

Universiteit Utrecht

MSC THEORETICAL PHYSICS

MASTER THESIS

Quantum Simulations and Many Body Localization

Author:
Philip Verduyn Lunel

Supervisors:
Prof. dr. C.J.M. Schoutens
dr. M. Walter
Prof. dr. C. Morais Smith

Examination date:
July 3, 2019

Universiteit Utrecht



Universiteit Utrecht

QuSoft

QuSoft
Research Center for Quantum Software

Abstract

In this thesis we investigated Anderson Localization and Many Body Localization [1] in two fermionic Hamiltonian models: the Heisenberg XXZ-model and the $\mathcal{N} = 2$ supersymmetric Hamiltonian [2]. Our main question is whether we can find a MBL phase in the supersymmetric Hamiltonian.

In the first part of the thesis the implementation of the time evolution procedure under a Hamiltonian has been implemented in the framework of ProjectQ, a quantum computer simulator. For this the Suzuki-Trotter approximation was used and its performance was compared to direct calculations. We implemented a quantum algorithm that estimates the second Renyi entanglement entropy of a state. We have implemented an algorithm based on the Swap test and an improved algorithm using Quantum Amplitude Estimation. The performance of both methods was analyzed.

We have simulated the entanglement entropy through time and we have reproduced previous results that show indication of Anderson Localization and MBL within the XXZ model [3]. We have imposed randomness in the supersymmetric Hamiltonian and found strong indications of an MBL phase in the supersymmetric Hamiltonian.

Contents

1	Introduction	3
2	Quantum Computations and Algorithms	5
2.1	Preliminaries Quantum Computations	5
2.2	Preliminaries ProjectQ	8
2.3	Quantum Fourier Transform	9
2.4	Quantum Phase Estimation	13
2.5	Quantum Amplitude Estimation	16
3	Quantum Simulations	18
3.1	Hamiltonian Simulation	19
3.1.1	Pauli basis time evolution	20
3.1.2	Trotter-Suzuki Decomposition	22
3.1.3	Jordan-Wigner Transformation	24
3.1.4	Hamiltonian Simulation in ProjectQ	27
4	Entanglement Entropy	31
4.1	Renyi Entropy and Entanglement Entropy	32
4.2	Swap Test	35
4.3	Estimation using QAE	39
4.3.1	Quantum circuit of operator O	42
4.4	Entanglement Entropy Simulation in ProjectQ	45
4.4.1	Swap Test	45
4.4.2	Quantum Amplitude Estimation	47
5	Numerical Simulations of Localization	49
5.1	XXZ Model	49
5.2	Supersymmetric Hamiltonian	51

5.3	Numerical Simulations	52
5.4	Eigenstate Thermalization Hypothesis	56
5.5	Anderson and Many Body Localization	58
5.6	Localization Supersymmetric Hamiltonian	61
6	Conclusion	66
	References	68
	Appendices	72
A	Supersymmetric Hamiltonian	73
	A.1	73
	A.2	74
	A.3	76
	A.4	77

Chapter 1

Introduction

Classical computers perform local operations on single states. Quantum systems can be in a superposition of states, and the system may be entangled. This makes it inherently difficult for classical computers to simulate quantum physics. In 1982 Richard Feynman proposed the first idea of a quantum computer, namely to simulate physics with another physical system. He, however, did not know how the application of such an idea would take place. With recent technical breakthroughs we now do know what these systems might look like. And small scale quantum computers with programmable qubits now actually exist.

In this thesis we will use the programming framework of ProjectQ to simulate quantum computers on a classical computer. A major application of quantum computers is the simulation of physics. Although we have a tremendous understanding of how the laws of physics work for the interactions of particles. Applying these laws and simulating the evolution of the system becomes exponentially hard in the of particles for a classical computer. We introduce the theory of how we can map Hamiltonians to quantum computers and how we can simulate the evolution of a state under such a Hamiltonian.

We will introduce two Hamiltonian systems which we will examine. The Heisenberg XXZ model and the $\mathcal{N} = 2$ supersymmetric Hamiltonian [2]. Of the XXZ model it is known that for strong disorder in the on site potential the system can become fully localized in a phase known as Anderson Localization. Introducing particle interactions we leave this phase, but reach a phase called Many Body Localization. This phase has been of great interest in recent years. For one, because it is the only known robust system that does not thermalize.

Secondly because it remains a perfect insulator even with local interactions.

We recreate these known results in the framework of ProjectQ, and we find indications of Many Body Localized behavior in the supersymmetric model after we introduce randomness.

Chapter 2

Quantum Computations and Algorithms

In this chapter we will introduce the basic notions of quantum computations a number of well-known Quantum Algorithms that will be use full in the remainder of this thesis. They build further on the preliminaries introduced before.

In this section we will introduce two algorithms that play an important role in the application of quantum simulations. The first is the Quantum Fourier Transformation (QFT), the quantum analog of the Fourier transform. The second is the Quantum Phase Estimation (QAE) algorithm, for a given eigenstate $|\psi\rangle$ of some unitary operator U it estimates the corresponding eigenvalue $U|\psi\rangle = \lambda|\psi\rangle$ with arbitrarily high precision. As we will see in the upcoming sections the phase estimation algorithm will be of great use, because many problems can be reduced to phase estimation.

2.1 Preliminaries Quantum Computations

Quantum computers consist of quantum bits or qubits, that can be described as vectors in $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{C}^2$ on the unit circle, i.e. $|\alpha|^2 + |\beta|^2 = 1$. Defining the Dirac notation for the basis states

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (2.1)$$

so a general state is written as $\alpha |0\rangle + \beta |1\rangle$. Systems with more qubits are tensor products of the single qubits for example with two qubits we get

$$\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\alpha_2 \\ \beta_1\beta_2 \end{pmatrix} = \alpha_1\alpha_2 |00\rangle + \alpha_1\beta_2 |01\rangle + \beta_1\alpha_2 |10\rangle + \beta_1\beta_2 |11\rangle. \quad (2.2)$$

A general state of n qubits is written as

$$|\psi\rangle = \bigotimes_{i=1}^n (\alpha_i |0\rangle + \beta_i |1\rangle) = \sum_{i=0}^{N-1} c_i |i\rangle, \quad (2.3)$$

where $|i\rangle = |i_1\rangle \dots |i_n\rangle$ where $i_1 \dots i_n$ is the binary representation of i . A system with n qubits is an element of \mathbb{C}^{2^n} . This exponential growth of the Hilbert space is crucial for quantum computers and shows why it is impossible for computers to simulate them for larger amount of qubits.

The operators we can do on such states must leave the probabilities (i.e. the 1-norm) preserved. Operators U that this must be unitary i.e. $UU^\dagger = I$. In fact any unitary is a quantum gate, and every unitary can be applied on a quantum computer. Just like the local operations from a classical computer (AND, OR, NOT, etc.) we also want our quantum computer to be build from a small set of gates. The gate set used in this work are the Pauli matrices.

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.4)$$

These have the following operations on the basis states

$$X |0\rangle = |1\rangle \quad Y |0\rangle = i |1\rangle \quad Z |0\rangle = |0\rangle \quad (2.5)$$

$$X |1\rangle = |0\rangle \quad Y |1\rangle = -i |0\rangle \quad Z |1\rangle = -|1\rangle. \quad (2.6)$$

These gates are single qubit gates, we also have a 2-qubit gate. In the general description of gates we only need the CNOT gate this gate applies a controlled X -gate, i.e. we apply an X -gate on the second qubit if the first qubit is $|1\rangle$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (2.7)$$

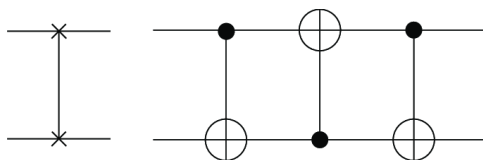


Figure 2.1: Application of 3 CNOT gates is the same as the Swap gate.

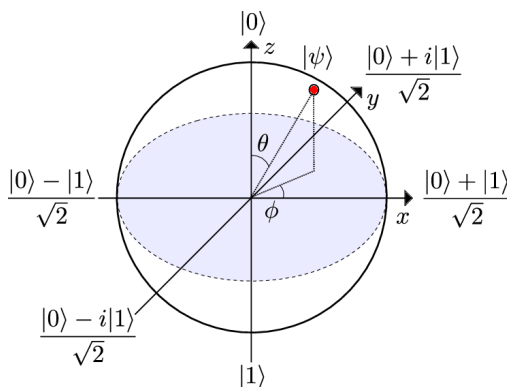


Figure 2.2: Image of the Bloch sphere, the axes are the rotation axes of the Pauli gates.

this maps the 2-qubit states as follows

$$\text{CNOT } |00\rangle = |00\rangle \qquad \text{CNOT } |10\rangle = |11\rangle \qquad (2.8)$$

$$\text{CNOT } |01\rangle = |10\rangle \qquad \text{CNOT } |11\rangle = |01\rangle, \qquad (2.9)$$

and in fact we can control any unitary on a qubit, i.e. $|c\rangle |t\rangle \xrightarrow{C-U} |c\rangle U^c |t\rangle$. A Swap gate swaps two qubits

$$\text{Swap } |a\rangle |b\rangle = |b\rangle |a\rangle. \qquad (2.10)$$

The Swap can be decomposed to three CNOT operations as shown in figure 2.1.

Consider the Bloch sphere in figure 2.2 in which all single qubit states represented. The axes in the Bloch sphere are the axes the rotation operators of the Pauli gates rotate around.

$$R_x(\theta) \equiv e^{-iX\theta/2} = 1 - iX\theta/2 + \frac{(iX\theta/2)^2}{2!} - \frac{(iX\theta/2)^3}{3!} + \frac{(iX\theta/2)^4}{4!} + \dots \quad (2.11)$$

$$= 1 - iX(\theta/2) - (\theta/2)^2 I + iX(\theta/2)^3 + (\theta/2)^4 I + \dots \quad (2.12)$$

$$= (1 - (\theta/2)^2 + (\theta/2)^4 + \dots) I - i(\theta/2 - (\theta/2)^3 + \dots) X \quad (2.13)$$

$$= \cos(\theta/2)I - i \sin(\theta/2)X, \quad (2.14)$$

thus we see a rotation around an axis in the Bloch sphere is a linear combination of Pauli matrices.

The fact that we can make every state combined with the CNOT operator, actually allows us to build any n qubit unitary operator [4]. This theorem is known as *strong universality*.

2.2 Preliminaries ProjectQ

Our quantum algorithms are implemented in the ProjectQ framework. ProjectQ is an open-source quantum simulator in Python developed at the ETH Zürich that uses an intuitive syntax. ProjectQ translates all the commands to the back-end, this can be a real quantum computer or their own quantum simulator.

To use the soft one initializes an engine, this is in our case a simulator. Next you initialize the amount of qubits you want, these are initialized in the $|0\rangle$ state

```
1 from projectq import MainEngine
2 eng = MainEngine()
3 q = eng.allocate_quireg(3)
```

Its syntax is similar to how one would write down the operation himself, e.g. if we apply a Hadamard gate

```
1 from projectq.ops import H
2 H | q[0]
```

A Hadamard gate brings basis states into superposition $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. So if one would measure the qubit in the computational basis we would get $|0\rangle$ or $|1\rangle$ with 50 % probability. Before measuring one should always perform the `flush()` command. This is because ProjectQ waits before applying the operations because it sometimes can optimize the calculations, when you `flush()` all operations that it still needs to do are being done/

```

1 Measure | q[0]
2 print(f"Measured {int(q[0])}")

```

in our case we measured a 0. A CNOT operator is implemented as follows

```

1 from projectq.ops import CNOT
2 CNOT | (q[0], q[1])

```

where $q[0]$ is the control qubit and $q[1]$ the target qubit. As we've mentioned before we can write any unitary with just the Pauli gates and CNOT operator. This means we can simulate any unitary in ProjectQ.

2.3 Quantum Fourier Transform

In mathematics an important tool is the Fourier transformation. It appears in a wide range of problems from signal processing to the diagonalization of a Hamiltonian in physics. It turns out that it is possible to do the quantum analog of a discrete Fourier transformations on a quantum computer. In mathematics the *discrete Fourier transform* is defined on a vector of complex inputs (x_0, \dots, x_{N-1}) of length N that outputs a vector (y_1, \dots, y_k) , where

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N}. \quad (2.15)$$

The *quantum Fourier transform* (QFT) is the same transformation but instead of a vector of inputs it transforms a state onto an orthonormal basis $|0\rangle, \dots, |N-1\rangle$

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle. \quad (2.16)$$

Where we now sum over k instead of over j by convention. We see that the quantum Fourier transform is the product of the matrix with entries $e^{2\pi i j k / N}$ at (k, j)

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} & & \vdots & & \\ \dots & & e^{2\pi i j k / N} & & \dots \\ & & \vdots & & \end{bmatrix} \quad (2.17)$$

An important property of the discrete Fourier transform is that any two columns j, j' are orthonormal

$$\begin{aligned}
& \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} e^{-2\pi i j' k/N} \frac{1}{\sqrt{N}} e^{2\pi i j k/N} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i (j' - j) k/N} = \frac{1}{N} \sum_{k=0}^{N-1} (e^{2\pi i (j' - j)/N})^k \\
&= \frac{1}{N} \frac{1 - e^{2\pi i (j' - j)}}{1 - e^{2\pi i (j' - j)/N}} = \begin{cases} 1 & \text{if } j' = j \\ 0 & \text{otherwise,} \end{cases} \quad (2.18)
\end{aligned}$$

where we have used Taylor's theorem for $x = e^{2\pi i (j' - j)/N}$ in the last step for $j' = j$

$$\frac{1}{N} \lim_{x \rightarrow 1} \frac{1 - x}{1 - x^{1/N}} = \frac{1}{N} \lim_{x \rightarrow 1} \frac{-1}{\frac{-1}{N} x^{\frac{N-1}{N}}} = \lim_{x \rightarrow 1} \frac{1}{x^{\frac{N-1}{N}}} = 1. \quad (2.19)$$

And if $j' - j \neq 0$ it is equal to an integer and the numerator becomes 0.

Since any two columns of F_N are orthonormal we have $F_N^\dagger F_N = I$, thus F is a unitary operator. Therefore it can be implemented on a quantum computer. We will derive its corresponding quantum circuit similar to [5]. To do this we rewrite the output of the quantum Fourier transform in its tensor product representation, taking $N = 2^n$, where n is the number of qubits we have. Expanding j in $|j\rangle$ in its binary representation

$$j = j_1 j_2 \dots j_n, \quad (2.20)$$

where all j_i are 0 or 1, we have

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0. \quad (2.21)$$

In the exponent of the quantum Fourier transform j is divided by $N = 2^n$. Note that j/N can be written as so-called binary fraction

$$0.j_1 j_2 \dots j_n = j_1/2 + j_2/2^2 + \dots + j_n/2^n.$$

Now we can rewrite

$$\begin{aligned}
F_N |j\rangle &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j (\sum_{l=1}^n k_l 2^{-l})} |k_1 \dots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1 \in \{0,1\}} \dots \sum_{k_n \in \{0,1\}} e^{2\pi i j (\sum_{l=1}^n k_l 2^{-l})} |k_1 \dots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1 \in \{0,1\}} \dots \sum_{k_n \in \{0,1\}} \bigotimes_{l=1}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle
\end{aligned}$$

$$\begin{aligned}
F_N |j\rangle &= \frac{1}{2^{n/2}} \bigotimes_{l=1}^n \sum_{k_l \in \{0,1\}} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^n (|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle) \\
&= \frac{(|0\rangle + e^{2\pi i j 2^{-1}} |1\rangle) \dots (|0\rangle + e^{2\pi i j 2^{-n}} |1\rangle)}{2^{n/2}} \\
&= \frac{(|0\rangle + e^{2\pi i j_1 j_2 \dots j_{n-1}} e^{2\pi i 0 \cdot j_n} |1\rangle) \dots (|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 \dots j_n} |1\rangle)}{2^{n/2}} \\
&= \frac{(|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle) (|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n} |1\rangle) \dots (|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 \dots j_n} |1\rangle)}{2^{n/2}}.
\end{aligned} \tag{2.22}$$

We see that on every qubit there has been a controlled application of multiple R_k gates

$$R_\phi \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{pmatrix} \tag{2.23}$$

The state of the first qubit $|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle = |0\rangle + (-1)^{j_n} |1\rangle$ is the same as a Hadamard on the qubit $|j_n\rangle$. For the second qubit the state is the same as applying

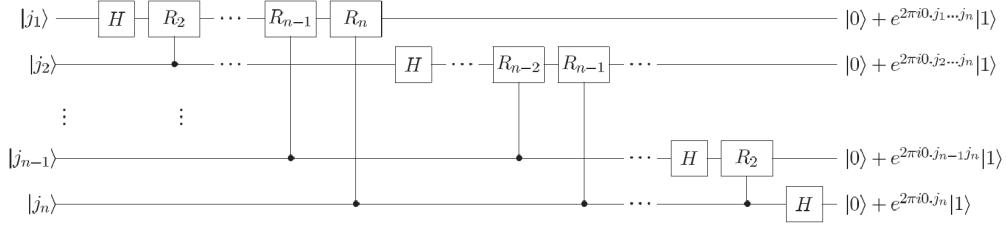


Figure 2.3: Quantum circuit for the implementation of the Quantum Fourier Transform. The order of the qubits is still reversed compared to equation (2.22). Also the outcomes on the rights should be multiplied by a normalization factor $\frac{1}{\sqrt{2}}$

. Image taken from [5].

a Hadamard and then a controlled R_n (before we applied the Hadamard on the first qubit),

$$|j_{n-1}\rangle \xrightarrow{H} (|0\rangle + e^{2\pi i 0.j_{n-1}} |1\rangle) \quad (2.24)$$

$$\xrightarrow{C_{j_n - R_2}} (|0\rangle + e^{2\pi i 0.j_{n-1}j_n} |1\rangle). \quad (2.25)$$

We see that we can create the m -th qubit from equation (2.22) by applying a Hadamard on $|j_{n-m}\rangle$ and then a series of appropriate controlled R gates on the qubits $|j_l\rangle$ for which $l > m$. The quantum circuit for this scheme is shown in fig 2.3.

However now we are left with the qubits in reversed order, to restore this we do a number of Swap operations to obtain the appropriate state. We can reverse all the qubits by just swapping $|j_l\rangle$ and $|j_{n-l}\rangle$ for all integer $l \leq n/2$.

Since we can apply a swap gate by applying 3 CNOT gates we need at most $3n/2$ extra gates to apply the swapping needed at the end. Until the swapping we have applied at most n gates to every qubit, since we have n qubits there are at most $\mathcal{O}(n^2)$ gates that we have to apply. This is exponentially faster than the best classical algorithm for the discrete Fourier transform. For example FFT, the Fast Fourier Transform, takes $\mathcal{O}(N \log(N)) = \mathcal{O}(n2^n)$. While this sounds like we found an exponential improvement over one of the most important algorithms in the world, this isn't quite true. We cannot directly access the amplitudes in the wave function since the whole state is in a quantum superposition. But for

some applications it is not necessary to know all the values of the amplitudes and we only use the properties of the Fourier transform, as we will see in the next algorithm.

2.4 Quantum Phase Estimation

Suppose we have a unitary operator U and an eigenvector $|u\rangle$ of U , such that $U|u\rangle = \lambda|u\rangle$, and we want to estimate the complex number λ . It turns out the Fourier transform gives us an efficient way to implement the estimation of λ in a quantum circuit. Since U is unitary the eigenvalue λ has norm 1, and can be written as $e^{2\pi i\phi}$ for some angle $\phi \in [0, 1)$. Suppose that $\phi = 0.\phi_1 \dots \phi_t$ can be written with t bits of precision. If this is not possible, we get arbitrarily close to ϕ by using a higher number of bits of precision. To see why this is possible take $\epsilon > 0$, then there always exists an N such that $\sum_{i=N+1}^{\infty} 1/2^i < \epsilon$. Then take the number of qubits to be N , and take as approximation $\tilde{\phi} = 0.\phi_1 \dots \phi_N$, then $\|\phi - \tilde{\phi}\| < \epsilon$. Approximating ϕ arbitrarily well implies we can do approximate λ well.

