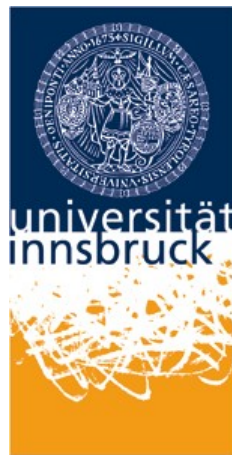Leopold-Franzens University Innsbruck
Faculty for Mathematics, Computer Science and
Physics

Institute for Computer Science



# Bachelor's thesis

**for the purpose of achieving the academic degree**

## Bachelor of Science

## Estimating ground state wavefunctions using QPE vs the VQE Method

by

Christoph Moser
(Matr.-Nr.: 11941436)

# Declaration on oath

*I hereby declare in lieu of oath by my own signature, that I have written this thesis on my own initiative and have not used any sources or aids other than those indicated. All places, that have been taken from the sources indicated, either verbatim or in contentual, are marked as such. I declare that I agree to the archiving of this bachelor thesis.*

Location and Date: _____

Signature: _____

# Contents

<center><i>Contents</i></center>

<center>I</center>

# List of Figures

# Abstract

One of the foundational equations of quantum mechanics is the Schrödinger equation, which enables the calculation of the energy state of a given closed quantum system. However, the equation is not analytically solvable for any system with n components interacting with each other (n-body-systems), where $n >= 3$. Hence researchers have developed approximation methods for the calculation of these energies. However, these approximation methods cannot be run on classical computers efficiently, that is, polynomial in the number of individual systems $n$.

With the advent of quantum computing in the recent years, multiple algorithms suitable for implementation on quantum computers have been proposed that promise a speedup in runtime complexity for this approximation. This thesis focuses on two such algorithms, the Variational Quantum Eigensolver (VQE) and the Quantum Phase Estimation Algorithm (QPE/ PEA). This thesis describes the process of mapping the general formalism of the Schrödinger equation onto the VQE and the QPE throughout various steps and transformations and also describes the concrete implementation of these on the example of two molecules, H2 and BeH2. Chapter 1 introduces the basics of quantum computing, quantum mechanics, and the functionality of the algorithms, Chapter 2 gives an overview of the technology stack used for tackling the problem, chapter 3 describes the methodology used. Chapter 4 offers an assessment of the results and chapter 5 discusses further work related to the algorithm and the quantum computing field in general.

# 1 Introduction

Quantum mechanics is the best experimentally verified theory of describing physical processes to date. In the domain of quantum mechanics, the energy level of systems are described by a fundamental differential equation, the Schrödinger equation. The energy levels for various small systems is of great interest in many fields of science, for example the behaviour of individual molecules dependent on these energy states plays a significant role in the likes of drug discovery, materials research and many more.

Thus by extension, solutions to the Schrödinger equation for certain atoms, molecules and general many-body systems are of high interest. However, the equation is only analytically solvable for very simple problems, namely for problems with only two interacting particles, for example a Hydrogen atom modelled by the interaction between its positively charged nucleus and the negatively charged electron. For molecules consisting of more than 2 elementary particles, no analytical solution to the Schrödinger equation can be found, but the exact solution can be numerically approximated by various techniques, such as the Hartree Fock method or the Slater method. Such "classical" techniques provide satisfactory results up until a certain amount of particles, but become computationally infeasible quickly due to the exponential scaling of the degrees of freedom for the mathematical description of a many-body-system.

As a result, good approximations for the Schrödinger equation for large many-body-systems are virtually impossible to calculate for classical computers. However, the

theory of quantum computing offers multiple algorithms that allow numerical approximations for such problems on a cheaper computational cost, among which are VQE and QPE, that are being dealt with in this thesis. Historically, this vision of simulating quantum systems by better controllable quantum systems was the motivating idea for quantum computers although many more use cases have since been identified.

This introductory chapter gives an outline of the basic mathematical framework of quantum computing, introduces the Schrödinger equation and the necessary framework to encode a many-body system into the state space of a quantum computer and illustrate the functionality of the used algorithms.

Within this thesis, the view onto quantum computing is restricted to an ideal, purely mathematical version, whereas current physical limitations of quantum computing, such as noise, decoherence, restrictions imposed by the paradigms etc. are not considered.

## 1.1 The basics of quantum computing

In the following sections, the basic mathematical foundation of quantum computation is introduced. The discussed concepts include qubits and composite systems in Hilbert spaces, operators and their action on qubits, qubit measurement, time evolution of operators, The encoding of qubits and operators into vectors and matrices and the circuit representation of quantum algorithms. While it was attempted to make this thesis accessible to readers without prior knowledge in the area, a complete introduction into quantum computing far exceeds its scope. The interested reader is therefore referred to the section on further reading 1.5 for references to introductory material on the topic.

## 1.1.1 Qubits and Hilbert spaces

Analogous to the bits of classical computers with two possible defined states, 0 and 1, quantum computers operate on *qubits*. However, a general qubit state $|\psi\rangle$ is allowed to be in a superposition of the two states

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

where $\alpha, \beta \in \mathbb{C}$ are complex coefficients called *probability amplitudes* with the constraint that $|\alpha|^2 + |\beta|^2 = 1$.

