the superblock Hamiltonian to find the state vector for this target state. Given that the superblock is in this state, we know from section 2.2 that the left enlarged block can be said to be in an incoherent mixture of the eigenstates of its density operator, with the weights given by the corresponding eigenvalues. Then, perhaps the eigenstates of the density operator with the largest weights would be suitable as a basis for truncation, since these are most important in describing the left enlarged block when the superblock is in the target state. This is exactly what White proposed when he first introduced the Density Matrix Renormalization Group. [8] To this end, we first construct the density matrix for the superblock in the usual manner, given that the superblock is in state $|\psi\rangle$:

$$\hat{\rho}_s = |\psi\rangle \langle\psi| \tag{4.5}$$

We now take the partial trace over the Hilbert space of the right enlarged block to find the reduced density matrix for the left enlarged block:

$$\hat{\rho}_{\text{left}} = \text{Tr}_{\text{right}} \left(\hat{\rho}_s\right) \tag{4.6}$$

In general, the reduced density matrix will not already be in diagonal form. So it needs to be diagonalized to find all its eigenstates $\{|u_\alpha\rangle\}$ and their corresponding weights w_α. We now have all the information needed to truncate the left enlarged block. The density operator eigenstates are ordered by descending weights and all but the first few eigenstates are discarded. A truncation operator $\hat{O}$ is now formed whose columns are these eigenstates with the largest weights. We then carry out the truncation:

$$\hat{H}_{\mathcal{B}(L+1,m')} = \hat{O}^\dagger \hat{H}_e \hat{O} \tag{4.7}$$

Here m' is the amount of density operator eigenstates we chose to keep and $\hat{H}_{\mathcal{B}(L+1,m')}$ is the starting block for the next renormalization step. If other operators were required to construct the enlarged block and superblock, then these also need to be projected onto the new basis. As one might recall from section 2.2, the trace of a density matrix is 1 and so the same holds for the sum of all the weights. The sum of the weights of the discarded states must then be equal to:

$$\epsilon_\rho = 1 - \sum_{\alpha=1}^{m'} w_\alpha \tag{4.8}$$

Here the weights are assumed to be in descending order of magnitude. This quantity is often referred to as the truncated weight and it is a measure for the error introduced by truncation. [10] There are different ways of choosing how many states to keep. In White's original paper a number m' of states to keep is chosen at the start and this is kept fixed throughout the algorithm. [8] But, an alternative way is to keep the truncated weight below a certain threshold value. At each renormalization step one can then determine the minimum number of states required to get a truncated weight below the threshold value and use this as the number of states to keep. We often want to keep the truncated weight below 10^{-4}, so this might be a possible choice for the threshold value. [9]

A few technical remarks with regard to the implementation of the algorithm. Many lattice models are described by Hamiltonians which can be pretty sparse. It can then be beneficial to store matrices in sparse form, which can cut down memory costs significantly. Since one is often interested in just a single eigenstate of the superblock, a lot of time can be saved by using sparse matrix diagonalization techniques, like the Lanczos algorithm, that allow the calculation of just a few eigenstates without having to determine the entire eigenspectrum. [9] The ground state is usually chosen as the target state, but this is not obligatory. [10] Any state that is accessible through the diagonalization technique used, can, in principle, be used as the target state. [10] This makes it possible to also use DMRG to extract information about excited states. One can even target multiple states simultaneously. The superblock is then said to be in an incoherent mixture of the targeted states, instead of in a pure state. [10] There is no well-defined procedure for determining the best choice of probabilities for this mixture, but people often simply weigh all targeted states roughly equally. [10] However, the best results are achieved when only a single state is targeted, since then the truncation is tailored to best represent that state. [8]

In summary, the renormalization transformation in DMRG is characterised by the following steps:

1. Construct an enlarged block by adding a site to the block of the previous iteration.
2. Construct a superblock by connecting the system block to an environment block.
3. Diagonalize the superblock to find the state vector of the target state and its corresponding energy.
4. Construct the density operator for the superblock.
5. Trace out the environment to get the reduced density operator for the system block.
6. Diagonalize the reduced density matrix to find its eigenstates and their corresponding weights.
7. Construct a truncation matrix from the m' density operator eigenstates with the largest weights.
8. Truncate all relevant operators.

As mentioned earlier, the system and environment block are the left (enlarged) and right (enlarged) block respectively.

We have already remarked that the truncated weight is a measure of the error introduced by truncation. This can be made a little more precise. Say that we want to know the expectation value of some local operator $\hat{A}$, which acts in the system block. Given that $\{|u_\alpha\rangle\}$ and w_α are eigenvectors and weights of the system's density operator, we know from section 2.2 that the expectation value of $\hat{A}$ is given by:

$$\begin{aligned}\langle\hat{A}\rangle = \mathrm{Tr}_{\mathrm{system}}\left(\hat{\rho}_{\mathrm{system}}\hat{A}\right) &= \sum_{\alpha,\beta}\langle u_\alpha|\, w_\beta\,|u_\beta\rangle\,\langle u_\beta|\,\hat{A}\,|u_\alpha\rangle \\ &= \sum_{\alpha,\beta} w_\beta\,\langle u_\alpha|u_\beta\rangle\,\langle u_\beta|\,\hat{A}\,|u_\alpha\rangle = \sum_{\alpha=1}^{N} w_\alpha\,\langle u_\alpha|\hat{A}|u_\alpha\rangle \qquad (4.9)\end{aligned}$$

Here N is the number of states before truncation. After the truncation, the expectation value will then be given by:

$$\langle\hat{A}\rangle_{\mathrm{approx}} = \sum_{\alpha=1}^{m'} w_\alpha\,\langle u_\alpha|\hat{A}|u_\alpha\rangle \qquad (4.10)$$

We, therefore, find an upper bound for the error in the expectation value:

$$\begin{aligned}\left|\langle\hat{A}\rangle - \langle\hat{A}\rangle_{\mathrm{approx}}\right| &= \sum_{\alpha=1}^{N} w_\alpha\,\langle u_\alpha|\hat{A}|u_\alpha\rangle - \sum_{\alpha=1}^{m'} w_\alpha\,\langle u_\alpha|\hat{A}|u_\alpha\rangle \\ &= \sum_{\alpha>m'}^{N} w_\alpha\,\langle u_\alpha|\hat{A}|u_\alpha\rangle \\ &\leq \max_\alpha\,\langle u_\alpha|\hat{A}|u_\alpha\rangle \sum_{\alpha>m'}^{N} w_\alpha = c_A\left(1-\sum_{\alpha=1}^{m'} w_\alpha\right) = c_A\epsilon_\rho \qquad (4.11)\end{aligned}$$

Here $c_A = \max_\alpha\,\langle u_\alpha|\hat{A}|u_\alpha\rangle$ is the maximum expectation value of $\hat{A}$ for one of the density operator eigenstates. So, the error in the expectation value can be said to be roughly proportional to the truncated weight. There is an additional correction factor of $(1-\epsilon_\rho)^{-1}$ to account for the fact that the norm of the target state $\langle\psi|\psi\rangle$ is no longer unity after truncation, but this correction is negligible when $\epsilon_\rho \to 0$. [10] In this sense, the Density Matrix Renormalization Group can be seen as a technique which optimises expectation values.

We can also look at the state vector after truncation. Before truncation, the state vector will be given by (see section 4.1):

$$|\psi\rangle = \sum_k\sum_l c_{kl}\,|b_k^e\rangle\otimes|b_l^e\rangle \qquad (4.12)$$

After projection of the left enlarged block onto a truncated basis of m' orthonormal states $\{|\alpha\rangle\}$, the approximate state vector is given by:

$$|\widetilde{\psi}\rangle = \sum_{\alpha}\sum_{l} a_{\alpha l}\,|\alpha\rangle \otimes |b_l^e\rangle \tag{4.13}$$

We then want to find a set of states which minimises the distance in the quadratic norm:

$$\Big|\Big||\psi\rangle - |\widetilde{\psi}\rangle\Big|\Big|^2 \tag{4.14}$$

It turns out that the solution to this problem is known from linear algebra and is given by a Singular Value Decomposition. [8] The optimal set of states are the eigenstates of the density operator with the largest weights and the distance in the quadratic norm is then given by: [10]

$$\Big|\Big||\psi\rangle - |\widetilde{\psi}\rangle\Big|\Big|^2 = 1 - \sum_{\alpha=1}^{m'} w_\alpha = \epsilon_\rho \tag{4.15}$$

So we see that the DMRG method indeed produces states which are optimal in some sense and in this language the algorithm can also be seen as a variational algorithm for the state vector. [10]

Unless the system block is embedded in the final and exactly described environment at each step in the algorithm, the truncation error will not be the only error. [10] Since this environment is not known at the start of the algorithm, the system is instead embedded in an environment which is smaller than the final environment at most steps and this environment is also described by an approximate truncated representation. This introduces additional errors, which are collectively referred to as environmental errors. [10] These errors are a lot harder to quantify and are often the reason that observables calculated by DMRG show errors much larger than the truncated weight. [10]

4.3 An example: The renormalization of the Transverse Field Ising Model

As an illustration of the renormalization transformation, we will now work through a full step of the DMRG algorithm for the Transverse Field Ising Model. More on the Transverse Field Ising Model can be found in section 5.1.2, but for now it is sufficient to know that our Hamiltonian has the form:

$$\hat{H} = -\sum_i \sigma_i^x \sigma_{i+1}^x - h\sum_i \sigma_i^z \tag{4.16}$$

Here σ^α are the Pauli spin matrices. We start the renormalization step with a block containing a single site, this is our system block. Its Hamiltonian is given by:

$$\hat{H}_{\mathcal{B}(1,2)} = -h\sigma^z = \begin{pmatrix} -h & 0 \\ 0 & h \end{pmatrix} \tag{4.17}$$

The Hamiltonian for a single site is identical:

$$\hat{H}_{\text{site}} = \begin{pmatrix} -h & 0 \\ 0 & h \end{pmatrix} \tag{4.18}$$

We now form the enlarged block by adding a site to the right of our system block. The site in our system block and the additional site will interact through their spins. This interaction can be written:

$$\hat{H}_{\text{interaction}} = -\sigma^x \otimes \sigma^x = -\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = -\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{4.19}$$

We can now determine the Hamiltonian of the (left) enlarged block:

$$
\begin{aligned}
\hat{H}_e^L &= \hat{H}_{\mathcal{B}(1,2)} \otimes \mathbb{1}_{\text{site}} + \mathbb{1}_{\mathcal{B}(1,2)} \otimes \hat{H}_{\text{site}} + \hat{H}_{\text{interaction}} \\
&= \begin{pmatrix} -h & 0 \\ 0 & h \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} -h & 0 \\ 0 & h \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} -h & 0 & 0 & 0 \\ 0 & -h & 0 & 0 \\ 0 & 0 & h & 0 \\ 0 & 0 & 0 & h \end{pmatrix} + \begin{pmatrix} -h & 0 & 0 & 0 \\ 0 & h & 0 & 0 \\ 0 & 0 & -h & 0 \\ 0 & 0 & 0 & h \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} -2h & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 2h \end{pmatrix}
\end{aligned} \tag{4.20}
$$

