Numerical analysis of a Josephson junction qubit using Matrix Product States

by

J.T. Teitsma

to obtain the degree of Bachelor of Science
at the Delft University of Technology

Student number:	4327454	
Supervisors:	J.M. Thijssen,	Faculty of Applied Sciences, TU Delft
	C. Vuik,	Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft
Co-readers thesis committee:	M.T. Wimmer,	Faculty of Applied Sciences, TU Delft
	R. van der Toorn,	Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft

TU Delft

Acknowledgments

Writing this thesis has learned me valuable lessons in many areas. First of all, getting inside knowledge in many topics ranging from quantum mechanics to numerical analysis and trying to combine both worlds into one project, has learned an indescribable lot about the two fields of study that I took my bachelors in. Maybe even more important are the lessons that I learned on a personal level.

I wish to thank my supervisors Jos Thijssen and Kees Vuik for the discussions we had, but also for keeping me on track during our regular meetings and your personal attention. I cannot imagine how I would have finished this project without your flexibility and patience.

Also a special thanks to Michael Wimmer and Ramses van der Toorn for taking the time to read and evaluate this thesis.

For my friends and family, who have supported me throughout the process in any possible way — although these two words are not enough to express my gratitude: thank you.

J.T. Teitsma
Delft, 27 August 2018

Contents

1 Introduction

Technological developments in the last couple of decades have been marked by the rise of computing machines. Since the introduction of so called supercomputers in the 1960s, the calculation power of these machines have doubled nearly every two years, as described by Moore's Law. This trend opened many possibilities in nearly every field of science. As computing power kept growing larger and the devices kept growing smaller, computers now have made their way into many people's homes and even pockets, affecting personal lives on a daily basis. However, computer chips now are reaching a such a small scale, that physical limits on performance arise. Leading chip manufacturer Intel has stated [2] that the pace at which they will introduce new chip-making technology will slow in the coming decade.

Simultaneously, an alternative type of computers is being developed, namely quantum computers. The field of quantum computing and quantum information has been developing since the 1980s, when among others David Deutsch started his research on this topic [3]. The quantum computer is able to perform many computations simultaneously by making use of quantum mechanical principles such as entanglement and superposition. The potential of quantum computers is enormous: many problems that are believed to be insolvable by classical computation could be solved using quantum computers and quantum algorithms. One of the most promising applications of quantum computing is in quantum chemistry. Recently, researchers of the University of Sydney developed a way model chemical bonds and reactions using quantum computers [4]. This could lead to new ways to solve problems in for example medicine. There are numerous other possible applications for quantum computing, as for example listed by Networked Quantum Information Technologies (NIQT) [5].

Classical computers are built up from bits representing a value of 0 or 1, on which they can perform one operation (or calculation) at the time. In contrast, quantum computers consist of quantum bits ('*qubits*'). The Hilbert space of a qubit is two-dimensional. The basis is formed by states $|0\rangle$ and $|1\rangle$. A qubit can be in superposition of the two ground states. This property allows for parallel computations, increasing computing power compared to classical computers. In order to preserve accuracy of calculations on qubits, another important requirement is that they are robust, meaning that they are not influenced by perturbations from outside, which is called decoherence. For example, qubits could decohere after single light particles (called '*photons*') disturb them out of their original superposition state, limiting the performance of the qubit device [6]. Developing robust, reliable qubits is thus essential to the development of the first quantum computer. Currently, many research teams around the world are participating in a research race to do so, among them the QuTech institute, in collaboration with the department of Quantum Nanoscience at the TU Delft.

In this thesis, we study one possible realization of a qubit. This realization was proposed by Hans Mooij, emeritus professor at the TU Delft. The qubit consists of a superconducting circuit, with several non superconducting or isolating junctions included. The structure composed of two pieces of superconducting material, separated by non superconducting materials, is called a Josephson junction, named after Brian David Josephson, who discovered and described the effects that occur at these junctions [7]. When exposed to an external magnetic flux, a circuit containing one or more Josephson junctions conserves discrete quanta of flux enabling the formation of two required ground states, which could be used as basis states for a qubit.

As will be described in later chapters, the Josephson junctions can be characterized by the phase difference between the two superconducting materials on both sides of the junction. Therefore, the wave function of the circuit — needed to find the two ground states of the qubit — depends on the phase differences of all junctions.

One way to analyze the qubit is to view each Josephson junction as a site in a one-dimensional quantum chain [8]. For the analysis of such a system, we are interested in the possible configuration *'states'* of our system. The space in which these states are included, is called the Hilbert space. Generally, the wave function of any one-dimensional quantum chain is a linear combination of the wave function of basis states of the Hilbert space. For a chain of only two sites, the Hilbert space can be explored quite straightforwardly. However, every time one adds another site to the chain, the number of possible states increases by the same factor - hence the dimension of the Hilbert space increases exponentially with the length of the chain. This also increases the size of the vector containing all linear combination coefficient, making the calculations of systems with multiple sites quite hard. Ironically enough, the availability of a quantum computer would make the exploration of exponentially growing number of states a lot easier.

Over the years, many numerical methods have been developed to analyze one-dimensional quantum chains. Most notably the density matrix renormalization group (DMRG) algorithm [9] has delivered great results. In recent research, the formalism of matrix product states (MPS) has been successfully combined with the DMRG algorithm, significantly improving its performance and providing solutions to other problems[10]. In MPS, the linear combination coefficient vector is rewritten as a product of matrices, which offers possibilities to further increasing the efficiency of numerical simulations. The time-evolving block decimation (TEBD) algorithm is a way to simulate time-evolution of quantum states. This procedure also allows us to evolve a random quantum state into the ground state of a system. Both MPS and TEBD will be at the core of this thesis.

The main goal of this thesis to understand the workings of the MPS formalism and apply the TEBD algorithm to systems of superconducting Josephson junctions. The build-up is as follows: in chapter 2 I lay the theoretical framework behind the Josephson junction qubits. We will start with a brief overview of the theory of quantum mechanics, needed to understand qubits at a small scale and to introduce the notation used in later calculations. We dive into a specific subset of quantum systems, namely one-dimensional chains, and learn about numerical methods that reveal properties of these systems. Next, the Josephson effect is explained in detail. In chapter 3, the mathematical foundation for our analysis is laid out, when we discuss the earlier-mentioned DMRG and MPS, that led to the development of the TEBD algorithm. Chapter 4 contains the results of all simulations that were done in this thesis. We start with the application of TEBD to one-dimensional models as the Ising model and Josephson junction chains, before we move on to the analysis of the proposed qubit.

2

Theoretical foundation

In order to study the system of interest in this report, we need to build an understanding of quantum systems. For that, the origin, history and most important principles of quantum mechanics are described in this chapter.

2.1. Quantum systems

Quantum mechanics is the study of the behavior of energy and matter at the the smallest scale. The theory behind quantum mechanics is built on several postulates, of which some are introduced in this section. After a short introduction about the origin of quantum mechanics, we elaborate on the underlying principles, we shall introduce concepts that are needed in later chapters.

2.1.1. Origin of quantum mechanics

At the end of the 19th century, some phenomena that could not be described by classical mechanics were discovered; most notably, the observation of light spectra coming from radiating blackbodies. Classical physics did not provide a way to predict the spectrum of the light coming off such a metal body. Max Planck found a way to predict these spectra, using the assumption that the energy in heat radiation is quantized,

$$E = nhf, \quad n \in \mathbb{N} \tag{2.1}$$

where f is the frequency of the emitted photons and h is now known as *Planck's constant.* The value of $h = 6.626 \times 10^{-34} J \cdot s$. Later Einstein discovered that this quantization assumption could also explain the photoelectric effect, which describes the liberation of photoelectrons when photons fall on a cathode. Following these discoveries, the assumption of quantized energy levels led to the development of a whole new field of physics, intuitively called *quantum* mechanics.

2.1.2. Wave function of quantum systems

From a classical perspective, it is possible to exactly describe a system by a set of its measurable physical quantities (also called *observables*), such as position, momentum or energy at initial time $t = t_0$. We call this configuration of quantities the *state* of the system. Time evolution is then determined by Hamilton's equations. Next to that, quantum mechanical systems can be in different states, each representing a possible system configuration. However, each state is now represented by a complex function called the *wave function.* The wave function can be represented by a *ket* $|\psi\rangle$ in a complex vector space of states. Its complex conjugate is represented by the *bra* $\langle\psi|$. All states are represented in *bra-ket notation* in this report. The vector space of states is called the *Hilbert space* $\mathscr{H}$.

To illustrate this, we look at the example of a single electron, which will be used throughout the rest of this chapter. Electrons have a fundamental property *s*, *spin*, which is comparable to an angular momentum intrinsic to the particle and not determined by its motion. The spin of the electron is quantized and can have two directions, *up* or *down.* In this case, *up* and *down* are the possible states that the system of interest (the electron) can be in. We denote these states by $|\psi_{\text{up}}\rangle$ and $|\psi_{\text{down}}\rangle$.

The two states $|\psi_{\text{up}}\rangle$ and $|\psi_{\text{down}}\rangle$ span up the Hilbert space for an electron spin, so the Hilbert space has dimension 2. The most intuitive - although generally not the most convenient - way to denote the states is to use arrows, so $|\psi_{\text{up}}\rangle = |\uparrow\rangle$ and $|\psi_{\text{down}}\rangle = |\downarrow\rangle$. Another possibility is to number the states, denoted as $|\psi_{\text{up}}\rangle = |0\rangle$ and $|\psi_{\text{down}}\rangle = |1\rangle$.

The vector space nature of the Hilbert space reveals another property of quantum mechanical system states. Mathematically, a complex vector space V is closed under linear combinations, meaning that for two states $|\psi_1\rangle$ and $|\psi_2\rangle$ in V, any linear combination $c_1|\psi_1\rangle + c_2|\psi_2\rangle$, with $c_1, c_2 \in \mathbb{C}$, is also in V. In quantum mechanics, this means that the so called *superposition* of $|\psi_1\rangle$ and $|\psi_2\rangle$, $|\psi_{\text{superposition}}\rangle = c_1|\psi_1\rangle + c_2|\psi_2\rangle$, is also a possible state of the system. In the mathematics of complex vector spaces, there are no restrictions on c_1 and c_2 as long as they are complex numbers. In quantum mechanics however, a normalization condition is imposed on coefficients $|c_i|^2$, as they are a measure of finding the corresponding state. The total probability of finding any state is 1, leading to the requirement that $|c_1|^2 + |c_2|^2 = 1$, or in general $\sum_{i=1}^{N} |c_i|^2 = 1$.

For more complex systems, the dimension of $\mathcal{H}$ increases. A Hilbert space of dimension N is spanned by N linear independent basis vectors $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_{\text{N}}\rangle \in \mathcal{H}$, also denoted as $|1\rangle, |2\rangle, \ldots, |N\rangle \in \mathcal{H}$. The *vector notation* represents each basis state as a unit vector of dimension N. Following the principle of superposition, each possible state in $\mathcal{H}$ can be written as a linear combination of the basis states:

$$\begin{aligned} |\psi\rangle &= c_1|\psi_1\rangle + \ldots + c_N|\psi_{\text{N}}\rangle = \begin{bmatrix} c_1 & c_2 & \ldots & c_N \end{bmatrix} (|\psi_1\rangle + |\psi_2\rangle + \ldots + |\psi_N\rangle) \\ &= \sum_{i=1}^{N} c_i|\psi_i\rangle \quad \text{with} \sum_{i=1}^{N} |c_i|^2 = 1. \end{aligned} \tag{2.2}$$

In our example of an electron spin, the states $|\psi_{\text{up}}\rangle$ and $|\psi_{\text{down}}\rangle$ are the basis states and any state $|\psi\rangle$ of the electron spin can be a superposition of these two states, with the condition that $|c_{\text{up}}|^2 + |c_{\text{down}}|^2 = 1$. This leads to the following notation:

$$\begin{aligned} |\psi_{\text{up}}\rangle &= |\uparrow\rangle = |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ |\psi_{\text{down}}\rangle &= |\downarrow\rangle = |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ so} \\ |\psi\rangle &= c_{\text{up}}|\psi_{\text{up}}\rangle + c_{\text{down}}|\psi_{\text{down}}\rangle = \begin{bmatrix} c_{\text{up}} \\ c_{\text{down}} \end{bmatrix} \end{aligned} \tag{2.3}$$

2.1.3. Density matrix

The wave function representation $|\psi\rangle$ of a quantum state - as defined in equation (2.2) - can be used when we have complete knowledge of the quantum state. However, when there is uncertainty in which state a system is, the wave function is unknown. This occurs for example when the system of interest is not isolated anymore, but interacts with an environment. In this case, it might not be possible anymore to find a single wave function of the system itself, for example because there the system can reside in multiple states. We call this a *mixed state*. In that case, another representation of the wave function turns out to be useful: the *density matrix* ρ. The density matrix is defined as:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{2.4}$$

Remember that $\langle\psi_i|$ is the complex conjugate of $|\psi_i\rangle$, so ρ is defined sum of the outer products of all possible states $|\psi_i\rangle$ that the system can be in. In this equation, the coefficients p_i are probabilities corresponding to the possible states $|\psi_i\rangle$. When the system is in a pure, known state $|\psi\rangle$, the density matrix reduces to $\rho = |\psi\rangle\langle\psi|$.

When measuring a quantum system that is initially in a pure state, the wave function collapses to a single state $|\phi\rangle$. Before a measurement has taken place, it is uncertain to which state the system will collapse. The probability of finding the system in state $|\phi\rangle$ is given by the square of the inner product of $|\phi\rangle$ and $|\psi\rangle$: $P_\phi =$

$|\langle\phi|\psi\rangle|^2$. If the system is in a mixed state, this probability is given by $P_\phi = \sum_i p_i|\langle\phi|\psi_i\rangle|^2$. This expression can be rewritten in order to reveal an important property of the density matrix:

$$P_\phi = \sum_i p_i|\langle\phi|\psi_i\rangle|^2 = \sum_i p_i\langle\phi|\psi_i\rangle\langle\psi_i|\phi\rangle = \langle\phi|(\sum_i p_i|\psi_i\rangle\langle\psi_i|)|\phi\rangle = \langle\phi|\rho|\phi\rangle \tag{2.5}$$

Using the definition of the density matrix, this equation shows that the density matrix of a mixed state contains enough information to calculate the possibility of finding that system in state $|\phi\rangle$ in measurement. Remember that all probabilities should add up to 1, so we can say that $\sum_i p_i = 1$.

2.1.4. Operators and observables

In the previous section, it is argued that quantum states are represented by a wave function or density matrix. We now turn to the quantum description of physical observables, such as position x or momentum p. In general, every observable q corresponds to a hermitian linear operator $\hat{Q}$. $\hat{Q}$ works on $|\psi\rangle$ via left multiplication [11]:

$$\hat{Q} : |\psi\rangle \rightarrow |\psi'\rangle = \hat{Q}|\psi\rangle \tag{2.6}$$

The fact that $\hat{Q}$ is hermitian means that $\hat{Q}$ is equal to its hermitian conjungate $\hat{Q}^\dagger$. The *spectral theorem* states the following about hermitian operators:

Theorem 1 *Suppose $\mathscr{H}$ is a finite-dimensional complex vector space of dimension N. If linear operator $\hat{Q} \in \mathscr{L}(\mathscr{H})$ is hermitian, there exists an orthogonal basis $\{|\psi_i\rangle\}_{i=1}^N$ for $\mathscr{H}$ consisting of the eigenvectors of $\hat{Q}$.*

This basis can be used as the set of linearly independent basis states $|\psi_i\rangle$ as used in equation (2.2) [12]. The N eigenvalues λ_i of $\hat{Q}$ are the values that can be found in a measurement of observable q. To illustrate this concept, we look at the example of electron spin s. We introduce the *spin operator* $\hat{\sigma}^z$, that represents s, the spin component along the z-direction:

$$\hat{\sigma}^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.7}$$

Applying $\hat{\sigma}^z$ to the states *up* $|0\rangle$ and *down* $|1\rangle$ - as defined in equation 2.3 - we find $\hat{\sigma}^z|0\rangle = +|0\rangle$ and $\hat{\sigma}^z|1\rangle = -|0\rangle$. We conclude that $|0\rangle$ and $|1\rangle$ are eigenstates of $\hat{\sigma}^z$, respectively corresponding to eigenvalues $+1$ and -1. Next to $\hat{\sigma}^z$, operators $\hat{\sigma}^x$ and $\hat{\sigma}^y$ are defined, corresponding to the spin components in the x- and y-direction:

$$\hat{\sigma}^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.8}$$

$$\hat{\sigma}^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{2.9}$$

The first eigenvalue of $\hat{\sigma}_x$ is $+1$, corresponding with eigenstate $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (or $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$). The other eigenvalue is -1, corresponding to eigenstate $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ (or $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$). These eigenstates are of interest in the next sections.