The notation of $|\cdot\rangle$ is called a *ket*, essentially signifying that the respective object is a quantum mechanical state and not a mere scalar value. In general such kets are elements of a *Hilbert space* $\mathcal{H}$, a special kind of complex vector space equipped with an inner product. Due to conciseness, no exact definition of a Hilbert space is given in the thesis, but some important properties of the used Hilbert space (the Hilbert space of complex n-tuples) and its elements are shown throughout this section and can be considered as axioms for the purpose of this thesis.

- Elements $|v\rangle \in \mathcal{H}$ are called kets and can be represented as complex n-tuple

$$|v\rangle \equiv \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

- The inner product $\langle v, w\rangle$ of two kets $|v\rangle, |w\rangle$ is defined as:

$$\langle v, w\rangle = \langle v|w\rangle = \begin{bmatrix} v_1^* & v_2^* & \dots & v_n^* \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

  where $v_i^*$ denotes the complex conjugate of $v_i$.

- Two kets $|v\rangle, |w\rangle$ are called *orthogonal* if $\langle v|w\rangle = 0$

- A ket $|v\rangle$ is *normalized* if $\||v\rangle\| = \sqrt{\langle v|v\rangle} = 1$

## 1.1.2 Operators and state evolution

Qubit states are transformed by *unitary operators*, which are defined by the condition that an operator $U$ is unitary if and only if

$$U^\dagger\, U = U\, U^\dagger = I$$

where I is the identity operator. This unitarity constraint is a result stemming from the reversibility of the evolution of quantum systems.

On the example of a qubit in an initial state $|\psi\rangle$ that is transformed by some operator $U$, one would simply write the transformation as

$$U\,|\psi\rangle = |\phi\rangle$$

where $|\phi\rangle$ is the result of the transformation.

Another important type of operators are the Hermitian operators, who are defined by the condition that $V$ is *hermitian* if and only if

$$V^\dagger = V$$

These hermitian operators can be used as observables, i.e. as a way to measure a specific property of a given state quantum state $|\psi\rangle$.

**Composite systems with multiple Qubits**

Systems consisting of multiple qubits can be mathematically defined by composing the individual states via the Tensor Product $\otimes$. For two states $|\psi\rangle$ and $|\phi\rangle$, the composite state is then given as $|\psi\rangle \otimes |\phi\rangle$, which can be abbreviated as $|\psi\rangle\,|\phi\rangle$ or $|\psi\phi\rangle$.

For example, A state consisting of three qubits which are all in the same state $|0\rangle$ as the composite system $|0\rangle \otimes |0\rangle \otimes |0\rangle$ can be represented simply as $|000\rangle$. Extending the example, one can assume that three operators $U$, $V$, $T$ act on the first, second and third qubit respectively. This results in the state

$$U\,|0\rangle \otimes V\,|0\rangle \otimes T\,|0\rangle = U \otimes V \otimes T\,|000\rangle$$

where $U \otimes V \otimes T$ is now a new composite operator acting on the three qubit state $|000\rangle$.

**Matrix-vector representation and basis sets**

A useful way to represent quantum states is in terms of their vector representation as complex n-tuple. By elementary linear algebra it is known that for a vectorspace $V$, any vector $\vec{v}$ can be represented by a linear combination of Basis vectors from a basis sets $\mathcal{B}$. While in theory all basis set are equally potent, in the realm of quantum computing, often the computational basis is preferred. In the computational basis, a ket $|n\rangle$ is represented as its binary representation $|b_0 b_1 \ldots b_m\rangle$, and the vector representation is of size $2^n$ consists of 0 everywhere except a 1 at index $n$.

$$|1\rangle \leftrightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad |3\rangle \equiv |10\rangle \leftrightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

## 1.1.3 The quantum circuit representation

On top of the precise mathematical formalism, a more intuitive way to represent qubits and operators acting onto them is the circuit representation. In this representation, qubits are represented as wires, where the evolution throughout time is from left to right, and operators are logic gates acting onto them. Figure 1.1 shows an example where three qubits are situated next to each other where the first two

have a so called Hadamard gate applied onto them and the third one an X gate. Afterwards, a CNOT gate acts onto qubits one and two, before lastly, qubits two and three are readout by measurements M1 and M2. These measurements results are then stored in classical channels i.e. usual registers on a classical computer, which may be represented by wires with two lines. By convention, if no other state $|\psi\rangle$ is signified on the left side of the wires, they are considered to be in the $|0\rangle$ state initially.
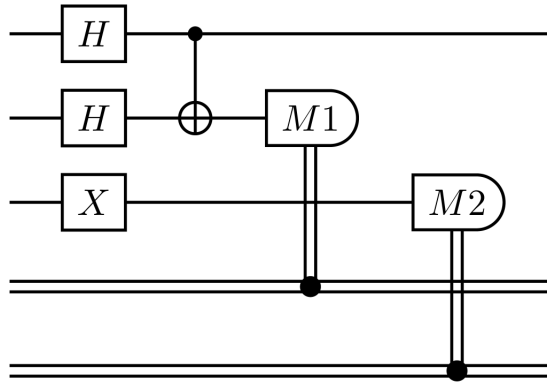


Figure 1.1: Example for the circuit representation

## 1.1.4 Universal set of quantum gates

Similar to how classical computation requires a universal set of logic gates, quantum computing also requires a universal set of operators, often also called a universal set of quantum gates. While many such universal sets exist, for historical reasons we will focus on the following set of important quantum gates:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

While it should be pointed that in general different and potentially subsets of quantum gates may be chosen, this set is already universal.