The quantum phase estimation algorithm works as follows

- Initialize $|0^t\rangle|u\rangle$
- Apply the $H^{\otimes t}$ to the first register $\longrightarrow \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle|u\rangle$
- Apply the map $|k\rangle|u\rangle \mapsto |k\rangle U^k|u\rangle$, obtaining $\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle U^k|u\rangle = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i k\phi} |k\rangle|u\rangle$
- Apply $\text{QFT}^{-1} = \text{QFT}^\dagger$ to the first register $\longrightarrow |\phi\rangle|u\rangle$
- Measure the first register ϕ
- Calculate $\lambda = e^{2\pi i\phi}$

The quantum circuit of the Quantum Phase Estimation is given in Figure 2.4. Note that the controlled operations of U in the second register perform the op-

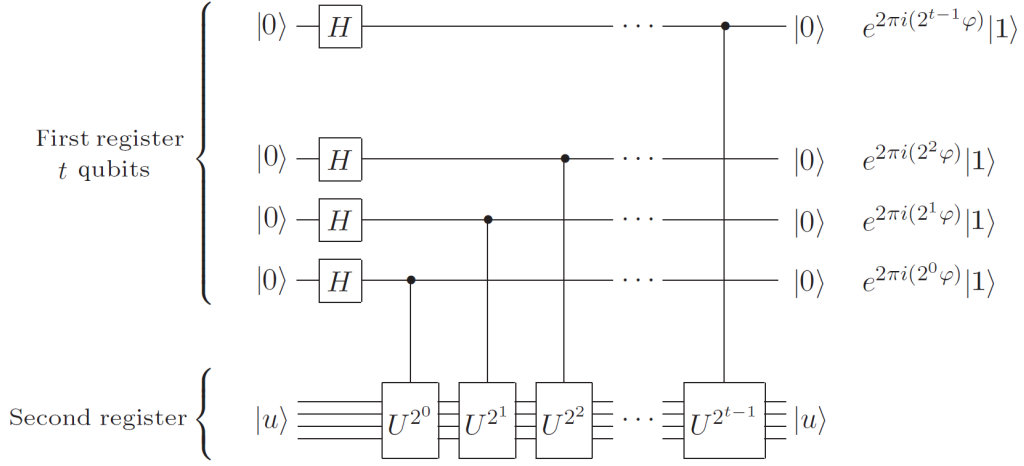


Figure 2.4: Quantum circuit for the estimation of the phase ϕ . Note that we have not performed the inverse QFT and the states on the right have to be multiplied by $\frac{1}{\sqrt{2}}$ to be normalized. Image taken from [5].

eration

$$|k\rangle |u\rangle = |k_1 k_2 \dots k_t\rangle |u\rangle \quad (2.26)$$

$$\rightarrow |k_1 k_2 \dots k_t\rangle U^{k_1 2^0 + k_2 2^2 + \dots + k_t 2^{t-1}} |u\rangle \quad (2.27)$$

$$= |k_1 k_2 \dots k_t\rangle U^{k_1 2^0 + k_2 2^2 + \dots + k_t 2^{t-1}} |u\rangle \quad (2.28)$$

$$= |k\rangle U^k |u\rangle. \quad (2.29)$$

Thus when applying the controlled operation U , we have to apply this operator 2^k times on the k -th qubit. It could be that the multiplication of U can be done more efficiently, for example if the operator was $U = ADA^\dagger$, we have $U^k = AD^k A^\dagger$, since $AA^\dagger = I$, such optimizations are implemented automatically by ProjectQ.

If we would implement the phase estimation algorithm on a quantum computer the pseudo code would be the following.

In short, the reason phase estimation is useful is because it can be used to estimate eigenvalues of unitary operators. The problem of solving a phase estimation is so common that often the short notation of the quantum circuit in fig 2.5 is used.

Algorithm 1: Quantum Phase Estimation algorithm to estimate the eigenvalue of some eigenvector of a unitary

```

1 function QuantumPhaseEstimation( $U, |u\rangle, t$ )
   Input : Unitary  $U$ , eigenstate  $|u\rangle$ , estimation precision  $t$  qubits initialized to  $|0\rangle$ 
   Output: Approximation of the corresponding eigenvalue of the  $\lambda$ 
2  $|0^t\rangle |u\rangle$  ; // initial state
3  $\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle |u\rangle$  ; //  $H^{\otimes t}$  on first register
4  $\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle U^k |u\rangle$  ; // apply controlled power of  $U$ 
5  $\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i k \phi} |k\rangle |u\rangle$  ; // result of  $U |u\rangle$ 
6  $|\phi_1 \dots \phi_t\rangle |u\rangle$  ; // apply inverse QFT on first register
7  $\phi_1 \dots \phi_t$ 

```

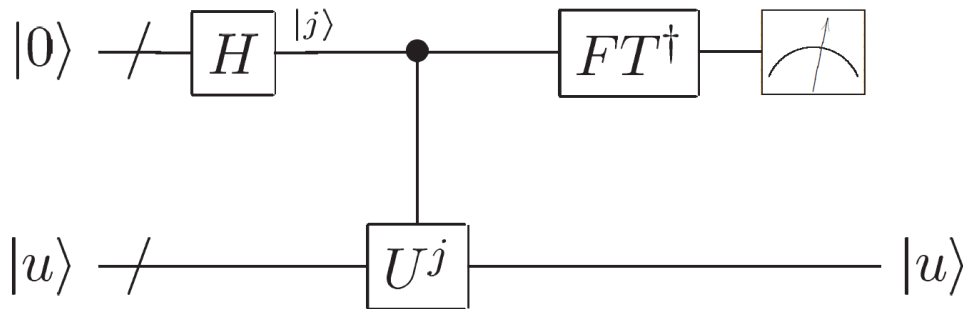


Figure 2.5: Schematic overview of Quantum Phase Estimation, the slashed line denotes a bundle of qubit wires. The state $|u\rangle$ is an eigenstate of U and the output in the measurement is the estimation of the angle ϕ of the eigenvalue $\lambda = e^{2\pi i \phi}$ corresponding to the $U |u\rangle$. Image from [5]

The difficult problem, however, is finding the corresponding eigenvector $|u\rangle$ of some unitary operator U . It is again important to realize that we do not necessarily need to know all the amplitudes in the wave function of $|u\rangle$, only that it is an eigenstate of U . This reasoning can even be made stronger, since the eigenvectors of a unitary form an orthonormal basis $\{|u_i\rangle\}_i$ we can write a general state $|\psi\rangle$ as a linear combination of the eigenvectors $|\psi\rangle = \sum_i s_i |u_i\rangle$. Performing the quantum phase estimation procedure we obtain

$$\begin{aligned}
|0^t\rangle |\psi\rangle &= \sum_i s_i |0^t\rangle |u_i\rangle \\
&\longrightarrow \sum_i s_i \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle |u_i\rangle \\
&\longrightarrow \sum_i s_i \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle U^k |u_i\rangle \\
&= \sum_i s_i \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i k \phi_i} |k\rangle |u_i\rangle \\
&\longrightarrow \sum_i s_i |\phi_i\rangle |u_i\rangle.
\end{aligned} \tag{2.30}$$

Measuring the first register in the computational basis collapses the register to some state $|\phi_j\rangle$, and we get as output an eigenvalue of some random eigenvector of U distributed according to the distribution of $|s_i|^2$.

2.5 Quantum Amplitude Estimation

The technique of Quantum Amplitude Estimation (QAE) introduced in [6] can be used to estimate the amplitude of state in superposition provided that we can reflect through this state. The technique can be described as follows:

Let \mathcal{H} denote the Hilbert space representing the state space of a quantum system, and suppose we have a Boolean function $\chi : \mathbb{Z} \rightarrow \{0, 1\}$ that induces a partition of \mathcal{H} into a direct sum of two subspaces, a good subspace and a bad subspace. The good subspace is spanned by the basis states $|x\rangle$ for which $\chi(x) = 1$, and the bad subspace is the orthogonal complement. Let \mathcal{A} be a unitary transformation that creates $|\Phi\rangle = \mathcal{A} |0^n\rangle$. Then write $|\Phi\rangle = |\Phi_1\rangle + |\Phi_0\rangle$ as a superposition of the (non-normalized) good ($|\Phi_1\rangle$) and bad ($|\Phi_0\rangle$) components of $|\Phi\rangle$.

Then *amplitude estimation* is the problem of estimating $a = \langle \Phi_1 | \Phi_1 \rangle$, the probability that a measurement of $|\Phi\rangle$ gives the good state. For this we introduce the operator \mathbf{Q}

$$\mathbf{Q} = -\mathcal{A} \mathbf{S}_0 \mathcal{A}^{-1} \mathbf{S}_\chi, \quad (2.31)$$

where \mathbf{S}_0 is the reflection that maps $|0^n\rangle \rightarrow -|0^n\rangle$ and leaves all other states untouched. And \mathbf{S}_χ changes the sign if the state is a good state

$$|x\rangle \rightarrow \begin{cases} -|x\rangle & \text{if } \chi(x) = 1 \\ |x\rangle & \text{if } \chi(x) = 0. \end{cases} \quad (2.32)$$

The operator \mathbf{Q} is unitary, hence the eigenvalues have norm 1, and it consists of two reflections thus it has determinant 1. Its eigenvalues can be written as $\lambda_\pm = -e^{2i\theta}$ where $\cos^2(\theta) = \langle \Phi_1 | \Phi_1 \rangle = a$. By applying the quantum phase estimation procedure from the previous section to the operator \mathbf{Q} we can get arbitrarily close, by using more ancillary qubits, to estimate the angle θ by only using one state $|\Phi\rangle$. With an estimation of θ we can then calculate the value c we are interested in.

Chapter 3

Quantum Simulations

Simulations of quantum chemistry and quantum many-body systems are important applications. For example simulating the molecular interaction of a medicine in our body increases our understanding of the working of the medicine. The search for catalysts in the industry relies on knowing how materials interact. Catalysts lower the activation energy needed for a chemical interaction. Understanding chemical interactions better can help us find new catalysts. The governing quantum physical equations that describe these interactions are well known and understood. However the implementation of these laws yields exponentially many equations that need to be solved. Classical simulations are inherently local operations that work on only a few bits at a time, while quantum physics consists of superpositions and is inherently non-local. Already for systems of 50 particles it becomes impossible to exactly simulate the quantum physics because the state space becomes too large to even store on a computer.

In 1982 Richard Feynman proposed to tackle these problems by using standard physical systems to simulate other physical systems [7]. This led to the idea of doing quantum simulations with quantum operations on a quantum computer [8]. The Schrödinger equation describes the time evolution of a quantum state $|\psi(0)\rangle$ under some Hamiltonian H . In the Schrödinger picture this is written as

$$e^{-iHt} |\psi(0)\rangle = |\psi(t)\rangle. \quad (3.1)$$

The question we will answer in this chapter is how one can use quantum computers to simulate physics and how we can build a circuit of the time evolution of some general Hamiltonian H to be applied on a state.

To do this we first introduce two important quantum algorithms that we will be used often in this thesis. The Quantum Fourier Transform (QFT) and the Quantum Phase Estimation Algorithm. Furthermore we will show how the implementation of Quantum Simulations in the quantum programming framework of ProjectQ, for this we will use the library of OpenFermion which translates the physics of a problem to the language of quantum simulators.

3.1 Hamiltonian Simulation

A potentially important area of quantum computation is the simulation of the time evolution of a physical system described by a Hamiltonian. If we have an initial state $|\psi\rangle$ governed under a Hamiltonian H and if H itself is time independent, the the time evolution of a general state is given by the time dependent Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H |\psi(t)\rangle. \quad (3.2)$$

For simplicity we will set the constant \hbar to 1 (this rescales the time into time steps of \hbar). We can rewrite equation (3.2) to its equation in the Schrödinger picture

$$e^{-iHt/\hbar} |\psi(0)\rangle = |\psi(t)\rangle, \quad (3.3)$$

here $e^{-iHt/\hbar}$ is the operator that time evolves some state by a time step t . For simplicity we can set the constant \hbar to 1 (this rescales the time into time steps of \hbar) to get

$$e^{-iHt} |\psi(0)\rangle = |\psi(t)\rangle. \quad (3.4)$$

Note that at every time step the operator is unitary since $e^{-iHt} e^{iHt} = I$. As mentioned before we can approximate any unitary with a quantum circuit [4]. The approach we will follow is to build the unitary $e^{-iH\Delta t}$ for some fixed time step Δt as a quantum circuit of gates, time evolution would then just be applying this unitary to a state a certain amount of times.

There are many applications where it is necessary to solve this problem. In quantum chemistry one wants to simulate the interactions between molecules to e.g. understand the interaction between a medicine and a receptor on a cell. Although the governing quantum mechanical equations of the interactions between particles are well known and can be written down in a Hamiltonian. Classically it is hard to simulate the time evolution of a larger system because the

size of the Hamiltonian is exponential in the number of particles.

If one would simulate the interaction of 40 fermionic modes for example, we would need to save and do operations on 2^{40} different amplitudes, if saving one amplitude takes 16 Bytes of RAM memory we would need 16 TB of RAM memory to save the wave function. It is clear that large systems, become impossible to simulate. While for a quantum computer we can have this, because to simulate 40 fermionic modes a quantum computer only needs 40 qubits. Between electron spins and qubits there is a one to one mapping, this is not true for a general particle but one can always use multiple qubits to simulate the particle interactions.

3.1.1 Pauli basis time evolution

We claimed it is possible to simulate the unitary $e^{-iH\Delta t}$ on a quantum computer, but it might not seem clear how this can be done for a general Hamiltonian. As shown in equation (2.11) we can calculate the exponents of Pauli gates. If our Hamiltonian is in the Pauli basis we could calculate the exponent of all the terms and find the time evolution unitary. To illustrate this we will now first show how to simulate a Hamiltonian that consists of single Pauli gates on every qubit. Consider the all Z gate Hamiltonian

$$H = Z_1 \otimes \cdots \otimes Z_n, \quad (3.5)$$

then the action of the unitary $e^{-iH\Delta t}$ applied to a state $|\psi\rangle$ yields

$$\begin{aligned} e^{-iH\Delta t} |\psi\rangle &= \sum_k \frac{1}{k!} (-iH(\Delta t))^k |\psi\rangle \\ &= \left(1 - iH(\Delta t) - \frac{(\Delta t)^2}{2!} + \frac{iH(\Delta t)^3}{3!} + \frac{(\Delta t)^4}{4!} + \dots \right) |\psi\rangle \\ &= \left(1 - \frac{(\Delta t)^2}{2!} + \frac{(\Delta t)^4}{4!} + \dots \right) |\psi\rangle - i \left((\Delta t) - \frac{(\Delta t)^3}{3!} + \dots \right) H |\psi\rangle \\ &= \cos((\Delta t)) |\psi\rangle - i \sin((\Delta t)) (-1)^{\text{Parity}} |\psi\rangle \\ &= e^{-\text{Parity } i t} |\psi\rangle, \end{aligned} \quad (3.6)$$

where Parity denotes the parity of the state in the computational basis, and we have used that $H^2 = I$. We can conclude that time evolution in this case is just the multiplication of the phase $e^{-\text{Parity } i t}$. We can realize this in a quantum circuit

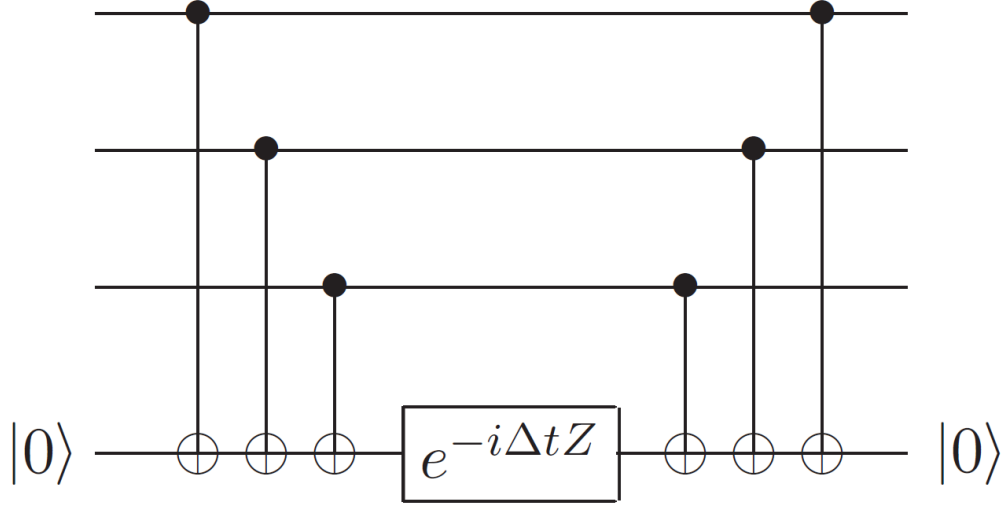


Figure 3.1: Quantum circuit that applies the time evolution operator e^{-iHt} , where $H = Z_1 \otimes \cdots \otimes Z_n$. Image from [5].

by setting up an ancilla qubit as the parity of the system and then applying $e^{-i\delta t Z}$ unitary to the ancilla qubit. Setting up the ancilla as the parity of the system can be done by applying a CNOT between every qubit and the ancilla where the ancilla qubit is the target. This circuit is shown for 3 qubits in Figure 3.1.

In fact with this procedure we can simulate any Hamiltonian that consists of a single Pauli or Identity gate per qubit

$$H = \bigotimes_{k=1}^n \sigma_i^k, \quad (3.7)$$

where k denotes the k th qubit and $\sigma \in \{I, X, Y, Z\}$. If in equation (3.1) one of the local operators was a I instead of a Z we would just not apply the CNOT gate to this qubit and the time evolution would still be exact. By using the following relations of the Pauli matrices

$$X = HZH \quad (3.8)$$

$$Y = R_x(\pi/2) Z R_x(\pi/2)^\dagger, \quad (3.9)$$

we can apply a unitary on a X or Y term to transform it to a Z gate and the

previously mentioned scheme can be applied.

3.1.2 Trotter-Suzuki Decomposition

We have seen that for a Hamiltonian given by a single term consisting of many local Pauli operators we can make an exact quantum circuit to simulate its time evolution on a state. In general, a Hamiltonian is just a sum of Hamiltonians of the form (3.8), and often consists only of single-qubit and 2-qubit gates. Examples of such Hamiltonians are the Ising model and the XXZ model [9]. Suppose we have an Hamiltonian that can be written as a sum of such local Hamiltonians

$$H = \sum_{j=1}^m H_j, \quad (3.10)$$

where m is polynomial in the number of qubits n . Our goal is to write a quantum circuit for this general Hamiltonian similar as we have done for the single local Hamiltonian. We would like to expand the time evolution of 3.10 as

$$e^{-i\sum_{j=1}^m H_j t} \stackrel{?}{=} \prod_{j=1}^m e^{-iH_j t}, \quad (3.11)$$

but this is not correct, since exponentials of matrices do not commute in general, i.e. $e^{A+B} \neq e^A e^B$. To solve this we can use the Trotter-Suzuki approximation [10],

$$e^{A+B} = e^A e^B + O(\|A\|_{op}\|B\|_{op}), \quad (3.12)$$

where $\|\cdot\|_{op}$ denotes the operator norm. So if A and B have a small operator norm we can get approximately equal. We can use this approximation in the Hamiltonian evolution operator by making the operator norm smaller, we do have

$$e^{-i\sum_{j=1}^m H_j t} = (e^{-i\sum_{j=1}^m H_j t/q})^q, \quad (3.13)$$

since $\sum_{j=1}^m H_j$ commutes with itself. Then the error for one time slice t/q becomes [11]

$$e^{-i\sum_{j=1}^m H_j t/q} = \prod_{j=1}^m e^{-iH_j t/q} + O\left(\frac{\|H_1\|_{op}\|\dots\|_{op}\|H_q\|t^2}{q^2}\right), \quad (3.14)$$

$$(3.15)$$

and the error for the time step evolution t becomes

$$(e^{-i\sum_{j=1}^m H_j t/q})^q = \left(\prod_{j=1}^m e^{-iH_j t/q} + O\left(\frac{\|H_1\|_{op} \dots \|H_q\|_{op} t^2}{q^2}\right) \right)^q, \quad (3.16)$$

$$= \left(\prod_{j=1}^m e^{-iH_j t/q} \right)^q + O\left(\frac{\|H_1\|_{op} \dots \|H_q\|_{op} t^2}{q^2}\right), \quad (3.17)$$

where in the last line we used that the product of the unitaries on the left add at most linearly in the error so the dominating term in the error will be

$$q O\left(\frac{\|H_1\|_{op} \dots \|H_q\|_{op} t^2}{q^2}\right).$$

This is the first Suzuki-Trotter approximation of the time evolution unitary and this approach was first introduced by Lloyd [8] in 1996. And this approximation is the foundation of quantum time evolution and it makes it possible for us to simulate quantum physics efficiently within error bounds.