2.1.5. Properties of the density matrix

When working with observables, their average value $\langle\hat{Q}\rangle$ can be of interest. There are multiple ways to obtain this average value. If the system can be represented by its wave function $|\psi\rangle$, $\langle\hat{Q}\rangle$ is found by taking its expected value $\langle\psi|\hat{Q}|\psi\rangle$. However, when we have a mixed state in density matrix representation, $\langle\hat{Q}\rangle$ can also be expressed in terms of ρ. For this derivation we use a set of orthonormal eigenvectors $|\phi_k\rangle$ of the Hilbert space of this system. As all $|\phi_k\rangle$ are orthonormal, the identity $\sum_k |\phi_k\rangle\langle\phi_k| = \mathbb{1}$ holds:

$$\begin{aligned} \langle\hat{Q}\rangle = \langle\psi|\hat{Q}|\psi\rangle = \langle\psi|\hat{Q}| \left[\sum_k |\phi_k\rangle\langle\phi_k|\right] |\psi\rangle = \sum_k \langle\psi|\hat{Q}|\phi_k\rangle\langle\phi_k|\psi\rangle \\ = \sum_k \langle\phi_k|\psi\rangle\langle\psi|\hat{Q}|\phi_k\rangle = \sum_k \langle\phi_k|\hat{\rho}\hat{Q}|\phi_k\rangle = \mathrm{Tr}(\hat{\rho}\hat{Q}) \end{aligned} \tag{2.10}$$

In equation 2.10, the *trace operator* is introduced. In general, the trace of operator $\hat{Q}$ is defined as:

$$\mathrm{Tr}(\hat{Q}) = \sum_k \langle\phi_k|\hat{Q}|\phi_k\rangle \tag{2.11}$$

The trace operator in practice sums over all diagonal entries of a square matrix. The trace will be of later use in this chapter, where also the *partial trace* is introduced.

We know from section 2.1.3 that all probabilities in the definition of the density matrix (equation 2.4) add up to 1. This reveals another important property of the trace of the density matrix:

$$\begin{aligned}\mathrm{Tr}(\hat{\rho}) &= \sum_k \langle\phi_k|\hat{\rho}|\phi_k\rangle \\ &= \sum_k \langle\phi_k| \left[\sum_i p_i |\psi_i\rangle\langle\psi_i|\right] |\phi_k\rangle \\ &= \sum_i p_i \sum_k |\langle\phi_k|\psi_i\rangle|^2 = 1\end{aligned} \tag{2.12}$$

where we used the normality of bases $\{|\psi_i\rangle\}$ and $\{|\phi_k\rangle\}$ in the last step.

2.1.6. Hamiltonian of quantum systems

One operator in particular is of significant interest in quantum mechanics: the *Hamiltonian*, which is closely related to time evolution of a quantum system, given by the time-dependent Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = \hat{H}|\psi\rangle \tag{2.13}$$

A common example is the Hamiltonian of a particle in potential U; $\hat{H} = -\frac{\hbar^2}{2m}\nabla^2 + U$, (2.3). The Schrödinger equation is then written as.

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = -\frac{\hbar^2}{2m}\nabla^2|\psi\rangle + U|\psi\rangle \tag{2.14}$$

The Hamiltonian corresponds to the total energy of the system, which means that when $\hat{H}$ is applied to with one of its eigenstates $|\psi_i\rangle$, the returned eigenvalue λ_i has the value of E_i, the energy of the corresponding eigenstate:

$$\hat{H}|\psi_i\rangle = \lambda_i|\psi_i\rangle = E_i|\psi_i\rangle$$

Now we know how to find the energy of the system from its Hamiltonian, it is possible to look at one eigenstate of significant interest, namely the eigenstate $|\psi_i\rangle$ with the lowest energy E_i. This state is called the *ground state* of the system, with the corresponding energy being called *ground state energy*. If there are more than one states with the same lowest energy, the ground state is said to be *degenerate*.

2.2. One-dimensional quantum chains

Section 2.1 introduced some important elements of quantum systems, such as the *wave function*, *superposition*, *observables* and *operators*; specifically the *Hamiltonian*. In this section, we focus on a specific subset of quantum systems, namely one-dimensional (1D) quantum chains. Methods used in this report specifically make use of the characteristics of these systems.

Quantum systems consisting of only one body - such as the single spin used as an example - can be extended by adding similar bodies to the system. When the locations of the additional bodies can be labeled by a single index, corresponding to the position on a one-dimensional line, we speak of a quantum chain.

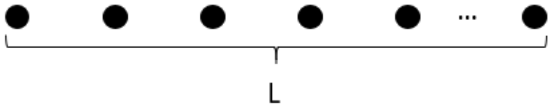

Figure 2.1: A simple graphical representation of a quantum chain of L bodies

As Thijssen [13] points out, there are several reasons why one-dimensional quantum chains are interesting to study. There exist experimental realizations of quasi-one dimensional systems in some crystals, but more importantly, the analytical and computational tools to study one-dimensional systems successfully may be useful for higher dimensions as well. This is exactly what we try to achieve in this thesis using the matrix product state method.

All units in a chain have their own Hilbert space $\mathscr{H}_i$, of which the dimension is denoted by d. In case of the electron spin, $d = 2$, as the electron spin has two possible configurations: *up* and *down*. The Hilbert space of a chain $\mathscr{H}$ is the direct product of all single Hilbert spaces $\mathscr{H}_i$. This means that a chain of L units, has d^L possible configurations. We conclude the dimension of the Hilbert space of the system grows exponentially with the number of units.

For many-body systems, a Hamiltonian can be found as well. Generally, this Hamiltonian contains interaction terms between the different bodies in the chain. This means that the wave function of the chain can not be written as a direct product of single-site wave functions anymore. Instead, we need to look at the wave function of the system as a whole. This phenomenon, called *entanglement*, is a central topic in this thesis.

The notation for wave function of a multi-body system is similar to the notation introduced in section 2.1 for single-body systems. In the case when we have two bodies A and B, the wave function of the system is obtained by taking the tensor product of the single-body wave functions $|\psi\rangle_A$ and $|\psi\rangle_B$, i.e.

$$|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B. \tag{2.15}$$

For simplicity, expression 2.15 may be abbreviated to $|\psi_{AB}\rangle = |\psi_A\rangle|\psi_B\rangle$ or $|\psi_A\psi_B\rangle$ throughout this thesis, by omitting the ⊗-sign. So in the case of a system consisting of electron spins that all point up ($|0\rangle$ or $|\uparrow\rangle$), the wave function is written as $|\psi\rangle = |00...0\rangle$ or $|\downarrow\downarrow ... \downarrow\rangle$ etcetera.

2.2.1. Ising model with transverse field

To get a better grasp of the concept of quantum chains, we consider the *1D Ising model* in more detail. The Ising model consists of a chain of N electron spins, which are exposed to a uniform external magnetic field. All electron spins have two possible configurations, *up* and *down*. Hence the number of possible system configurations - and the dimension of the Hilbert space - is $N = d^L = 2^L$. The energy of the system is determined by the Hamiltonian. Note that if we write Hamiltonian in its matrix representation, this matrix would have dimensions $2^L \times 2^L$, and also grows exponentially with system size. The Hamiltonian of the Ising chain reads:

$$\hat{H} = -J\sum_i \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z - g\sum_i \hat{\sigma}_i^x \tag{2.16}$$

$\hat{H}$ consists of two terms. The first term represents the coupling between two neighbouring spins in the chain via coupling constant J. The operators $\hat{\sigma}_i^z$ $\hat{\sigma}_i^x$ are the spin operators introduced in section 2.1.4, acting on site i. The summation of the first term is performed over all neighbouring pairs. In case of a positive value for J, all spins prefer to be surrounded by spins with the same orientation, as this lowers the total energy. This is called *ferromagnetic*. On the other hand, for negative J, the spins spend tend to orient in opposite direction, the *antiferromagnetic* model. The second term accounts for the influence of the external field g. The sign of g determines the direction of the field; in case g is positive, the eigenstate of $\hat{\sigma}_i^x$ belonging to eigenvalue $+1$, which is $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$, is favored in the configuration of the chain. Similarly, $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$ is less preferred for positive g. For negative g, this reasoning flips.

There are two possible ways to define boundary conditions on this model: *open* and *periodic*. In case of open boundary conditions, the model represents a finite chain, where the bodies on both ends of the chain only interact with one neighbour. For periodic boundary conditions, we need to make sure the first and last bodies interact by setting $\hat{\sigma}_{L+1}^z \equiv \hat{\sigma}_1^z$. For periodic boundary conditions, this model has been solved analytically by Pfeuty [1]. The analytical solution is not explored in detail in this thesis, but the values of the ground state energy are used to benchmark our own simulations of this model in chapter 4.

2.2.2. Heisenberg model

Quite similar to the Ising model from the previous section, is the *Heisenberg model*. The Heisenberg model is another one-dimensional quantum system that consists of electron spins that can point up and down. This model is governed by the slightly different Hamiltonian [8]:

$$\hat{H} = J\sum_{i=1}^{L-1} \mathbf{S}_i \cdot \mathbf{S}_{i+1} = \frac{\hbar^2 J}{4}\sum_{i=1}^{L-1} \sigma_i^x\sigma_{i+1}^x + \sigma_i^y\sigma_{i+1}^y + \sigma_i^z\sigma_{i+1}^z \tag{2.17}$$

In this Hamiltonian, $\mathbf{S}_i = (S^x, S^y, S^z) \equiv \frac{\hbar}{2}(\sigma_i^x, \sigma_i^y, \sigma_i^z)$ is the spin vector at site i. The Heisenberg model also has 2^L possible configurations, similar to the Ising model in the previous section.

2.3. Numerical methods to find the ground state energy

Suppose we are interested in the ground state energy of our Ising model, or large quantum systems in general. As we know that the Hamiltonian has all possible values for the energy as its eigenvalues, the straightforward procedure would be to completely diagonalize the Hamiltonian matrix. For direct diagonalization, many algorithms are available. The approach described in [14] is to transform the matrix in a tridiagonal form, and then diagonalize this tridiagonal matrix using the QL or QR algorithm. The computational cost of this approach scales as $\frac{2}{3}L^3$ [15]. During this approach, the entire matrix has to be stored and diagonalized, imposing limits on the maximum chain size. This makes direct diagonalization not suitable for computation of large chains.

For larger systems, there are many numerical methods for matrix computations available in standard textbooks [16]. One of these methods is the *Lanczos* method [17]: an iterative diagonalization method, which is often used in physics. The Lanczos algorithm not only can be used to diagonalize the Hamiltonian directly, but it is also used in the DMRG algorithm, discussed later in this chapter. In this section we discuss the ideas behind Lanczos diagonalization briefly.

Before we describe the Lanczos algorithm, we first look into the *power method*, which is at the basis of many iterative diagonalization methods. In the power method, $\hat{H}$ is repeatedly applied to a random initial state $|\psi^0\rangle$. Remember from section 2.1.2 that this random state can be expanded in the basis of eigenstates $|\psi_i\rangle$:

$$\begin{aligned}
|\psi^0\rangle &= \sum_i c_i|\psi_i\rangle \\
|\psi^n\rangle &= \hat{H}^n|\psi^0\rangle \\
|\psi^n\rangle &= \hat{H}^n\sum_i c_i|\psi_i\rangle \\
|\psi^n\rangle &= \sum_i c_i\hat{H}^n|\psi_i\rangle \\
|\psi^n\rangle &= \sum_i c_i\lambda_i^n|\psi_i\rangle
\end{aligned} \tag{2.18}$$

It then follows immediately that the state $|\psi_{\text{max}}\rangle$ with the largest eigenvalue λ_{max} dominates $|\psi^n\rangle$ after sufficient iterations, provided that c_{max} is unequal to zero - i.e. the initial state $|\psi^0\rangle$ has overlap with $|\psi_{\text{max}}\rangle$. The convergence of the power method depends on the difference between λ_{max} and the next-largest eigenvalue. The power method generally converges more slowly than other diagonalization methods - such as Lanczos diagonalization. However, it is easy to implement and very memory efficient as only $|\psi^n\rangle$ and $|\psi^{n-1}\rangle$ need to be stored in memory.

2.3.1. Lanczos Diagonalization

The power method serves another goal, as all steps generated in this procedure form a subspace - the *Krylow space* - that is a starting point for other procedures. The Krylow space is defined as follows:

$$\{|\psi^0\rangle, |\psi^1\rangle, |\psi^2\rangle, ..., |\psi^n\rangle\} = \{|\psi^0\rangle, \hat{H}|\psi^0\rangle, \hat{H}^2|\psi^0\rangle, ..., \hat{H}^n|\psi^0\rangle\} \tag{2.19}$$

The Lanczos method produces a matrix representation of $\hat{H}$ that is tridiagonal, by projecting $\hat{H}$ on the Krylow space. The algorithm works as follows [18]:

1. Choose a random normalized initial vector $|\psi^0\rangle$ that has non-zero overlap with the Hamiltonian's eigenstates.

2. Generate a sequence of Lanczos states with the following recursion relation:

$$|\psi^{n+1}\rangle = H|\psi^n\rangle - a_n|\psi^n\rangle - b_n^2|\psi^{n-1}\rangle$$

$$\text{with } a_n = \frac{\langle\psi^n|\hat{H}|\psi^n\rangle}{\langle\psi^n|\psi^n\rangle}$$

$$\text{and } b_n^2 = \frac{\langle\psi^n|\psi^n\rangle}{\langle\psi^{n-1}|\psi^{n-1}\rangle}$$

3. Iteratively repeat step 2 until the conditions $\langle\psi^n|\psi^n\rangle < \epsilon$ or $n = M$ (maximum number of iterations) are met.

4. Compute the following tridiagonal matrix T:

$$T = \begin{pmatrix} a_0 & b_1 & & & \\ b_1 & a_1 & b_2 & & \\ & b_2 & a_2 & \ddots & \\ & & \ddots & \ddots & b_n \\ & & & b_n & a_n \end{pmatrix}$$

Let V be the orthonormal matrix with columns $|\psi^0\rangle, ..., |\psi^n\rangle$, then it follows that $\hat{H} = V^\dagger T V^1$. Observe that any eigenvalue λ_i of T is also an eigenvalue of $\hat{H}$ as V and $V^\dagger$ are orthonormal.

5. Diagonalize T using the QL algorithm to find eigenvalues $\lambda_0, ..., \lambda_n$ of $\hat{H}$.

The procedure above briefly describes the Lanczos algorithm to find eigenvalues of the Hamiltonian of a large quantum system. As matrix T is tridiagonal, finding eigenvalues using the diagonalization in step *5* is more efficient to carry out. The Lanczos algorithm is a widely used method that can be applied to many systems. As with the power method, the convergence of the largest eigenvalues happens first, making the scheme especially suitable to find these states. For excited states of higher order, the number of iterations required to converge rises. In addition, as the dimension of the Hilbert space grows exponentially with the length L of one-dimensional systems, the growth of vectors $|\psi^i\rangle$ needs to grow evenly, reducing the speed of the method and making it less useful for large systems. In the next section, we look into a method that overcomes this exponential dimension growth. Despite these disadvantages, Lanczos is still used as the preferred method to benchmark other methods.

2.3.2. Density matrix formulation of quantum states

Although diagonalization using the Lanczos algorithm can provide numerically exact solutions to ground-state problems of one-dimensional quantum systems, the scheme becomes less powerful when the system size increases, as pointed out in the previous section. Therefore, there has been interest in other ways to represent a system state, in order to analyze systems of larger size. One such method is the density matrix renormalization group (DMRG), that was originally developed by White [9]. As the name suggests, this method makes use of the density matrix formulation, which is extended in this section.

One important step of the DMRG method is to split the system into two parts, as illustrated in figure 2.2. Mathematically, this is equivalent to bipartitioning the Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$. When we take $\{|i\rangle_A\}$ and $\{|j\rangle_B\}$ as orthonormal bases for $\mathcal{H}_A$ and $\mathcal{H}_B$, any wave function of the complete system $|\psi\rangle_{AB}$ can be written as a composition of the possible wave functions of subsystems A and B:

$$|\psi\rangle_{AB} = \sum_{i,j} c_{ij}|i\rangle_A \otimes |j\rangle_B \tag{2.20}$$

Subspaces $\mathcal{H}_A$ and $\mathcal{H}_B$ might have different dimensions d^{N_A} and d^{N_B} if the size of subsystems A and B differ. If we now define

$$|\tilde{i}\rangle_B = \sum_j c_{ij}|j\rangle_B, \tag{2.21}$$

[1] SOURCE

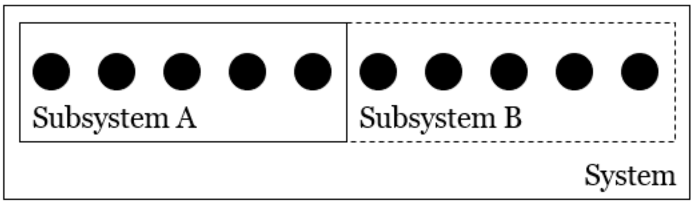

Figure 2.2: A quantum chain of 10 sites is split up in subsystems A and B, both consisting of 10 sites

the expression for $|\psi\rangle_{AB}$ in this equation reduces to a single sum:

$$|\psi\rangle_{AB} = \sum_i |i\rangle_A|\tilde{i}\rangle_B \tag{2.22}$$

Preskill shows in his lecture notes on quantum computation [19] that $\{|\tilde{i}\rangle\}$ is an orthogonal basis for $\mathscr{H}_B$. To arrive at an orthonormal basis for $\mathscr{H}_B$, we impose a normality condition on $\{|\tilde{i}\rangle\}$:

$$|i'\rangle = \frac{|\tilde{i}\rangle}{||\,|\tilde{i}\,\rangle||} \equiv \lambda_i|\tilde{i}\rangle \tag{2.23}$$

$$|\psi\rangle_{AB} = \sum_i \lambda_i|i\rangle_A|i'\rangle_B \tag{2.24}$$

The expression in 2.24 is called the *Schmidt decomposition* [20]. In the sum, index i and has $\min(d^{N_A}, d^{N_B})$ values. The coefficients λ_i are called the *Schmidt coefficients.*

The Schmidt decomposition allows us to write $|\psi\rangle_{AB} \in \mathscr{H}$ in terms of eigenvectors of subsystems A and B, with $\{|i\rangle_A\} \subset \mathscr{H}_A$ and $\{|i'\rangle_B\} \subset \mathscr{H}_B$. The Schmidt decomposition offers a more convenient representation of $|\psi\rangle_{AB}$ which is also used in the DMRG algorithm and when we explore matrix product states in the next chapter.