**Circuit representation of the universal set**

The circuit representations of the universal quantum gate set are shown below:
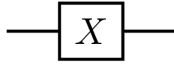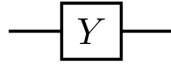


Figure 1.2: X Gate          Figure 1.3: Y Gate          Figure 1.4: Z Gate
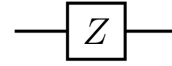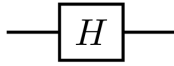


Figure 1.5: Hadamard Gate          Figure 1.6: CNOT Gate
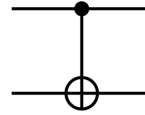
## 1.2 Basics of Quantum mechanics

In essence, Quantum mechanics can be described as a mathematical framework for describing and developing physical theories. The following four postulates, loosely adapted from [15], form the foundation of quantum mechanics.

1. Any isolated physical system can be described by a complex vector $|\psi\rangle$ living in an associated state space mathematically represented by some Hilbert space $\mathcal{H}$. The statevector $|\psi\rangle$ fully stores any information of the physical system.

2. The evolution of a closed quantum system is described by the Schrödinger equation

$$i\hbar\frac{d}{dt}\ket{\psi} = \hat{H}\ket{\psi}$$

   A closer analysis of the Schrödinger equation will be given in section 1.2.1

3. Measurements on quantum states i.e. state vectors are described by a collection $M_m$ of measurement operators, where the index $m$ refers to the different probabilistic outcomes possible when performing measurement M.

4. The state space of a composite physical system is the tensor product of the state spaces of the individual systems.

## 1.2.1 The Schrödinger equation

In the quantum mechanics formalism, the time evolution of quantum state can be described by the *Schrödinger equation*[21]

$$i\hbar\frac{d}{dt}\ket{\psi} = \hat{H}\ket{\psi}$$

where $\hat{H}$ denotes the *Hamiltonian*, i.e. the energy operator of the system. The states $\{\ket{E_i}\}$ which fulfill the eigenvalue equation of the Hamiltonian are called the Eigenstates of the Hamiltonian with corresponding Eigenvalues $E_i$.

$$\hat{H}\ket{E_i} = E_i\ket{E_i}$$

For any given Hamiltonian, there exists an eigenstate $\ket{E_0}$ where the corresponding energy $E_0$ has the smallest value. This so-called *ground state* and its corresponding energy value are of special interest, since it can be used as a reference state for determining the higher energy states (i.e. the *excited states*).

**Classical solution approaches**

In the most common classical approach to the problem of solving the Schrödinger equation, the Hamiltonian is represented as a matrix, the wavefunctions of the quantum states are represented as vectors and the eigenvalues are then calculated using linear algebra techniques. However, while eigenvalue calculation of a general $k \times k$ matrix is polynomial in $k$, the size of the vector description for some *n-body system* will already be of order $2^n$, making classical ground state estimation a problem of order $O(2^{p(n)})$ and therefore exponential in the number of particles.

Throughout the many different techniques for solving the Schrödinger equation, all known classical algorithms are constrained by this problem. For practical applications, this results in a strong restriction on the size of the systems that can be calculated.

## 1.2.2  Quantum simulation

Quantum simulation as a means of solving the Schrödinger equation has its roots in the realization that the Schrödinger equation for the case of a time-independent Hamiltonian $\hat{H}$ as in equation 1.1 can be rewritten as follows [15]:

$$|\psi(t)\rangle = e^{-i\hat{H}t} |\psi(0)\rangle$$

While the exponentiation $e^{-i\hat{H}t}$ is still computationally hard for general operators $\hat{H}$, it can be tractable for certain subsets of operators, namely those that can be decomposed into a sum of local interactions

$$\hat{H} = \sum_k H_k$$

This technique of applying the exponential of an operator will be useful in the realization of both the VQE and the QPE to map the Hamiltonian operator into the state of the quantum computer.

## 1.2.3 The ab initio Hamiltonian

Ab initio methods seek to construct the problem formulations by only employing information from the underlying theoretical framework and first principles, without any additional prior assumptions or empirical parameters. Specifically, the ab-initio Hamiltonian for a molecule with i electrons and I nuclei is given by [13]:

$$H = \underbrace{-\sum_i \frac{\hbar^2}{2m_e}\nabla_i^2}_{\text{electrons kin. energy}} \underbrace{-\sum_I \frac{\hbar^2}{2M_I}\nabla_I^2}_{\text{nucleis kin. energy}} \underbrace{-\sum_{i,I} \frac{e^2}{4\pi\epsilon_0}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|}}_{\text{pot. energy electron-nuclei}}$$
$$\underbrace{+\frac{1}{2}\sum_{i\neq j} \frac{e^2}{4\pi\epsilon_0}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}}_{\text{pot. energy electron-electron}} \underbrace{+\frac{1}{2}\sum_{I\neq J} \frac{e^2}{4\pi\epsilon_0}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}}_{\text{pot. energy nucleus-nucleus}}$$

(1.1)

where $M_I, \mathbf{R}_I, Z_I$ denote mass, position and atomic number of the $I$th nucleus respectively, and $m_e, \mathbf{r}_i$ denote electron mass and position of the $i$th electron. By the Born-Oppenheimer approximation [5], one can regard the nuclei as classical point charges static in space (and thus neglectable), due to their far greater mass in comparison to the electrons. This leaves the electronic Hamiltonian

$$H_e = -\sum_i \frac{\hbar^2}{2m_e}\nabla_i^2 - \sum_{i,I} \frac{e^2}{4\pi\epsilon_0}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \frac{1}{2}\sum_{i\neq j} \frac{e^2}{4\pi\epsilon_0}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \qquad (1.2)$$

This approximation is a frequently used simplification for molecules and other many-body-systems. Going forward, $H_e$ will be considered as the relevant Hamiltonian for the Schrödinger equation.