We will need the operator representing the rightmost spin of our enlarged block to determine the interactions in the superblock, so this also has to be constructed:

$$
\sigma_e^{x,L} = \mathbb{1}_{\mathcal{B}(1,2)} \otimes \sigma^x = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.21}
$$

For the environment we will use an identical but reflected enlarged block. So our superblock will contain four sites, two sites from both enlarged block. The Hamiltonian for the environment block will be identical after reflection of the system block, but we now need the operator representing the spin on the leftmost site:

$$
\sigma_e^{x,R} = \sigma^x \otimes \mathbb{1}_{\mathcal{B}(1,2)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{4.22}
$$

The interactions between both enlarged blocks will be due to the two adjacent spins at the connection between the blocks:

$$
\hat{H}_{\text{interaction}} = -\sigma_e^{x,L} \otimes \sigma_e^{x,R} \tag{4.23}
$$

This allows us to construct the superblock Hamiltonian:

$$
\begin{aligned}
\hat{H}_s &= \hat{H}_e^L \otimes \mathbb{1}_e^R + \mathbb{1}_e^L \otimes \hat{H}_e^R + \hat{H}_{\text{interaction}} \\
&= \begin{pmatrix} -2h & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 2h \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} -2h & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 2h \end{pmatrix} \\
&- \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix}
-4h & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -2h & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & -2h & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
-1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & -1 & 0 & -2h & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & 2h & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -2h & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 2h & 0 & -1 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2h & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 2h & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 4h
\end{pmatrix}
\end{aligned}
$$

This clearly shows just how fast these matrices grow for a simple lattice model. For simplicity, we will now set $h = 2$. If we were to keep h in the matrix, then the eigenvectors would become horribly complicated. We will target the ground state in this example. For this, we first diagonalize the superblock Hamiltonian. We find the following ground state and energy, given that $h = 2$:

$$
|\psi_0\rangle = \begin{pmatrix} 0.976082 \\ 0. \\ 0. \\ 0.122 \\ 0. \\ 0.0304289 \\ 0.123787 \\ 0. \\ 0. \\ 0.00911006 \\ 0.0304289 \\ 0. \\ 0.122 \\ 0. \\ 0. \\ 0.0154554 \end{pmatrix}, \qquad E_0 = -8.3768 \tag{4.24}
$$

The reduced density operator for the left enlarged block can now easily be calculated by first constructing the density operator for the superblock and then tracing out the environment:

$$\hat{\rho}_s = |\psi_0\rangle \langle\psi_0| \tag{4.25}$$

$$\begin{aligned}\hat{\rho}_L &= \mathrm{Tr}_R\left(\hat{\rho}_s\right)\\ &= \begin{pmatrix} 0.967619 & 0. & 0. & 0.120967 \\ 0. & 0.0162492 & 0.00404391 & 0. \\ 0. & 0.00404391 & 0.00100891 & 0. \\ 0.120967 & 0. & 0. & 0.0151228 \end{pmatrix}\end{aligned} \tag{4.26}$$

The last thing that we need to do before we can carry out the truncation, is to diagonalize this reduced density matrix:

Eigenstates	$\begin{pmatrix} -0.992276 \\ 0. \\ 0. \\ -0.12405 \end{pmatrix}$	$\begin{pmatrix} 0. \\ 0.970392 \\ 0.241536 \\ 0. \end{pmatrix}$	$\begin{pmatrix} 0. \\ 0.241536 \\ -0.970392 \\ 0. \end{pmatrix}$	$\begin{pmatrix} -0.12405 \\ 0. \\ 0. \\ 0.992276 \end{pmatrix}$
Weights	0.982742	0.0172557	2.35975×10^{-6}	4.14343×10^{-8}

Say that we want to keep two states. We then construct the truncation matrix as follows:

$$\hat{O} = \begin{pmatrix} -0.992276 & 0 \\ 0. & 0.970392 \\ 0. & 0.241536 \\ -0.12405 & 0. \end{pmatrix} \tag{4.27}$$

Lastly, we carry out the truncation:

$$\begin{aligned}\hat{H}_{\mathcal{B}(2,2)} &= \hat{O}^\dagger \hat{H}_e \hat{O}\\ &= \begin{pmatrix} -0.992276 & 0. & 0. & -0.12405 \\ 0 & 0.970392 & 0.241536 & 0. \end{pmatrix} \begin{pmatrix} -4. & 0. & 0. & -1. \\ 0. & 0. & -1. & 0. \\ 0. & -1. & 0. & 0. \\ -1. & 0. & 0. & 4. \end{pmatrix} \begin{pmatrix} -0.992276 & 0 \\ 0. & 0.970392 \\ 0. & 0.241536 \\ -0.12405 & 0. \end{pmatrix}\\ &= \begin{pmatrix} -4.12308 & 0 \\ 0 & -0.468768 \end{pmatrix}\end{aligned} \tag{4.28}$$

$$\begin{aligned}\sigma^x_{\mathcal{B}(2,2)} &= \hat{O}^\dagger \sigma_e^{x,L} \hat{O}\\ &= \begin{pmatrix} -0.992276 & 0. & 0. & -0.12405 \\ 0 & 0.970392 & 0.241536 & 0. \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} -0.992276 & 0 \\ 0. & 0.970392 \\ 0. & 0.241536 \\ -0.12405 & 0. \end{pmatrix}\\ &= \begin{pmatrix} 0. & -0.992859 \\ -0.992859 & 0. \end{pmatrix}\end{aligned} \tag{4.29}$$

This concludes the renormalization transformation. We constructed an effective interaction $\hat{H}_{\mathcal{B}(2,2)}$ for a block of two spins with only two degrees of freedom. The truncated weight for this truncation is 2.40119×10^{-6}, which is pretty good. $\hat{H}_{\mathcal{B}(2,2)}$ will serve as the starting point of the next renormalization step.

We are now ready to discuss how several renormalization steps are combined to calculate the properties of a quantum lattice model.

4.4 The infinite system algorithm

The first method proposed by White is known as the infinite system algorithm. [8] Its aim is to calculate the properties of the system in the thermodynamic limit, when the chain length goes to infinity. [8] The method is characterised by the simultaneous growth of the system and environment blocks. [8] At each step the system and environment will be of equal size. The algorithm starts with a system and environment block which both contain just a single site. The blocks are grown until the energy per lattice site or some other observable converges. Assuming the system is reflection symmetric, the algorithm can be formulated as follows: [8]

1. Start with two blocks containing a single site, these are the first system block and environment block.
2. Form two enlarged blocks by adding a site to the right of the system block and to the left of the environment block.
3. Form the superblock by connecting the system block to the environment block.
4. Diagonalize the superblock to find the state vector $|\psi\rangle$ and energy of the target state.
5. Form the reduced density matrix for the enlarged system block by first constructing the density matrix for the superblock $\hat{\rho}_s = |\psi\rangle \langle\psi|$ and then taking the partial trace over the environment $\hat{\rho}_L = \mathrm{Tr}_R\left(\hat{\rho}_s\right)$.
6. Diagonalize the reduced density matrix and order its eigenvectors by descending weight. Keep only the m' eigenvectors with the largest weights.
7. Form the truncation operator $\hat{O}$ from the m' eigenvectors with the largest weights and project all operators onto this truncated basis $\hat{A}_{L+1} = \hat{O}^{\dagger}\hat{A}_e\hat{O}$.
8. Reflect the system block to get the new environment block.
9. Repeat steps 2 - 8 until the algorithm has converged.

One step in the infinite system algorithm is depicted schematically in figure 4.2.

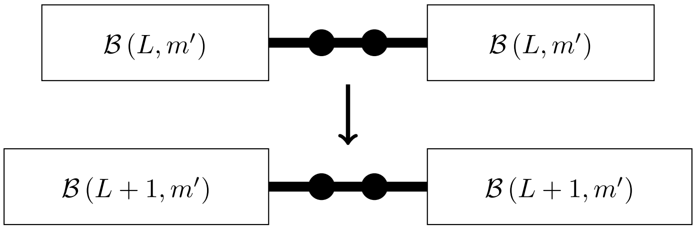


Figure 4.2: *A single step in the infinite system algorithm, which consists of the simultaneous growth of the system and environment block.*

4.5 The finite system algorithm

Complementary to the infinite system algorithm, White proposed the finite system algorithm as a means to accurately calculate the properties of finite size systems. [8] The first part of the algorithm is similar to the infinite system variant, but instead of growing both blocks until the algorithm has converged, we stop at a predetermined superblock length L. One could stop here and use the current blocks as an approximate representation of the finite system. But, practice has shown that this often leads to unsatisfactory results. [10] This is mainly due to the fact that at the early stages of the infinite system algorithm, the environment block is too small to accurately represent the embedding in the final system, which leads to large environmental errors. [10] To correct for this, White designed a method of sweeping over the finite system. During these sweeps, one proceeds with renormalization steps, but instead of growing both blocks, one block is grown at the expense of the other, keeping the superblock size constant. The renormalization transformation is applied to the growing block. This growth is continued until the shrinking block becomes just a single site. At this point, the situation is reversed and growth happens in the opposite direction. A single sweep consists of a full growth and shrinkage cycle for both blocks, so that for every size the corresponding block is updated once. We need a way to retrieve the operators for the shrinking block, so in the finite system algorithm all operators for all block sizes need to be stored. Whenever a renormalization is carried out for a certain block size, the operators representing this block have to be updated in storage. If the system is reflection symmetric, then the reflection of the left block at a certain size can be used to represent the right block at that size. In this case, the growth of the environment block can be replaced by simply setting the size of the system block back to 1 and proceeding with the growth of that block.

The idea behind these sweeps is that they will improve the representation of all the different size blocks, because they can now be embedded in the final system. [8] And this embedding only becomes better when the representations of blocks become more accurate. By repeatedly sweeping over the system, we are able to significantly reduce the environmental errors and bring down the total error to almost the truncation error. [10]

For a reflection symmetric system, the finite system algorithm can be formulated as follows: [8]

1. Carry out the steps of the infinite system algorithm until the superblock reaches a predetermined size L. The size of the system block should now be $l = L/2$. Blocks should be stored for all sizes, from $l = 1$ to $l = L/2$.
2. [Start of sweep] Grow the system block to size $l + 1$. Retrieve a block of size $L - l - 2$ from storage and reflect it to get an environment block. Grow the environment block to size $L - l - 1$.
3. Carry out steps 3 - 7 of the infinite system algorithm.
4. Store the truncated block as the new block $\mathcal{B}\,(l + 1, m')$.
5. Set $l = l + 1$.
6. Repeat steps 2 - 5 until $l = L/2 - 2$. At this point, set $l = 1$ and retrieve a block of size 1 to be the new system block.
7. [End of sweep] Repeat steps 2 - 5 until $l = L/2$. This concludes a sweep.
8. Repeatedly sweep over the system (steps 2 - 7) until the algorithm has converged.