Just as density matrix ρ_{AB} is a way to represent the complete system, *reduced density matrices* ρ_A and ρ_B represent the respective subsystems A and B. We see in this section how to derive ρ_A and ρ_B from ρ_{AB}. We start off by plugging the expression for $|\psi\rangle_{AB}$ from equation 2.20 into the definition for the density matrix:

$$\rho_{AB} = |\psi\rangle_{AB\,AB}\langle\psi| = \sum_{i,j} c_{ij}|i\rangle_A|j\rangle_B \sum_{i,j} c^*_{i,j}\ {}_A\langle i|_B\langle j| \tag{2.25}$$

Now we use an orthonormal basis $\{|j\rangle_B\} \in \mathscr{H}_B$ for subsystem B, and take the *partial trace* Tr_B over B, which is defined similarly to the trace in equation 2.11.

$$\mathrm{Tr}_B(\rho_{AB}) = \sum_j {}_B\langle j|\rho_{AB}|j\rangle_B \tag{2.26}$$

When we plug in the expression for ρ_{AB} from equation 2.25 into 2.26 and use the orthonormality of basis $\{|j\rangle_B\}$, we are left with the following expression:

$$\mathrm{Tr}_B(\rho_{AB}) = \sum_i c_i c_i^* |i\rangle_{A\,A}\langle i| \tag{2.27}$$

After taking $p_i = c_i c_i^*$, we conclude that we derived another expression for the reduced density matrix ρ_A of subsystem A. A similar expression can be found for ρ_B:

$$\begin{aligned} \rho_A &= \mathrm{Tr}_B \rho_{AB} \\ \rho_B &= \mathrm{Tr}_A \rho_{AB} \end{aligned} \tag{2.28}$$

With knowledge of the partial trace, we can show that the Schmidt coefficients relate to the eigenvalues of ρ_A, p_i. To show this, we compute ρ_A by taking the partial trace of ρ_{AB} using the Schmidt decomposition in equation 2.24:

$$\begin{aligned} \rho_A = \mathrm{Tr}_B\rho_{AB} = \mathrm{Tr}|\psi\rangle_{AB\,AB}\langle\psi| &= \mathrm{Tr}_B \sum_i \lambda_i |i\rangle_A|i'\rangle_B \sum_i \lambda_i^*\ {}_A\langle i|_B\langle i'| \\ &= \mathrm{Tr}_B \sum_i \lambda_i^2 |i'\rangle_B|i\rangle_A\ {}_A\langle i|_B\langle i'| = \sum_i \lambda_i^2|i\rangle_A\ {}_A\langle i| \equiv \sum_i p_i|i\rangle_A\ {}_A\langle i| \end{aligned} \tag{2.29}$$

It follows that $\lambda_i^2 = p_i$.

2.3.3. Density matrix renormalization group (DMRG) algorithm

As discussed before, the complexity of large one-dimensional quantum systems lies in the dimension of the Hilbert space of such systems, as the dimension of $\mathscr{H}$ grows exponentially with system size. The idea behind infinite DMRG is to start with a small (two-site) system and gradually expand this by adding new sites, all while keeping the dimension of the vectors and matrices low. This can be achieved by diagonalizing the density matrix in every iteration and truncating it by only keeping the eigenvectors corresponding to the largest eigenvalues. If the sum of these eigenvalues is close to 1, these eigenvalues are supposed to represent the physics well enough [21]. This procedure is explained in more detail below and visualized in figure 2.3.

1. Set up a one-dimensional chain of identical quantum sites with dimension d and total length $L = 2$, and define its Hamiltonian H. As this system is still small, it is possible to directly diagonalize the Hamiltonian and obtain the ground state of the system $|\psi\rangle_U$ and the corresponding eigenvalue.

2. Call the total chain *universe (U)* and split this up into two subsystems, the *system (S)* and the *environment (E)*. Note that this is equivalent to the use of subsystems A and B in the previous section. We use $\{|i\rangle_S\}$ and $\{|l\rangle_E\}$ as bases for the Hilbert spaces $\mathscr{H}_S$ and $\mathscr{H}_E$. Similar to 2.20, we write the wave function of the ground state $|\psi\rangle_U$ in terms of eigenfunctions of S and E:

$$|\psi\rangle_U = \sum_{i,l} c_{il}|i\rangle_S|l\rangle_E \tag{2.30}$$

   In this sum, indices i and l are chosen in such a way that they still are the most intuitive in later steps. Note that in the initial setup of the chain, the dimension of the Hilbert spaces $\mathscr{H}_S$ and $\mathscr{H}_E$ — N^S and N^E respectively — is d, and the dimension of the Hilbert space $\mathscr{H}_U$ — N^U — of the total chain is d^2. Hamiltonian H_U of the chain can be split up into three parts:

$$H_U = H_S + H_E + H_{SE} \tag{2.31}$$

   where H_S is the Hamiltonian of the system S, H_E is the Hamiltonian of environment E and H_{SE} contains the interacting terms between S and E. If there is only next-neighbour interaction between S and E, H_{SE} is a two-site operator connecting the two subsystems.

3. After splitting up U in S and E, add two identical sites of dimension d in between them. The basis states of the new sites again have dimension d and are denoted by $|j\rangle_S$ and $|k\rangle_E$. Using Lanczos diagonalization, we find the wave function of the ground state of the extended system, and write this function in terms of $|i\rangle_S$, $|j\rangle_S$, $|k\rangle_E$ and $|l\rangle_E$. The wave function of the ground state of the extended system is:

$$|\psi\rangle_U = \sum_{i,j,k,l} c_{ijkl}|i\rangle_S|j\rangle_S|k\rangle_E|l\rangle_E \tag{2.32}$$

4. Update the wave functions for system (S) and environment (E), by taking the tensor products $|\tilde{i}\rangle_S = |i\rangle_S \otimes |j\rangle_S$ and $|\tilde{l}\rangle_E = |k\rangle_E \otimes |l\rangle_E$ respectively. The updated wave function of the ground state of the system $|\tilde{\psi}_U\rangle$ in the updated formulation is:

$$|\tilde{\psi}\rangle_U = \sum_{i,l} c_{il}|\tilde{i}\rangle_S|\tilde{l}\rangle_E \tag{2.33}$$

   Dimension $\tilde{N}^S$ grows to $N^S \times d$ (or d^2 in the first iteration). The same holds for $\tilde{N}^E$. Again H_U can be split up in H_S, H_E and H_{SE}. The dimensions of H_S and H_E similarly grow to $\tilde{N}^S$ and $\tilde{N}^E$.

5. Now it is important to reduce $\tilde{N}^S$ and $\tilde{N}^E$, as we grow the system size iteratively. We need to find a basis of the lowest lying eigenstates of the density matrices of both S and E, in order to update H_S, H_{SE} and H_E into matrices of smaller size. As described in section 2.1.3, S and E cannot explicitly be represented by a wave function, as they interact with each other. Therefore we take a look at the density matrices ρ_S and ρ_E. To obtain these, we define $\rho_U = |\tilde{\psi}\rangle_{UU}\langle\tilde{\psi}|$. As in equation 2.28, ρ_S can be found by taking the partial trace of ρ_U:

$$\rho_S = \mathrm{Tr}_E \rho_U, \tag{2.34}$$

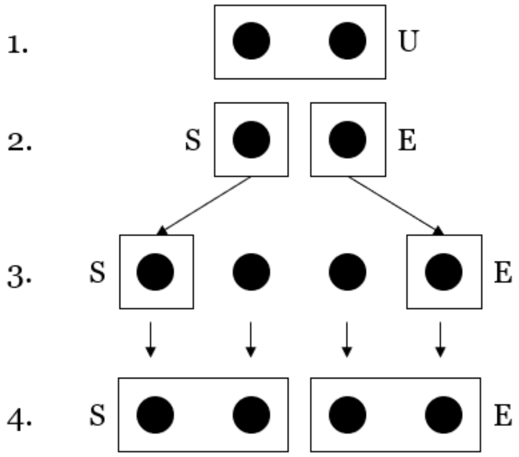

Figure 2.3: Step-by-step visualization of the DMRG algorithm

We find the eigenbasis $\mathscr{B}_S$ of ρ_S consisting of eigenvectors $|w\rangle$ using SVD. This eigenbasis is ordered by weights w, that sum up to 1, thanks to property 2.12. Now we truncate ρ_S by creating a new eigenbasis $\mathscr{B}'_S$ that consists of only χ eigenvectors from $\mathscr{B}_S$ with the highest weights w. The number χ is chosen in such a way that the sum of the weights of the vectors in $\mathscr{B}'_S$ is sufficiently close to 1, in order to keep accuracy. We use our newly found basis to truncate H_S into H_S^{tr} to an operator of size $\chi \times \chi$ by transforming to the reduced basis is explained by Schollwöck [21]. The same routine can be followed for ρ_E and H_E to obtain H_E^{tr}. After finding truncated eigenbases $\mathscr{B}'_S$ and $\mathscr{B}'_E$ for ρ_S and ρ_E, it is possible to rewrite H_{SE} intro H'_{SE} in truncated eigenbase $\mathscr{B}'_U = \mathscr{B}'_S \otimes \mathscr{B}'_E$. The new expression for the Hamiltonian is:

$$H_U^{tr} = H_S^{tr} + H_E^{tr} + H_{SE}^{tr} \tag{2.35}$$

6. With the new truncated Hamiltonian H_U^{tr}, it is possible to directly calculate the ground state and its energy via direct or Lanczos diagonalization. After this, we return to step 3, where we again insert two sites in the chain and run the procedure until a sufficient accuracy for the ground state energy is found.

2.4. Josephson junctions

The qubit that we study in this thesis is made out of a circuit of Josephson junctions (JJ). In this section, we take a look at the underlying physics of Josephson junctions, in order to understand the behaviour of this device. Later in this thesis, this knowledge is used to construct the Hamiltonian of the qubit and calculate its ground state energy.

2.4.1. Superconductivity

When electrons travel through metals, they generally experience some form of electrical resistance, due to interactions with the material. However, in certain materials, when the temperature is below a *critical temperature*, this resistance drops to zero. This phenomenon is called *superconductivity*. Similar to the systems previously described, superconducting materials can also be described by a wave function, called the *macroscopic wave function* [22]:

$$\psi(\mathbf{r}, t) = \sqrt{\rho(\mathbf{r}, t)}e^{i\theta(\mathbf{r}, t)} \tag{2.36}$$

In this relation, ρ is the charge density in the material and θ is the *quantum phase* of the material. Within superconducting materials, *Cooper pairing* [23] occurs. Cooper pairing is the mechanism where two electrons with opposite spin and momentum may together form a bound state because of the interaction between electrons and phonons.

2.4.2. Josephson effect

The coupling of two superconducting materials with a sufficiently thin layer of non-superconducting or insulating material in between, is called a *Josephson junction*. Both superconducting materials have a macro-

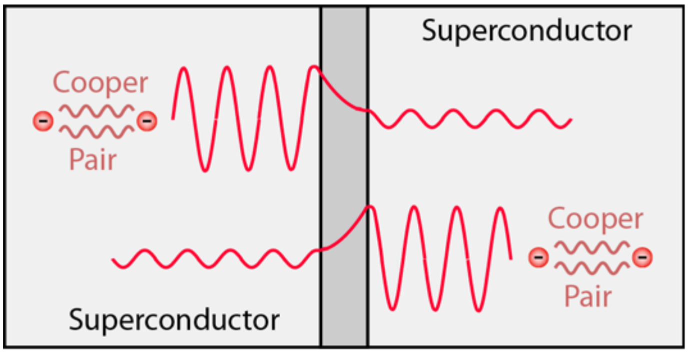

Figure 2.4: Schematic illustration of a Josephson junction, constructed out of two parts of superconducting material seperated by insulating material. *(Source: http://hyperphysics.phy-astr.gsu.edu/hbase/Solids/Squid.html)*

scopic wave functions ψ_1, ψ_2 and quantum phases θ_1, θ_2. Brian Josephson predicted [7] in his 1962 paper that the aforementioned Cooper pairs might tunnel through the insulating layer - even when zero voltage is applied over the junction - and lead to a superconducting current. This effect is called the DC Josephson effect. The Josephson current depends on the phase difference $\delta = \theta_1 - \theta_2$ between the two materials [24]:

$$I_J = I_0 \sin\delta \tag{2.37}$$

In this equation, I_0 is the maximum supercurrent that the junction can carry. Another phenomenon occurs when a voltage is applied over a Josephson junction. The first time derivative of δ depends linearly on the applied voltage V:

$$\frac{d\delta}{dt} = \frac{2eV}{\hbar} \tag{2.38}$$

From equations 2.37 and 2.38 we derive the classical Hamiltonian for the Josephson junction. Classically, the Hamiltonian consists of two parts: the *induction energy* and *capacitive energy*. The contribution of the inductance is:

$$U_J = \int_0^t I_J V dt = \frac{I_0\hbar}{2e}\int_0^t \sin\delta' \frac{d\delta'}{dt} dt = \frac{I_0\hbar}{2e}\int_0^t \sin\delta' d\delta' = \frac{I_0\hbar}{2e}(1-\cos\delta) = E_J(1-\cos\delta) \tag{2.39}$$

where we defined *Josephson energy* $E_J = \frac{I_0\hbar}{2e}$. If we look at the second part, the capacitive energy, we can find this by:

$$U_C = \frac{1}{2}CV^2 = \frac{1}{2}\frac{Q^2}{C} = \frac{1}{2}\frac{(2en)^2}{C} = \frac{2e^2n^2}{C} = E_C n^2 \tag{2.40}$$

where we defined *charging energy* $E_C = \frac{2e^2}{C}$ and introduced the difference in Cooper pairs between the two superconducting materials n. Combining these two terms leads to the classical Hamiltonian for the Josephson junction:

$$H = E_C n^2 + E_J(1-\cos\delta) \tag{2.41}$$

Generally, we distinguish two regimes: the *phase regime* ($E_C \ll E_J$) and the *charge regime* ($E_J \ll E_C$). It is possible to rewrite the Hamiltonian in equation 2.41 in terms of phase difference δ. To achieve this, we transform the classical Hamiltonian above, where n and δ are regarded as continuous variables, by quantizing n and δ in 2.41 into quantum operators ($\hat{n}$ and $\hat{\delta}$). Now, they obey a commutation relation between them [8]:

$$[\hat{\delta}, \hat{n}] = i \tag{2.42}$$

This relation is similar to the commutation relation between position $\hat{x}$ and momentum $\hat{p}$ ($[\hat{x},\hat{p}] = i\hbar$). We extend this similarity to find a relation between $\hat{n}$ and $\hat{\delta}$:

$$\hat{n} = -i\frac{d}{d\delta} \tag{2.43}$$

The Hamiltonian operator in terms of $\hat{\delta}$ is:

$$\hat{H} = -E_C\frac{d^2}{d\delta^2} + E_J(1-\cos\delta) \tag{2.44}$$

2.4.3. Josephson junction chains

Next to individual Josephson junctions, it is possible to create devices that contain multiple connected Josephson junctions. Therefore, we look at how these junctions interact with each other. We start off with a one-dimensional array of Josephson junctions. In the previous section we have seen that the Hamiltonian of Josephson junctions depend on the phase difference δ over the junction. This allows us to rewrite the Hamiltonian in terms of ϕ_i, the phase at a certain island between two junctions. It holds that $\delta_i = \phi_i - \phi_{i+1}$. For a chain of L Josephson junctions, there are $L-1$ intermediate islands, meaning that this frees up a degree of freedom [8]. As a result, we can impose one of the islands to a fixed phase. It is convenient to set the first phase $\phi_0 = 0$. We can impose a fixed boundary condition by setting $\phi_{L+1} = \Delta\theta$. The Hamiltonian from equation 2.44 for junction i as a function of ϕ_i becomes:

$$\hat{H}_i = -E_C \frac{d^2}{d\phi_i^2} + 1 - \cos\left(\phi_i - \phi_{i+1}\right) \tag{2.45}$$

where we set $E_J = 1$. For the complete chain, the Hamiltonian becomes:

$$\begin{aligned} \hat{H} = \sum_{i=1}^{L} \hat{H}_i &= \sum_{i=1}^{L} \left[-E_C \frac{d^2}{d\phi_i^2} + 1 - \cos\left(\phi_i - \phi_{i+1}\right) \right] \\ &= 2 - \cos\left(\phi_1\right) - \cos\left(\phi_L - \Delta\theta\right) + \sum_{i=2}^{L-1} \left[-E_C \frac{d^2}{d\phi_i^2} + 1 - \cos\left(\phi_i - \phi_{i+1}\right) \right] \end{aligned} \tag{2.46}$$

The chain of JJ with boundary condition $\phi_{L+1} = \Delta\theta$ has a classical analogy of a chain of springs that are stretched over a fixed distance $\Delta\theta$. This approximation only holds when δ is sufficiently small, as $\cos\delta \approx \delta^2$ in that case. The chain of stretched strings is in its lowest-energy configuration when all strings are stretched uniformly. The same holds for the chain of Josephson junctions. Therefore it is convenient to make the following transformation:

$$\phi_i \to \tilde{\phi}_i = \phi_i + i\Delta\theta/L \tag{2.47}$$

Instead of dealing with the boundary condition of $\Delta\theta$, the total "stretch" is now uniformly imposed at each junction, leading to the following Hamiltonian:

$$\hat{H} = \sum_{i=1}^{L} \left[-E_C \frac{d^2}{d\tilde{\phi}_i^2} + 1 - \cos\left(\tilde{\phi}_i - \tilde{\phi}_{i+1} + \Delta\theta/L\right) \right] \tag{2.48}$$

This Hamiltonian will be used in simulations using Josephson junctions in chapter 4.