## 1.2.4 Hamiltonian Mapping

In order to be able to use the quantum simulation algorithm as introduced in section 1.2.2 $H_e$ has to be transformed into the linear combinations of local interactions $\hat{H}_e = \sum_k H_k$. To this end, multiple steps have to be fulfilled, which are motivated throughout the following sections.

**Basis set**

Since qubits are mathematically described in a so-called Hilbert space $\mathcal{H}$, an appropriate basis set $\{|\phi_i\rangle\}$ spanning that space has to be chosen. A frequently used approach is that of utilizing molecular orbitals. Only the mathematical representations of these orbitals are considered in the following sections, a motivation on the physical significance can be found in [8].

**Selection of the basis set** There exist many ways of representing molecular orbitals in terms of bases. We will utilize a Slater type orbital, namely the **STO-3G** procedure [8]. For that, the molecular orbital will be constructed from the Slater determinant of the electronic orbitals for each of the individual electrons. These in term can be approximated via a linear combination (also called contraction) of 3 Gaussian primitives (hence the 3G in the methods name). So the electronic orbitals will have the form:

$$\chi = \sum_{i=1}^{N=3} c_i \phi_i$$

where $\phi_i$ represent the primitive Gaussian functions:

$$\phi_1 = \left(\frac{2\alpha_1}{\pi}\right)^{\frac{3}{4}} e^{-\alpha_1 r^2} \quad \phi_2 = \left(\frac{2\alpha_2}{\pi}\right)^{\frac{3}{4}} e^{-\alpha_2 r^2} \quad \phi_3 = \left(\frac{2\alpha_3}{\pi}\right)^{\frac{3}{4}} e^{-\alpha_3 r^2}$$

with each $\alpha$ and the $c_i$ being experimentally determined. The relevant coefficients for $\alpha_i$ and $c_i$ for the elements H and Be were taken from Basis Set Exchange [19] and are tabulated in table 1.1. Computing the respective molecular orbitals then

| Element | Orbital | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $c_1$ | $c_2$ | $c_3$ |
|---------|---------|-----------|-----------|-----------|-----------|-----------|-----------|
| H | 1s | 3.425251 | 0.623914 | 0.168855 | 0.154328 | 0.535328 | 0.444635 |
| Be | 1s | 30.167871 | 5.495115 | 1.487192 | 0.154328 | 0.535328 | 0.444635 |
| | 2s | 1.314833 | 0.305538 | 0.099371 | -0.099967 | 0.399513 | 0.700115 |
| | 2p | 1.314833 | 0.305538 | 0.099371 | 0.155916 | 0.607684 | 0.391957 |

Table 1.1: STO-3G Coefficients for the 1s, 2s and 2p orbitals

yields the following results:

$$\chi_H^{1s} = 0.276934 \ \exp(-3.425251 \ r^2) + 0.267839 \ \exp(-0.623914 \ r^2) + 0.083474 \ \exp(-0.168855 \ r^2)$$

$$\chi_{Be}^{1s} = 1.415846 \ \exp(-30.167871 \ r^2) + 1.369345 \ \exp(-5.495115 \ r^2) + 0.426765 \ \exp(-1.487193 \ r^2)$$

$$\chi_{Be}^{2s} = -0.087482 \ \exp(-1.314833 \ r^2) + 0.117015 \ \exp(-0.305539 \ r^2) + 0.088313 \ \exp(-0.099371 \ r^2)$$

$$\chi_{Be}^{2p} = 0.136444 \ \exp(-1.314833 \ r^2) + 0.177987 \ \exp(-0.305539 \ r^2) + 0.049442 \ \exp(-0.099371 \ r^2)$$

**Antisymmetry of the wavefunction**   The Pauli exclusion principle as another fundamental constraint of quantum mechanics states that no two particles can be in the same quantum state (i.e. be described by the same wavefunction $\psi$)at the same time. As a consequence, the total wavefunction composed from the individual wavefunctions of fermions has to be antisymmetric under permutation of the fermions, and in this case specifically for electrons:

$$\psi_1(e_1)\psi_2(e_2) = -\psi_1(e_2)\psi_2(e_1)$$

where $\psi(e)$ denotes the quantum state of an electron $e$.

There are in general two options to introduce this condition into the mathematical description of the system, namely **first quantization**, in which the antisymmetry property is encoded within the wavefunction, and **second quantization**, which encodes the antisymmetry condition into the Hamiltonian via the algebra of anticommutators. In the realm of quantum computing, second quantization is considered as the standard for many quantum algorithms, and was accordingly used throughout this thesis.