We have spliced the time evolution step t into q smaller blocks to remain with a time evolution of a time step t . We could also rewrite the approximation for the spliced time step $\Delta t = t/q$. This approximation would then be

$$e^{-i\sum_{j=1}^m H_j \Delta t} = \prod_{j=1}^m e^{-iH_j \Delta t} + O\left(\|H_1\|_{op} \dots \|H_q\|_{op} (\Delta t)^2\right). \quad (3.18)$$

We can write down a scheme for quantum time evolution for a time step t .

- Initialize a state $|\psi(0)\rangle$
- Generate the circuit $U_{\Delta t}$ of the time evolution operator $e^{-i\sum_{j=1}^m H_j \Delta t}$
- Apply the circuit $U_{\Delta t} |\psi(0)\rangle = |\psi(\Delta t)\rangle$
- Loop the circuit $t/\Delta t$ times
- Return the final result $|\psi(t)\rangle$

In conclusion we have shown that it is possible to simulate the time evolution of n particles evolving under an Hamiltonian in the Pauli basis with an accuracy we using only $O(n)$ qubits.

3.1.3 Jordan-Wigner Transformation

In the previous section we derived a circuit for a general Hamiltonian in the Pauli operator basis. However in fermionic quantum many-body systems the Hamiltonian is often stated in the second quantization formalism. In this formalism creation and annihilation operators are introduced which work on states representing modes in the Fock state basis. Where a $|0\rangle$ means there is not particle in the mode and $|1\rangle$ that there is. The creation (c^\dagger) and annihilation (c) operators are applied on the fermionic modes in the Fock state basis

$$c^\dagger |0\rangle = |1\rangle \quad c^\dagger |1\rangle = 0 \quad (3.19)$$

$$c |0\rangle = 0 \quad c |1\rangle = |0\rangle. \quad (3.20)$$

Here in the Fock state basis a state is given $|n_0 \dots n_j\rangle$ where n_i denotes the number of particles in the i -th state. In our case we only consider the fermionic states so $n_i \in \{0, 1\}$. The creation and annihilation operators have fermionic anticommutation relations

$$\{c_j, c_i\} = 0 \quad (3.21)$$

$$\{c_j^\dagger, c_i^\dagger\} = 0 \quad (3.22)$$

$$\{c_j^\dagger, c_i\} = \delta_{i,j} I, \quad (3.23)$$

where $\{A, B\} \equiv AB + BA$ is the anticommutator and $\delta_{i,j}$ is the Kronecker delta. These creation and annihilation operators are very useful for describing quantum many-body systems. The counting operator that counts how many particles there are on at site i can be described by $c_i^\dagger c_i$. If there is no particle this operator returns 0, if there is particle it returns 1. The operator $c_i^\dagger c_j$ describes particle hopping, by annihilating a particle at site j then creating one at site i .

The most general Hamiltonian in second quantization in the Born-Oppenheimer approximation [12], ignoring the nuclear kinetic en nuclear nuclear interaction terms, can be written as

$$H = \sum_{pq} h_{pq} c_p^\dagger c_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} c_p^\dagger c_q^\dagger c_r c_s, \quad (3.24)$$

where the first term represents the single-electron and the second term the Coulomb electron-electron interaction.

We want to translate this formalism to gates and states on a quantum computer so that we can simulate the many-body physics corresponds to We first havet to describe the Fock space and the creation and annihilation operators in the framework we used in the last section. The fermionic states in the Fock space can already be perfectly represented by the computational basis states of a qubit. The *Jordan-Wigner transformation* introduced by Pascual Jordan and Eugene Wigner in 1927 to map the spin operators to fermionic operators can be used the other way round to map the fermionic operators in the second quantization to the spin-operators in the Pauli basis [13].

As an initial guess we could set $c_j = |0\rangle\langle 1|$ and $c_j^\dagger = |1\rangle\langle 0|$ where the brackets are applied to the j -th qubit. From the relations in (3.21) we can write using Pauli matrices

$$|0\rangle\langle 1| = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad (3.25)$$

$$= \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (3.26)$$

$$= \left(\frac{\sigma_i^x + i\sigma_i^y}{2} \right) \quad (3.27)$$

$$|1\rangle\langle 0| = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad (3.28)$$

$$= \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (3.29)$$

$$= \left(\frac{\sigma_i^x - i\sigma_i^y}{2} \right). \quad (3.30)$$

Although they obey the commutation relation for equal index

$$\{c_j, c_j^\dagger\} = |0\rangle\langle 1| |1\rangle\langle 0| + |1\rangle\langle 0| |0\rangle\langle 1| \quad (3.31)$$

$$= |0\rangle\langle 0| + |1\rangle\langle 1| = I, \quad (3.32)$$

they also commute if they have different indices $[c_j^\dagger, c_i]$, while fermionic operators must anticommute. By multiplying the operators by a phase dependent on the general configuration we can recover the correct commutation relations, this phase dependent on the configuration is a chain of $-\sigma^z$ operators working

applied to every site before the i -th site

$$c_i = \prod_{k=0}^{i-1} (-\sigma_i^z) |0\rangle \langle 1|_i = \prod_{k=0}^{i-1} (-\sigma_i^z) \left(\frac{\sigma_i^x + i\sigma_i^y}{2} \right) \quad (3.33)$$

$$c_i^\dagger = \prod_{k=0}^{i-1} (-\sigma_i^z) |1\rangle \langle 0|_i = \prod_{k=0}^{i-1} (-\sigma_i^z) \left(\frac{\sigma_i^x - i\sigma_i^y}{2} \right) \quad (3.34)$$

$$2c_i^\dagger c_i - 1 = 2|0\rangle \langle 0| - I = |0\rangle \langle 0| - |1\rangle \langle 1| = \sigma_i^z. \quad (3.35)$$

These still obey the commutation relation for equal index, but now for different indices we get phases, where we use $\prod_{k=0}^{i-1} (-\sigma_i^z) = e^{-i\pi \sum_{k=0}^{i-1} c_k^\dagger c_k}$ (omitting the σ^x, σ^y and using $i > j$ without loss of generality)

$$\{c_i, c_j^\dagger\} = e^{-i\pi \sum_{k=0}^{i-1} c_k^\dagger c_k} e^{i\pi \sum_{k=0}^{j-1} c_k^\dagger c_k} + e^{-i\pi \sum_{k=0}^{j-1} c_k^\dagger c_k} e^{i\pi \sum_{k=0}^{i-1} c_k^\dagger c_k} \quad (3.36)$$

$$= e^{-i\pi \sum_{k=j}^{i-1} c_k^\dagger c_k} + e^{i\pi \sum_{k=j}^{i-1} c_k^\dagger c_k} = 0 \quad (3.37)$$

$$\{c_i, c_j\} = 0 \quad (3.38)$$

$$\{c_i^\dagger, c_j^\dagger\} = 0, \quad (3.39)$$

as desired.

Summarizing, the creation and annihilation operators transform similarly to what one would expect from looking at the Pauli operators, but to preserve the anticommutation relations they are multiplied with a chain of σ^z operators. From this last definition we can derive nice expressions for the Jordan-Wigner transform back to second quantization,

$$\sigma_i^x = -(\sigma_1^z \dots \sigma_{i-1}^z)(c_i + c_i^\dagger) \quad (3.40)$$

$$\sigma_i^y = i(\sigma_1^z \dots \sigma_{i-1}^z)(c_i^\dagger - c_i) \quad (3.41)$$

$$\sigma_i^z = |0\rangle \langle 0|_j - |1\rangle \langle 1|_j = (c_i c_i^\dagger - c_i^\dagger c_i). \quad (3.42)$$

This shows that for adjacent Pauli operators we can rewrite the Pauli z chain as

$$\sigma_i^x \sigma_{i+1}^x = (c_i + c_i^\dagger) \sigma_i^z (c_{i+1} + c_{i+1}^\dagger) \quad (3.43)$$

$$= (c_i^\dagger - c_i)(c_{i+1} + c_{i+1}^\dagger) \quad (3.44)$$

$$\sigma_i^y \sigma_{i+1}^y = -(c_i^\dagger - c_i) \sigma_i^z (c_{i+1}^\dagger - c_{i+1}) \quad (3.45)$$

$$= -(c_i + c_i^\dagger)(c_{i+1}^\dagger - c_{i+1}), \quad (3.46)$$

note that the σ_i^z operator was applied from right to left on the second quantized operators.

This that we can map the operators and the states in second quantization to a system in a Pauli basis. Since we have seen that in the previous section we can simulate the time evolution of these Hamiltonians in the Pauli basis to arbitrary precision, we now can simulate the time evolution of second quantized Hamiltonians as well. We have shown that Feynman’s conjecture that any local quantum system can be simulated by a quantum computer is indeed true.

3.1.4 Hamiltonian Simulation in ProjectQ

So far we have given a theoretical basis of on the implementation of a quantum circuit that evolves a quantum state in time. We have shown how a Hamiltonian in terms of the Pauli operators can be simulated, and we have shown that there exists a mapping from the second quantization formalism in the fermionic Fock space to Pauli operators in the computational basis. These tools would allow us to simulate quantum physics on a quantum computer with an exponential increase in speed compared to classical simulations.

One of the main interest of this thesis is the implementation of the quantum circuit for time evolution on a quantum simulator. In this section we will show how to implement the procedure for time evolution of a second quantized Hamiltonian in ProjectQ.

We will use *OpenFermion*, an open-source chemistry package in Python for quantum computers initiated by Google [14]. The goal of *OpenFermion* is to provide a tool to generate and compile equations from quantum chemistry and quantum many-body systems which can be translated to a quantum circuit to be used on a quantum computer. It serves as a general framework that can be used in different quantum computing programming frameworks.

In the computer simulations in this thesis we will start from a Hamiltonian in second quantized form. We will show how *OpenFermion* translates this into code used in the ProjectQ framework. Fermionic creation and annihilation operators a_p^\dagger and a_q for e.g. $p = 2, q = 0$ are represented as

```
1 from openfermion.ops import FermionOperator
2 a_p_dagger = FermionOperator('2^')
```

```
3 a_q = FermionOperator('0')
```

The anticommutation relations from (3.21) are automatically taken into account in future operations. An arbitrary fermionic operator working on multiple sites such as $W = a_3 a_2^\dagger a_5^\dagger a_9$ is written as follows

```
1 from openfermion.ops import FermionOperator
2 W = FermionOperator('3 2^ 5^ 9')
```

As shown before we can perform a Jordan-Wigner transformation on the Fock space operators to get their representation in the Pauli basis, this representation can then be used in ProjectQ. The Pauli basis can also be written as a sparse matrix. This representation is the classical representation and can be used for classical simulations. In OpenFermion both these transformation are supported. For a simple operator $Q = a_1$ we obtain

```
1 from openfermion.transforms import jordan_wigner,
   get_sparse_operator
2 Q = FermionOperator('1')
3 ham_jw = jordan_wigner(W)
4 ham_matrix = get_sparse_operator(W)
```

the output of the Jordan Wigner transformation equals

```
1 0.5 [Z0 X1] +
2 0.5 j [Z0 Y1]
```

The output of the matrix will be in the sparse matrix form of SciPy, since there are many more efficient algorithms for sparse matrices. In the dense matrix form we would get

```
1 matrix([[ 0.+0.j,  1.+0.j,  0.+0.j,  0.+0.j ],
2         [ 0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j ],
3         [ 0.+0.j,  0.+0.j,  0.+0.j, -1.+0.j ],
4         [ 0.+0.j,  0.+0.j,  0.+0.j,  0.+0.j ]])
```

With the Jordan-Wigner transformed operator we can calculate the time evolution e^{-iQt} of the operator Q . The circuit for this evolution can be calculated exactly as shown in (3.7). To do this in OpenFermion we include

```
1 from openfermion.utils import trotterize_exp_qubop_to_qasm as
   trot
2 ham_trot = trot(jordan_wigner(Q))
```

this returns a generator object which generates

```
1 ['H 1',
2  'CNOT 0 1',
```

```

3 'Rz 0.5 1',
4 'CNOT 0 1',
5 'H 1',
6 'Rx 1.5707963267948966 1',
7 'CNOT 0 1',
8 'Rz 0.0 1',
9 'CNOT 0 1',
10 'Rx -1.5707963267948966 1']

```

This is what we expect from the calculation in (3.7). If our Hamiltonian would not have been local then OpenFermion would have applied the first order Suzuki-Trotter approximation. To get a better precision by splicing the time step we can divide the Hamiltonian by m , get the circuit and then loop the circuit m times.

The matrix form of the Hamiltonian that we can generate in OpenFermion gives the operator in the classical case. To create the time-evolution operator e^{-iHt} we can do direct matrix exponentiation of H . Since H will be sparse we can use optimized algorithms to do this efficiently.

```

1 import scipy
2 time_evo = scipy.linalg.sparse.expm(ham_matrix)

```

It is important to note that our numerical simulations are underneath still classical. ProjectQ, after optimization of the circuit, applies all operations on the entire wave function as classical operators. This means that all our numerical quantum simulations are still classically implementable. In fact, the only difference between the numerical quantum simulation and the classical simulation of directly exponentiating the matrix is the error of the Suzuki-Trotter approximation. Thus, the numerical quantum simulation is the classical information with an error. The quantum simulation should therefore by construction always approach the solution of the classical simulation.

The algorithm to time evolve an initial state $|\psi(0)\rangle$ by some time step t and q splices is given below.

Algorithm 2: Time evolution procedure of a Hamiltonian in second quantization.

```

1 function TimeEvolution ( wavefunction, Hamiltonian, t, q )
   Inputs      : The coefficients of the states in the wave function of  $|\psi(0)\rangle$ ,
                  Hamiltonian in second quantization, time t, number of splices
                  q
   Outputs    : Time evolved state  $|\psi(t)\rangle$ 
2  $|\psi(0)\rangle \leftarrow \text{INITIALZIE}(\text{wavefunction})$ ; // initial state
3  $H \leftarrow \text{FermionicOperator}(\text{Hamiltonian})$ ; // initialize Hamiltonian
4  $H_{jw} \leftarrow \text{JordanWigner}(H)$ ; // Jordan-Wigner transformation
5  $\text{Evo}_q \leftarrow \text{Trot}(H_{jw}/q)$ ; // Creation of  $e^{-iH/q}$ 
6 for  $i = q$  to  $i = 0$  do
7    $|\psi(t' + q)\rangle \xleftarrow{\text{Evo}_q} |\psi(t')\rangle$ ; // Loop spliced time evolution
8 return  $|\psi(t)\rangle$ ; // Return time evolved state

```

One might have the impression that the time evolution procedure of a state on a quantum computer is the same as the classical evolution but only quicker. This is however not the case, after the quantum time evolution procedure we have the time evolved state but in a superposition. Therefore we do not know the full wave function, in contrast to a classical time evolution case. In the case of quantum time evolution, and many quantum algorithms in general, the fact that we can apply the algorithm is not enough. We also need to be able to extract some useful information out of the final system. An example of a way to do this is the application of the Quantum Phase Estimation algorithm. If we know that our time evolved state $|\psi(t)\rangle$ is the eigenvector of some unitary operator U we can approximate its eigenvectors efficiently using only one initialization of $|\psi(t)\rangle$. A way to do this will be introduced in the next chapter.

Chapter 4

Entanglement Entropy

Entropy is an important concept in quantum information theory. It measures how much uncertainty there is in the state of a system and is related to the amount of microstates there can be in a known system known characterized by *Boltzmann's entropy formula*

$$S = k_B \ln W, \quad (4.1)$$

where W is the number of microstates there are corresponding to a given macrostate. The second law of thermodynamics states that the entropy of an isolated system may never decrease over time and thus such systems evolve towards a thermodynamic equilibrium. In this chapter we will use the concept of entropy as defined in quantum information theory as a measure of entanglement between two systems.

In 1948 Claude Shannon introduced the concept of classical information entropy, known as the *Shannon Entropy* [15]

$$H(X) \equiv - \sum_x p_x \log p_x, \quad (4.2)$$

where p_1, \dots, p_n is the probability distribution of some discrete variable X . The concept of information entropy can be used to quantify the amount of information there is in X .

The Shannon entropy is associated with a classical probability distribution. The entropy of quantum states was defined similarly already in 1932 when von

Neumann introduced the concept of entropy of a quantum state as known as the *Von Neumann Entropy* [16]

$$S(\rho) \equiv -\text{Tr} [\rho \log \rho], \quad (4.3)$$

where ρ is a density matrix. In this definition classical probability distributions are replaced by density operators. A useful way to rewrite this definition is

$$S(\rho) = -\sum_x \lambda_x \log \lambda_x, \quad (4.4)$$

where λ_x are the eigenvalues of the density matrix ρ in a Hilbert space \mathcal{H} .

The concept of the von Neumann entropy is to quantify how much a state is away from a pure state. For a pure state one has $\rho = |\psi\rangle \langle\psi|$ which has eigenvector $|\psi\rangle$ with eigenvalue 1 and thus it has von Neumann entropy 0. The entropy is maximal if all λ_x are equal, in that case we have

$$\begin{aligned} S(\rho) &\leq -\sum_x \frac{1}{\dim(\mathcal{H})} \log \left(\frac{1}{\dim(\mathcal{H})} \right) \\ &\leq \frac{\dim(\mathcal{H})}{\dim(\mathcal{H})} \log (\dim(\mathcal{H})) \\ &\leq \log (\dim(\mathcal{H})). \end{aligned} \quad (4.5)$$

If a density matrix is a probabilistic sum of pure states, i.e.

$$\rho = \sum_x p_x |\psi_x\rangle \langle\psi_x|, \quad (4.6)$$

we say that the state ρ is mixed. The von Neumann entropy then gives a measure to how mixed the state actually is.

4.1 Renyi Entropy and Entanglement Entropy

The concept of the von Neumann entropy can be extended to a broader family of information measures known as the Renyi entropies. These are defined as

$$R_\alpha = \frac{1}{1-\alpha} \log(\text{Tr}[\rho^\alpha]), \quad (4.7)$$

which in the limit for $\alpha \rightarrow 1$ converges to the von Neumann entropy. The Renyi entropies behave similarly as the von Neumann entropy and for $\alpha > 1$, R_α is bounded by the von Neumann entropy. As we will see later in this chapter, the Renyi entropies are useful because for integer values $\alpha \geq 2$ it is possible to calculate them on a quantum computer.