2.5. A superconducting qubit of Josephson junctions

The system of particular interest in this thesis is a superconducting qubit, which was originally proposed by Hans Mooij. A schematic representation of the qubit is given in figure 2.5. This qubit is exposed to an external magnetic flux of strength $f\Phi$. When the flux is exactly two flux quanta Φ_0, the structure behaves as a qubit with two ground states, that represent the situation where two non-adjacent inner squares — i.e. n_1 and n_4 or n_2 and n_3 — both contain one flux quantum. Any closed loop of Josephson junctions is subject to the *fluxoid quantization condition* [8], which means that the sum of the phase difference of all junctions on this loop should be a multiple of 2π. Formally, this condition is written as:

$$\sum_{\text{contour}} \gamma_i = 2\pi(n - f) \tag{2.49}$$

where γ_i is the phase difference over each junction on the loop and n is the fluxoid number that is contained in the loop. This means that the frustration f should be distributed over the junctions in outer branch of the loop.

A high number of θ_i junctions on the outer branch of the structure makes the qubit "better defined". If the number is large, the phase difference over the individual junctions drops. However, these junctions come at a high cost in terms of computability. For a simpler version of the qubit, the number of junctions θ_i may be reduced to for example 2 per quarter. Furthermore, the ψ_i branches could be removed. This leaves us with a

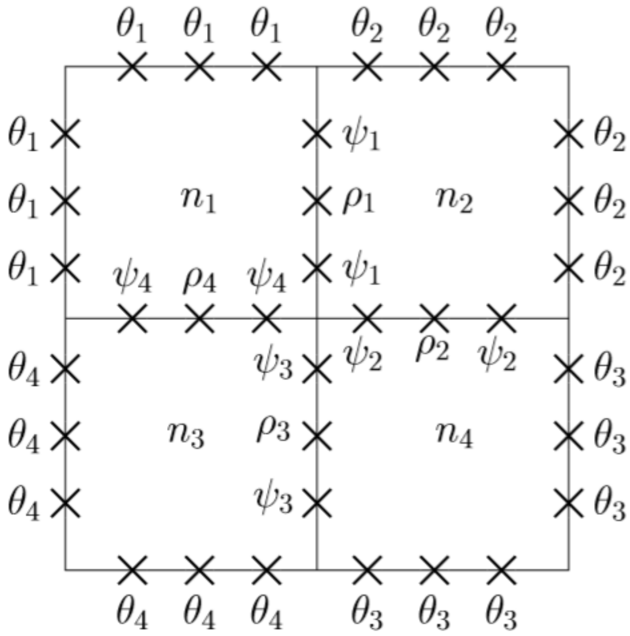
θ_1 θ_1 θ_1 θ_2 θ_2 θ_2
θ_1 θ_1 θ_1
ψ_1 ρ_1 ψ_1
n_1 n_2 n_3 n_4
θ_2 θ_2 θ_2
ψ_4 ρ_4 ψ_4
ψ_2 ρ_2 ψ_2
ψ_3 ρ_3 ψ_3
θ_4 θ_4 θ_4
θ_3 θ_3 θ_3
θ_4 θ_4 θ_4 θ_3 θ_3 θ_3

Figure 2.5: Schematic representation of the qubit proposed by Hans Mooij. All crosses represent a possible Josephson junction, depending on the exact configuration of the qubit. *(From [8])*

much simpler proposition of the qubit, as can be seen in figure 2.6.

Note that only 12 junctions remain in this simplified qubit. It is now possible to write the Hamiltonian in the *charge regime*, i.e. $E_J \ll E_C$, in terms of the phases of islands ϕ_i:

$$\hat{H} = \sum_{i=1}^{12} 1 - \cos\left(\delta_i - f_i\right) = \sum_{i \text{ outer junctions}}^{8} 1 - \cos\left(\delta_i - f/8\right) + \alpha \sum_{i \text{ inner junctions}}^{4} 1 - \cos\left(\delta_i\right) \tag{2.50}$$

We are allowed to set the phase of the central island to ϕ_c, so the expression above reduces to [8]:

$$\hat{H} = 12 + \sum_{i=1}^{8} 1 - \cos\left(\phi_i - \phi_{i+1} - f/8\right) + \sum_{i=1,3,5,7} \cos\left(\phi_i\right) \tag{2.51}$$

We will use this expression in our analysis of the qubit in chapter 4.

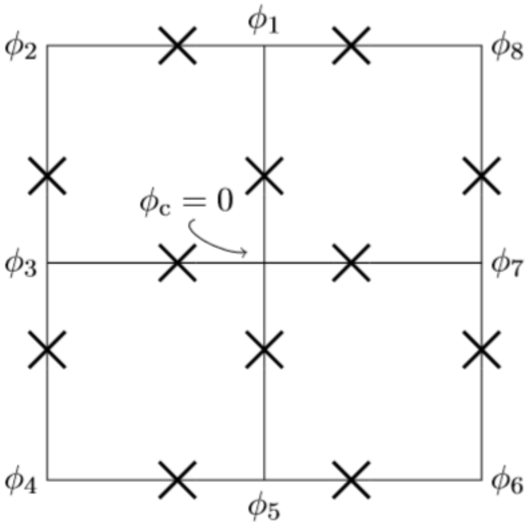

Figure 2.6: Schematic representation of a simplified version of the qubit proposed by Hans Mooij. There are only 2 θ_i junctions per quarter and all ψ_i junctions are removed. *(From [8])*

3
Matrix product states and time-evolving block decimation

The one-dimensional interacting systems described in the previous chapters tend to grow exponentially with system size. In order to analyze these systems, it is important to keep the computational costs low by reduce the required amount of stored data. This has led to the development of several numerical methods for this purpose, of which some are outlined by Noack [15]. White's DMRG method, that is introduced in chapter 2, is the best known method to analyze such systems. At the core of the DMRG algorithm, is the decomposition of the chain into a bipartite system. This decomposition could be done repeatedly, leading to another class of methods, namely that of *matrix product states (MPS)* [10, 21]. We first introduce the concept of MPS in section 3.1 and then explain a related algorithm for finding the ground-state energy, called *time-evolving block decimation (TEBD)* in section 3.2. In the next chapter 4.1, the TEBD algorithm is applied to both the Heisenberg model and a chain of Josephson junctions in order to explore the workings of the method, before we analyze the qubit of our interest.

3.1. Matrix product states

Consider a one-dimensional quantum system that consists of L sites that all have a Hilbert space of dimension d, with basis $\{|\sigma_i\rangle\}_{i=1,2,...,d}$. The system has a Hilbert space $\mathscr{H}$ of dimension d^L, and any wave function $|\psi\rangle \in \mathscr{H}$ can be written as follows:

$$|\psi\rangle = \sum_{\sigma_1,\sigma_2,...,\sigma_L} c_{\sigma_1\sigma_2...\sigma_L}|\sigma_1\sigma_2...\sigma_L\rangle \tag{3.1}$$

In this representation, the sum runs over the d^L basis states of $\mathscr{H}$, hence $c_{\sigma_1\sigma_2...\sigma_L}$ is a vector of size d^L. In the formalism of matrix product states, $|\psi\rangle$ can be rewritten as:

$$|\psi\rangle = \sum_{\sigma_1,\sigma_2,...,\sigma_L} A^{\sigma_1}A^{\sigma_2}...A^{\sigma_L}|\sigma_1\sigma_2...\sigma_L\rangle \tag{3.2}$$

In equation 3.2, A^{σ_1} is a row vector of size χ_1, A^{σ_i} are matrices of size $\chi_{i-1} \times \chi_i$ and A^{σ_L} is a column vector of size χ_L. The variables χ_i are known as the *bond dimensions* and determine the amount of stored information. By choosing variables χ_i in the right way, the product of A matrices is a reduction in size in comparison to the c tensor in equation 3.1, without losing accuracy. Just as the *ket representation* or the *density matrix*, the matrix product state of equation 3.2 is a method to represent a state of the system, that stores enough information about the state to use for analysis.

3.1.1. Constructing matrix product states

First, we show that there exists an exact equivalence between the wave function in equation 3.1 and a matrix product representation. The procedure to show this is described in detail by Schollwöck [20]. In section 3.1.4, we show that the reduction in size can be obtained by truncating the matrix product representation to the form in equation 3.2.

The procedure starts by rewriting the column vector $c_{\sigma_1\sigma_2...\sigma_L}$ as matrix $\Psi_{\sigma_1,(\sigma_2...\sigma_L)}$ of dimension $d\times d^{L-1}$. The next step is to use singular value decomposition (SVD) to expand Ψ into a product of matrices:

$$\begin{aligned} c_{\sigma_1\sigma_2...\sigma_L} &= \Psi_{\sigma_1,(\sigma_2...\sigma_L)} \\ &= \sum_{a_1}^{r_1} U_{a_1\sigma_1}\Lambda_{a_1a_1}V^{\dagger}_{a_1(\sigma_2...\sigma_L)} \\ &= \sum_{a_1}^{r_1} A^{\sigma_1}_{a_1}\Lambda_{a_1a_1}V^{\dagger}_{a_1(\sigma_2...\sigma_L)} \\ A &= U\Lambda V^{\dagger} \end{aligned} \tag{3.3}$$

In this expression, $\Lambda_{a_1a_1}$ contains the r_1 singular values of $\Psi_{\sigma_1,(\sigma_2...\sigma_L)}$, where r_1 is the rank of $\Psi_{\sigma_1,(\sigma_2...\sigma_L)}$. $r_1 \le d$, so a_1 runs from 1 to d at most. In the last step, matrix $U_{a_1\sigma_1}$ is reshaped into d vectors of dimension r_1 which we call $A^{\sigma_1}_{a_1}$. The next step is to multiply Λ and $V^{\dagger}$ and reshape the result into $\Psi_{(a_1\sigma_2),(\sigma_3...\sigma_L)}$, which has dimension $(r_1d\times d^{L-2})$. We again apply singular value decomposition, this time on $\Psi_{(a_1\sigma_2),(\sigma_3...\sigma_L)}$ to further expand our initial wave function:

$$\begin{aligned} c_{\sigma_1\sigma_2...\sigma_L} &= \sum_{a_1}^{r_1} A^{\sigma_1}_{a_1}\Psi_{(a_1\sigma_2),(\sigma_3...\sigma_L)} \\ &= \sum_{a_1}^{r_1}\sum_{a_2}^{r_2} A^{\sigma_1}_{a_1}U_{(a_2\sigma_1)a_2}\Lambda_{a_2a_2}V^{\dagger}_{a_2(\sigma_3...\sigma_L)} \\ &= \sum_{a_1}^{r_1}\sum_{a_2}^{r_2} A^{\sigma_1}_{a_1}A^{\sigma_2}_{a_1a_2}\Lambda_{a_2a_2}V^{\dagger}_{a_2(\sigma_3...\sigma_L)} \\ &= \sum_{a_1}^{r_1}\sum_{a_2}^{r_2} A^{\sigma_1}_{a_1}A^{\sigma_2}_{a_1a_2}\Psi_{(a_2\sigma_3),(\sigma_4...\sigma_L)} \end{aligned} \tag{3.4}$$

$\Lambda_{a_2a_2}$ contains the r_2 singular values of $\Psi_{(a_1\sigma_2),(\sigma_3...\sigma_L)}$ and $U_{(a_1\sigma_1)a_2}$ is reshaped into d matrices of dimension $(r_1\times r_2)$. $\Psi_{(a_2\sigma_3),(\sigma_4...\sigma_L)}$ is the product of $\Lambda_{a_2a_2}$ and $V^{\dagger}_{a_2(\sigma_3...\sigma_L)}$ and has dimension $(r_2d\times d^{L-3})$, where $r_2 \le r_1d \le d^2$. After performing the SVD $L-1$ times, we arrive at the following form:

$$c_{\sigma_1\sigma_2...\sigma_L} = \sum_{a_1,a_2,...a_{L-1}} A^{\sigma_1}_{a_1}A^{\sigma_2}_{a_1a_2}...A^{\sigma_{L-1}}_{a_{L-2},a_{L-1}}A^{\sigma_L}_{a_{L-1}} \tag{3.5}$$

where all sums over a_i run over r_i. After summing over all a_i, the expression above reduces to

$$c_{\sigma_1\sigma_2...\sigma_L} = A^{\sigma_1}A^{\sigma_2}...A^{\sigma_L} \tag{3.6}$$

We substitute this expression in equation 3.1 to arrive at the desired MPS form in equation 3.2.

3.1.2. Canonical form

To see why writing the wave function in matrix product state form is relevant, we need to perform an additional step, namely bringing the expression into *canonical form.* To achieve this, we reuse the diagonal matrices $\Lambda^{[i]} = \Lambda_{a_ia_i}$ that were created in the SVD steps, and insert the identity $\Lambda^{[i]}(\Lambda^{[i]})^{-1}$ between the matrix products in equation 3.2:

$$|\psi\rangle = \sum_{\sigma_1,\sigma_2,...,\sigma_L} A^{\sigma_1}\Lambda^{[1]}(\Lambda^{[1]})^{-1}A^{\sigma_2}...\Lambda^{[L-1]}(\Lambda^{[L-1]})^{-1}A^{\sigma_L}|\sigma_1\sigma_2...\sigma_L\rangle \tag{3.7}$$

Now we perform the following transformation

$$\begin{aligned} \Gamma^{\sigma_1} &= A^{\sigma_1} \\ \Gamma^{\sigma_i} &= \Lambda^{[i-1]}A^{\sigma_i} \text{ for } i = 2,...,L \end{aligned} \tag{3.8}$$

to arrive at another representation of matrix product states, the *canonical form*:

$$|\psi\rangle = \sum_{\sigma_1,\sigma_2,...,\sigma_L} \Gamma^{\sigma_1}\Lambda^{[1]}\Gamma^{\sigma_2}...\Gamma^{\sigma_{L-1}}\Lambda^{[L-1]}\Gamma^{\sigma_L}|\sigma_1\sigma_2...\sigma_L\rangle \tag{3.9}$$

This canonical MPS form of the wave function becomes very useful for DMRG, as it can be rewritten into the form of the Schmidt decomposition, as Schollwöck [21] shows. The Schmidt decomposition of a bipartition on site i takes on the following form [8]:

$$|\psi\rangle = \sum_{a_i}^{r_i} \Lambda^{[i]}_{a_i a_i} |\eta_{a_i}\rangle_A |\eta_{a_i}\rangle_B \tag{3.10}$$

where $|\eta_{a_i}\rangle_A$ and $|\eta_{a_i}\rangle_B$ are the following *Schmidt vectors*:

$$|\eta_{a_i}\rangle_A = \sum_{\sigma_1,\sigma_2,\ldots,\sigma_i} \sum_{a_1,a_2,\ldots,a_{i-1}} \Gamma^{\sigma_1}_{a_1} \Lambda^{[1]}_{a_1 a_1} \Gamma^{\sigma_2}_{a_1 a_2} \ldots \Lambda^{[i-1]}_{a_{i-1} a_{i-1}} \Gamma^{\sigma_i}_{a_{i-1} a_i} |\sigma_1 \sigma_2 \ldots \sigma_i\rangle \tag{3.11}$$

$$|\eta_{a_i}\rangle_B = \sum_{\sigma_{i+1},\sigma_{i+2},\ldots,\sigma_L} \sum_{a_{i+1},a_{i+2},\ldots,a_L} \Gamma^{\sigma_{i+1}}_{a_i a_{i+1}} \Lambda^{[i+1]}_{a_{i+1} a_{i+1}} \Gamma^{\sigma_{i+2}}_{a_{i+1} a_{i+2}} \ldots \Lambda^{[L-1]}_{a_{L-1} a_{L-1}} \Gamma^{\sigma_L}_{a_{L-1} a_L} |\sigma_{i+1} \sigma_{i+2} \ldots \sigma_L\rangle \tag{3.12}$$

3.1.3. Graphical representation of matrix product states

The derivation of MPS in the previous sectors is highly detailed and therefore a clear visualization could be helpful. We start with the representation of the original wave function in equation 3.1. Here, the complex coefficient $c_{\sigma_1 \sigma_2 \ldots \sigma_L}$ is written as a block with L legs, that all represent a site of the one-dimensional chain (Figure 3.1). After that, the wave function is iteratively converted into MPS form through multiple SVD, which is described by equation 3.3. The graphical representation of this process can be found in figure 3.2. Matrix A^{σ_1} is represented by the black dot, diagonal matrix $\Lambda^{[1]}$ by the transparant rhombus and the remainder of the coefficient is the black block. After $L-1$ iterations, the matrix product state is represented by the L dots, connected by $L-1$ black lines. This can again be transformed to the canonical form, which is depicted in figure 3.3.