**Constructing the molecular basis**   Based on the calculated orbitals, one can go about representing the full statespace of the wavefunction in the computational basis, satisfying the fermionic antisymmetry relations. In the simple method, called

*first quantization*, one enforces these through the Slater determinant of the wave-functions for individual electrons:

$$\Psi(e_1, \ldots, e_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(e_1) & \chi_2(e_1) & \cdots & \chi_N(e_1) \\ \chi_1(e_2) & \chi_2(e_2) & \cdots & \chi_N(e_2) \\ \vdots & \cdots & \ddots & \vdots \\ \chi_1(e_N) & \chi_2(e_N) & \cdots & \chi_N(e_N) \end{vmatrix} \tag{1.3}$$

Expanding this determinant gives a linear combination with factor $\frac{1}{\sqrt{N!}}$ of all $N!$ exchange permutations over a tuple $(\chi_1, \chi_2, \cdots, \chi_N)$, where the permutation $\sigma$ is in terms of the indices of the individual functions. This linear combination with equal coefficients for all terms encodes the antisymmetry of the wavefunction due to the **indistinguishability principle**, since no configuration of one wavefunctions should be more probable than any other at this point as the labelling of the electrons is arbitrary.

However, this represents a cumbersome way of producing the wavefunction asymmetry, as the linear combination of these wavefunctions for $N$ particles grows in the order of $N!$. Therefore a different way of encoding antisymmetry called *second quantization* was used. In second quantization, the antisymmetry condition is enforced onto the operators rather than the wavefunctions, as will be seen later in this section, whereas the wavefunction will simply be represented in the *occupation number representation* as follows:

$$\Psi = (\chi_1, \chi_1, \cdots, \chi_k) \longrightarrow (\chi_1 \chi_1 \cdots \chi_k) \longrightarrow |n_1 n_1 \cdots n_k\rangle \tag{1.4}$$

where $n_i$ in the ket signalizes that the orbital $i$ is occupied by $n_i$ electrons. For a $N$ electron system, $\sum_i n_i = N$ must hold. On top of that, for fermionic systems, $n_i \in \{0, 1\}$ must hold, i.e. an orbital can either be unoccupied or occupied by exactly 1 electron. Of course one has to decide on how many orbitals one includes in the representation. This can be decided by looking at the amount of electrons

that shells in the respective atoms can hold:

- H only has the 1s shell which can contain 2 electrons

- Be in the ground state has, 1s, 2s and 2p shells, the latter of which can contain 2 and 6 electrons, respectively.

In total we get $1 \times 1s, 1 \times 2s, 3 \times 2p$ shells and thus 14 available electron orbitals, a mapping which is consistent with others in the literature, for example as in [22]. These orbital configurations $\chi_i, \cdots, \chi_k$ can then simply be mapped onto the computational basis as follows:

$$\chi_1 \longleftrightarrow \underbrace{|10000000000000\rangle}_{14 \text{ digits}}$$

$$\chi_2 \longleftrightarrow |01000000000000\rangle$$

$$\vdots$$

$$\chi_{14} \longleftrightarrow |00000000000001\rangle$$

Note that the basis kets themselves are not actually valid states of our system, since $\sum_i n_i = N$ is not satisfied. Instead, they merely *span* the space of the possible states.

**Second quantization of the electronic Hamiltonian**

The basic idea in second quantization is that the antisymmetry of the wavefunction is encoded into the representation of the Hamiltonian rather than into the wavefunction state. This is done by introducing the so-called **creation**- and **annihilation operators** with an corresponding operator algebra as described in Table 1.2.

The electronic structure Hamiltonian $\hat{H}_e$ in the second quantized form can then be written as [30]:

| Second quantization | | |
|---|---|---|
| Creation operator | $a_i^\dagger \lvert j_1, \ldots, 0_i, \ldots, j_n \rangle = \Gamma_i^{\mathbf{j}} \lvert j_1, \ldots, 1_i, \ldots, j_n \rangle$ | with $\Gamma_i^{\mathbf{j}} = \prod_{k=1}^{i-1} (-1)^{j_k}$ |
| | $a_i^\dagger \lvert j_1, \ldots, 1_i, \ldots, j_n \rangle = 0$ | |
| Annihilation Operator | $a_i \lvert j_1, \ldots, 1_i, \ldots, j_n \rangle = \Gamma_i^{\mathbf{j}} \lvert j_1, \ldots, 0_i, \ldots, j_n \rangle$ | |
| | $a_i \lvert j_1, \ldots, 0_i, \ldots, j_n \rangle = 0$ | |
| Commutation relations | $\{a_i^\dagger, a_j\} = \delta_{ij} \mathbf{1}$ | where $\mathbf{1} = \sum_i \lvert i \rangle \langle i \rvert$ the identity |
| | $\{a_i a_j\} = 0$ | |

Table 1.2: Overview of anticommutator algebra [30]

$$\hat{H} = \sum_{p,q} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \tag{1.5}$$

where the first term describes the kinetic energy of the individual electrons in interaction with the electric field of the nuclei, and the second term describes the interactions between different electrons. Specifically, $h_{pq} a_p^\dagger a_q$ describes the process of an electron from orbital $q$ moving to orbital $p$, whereas the term $h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$ describes the process of two electrons from orbitals $r, s$ interacting with each other and ending up in orbitals $p, q$ [1]. The coefficients $h_{pq}, h_{pqrs}$ can be found by evaluating the **one-electron integrals** and the **two-electron integrals** respectively:

**Molecular integrals** The coefficients $h_{pq}$ and $h_{pqrs}$ are calculated by evaluating the molecular integrals defined as follows:

$$h_{pq} = \int \chi_p^*(\mathbf{x}) \left( -\frac{\nabla^2}{2} - \sum_I \frac{Z_I}{\lvert \mathbf{r} - \mathbf{R} \rvert} \right) \chi_q(\mathbf{x}) d\mathbf{x} \tag{1.6}$$

$$h_{pqrs} = \int \int \frac{\chi_p^*(\mathbf{x_1} \chi_q^*(\mathbf{x_2}) \chi_r(\mathbf{x_2}) \chi_s(\mathbf{x_2})}{\lvert \mathbf{r_1} - \mathbf{r_2} \rvert} d\mathbf{x_1} d\mathbf{x_2} \tag{1.7}$$

These integrals have been evaluated for all the values of p,q,r,s using PySCF [25], the exact values can be found in the supplementary materials.