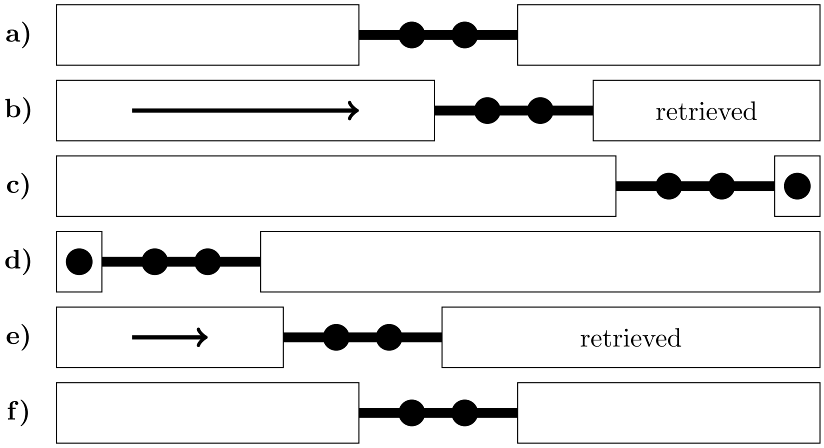


Figure 4.3: *A single sweep in the finite system algorithm for a reflection symmetric system. At the start of a sweep, the superblock is in a symmetric configuration as in (a). In (b) the left block is then grown at the expense of the right block. Here the right blocks are retrieved from storage. This growth continues until the right block becomes a single site as in (c). We then reflect the system so that the left block becomes a single site as in (d). In (e) the left block is grown again at the expense of the right block. Here the right blocks are again retrieved from storage. This growth continues until we arrive at the symmetric configuration in (f). This marks the end of the sweep. During a single sweep, the blocks get updated once for each size.*

A single sweep in the finite system algorithm is depicted schematically in figure 4.3. For spin models, less than 10 sweeps are usually required to reach convergence, while many more sweeps may be necessary for electronic models. [9] Sometimes, the algorithm might seem to have converged for several sweeps before suddenly improving further. [10] It is, therefore, advisable to first study the convergence behaviour of a model by running calculations for small system sizes, which are relatively inexpensive. One can then decide how to determine if the algorithm has truly converged.

4.6 Observables

Because the DMRG algorithm produces a state vector for the target state being studied, we are also able to calculate expectation values for observables other than the energy per site. This is done in the usual manner. Given that the superblock is in state $|\psi\rangle$, we calculate the expectation value for an operator $\hat{A}$ as:

$$\langle\hat{A}\rangle = \langle\psi|\hat{A}|\psi\rangle \tag{4.30}$$

This does require that we have a representation of the operator $\hat{A}$ in the same truncated basis as is being used for the superblock. For a local operator $\hat{A}_i$ acting on site i of the lattice, there are two cases that need to be discerned. The easiest one is the case where the operator is acting on one of the two additional single sites between the two enlarged blocks. So on of the sites that was added when forming an enlarged block. If this is the case, then we can simply construct the right representation by taking the Kronecker product between that operator and the appropriate identity matrices. Say, for example, that the operator is acting on the site that was added to the left block, then its representation in the superblock basis is given by:

$$\hat{A}_i^s = \mathbb{1}^L \otimes \hat{A}_i \otimes \mathbb{1}_{\text{site}} \otimes \mathbb{1}^R \tag{4.31}$$

Here $\mathbb{1}^L$, $\mathbb{1}_{\text{site}}$ and $\mathbb{1}^R$ are the identity matrices for the left block, single site and right block respectively. The other case is when the operator is acting in one of the two blocks. We now have to construct its representation alongside the block, keeping track of its representation at every step in the algorithm. Whenever the block is truncated, we also have to truncate the representation of the operator:

$$\hat{A}_{i,\mathcal{B}(L+1,m')} = \hat{O}^\dagger \hat{A}_{i,e} \hat{O} \tag{4.32}$$

And when a site is added to a block, the representation of the operator also has to be updated:

$$\hat{A}_{i,e} = \hat{A}_{i,\mathcal{B}(L,m')} \otimes \mathbb{1}_{\text{site}} \tag{4.33}$$

Since this requires additional computational effort, the construction is done usually as late as possible in the calculation. In the infinite system algorithm, this can be done by restricting oneself to sites near the free sites in the middle of the superblock, so that we only have to keep track of the operator representation in the last few steps before convergence. [9] And in the finite system case, the construction can be postponed until the last sweep, where you then start keeping track of the operator representation once the free site of the enlarged block is equal to the site where we want to evaluate the operator. [9] Once you have the representation of the operator in the basis of the block, its representation in the basis of the superblock is again constructed by taking the Kronecker product with the appropriate identity matrices. Assuming the operator acts in the left block:

$$\hat{A}_i^s = \hat{A}_{i,\mathcal{B}(L,m')} \otimes \mathbb{1}_{\text{site}} \otimes \mathbb{1}_{\text{site}} \otimes \mathbb{1}^R \tag{4.34}$$

For non-local operators the situation is a bit more complicated. There are four distinct cases for two-point correlators, for example. Analogous to the local operator case, the situation is easiest when the two points are the free sites in the middle of the superblock. The representation is then simply given by:

$$\left(\hat{A}_i \hat{A}_{i+1}\right)^s = \mathbb{1}^L \otimes \hat{A}_i \otimes \hat{A}_{i+1} \otimes \mathbb{1}^R \tag{4.35}$$

The second case is when one of the operators acts in one of the blocks and the other on one of the free sites. The representation of the operator acting in the block then has to be constructed the same way as for a local operator acting in that block. After this representation is constructed, the superblock representation for a two-point correlator where one operator acts in the left block and the other on the left free site, is given by:

$$\left(\hat{A}_i\hat{A}_j\right)^s = \hat{A}_{i,\mathcal{B}(L,m')} \otimes \hat{A}_j \otimes \mathbb{1}_{\text{site}} \otimes \mathbb{1}^R \tag{4.36}$$

We can also have a two-point correlator where one operator acts in the left block and the other in the right block. One then has to construct the representation for both operators in their respective blocks. If you have a reflection symmetric configuration (a reflection symmetric superblock, identical operators and the sites on which the operators act have the same distance to the middle of the superblock), then it suffices to determine the representation of the operator in only one of the blocks. The other representation can be found by reflection. The superblock representation is given by:

$$\left(\hat{A}_i\hat{A}_j\right)^s = \hat{A}_{i,\mathcal{B}(L,m')} \otimes \mathbb{1}_{\text{site}} \otimes \mathbb{1}_{\text{site}} \otimes \hat{A}_{j,\mathcal{B}(L',m'')} \tag{4.37}$$

When both operators act in the same block, the situation is most difficult. For then, it isn't sufficient to construct the representations of the operators in the basis of the block separately. They have to be treated as a compound object and their representation has to be constructed as such. [10] Since it can be quite complicated to keep track of such a compound object, it is often more convenient to restrict oneself to the other three cases. [8] Given the representation of the compound object in the left block, its superblock representation will be:

$$\left(\hat{A}_i\hat{A}_j\right)^s = \left(\hat{A}_i\hat{A}_j\right)_{\mathcal{B}(L,m')} \otimes \mathbb{1}_{\text{site}} \otimes \mathbb{1}_{\text{site}} \otimes \mathbb{1}^R \tag{4.38}$$

For higher-order correlators the situation will be similar, just with more difficult combinations. The general rule is that operators acting in the same block have to be treated as a compound object.

When expectation values are calculated in the finite system algorithm, the evaluation of (4.30) can, in principle, happen at each step in a sweep. Practice has shown, however, that the results will be most accurate when you have a symmetric superblock configuration, where both blocks have equal sizes. [8] It is, therefore, advisable to use this configuration.

As already has been discussed (see (4.11)), the error in observables will roughly be proportional to the truncation error. It is often found, however, that the error in the values of the observables are larger than the error in the energy for the same state vector. [9, 10]

4.7 Boundary conditions

We have yet to mention something about boundary conditions. The DMRG procedure described in the previous subsections assumes the use of open boundary conditions (OBC), which are used most often in DMRG. The reason for this is that practice has shown that with the same number of states kept the results obtained with open boundary conditions are much more accurate than those obtained with periodic boundary conditions, sometimes up to several orders of magnitude. [8] White's explanation for this lies in the nature of the density operator eigenstates. He argued that each eigenstate roughly corresponds to the response of the block to a particular quantum fluctuation in the rest of the chain. [8] And while with open boundary conditions, a block only has to react to the quantum fluctuations at one boundary, with periodic boundary conditions the block will have to respond to nearly independent fluctuations from both ends. [8] The effect of this, he concluded, was that m^2 states were required with periodic boundary conditions to achieve the same accuracy as when m states are used with open boundary conditions. [8]

From a physical perspective, however, periodic boundary conditions are preferred, since they allow us to reduce the impact of the undesirable edge effects which often plague open boundaries. White, therefore, also proposed a way to use periodic boundary conditions in DMRG. [8] For periodic boundary conditions, one has to use a different superblock configuration. Instead of having the free sites in the middle, the blocks are configured as in figure 4.4. If one were to use the same configuration as with open boundary conditions, then the two large blocks would be neighbours. This, in turn, can greatly reduce the sparseness of the Hamiltonian, which makes the calculation much more demanding. [8] Apart from this, the rest of the DMRG algorithm will remain largely the same. One only needs to make sure that the blocks are grown at the appropriate ends.

A final thing to mention is that analogous to reflection symmetry in open boundary conditions, translation symmetry can be used to prevent the need of independently constructing the blocks for the environment. If the system is translation symmetric, then the left blocks can also simply be used as right blocks. Contrary to systems with reflection symmetry, no reflection of blocks is required.

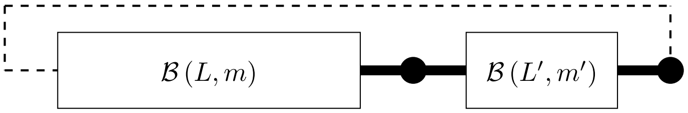


Figure 4.4: *The preferred superblock configuration when using periodic boundary conditions. The dashed line indicates that for periodic boundary conditions, the rightmost site of the right enlarged block is also connected to the leftmost site of the left enlarged block.*

4.8 Further improvements and extensions

We have now discussed everything necessary for someone to create their own basic implementation of the Density Matrix Renormalization Group algorithm. A lot can still be done, however, to improve the speed of the algorithm. Some of these techniques will be discussed shortly here.