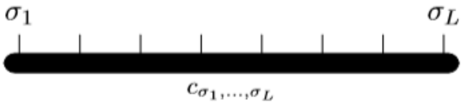

Figure 3.1: Graphical representation of wave function in the original complex vector form (*from* [8])

3.1.4. Entanglement and MPS truncation

Any of the most important advantages of the MPS representation, is its ability to truncate the wave function. For a better understanding, we take a closer look at the concept *entanglement*. For this purpose, we follow the reasoning of Feiguin [18]. Entanglement is property of a quantum system that consists of two or more units. When the system is entangled, this means that you cannot describe one unit of the system — or a part of the system — without having knowledge of other parts of the system. To illustrate this, we look at a simple system consisting of two electron spins. Suppose this system is in an equal superposition of all possible configurations:

$$|\psi\rangle = |\uparrow\uparrow\rangle + |\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle + |\downarrow\downarrow\rangle \tag{3.13}$$

It is straightforward that this superposition can be rewritten as the product of the states of the two individual spins:

$$|\psi\rangle = (|\uparrow\rangle + |\downarrow\rangle) \otimes (|\uparrow\rangle + |\downarrow\rangle) \tag{3.14}$$

This means that one spins can be in both states, without any dependence on the state of the other spin. This changes when we look at the next possible state, the so called *Bell state*:

$$|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle \tag{3.15}$$

In the Bell state, the system is in a superposition of only two possible configurations and it is clear that both spins are completely dependent on each other. If we measure the state of the first spin to be $|\uparrow\rangle$, the second spin can only be $|\uparrow\rangle$ as well, and vice versa. We call this system entangled. The advantage of entangled systems is that information about a small part of the system suffices to have complete knowledge about the complete system. The more a system is entangled, the more efficient its information can be stored.

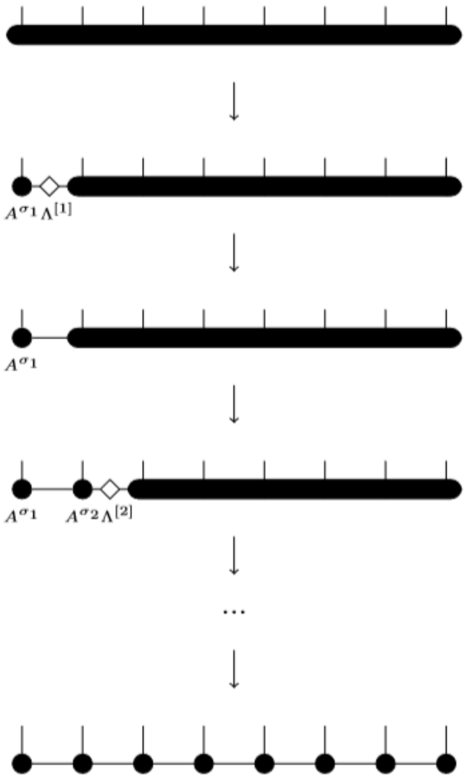
A^{σ_1} $\Lambda^{[1]}$
A^{σ_1}
A^{σ_1} A^{σ_2} $\Lambda^{[2]}$
...

Figure 3.2: Graphical conversion of wave function into MPS form (*from* [8])

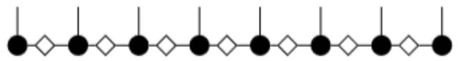

Figure 3.3: MPS wave function in canonical form (*from* [8])

It is possible to generalize this concept to larger systems that can be partitioned into subsystems. As we have seen in the previous section, any wave function that can be written in MPS representation, can be written as a Schmidt decomposition at any site i (equation 3.10):

$$|\psi\rangle = \sum_{a_i}^{r_i} \Lambda^{[i]}_{a_i a_i} |\eta_{a_i}\rangle_A |\eta_{a_i}\rangle_B \tag{3.16}$$

In this expression, all $\Lambda^{[i]}_{a_i a_i}$ are the Schmidt coefficients of this decomposition, obtained through the SVD at site i. Remember from section 2.3.2 that these values are related to the eigenvalues of ρ_A, by $p_i = (\Lambda^{[i]}_{a_i a_i})^2$. Therefore, the values of $(\Lambda^{[i]}_{a_i a_i})^2$ cannot exceed 1 and should add up to 1. This is a crucial point in understanding why the MPS representation can be used to truncate states efficiently. If the coefficients decay fast enough, there occurs only a small error if we discard the states with the smallest coefficients, by keeping only the χ_i largest of the r_i singular values. All χ_i's can be chosen to be either a fixed number, or by choosing truncation errors ϵ_i and retaining the minimum number of coefficients that reach the appropriate accuracy. Verstraete and Cirac show that an upper bound on the total truncation error can be guaranteed in this way [25]:

$$|| \, |\psi\rangle - |\psi_{\text{trunc}}\rangle||^2 \leq 2 \sum_{i=1}^{L-1} \epsilon_i \tag{3.17}$$

Now we know why it is allowed to truncate matrix product states, it is interesting to look at the advantage in terms of storage complexity. In the original wave function in the form of equation 3.1, the coefficient $c_{\sigma_1\sigma_2\ldots\sigma_L}$ has complexity d^L, with d being the possible number of states of one chain, and L the length of the chain. Now, the truncated matrix product state in canonical form of equation 3.9 contains $L-1$ diagonal

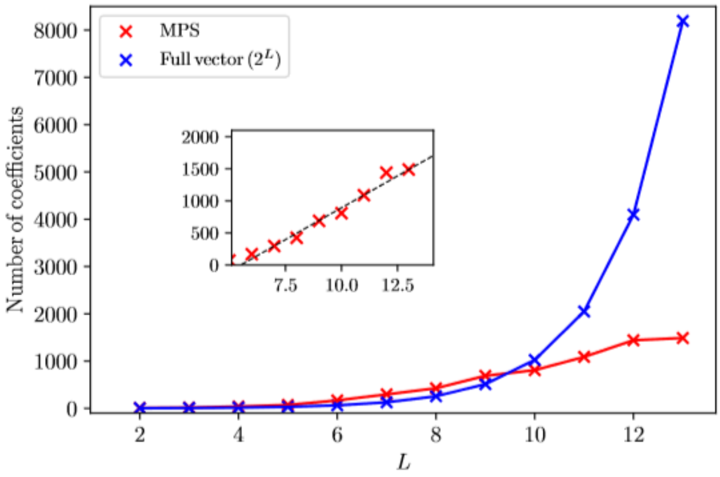

Figure 3.4: The number of coefficients needed to represent a Heisenberg chain ground state using the full vector and MPS representation. In the latter representation, truncation error $\epsilon_i = 10^{-6}$ is used to determine the number of states that need to be kept. (*from* [8]).

matrices $\Lambda^{[i]}_{a_i a_i}$, each with χ_i entries. Next to that, the MPS contains dL rectangular matrices $\Gamma^{\sigma_i}_{a_{i-1}a_i}$ with dimensions $\chi_{i-1} \times \chi_i$. If χ is fixed or if we suppose that $\chi = \max\{\chi_i\}$ exists — we explore this in the next sections — the overall complexity of the truncated matrix product state reduces to $O(dL\chi^2)$, which means that the storage complexity scales linearly with system size. It becomes clear that when χ is sufficiently small, that matrix product states become much more efficient then complete wave functions if L becomes larger. Melo illustrates this [8] by comparing the number of coefficients needed to represent the ground state of a Heisenberg chain by both the full vector and the MPS representation, as can be seen in figure 3.4. For more complex systems, with higher single-site dimension d, the advantages of MPS representation become even more clear.

3.1.5. Entropy of ground states: the area law

In the previous section we saw that MPS representation has advantages in storage complexity, because entanglement makes it possible to truncate the individual matrices to only χ states. Now it is important to see that it is allowed to truncate the Schmidt decomposition without loss in accuracy, and that χ is indeed smaller than d for the states within our interest — ground states of one-dimensional quantum chains.

To see why this is true, we take a look at the *entropy* of ground states. Entropy is a measure that is related to the amount of possible system configurations and therefore it is related to the entanglement of the system. In general, entropy is an extensive property, meaning that it scales with system volume. This also holds for quantum systems [26]. Entropy follows what is called a *volume law*. However, it can be shown [27] that the ground states of entangled systems is not extensive, but rather proportional to the area of the surface of the boundary between the partitions, obeying the *area law*. For 1D systems, this implies that the entropy of the ground state is a constant.

States that follow the area law, generally have lower entropy than other possible states of the system, as they are highly entangled. This impacts the Schmidt decomposition of such states. For highly entangled states with low entropy, one can expect that such states can be decomposed accurately using only a small number of states, in comparison to states with high entropy. If this is the case, it is indeed possible to truncate the Schmidt decomposition to an sufficient number of states (which is χ) as is suggested in the previous section. To illustrate this, Melo [8] compares the ground state and a random state of the aforementioned Heisenberg chain, which is included in figure 3.5. Figure 3.5a) shows the Schmidt coefficients of both states. For the random state, the significance of the Schmidt coefficients stays relatively constant, whereas this decays rapidly for the ground state. Vidal [28] shows that the Schmidt coefficients of a ground state of one-dimensional Hamiltonians generally decay exponentially. We can see this relation in figure 3.5b), where only a small number of states are needed to achieve a truncation accuracy close to 1 for the ground state. This suggests that it is possible to truncate the Schmidt decomposition of states that follow the area law to a finite number of χ states, as is done in the previous section.

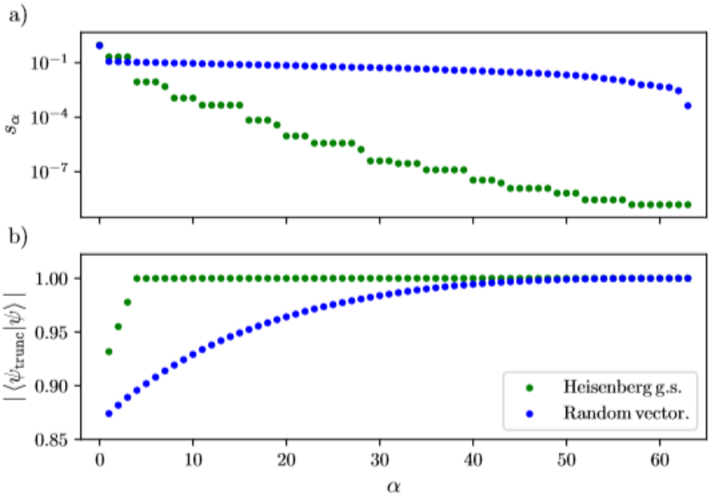

Figure 3.5: Comparison of entanglement spectra of the ground state and a random state of a Heisenberg chain with $L = 12$ sites. a) Schmidt coefficients in decreasing order b) accuracy of truncated state including α terms (*from* [8]).

3.1.6. MPS and the area law

The consequence of this argument is significant; when looking for ground states of a one-dimensional quantum system, it suffices to look into a small subset of the states in the Hilbert space: those that follow the area law. To see why matrix product states are suitable to represent ground states of one-dimensional quantum systems, we look at the entropy of MPS. The entanglement of a state $|\psi\rangle$ with density matrix ρ can be quantified through the *Von Neumann entropy*:

$$S(\rho) = -\mathrm{Tr}(\rho \log_2 \rho) \tag{3.18}$$

For a bipartition at any site i, the truncated MPS in the Schmidt decomposition follows from equation 3.16:

$$|\psi\rangle = \sum_{a_i}^{\chi} \Lambda^{[i]}_{a_i a_i} |\eta_{a_i}\rangle_A |\eta_{a_i}\rangle_B \tag{3.19}$$

Note that the sum only runs until the number χ is reached. When taking the density matrix of above expression and plug it into equation 3.18, we find the Von Neumann entropy for the truncated MPS to be [27]:

$$S(\rho) = -\mathrm{Tr}\left[\Lambda^{[i]}_{a_i a_i}\right]^2 \log_2 (\Lambda^{[i]}_{a_i a_i})^2 \leq O(\log_2 \chi) \tag{3.20}$$

Is χ is chosen as a constant, this entropy is bounded by a constant, meaning that the MPS also naturally follow an area law. This makes them suitable for exploring ground states of one-dimensional quantum states.

3.1.7. Applying operators on matrix product states

Next to efficient truncation of ground states, MPS have another advantage in the efficient performance of operations. In the rest of this thesis, we work with MPS to perform TEBD simulations on one-dimensional chains of electron spins and Josephson junctions. Therefore, we have a look into how operators can be applied to MPS. The Hamiltonians that we work with generally consist of one- and two-site operators. We restrict ourselves to nearest neighbour two-site operators in this section, as one-site operators can be generalized to this situation by taking a tensor product with a unit operator. This procedure is also described in [8]

If we want to apply two-site operator $O^{[i,i+1]}$ to sites i and $i+1$, we start with the MPS representation of the complete system in canonical form:

$$|\psi\rangle = \sum_{\sigma_1,\sigma_2,...,\sigma_L} \sum_{a_1,a_2,...,a_L} \Gamma^{\sigma_1}_{a_1} \Lambda^{[1]}_{a_1 a_1} \Gamma^{\sigma_2}_{a_1 a_2} ... \Gamma^{\sigma_{L-1}}_{a_{L-2} a_{L-1}} \Lambda^{[L-1]}_{a_{L-1} a_{L-1}} \Gamma^{\sigma_L}_{a_{L-1} a_L} |\sigma_1 \sigma_2 ... \sigma_L\rangle \tag{3.21}$$

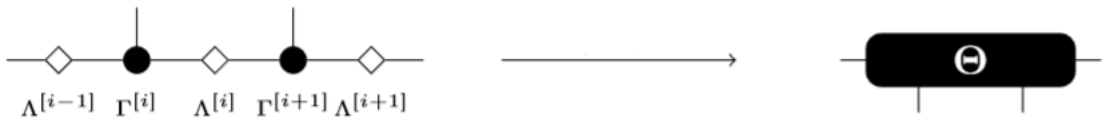

Figure 3.6: Graphical representation of the creation of tensor $\Theta^{\sigma_i \sigma_{i+1}}$ (*from* [8]).