**Jordan Wigner Transformation** After the form of the Hamiltonian was fully derived and the necessary coefficients were calculated, the question remains of how to

map the full Hamiltonian $\hat{H}_e$, that acts on the $N$-dimensional Hilbert space $\mathcal{H}$, onto a universal set of qubit gates. This can be achieved by decomposing the Hamiltonian according to the *Jordan-Wigner transformation* as described in [27]. Therefore, the creation operator $\hat{a}_j^\dagger$ and the annihilation operator $\hat{a}_j$ acting on the $j$th orbital in the occupation number representation can be decomposed by the Jordan-Wigner transformation as follows:

$$\hat{a}_j^\dagger \to \frac{\sigma_x^j - i\sigma_y^j}{2} \otimes \sigma_z^0 \otimes \ldots \otimes \sigma_z^{j-1} = \left(\prod_{k=1}^{n-1} \sigma_z^k\right) \sigma_+^j \tag{1.8}$$

$$\hat{a}_j \to \frac{\sigma_x^j + i\sigma_y^j}{2} \otimes \sigma_z^0 \otimes \ldots \otimes \sigma_z^{j-1} = \left(\prod_{k=1}^{n-1} \sigma_z^k\right) \sigma_-^j \tag{1.9}$$

where $\sigma_x^k, \sigma_y^k, \sigma_z^k$ are the Pauli gates and $\sigma_+^k$ and $\sigma_-^k$ are the creation and annihilation operator acting on a single qubit on position $k$ respectively. This mapping of the ladder operators inherently encodes the antisymmetry required by second quantization.

Using these identities 1.8, 1.9 onto the Hamiltonian $\hat{H}_e$ 1.5, it then can be decomposed as shown by Ovrum and Jensen [17]. Considering the one-electron and two-electron interaction term separately, one gets:

$$\hat{H}_1 = \sum_{pq} h_{pq} a_p^\dagger a_q = \sum_{pp} h_{pp} a_p^\dagger a_p + \sum_{p<q} (a_p^\dagger a_q + a_q^\dagger a_p) \tag{1.10}$$

$$\hat{H}_2 = \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \tag{1.11}$$

For conciseness, the rest of the decomposition is not explained in detail here. The full derivation can be found in the paper by Ovrum and Jensen.

# 1.3 The VQE algorithm

The VQE algorithm was first introduced by Peruzzo et al. [18] and presents an iterative quantum-classical hybrid approach for calculating the eigenstates of given Hamiltonians, most prominently used in quantum chemistry with a wide array of potential use cases in drug discovery [3], material science [12]and various other scientific areas [7][4].

## 1.3.1 Overview

The variational quantum eigensolver [18] is a hybrid quantum classical algorithm which is used to iteratively approach the eigenvalue as an optimization problem.

The Hamiltonian of interest $\hat{H}_e$ will be mapped onto unitary operations, which in term can be mapped onto a universal set of quantum gates which can be run on a quantum computer. In this case this will be the single and double excitation fermionic Hamiltonian which is already derived from the many body problem in section 1.2.4.

The general procedure of the VQE algorithm is summarized below:

1. A register of qubits is prepared in a well defined reference state $|\psi_{\text{ref}}\rangle$. It is vital that the size of the register, i.e. the number of qubits fits the required requirements imposed by the encoding of the Hamiltonian.

2. A parameterized Ansatz state is prepared. This parameterized state will throughout the algorithm runtime be altered through an optimization process to approach the desired state.

3. The expectation value of Hamiltonian Operator, i.e. the energy state of the prepared state is being measured.

4. A classical computation of the measurements is being done to compute the parameters for the next iterations.
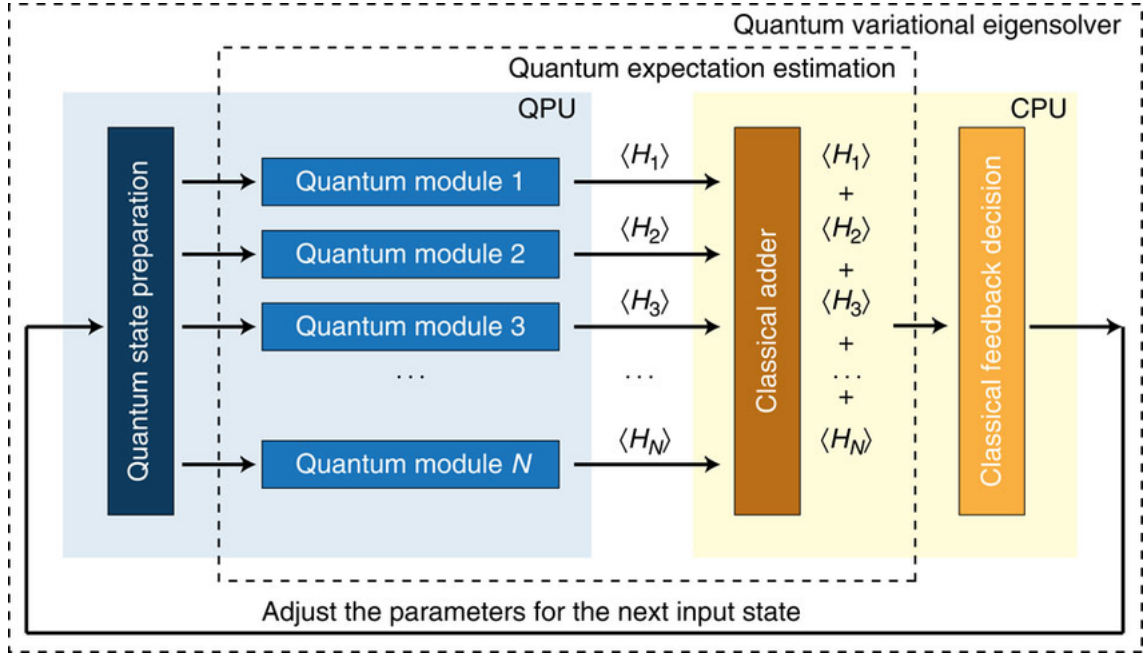
Figure 1.7: Schema of the VQE algorithm [18]