The first widely used technique is the exploitation of symmetries in your system. We have already encountered one example of this: the use of reflection and translational symmetry. These symmetries allowed us to cut calculation time and storage requirements in half, by dispensing with the need of independently constructing the system and environment blocks. But other symmetries are also widely used. Symmetries that lead to the conservation of quantum numbers, for example, often allow us to write the operators in block form, where blocks correspond to symmetry sectors. [10] If the DMRG approach respects this conservation, then each symmetry sector can be treated and stored independently. [10] Once again, this might greatly reduce computation time and storage requirements. [10]

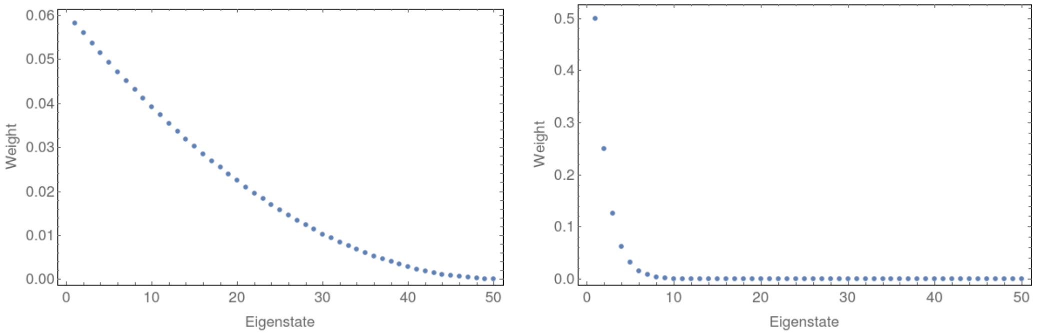
Another technique, which can be used in the finite system algorithm, makes use of the fact that the first iterations of the algorithm are only required to produce a good set of starting blocks. [9] Subsequent sweeps are then used to improve these starting blocks. Efficiency of the algorithm can then be improved by only keeping a few states in the first part of the algorithm and increasing this number when convergence is approached. [9]

The last technique we will mention here is used to speed up the convergence of iterative diagonalization algorithms. These methods will converge significantly faster when there is a large overlap between the starting vector and an eigenstate of the matrix. [10] So, instead of using a random starting vector, one can use the state vector of the previous renormalization step as the starting vector. [10] For the infinite system algorithm this poses a problem, however, because the size of the superblock changes each step. But, there is a technique which can transform the state vector of the previous iteration and get an accurate prediction of the state vector for the new superblock. [10]

Besides improving the speed and efficiency of DMRG, the method can also be extended to be used to solve other problems. One obvious extension, would be the extension to higher dimensions. While DMRG unfortunately displays its full force only for one-dimensional systems, much progress has been made to extend its use to two- and three-dimensional systems. [10] One can also look beyond static properties and try to determine dynamic properties instead. We then get in the territory of time-dependent DMRG, which is also widely used nowadays. [10] It can be said that the core of the DMRG algorithm turned out to be very versatile and it has even been extended to such fields as quantum chemistry, equilibrium and far-from-equilibrium problems and quantum statistical mechanics. [10]

4.9 When and why does DMRG work?

When White first introduced the Density Matrix Renormalization Group, he did provide some indications of why he thought density matrix eigenstates were a natural choice to use as the basis of truncation, and he showed that this choice leads to a minimisation of the distance of the quadratic norm of the wave function before and after truncation, as was already discussed in section 4.2. But there was not yet a solid theoretical background which could predict how well DMRG would perform for a certain system. Practitioners of DMRG usually resorted to an empirical survey of the algorithm's behaviour and used that to decide whether the results were reliable or not. [10] One could, for example, look at the decay of the density matrix eigenvalue spectrum. If it decays very slowly as in figure 4.5a, then it safe to say that DMRG will most likely perform poorly, since very many states would have to be kept to end up with an acceptable truncation error. And while a fast decaying spectrum as in figure 4.5b might not guarantee accurate results, it can be a good first indication. Since then, a lot of theoretical work has been done and it can be said that we now have a fairly good understanding of the theoretical foundations underlying DMRG. [10] Here we will briefly mention two aspects of this theory. For an in-depth discussion, the paper by Schollwöck is an excellent starting point. [10]



(a) *A slow decaying spectrum*

(b) *A fast decaying spectrum*

Figure 4.5: *Two possibilities for a density matrix eigenvalue spectrum for a system with a 50-dimensional Hilbert space, where the eigenvalues are in descending order. The spectrum in (a) was modelled after a polynomial function with the restriction that the sum of all weights must be equal to 1. The spectrum in (b) was modelled after an exponential function with the same restriction. Of course, these spectra paint a very idealised picture. There is no guarantee that the eigenvalue spectrum of an arbitrary system can be modelled as such a nice analytical function.*

The DMRG algorithm produces ansatz states of a particular kind. So-called matrix product states. [10] A matrix product state is a generalisation of a product state (a state written in the product basis):

$$|\sigma\rangle = |\sigma_1\rangle \otimes |\sigma_2\rangle \otimes \cdots \otimes |\sigma_L\rangle \tag{4.39}$$

This generalisation is achieved by introducing state-dependent linear operators $\hat{A}_i\left[\sigma_i\right]$, where these operators map from one M-dimensional auxiliary state space $\{|\beta\rangle\}$ to another M-dimensional auxiliary state space $\{|\alpha\rangle\}$: [10]

$$\hat{A}_i\left[\sigma_i\right] = \sum_{\alpha,\beta} \left(\hat{A}_i\left[\sigma_i\right]\right)_{\alpha\beta} |\alpha\rangle \langle\beta| \tag{4.40}$$

Furthermore, these operators have to obey: [10]

$$\sum_{\sigma_i} \hat{A}_i\left[\sigma_i\right] \hat{A}^{\dagger}\left[\sigma_i\right] = \mathbb{1} \tag{4.41}$$

For periodic boundary conditions, a matrix product state for an L-site system can then be written as:

$$|\psi\rangle = \sum_{\{\sigma\}} \mathrm{Tr}\left(\prod_{i=1}^{L} \hat{A}_i\left[\sigma_i\right]\right) |\sigma\rangle \tag{4.42}$$

With this knowledge, we can then conclude that DMRG will perform well if a true state of the system can be well approximated by this kind of states. [10]

Another approach to understanding why DMRG works comes from quantum information theory. White already argued that one of the reasons that DMRG works well is that it doesn't treat blocks as being isolated, but as being connected to a larger system. [8] As it turns out, by choosing the density operator eigenstates with the highest weights, you are maximising the preservation of entanglement between system and environment. [10] We will not give a detailed description of entanglement here, but it can be noted that entanglement can be measured by the von Neumann entropy: [1]

$$S_{vN} = -\mathrm{Tr}\hat{\rho}\ln\hat{\rho} \tag{4.43}$$

This is the quantity that is being maximised. [10]

Hopefully, this brief overview has given some insights in the rich theoretical foundations underlying DMRG.

5 The Ising Model

5.1 Theory

One of the standard and most well-known models in Statistical Mechanics that has been studied extensively in the context of phase transitions, is the Ising Model. It derives its name from Ernst Ising, who solved the model in one dimension as part of his doctoral research, but as Ising mentions in this 1924 dissertation, the model was actually suggested earlier in 1920 by his research director, Wilhelm Lenz. [11] Originally intended as a model to study ferromagnetism, the Ising model gained widespread popularity as a standard theory for studying cooperative behaviour. [11] Is has since seen applications in many fields, including physics, metallurgy, chemistry and even biology. [11]

5.1.1 The model

In the context of magnetism, the Ising model can be formulated as a system of L particles on a lattice, where each particle has a spin and interacts with its nearest neighbours through this spin. The Hamiltonian for the system can be written as: [11]

$$H = -\frac{J}{2}\sum_{i=1}^{L}\sum_{\langle i,j\rangle} S_i S_j - \Gamma\sum_{i=1}^{L} S_i \tag{5.1}$$

Here S_i denotes a spin on site i which can be either up $S_i = +1$ or down $S_i = -1$. $\sum_{\langle i,j\rangle}$ is the sum over the nearest neighbours of site i and the factor of $1/2$ is included to make sure that every pair is only counted once. J is the coupling between neighbouring spins. If $J > 0$, neighbouring particles prefer to align their spins in the same direction, while for $J < 0$, particles prefer to have their spins anti-parallel. Γ is an external field which tends to align the spins in the same direction as the field. For a one-dimensional chain, a possible configuration of the model can be seen in figure 5.1.

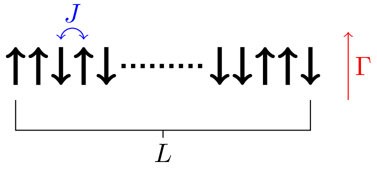


Figure 5.1: *An example configuration of a one-dimensional Ising chain containing L spins. The interaction J between neighbouring spins and the external field Γ have also been indicated.*

This form of the Ising model has mainly been used to study the phenomenon of ferromagnetism, which leads to the spontaneous alignment of spins in a material below a certain temperature. Ising showed in his dissertation that the one-dimensional Ising chain shows no phase transition at finite temperature between an ordered ferromagnetic phase and a disordered phase. [11] The intuition behind this is that if we were to flip only one spin in an ordered phase of the Ising chain, then the left half would become completely disconnected from the right half. And since some spins will always be randomly flipped at finite temperatures, any long range order will be unstable. [11] It has been shown that, contrary to the one-dimensional chain, higher-dimensional Ising models do exhibit phase transitions. [11] The most prominent example of this is Lars Onsager's exact solution of the two-dimensional square lattice Ising model. Despite much effort, Onsager's solution has never been extended to higher dimensions. Nevertheless, the existence of a phase transition in the three-dimensional model has been verified through numerical techniques. [12]

The Ising model can also be treated as a quantum mechanical model. Classical spins will be replaced by one of the Cartesian components of the quantum mechanical spin. If we choose this component to be the z-component, then the Hamiltonian operator can be written as:

$$\hat{H} = -\frac{J}{2}\sum_{i=1}^{L}\sum_{\langle i,j\rangle}\sigma_i^z\sigma_j^z - \Gamma\sum_{i=1}^{L}\sigma_i^z \tag{5.2}$$

Here σ_i^z is one of the Pauli spin matrices. We can easily determine the eigenstates of this Hamiltonian. They correspond to simple product states of the spin matrix eigenstates with one eigenstate for every lattice site:

$$|\psi\rangle = |\sigma_1^z\rangle \otimes |\sigma_2^z\rangle \otimes \cdots \otimes |\sigma_L^z\rangle\,, \qquad |\sigma_i^z\rangle \in \{|+\rangle\,, |-\rangle\}\,, \qquad \sigma_i^z\,|\pm\rangle = \pm\,|\pm\rangle \tag{5.3}$$

There is a variant of this model which introduces interesting new properties and behaviour. This variant is known as the Transverse Field Ising Model (TFIM) and will be discussed in the next section.