As we are only interested in sites i and $i+1$, we can simplify this form by multiplying all matrices left of $\Lambda^{[i-1]}$ and right of $\Lambda^{[i+1]}$. For now, we discard these parts for the sake of readability and refer to [8] for a detailed account of what happens to the complete MPS. We continue with the part that remains and create tensor $\Theta^{\sigma_i \sigma_{i+1}}$ by contracting the remaining matrices over indices a_{i-1}, a_i and a_{i+1}:

$$\Theta^{\sigma_i \sigma_{i+1}}_{a_{i-1} a_{i+1}} |\sigma_i \sigma_{i+1}\rangle = \sum_{a_i} \Lambda^{[i-1]}_{a_{i-1} a_{i-1}} \Gamma^{\sigma_i}_{a_{i-1} a_i} \Lambda^{[i]}_{a_i a_i} \Gamma^{\sigma_{i+1}}_{a_i a_{i+1}} \Lambda^{[i+1]}_{a_{i+1} a_{i+1}} |\sigma_i \sigma_{i+1}\rangle \tag{3.22}$$

Analogous to the visual representation of matrix product states in section 3.1.3, figure 3.6 shows the creation of $\Theta^{\sigma_i \sigma_{i+1}}_{a_{i-1} a_{i+1}}$ from the contraction of $\Lambda^{[i-1]}_{a_{i-1} a_{i-1}} \Gamma^{\sigma_i}_{a_{i-1} a_i} \Lambda^{[i]}_{a_i a_i} \Gamma^{\sigma_{i+1}}_{a_i a_{i+1}} \Lambda^{[i+1]}_{a_{i+1} a_{i+1}}$. Now we apply operator $O^{[i,i+1]}$ and insert unit operator $\mathbb{1} = \sum_{\sigma'_i \sigma'_{i+1}} |\sigma'_i \sigma'_{i+1}\rangle\langle\sigma'_i \sigma'_{i+1}|$ to calculate the updated tensor $\tilde{\Theta}^{\sigma_i \sigma_{i+1}}_{a_{i-1} a_{i+1}}$:

$$\tilde{\Theta}^{\sigma_i \sigma_{i+1}}_{a_{i-1} a_{i+1}} |\sigma'_i \sigma'_{i+1}\rangle = \sum_{\sigma_i \sigma_{i+1}} \Theta^{\sigma_i \sigma_{i+1}}_{a_{i-1} a_{i+1}} \langle \sigma'_i \sigma'_{i+1} | O^{[i,i+1]} | \sigma_i \sigma_{i+1}\rangle \tag{3.23}$$

To restore the MPS representation of the updated system, we need to restore $\tilde{\Theta}^{\sigma_i \sigma_{i+1}}_{a_{i-1} a_{i+1}}$ into the original form of the matrix products, we reshape it to a matrix and apply SVD. Some minor alterations are needed to restore the final MPS form, for which we refer to [8]. The final result reads as follows:

$$O^{[i,i+1]}|\psi\rangle = \sum_{\sigma_1,\sigma_2,\ldots,\sigma_L} \sum_{a_1,a_2,\ldots,a_L} \Gamma^{\sigma_1}_{a_1} \ldots \tilde{\Gamma}^{\sigma'_i}_{a_{i-1} a_i} \tilde{\Lambda}^{[i]}_{a_i a_i} \tilde{\Gamma}^{\sigma'_{i+1}}_{a_i a_{i+1}} \ldots \Gamma^{\sigma_L}_{a_{L-1} a_L} |\sigma_1 \ldots \sigma'_i \sigma'_{i+1} \ldots \sigma_L\rangle \tag{3.24}$$

or in simplified form:

$$O^{[i,i+1]}|\psi\rangle = \sum_{\sigma_1,\sigma_2,\ldots,\sigma_L} \Gamma^{\sigma_1} \ldots \tilde{\Gamma}^{\sigma'_i} \tilde{\Lambda}^{[i]} \tilde{\Gamma}^{\sigma'_{i+1}} \ldots \Gamma^{\sigma_L} |\sigma_1 \ldots \sigma'_i \sigma'_{i+1} \ldots \sigma_L\rangle \tag{3.25}$$

3.2. Time-evolving block decimation (TEBD)

We have seen in the previous section that MPS are an effective way to represent quantum states that follow the area law and that the ground states of entangled quantum systems follow the same law. The DMRG algorithm, introduced in chapter 2, has established itself as an important technique to numerically analyze the ground state of many-body quantum states. However, the use of DMRG is mainly restricted to the analysis of the static properties of the ground states [29], whereas the time-evolution of these states is another problem of interest. Furthermore, the time evolution of the time-dependent Schrödinger equation can in fact be used as a method to reduce a random state to the ground state of the system.

The development of MPS opened the door to new techniques that can numerically simulate quantum states. The technique of interest in this thesis is time-evolving block decimation (TEBD), introduced by Vidal [29]. In this section, we introduce the algorithm and its underlying principles, and illustrate the application of TEBD on the Ising model and a chain of Josephson junctions.

3.2.1. Time evolution of a random quantum state

The time evolution of quantum state $|\phi\rangle$ is governed by the time-dependent Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \hat{H} |\psi\rangle \tag{3.26}$$

We start with an initial state $|\phi_0\rangle$, that can be described as a superposition of eigenstates of the Hamiltonian $|\psi_i\rangle$, that all have eigenvalues E_i:

$$|\phi_0\rangle = \sum_i c_i |\psi_i\rangle \tag{3.27}$$

The straightforward solution of the Schrödinger equation shows that $|\phi_t\rangle$ at time t comes in the following form:

$$|\phi_t\rangle = e^{-\frac{i\hat{H}t}{\hbar}}|\phi_0\rangle \tag{3.28}$$

For TEBD, we are only interested in *relative* time evolution, so in that case it is allowed to omit constant $\hbar$. When we introduce *imaginary time* $\tau = it$, we arrive at the *imaginary time evolution operator*: $e^{-\hat{H}\tau}$. Next, we apply this operator to the random initial state $|\phi_0\rangle$:

$$e^{-\hat{H}\tau}|\phi_0\rangle = \sum_i c_i e^{-\hat{H}\tau}|\psi_i\rangle = \sum_i c_i e^{-E_i\tau}|\psi_i\rangle \tag{3.29}$$

When the energy of the ground state E_g is smaller than all other energies E_i, this expression becomes proportional to the ground state $|\psi_g\rangle$, as the contributions of all other terms diminish relatively. Only when the ground state is degenerate, meaning that there are multiple states with the same lowest energy, the initial state converges to a superposition of these degenerate states.

3.2.2. TEBD algorithm

The exponentially increasing dimension of the Hilbert space for one-dimensional systems makes the time evolution operator hard to compute. However, the nature of the Hamiltonians of interest in this thesis could be split up into small components, which makes it much easier to compute. The Hamiltonians are built up from single- and two-site operators $O^{[i]}$ and $O^{[i,i+1]}$, making it possible to split them up into an even and odd part:

$$\hat{H} = \sum_{i=1}^{L-1} O^{[i,i+1]} + \sum_{i=1}^{L} O^{[i]} = \hat{H}_{\text{even}} + \hat{H}_{\text{odd}} \tag{3.30}$$

with

$$\hat{H}_{\text{even}} = \sum_{\text{even } i} O^{[i,i+1]} + O^{[i]} = \sum_{\text{even } i} \hat{H}^{[i]}_{\text{even}} \tag{3.31}$$

and

$$\hat{H}_{\text{odd}} = \sum_{\text{odd } i} O^{[i,i+1]} + O^{[i]} = \sum_{\text{odd } i} \hat{H}^{[i]}_{\text{odd}} \tag{3.32}$$

This decomposition could be used to transform $e^{-\hat{H}\tau}$ into a form that could be handled by the TEBD algorithm. Note that all operators $\hat{H}_{\text{even}}$ and $\hat{H}_{\text{odd}}$ have dimension d^2, as they work on a maximum of two sites with dimension d. Next we introduce $\Delta\tau = \tau/N$, where $N \gg 1$ is the number of time steps in the simulation. Then

$$e^{-\hat{H}\tau} = \left(e^{-\hat{H}\Delta\tau}\right)^N \tag{3.33}$$

If we choose $\Delta\tau$ sufficiently small, it is allowed to decompose $e^{-\hat{H}\Delta\tau}$ even further, using the second-order *Suzuki-Trotter decomposition*, which has an error of $O(\Delta\tau^3)$ [30]:

$$e^{-\hat{H}\Delta\tau} = e^{-(\hat{H}_{\text{even}}+\hat{H}_{\text{odd}})\Delta\tau} \approx e^{-\hat{H}_{\text{even}}\Delta\tau/2} e^{-\hat{H}_{\text{odd}}\Delta\tau} e^{-\hat{H}_{\text{even}}\Delta\tau/2} = \hat{S}_{\text{even}}\hat{S}_{\text{odd}}\hat{S}_{\text{even}} \tag{3.34}$$

In the last equality, we introduced $\hat{S}_{\text{even}}$ and $\hat{S}_{\text{odd}}$. All $\hat{H}^{[i]}_{\text{even}}$ and $\hat{H}^{[i]}_{\text{odd}}$ are mutually commuting — $\left[\hat{H}^{[i]}_{\text{even}}, \hat{H}^{[j]}_{\text{even}}\right] = \left[\hat{H}^{[i]}_{\text{odd}}, \hat{H}^{[j]}_{\text{odd}}\right] = 0$ — so both $\hat{S}_{\text{even}}$ and $\hat{S}_{\text{odd}}$ be written as products of individual components of the original time evolution operator:

$$\hat{S}_{\text{even}} = e^{-\hat{H}_{\text{even}}\Delta\tau/2} = e^{-\sum_{\text{even } i}\hat{H}^{[i]}_{\text{even}}\Delta\tau/2} = \prod_{\text{even i}} e^{-\hat{H}^{[i]}_{\text{even}}\Delta\tau/2} \tag{3.35}$$

and

$$\hat{S}_{\text{odd}} = e^{-\hat{H}_{\text{odd}}\Delta\tau} = e^{-\sum_{\text{odd } i}\hat{H}^{[i]}_{\text{odd}}\Delta\tau} = \prod_{\text{odd i}} e^{-\hat{H}^{[i]}_{\text{odd}}\Delta\tau} \tag{3.36}$$

The transformations above allow us to write the original time evolution operator $e^{-\hat{H}\Delta\tau}$ as product of local operators $e^{-\hat{H}^{[i]}_{\text{even}}\Delta\tau/2}$ and $e^{-\hat{H}^{[i]}_{\text{odd}}\Delta\tau}$ of dimension d^2. The TEBD algorithm now works as follows:

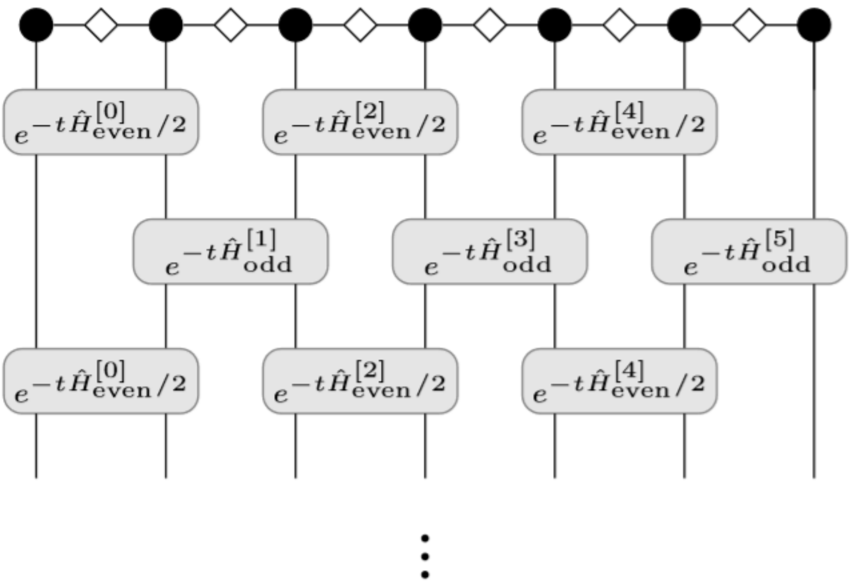
$e^{-t\hat{H}^{[0]}_{\text{even}}/2}$
$e^{-t\hat{H}^{[2]}_{\text{even}}/2}$
$e^{-t\hat{H}^{[4]}_{\text{even}}/2}$
$e^{-t\hat{H}^{[1]}_{\text{odd}}}$
$e^{-t\hat{H}^{[3]}_{\text{odd}}}$
$e^{-t\hat{H}^{[5]}_{\text{odd}}}$
$e^{-t\hat{H}^{[0]}_{\text{even}}/2}$
$e^{-t\hat{H}^{[2]}_{\text{even}}/2}$
$e^{-t\hat{H}^{[4]}_{\text{even}}/2}$

Figure 3.7: Graphical representation of the time evolution over $\Delta\tau$ of a random initial quantum state $|\phi_o\rangle$ using the TEBD algorithm. $|\phi_o\rangle$ is depicted in MPS in canonical form (similar to figure 3.3), the grey blocks represent the two-site elements of $\hat{H}$. (*from* [8]).

1. Create a random initial state $|\phi_0\rangle$ and transform this into MPS form.
2. Write the Hamiltonian of the system in terms of two-site operators $e^{-\hat{H}^{[i]}_{\text{even}}\Delta\tau/2}$ and $e^{-\hat{H}^{[i]}_{\text{odd}}\Delta\tau}$. Write the operators in MPS form as well.
3. Evolve $|\phi_0\rangle$ in imaginary time over $\Delta\tau$ by applying $\hat{S}_{\text{even}}$, $\hat{S}_{\text{odd}}$ and again $\hat{S}_{\text{even}}$. Operators $\hat{S}$ are applied using so called *Trotter sweeps*, where the two-site operators sweep through the one-dimensional chain from left to right. The application of the two-site operators is explained in section 3.1.7.
4. Repeat step 3. until a desired accuracy for ground state $|\psi_g\rangle$ is reached.

The procedure is graphically presented in figure 3.7.

4

Simulation results

The foundation for all simulations in this thesis is laid in the previous chapters. The underlying principles of quantum systems in general are outlined in chapter 2, focusing on one-dimensional quantum chains of electron spins and Josephson junctions. In chapter 3.1 MPS formalism was introduced, with special attention for the entanglement and truncation of matrix product states. This led to the development of the TEBD algorithm, explained in chapter 3.2. Next, the superconducting Josephson junction qubit was introduced in chapter 4.

The simulations in this thesis can be split up in two parts. Firstly in section 4.1 we illustrate the TEBD algorithm and look at its inner workings by applying it to both the Ising model with a transverse field and a one-dimensional chain of Josephson junction. Secondly, we analyze the superconducting qubit, also using TEBD, in order to find its ground state energy. All simulations are done using Python software. All used scripts can be found in appendix A. For TEBD simulations, the used scripts are based on the example that Pollmann offers in his lecture slides [31].

4.1. Application of TEBD

In order to illustrate the TEBD algorithm outlined in chapter 3, we apply it to the Ising model and a one-dimensional chain of Josephson junctions. Specifically, we are interested in the performance of the model and try to find an optimal set of parameters for the analysis of the qubit, in the next section.

4.1.1. Ising model with a transverse field

The Ising model with transverse field that we simulate is governed by the following Hamiltonian:

$$\hat{H}_{\text{sim}} = -J_{\text{sim}} \sum_i \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z - g_{\text{sim}} \sum_i \hat{\sigma}_i^x \tag{4.1}$$

We are interested in the ground state energy, which we can compare to the analytical solution from [1]. The model parameters of this model are spin coupling J_{sim} and field coupling g_{sim}. The simulation parameters are site dimension d — which is 2 for the electron spins in this model — bond dimension χ, imaginary time step $\Delta\tau$ and number of iterations N. We introduced superscripts for the model that we simulate in this thesis, in order to avoid confusion when we compare with the Hamiltonian in [1], which is as follows:

$$\hat{H}_{\text{Pfeuty}} = -J_{\text{Pfeuty}} \sum_i \hat{S}_i^x \hat{S}_{i+1}^x - \Gamma_{\text{Pfeuty}} \sum_i \hat{S}_i^z \tag{4.2}$$

There are two important differences between $\hat{H}_{\text{sim}}$ and $\hat{H}_{\text{Pfeuty}}$. Firstly, Pfeuty interchanges x and z. This has no impact on the ground state energy, as this is only a change in dimensions. Furthermore, Pfeuty uses $\hat{S}_x$ and $\hat{S}_z$ operators, whereas we simulate using $\hat{\sigma}_x$ and $\hat{\sigma}_z$. Remember from section 2.2.2 that the difference between $\hat{S}$ and $\hat{\sigma}$ is a factor $\hbar/2$. We omit constant $\hbar$ as this only effects the energy unit and establish the following relations for correct comparison of the results:

$$\begin{aligned} J_{\text{Pfeuty}} &= 4J_{\text{sim}} \\ \Gamma_{\text{Pfeuty}} &= 2g_{\text{sim}} \end{aligned} \tag{4.3}$$

Table 4.1: Analytic and numerical calculation of ground state energy E_g of infinite-chain Ising model with transverse field for different values of J and g. Numerical calculation is done with the TEBD algorithm with $N = 1000$, $\tau = 0.01$ and $\chi = 5$.

λ	J_{Pfeuty}	Γ_{Pfeuty}	$E_{g,\text{Pfeuty}}$	J_{sim}	g_{sim}	$E_{g,\text{sim}}$	$\lvert E_{g,\text{Pf}} - E_{g,\text{sim}}\rvert / \lvert E_{g,\text{Pf}}\rvert$
0.5	2	2	-1.06354440997	0.5	1	-1.06354575445	1.3×10^{-6}
1	4	2	-1.27323954474	1	1	-1.27292632397	2.5×10^{-4}
1.5	6	2	-1.67192622154	1.5	1	-1.67192780049	9.4×10^{-7}
2	8	2	-2.12708881995	2	1	-2.12708993556	5.2×10^{-7}
2	4	1	-1.06354440997	1	0.5	-1.06354454877	1.3×10^{-7}

Firstly, we want to show that our implementation of the Ising model is correct by comparing the ground state energy with the analytic solution. Pfeuty solves the Ising model for a chain with periodic boundary conditions. The analytic solution for the ground state energy is given by (equation 3.2 in[1]):

$$\frac{-E_g}{\Gamma N} = \frac{2}{\pi}(1+\lambda)E\left(\frac{\pi}{2},\theta\right) \tag{4.4}$$

where $\lambda = J_{\text{Pfeuty}}/2\Gamma_{\text{Pfeuty}} = J_{\text{sim}}/g_{\text{sim}}$ and E is an elliptic integral of the second kind. The script used to calculate the analytic solution can be found in appendix A.1.