Extending the concept of quantum entropies we introduce the entropy of entanglement. This is a measure for a many-body quantum state and measures how entangled two parts of the system are. In the case we create a bi-partition A, B of the state ρ where the two parts together make the entire state, the von Neumann entanglement entropy is defined as the Von Neumann entropy on the reduced states ρ_A or ρ_B . The reduced states can be obtained by performing a partial trace on the other subsystem

$$\rho_A = \text{Tr}_B[\rho] \quad \rho_B = \text{Tr}_A[\rho]. \quad (4.8)$$

The entanglement entropy then becomes

$$S(\rho_A) = -\text{Tr}[\rho_A \log \rho_A], \quad (4.9)$$

which again can be extended to a broader family of entanglement entropies known as the Renyi entanglement entropies

$$R_\alpha(A) = \frac{1}{1-\alpha} \log(\text{Tr}[\rho_A^\alpha]) = \frac{1}{1-\alpha} \log\left(\sum_i \lambda_i^\alpha\right), \quad (4.10)$$

where λ_i are the eigenvalues of ρ_A . By the Schmidt decomposition we can write any state $|\Psi\rangle_{A,B} \in A \otimes B$ as [17]

$$|\psi\rangle = \sum_i s_i |e_i\rangle \otimes |f_i\rangle, \quad (4.11)$$

where $s_i > 0$, and $\{|e_i\rangle\}_i$ and $\{|f_i\rangle\}_i$ are two orthonormal sets. In this basis the reduced states ρ_A and ρ_B can be rewritten as

$$\rho_A = \sum_i s_i^2 |e_i\rangle \langle e_i| \quad \rho_B = \sum_i s_i^2 |f_i\rangle \langle f_i|, \quad (4.12)$$

and we see that the eigenvalues of both reduced states are equal, and thus

$$R_\alpha(A) = R_\alpha(B)$$

for all values of α .

In this thesis we are interested in the second Renyi entanglement entropy

$$R_2(\rho) = -\log(\text{Tr}(\rho_A^2)) \quad (4.13)$$

The reason is that we can calculate this quantity on a quantum computer, as we will show later. What we will do for a general state $|\Psi\rangle$ is to time evolve this by the time evolution operator e^{-iHt} , where H is the Hamiltonian of the system, and calculate the second Renyi entanglement entropy at every time step we make. Using this approach we can make statements about how the system evolves through time.

To show the non-triviality behind the entanglement entropy we will calculate the entanglement entropy of $|\psi\rangle = \frac{1}{\sqrt{3}}(|00\rangle + |01\rangle + |10\rangle)$, where system A has the first qubit and system B the second. This state is clearly entangled since measuring $|1\rangle$ for the first qubit tells us what the second qubit is, however it is not fully entangled, since measuring $|0\rangle$ does not give us information. Writing out $\rho = |\psi\rangle\langle\psi|$ we get

$$\begin{aligned} \rho &= \frac{1}{3} (|00\rangle\langle 00| + |00\rangle\langle 01| + |00\rangle\langle 10| + |01\rangle\langle 00| + |01\rangle\langle 01| + \\ &\quad |01\rangle\langle 10| + |10\rangle\langle 00| + |10\rangle\langle 01| + |10\rangle\langle 10|) \quad (4.14) \\ &= \frac{1}{3} (|0\rangle\langle 0|_A \otimes |0\rangle\langle 0|_B + |0\rangle\langle 0|_A \otimes |0\rangle\langle 1|_B + |0\rangle\langle 1|_A \otimes |0\rangle\langle 0|_B \\ &\quad + |0\rangle\langle 0|_A \otimes |1\rangle\langle 0|_B + |0\rangle\langle 0|_A \otimes |1\rangle\langle 1|_B + |0\rangle\langle 1|_A \otimes |1\rangle\langle 0|_B \\ &\quad + |1\rangle\langle 0|_A \otimes |0\rangle\langle 0|_B + |1\rangle\langle 0|_A \otimes |0\rangle\langle 1|_B + |1\rangle\langle 1|_A \otimes |0\rangle\langle 0|_B), \quad (4.15) \end{aligned}$$

tracing out the B subsystem we get

$$\begin{aligned}
&= \frac{1}{3} (|0\rangle \langle 0|_A \otimes |0\rangle \langle 0|_B + |0\rangle \langle 0|_A \otimes |1\rangle \langle 1|_B + |0\rangle \langle 1|_A \otimes |0\rangle \langle 0|_B \\
&\quad + |0\rangle \langle 0|_A \otimes |1\rangle \langle 0|_B + |0\rangle \langle 0|_A \otimes |1\rangle \langle 1|_B + |0\rangle \langle 1|_A \otimes |1\rangle \langle 0|_B \\
&\quad + |1\rangle \langle 0|_A \otimes |0\rangle \langle 0|_B + |1\rangle \langle 0|_A \otimes |0\rangle \langle 1|_B + |1\rangle \langle 1|_A \otimes |0\rangle \langle 0|_B)
\end{aligned} \tag{4.16}$$

$$\begin{aligned}
\rho_A &= \frac{1}{3} (|0\rangle \langle 0|_A + |0\rangle \langle 1|_A + |0\rangle \langle 0|_A + |1\rangle \langle 0|_A + |1\rangle \langle 1|_A) \\
&= \frac{1}{3} (2|0\rangle \langle 0|_A + |0\rangle \langle 1|_A + |1\rangle \langle 0|_A + |1\rangle \langle 1|_A)
\end{aligned} \tag{4.17}$$

$$\begin{aligned}
\rho_A^2 &= \frac{1}{9} (4|0\rangle \langle 0| + 2|0\rangle \langle 1| + |0\rangle \langle 0| + |0\rangle \langle 1| + 2|0\rangle \langle 1| \\
&\quad + |1\rangle \langle 1| + |1\rangle \langle 0| + |1\rangle \langle 1|)
\end{aligned} \tag{4.18}$$

$$= \frac{1}{9} (5|0\rangle \langle 0| + 3|0\rangle \langle 1| + 2|1\rangle \langle 0| + 2|1\rangle \langle 1|) \tag{4.19}$$

$$R_2(A) = -\log \text{Tr}[\rho_A^2] = -\log \left(\frac{7}{9} \right) \approx 0.363. \tag{4.20}$$

So determining the entanglement entropy is not trivial. Classically this becomes increasingly more difficult for larger states, because the states become exponentially larger. However, as we will see in this chapter, it is possible to calculate the entanglement entropy on a quantum computer.

4.2 Swap Test

In this section we will show how to calculate integer Renyi entropies for a general quantum state. Classically one would need to know the full wave function of a state to calculate the entropy and this is not always possible. There is however a quantum algorithm that can calculate the Renyi entropy on a quantum computer provided we have two copies of the quantum state. Since by the no-cloning theorem it is not possible to clone an arbitrary state one needs to have a routine from some initial state that we can do twice. For example in the case of the time evolution of an initial state, we could prepare the initial state twice and perform the same time evolution on both systems to get two copies of the time evolved state.

The algorithm makes use of the Swap gate. One can also apply the Swap operator to a subsystem A of the state. This operator is denoted as Swap_A and when applied on two equal states $|\phi\rangle_{A,B} |\phi\rangle_{A',B'}$ we get

$$\text{Swap}_A |\phi\rangle_{A,B} |\phi\rangle_{A',B'} = |\phi\rangle_{A',B} |\phi\rangle_{A,B'}. \quad (4.21)$$

Note that even though we applied the Swap_A operator to the same state the new state is completely different from the old one. This becomes clear in the next example where subsystem A is the first qubit and B the second

$$\begin{aligned} \text{Swap}_A \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ = \frac{1}{2} \text{Swap}_A(|00\rangle |00\rangle + |00\rangle |11\rangle + |11\rangle |00\rangle + |11\rangle |11\rangle) \\ = \frac{1}{2}(|00\rangle |00\rangle + |10\rangle |01\rangle + |10\rangle |01\rangle + |11\rangle |11\rangle) \end{aligned} \quad (4.22)$$

It turns out that the expectation of the Swap gate $\langle \Psi | \text{Swap} | \Psi \rangle = \text{Tr}(\rho^2)$, where $|\Psi\rangle = |\psi\rangle |\psi\rangle$. And $\langle \Psi | \text{Swap}_A | \Psi \rangle = \text{Tr}(\rho_A^2)$. We will show this first for a cyclic operator C_n that cyclically permutes the states as

$$C_n |a_1, a_2, \dots, a_n\rangle = |a_2, a_3, \dots, a_n, a_1\rangle, \quad (4.23)$$

then for $\text{Tr} \rho^{\otimes n} C_n$ we obtain

$$\begin{aligned} \text{Tr} \rho^{\otimes n} C_n &= \sum_{a_1, \dots, a_n} \langle a_1, \dots, a_n | \rho^{\otimes n} C_n | a_1, \dots, a_n \rangle \\ &= \sum_{a_1, \dots, a_n} \langle a_1, \dots, a_n | \rho^{\otimes n} | a_2, \dots, a_n, a_1 \rangle \\ &= \sum_{a_1, \dots, a_n} \langle a_1 | \rho | a_2 \rangle \langle a_2 | \rho | a_3 \rangle \dots \langle a_{n-1} | \rho | a_n \rangle \langle a_n | \rho | a_1 \rangle \\ &= \sum_{a_1} \langle a_1 | \rho^n | a_1 \rangle \end{aligned} \quad (4.24)$$

$$= \text{Tr} \rho^n, \quad (4.25)$$

where in (4.24) we have used the completeness relation of the Hilbert state that $\sum_a |a\rangle \langle a| = I$.

Since the Swap gate is the cyclic operator C_2 we now have

$$\begin{aligned}
\langle \Psi | \text{Swap} | \Psi \rangle &= \text{Tr}(\langle \psi | \langle \psi | \text{Swap} | \psi \rangle) | \psi \rangle \\
&= \text{Tr}(|\psi\rangle \langle \psi| \otimes |\psi\rangle \langle \psi| \text{Swap}) \\
&= \text{Tr}(\rho^{\otimes 2} C_2) \\
&= \text{Tr}(\rho^2).
\end{aligned} \tag{4.26}$$

Note that for any integer $n > 1$ we can calculate the n -th Renyi entropy in this way by simply taking n copies of the density matrix to obtain (4.25)

$$\text{Tr}[\rho^n] = \langle \psi |^{\otimes n} C_n | \psi \rangle^{\otimes n}. \tag{4.27}$$

For the Swap_A gate by separating $|\psi\rangle$ to $|\psi\rangle_A |\psi\rangle_B$ we see that in the trace $\text{Tr}(\rho^{\otimes 2} \text{Swap}_A)$ the subsystem B part is traced out so we get

$$\langle \Psi | \text{Swap}_A | \Psi \rangle = \text{Tr}(\rho_A^2).$$

A short quantum algorithm to calculate $\text{Tr}(\rho_A^2)$ would then be gives as in Figure 4.1.

In this circuit we bring a qubit in a superposition and apply a controlled Swap_A on the two systems $|\psi\rangle |\psi\rangle$. The application of the algorithm yields

$$\begin{aligned}
|0\rangle |\Psi\rangle &\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) |\Psi\rangle \\
&\rightarrow \frac{1}{\sqrt{2}}(|0\rangle |\Psi\rangle + |1\rangle \text{Swap}_A |\Psi\rangle) \\
&\rightarrow \frac{1}{2}((|0\rangle + |1\rangle) |\Psi\rangle + (|0\rangle - |1\rangle) \text{Swap}_A |\Psi\rangle) \\
&\rightarrow \frac{1}{2}(|0\rangle (|\Psi\rangle + \text{Swap}_A |\Psi\rangle) + |1\rangle (|\Psi\rangle - \text{Swap}_A |\Psi\rangle)).
\end{aligned} \tag{4.28}$$

After measuring the first qubit we get the probabilities

$$\mathbb{P}_{|0\rangle} = \frac{1 + \langle \Psi | \text{Swap}_A | \Psi \rangle}{2} \tag{4.29}$$

$$\mathbb{P}_{|1\rangle} = \frac{1 - \langle \Psi | \text{Swap}_A | \Psi \rangle}{2}. \tag{4.30}$$

Therefore after repeated measurements we can estimate the probabilities and thus the value of

$$\langle \Psi | \text{Swap}_A | \Psi \rangle = \text{Tr}(\rho_A^2).$$

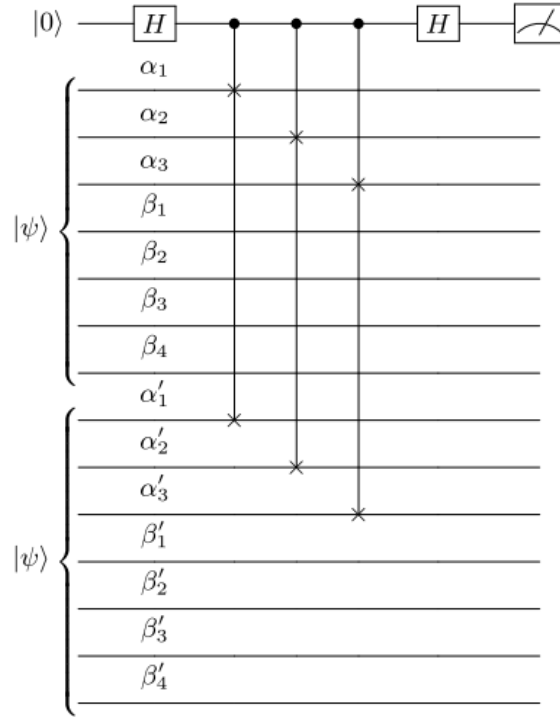


Figure 4.1: Quantum circuit that applies a controlled Swap gate on a subsystem of two equal states. Image taken from [18]

This can be used to calculate $R_2 = -\log(\text{Tr}(\rho_A^2))$. By the Central Limit Theorem this converges to the mean probability with an accuracy of $\epsilon \sim 1/\sqrt{N}$, where N is the number of measurements we have made. So for a given accuracy $\epsilon > 0$ we would need $\mathcal{O}(N^2)$ measurements.

This shows that after measuring our resulting state is

$$\frac{1}{\sqrt{2}} (|\Psi\rangle \pm \text{Swap}_A |\Psi\rangle), \quad (4.31)$$

so it is not clear how to extract further information from it anymore. Therefore we have to repeatedly initialize and time evolve the system, it is clear that this method is not an efficient way to calculate the Renyi entropies. However as we will see, there exists a technique called Quantum Amplitude Estimation that can efficiently calculate this probability and only need few measurements as shown

in [18].

4.3 Estimation using QAE

In our previous algorithm the aim was to estimate the value of $c = \langle \Psi | \text{Swap}_A | \Psi \rangle$. If we define the state $|\tilde{\Psi}\rangle = \text{Swap}_A |\Psi\rangle$ This is a problem that can be solved effectively by the Quantum Amplitude Estimation procedure from section 2.5. We can decompose this in the good part and the bad part as we have done before in the QAE procedure

$$|\tilde{\Psi}\rangle = c |\Psi\rangle + \sqrt{1 - c^2} |\Psi^\perp\rangle, \quad (4.32)$$

where $|\Psi\rangle$ is the good part and $|\Psi^\perp\rangle$ the bad part orthogonal to the good part and we want to estimate c . To do this we follow the method used in [18] but now with a slightly different operator \mathbf{Q} . We use the same idea to apply two reflections to get an operator of which the eigenvalues are dependent on c and we can find these eigenvalues by the phase estimation procedure.

The operator O used to do this is defined as

$$O = \text{Swap}_A V, \quad (4.33)$$

where

$$V = \mathbb{I} - 2 |\Psi\rangle \langle \Psi| \quad (4.34)$$

this operator maps $|\Psi\rangle$ to $-|\Psi\rangle$ and all other states to itself. Since applying Swap_A twice equals the identity, the Swap_A operation is indeed a reflection. Swap_A reflects around an axis in the space spanned by $|\Psi\rangle$ and $|\Psi^\perp\rangle$ as shown in Figure 4.2.

The operator O consists of two reflections and thus O itself is a rotation with eigenvalues $-e^{2i\theta}$, where

$$\cos^2(\theta) = \frac{\langle \Psi | \text{Swap}_A | \Psi \rangle + 1}{2}. \quad (4.35)$$

Using the quantum phase estimation procedure we can get the value of the eigenvalue with arbitrarily high precision.

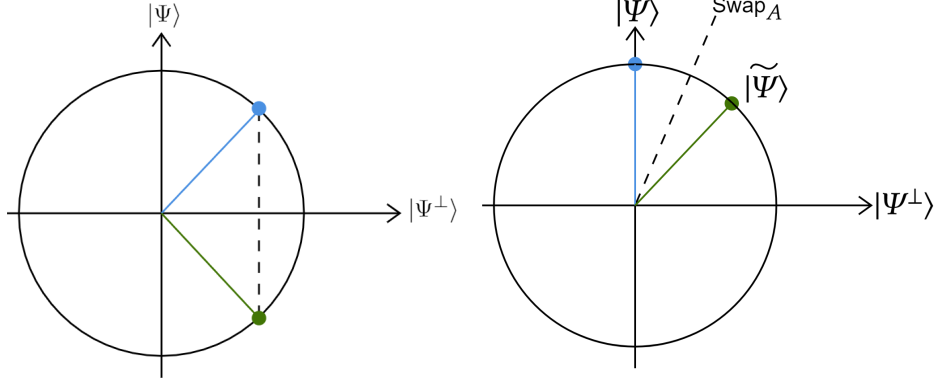


Figure 4.2: Left: Reflection of a state through the $|\Psi^\perp\rangle$ axis. Right: Reflection of a state around the dashed line that represents the Swap_A operation.

We now derive expression (4.35). In matrix-form the operation of the operator on the subspaces $|\Psi\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|\Psi^\perp\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ can be written as

$$V = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (4.36)$$

Its operation in the space spanned by $|\Psi\rangle$ and $|\Psi^\perp\rangle$ is shown in fig. 4.2.

To get the matrix form of the Swap_A gate we calculate its operation on $|\Psi^\perp\rangle$. Since applying the Swap operation twice is the same as the identity mapping we get for $|\Psi^\perp\rangle$

$$\begin{aligned} |\Psi\rangle &= c\text{Swap}_A |\Psi\rangle + \sqrt{1-c^2} \text{Swap}_A |\Psi^\perp\rangle \\ &= c^2 |\Psi\rangle + c\sqrt{1-c^2} |\Psi^\perp\rangle + \sqrt{1-c^2} |\Psi^\perp\rangle \\ (1-c^2) |\Psi\rangle + c\sqrt{1-c^2} |\Psi^\perp\rangle &= \sqrt{1-c^2} \text{Swap}_A |\Psi^\perp\rangle \\ |\tilde{\Psi}^\perp\rangle = \text{Swap}_A |\Psi^\perp\rangle &= \sqrt{1-c^2} |\Psi\rangle + c |\Psi^\perp\rangle. \end{aligned} \quad (4.37)$$

From this we see that the Swap gate is a reflection with matrix

$$\text{Swap}_A = \begin{pmatrix} c & \sqrt{1-c^2} \\ \sqrt{1-c^2} & -c \end{pmatrix}. \quad (4.38)$$

The operator O has as a matrix form

$$O = \begin{pmatrix} -c & \sqrt{1-c^2} \\ -\sqrt{1-c^2} & -c \end{pmatrix}, \quad (4.39)$$

of which the eigenvalues are

$$\begin{aligned} 0 &= \det \begin{pmatrix} -c - \lambda & \sqrt{1-c^2} \\ -\sqrt{1-c^2} & -c - \lambda \end{pmatrix} \\ 0 &= (-c - \lambda)(-c - \lambda) + 1 - c^2 \\ 0 &= \lambda^2 + 2\lambda c + 1 \\ (\lambda + c)^2 &= c^2 - 1 \\ \lambda &= -c \pm i\sqrt{1-c^2}. \end{aligned} \quad (4.40)$$

Since $c \in [0, 1]$ we can write $c = \cos(2\phi)$ for some $\phi \in [0, \pi)$, then the eigenvalues become

$$\lambda_{\pm} = -\cos(2\phi) \pm i\sqrt{1 - \cos(2\phi)^2} = -\cos(2\phi) \pm i\sin(2\phi) \quad (4.41)$$

$$= -(\cos(2\phi) \pm i\sin(2\phi)) = -e^{\pm i2\phi}, \quad (4.42)$$

where $\cos(\phi)^2 = \frac{1+\cos(2\phi)}{2} = \frac{1+\langle\Psi|\text{Swap}_A|\Psi\rangle}{2}$, and the statement in eq. (4.35) has been verified. Note that

$$-\frac{1}{2}(\lambda_{\pm} + \bar{\lambda}_{\pm}) = \frac{1}{2}(e^{\pm 2i\phi} + e^{-\pm 2i\phi}) \quad (4.43)$$

$$= \cos(2\phi) = \langle\Psi|\text{Swap}_A|\Psi\rangle = \text{Tr}[\rho_A^2] \quad (4.44)$$

Thus independent from what eigenvalue we get from the quantum phase estimation we can always calculate the Renyi entropy by adding the complex conjugate.