5.1.2 Ising Model with a Transverse Field

The Ising Model with a Transverse Field is obtained by rotating the external field over an angle of 90°, so that it is perpendicular to the direction of the interacting spin component. If we now choose that the particles interact through the x-component of their spins and that the external field is in the z-direction, then the Hamiltonian becomes:

$$\hat{H} = -\frac{J}{2}\sum_{i=1}^{L}\sum_{\langle i,j\rangle}\sigma_i^x\sigma_j^x - \Gamma\sum_{i=1}^{L}\sigma_i^z \tag{5.4}$$

The spin interaction J still tends to order the spins in the x-direction, while the inclusion of a transverse term in the Hamiltonian has the effect of flipping the x-component, destroying this order and thereby introducing quantum fluctuations. Because of this, the simple product states (5.3) will no longer be eigenstates of the model. For the remainder of this section, we will restrict ourselves to the one-dimensional Transverse Field Ising Chain. When going to one dimension, the Hamiltonian can be rewritten in a slightly simpler way:

$$\hat{H} = -J\sum_{i=1}^{L}\sigma_i^x\sigma_{i+1}^x - \Gamma\sum_{i=1}^{L}\sigma_i^z \tag{5.5}$$

By letting the first sum run to L, we are assuming periodic boundary conditions, so that $\sigma_{L+1}^x = \sigma_1^x$. If we wanted to use open boundary conditions, the sum would have to stop at $L-1$. If we introduce the new variable $h \equiv \Gamma/J$, we can factor out the J:

$$\hat{H} = -J\left(\sum_{i=1}^{L}\sigma_i^x\sigma_{i+1}^x + h\sum_{i=1}^{L}\sigma_i^z\right) \tag{5.6}$$

When using periodic boundary conditions, this model can actually be solved analytically by mapping it to a chain of spinless fermions, a so-called Jordan-Wigner transformation. Here we will just mention that the ground state energy is given by: [13]

$$E_0 = -\frac{1}{2} \sum_{n=-L/2}^{L/2-1} \epsilon\left(k_n\right) \tag{5.7}$$

Where: [13]

$$\epsilon\left(q\right) = 2J\sqrt{1+h^2-2h\cos\left(q\right)}, \qquad k_n = \frac{2\pi\left(n+1/2\right)}{L}, \qquad p_n = \frac{2\pi n}{L}, \qquad n = -\frac{L}{2}, \cdots, \frac{L}{2}-1 \tag{5.8}$$

Energies for excited states can be found by adding an even number of $\epsilon\left(k_n\right)$ to the ground state energy or an odd number of $\epsilon\left(p_n\right)$ to E_0^R, for distinct values of k_n and p_n, where E_0^R is given by: [13]

$$E_0^R = -\frac{1}{2} \sum_{n=-L/2}^{L/2-1} \epsilon\left(p_n\right) \tag{5.9}$$

An interesting property of the Transverse Field Ising Model is that it is one of the simplest one-dimensional models which shows a quantum phase transition at a finite external field. [14] This phase transition marks a change from a ferromagnetic phase, where the x-components of the spin are ordered, to a paramagnetic phase, where the x-components of the spin are disordered. [15] This transition happens at the point where the strength of the spin coupling and external field are equal, $J = \Gamma$ or equivalently $h = 1$. [15] At and near the critical point, where $h = 1$, the spin observables of the chain display characteristic behaviour. First introducing some quantities:

- Magnetic order parameters: $M_x = \langle\psi|\sigma_i^x|\psi\rangle$, $M_z = \langle\psi|\sigma_i^z|\psi\rangle$
- Correlation functions: $\rho_n^x = \langle\psi|\sigma_i^x\sigma_{i+n}^x|\psi\rangle$, $\rho_n^z = \langle\psi|\sigma_i^z\sigma_{i+n}^z|\psi\rangle - M_z^2$
- Mass gap: $\Delta = E_1 - E_0$

The order parameter and correlation functions could depend on where in the chain they are evaluated, but this dependency will go away in the limit of very large chains. Using the analytical solution of the Transverse Field Ising Chain, the behaviour of these quantities can be explored. The behaviour of the mass gap, for example, follows readily from (5.8). Since $q \in [-\pi, \pi]$, $\epsilon\left(q\right)$ will have its minimum at $q = 0$, the lowest excited state is the one where we add a single $\epsilon\left(p_0\right)$ to E_0^R. Given that $E_0 = E_0^R$ as the chain length goes to infinity, the mass gap is then given by:

$$\Delta = \epsilon\left(p_0\right) = 2J\sqrt{1+h^2-2h} = 2J\left|1-h\right| \tag{5.10}$$

We see that the mass gap goes to zero at the critical point. The behaviour of the other quantities can also be extracted from the analytical solution, but the derivation is more difficult. We will only mention the results here. At the critical point $h = 1$, the correlation functions are given by: [15]

$$\rho_n^x = \left(\frac{2}{\pi}\right)^n 2^{2n(n-1)} \frac{H\left(n\right)^4}{H\left(2n\right)} \tag{5.11}$$

$$\rho_n^z = \frac{4}{\pi^2}\frac{1}{4n^2-1} \tag{5.12}$$

Where:

$$H\left(n\right) = \prod_{i=1}^{n-1} i^{n-i} \tag{5.13}$$

Away from the critical point, we can look at the behaviour of the correlations in the x-component of the spin when we consider two spins which are infinitely far apart. The results are: [15]

$$\lim_{n\to\infty} \rho_n^x = \begin{cases} 0 & h > 1 \\ \left(1-h^2\right)^{1/4} & h < 1 \end{cases} \tag{5.14}$$

This confirms that in the ferromagnetic phase $h < 0$ there is long-range order, which disappears in the paramagnetic phase $h > 1$. Lastly, we can look at the order parameter for the magnetisation in the x-direction in the limit of infinite chains. It is given by: [15]

$$M_x = \begin{cases} 0 & h \geq 1 \\ \left(1 - h^2\right)^{1/8} & h < 1 \end{cases} \tag{5.15}$$

We will explore this phase transition with the Density Matrix Renormalization Group and see if it can reproduce these results.

5.2 DMRG treatment

The Density Matrix Renormalization Group algorithm, as described in section 4, was implemented in C++ for the Transverse Field Ising Chain with both open and periodic boundary conditions. For linear algebra functionality, the Armadillo C++ library was used. [16] For the calculations the Hamiltonian was divided by the coupling constant J, so that all energies are determined in units of J:

$$\hat{H}/J = -\sum_{i=1}^{L} \sigma_i^x \sigma_{i+1}^x - h \sum_{i=1}^{L} \sigma_i^z \tag{5.16}$$

5.2.1 Errors

The first thing that was explored, were the errors of the DMRG calculations and how they depended on certain parameters. To this end, we can first look at the density matrix eigenspectrum. The spectrum was calculated for a left enlarged block of size 6 connected to an identical block, so that the superblock contained 12 sites. The external field parameter was set to $h = 1$. Results of the calculations for both open and periodic boundary conditions can be seen in figure 5.2.

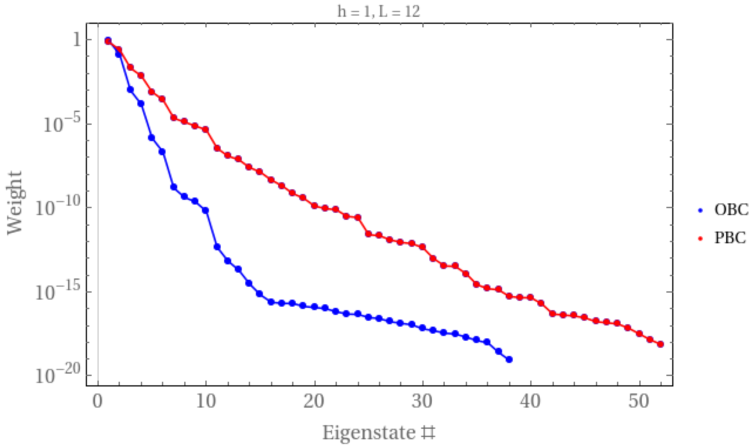
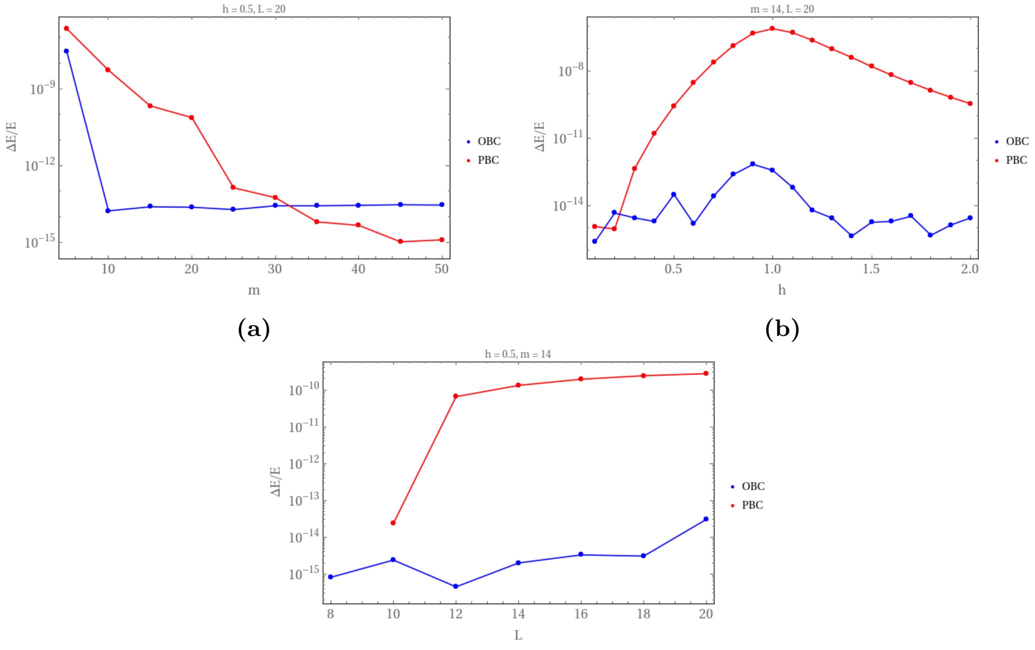


Figure 5.2: *Density matrix eigenspectra for a 6-site left enlarged block embedded in a 12 site superblock of the Transverse Field Ising Chain for both open boundary conditions (OBC) and periodic boundary conditions (PBC). The external field was set to $h = 1$. The fact that the spectra drop down to very low values after as little as two or three eigenstates is a good first indication that DMRG might perform very well for this model. For example, one would only need to keep 10 states to get a truncation error of around 10^{-10} with open boundary conditions. This plot also clearly shows that one would need to keep considerably more states to get the same truncation error with periodic boundary conditions.*

There are two things that need to be pointed out here. Firstly, the spectrum drops down to very low values very quickly, even at the critical field $h = 1$. Since this means that you only have to keep a few states to get a good truncation error, it is a good first indication that DMRG might perform well for this model. Moreover, the density spectrum with open boundary conditions seems to be consistently below the one with periodic boundary conditions. This confirms that one has to use considerably more states to achieve the same truncation error with periodic boundaries as with open boundaries.