The chain with periodic boundary conditions is analogous to an infinite chain. Because of translation invariance, this means that the solution of the ground state should be the same for all individual sites. However, as we deal with two-site operators, we partially break this translational invariance by defining the random initial state $|\phi_0\rangle$ on two sites. Then we perform the TEBD algorithm outlined in section 3.2 for $N = 1000$ time steps of $\tau = 0.01$, applying two-site operators as described in section 3.1.7. The value for χ is chosen to be 5 based on the example by Pollmann [31]. For the Ising model, the individual spins have dimension $d = 2$, and the operators have dimension $d^2 = 4$, so $\chi = 5$ suffices for our purpose now.

As the outcome of every iteration is normalized, $|\phi_0\rangle$ has converged to $e^{-\hat{H}\Delta\tau N}|\psi_g\rangle$ when the procedure ends. Next we take the expectation value of $e^{-\hat{H}\Delta\tau N}$ — the sum of squares of all matrix elements [8] — which yields $e^{-E_g\Delta\tau N}$. From this we can calculate the ground state energy per site by dividing by 2. The script that is used can be found in appendix A.2.

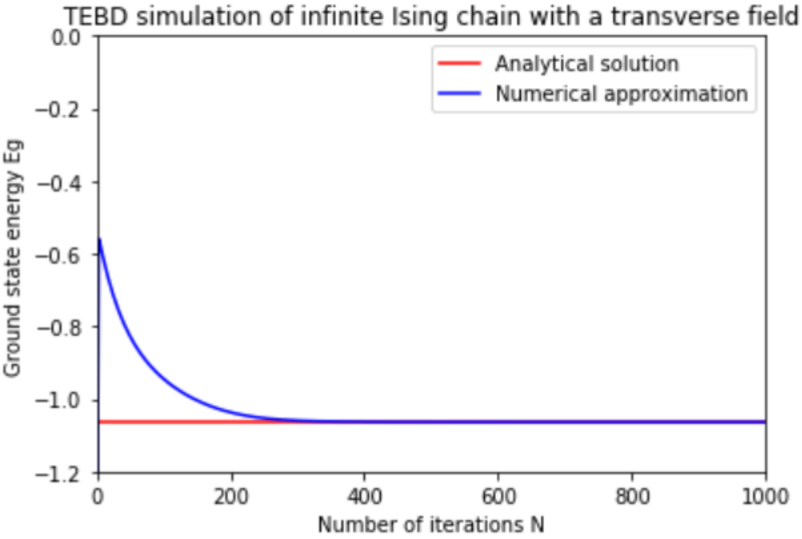

Figure 4.1: Numerical simulation of infinite Ising chain with a transverse field with parameters $J = 1$ and $g = 0.5$ using TEBD algorithm with $\Delta\tau = 0.01$, $N = 1000$ and $\chi = 5$.

From [1], it becomes clear that E_g depends on the ratio λ and the value of Γ_{Pfeuty}, which correlates with the strength of the external field. For our simulations, we compare the analytical result and the simulated result for $\lambda = 0.5$, 1, 1.5 and 2. As the sign of J and g has no effect on the value of the ground state energy, only on the configuration of the ground state, we have not varied the sign of these parameters.

In table 5.1 we compare the analytic solution for the ground state energy with the results of the TEBD simulation for multiple values of λ after $N = 1000$ iterations. For all values of λ, the TEBD gives a good approximation for the analytical solution. Figure 4.1 shows that the TEBD indeed converges to $E_g = -1.06354440997$ for $J = 1$ and $g = 0.5$ and that $N = 1000$ iterations are enough to approximate the ground state energy. In fact, less iterations already suffice to reach comparable accuracy.

In the simulation above, we used the TEBD algorithm to approximate the infinite-chain Ising model using partial translation invariance. In this way, we only need to include two sites in our model. However, the real power of MPS formulation and the TEBD algorithm lies in longer chains, where exponentially growing Hilbert space can be efficiently explored using the truncation of the matrix product states. Therefore, it is interesting to look at the behaviour of longer quantum chains, with open boundary conditions.

For this, we slightly change our script by introducing chain length L and running the sweeps over even and odd sites alternately. Also, we store the energy per site in every iteration, and average over the complete chain to find the average ground state energy per site of the chain. The adjusted script can be found in appendix A.3.

Table 4.2: Numerical calculation of ground state energy E_g of finite-chain Ising model with transverse field for different values of L. Numerical calculation is done with the TEBD algorithm with $J = 1$, $g = 0.5$, $N = 1000$, $\tau = 0.01$ and $\chi = 5$.

L	$E_{g,\text{sim}}$ at $N = 1000$	Relative error to analytical solution
3	-1.11696246824	5.0×10^{-2}
10	-1.06389273293	3.3×10^{-4}
50	-1.0635935535	4.6×10^{-5}
100	-1.06356853454	2.3×10^{-5}

We run the simulation for $L = 3$, 10 and 50. The results can be found in table 5.2 and figure 4.2. Some things stand out from these plots. Firstly, the simulation for $L = 3$ does not converge to the analytical value for the infinite Ising chain. The reason for this is the fact that we simulate with open boundary conditions. As the chain is only one site larger than the two-site chain that we used in the previous simulation, the open boundary conditions clearly have their impact on the ground state energy. The sites at the boundaries are less restricted than before, which allows them to take on lower-energy configurations in the ground state. A detailed explanation of this effect is given by [32]. As the chain length grows to $L = 10$ or more, these advantages become relatively smaller, allowing the ground state energy to approximate the ground state energy for the infinite chain. However, the effects will always have a slight advantage at the boundaries, meaning that the ground state energy for the finite chain with open boundaries is always lower than the infinite result. We can see this from table 5.2, as all simulated ground state energies are lower than the analytical value of $E_g = -1.06354440997$.

Next to that, we see that the simulation converges at roughly the same rate as the infinite chain, for $L = 10$ and $L = 50$. This indicates the linear character of the TEBD algorithm, as increasing L only increases the computation time per iteration, but not the overall converging speed. As we pointed out in chapter 2, methods such as Lanczos diagonalization can handle quantum chains up to roughly 22 sites, and at longer chain lengths, the TEBD algorithm really shows its power.

4.1.2. One-dimensional chain of Josephson junctions

We extend our knowledge of Josephson junctions and TEBD by applying the algorithm to a one-dimensional chain of Josephson junctions. As we have seen in chapter 2, a chain of Josephson junctions with fixed boundary condition is governed by the following Hamiltonian:

$$\hat{H} = \sum_{i=1}^{L} \left[-E_C \frac{d^2}{d\phi_i^2} + 1 - \cos\left(\phi_i - \phi_{i+1} + \Delta\theta / L\right) \right] \tag{4.5}$$

In order to analyze this Hamiltonian numerically, we need to discretize ϕ_i. We discretize ϕ_i uniformly into d states $\{|\phi_j\rangle_i\}_{j=1}^{d}$, where the difference between the states is $\Delta\phi$. Here, i is the index for the position of the junction, and index j represents the discrete state of a junction. For an accurate description of the system, we will need a dimension d that is much higher than $d = 2$, which we used for the analysis of the Ising model. The first part of the Hamiltonian can now be discretized as follows (location index i is left out for simplicity):

$$-E_C \frac{d^2}{d\phi^2} \approx \frac{-E_C}{(\Delta\phi)^2} \left(\sum_j |\phi_{j+1}\rangle\langle\phi_j| - 2|\phi_j\rangle\langle\phi_j| + |\phi_{j-1}\rangle \right) \tag{4.6}$$

The second part of the Hamiltonian is built up straightforwardly, which can be seen in the script in appendix A.4.

Again, we are first interested in the accuracy of our model. Although we could not compare our results to an analytical solution, Melo [8] shows graphs containing the ground state energy of Josephson junction chains with varying values for length L, charging energy E_C and boundary condition $\Delta\theta$. The data underlying these graphs is not available, but reconstruction of the same graphs gives at least an indication of the accuracy of our model.

We reconstruct the simulations for a Josephson junction chain of length $L = 4$, with $E_C = 0.1$, $E_J = 1$, $d = 10$ and $\chi = 10$ and $N = 500$. The total dimension of this chain is 10^4, which is significantly higher than that of most of the Ising model simulations in the previous section. $\chi = 10$ might seem to be on the low side, but for a first impression of the performance of the model, it suffices. The same holds for the choice of N. We vary $\Delta\theta$ from $\pi/10$ to π in order to reconstruct the graph from [8]. The results of our simulation can be found in figure 4.3. In this figure, we see that the results from our simulation roughly compare to that of Melo, except at small $\Delta\theta$.

4.1.3. Chain of Josephson junctions with periodic boundary conditions

To be filled after discussion with Jos

4.2. Josephson junction qubit

To be filled after discussion with Jos

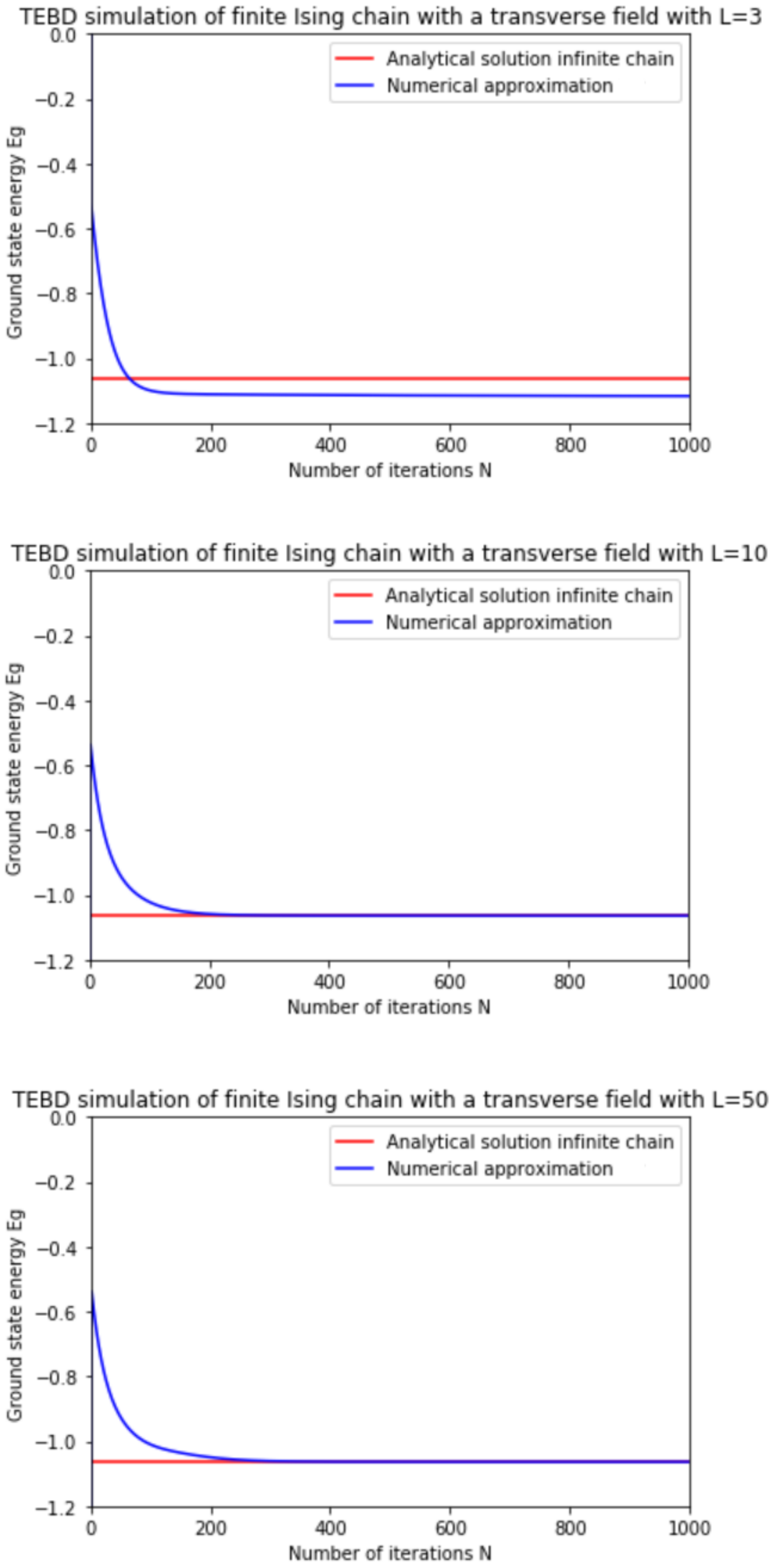

Figure 4.2: Numerical simulation of finite Ising chain with a transverse field of lengths $L = 3$, $L = 10$ and $L = 50$ using TEBD algorithm. Parameters are $J = 1$ and $g = 0.5$ with $\Delta\tau = 0.01$, $N = 1000$ and $\chi = 5$. The plot for $L = 100$ is almost identical to the $L = 50$ plot and is left out for clarity.

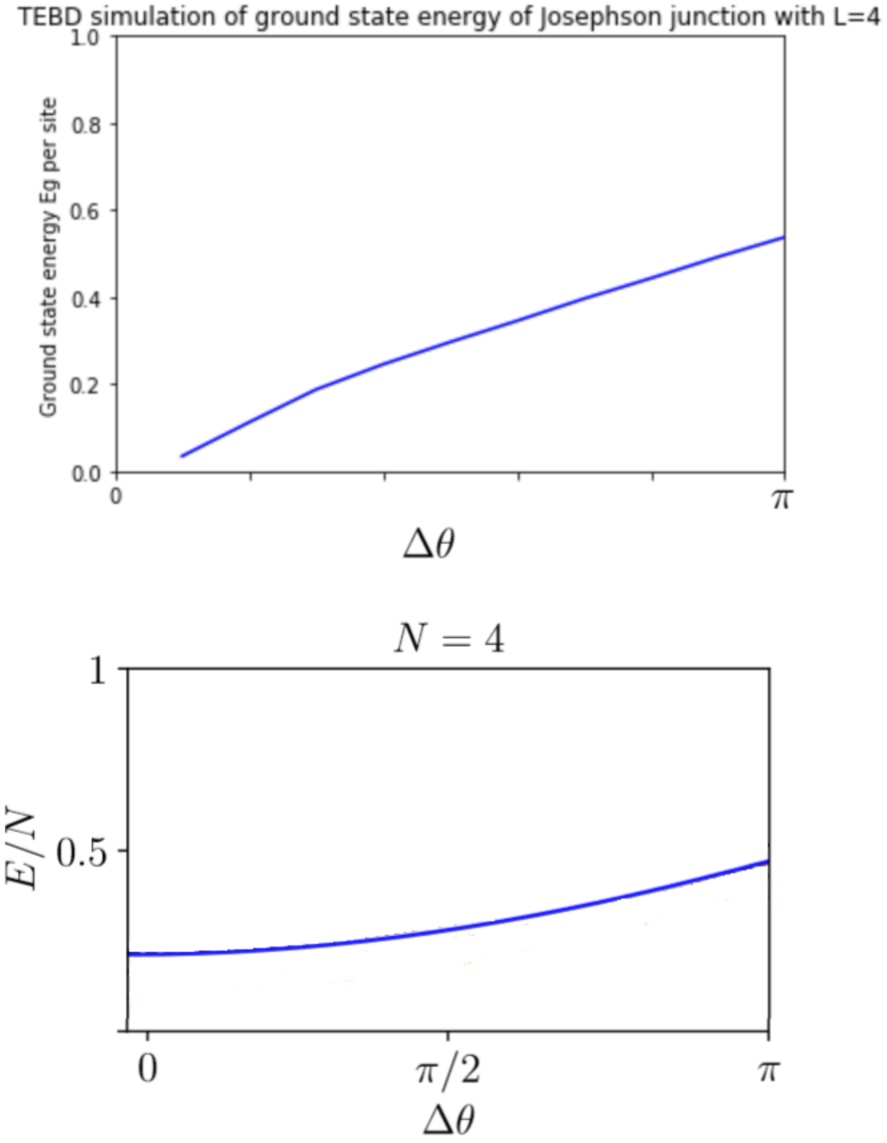

Figure 4.3: Top: Numerical simulation of Josephson junction chain of length $L = 4$ using TEBD algorithm. Parameters are $E_C = 0.1$, $E_J = 1$, $d = 10$ and $\chi = 10$ and $N = 500$. Bottom: plot from [8] included to compare performance of numerical simulation for Josephson junction chains.

5
Conclusion and discussion

Simulations on the Heisenberg model have shown that TEBD is indeed a very efficient way to perform ground state calculations on low-entanglement quantum systems. We have been able to implement example schemes of the TEBD algorithm and adjust them accordingly, for example to include open boundary conditions. Results were in line with the analytical solution offered by Pfeuty [1].

The extension to Josephson junctions chains confronted us with quantum systems that have Hilbert spaces of even higher dimensions. Although we have successfully implemented a scheme to analyze these chains using TEBD, the limitations on laptop computing power eventually limited us to perform further analysis on these systems. Interesting questions that remain involve the optimal truncation of χ and the required dimension for the discretization of $|\phi_j\rangle$, both ensuring optimal performance of the algorithm. Melo does offer answers to these questions, stating that TEBD with $\chi = 20$ and $d = 20$ in cases with few junctions offers remarkable results, although the method does not fully capture the correct behaviour of larger systems [8].