We see that using QAE we can estimate the value $\langle\Psi|\text{Swap}_A|\Psi\rangle$ with arbitrary precision and using only one initialization of $|\Psi\rangle$. This is a huge improvement compared to the algorithm showed previously in terms of the number of initialization we have to make, but this comes at the price of using more qubits.

4.3.1 Quantum circuit of operator O

We will now show how one can implement the scheme of the QAE algorithm into gates to apply on a quantum computer, where we follow the implementation as it was done in [18]. The main part of the quantum circuit has to do with the implementation of the reflection operator V in O , since the Swap_A operator consists of 3 CNOT gates its implementation is easy to do. The operator O is controlled implemented on an ancillary qubit, since we will use the operator for quantum phase estimation.

The method used to prepare the reflection V is similar to the $\mathcal{A} S_0 \mathcal{A}^{-1}$ operator that only reflects the state $|\Psi\rangle = \mathcal{A} |0^{2n}\rangle$. This operator first 'unpacks' the state $|\Psi\rangle$ to the all zero-state then sends only $|0^{2n}\rangle \rightarrow -|0^{2n}\rangle$, then performing a controlled Z -gate on the first qubit, where this gate is controlled on all other qubits being in the $|0\rangle$ state. Then we bring the zero-state back to the state $|\Psi\rangle$. This procedure can be made more general, if we have an operator Q of which $|\Phi\rangle$ is an eigenstate with known eigenvalue q_Ψ . Setting up the Fourier Transform of the eigenvalue $|\mathcal{F}[q_\Psi]\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \phi k} |k\rangle$, where $\phi \in [0, 1)$ is such that $q_\Psi = e^{2\pi i \phi}$. We then do a reverse application of the Quantum Phase Estimation procedure as shown in section 1 and bring $|\Psi\rangle |\mathcal{F}[q_\Psi]\rangle$ to $|\Psi\rangle |0^m\rangle$, where m is the number of qubits we use for the Fourier Transform of q_Ψ . The quantum circuit for this procedure is shown in fig. 4.3

We will show how the quantum circuit for the desired input state Ψ that we use. Firstly note that if $Q|\Psi\rangle = q_\Psi|\Psi\rangle$, then $Q^\dagger|\Psi\rangle = \frac{1}{q_\Psi}|\Psi\rangle$, since Q is unitary. Now the controlled implementation of the operator O works as follows, for the

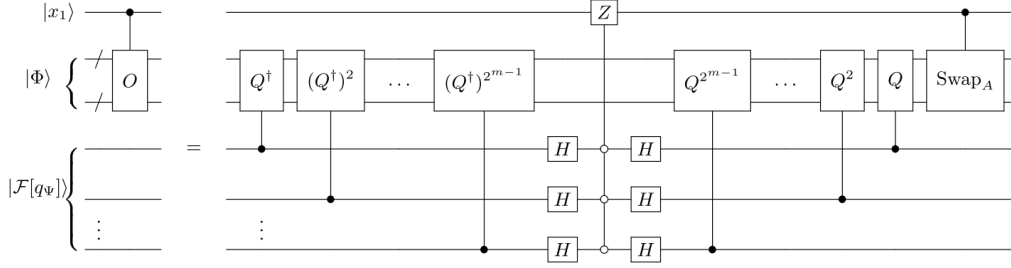


Figure 4.3: Quantum circuit of the controlled implementation of the operator $O = \text{Swap}_A V$, where $V = 1 - 2|\Psi\rangle\langle\Psi|$. The operator Q has the desired state $|\Psi\rangle$ as an eigenstate and the corresponding eigenvalue q_Ψ is known beforehand. The Fourier Transform of the eigenvalue is stored in ancillary qubits, and the inverse of the Quantum Phase Estimation procedure is applied. The open balls in the controlled Z gate mean that the gate is controlled on the $|0\rangle$ state instead of the $|1\rangle$ state.

desired state input state $|\Psi\rangle$ and an ancillary qubit $|x\rangle$:

$$\begin{aligned}
|x\rangle|\Psi\rangle|\mathcal{F}[q_\Psi]\rangle &= |x\rangle|\Psi\rangle\sum_{k=0}^{2^m-1}e^{2\pi i\phi k}|k\rangle \\
&\xrightarrow{C-(Q^\dagger)^k}|x\rangle\sum_{k=0}^{2^m-1}(Q^\dagger)^k|\Psi\rangle e^{2\pi i\phi k}|k\rangle \\
&= |x\rangle|\Psi\rangle\sum_{k=0}^{2^m-1}\frac{1}{q_\Psi^k}e^{2\pi i\phi k}|k\rangle \\
&= |x\rangle|\Psi\rangle\sum_{k=0}^{2^m-1}e^{-2\pi i\phi k}e^{2\pi i\phi k}|k\rangle = |x\rangle|\Psi\rangle\sum_{k=0}^{2^m-1}|k\rangle \\
&\xrightarrow{H}|x\rangle|\Psi\rangle|0^m\rangle \\
&\xrightarrow{C-Z}(-1)^x|x\rangle|\Psi\rangle|0^m\rangle \\
&\xrightarrow{C-(Q)^k}(-1)^x|x\rangle|\Psi\rangle|\mathcal{F}[q_\Psi]\rangle \\
&\xrightarrow{C-\text{Swap}_A}(-1)^x|x\rangle\text{Swap}_A^x|\Psi\rangle|\mathcal{F}[q_\Psi]\rangle,
\end{aligned}$$

so we see that the circuit works as expected for the desired input state $|\Psi\rangle$. For some general state input $|\Phi\rangle$ only its overlap with the $|\Psi\rangle$ would be reflected,

while the rest would remain the same. Writing out a general state in the overlapping part with $|\Psi\rangle$ and a non-overlapping part we get $|\Phi\rangle = c_\Psi |\Psi\rangle + \sum_i c_i |\phi_i\rangle$, where $|\phi_i\rangle$ span the basis of $|\Psi^\perp\rangle$. Then applying the quantum circuit the overlapping part would be controlled reflected while the rest stays the same so

$$|x\rangle |\Phi\rangle |\mathcal{F}[q_\Psi]\rangle = |x\rangle (c_\Psi |\Psi\rangle + \sum_i c_i |\phi_i\rangle) |\mathcal{F}[q_\Psi]\rangle \quad (4.45)$$

$$\longrightarrow |x\rangle \left((-1)^x c_\Psi |\Psi\rangle |0^m\rangle + \sum_{i \neq 0} c_i |\phi_i\rangle |i\rangle \right) \quad (4.46)$$

$$= |x\rangle \left((1 - 2x) c_\Psi |\Psi\rangle |0^m\rangle + \sum_{i \neq 0} c_i |\phi_i\rangle |i\rangle \right) \quad (4.47)$$

$$\longrightarrow |x\rangle (|\Phi\rangle - 2x c_\Psi |\Psi\rangle) |\mathcal{F}[q_\Psi]\rangle \quad (4.48)$$

$$\longrightarrow |x\rangle \text{Swap}_A^x (|\Phi\rangle - 2x c_\Psi |\Psi\rangle) |\mathcal{F}[q_\Psi]\rangle. \quad (4.49)$$

With the controlled operator written as a quantum circuit we can now apply the Quantum Phase Estimation Algorithm to estimate the eigenvalue of the operator O to an arbitrarily high precision $\epsilon > 0$ using only one initialization of $|\Psi\rangle$ and $\mathcal{O}(\log(1/\epsilon))$ qubits to store its value.

We apply the same circuit for the Quantum Phase Estimation Algorithm as in figure 2.4 to estimate the eigenvalues of operator O . Note that compared to the algorithm in fig 4.1 we now use many more ancillary qubits, we need $\mathcal{O}(1/\epsilon)$ qubits to store the eigenvalue of O and we need to store the Fourier Transform of the eigenvalue of Q . This trade-off is justified because the new algorithm only needs one initialization of $|\Psi\rangle$. However finding an operator Q of which $|\Psi\rangle$ is an eigenstate is not an easy task because we don't know the full wave function of $|\Psi\rangle$ (if we did we could classically calculate $\text{Tr}[\rho_A^2]$).

In practice the only operator Q of which we know $|\Psi\rangle$ to be an eigenstate and of which we know the eigenvalue is to reverse the unitary initialization of $|\Psi\rangle$ and do some operation on the original state and then initialize $|\Psi\rangle$ again. This is in fact the regular Quantum Amplitude Estimation algorithm. This does have the advantage that we can choose the eigenvalue of the operator Q ourselves, and by choosing $q_\Psi = -1$ as an eigenvalue we only need 1 qubit to store its Fourier Transform. Also the implementation of the phase estimation of the operator O after we have created a physical quantum circuit of O should not be difficult. Performing O^{2^j} just means we loop the qubits in the same circuit 2^j times.

4.4 Entanglement Entropy Simulation in ProjectQ

We've now seen two algorithms that can estimate the value of $\text{Tr}[\rho_A^2]$ from two copies of the state $|\psi\rangle$. In this section we present an implementation of both algorithms using ProjectQ, an open source software framework for quantum computing in Python [19]. The idea of ProjectQ is that it is a compiler framework capable of compiling a quantum algorithm to a quantum simulator or various types of hardware. In our case we will simulate the algorithms using a quantum simulator that keeps track of the full wave function.

4.4.1 Swap Test

The implementation of the first algorithm using the Swap test is as follows

Algorithm 3: Swap Test Procedure to estimate the entanglement entropy between two subsystems A and B

```

1 function SwapTest ( wavefunction, A, B)
  Inputs      : The coefficients of the states in the initial wave function,
                  Subsystem A, Subsystem B, Number of runs Nruns
  Outputs    : Approximation of  $\text{Tr}[\rho_A^2]$ 
2 for  $i = Nruns$  to  $i = 0$  do
3    $|qAB\rangle \leftarrow \text{INITIALIZE}(\text{wavefunction})$ ;           // initial state
4    $|qAB2\rangle \leftarrow \text{INITIALIZE}(\text{wavefunction})$ ;       // initial state
5    $|0\rangle \leftarrow |q0\rangle$ ;                                 // initialize ancilla
6    $H(|q0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ;           // Hadamard gate
7   if  $|q0\rangle$  then
8      $\text{Swap}_A(|qAB\rangle, |qAB2\rangle)$ ;                       // Control Swap-gate
9      $q0 \leftarrow |q0\rangle$ ;                                 // Measure ancilla
10 return  $(\#(q0 = 0) - \#(q0 = 1))/Nruns$ 

```

We've used the conventions for quantum pseudocode from [5].

In fig. 4.4 the estimation of $\text{Tr}[\rho_A^2]$ is plotted against the number of runs. The value of $\text{Tr}[\rho_A^2]$ is calculated from (4.29)

$$\text{Tr}[\rho_A^2] = \mathbb{P}_{|0\rangle} - \mathbb{P}_{|1\rangle}, \quad (4.50)$$

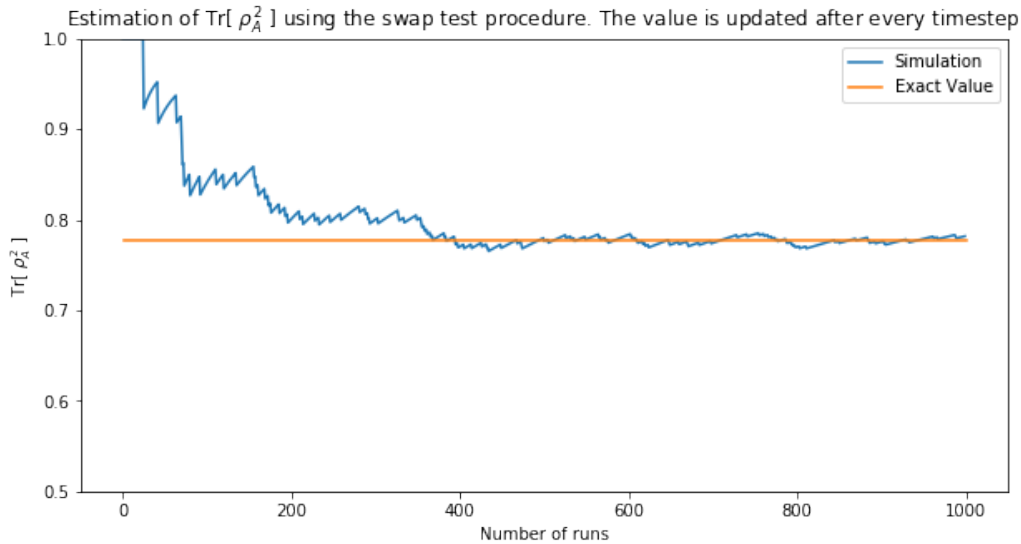


Figure 4.4: Output of the implementation of Algorithm 3 in ProjectQ. The value of the estimation of $\text{Tr}[\rho_A^2]$ is plotted for the state $|\psi\rangle = \frac{|00\rangle + |10\rangle + |01\rangle}{\sqrt{3}}$, note that after ~ 400 runs the estimation has converged to around the true value.

the estimation of the probability of measuring $|0\rangle$ or $|1\rangle$ is updated after every run. The convergence of this probabilistic scheme is not very fast, especially considering for every time step we need to initialize the state $|\Psi\rangle$ again.

In the simulator every gate and operation is implemented as an operation on the wave function and during the simulation it is possible to access the full wave function. This means that we can 'cheat' in our simulations to make them quicker. We could, for example, directly access the probability of measuring $|0\rangle$ or $|1\rangle$ in the first algorithm 4.28. In ProjectQ we implement this by accessing the backend:

```

1 eng.backend.get_probability('0', q0) -
2   eng.backend.get_probability('1', q0)

```

This returns a value of 0.777777777777775295, which coincides with our calculation of $\frac{7}{9}$. Note that these operations are not possible to do on a real quantum computer and are *only* done to improve the speed of the algorithm, by applying enough runs we could estimate this value in the same way.

4.4.2 Quantum Amplitude Estimation

The implementation for the Quantum Amplitude Estimation scheme is as follows

Algorithm 4: Quantum Amplitude Estimation procedure to estimate the entanglement entropy between two subsystems A and B

```

1 function QuantumAmplitudeEstimation(wavefunction, A, B, O, d)
   Input : wavefunction, subsystemA, subsystemB, OperatorO, precision
   Output: Approximation of  $\text{Tr}[\rho_A^2]$ 
2  $|qAB\rangle \leftarrow \text{Initialize}(\text{wavefunction})$ ; // initial state
3  $|qAB2\rangle \leftarrow \text{Initialize}(\text{wavefunction})$ ; // initial state
4  $|q0\rangle \leftarrow |0^d\rangle$ ; // initialize ancillas
5  $\mathcal{H}(|q0\rangle)$ ; // bring ancilla in superposition
6 for  $i = \text{precision} - 1$  to  $i = 0$  do
7     if  $|q0[i]\rangle$  then
8          $O^{2^i}(|qAB\rangle, |qAB2\rangle) \leftarrow |qAB\rangle, |qAB2\rangle$ ; // controlled  $O$  in the fourier
           space
9  $q0 \leftarrow \text{MeasuredFourier}(|q0\rangle, d)$  return  $q0$ 

```

This function outputs the values of the ancilla qubits used in the quantum phase estimation algorithm. These qubits are measured in the computational basis, and are the value of the angle ϕ of the eigenvalue of O namely $\lambda_{\pm} = e^{2\pi i\phi}$. By adding the complex conjugate as done in (4.43) we get the value of $\text{Tr}(\rho_A^2)$

$$\frac{1}{2} (\lambda_{\pm} + \bar{\lambda}_{\pm}) = \text{Tr}(\rho_A^2). \quad (4.51)$$

From the implementation in ProjectQ in fig 4.5 we see for our example of the state $|\psi\rangle = \frac{|00\rangle + |10\rangle + |01\rangle}{\sqrt{3}}$ that the convergence of the estimation of $\text{Tr}(\rho_A^2)$ to the analytical value $\frac{7}{9}$ goes very quickly, and we only need to initialize $|\Psi\rangle$ once. We do need extra ancillary qubits. If we would have a quantum computer with a large amount of qubits this would not be a problem, especially since the amount of ancillary qubits only depends on the precision of the estimation, i.e. a very large and complex state requires the same number of ancillary as our small problem if we want equal precision in its estimation. However in our simulator this can be a problem, since we can only simulate 25 qubits on a regular computer. To implement this procedure we would lose relatively many qubits compared to

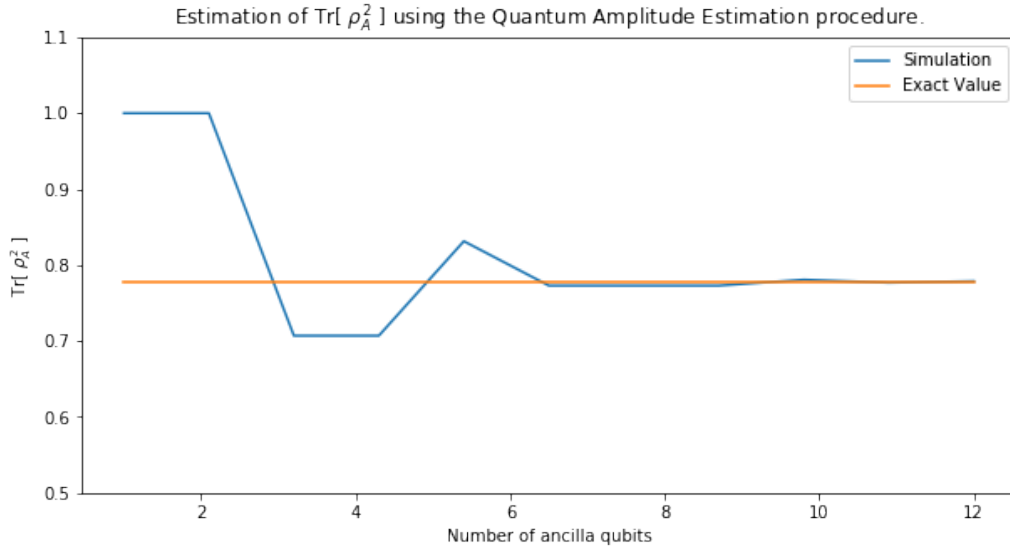


Figure 4.5: Output of the implementation of Algorithm 4.3 in ProjectQ. The value of the estimation of $\text{Tr}[\rho_A^2]$ is plotted for the state $|\psi\rangle = \frac{|00\rangle + |10\rangle + |01\rangle}{\sqrt{3}}$. The convergence to the exact value of $\frac{7}{9}$ converges exponentially good in the number of ancilla used.

how much we can use for the state we are interested in.

The implementations of these functions in ProjectQ can be used to access the entanglement entropy for every wave function. In the algorithms that follow we will perform operations on quantum states and use these algorithms to access the wave functions. As mentioned before we are simulating a quantum computer and it is possible to save the entire wave function, something that is not possible on a real quantum computer. Therefore to make the simulations more efficient we will not use the Quantum Amplitude Estimation procedure, but we will do the Swap test and cheat to get the probabilities.