Next we can look at the correlation between the relative error in the energy and certain algorithm parameters. The relative error was calculated as a function of the number of states kept m, the external field h and the size of the superblock L. The results are summarised in figure 5.3. One obvious thing that stands out for all the plots, is that for most values of the parameters the energies obtained with open boundaries are several orders of magnitude more accurate than those obtained with periodic boundaries. This is in line with what the density matrix eigenspectra predicted. But even with periodic boundary conditions, the results always seem to be very accurate with a worst relative error of around 10^{-6} at the critical field. We will discuss each plot separately.



(a)

(b)

(c)

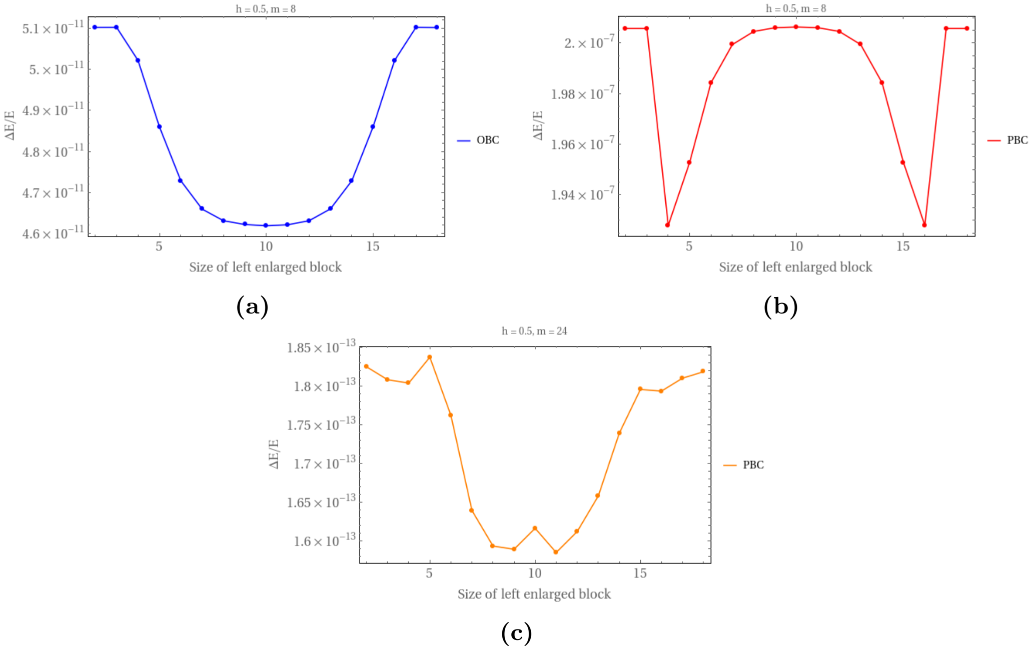
Figure 5.3: *Plots of the relative error in the energy versus (a) the number of states kept, (b) the external field and (c) the size of the superblock for both open boundaries (OBC) and periodic boundaries (PBC). The values $h = 0.5$, $m = 14$ and $L = 20$ were used for the parameters that were not being changed in a calculation. The relative error was defined as the absolute difference between the exact energy and the calculated energy, divided by the exact energy. Exact energies for periodic boundary conditions were obtained from* (5.7)*, while for open boundary conditions they were obtained by diagonalising the full Hamiltonian of the chain using an iterative diagonalization algorithm.*

Figure 5.3a shows the correlation between the relative error and the number of states kept in the truncation. For open boundaries, the error seems to plateau after 10 states, while for periodic boundaries, the error just keeps decreasing with the number of states, even dipping below the error of OBC at around 32 states. One possible explanation for this, is that the open 20-site chain is plagued by a significantly larger environmental error than the periodic chain. And since with open boundaries we start with a lower truncation error, we quickly reach a point where the environmental error becomes the limiting factor. Meanwhile, with periodic boundaries we start with a higher truncation error, but if the environmental error is smaller, the error can decrease further before levelling out. It might also be possible that there is some inherent additional error in the ”exact” energies used to calculate the relative errors with open boundary conditions, since these were calculated via an iterative diagonalization scheme and not via an analytical solution.

The dependence of the error on the strength of the external field is presented in figure 5.3b. For both boundary conditions the error seems to peak around the critical field $h = 1$. This is a reasonable result, since the quantum fluctuations will diverge at the critical point and one would need to keep many states to accurately represent the system. So, if the number of states is kept constant, the error will go up when approaching the critical point. The periodic chain seems to suffer most from this increase in error, with an increase of up to 9 orders of magnitude when going from zero field to the critical field. Maybe White's argument about a periodic chain having to respond to nearly independent fluctuations at both ends could also explain this.

One can also look at the scaling of the error with the size of the superblock, see figure 5.3c. The general trend here seems to be that the error increases almost monotonically with the superblock size. Again, this seems quite reasonable, since with the same number of states kept, a larger chain means more degrees of freedom have been integrated out. This could in turn lead to a higher truncation error. It is unclear why there is such a steep jump in the error of PBC when going from a superblock of size 10 to a superblock of size 12. Perhaps the superblock becomes large enough after 12 sites that the properties don't change much when increasing the size further.

The last error dependency that will be investigated is the dependence on the superblock configuration. More precisely, the dependence on the size of the left block given that the size of the superblock remains constant. The relative error was, therefore, calculated for all sizes of the left enlarged block for a superblock of 20 sites. The results can be found in figure 5.4. For open boundary conditions (figure 5.4a), the results clearly show a minimum at the symmetric superblock configuration, just as was observed by White for the Heisenberg chain. [8] With periodic boundaries (figure 5.4b), the situation is, surprisingly, the exact opposite if you use the same number of states ($m = 8$). Triple the number of states kept, however, and the situation is again similar to the one with open boundaries (figure 5.4c), where the energy is lowest for enlarged blocks of similar sizes. The same result doesn't arise for open boundary conditions by changing the number of states kept. This result was completely unexpected and there is no explanation for it yet. The relative difference between the peak and the lowest point in figure 5.4b is quite small however, much smaller than the same difference in figures 5.4a and 5.4c. So, perhaps the shape is a coincidence and the line should be considered mostly flat.



(a)

(b)

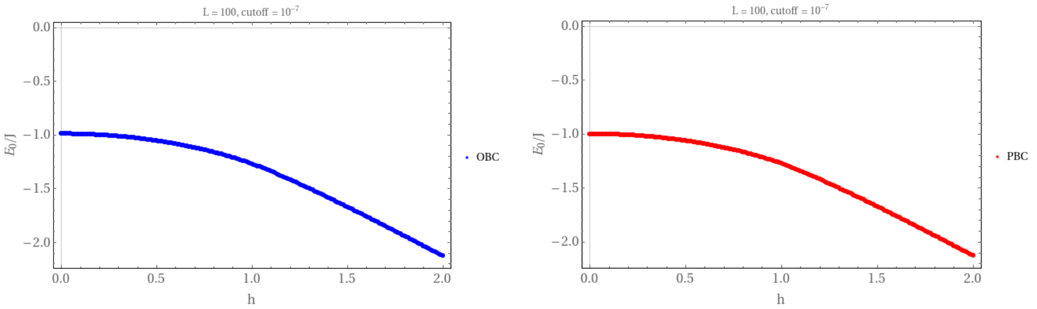
(c)

Figure 5.4: *Plots of the relative error in the energy versus the size of the left enlarged block for (a) open boundaries with $m = 8$, (b) periodic boundaries with $m = 8$, (c) periodic boundaries with $m = 24$. The superblock size is $L = 20$ and the external field is $h = 0.5$. For periodic boundaries the plot is given for two different values of m because they show such a striking difference. The relative energies were obtained in the same way as described in the caption of figure 5.3.*

5.2.2 Phase transition

We next investigated the behaviour of the model around the critical field by calculating the value of the quantities introduced in section 5.1.2 for the external field h ranging from 0 to 2 with steps of 0.01. The calculations were performed for a chain of size $L = 100$. We wanted to use as a large a chain as possible to be able to accurately compare them with the theoretical results in (5.10), (5.11), (5.12), (5.14) and (5.15), which are for an infinite chain. And $L = 100$ was the largest size we could reach for which all calculations could be completed with good accuracy within reasonable time. We chose not to keep the number of states constant, but to keep the truncation error below a certain cut-off value, the algorithm could then choose itself how many states were required at each step. This improved the efficiency of the algorithm, because now it wouldn't use a lot of states at steps where it wasn't really necessary. For every calculation, the cut-off was set to 10^{-7}. All calculations were performed for both open and periodic boundaries. The first result, the ground state energy as a function of the field, can be seen in figure 5.5.

The energy seems to vary smoothly over the entire range of field strengths. The limiting cases are $h = 0$, where the TFIM reduces to the regular Ising Model, and high field strengths $h \gtrsim 1.5$, where the external field dominates. In the first case, the ground state should have all spins aligned so that there is an energy $-J$ per spin. In the other case, the ground state will have roughly all spins in the direction of the field so that the energy is proportional to the field. Both cases are accurately predicted by the DMRG calculations.