In chapter 2, we gave a detailed overview of the DMRG algorithm, as it was this method that led to the development of MPS and eventually TEBD. However, the direct comparison of 'classic' DMRG, MPS-style DMRG and other MPS-based methods such as TEBD in terms of performance of solving the same problems could be very interesting. Under what circumstances is any of the methods preferred over the others? In this thesis, we have been primarily focussed on the computation of ground-state energies, but there might be other problems of interest that require other methods.

In our comparison of the simulation results with the analytical solution by Pfeuty, we did not dive deep into the background of his solution, although this could reveal a lot about the structure of the ground state. Further study would be required to add these insights. The same could be said about an analytical solution for the Josephson junction chain. Further study is required to find an analytical solution to this problem, which in turn could be used to benchmark and optimize the methods used.

In the simulations in this thesis, we have only encountered situations in which the algorithms used behaved stable. However, to ensure that TEBD is stable in any case, for any system and any Hamiltonian, a stability study should be performed.

Matrix product states arise naturally from physics, as it incorporates the system property entanglement. Its beneficial truncation and operator properties raise the question whether the method could be explored outside physics as well, offering efficiency costs in other field within science or engineering.

The field of low-entanglement states has received an increasing amount of attention over the past years. Next to matrix product states, the more general *Tensor network states* are another area of research interest. Projected entangled pair states (PEPS) have another advantage over MPS, namely that it is not restricted to sequential entanglement, but can handle arbitrary networks of entanglement instead, making it more suitable for other applications [33].

A

Python scripts

A.1. Calculating the analytic solution of the Ising model from Pfeuty [1]

```
import numpy as np
import math
import matplotlib.pyplot as plt
from scipy.linalg import expm, sinm, cosm
from scipy.special import ellipeinc

Jpollmann=1;
gpollmann=0.5;
J=4*Jpollmann;
g=2*gpollmann;
l=J/2/g;
N=100;
k=np.zeros(N);

for m in range (0,N):
k[m]=2*math.pi*m/N        #use periodicity here: cos(k)=cos(2+2*pi)

lambda_sq=np.zeros(N);
for i in range(0,N):
lambda_sq[i]=1+l**2+2*l*math.cos(k[i])

print(lambda_sq)

lambda_fin=lambda_sq**0.5

#print(sum(lambda_fin))

E_gr=-g/2*sum(lambda_fin)/N #(2.13) in Pfeutty

print(E_gr)

theta=4*l/(1+l)**2
print(2*(1+l)/math.pi*ellipeinc(math.pi/2,theta)*-0.5*g) #(3.2) in Pfeutty
```

A.2. Simulation of translational invariant Ising model ground state

```
import numpy as np
import matplotlib.pyplot as plt
import math
import scipy.linalg
from scipy.linalg import expm, sinm, cosm

#model parameters
J=1.0; g=0.5; chi=5; d=2; delta=0.01; N=1000;
G = np.random.rand(2,d,chi,chi); l=np.random.rand(2,chi)

#Generate the two-site time evolution operator
H = np.array([[J,-g/2,-g/2,0], [-g/2,-J,0,-g/2], [-g/2,0,-J,-g/2], [0,-g/2,-g/2,J]])
U = np.reshape(expm(-delta*H),(2,2,2,2))
E = np.empty(N);
Ean = np.empty(N);
Ean.fill(-1.06354440997)

#Perform the imaginary time evolution alternating on A and B bonds
for step in range(0, N):
A = np.mod(step,2); B = np.mod(step+1,2)

#Construct theta 1
theta = np.tensordot(np.diag(l[B,:]),G[A,:,:,:],axes=(1,1))

#Construct theta 2
theta = np.tensordot(theta,np.diag(l[A,:],0),axes=(2,0))

#Construct theta 3
theta = np.tensordot(theta,G[B,:,:,:],axes=(2,1))
#print(theta)
#print(theta.shape)

#Construct theta 4
theta = np.tensordot(theta,np.diag(l[B,:],0),axes=(3,0))

#Apply U
theta = np.tensordot(theta,U,axes=([1,2],[0,1]))

#SVD
theta = np.reshape(np.transpose(theta,(2,0,3,1)),(d*chi,d*chi))
X, Y, Z = np.linalg.svd(theta); Z = Z.T

#Truncate
l[A,0:chi]=Y[0:chi]/np.sqrt(sum(Y[0:chi]**2)) #normalize

X=np.reshape(X[0:d*chi,0:chi],(d,chi,chi))
G[A,:,:,:]=np.transpose(np.tensordot(np.diag(l[B,:]**(-1)),X,axes=(1,1)),(1,0,2))

Z=np.transpose(np.reshape(Z[0:d*chi,0:chi],(d,chi,chi)),(0,2,1))
G[B,:,:,:]=np.tensordot(Z,np.diag(l[B,:]**(-1)),axes=(2,0))
E[step]=-np.log(np.sum(theta**2))/delta/2

print ("E_iTEBD␣=", -np.log(np.sum(theta**2))/delta/2)
```

A.3. Simulation of finite size Ising model ground state

```
import numpy as np
import matplotlib.pyplot as plt
import math
import scipy.linalg
from scipy.linalg import expm, sinm, cosm

#model parameters
J=1.0; g=0.5; chi=5; d=2; delta=0.01; N=1000; L=10;
G = np.random.rand(L,d,chi,chi); l=np.random.rand(L,chi)

#Generate the two-site time evolution operator
H = np.array([[J,-g/2,-g/2,0], [-g/2,-J,0,-g/2], [-g/2,0,-J,-g/2], [0,-g/2,-g/2,J]])
U = np.reshape(expm(-delta*H),(2,2,2,2))
Esite = np.empty(L-2); #store energy per site
Estep = np.empty(N); # store average energy of the chain per iteration
Ean = np.empty(N);
Ean.fill(-1.06354440997)

#Perform the imaginary time evolution on a finite chain with open ends
for step in range(0, N):
for site in range(0,L-2,2): #even sites
A = site; B = site+1; C = site+2;

#Construct theta 1
theta = np.tensordot(np.diag(l[A,:]),G[A,:,:,:],axes=(1,1))

#Construct theta 2
theta = np.tensordot(theta,np.diag(l[B,:],0),axes=(2,0))

#Construct theta 3
theta = np.tensordot(theta,G[B,:,:,:],axes=(2,1))

#Construct theta 4
theta = np.tensordot(theta,np.diag(l[C,:],0),axes=(3,0))

#Apply U
theta = np.tensordot(theta,U,axes=([1,2],[0,1]))

#SVD
theta = np.reshape(np.transpose(theta,(2,0,3,1)),(d*chi,d*chi))
X, Y, Z = np.linalg.svd(theta);Z = Z.T

#Truncate
l[B,0:chi]=Y[0:chi]/np.sqrt(sum(Y[0:chi]**2))

X=np.reshape(X[0:d*chi,0:chi],(d,chi,chi))
G[A,:,:,:]=np.transpose(np.tensordot(np.diag(l[A,:]**(-1)),X,axes=(1,1)),(1,0,2))

Z=np.transpose(np.reshape(Z[0:d*chi,0:chi],(d,chi,chi)),(0,2,1))
G[B,:,:,:]=np.tensordot(Z,np.diag(l[C,:]**(-1)),axes=(2,0))

Esite[site]=-np.log(np.sum(theta**2))/delta/2 #store energy per site for this iteration

for site in range(1,L-2,2): #odd sites
```

```
A = site; B = site+1; C = site+2;

#Construct theta 1
theta = np.tensordot(np.diag(l[A,:]),G[A,:,:,:],axes=(1,1))

#Construct theta 2
theta = np.tensordot(theta,np.diag(l[B,:],0),axes=(2,0))

#Construct theta 3
theta = np.tensordot(theta,G[B,:,:,:],axes=(2,1))

#Construct theta 4
theta = np.tensordot(theta,np.diag(l[C,:],0),axes=(3,0))

#Apply U
theta = np.tensordot(theta,U,axes=([1,2],[0,1]))

#SVD
theta = np.reshape(np.transpose(theta,(2,0,3,1)),(d*chi,d*chi))
X, Y, Z = np.linalg.svd(theta);Z = Z.T

#Truncate
l[B,0:chi]=Y[0:chi]/np.sqrt(sum(Y[0:chi]**2))

X=np.reshape(X[0:d*chi,0:chi],(d,chi,chi))
G[A,:,:,:]=np.transpose(np.tensordot(np.diag(l[A,:]**(-1)),X,axes=(1,1)),(1,0,2))

Z=np.transpose(np.reshape(Z[0:d*chi,0:chi],(d,chi,chi)),(0,2,1))
G[B,:,:,:]=np.tensordot(Z,np.diag(l[C,:]**(-1)),axes=(2,0))

Esite[site]=-np.log(np.sum(theta**2))/delta/2 #store energy per site for this iteration

Estep[step]=np.average(Esite) #store average energy in the chain for this iteration

print("E_iTEBD␣=",Estep[step]) #approximate average ground state energy after N iterations
print("E_analytical=",Ean[1]) #analytical solution to the infinite Ising chain
print("relative_error=",abs((Estep[step]-Ean[1])/Ean[1]))
```

A.4. Simulation of chain of Josephson junctions

```
import numpy as np
import matplotlib.pyplot as plt
import math
import scipy.linalg
from scipy.linalg import expm, sinm, cosm

#model parameters
Ec=0.1;
Ej=1.0;
chi=10; d=10; delta=0.01; N=1000; L=4; dtheta=math.pi;
dphi=2*math.pi/d;
G = np.random.rand(L,d,chi,chi); l=np.random.rand(L,chi);
Esite = np.empty(L); #store energy per site
Estep = np.empty(N); # store average energy of the chain per iteration

#Generate the one-site second derivative operator
diag=np.eye(d,k=1)+np.eye(d,k=-1)-2*np.eye(d,k=0)
diag[0,d-1]=1
diag[d-1,0]=1
print(diag)

#create Hphase two-site operator
Hphase=np.kron(diag,np.eye(d))
Hphase=-Ec/(dphi**2)*Hphase

#create Hcharge
Hcharge=np.zeros((d,d))
step = 1
while step<d/2:
Hcharge=Hcharge + step*np.eye(d,k=step)+step*np.eye(d,k=d-step)+step*np.eye(d,k=-step)+step*np.eye(d,k
step+=1
if d%2==0:
Hcharge=Hcharge + step*np.eye(d,k=int(step)) + step*np.eye(d,k=-int(step))

#reshape Hcharge
Hcharge=2*np.pi/d*Hcharge*+np.ones((d,d))*dtheta/(L-1)
Hcharge=np.ones((d,d))-np.cos(Hcharge)
Hcharge=Ej*np.diag(np.reshape(Hcharge,d**2))

#create H
H=Hphase+Hcharge

#create U
U = np.reshape(expm(-delta*H),(d,d,d,d))

#Perform the imaginary time evolution alternating on A and B bonds
for step in range(0, N):

for site in range(0,L,2):
A = site; B = (site+1)%L; C = (site+2)%L;

#Construct theta 1
theta = np.tensordot(np.diag(l[A,:]),G[A,:,:,:],axes=(1,1))
```

```
#Construct theta 2
theta = np.tensordot(theta,np.diag(l[B,:],0),axes=(2,0))

#Construct theta 3
theta = np.tensordot(theta,G[B,:,:,:],axes=(2,1))

#Construct theta 4
theta = np.tensordot(theta,np.diag(l[C,:],0),axes=(3,0))

#Apply U
theta = np.tensordot(theta,U,axes=([1,2],[0,1]))

#SVD
theta = np.reshape(np.transpose(theta,(2,0,3,1)),(d*chi,d*chi))
X, Y, Z = np.linalg.svd(theta);Z = Z.T

#Truncate
l[B,0:chi]=Y[0:chi]/np.sqrt(sum(Y[0:chi]**2))

X=np.reshape(X[0:d*chi,0:chi],(d,chi,chi))
G[A,:,:,:]=np.transpose(np.tensordot(np.diag(l[A,:]**(-1)),X,axes=(1,1)),(1,0,2))

Z=np.transpose(np.reshape(Z[0:d*chi,0:chi],(d,chi,chi)),(0,2,1))
G[B,:,:,:]=np.tensordot(Z,np.diag(l[C,:]**(-1)),axes=(2,0))

for site in range(1,L,2):
A = site; B = (site+1)%L; C = (site+2)%L;

#Construct theta 1
theta = np.tensordot(np.diag(l[A,:]),G[A,:,:,:],axes=(1,1))

#Construct theta 2
theta = np.tensordot(theta,np.diag(l[B,:],0),axes=(2,0))

#Construct theta 3
theta = np.tensordot(theta,G[B,:,:,:],axes=(2,1))

#Construct theta 4
theta = np.tensordot(theta,np.diag(l[C,:],0),axes=(3,0))

#Apply U
theta = np.tensordot(theta,U,axes=([1,2],[0,1]))

#SVD
theta = np.reshape(np.transpose(theta,(2,0,3,1)),(d*chi,d*chi))
X, Y, Z = np.linalg.svd(theta);Z = Z.T

#Truncate
l[B,0:chi]=Y[0:chi]/np.sqrt(sum(Y[0:chi]**2))

X=np.reshape(X[0:d*chi,0:chi],(d,chi,chi))
G[A,:,:,:]=np.transpose(np.tensordot(np.diag(l[A,:]**(-1)),X,axes=(1,1)),(1,0,2))

Z=np.transpose(np.reshape(Z[0:d*chi,0:chi],(d,chi,chi)),(0,2,1))
G[B,:,:,:]=np.tensordot(Z,np.diag(l[C,:]**(-1)),axes=(2,0))
```

```
Estep[step]=-np.log(np.sum(theta**2))/delta #store energy per junction at this iteration

print("E_iTEBD =", -np.log(np.sum(theta**2))/delta)
```

Bibliography

[1] Pierre Pfeuty. The one-dimensional ising model with a transverse field. *Annals of Physics*, 57(1):79 – 90, 1970.

[2] Tom Simonite. Intel puts the brakes on moore's law, 2016.

[3] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[4] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt, and Christian F. Roos. Quantum chemistry calculations on a trapped-ion quantum simulator. *Phys. Rev. X*, 8:031022, Jul 2018.

[5] Networked Quantum Information Technologies. Applications of quantum computing, 2018.

[6] Luca Chirolli and Guido Burkard. Decoherence in solid-state qubits. *Advances in Physics*, 57(3):225–285, 2008.

[7] B.D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251 – 253, 1962.

[8] Andre Melo. Numerical study of a superconducting qubit for the realization of quantum ising chains using matrix product state techniques. Master's thesis, TU Delft.

[9] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992.

[10] Garnet Kin-Lic Chan. Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *The Journal of Chemical Physics*, 145, Jul 2016.

[11] R. L. Jaffe. Supplementary lecture notes on dirac notation, quantum states, etc., 1996.

[12] N. Dorey. Lecture notes on quantum mechanics, 2007.

[13] Jos Thijssen. *Computational Physics*. Cambridge University Press, Delft, The Netherlands, 2007.

[14] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[15] Reinhard M. Noack and Salvatore R. Manmana. Diagonalization- and numerical renormalization-group-based methods for interacting quantum systems. *AIP Conference Proceedings*, 789:93, 2005.

[16] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[17] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand. B*, 45:255–282, 1950.

[18] Adrian E. Feiguin. *The Density Matrix Renormalization Group*, pages 31–65. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[19] John Preskill. Lecture notes on quantum computation, 2015.

[20] Artur Ekert and Peter L. Knight. Entangled quantum systems and the schmidt decomposition. *American Journal of Physics*, 63(5):415–423, 1995.

[21] Ulrich Schöllwock. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326:96–192, 2010.

[22] Michael Tinkham. *Introduction to Superconductivity: Second Edition* . Dover Publications, second edition edition, June 2004.

[23] Leon N. Cooper. Bound electron pairs in a degenerate fermi gas. *Phys. Rev.*, 104:1189–1190, Nov 1956.

[24] Serge Haroche. Lecture 5: An introduction to circuit qed describing josephson junctions as qubits and lc circuits as quantum oscillators. URL: `https://www.quantumlah.org/media/lectures/QT5201E-Haroche-Slides-1.pdf`.

[25] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73:094423, Mar 2006.

[26] Don N. Page. Average entropy of a subsystem. *Phys. Rev. Lett.*, 71:1291–1294, Aug 1993.

[27] J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82:277–306, Feb 2010.

[28] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev. Entanglement in quantum critical phenomena. *Phys. Rev. Lett.*, 90:227902, Jun 2003.

[29] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93:040502, Jul 2004.

[30] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Physics Letters A*, 146(6):319 – 323, 1990.

[31] Frank Pollman. Introduction to matrix product states and algorithms. URL: `http://quantumtensor.pks.mpg.de/wp-content/uploads/2016/06/slides_1.pdf`.

[32] G. G. Cabrera and R. Jullien. Role of boundary conditions in the finite-size ising model. *Phys. Rev. B*, 35:7062–7072, May 1987.

[33] Garnet Kin-Lic Chan. Low entanglement wavefunctions. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2(6):907–920.