Chapter 5

Numerical Simulations of Localization

An important statement in classical statistical physics is the concept of thermalization. One of the statements we can make is that systems over time converge to their thermal equilibrium, this is due to the ergodicity theorem that over time every microstate becomes equally likely to be reached. In this chapter we will extend the concept of thermalization to quantum states.

There are physical systems that do not thermalize over time. We will introduce the Heisenberg XXZ model in different configurations that are known to exhibit localization instead of thermalization. We will introduce a model called the supersymmetric Hamiltonian for which we will investigate if there is localization over time.

The method used to find localization in these systems is a combination of the previous chapters. We will time evolve an initial state under the Hamiltonian and calculate the entanglement entropy at every time step. The entanglement holds crucial information for the characterization of different forms of localization.

5.1 XXZ Model

In this section we will introduce and the one-dimensional spin-1/2 Heisenberg XXZ model in a magnetic field. This model has been studied in great detail and we will use it to reference our findings with known results. Our goal is to simulate

the time evolution of this system. The XXZ model is an instance of the general 1d spin-1/2 Heisenberg model which for a lattice of size N (where the first site has label 1) can be written in the form

$$H = \sum_{j \in \{x,y,z\}} J_j \sum_{i=1}^N \sigma_i^j \sigma_{i+1}^j + h \sum_i \sigma_i^z. \quad (5.1)$$

The model describes nearest neighbor interactions in a spin chain, where the interactions are Pauli operators on adjacent lattice sites with corresponding interaction strength J_j . The entire lattice is placed in a magnetic field with strength h . We can introduce periodic or open boundary conditions by setting $\sigma_{N+1} = \sigma_1$ or $\sigma_{N+1} = 0$ respectively. Note that often the Hamiltonian is defined as minus our definition, this coincides with whether we see the completely aligned state as the state with minimal or maximal energy. In our setup we define it as the state with maximal energy, alas the plus sign in our definition.

The Heisenberg XXZ model is an instance of the general model in which $J = J_x = J_y \neq J_z = \Delta$. In the following parts we will modify the model by allowing the magnetic field to be random at every lattice site denoted by h_i where the field strength h_i will be chosen uniformly from the interval $[-h/2, h/2]$ for some h . Setting open boundary conditions we then get

$$H = J \sum_{i=1}^{N-1} \sigma_i^x \sigma_{i+1}^x + \sigma_{i+1}^y \sigma_i^y + \Delta \sum_{i=1}^{N-1} \sigma_i^z \sigma_{i+1}^z + \sum_{i=1}^N h_i \sigma_i^z. \quad (5.2)$$

Since this model is already written in terms of Pauli operators we can immediately apply it to the quantum programming framework of ProjectQ as mentioned in section 3.1.4. But for the analysis in the upcoming sections it is useful to write down the XXZ model in second quantized form. Filling in the second quantized version of the product of neighboring Pauli operators was derived in equation (3.43) we get

$$H = 2J \sum_{i=1}^{N-1} (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i) + \Delta \sum_{i=1}^{N-1} n_i n_{i+1} - 2 \sum_{i=1}^N h_i n_i \quad (5.3)$$

where n_i denotes the counting operator $c_i^\dagger c_i$ and we have omitted the constant terms and constant terms dependent on the number of particles. These terms are not interesting in the time evolution of the operator since they only add a

global phase to e^{-iHt} . We see that the J term corresponds to a fermionic hopping strength and Δ corresponds to nearest neighbor particle interaction strengths.

When we implement the time operator note that in this operator e^{-iHt} a general coefficient of the Hamiltonian can be re-scaled in the time step. We will therefore set the coefficient $J = 1/2$, leaving the labels of the other coefficients the same for convenience. We can also omit the minus sign in the random magnetic field strength since we draw the coefficients uniformly symmetric around 0. The Hamiltonian then becomes

$$H = \sum_{i=1}^{N-1} (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i) + \Delta \sum_{i=1}^{N-1} n_i n_{i+1} + \sum_{i=1}^N h_i n_i, \quad (5.4)$$

where now we draw h_i from $[-h, h]$. From now on we will use the XXZ model in this form.

5.2 Supersymmetric Hamiltonian

In this section the supersymmetric lattice structure is introduced. The supersymmetric model describes spinless fermions on a one dimensional lattice of size N . We say a Hamiltonian is $\mathcal{N} = 2$ *supersymmetric* if it can be written as the anticommutator of two nilpotent fermionic generator Q, Q^\dagger [2]. In our model the fermionic generators Q and Q^\dagger , also called *supercharges*, are defined as

$$Q^\dagger = \sum_{i=1}^N (1 - n_{i-1}) c_i^\dagger (1 - n_{i+1}) \quad (5.5)$$

$$Q = \sum_{i=1}^N (1 - n_{i-1}) c_i (1 - n_{i+1}), \quad (5.6)$$

where as in the XXZ model we can impose periodic or open boundary conditions. Setting $n_0 = n_{N+1} = 0$ or $n_0 = n_{N+1}$ we get open or closed boundaries. These operators are nilpotent, $Q^2 = 0$ and $(Q^\dagger)^2 = 0$, see Appendix A.1. What both operators do when applied to a state is to add or take away a particle if its adjacent sites are empty. The corresponding Hamiltonian

$$H = \{Q, Q^\dagger\}, \quad (5.7)$$

therefore always leaves the amount of particles fixed. Supersymmetric Hamiltonians are always positive semi-definite, using that the superoperators are nilpotent we can show this

$$H = QQ^\dagger + Q^\dagger Q \quad (5.8)$$

$$= QQ^\dagger + Q^\dagger Q + QQ + Q^\dagger Q^\dagger \quad (5.9)$$

$$= (Q + Q^\dagger)(Q^\dagger + Q) = YY^\dagger, \quad (5.10)$$

where $Y = (Q + Q^\dagger)$ has $\det(Y) \neq 0$, thus H is positive semi-definite. Therefore the energy E of any state $H|\psi\rangle \geq 0$. Supersymmetric Hamiltonians can also be applied on higher level structures instead of chains and be implemented with other supercharges. However these are not within the scope of this thesis. We will further investigate the Hamiltonian with open boundary conditions introduced before and implement its time evolution in the quantum simulator framework of ProjectQ.

By using the anticommutation relations the Hamiltonian can be rewritten to

$$\sum_{i=1}^N (1 - n_{i-1})(c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i)(1 - n_{i+2}) + \sum_{i=1}^N (1 - n_{i-1})(1 - n_{i+1}), \quad (5.11)$$

the derivation of this expression is given in Appendix A.2. This statement has some parallels with the XXZ-model in equation (5.4). We can have hopping to nearest neighbors if the original site has no neighboring particles and the target state also does not have one. The second term can be expanded

$$\sum_{i=1}^N (1 - n_{i-1})(1 - n_{i+1}) = N - 2F + \sum_{i=1}^N n_{i-1}n_{i+1}, \quad (5.12)$$

where F is the amount of fermions in the system. We see next to nearest neighbor interactions instead of nearest neighbor interactions in the XXZ model.

5.3 Numerical Simulations

In this section we will show the results of the numerical implementation of the before mentioned Hamiltonians. The XXZ Hamiltonian and supersymmetric Hamiltonian in the ProjectQ framework using the method described in section

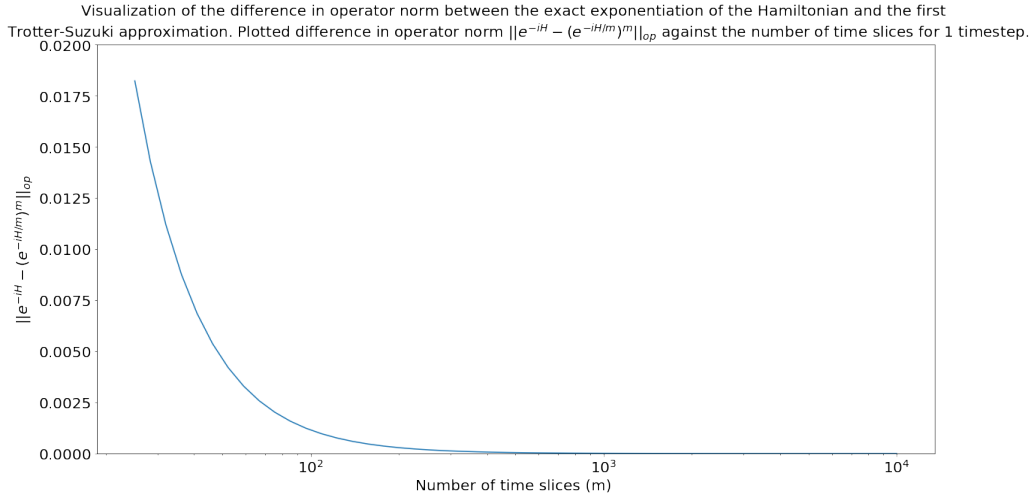


Figure 5.1: Difference in operator norm of the first Suzuki-Trotter approximation and with the direct exponentiation of the Hamiltonian. Hamiltonian used is a 8 site supersymmetric Hamiltonian as introduced in 5.12.

3.1.4. A url to the code can be found here [A.4](#). These Hamiltonians will always have non-commuting terms since different Pauli operators don't commute. As shown in equation (3.17) the error of the approximation should be inversely proportional to the number of slices we divide the time step. In figure 5.1 this result is has been calculated by comparing the operator norm of the direct matrix exponentiation with the circuit we get from OpenFermion.

We see that the error is inversely proportional to the number of splices we made of our time step. This was expected but it is a good confirmation that we can indeed simulate this model.

To visualize what action the time evolution of these Hamiltonians has on a basis state we have plotted the probability density of measuring a state in a certain instance of a configuration with $J = 1, \Delta = 1, h = 1$. This visualization is shown in figure 5.2. In the plot a 1 in the ket means the fermion is spin up, a 0 means spin down. We initialize the state in the $|001\rangle$ state and use open boundary conditions. We see that the number of spin up states is not conserved through time, after some time the $|111\rangle$ is a possible state to be measured, this means that in this instance we had 3 random magnetic field strengths that pointed in the

Probability density function of a single fermion in the starting state $|001\rangle$ evolved through time in the XXZ with $J = 1 = \Delta$, $h = 1$ model.

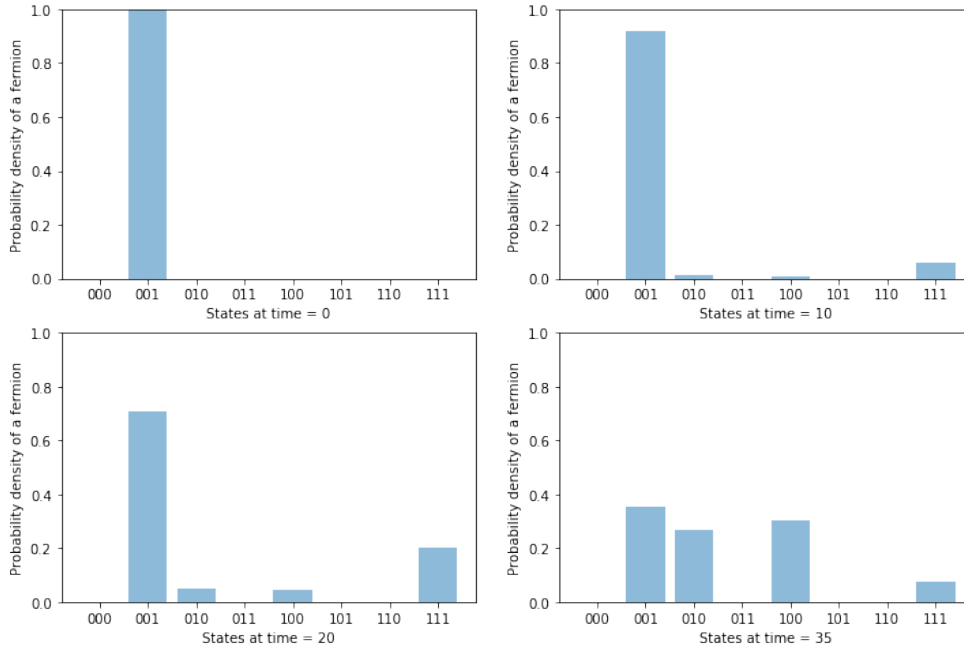


Figure 5.2: Visualization of the probability density at different instances in time of a particle initialized as $|001\rangle$ and time evolved under the XXZ model as described in equation (5.4). A 1 means the fermion has spin up, a zero spin down.

same direction. The visualization shows clearly what we would expect, we have hopping to nearest neighbors and boundary effects.

In figure 5.3 the time evolution of the initial state $|001\rangle$ is plotted. This result shows exactly what we would expect from the Hamiltonian in equation (5.12). After the initialization of a fermion at the third lattice site we see that it moves to the middle state, and only after it it can also move to the most left state.

As explained in the previous chapter an important quantity in physics that we can also measure on a quantum computer is the second Renyi entanglement entropy. The largest chain that we can simulate effectively on a normal CPU is of size 8, so in the remainder our plots will be of lattices of this size.

Probability density function of a single fermion in the starting state $|001\rangle$ evolved through time in the supersymmetric model.

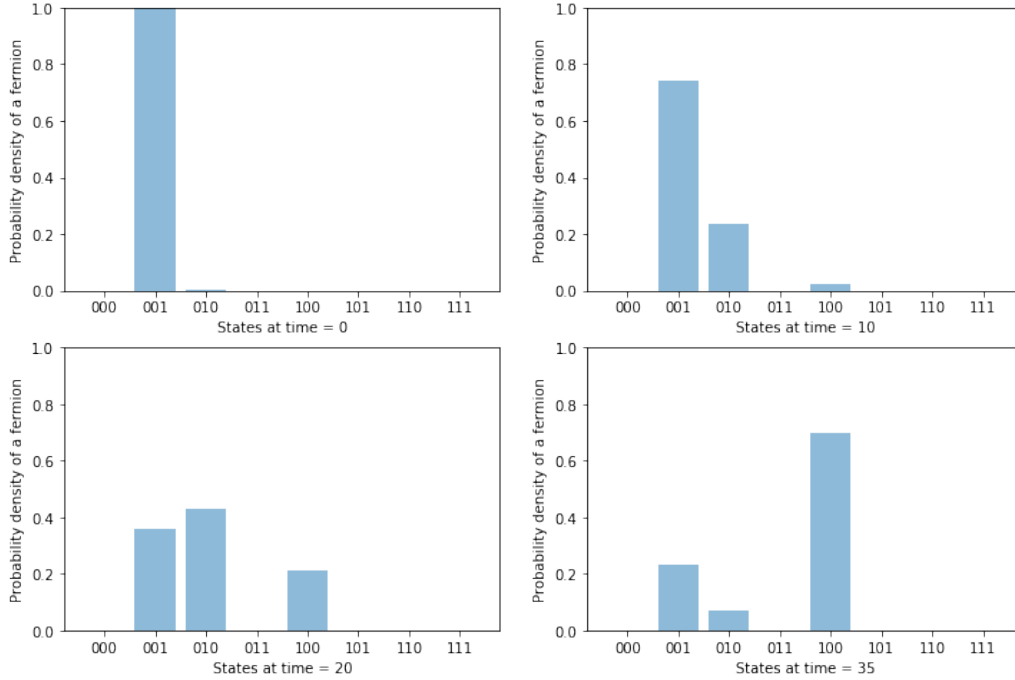


Figure 5.3: Visualization of the probability density at different instances in time of a particle initialized as $|001\rangle$ and time evolved under the supersymmetric Hamiltonian model as described in equation (5.12). A 1 means we have a fermion at that site in the lattice, a 0 means it is empty.

In figure 5.4 the time evolution of the entanglement entropy was plotted. The state $|01000010\rangle$ was initialized in the ProjectQ framework and time evolved by the method described in algorithm 2 and after every time step the second Renyi entropy was calculated by algorithm 3 where we cheated by looking at the underlying wave function and calculated the probabilities of measuring the $|0\rangle$ or $|1\rangle$ ancilla with high probability.

The time evolution of the second Renyi entropy in figure 5.4 gives a measure on how entangled the two subsystems A , B are evolved through time. A and B both consist of 4 qubits and they describe the first and second half of the system. Initially there is no entanglement in state $|01000010\rangle$, then when we

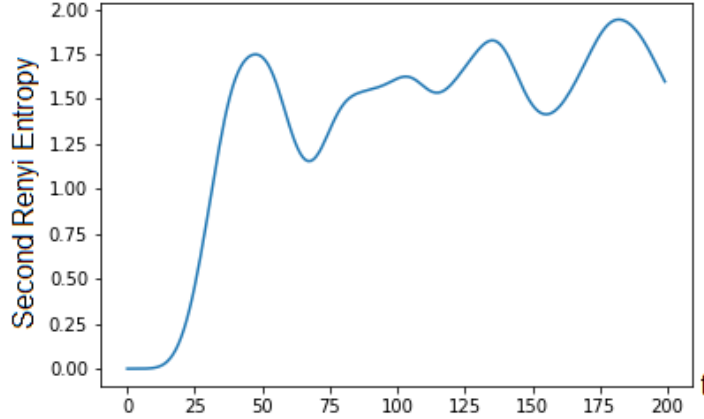


Figure 5.4: Quantum simulation of the evolution of the entanglement entropy of the state initial state $|01000010\rangle$ evolved under the supersymmetric Hamiltonian. Plotted on the x-axis are the time steps of size 1 against the second Renyi entropy, where the system was divided into two subsystems A, B consisting of the 4 qubits on the left and right. The Swap test procedure from Algorithm 3 was used here we cheated to get high precision values.

time evolve the state the fermions start to move probabilistically to neighboring states similar to as shown in figure 5.3 and we get a superposition of new states. Then whenever a particle crosses the boundary between subsystems A and B we build up entanglement in the state. We see that the entanglement entropy increases quickly and then remains high, this fact that it stops growing comes from the finite lattice size and the finite amount of fermions. This gives a saturation bound on the entanglement entropy, for a infinite lattice we expect the entanglement entropy to keep increasing. The behavior of fast entanglement growth to a saturation value coincides with the thermalization of states.

5.4 Eigenstate Thermalization Hypothesis

So far we have seen how we can time evolve general fermionic states under a Hamiltonian and we have calculated the entanglement entropy at every time step of this evolution. In this section we will show how the entanglement entropy is an indicator of the phenomena of *Anderson Localization (AL)* and *Many Body*

Localization (MBL) in condensed matter physics.

First we discuss thermalization in quantum systems. In classical statistical mechanics thermalization is the process in which a system reaches thermal equilibrium with maximal entropy. This description relies on the ergodicity theorem, that over time all microstates of the system are reached with equal probability. In quantum systems we cannot extend this reasoning because not all states with equal energy E are reached. In the time evolution of some state $|\psi(0)\rangle = \sum_{\alpha} A_{\alpha} |E_{\alpha}\rangle$, where $|E_{\alpha}\rangle$ are the eigenstates of the Hamiltonian the coefficients of the eigenstates only acquire a phase

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle = \sum_{\alpha} e^{-iE_{\alpha}t} A_{\alpha} |E_{\alpha}\rangle. \quad (5.13)$$

The probability of measuring the state in an eigenstate $|E_{\alpha}\rangle$ is $p_{\alpha} = |A_{\alpha}|^2$ will therefore not change over time and depend completely on the initial state. In contrast to classical states, which does explore different states through the state space by ergodicity.