(a) *Open Boundary Conditions*

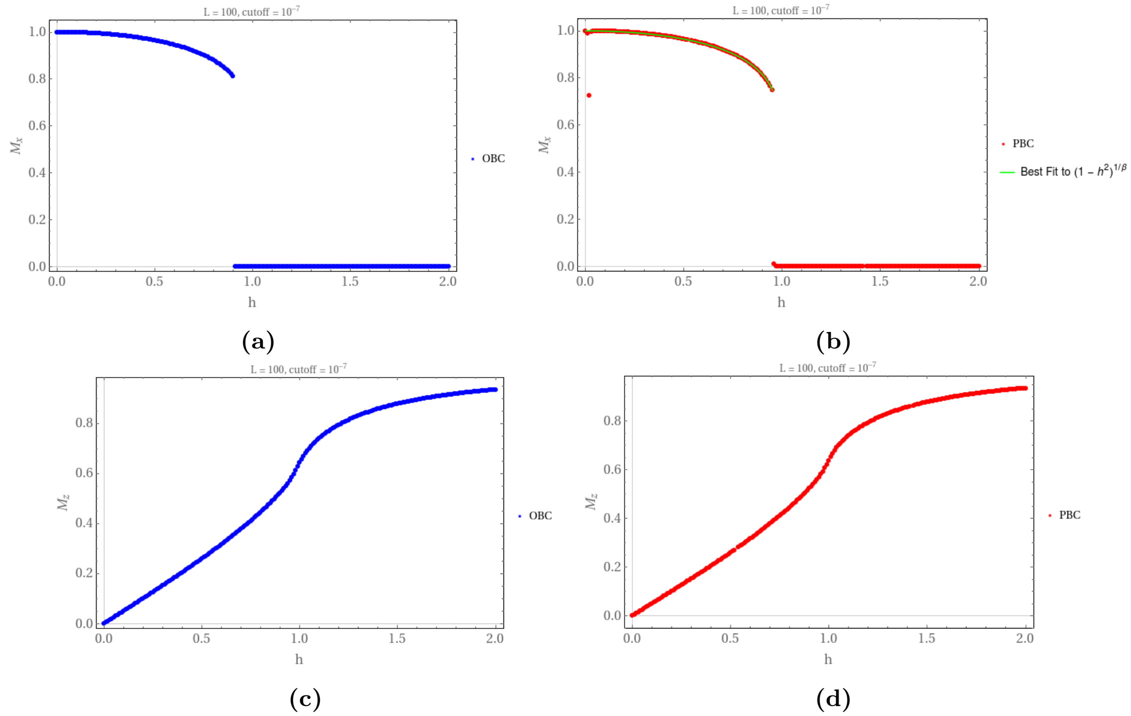
(b) *Periodic Boundary Conditions*

Figure 5.5: *Plot of the energy of the ground state in units of J versus the strength of the external field h for a chain of length $L = 100$. The truncation cut-off was set to 10^{-7}.*

We next calculated the expectation value for all relevant spin observables: M_x, M_z, ρ_1^x, ρ_1^z, ρ_n^x, ρ_n^z and ρ_{51}^x. The spin correlators were always calculated for spins spaced symmetrically around the centre, with an equal number of sites between both spins and the middle of the superblock. The value ρ_{51}^x was used as a measure of the long-range correlations in the system. A distance of 51 spins was chosen for this, because then both spins would still also be quite far from the left and right edges of the superblock, so that edge effects were hopefully minimal. The results for M_x and M_z are summarised in figure 5.6.

A few remarks about M_x are in order. The algorithm produced both positive and negative values for M_x, seemingly at random. This might be due to the fact that the model is invariant under a change of σ_x to $-\sigma_x$. [15] This might cause the algorithm to sometimes converge to a state where spins are preferably in the positive x-direction and other times to a state where spins are preferably in the negative x-direction. To produce the plot, the absolute value of M_x was taken, so that its sign was always the same. For an example of this sign problem, see figure 5.7. The plots of M_x (figures 5.6a and 5.6b) clearly show a phase transition from an ordered to a disordered state. Below a certain field strength, there is an average magnetisation of the chain in the x-direction, while above this field strength this magnetisation disappears. The transition doesn't seem to happen at the critical field $h = 1$, but actually at a slightly weaker field. This is especially true with open boundaries, where it already happens around $h = 0.9$. This might be due to the fact that errors in the energies are larger near the critical point, as already discussed, which could translate into even larger errors in the magnetisation. In the ferromagnetic region, the magnetisation was also fitted to $\left(1 - h^2\right)^{1/\beta}$. Only values before the point where it abruptly drops to zero were included. The best fit was given by $\beta = 8.0 \pm 0.3$, in relatively good agreement with (5.15). The standard deviation was quite high, though. This is caused by the one deviating value of M_x near $h = 0$. If this point is excluded, the best fit becomes $\beta = 8.009 \pm 0.006$, which is considerably better.

The magnetisation in the z-direction (figures 5.6c and 5.6d) varies smoothly from a value of 0 at zero field strength to a value near 1 at high field strengths. This agrees with our intuition that the z-component of the spin should be random when we don't apply a field and that all spins slowly start to point in the z-direction when the field becomes strong.



(a)

(b)

(c)

(d)

Figure 5.6: *Plots of the magnetic order parameters M_x ((a) and (b)) and M_z ((c) and (d)) versus the field strength h. The values were calculated for a chain of size $L = 100$ with a truncation error cut-off of 10^{-7}. Since the algorithm produced values of M_x for which the sign seemed to change at random, the absolute value of M_x was used for these plots.*

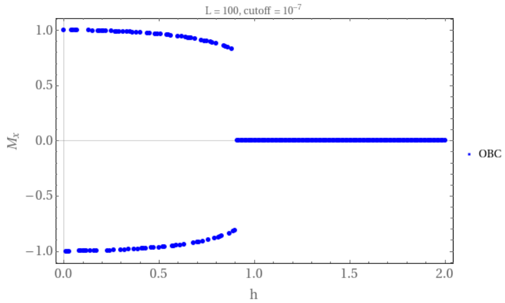


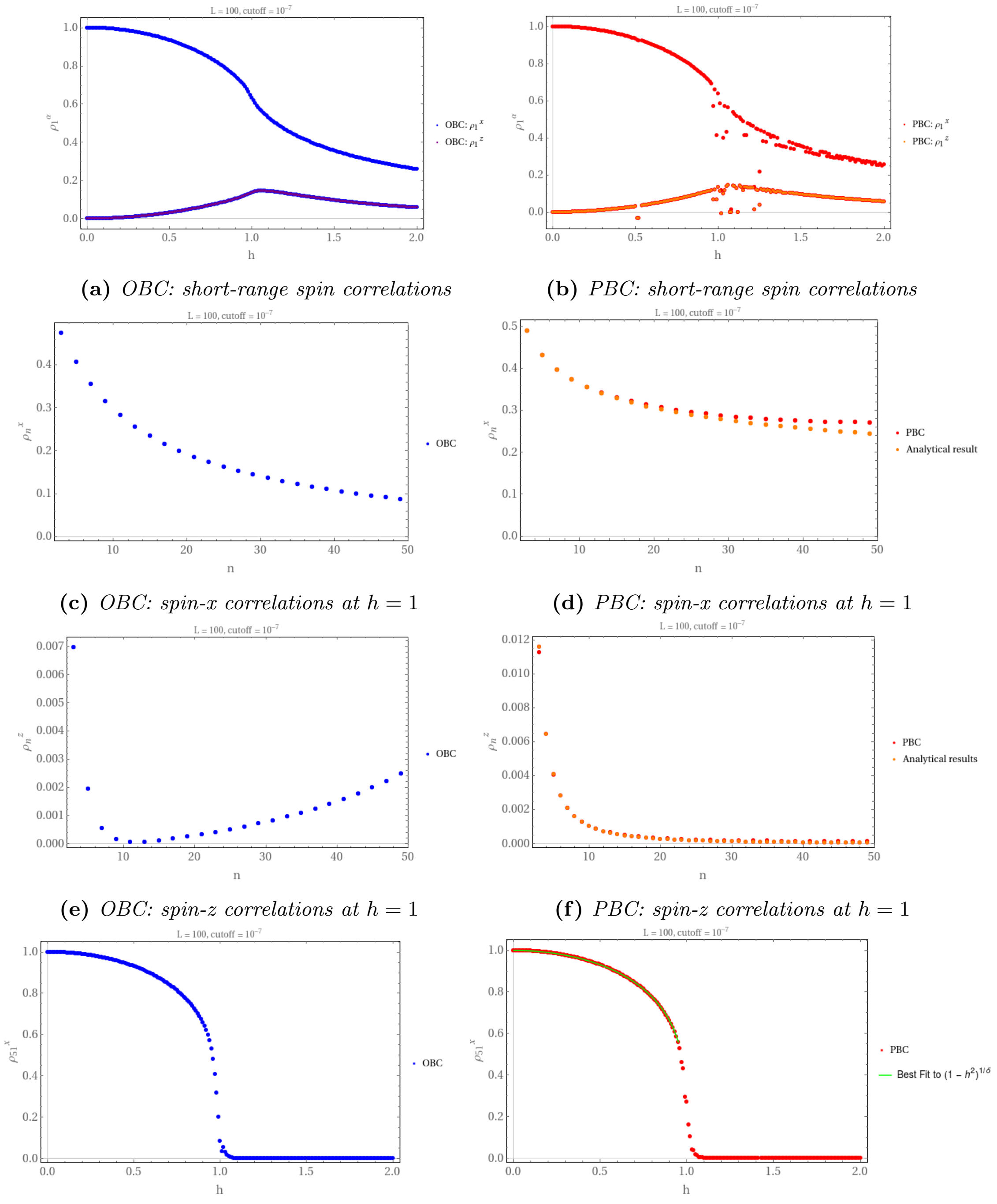
Figure 5.7: *The version of figure 5.6a before the absolute value was taken of the order parameter M_x. This plot clearly shows the sign problem that plagues the expectation values of the x-component of the spin.*

The spin-spin correlators were also calculated and the results can be found in figure 5.8. The spin-x correlators were plagued by the same randomly changing sign as the magnetic order parameter M_x, although to a lesser extent. For the spin-x correlators the problem is harder to explain by the invariance under the change of σ_x to $-\sigma_x$, since the spins should be predominantly aligned and not anti-parallel, regardless of the direction of the x-component of the spin. Perhaps the algorithm sometimes converges to some superposition of a state with spins predominantly in the positive x-direction and one with the spins predominantly in the negative x-direction, which then causes a negative value for the correlator, but the reason for this sign problem remains unclear. For the plots, it was chosen to take the absolute value of the spin-x correlators to account for this.

The first correlators to be discussed are ρ_1^x and ρ_1^z. These are a measure of the short-range order in the system. For the short-range correlation for the x-component varies smoothly from a value of 1 at zero field to lower values at strong fields. This is a reasonable result, since at zero field all spins should be aligned in the same direction, leading to a correlation of 1. And while there might be no average magnetisation in the x-direction at high fields, the spins do still prefer to be parallel due to the interaction J, so even at high field strengths there should be some short-range correlation. ρ^z is a measure of the difference between the spin-z correlations and the average magnetisation in the z-direction. A high ρ^z means a large correlation while the magnetisation is low, while a low ρ^z might both mean low correlation and low magnetisation or high correlation and high magnetisation. Its peak at the critical point seems to suggest that the correlations grow faster near the critical point than the magnetisation in the z-direction itself. The plot for periodic boundary conditions is also a striking example of the fact that errors in observables can be quite large even if the errors in the energies are very small.

At the critical point, the dependence of the correlators on the distance between spins was also determined. In three of the four cases, the correlations decreased as the distance increased. But, for the spin-z correlations with open boundaries, the correlations somehow curved back up after a certain point. It is unclear why this happens, but as can be seen by looking at the axes, the value of the correlator remains very small, so it is possible that this is due to errors. For periodic boundaries, the analytical results ((5.11) and (5.12)) were also plotted in the same figure. They are in good agreement for most small values of n, but they seem to diverge at larger n. This is probably an edge effect, since a 100-site chain is not really infinite and due to the periodic boundaries the spins can interact from two sides.

The last spin observable that was investigated was ρ_{51}^x as a measure for the long-range order of the system. As can be seen in the plots, there are strong long-range correlations below the critical field $h = 1$, which quickly disappear once the field strength passes the critical point. This is a strong indication of a phase transition from a ferromagnetic system to a paramagnetic system. The phase transition for the long-range correlation seems to happen nearer to the critical field than what the plots of the magnetic order parameters suggested. For periodic boundary conditions, the long-range correlators were also fitted to $(1-h^2)^{1/\delta}$ for the values below the critical field. A value of $\delta = 4.0032 \pm 0.0003$ was found for the fit parameter, once again in excellent agreement with the analytical result of (5.14).



(a) *OBC: short-range spin correlations*

(b) *PBC: short-range spin correlations*

(c) *OBC: spin-x correlations at $h = 1$*

(d) *PBC: spin-x correlations at $h = 1$*

(e) *OBC: spin-z correlations at $h = 1$*

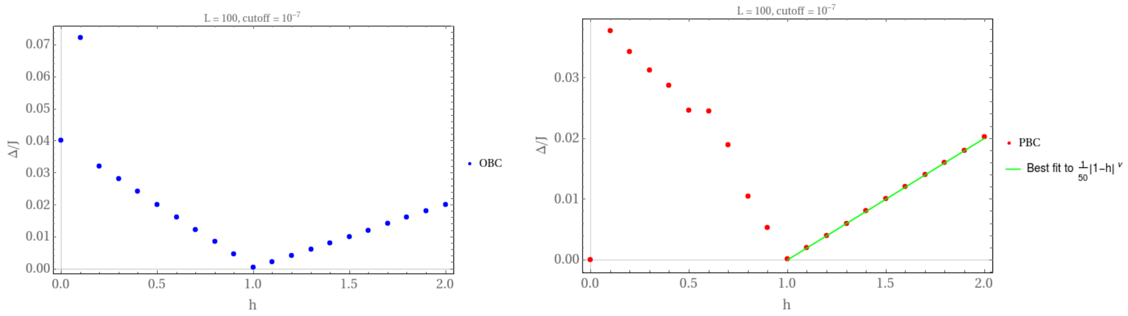
(f) *PBC: spin-z correlations at $h = 1$*

(g) *OBC: long-range spin-x correlations*

(h) *PBC: long-range spin-x correlations*

Figure 5.8: *Plots of several spin correlator functions. (a) and (b): ρ_1^x and ρ_1^z as a function of h. (c) and (d): ρ_n^x at $h = 1$ as a function of the distance n. (e) and (f): ρ_n^z at $h = 1$ as a function of the distance n. (g) and (h): $\rho_5 1^x$ as a function of h. All values were calculated for a chain of size $L = 100$ with a truncation cut-off of 10^{-7}. The spin-x correlators seemed to be plagued by the same sign problem as the magnetic order parameter M_x. To remedy this, we chose to take the absolute value of these spin correlators for the plots.*

To test whether the Density Matrix Renormalization Group was able to produce accurate excited energies, the mass gap was also calculated as a function of the field strength. For these calculations we also used a 100-site chain and a truncation cut-off of 10^{-7}. The results are presented in figure 5.9.



(a) *Open Boundary Conditions* **(b)** *Periodic Boundary Conditions*

Figure 5.9: *Plot of the mass gap $\Delta = E_1 - E_0$ in units of J versus the strength of the external field h for a chain of length $L = 100$. The truncation cut-off was set to 10^{-7}.*

The DMRG calculations accurately predict the mass gap to go to 0 at the critical point. And it also predicts linear behaviour in the paramagnetic region. The behaviour of the mass gap in the ferromagnetic region is harder to predict from the analytical results, but the calculations also seem to show roughly linear behaviour, but at different slopes than in the paramagnetic region. For periodic boundaries, the values of the mass gap above the critical field were also fitted to $1/50\,|1-h|^{\nu}$. The factor of 1/50 arises because we calculated the energies per site, so the equation for the mass gap (5.10) should also be divided by 100. The best fit was given by $\nu = 1.000 \pm 0.005$, which is in good agreement with (5.10).

6 Conclusion and Outlook

The Density Matrix Renormalization Group algorithm was introduced for one-dimensional quantum lattices. It was shown that the algorithm follows naturally from other Renormalization Group methods. Special attention was also paid to the product basis and density operator, which are essential for understanding DMRG. The algorithm was discussed from a mostly practical perspective.

As an example model, the Ising Chain with a Transverse Field was introduced. This model is interesting, because it is one of the simplest one-dimensional models that shows a phase transition at a finite external field. And since it can be solved analytically, results of the DMRG calculations could be easily checked. The behaviour of several key properties of the chain were discussed, as predicted by the analytical solution.

The DMRG algorithm was implemented from scratch in C++ for the Transverse Field Ising Model. This implementation was used to both investigate the behaviour of the errors of DMRG calculations and to explore the phase transition in the Transverse Field Ising Chain.

It was found that errors are larger when using periodic boundary conditions than when using open boundary conditions, sometimes up to several orders of magnitude. The errors in the energy seemed to decrease when increasing the number of states kept in the DMRG truncation, but this decrease levelled out for open boundary conditions, suggesting the existence of an environmental error, which limits the accuracy one can achieve. The errors were also found to depend on the strength of the applied external field. This dependence could yield significantly larger errors near the critical point of the phase transition, especially when using periodic boundary conditions. It was confirmed that a symmetric superblock configuration with enlarged blocks of equal sizes yields the most accurate results.

The phase transition in the Ising Model with a Transverse Field was explored through the calculation of the mass gap and of the expectation values of spin observables as a function of the external field. The existence of a phase transition from a ferromagnetic phase, with average magnetisation and long-range correlations for the x-component of the spin, to a paramagnetic phase, where the order in the x-component of the spin disappeared and the z-component of the spin was predominantly in the direction of the field. An unexpected finding was that the DMRG calculations seem to be plagued by a sign error, which causes the sign of a spin expectation value to randomly change sign for some values of the external field. In general, the results of the DMRG calculations were in good agreement with the analytical results, but the errors in the spin observables seemed to be significantly larger than the errors in energies when using the same parameters. Using the results of the calculations, the critical exponents for the Transverse Field Ising Chain were determined and these were in excellent agreement with the theory.

For further research, one could try to look for the origin of the aforementioned sign problem. The calculations could, for example, be performed with a lower truncation cut-off to increase the accuracy and see if this has any effect on the problem, but this will require a significant amount of computational effort. One could also try to implement one of the improvements mentioned in section 4.8 and see how these effect the results of the calculations. It could also be very interesting to extend the analysis to higher-dimensional Transverse Field Ising Models or to the time evolution of the model.

References

[1] Michel Le Bellac and Patricia de Forcrand-Millard. *Quantum Physics*. Cambridge University Press, Cambridge, 2006. ISBN 9780511616471. doi: 10.1017/CBO9780511616471. URL http://ebooks.cambridge.org/ref/id/CBO9780511616471.

[2] Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, 1991. ISBN 9780511840371. doi: 10.1017/CBO9780511840371. URL http://ebooks.cambridge.org/ref/id/CBO9780511840371.

[3] Jonas Maziero. Computing partial traces and reduced density matrices. *International Journal of Modern Physics C*, 28(01):1750005, 2017. ISSN 0129-1831. doi: 10.1142/S012918311750005X. URL http://arxiv.org/abs/1601.07458.

[4] U. Fano. Description of States in Quantum Mechanics by Density Matrix and Operator Techniques. *Reviews of Modern Physics*, 29(1):74–93, jan 1957. ISSN 0034-6861. doi: 10.1103/RevModPhys.29.74. URL https://link.aps.org/doi/10.1103/RevModPhys.29.74.

[5] Kenneth G. Wilson and J. Kogut. The renormalization group and the ϵ expansion. *Physics Reports*, 12 (2):75 – 199, 1974. ISSN 0370-1573. doi: https://doi.org/10.1016/0370-1573(74)90023-4. URL http://www.sciencedirect.com/science/article/pii/0370157374900234.

[6] Kenneth G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Rev. Mod. Phys.*, 47:773–840, Oct 1975. doi: 10.1103/RevModPhys.47.773. URL https://link.aps.org/doi/10.1103/RevModPhys.47.773.

[7] S. R. White and R. M. Noack. Real-space quantum renormalization groups. *Physical Review Letters*, 68(24):3487–3490, jun 1992. ISSN 00319007. doi: 10.1103/PhysRevLett.68.3487. URL https://link.aps.org/doi/10.1103/PhysRevLett.68.3487.

[8] Steven R. White. Density-matrix algorithms for quantum renormalization groups. *Physical Review B*, 48(14):10345–10356, oct 1993. ISSN 01631829. doi: 10.1103/PhysRevB.48.10345. URL https://link.aps.org/doi/10.1103/PhysRevB.48.10345.

[9] André Luiz Malvezzi. An introduction to numerical methods in low-dimensional quantum systems. *Brazilian Journal of Physics*, 33(1), 2003. ISSN 0103-9733. doi: 10.1590/S0103-97332003000100004. URL http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-97332003000100004&lng=en&nrm=iso&tlng=en.

[10] U Schollwöck. The density-matrix renormalization group. *Reviews of Modern Physics*, 77(1):259, 2005. ISSN 00346861. doi: 10.1103/RevModPhys.77.259. URL https://arxiv.org/pdf/cond-mat/0409292.pdf.

[11] Stephen G. Brush. History of the Lenz-Ising model. *Reviews of Modern Physics*, 39(4):883–893, oct 1967. ISSN 00346861. doi: 10.1103/RevModPhys.39.883. URL https://link.aps.org/doi/10.1103/RevModPhys.39.883.

[12] David Chandler. *Introduction to Modern Statistical Mechanics*. Oxford University Press, 1987. ISBN 978-0-19-504277-1. doi: citeulike-article-id:5967174. URL https://books.google.com/books?id=3taTh5D-CDsC.

[13] Pasquale Calabrese, Fabian H.L. Essler, and Maurizio Fagotti. Quantum quench in the transverse field Ising chain: I. Time evolution of order parameter correlators. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(7), 2012. ISSN 17425468. doi: 10.1088/1742-5468/2012/07/P07016. URL https://arxiv.org/pdf/1204.3911.pdf.

[14] R. J. Elliott, P. Pfeuty, and C. Wood. Ising model with a transverse field. *Physical Review Letters*, 25 (7):443–446, aug 1970. ISSN 00319007. doi: 10.1103/PhysRevLett.25.443. URL https://link.aps.org/doi/10.1103/PhysRevLett.25.443.

[15] Pierre Pfeuty. The one-dimensional Ising model with a transverse field. *Annals of Physics*, 57(1):79–90, mar 1970. ISSN 1096035X. doi: 10.1016/0003-4916(70)90270-8. URL http://linkinghub.elsevier.com/retrieve/pii/0003491670902708.

[16] Conrad Sanderson and Ryan Curtin. Armadillo: a template-based C++ library for linear algebra. *The Journal of Open Source Software*, 1(2):26, 2016. ISSN 1833-9646-4314. doi: 10.21105/joss.00026. URL http://joss.theoj.org/papers/10.21105/joss.00026.