We see that we need to develop a new notion of ergodicity to explain what thermalization means in a quantum many-body setting, a notion for this has been introduced in [20]. By comparison the classical microcanonical ensemble we want that starting from a physical initial state that the observables of the system reach the values of the microcanonical ensemble after infinite time. The expectation of an observable \hat{O} at infinite time is

$$\langle \hat{O} \rangle_{t \rightarrow \infty} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^{\infty} \langle \psi(t) | \hat{O} | \psi(t) \rangle dt \quad (5.14)$$

$$= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^{\infty} \sum_{\alpha, \beta} A_{\alpha}^{\dagger} A_{\beta} e^{-i(E_{\beta} - E_{\alpha})t} \langle E_{\alpha} | \hat{O} | E_{\beta} \rangle \quad (5.15)$$

$$= \sum_{\alpha} p_{\alpha} \langle E_{\alpha} | \hat{O} | E_{\alpha} \rangle, \quad (5.16)$$

where in the last line we used that the off diagonal terms oscillate at different frequencies and average out over long time [1]. Again we have the p_{α} are set by the initial state, so the way to define thermalization to the equilibrium is by assuming that the expectation values of the eigenstate and the operator $\langle E_{\alpha} | \hat{O} | E_{\alpha} \rangle$ are the as in the microcanonical ensemble $O_{mc}(E_{\alpha})$.

The explanation for thermalization, that in ergodic quantum many body systems the eigenstates have thermal observables identical to the microcanonical ensembles was introduced in [21] and [20], and is known as the Eigenstate Thermalization Hypothesis (ETH). While all known thermalizing systems obey ETH, it is not known whether ETH is necessary for thermalization [1].

An implication of ETH is that all observables within a subsystem A will have thermal expectation values. This implies that the reduced density matrix ρ_A is thermal and therefore the entanglement entropy is equal to the thermodynamic entropy of A [1]. It is known of the thermodynamic entropy that this grows linearly in time [22]. The growth of the entanglement in the time evolution of a system will be used to characterize its characteristics.

5.5 Anderson and Many Body Localization

Anderson localization also known as strong localization was introduced by Philip Anderson in 1958 [23]. He proposed a lattice model with strong random disorder, these disorders can be impurities defects in the lattice or strong different magnetic fields at each lattice point. What one would expect is after initializing a particle in such a lattice due to the disorder the wave function will scatter and its localization will diffuse on the lattice and the wave function will extend throughout the whole system. However what Anderson showed is that if the disorder is large enough the interference with particles and the disorder completely halt the particle and particles become exponentially localized in finite regions of space [24], and interactions become negligible. The concept of Anderson localization is important in the understanding of insulators and the transitions between insulating phases and metallic phases.

An indication of the occurrence of Anderson localization can be found in the entanglement entropy. When a system becomes localized the entanglement entropy will not grow anymore in time. We expect that in the occurrence of Anderson localization in a finite lattice such as in figure 5.4 after initialization in a pure state the entanglement will grow from 0 to some value higher value and remain constant through time. This value will be much lower than the saturation value of the entanglement entropy.

It may seem that Anderson Localization is a non-thermalizing system since by the localization the approach to equilibrium is not possible. Indeed theoretically an isolated Anderson Localized system is an insulator and has zero conductivity. However practically we will still have interactions in our system and it was Anderson himself who showed that it is possible by the decay of a single localized particle for the whole system to become delocalized again and break the Anderson Localization [25]. This brings the question whether there are systems that break ETH while allowing interactions.

It turns out such systems exist. In recent years the existence of localized phases at non-zero temperatures have been shown in low-dimensional disordered systems [26]. These systems that remain a perfect insulator while having interactions are called *Many Body Localized* (MBL) systems. In contrast to Anderson localization these systems remain robust within interactions and small perturbations. MBL systems are the only known robust systems that do not thermalize in a closed system [1]. An explanation to why MBL systems remain localized is the emergence of an integrable set of quasilocal integrals of motion (LIOMs), which by integrability remain during the dynamics of the system and prevent thermalization. A key characteristic of MBL is that the entanglement entropy of the system will grow logarithmically in time [1], this is a very unique characteristic that we will use in the remainder of this thesis.

There have been numerical simulations [3] using the Heisenberg XXZ model

$$H = \sum_{i=1}^{N-1} (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i) + \Delta \sum_{i=1}^{N-1} n_i n_{i+1} + \sum_{i=1}^N h_i n_i, \quad (5.17)$$

which under certain configurations of large enough disorder in the magnetic field showed Anderson Localization and Many Body Localization. The simulations can be seen in figure 5.5.

In the plot as an initial state the Néel state was chosen. The lattice size was $N = 50$, and plotted are two different Renyi entanglement entropies R_1 and $R_{0.5}$ of the same Hamiltonian initialization. Two different Hamiltonian initializations were made: the random coefficients were chosen from the interval $[-5, 5]$ and two values of Δ were chosen. $\Delta = 0$ (saturating curves) and $\Delta = 0.5$ (growing curves). The data perfectly coincides with the predictions made earlier, we would expect Anderson localization in the non-interacting case and indeed the entan-

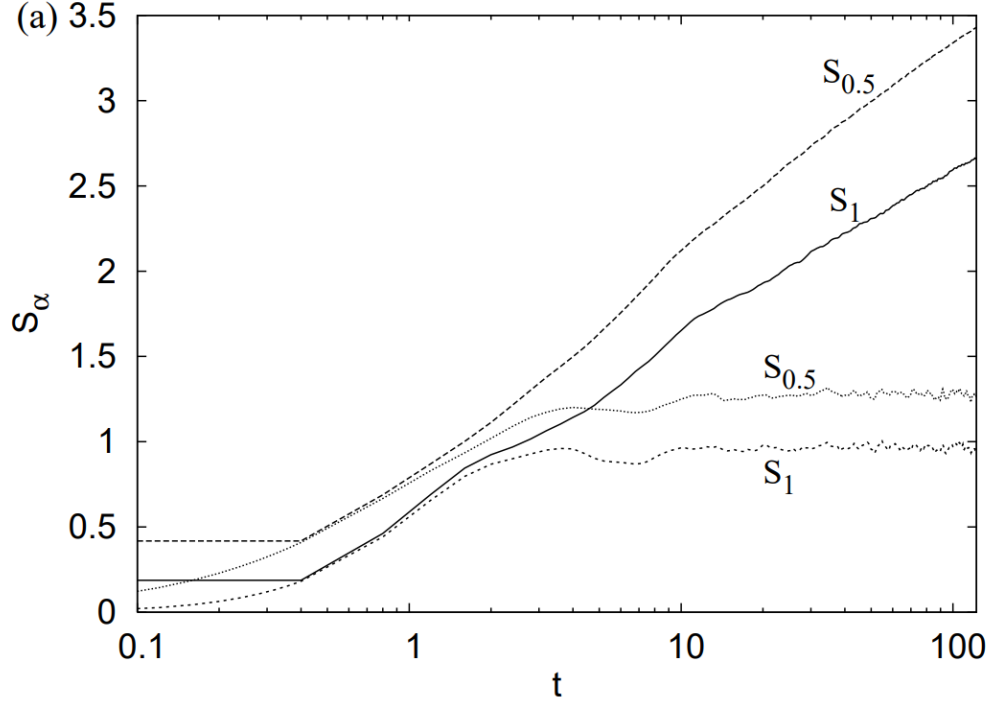


Figure 5.5: Logarithmic plot of time against the R_1 and $R_{0.5}$ entanglement entropies. The initial state was a Néel state with lattice size 50. The system was evolved under the XXZ Hamiltonian where the random magnetic fields were chosen uniformly from the interval $[-5,5]$ and plotted are two initializations of the interaction parameter Δ . $\Delta = 0$ (no interactions) corresponds to the saturating curves, $\Delta = 0.5$ corresponds to the growing curves. Both are averaged over 100 plots.

gument entropy remains constant. Turning on the interactions in the Hamiltonian gives logarithmically increasing entanglement entropy (linear in this plot because it is a log-plot), which shows the system is in the MBL phase.

We reproduced these simulations on the quantum computing framework of ProjectQ, for a lattice of size 8. This data is shown in figure 5.6.

This simulation shows the same behaviour of the entanglement entropy in the different cases such as in figure 5.5. We see after the localization that the

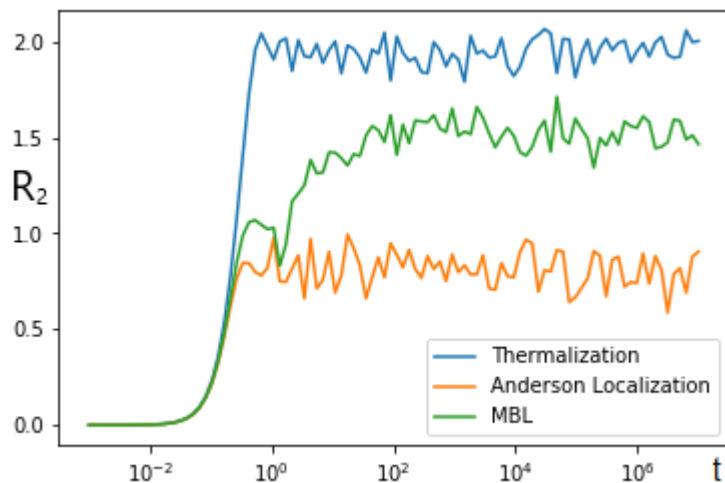


Figure 5.6: Logarithmic plot of the entanglement entropy from ProjectQ. Shown are three configurations of the XXZ-Hamiltonian. $\Delta = 1, h = 1$ shows the thermalizing phase, $\Delta = 0, h = 5$ shows the Anderson Localization phase and $\Delta = 0.5, h = 5$ shows the MBL phase. The plots are averaged over 20 initializations.

entanglement entropy of Anderson localizing phase remains constant, the MBL phase grows linearly and the thermalizing phase grows exponentially, which we expect because it grows linear in time and thus exponential in a logarithmic plot. The key difference being the lattice size, we used a lattice of 8 and in [3] they used specialized software to go to the higher lattice size of 50. This is visible in the MBL phase entanglement growth, this value also stabilizes after some time due to the smaller lattice size.

5.6 Localization Supersymmetric Hamiltonian

In the previous chapter we have seen that the XXZ model exhibits Anderson Localization and MBL. We posed the question if our supersymmetric Hamiltonian could also have these localization phases, since the Hamiltonians are similar. To this we must introduce random disorder in the supersymmetric Hamiltonian and the only way to do that without breaking the supersymmetry is by multiplication

of some random factor λ_i with every term in Q

$$Q = \sum_{i=1}^N \lambda_i (1 - n_{i-1}) c_i (1 - n_{i+1}). \quad (5.18)$$

The Hamiltonian then becomes

$$\begin{aligned} & \sum_{i=1}^N \lambda_i \lambda_{i-1} (1 - n_{i-2}) (c_i^\dagger c_{i-1} + c_{i-1}^\dagger c_i) (1 - n_{i+1}) + \sum_{i=1}^N \lambda_i^2 (1 - n_{i-1}) (1 - n_{i+1}) \\ &= \sum_{i=1}^N \lambda_i \lambda_{i-1} (1 - n_{i-2}) (c_i^\dagger c_{i-1} + c_{i-1}^\dagger c_i) (1 - n_{i+1}) + \sum_{i=1}^N \lambda_i^2 \\ & \quad + \sum_{i=1}^N \lambda_i^2 n_{i-1} n_{i+1} - \sum_{i=1}^N \lambda_i^2 (n_{i-1} + n_{i+1}), \end{aligned} \quad (5.19)$$

this derivation is done in Appendix A.3. Comparing the XXZ Hamiltonian

$$H = \sum_{i=1}^{N-1} (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i) + \Delta \sum_{i=1}^{N-1} n_i n_{i+1} + \sum_{i=1}^N h_i n_i, \quad (5.20)$$

we conject that the supersymmetric model will have no Anderson localization since we can only tune the λ_i parameter we will always have interactions in the model. To introduce disorder such as in the XXZ Hamiltonian, we need that the on site potential is large enough compared to the hopping parameter (which is 1 in the XXZ model). Since these effects are average effects, this condition of disorder translates in our case to

$$\mathbb{E}[\lambda^2] > c \mathbb{E}[\lambda]^2, \quad (5.21)$$

we want the second moment to be some factor larger than the mean squared. Rewriting this we get

$$\frac{\mathbb{E}[\lambda^2] - \mathbb{E}[\lambda]^2}{\mathbb{E}[\lambda]^2} > (c - 1) \quad (5.22)$$

$$\frac{\sigma^2}{\mu^2} > (c - 1) \leftrightarrow \frac{\sigma}{\mu} > \sqrt{c - 1}, \quad (5.23)$$

or in other words we want to draw our λ_i from a distribution which has the desired property that σ/μ is large enough. What we also cannot allow is for

λ_i to be negative since it would be not physical to have hopping parameters which are negative. There are distributions that are positive and of which μ and σ can be tuned as desired, the beta-distribution is an example of this. However taking a continuous distribution with the desired property will always be heavily centered to zero and have a large tail. This is something we don't want because if we have a λ_i that is almost 0 we break the hopping in the chain. To counter these problems we took a general normal distribution around 1, and cast all the negative values to 0.1. These new distribution are not continuous and have a new mean and sigma than before. We got some results as follows starting from $\mathcal{N}(1, \sigma)$.

$\sigma/1$	σ'/μ'
0	0
0.2	0.2
0.4	0.39
0.6	0.56
0.8	0.68

We time evolve an initial state under the supersymmetric Hamiltonian in which we introduce randomness as done in equation (5.19) and we have chosen the coefficients λ_i to be from the distributions we just created. We get the logarithmic plot in time of the entanglement entropy as shown in 5.7. These plots are created using the classical matrix exponentiation method as described in section 3.1.4. From the plots we see that for a higher value of σ/μ i.e. a higher disorder in the on-site potential compared to the hopping parameter, that the entropy begins to grow linearly after first growing exponentially. This suggests that our system is indeed in the MBL phase. This was not expected beforehand, in the supersymmetric model in equation (5.19) we see that the interactions and the onsite potential are both dependent on λ_i^2 and therefore we cannot tune these coefficients relative to each other. Also in the simulations in [3] the interaction parameter Δ was fixed while in our case it is randomly chosen.

An extra method to check if our system is in the MBL regime is to set the interactions in the Hamiltonian to zero. In the MBL phase this should result in a system in the Anderson localized phase. In our simulation in the ProjectQ we can set up the Hamiltonian without the interacting term $\sum_{i=1}^N \lambda_i^2 n_{i-1} n_{i+1}$, we also take away the interacting terms arising in the hopping term to get as final

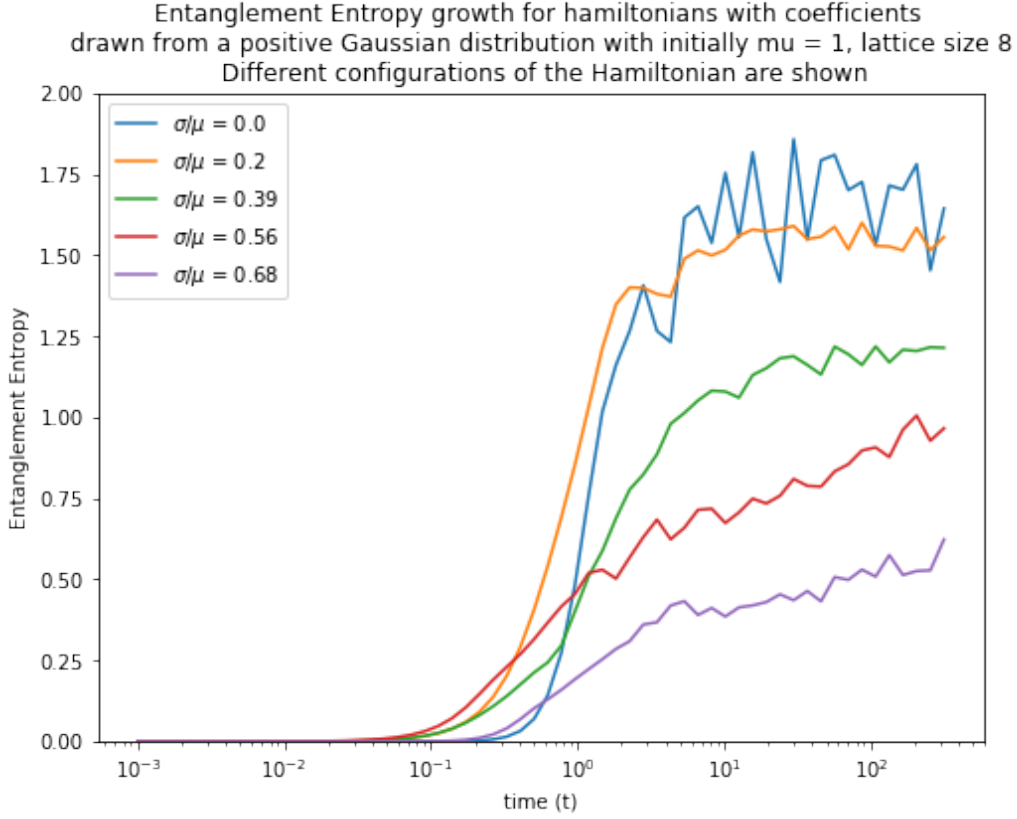


Figure 5.7: Logarithmic plot of the entanglement entropy from ProjectQ. Shown are different configurations of the supersymmetric Hamiltonian with a lattice size 8. The coefficients are drawn from a distribution initially a normal distribution $\mathcal{N}(1, \sigma)$ for different σ where all the negative values are cast to 0.1. The new σ/μ is shown in the legend. All the plots are averaged over 20 hamiltonian configurations and 3 different initial states with 2 qubits.

Hamiltonian

$$H = \sum_{i=1}^N \lambda_i \lambda_{i-1} (c_i^\dagger c_{i-1} + c_{i-1}^\dagger c_i) + \sum_{i=1}^N \lambda_i^2 - \sum_{i=1}^N \lambda_i^2 (n_{i-1} + n_{i+1}). \quad (5.24)$$

This breaks the supersymmetry of the Hamiltonian. The entanglement entropy time evolved under this Hamiltonian in the setup where $\sigma/\mu = 0.56$ is compared to the previous case in which we found indication for MBL in figure 5.8.

The figure shows what we expected, after growing initially the entanglement growth of the non-interacting case becomes constant, which indicates Anderson Localization. Also a thermalizing state is plotted as reference.

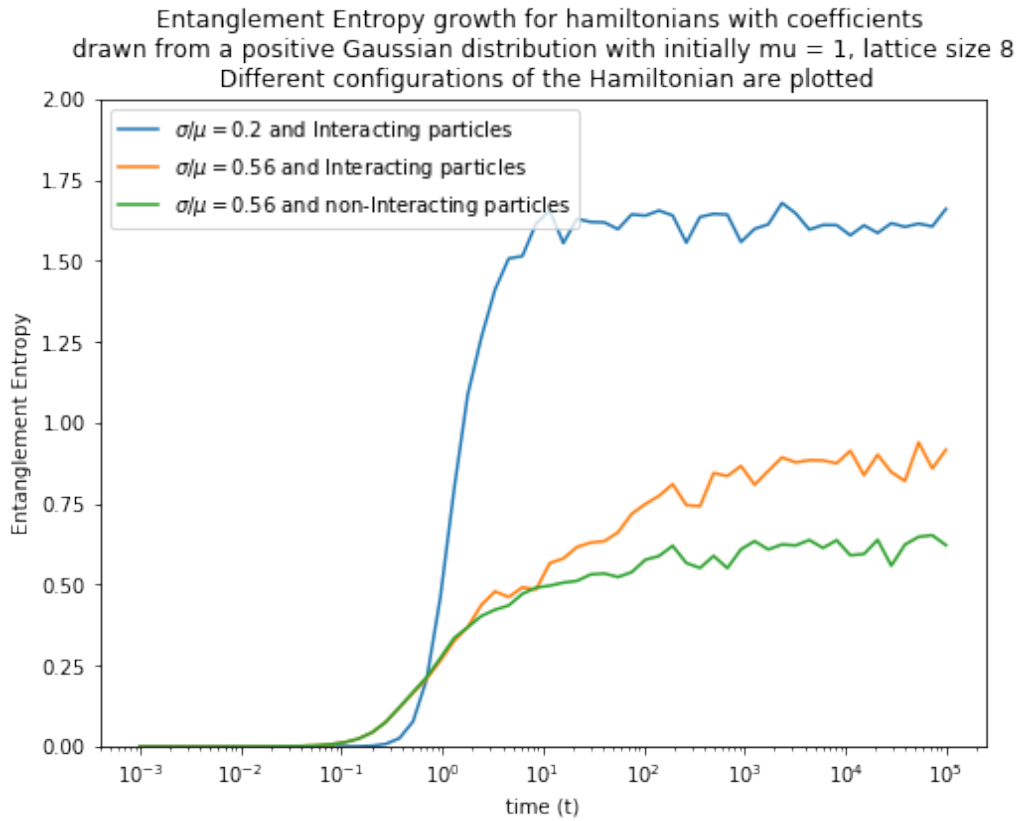


Figure 5.8: Logarithmic plot of the entanglement entropy from ProjectQ. Shown are 2 different configurations of the supersymmetric Hamiltonian with a lattice size 8 and 1 configuration of with no interactions. The coefficients are drawn from a distribution initially a normal distribution $\mathcal{N}(1, 0.6)$ for different σ where all the negative values are cast to 0.1. The new σ/μ value is 0.56. Two cases are from the interacting Hamiltonian

Chapter 6

Conclusion

One of the goals of this thesis was to implement a Hamiltonian system in the framework of ProjectQ and simulate quantum time evolutions. We have introduced the Jordan-Wigner transformation to bring Hamiltonians from second quantization to Pauli operators, for this we used the OpenFermion framework. With this we have presented the Suzuki-Trotter approximation to arbitrarily close generate the time evolution quantum circuit of such a Hamiltonian in Pauli operators. These quantum circuits were implemented in the ProjectQ framework and we have successfully simulated time evolution in the quantum simulator. Also a classical quantum time evolution procedure was created using OpenFermion and the performance of the approximate quantum circuit was checked against the classical direct calculation.

A second part of this thesis was to implement a quantum algorithm within ProjectQ that could be used to calculate the second Renyi entanglement entropy of a state. We have implemented the Swap test algorithm and the faster but needing more ancillary qubits algorithm based on Quantum Amplitude Estimation. The performance of both methods was analyzed.

Ultimately both concepts of quantum time evolution and entanglement entropy were combined in creating calculating the time evolution of the entanglement entropy of different systems. The systems of interest in this thesis were the Heisenberg XXZ model and the $\mathcal{N} = 2$ supersymmetric Hamiltonian. The concept of Anderson Localization and Many Body Localization (MBL) were discussed. In Anderson Localization particles don't interact and become completely localized on a lattice chain, a characterization of this phase is no entanglement

growth through time. However allowing for small interactions breaks the localization and the system can become Many Body Localized, this is characterised by logarithmic entanglement growth. The Many Body Localized state is proven to be a robust phase that does not vanish when one introduces small perturbations. This makes MBL phases the only known robust systems that violate the Eigenstate Thermalization Hypothesis, which gives a statement on the convergence of quantum states to thermal equilibrium.

It was shown in previous simulations that in configurations with enough disorder in the on site potential the Heisenberg XXZ model is in Anderson Localization phase if there are no interactions and becomes in the MBL phase when small interactions are introduced. This was shown with the entanglement entropy time evolution of the different configurations. We have reproduced these with the quantum simulator of the framework of ProjectQ. We conjectured that the supersymmetric Hamiltonian could also have an MBL phase if we introduced random disorder such that the site potential is large enough with respect to the expected value of the random coefficients of the hopping parameter. We found strong indication that this is indeed the case.

These results allow for a number of possible future research question. One concerning the the choice of the random coefficients in the supersymmetric Hamiltonian. These can be chosen periodically at random and otherwise they are fixed at 1, in the case of period of 3 lattice sites interesting kink dynamics arise. Other follow research includes out of time order correlators (OTOC's) these were investigated in this thesis, but we did not manage to use them in this context. The idea of OTOC's is however of importance in localized systems. One of the most important future researches is of course a physical implementation in a lab of these systems, a physical realization of a MBL state would be of great significance.

Bibliography

- [1] D. A. Abanin, E. Altman, I. Bloch, and M. Serbyn, “Colloquium: Many-body localization, thermalization, and entanglement,” *Reviews of Modern Physics*, vol. 91, no. 2, p. 021001, 5 2019. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.91.021001>
- [2] P. Fendley, K. Schoutens, and J. De Boer, “Lattice Models with $N = 2$ Supersymmetry,” Tech. Rep., 2002. [Online]. Available: <https://arxiv.org/pdf/hep-th/0210161.pdf>
- [3] M. M. Markožnidarič, T. Prosen, and P. Prelovšek, “Many body localization in Heisenberg XXZ magnet in a random field,” Tech. Rep., 2008. [Online]. Available: <https://arxiv.org/pdf/0706.2539.pdf>
- [4] D. Deutsch, A. Barenco, and A. Ekert, “Universality in Quantum Computation,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 449, no. 1937, pp. 669–677, 6 1995. [Online]. Available: <http://rspa.royalsocietypublishing.org/cgi/doi/10.1098/rspa.1995.0065>
- [5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2010. [Online]. Available: <http://ebooks.cambridge.org/ref/id/CBO9780511976667>
- [6] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation.” American Mathematical Society, 2002, pp. 53–74. [Online]. Available: <http://www.ams.org/conm/305/>
- [7] R. P. Feynman, “Simulating Physics with Computers,” Tech. Rep. 6, 1982. [Online]. Available: <https://people.eecs.berkeley.edu/~christos/classics/Feynman.pdf>

- [8] S. Lloyd, “Universal Quantum Simulators,” pp. 1073–1078. [Online]. Available: <https://www.jstor.org/stable/2899535>
- [9] M. Suzuki, “Relationship between d-Dimensional Quanta! Spin Systems and (d+ 1)-Dimensional Ising Systems,” Tech. Rep. 5, 1976. [Online]. Available: <https://academic.oup.com/ptp/article-abstract/56/5/1454/1860476>
- [10] —, “General theory of higher-order decomposition of exponential operators and symplectic integrators,” *Physics Letters A*, vol. 165, no. 5-6, pp. 387–395, 6 1992. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/037596019290335J>
- [11] R. De Wolf, “Quantum Computing: Lecture Notes.” [Online]. Available: <https://homepages.cwi.nl/~rdewolf/qcnotes.pdf>
- [12] C. Yung, M. hong, J. D. Whitfield, S. Boixo, D. Gabriel Tempel, and A. Aspuru-Guzik, “Introduction to Quantum Algorithms for Physics and Chemistry,” Tech. Rep. [Online]. Available: <http://nrs.harvard.edu/urn-3:HUL.InstRepos:11084801>
- [13] E. Fradkin, “Jordan-Wigner transformation for quantum-spin systems in two dimensions and fractional statistics,” *Physical Review Letters*, vol. 63, no. 3, pp. 322–325, 7 1989. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.63.322>
- [14] J. R. McClean, I. D. Kivlichan, K. J. Sung, D. S. Steiger, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby, P. Gokhale, T. Häner, T. Hardikar, V. Havlíček, C. Huang, J. Izaac, Z. Jiang, X. Liu, M. Neeley, T. O’Brien, I. Ozfidan, M. D. Radin, J. Romero, N. Rubin, N. P. D. Sawaya, K. Setia, S. Sim, M. Steudtner, Q. Sun, W. Sun, F. Zhang, and R. Babbush, “OpenFermion: The Electronic Structure Package for Quantum Computers,” Tech. Rep. [Online]. Available: www.openfermion.org
- [15] C. E. Shannon, “Prediction and Entropy of Printed English,” *Bell System Technical Journal*, vol. 30, no. 1, pp. 50–64, 1 1951. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773263>
- [16] J. V. Neumann, *Mathematical foundations of quantum mechanics: New edition*, 2018. [Online]. Available: <https://books.google.nl/books?hl=en&lr=&id=z3g9DwAAQBAJ&oi=>

find&pg=PP1&dq=Mathematical+Foundations+of+Quantum+Mechanics.
&ots=4VYeIyzgeT&sig=610QWNWdJO-JHAGY-zECa4QiHDE

- [17] A. Ekert and P. L. Knight, “Entangled quantum systems and the Schmidt decomposition,” *American Journal of Physics*, vol. 63, no. 5, pp. 415–423, 5 1995. [Online]. Available: <http://aapt.scitation.org/doi/10.1119/1.17904>
- [18] S. Johri, D. S. Steiger, and M. Troyer, “Entanglement spectroscopy on a quantum computer,” *Physical Review B*, vol. 96, no. 19, p. 195136, 11 2017. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.96.195136>
- [19] D. S. Steiger, T. Häner, and M. Troyer, “ProjectQ: An Open Source Software Framework for Quantum Computing,” Tech. Rep., 2018. [Online]. Available: <https://quantum-journal.org/wp-content/uploads/2018/01/q-2018-01-31-49.pdf>
- [20] M. Srednicki, “The approach to thermal equilibrium in quantized chaotic systems,” *Journal of Physics A: Mathematical and General*, vol. 32, no. 7, pp. 1163–1175, 2 1999. [Online]. Available: <http://stacks.iop.org/0305-4470/32/i=7/a=007?key=crossref.a70ec040b8a9210e92e2e0f5a09e67aa>
- [21] J. M. Deutsch, “Quantum statistical mechanics in a closed system,” *Physical Review A*, vol. 43, no. 4, pp. 2046–2049, 2 1991. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.43.2046>
- [22] P. Calabrese and J. Cardy, “Evolution of entanglement entropy in one-dimensional systems,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 04, p. P04010, 4 2005. [Online]. Available: <http://stacks.iop.org/1742-5468/2005/i=04/a=P04010?key=crossref.7e427f5b65ccbfcf3b82757d806c7b35>
- [23] P. W. Anderson, “Absence of Diffusion in Certain Random Lattices,” *Physical Review*, vol. 109, no. 5, pp. 1492–1505, 3 1958. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRev.109.1492>
- [24] B. Kramer and A. MacKinnon, “Localization: theory and experiment,” *Reports on Progress in Physics*, vol. 56, no. 12, pp. 1469–1564, 12 1993. [Online]. Available: <http://stacks.iop.org/0034-4885/56/i=12/a=001?key=crossref.77bed25384b7b7f3282dd2cf93159bf0>

- [25] L. Fleishman and P. W. Anderson, “Interactions and the Anderson transition,” *Physical Review B*, vol. 21, no. 6, pp. 2366–2377, 3 1980. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.21.2366>
- [26] I. V. Gornyi, A. D. Mirlin, and D. G. Polyakov, “Interacting Electrons in Disordered Wires: Anderson Localization and Low- T Transport,” *Physical Review Letters*, vol. 95, no. 20, p. 206603, 11 2005. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.95.206603>

Appendices

Appendix A

Supersymmetric Hamiltonian

A.1

At first sight it might not seem clear why $Q^2 = (Q^\dagger)^2 = 0$. We will show the result for Q , the result for Q^\dagger follows analogously. First we calculate the commutation relation between a creation or annihilation operator and $(1 - n_i)$.

$$\begin{aligned}c_i(1 - n_j) &= c_i - c_i c_j^\dagger c_j \\ &= c_i - (\delta_{ij} - c_j^\dagger c_i) c_j \\ &= c_i - \delta_{ij} c_j - c_j^\dagger c_j c_i \\ &= c_i - c_j^\dagger c_j c_i - \delta_{ij} c_j \\ &= (1 - n_j) c_i - \delta_{ij} c_j\end{aligned}$$

$$\begin{aligned}c_i^\dagger(1 - n_j) &= c_i^\dagger + c_j^\dagger c_i^\dagger c_j \\ &= c_i^\dagger + c_j^\dagger (\delta_{ij} - c_j c_i^\dagger) \\ &= c_i^\dagger - c_j^\dagger c_j c_i^\dagger + \delta_{ij} c_j^\dagger \\ &= (1 - n_j) c_i^\dagger + \delta_{ij} c_j^\dagger\end{aligned}$$

Note further that the product of two number operators always commute since their application on a state yields a number and keeps the state intact. Now we

can show $Q^2 = 0$

$$\begin{aligned}
Q^2 &= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})c_i c_j (1 - n_{i-1})(1 - n_{i+1}) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})c_i (1 - n_{i-1})c_j (1 - n_{i+1}) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})((1 - n_{i-1})c_i - \delta_{ij}c_j)c_j (1 - n_{i+1}) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})(1 - n_{i-1})((1 - n_{i+1})c_i - \delta_{ij}c_j)c_j \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})(1 - n_{i-1})(1 - n_{i+1})c_i c_j \\
&= \begin{cases} -\sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})(1 - n_{i-1})(1 - n_{i+1})c_j c_i, & \text{by commutation relations} \\ \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})(1 - n_{i-1})(1 - n_{i+1})c_j c_i, & \text{by relabelling } i \leftrightarrow j \end{cases}
\end{aligned}$$

and we conclude $Q^2 = 0$.

A.2

We will show how the Hamiltonian in equation (5.11) can be transformed into the Hamiltonian in equation (5.12).

With the commutation relations from the previous Appendix A.1 we will commute all the annihilation and creation operators to one side in the Hamil-

tonian.

$$\begin{aligned}
H &= \{Q^\dagger, Q\} = Q^\dagger Q + Q Q^\dagger \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1}) c_i c_j^\dagger (1 - n_{j-1})(1 - n_{j+1}) \\
&\quad + \sum_{i,j} (1 - n_{i+1})(1 - n_{i-1}) c_i^\dagger c_j (1 - n_{j-1})(1 - n_{j+1}) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1}) (c_i c_j^\dagger + c_i^\dagger c_j) (1 - n_{j-1})(1 - n_{j+1}) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1}) (c_i (1 - n_{j-1})(1 - n_{j+1}) c_j^\dagger + c_i^\dagger (1 - n_{j-1})(1 - n_{j+1}) c_j) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1}) ((1 - n_{j-1}) c_i - \delta_{i,j-1} c_{j-1}) (1 - n_{j+1}) c_j^\dagger + \\
&\quad ((1 - n_{j+1}) c_i^\dagger + \delta_{i,j-1} c_j^\dagger) (1 - n_{j+1}) c_j \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1}) (((1 - n_{j-1}) c_i (1 - n_{j+1}) c_j^\dagger + (1 - n_{j-1}) c_i^\dagger (1 - n_{j+1}) c_j) - \\
&\quad \delta_{i,j-1} c_{j-1}^\dagger (1 - n_{j+1}) c_j^\dagger + \delta_{i,j-1} c_{j-1} (1 - n_{j+1}) c_j) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1}) ((1 - n_{j-1}) ((1 - n_{j+1}) c_i c_j^\dagger - \delta_{i,j+1} c_{j+1} c_j^\dagger) \\
&\quad + (1 - n_{j+1}) c_i^\dagger c_j + \delta_{i-1,j} c_{j+1}^\dagger c_j) - \delta_{i,j-1} (1 - n_{j+1}) c_{j-1} c_j^\dagger + \delta_{i,j-1} (1 - n_{j+1}) c_{j-1}^\dagger c_j) \\
&= \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1}) (1 - n_{j-1})(1 - n_{j+1}) (c_i c_j^\dagger + c_i^\dagger c_j) \\
&\quad + \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})(1 - n_{j-1}) (-\delta_{i-1,j} c_{j+1} c_j^\dagger + \delta_{i,j+1} c_{j+1}^\dagger c_j) + \\
&\quad \sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})(1 - n_{j+1}) (-\delta_{i+1,j} c_{j-1} c_j^\dagger + \delta_{i+1,j} c_{j-1}^\dagger c_j) \\
&= \sum_i (1 - n_{i-1})(1 - n_{i+1}) (1 - n_{j-1})(1 - n_{j+1}) (c_i c_j^\dagger - c_j c_i^\dagger + \delta_{i,j}) \\
&\quad + \sum_i (1 - n_{i-1})(1 - n_{i+1})(1 - n_{i-2}) (-c_i c_{i-1}^\dagger + c_i^\dagger c_{i-1}) \\
&\quad + \sum_i (1 - n_{i+1})(1 - n_{i+2}) (-c_i c_{i+1}^\dagger + c_i^\dagger c_{i+1})
\end{aligned}$$

$$\begin{aligned}
&= \sum_{i,j} (1 - n_{i-1})^2 (1 - n_{i+1})^2 \\
&\quad + \underbrace{\sum_{i,j} (1 - n_{i-1})(1 - n_{i+1})(1 - n_{j-1})(1 - n_{j+1})c_i c_j^\dagger - (i \leftrightarrow j)}_{=0 \text{ by relabeling}} \\
&\quad + \sum_i (1 - n_{i+1})(1 - n_{i-2}) \underbrace{((1 - n_{i-1})c_{i-1}^\dagger c_i + c_i^\dagger c_{i-1})}_{\rightarrow 0} \\
&\quad + \sum_i (1 - n_{i-1})(1 - n_{i+2}) \underbrace{((1 - n_{i+1})c_{i+1}^\dagger c_i + c_i^\dagger c_{i+1})}_{\rightarrow 0} \\
&= \sum_i (1 - n_{i-1})^2 (1 - n_{i+1})^2 - \sum_i \underbrace{(1 - n_{i-1})(1 - n_{i-1})(1 - n_{i-2})}_{\rightarrow 1 \text{ by } c_{i-1}} c_i^\dagger c_{i-1} \\
&\quad + \sum_i (1 - n_{i-1}) \underbrace{(1 - n_{i+1})(1 - n_{i+2})}_{\rightarrow 1 \text{ by } c_{i+1}} c_i^\dagger c_{i+1} \\
&= \sum_i (1 - n_{i-1})^2 (1 - n_{i+1})^2 + \sum_i (1 - n_{i-2}) c_i^\dagger c_{i-1} (1 - n_{i-1}) \\
&\quad + \sum_i (1 - n_{i-1}) c_i^\dagger c_{i+1} (1 - n_{i+2}) \\
&= \sum_i (1 - n_{i-1})(1 - n_{i+1}) + \sum_i (1 - n_{i-2})(c_i^\dagger c_{i-1} + c_{i-1}^\dagger c_i)(1 - n_{i+1})
\end{aligned}$$

A.3

As mentioned the only randomness we can introduce in the Hamiltonian while not breaking the supersymmetry (i.e. keeping the Q, Q^\dagger invariant) is multiplying each term in the sum of Q .

$$Q = \sum_i \lambda_i (1 - n_{i-1}) c_i (1 - n_{i+1}),$$

this is the same as sending every creation and annihilation operator at site i to λ_i times this operator. If we track the calculations in Appendix A.2 before we set

any term to zero we get

$$\begin{aligned} \sum_{i,j} \lambda_i \lambda_j (1 - n_{i-1})(1 - n_{i+1})(1 - n_{j-1})(1 - n_{j+1})(c_i c_j^\dagger - c_j c_i^\dagger + \delta_{i,j}) \\ + \sum_i \lambda_i \lambda_j (1 - n_{i-1})(1 - n_{i+1})(1 - n_{i-2})(-c_i c_{i-1}^\dagger + c_i^\dagger c_{i-1}) \\ + \sum_i \lambda_i \lambda_j (1 - n_{i+1})(1 - n_{i+2})(-c_i c_{i+1}^\dagger + c_i^\dagger c_{i+1}), \end{aligned}$$

tracking these factors in the summations over the Kronecker delta terms we get

$$\sum_i \lambda_i^2 (1 - n_{i-1})(1 - n_{i+1}) + \sum_i \lambda_i \lambda_{i-1} (1 - n_{i-2})(c_i^\dagger c_{i-1} + c_{i-1}^\dagger c_i)(1 - n_{i+1})$$

A.4

Link to the github with the code used in this project. <https://github.com/philiplunel?tab=projects>