5. The above steps are performed until a defined converge threshold is reached.

## 1.3.2 Ansatz selection

For the Ansatz state selection, numerous different possibilities exist, each of which have different advantages and disadvantages related to the problem at hand, but also the circuit depth, i.e. number of gate operations of the corresponding quantum circuit representation. For n-body problems usually the k-UCC Ansatz inspired from principles of unitary coupled cluster theory promises good results [23] and was therefore chosen. Other possible Ansatz states include the Hardware Efficient Ansatz (HEA), or the Two-Local Ansatz. An overview can be found in [27].

## 1.3.3 Optimization

Likewise there exists a wide array of different classical optimizers for the VQE [27]. They can roughly be grouped into three categories:

- First order gradient optimizers, such as *Gradient Descent, ADAM* and variations of these.

- Second order gradient optimizers, such as the *BFGS algorithm*

- Gradient-free optimizers, for example the *Nelder-Mead algorithm*

For a detailed overview of the optimizers aswell as the respective dedicated use cases, one can consult [27]. Based on the information that the *Nelder-Mead algorithm* performs well on low-noise environments and some empiric modelling on smaller problems, I chose it as the optimizer for the concrete implementation.

## 1.4 The QPE algorithm

In general terms, the phase estimation algorithm enables the estimation of the eigenvalue $\lambda$ of an unitary operator $U$ and corresponding to the eigenstate $|\lambda\rangle$ , where $\lambda$ can also mathematically be represented as $e^{2\pi i\phi}$. Thus an estimation of the phase parameter $\phi$ will also allow for the calculation of the full eigenvalue. In the case of calculating the ground state for the Hamiltonian $\hat{H}$, the unitary $U$ will again be the the evolution operator of the Hamiltonian, $U = e^{-i\hat{H}t}$.

### 1.4.1 Overview

Firstly, a state register $|S\rangle$ is prepared, intended to store the eigenstate $|\lambda\rangle$ of the unitary $U$, as well as a readout register $|R\rangle$ with the purpose of storing the eigenvalue. Since the register consists of qubits, the representation , resulting in the total register state $|R\rangle \otimes |S\rangle$. Typically the readout register will initially consist of $k$ qubits in the $|0\rangle$ state. These readout qubits are then brought into an equal superposition of $|0\rangle$ and $|1\rangle$ by applying a Hadamard transform.
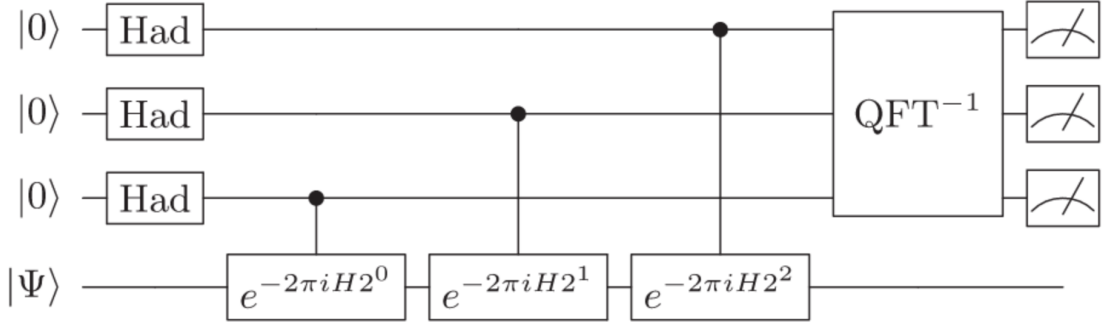
Figure 1.8: Schema of the QPE algorithm [15]

$$|R\rangle \otimes |S\rangle \to |0\rangle^{\otimes k} |\psi\rangle \xrightarrow{H^{\otimes k}} \left[\frac{1}{\sqrt{2^k}} \bigotimes_{i=0}^{k}(|0\rangle + |1\rangle)\right] |\psi\rangle \qquad (1.12)$$

Then for each of the control qubits $j \in \{0, ..., k\}$, the jth power of the unitary operator $U \equiv e^{i\hat{H}tj}$, is conditionally (labelled as $C - U$ operator) applied onto the state register $|S\rangle$. Since $|\psi\rangle$ is an eigenstate of $U$, the action of the $C - U$ operator will have one of the following two effects, depending onto which state the ith qubit collapses:

1. If qubit i collapses into $|0\rangle$, no action (i.e. the identity operator) is applied

2. If qubit i collapses into $|1\rangle$, the action onto the eigenstate is equivalent to a multiplication with the eigenvalue $e^{i\pi\phi j}$

Considering only the case of a single jth control qubit for conciseness, the state will then be:

$$\frac{1}{\sqrt{2^k}}(|0\rangle |\psi\rangle + |1\rangle e^{i\pi\phi j} |\psi\rangle) = \frac{1}{\sqrt{2^k}}(|0\rangle + e^{i\pi\phi j} |1\rangle) |\psi\rangle \qquad (1.13)$$

Thus, even though the controlled operation was applied onto the state register $|S\rangle$, the phase will be observable within the readout register $|R\rangle$. This is called a phase kickback in the literature.

To recover the relevant phase information $\phi$, a subprocess called inverse Quantum Fourier Transformation (inverse QFT) will be applied. Inverse QFT is discussed briefly in section 1.4.3. For the purpose of this overview, it suffices to clarify that the result of the process will be that the readout register $|R\rangle$ will store a binary representation of $\phi$. Thus a simple measurement in the computational basis will give the value $\phi$ as desired.

## 1.4.2 Register preparation

Quantum phase estimation requires an initial preparation of the state register $|S\rangle$ in some eigenstate of the Hamiltonian $\hat{H}_e$. This can be achieved by a method called adiabatic state preparation. The process can be summarized in 5 steps [24]:

1. Calculation of the RHF (Restricted Hartree-Fock) molecular integrals as seen by equation 1.6, 1.7.

2. Division of the electronic Hamiltonian 1.2 into the fermionic representation as described by equations 1.10, 1.11.

3. Prepare the necessary computational parameters for simulating the Hamiltonian in question. This may consist of doing Trotterization of the Hamiltonian as first developed mathematically by Trotter et al.[29] and described in more practical terms for quantum computing purposes for example in [16].

4. Computation of the trotterized Hamiltonian at each discrete timestep and construction of a corresponding quantum circuit.

5. Encoding of the starting wave function onto the state register $|S\rangle$, which can be done for the occupation number representation by applying X gates to $N$ qubits, where $N$ is the number of electrons occupying the molecular orbitals.

A more mathematically rigorous description of the process can be found in [24]. Already however it should be noted that these steps for adiabatic ground state preparation are closely related to the encoding scheme of the electronic Hamiltonian

within the VQE algorithm.

### 1.4.3 The inverse QFT subroutine

Quantum Fourier Transform is a quantum algorithm derived from the classical discrete Fourier Transform. It can mathematically represented as follows [15]:

where $|j\rangle$ is the input state, $|k\rangle$ is the transformed state and $N$ is the number of states. Thus the action of the transformation is to retrieve information of $|j\rangle$ by extracting it into phase information $e^{2\pi ijk/N}$, and changing the state to $k$ in the process.

The *inverse* QFT simply refers to the opposite action, i.e. encoding phase information of some state $|k\rangle$ into a state $|j\rangle$. Therefore it can be mathematically represented as:

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \longrightarrow |j\rangle \tag{1.14}$$

The usefulness of the inverse QFT lies in the fact that phase information is not directly measurable, while quantum states on the other hand are. Therefore the transformation describes a way to retrieve phase information from the state space of the quantum computer. A detailed discussion of (inverse) QFT and its applications is given by Nielsen and Chuang [15].

## 1.5 Further reading

Quantum computing is a wide-ranging topic with ties deep into physics, computer science, mathematics. The best way of approaching the topic is therefore arguably dependent on the background of the reader at hand. A gentle introduction to the field from a computer science perspective is given by Hidary [9]. The definitive work in quantum computing is widely regarded to be Nielsen and Chuang's textbook [15].

*1 Introduction*

For basic intuition in quantum mechanics, Susskind [26] offers a simple introduction, with far more mathematically precise and advanced topics covered in [28] and [21]. A solid starting point for approaching the many-body Hamiltonian problem and electronic structure theory is the paper by McArdle et al. [13]. Additional information on electronic structure theory, molecular basis sets and related topics are given in the work by [8].

A great overview on the VQE and many of its modifications was published by Tilly et al. [27]. A concrete example for ground state evaluation of molecule using the VQE has been shown by [11]. For the QPE, a general outline of the algorithm has also been given by Nielsen and Chuang [15], a concrete mapping to molecular energy calculation has been given for example by Aspuru-Gzik et al.[2].

# 2 Technology

## 2.1 Python

Python 3 is a strongly, dynamically typed programming language intended for quick and easy-to-read development. It has become one of the main programming languages in many scientific fields and provides a large number of non-commercial libraries for scientific computing. Especially in quantum computing, Python is the de-facto standard language due to the popularity of quantum computing packages such as qiskit or cirq.

## 2.2 Qiskit

Qiskit [20] is an open-source framework created and maintained by IBM which offers both high-level and low-level support for programs entailing quantum computing aspects. The centerpiece of API is the QuantumCircuit object, which can be equipped with a wide variety of gates to create the respective quantum algorithms. It also offers strong support of linear algebra tools to create arbitrary unitary operators and offers possibilities to transform said operators into the circuit formalism.

This thesis project uses Qiskit as centerpiece for building up the described algorithms, run them by simulation and visualize results.

## 2.3 PySCF

PYSCF [25] is an open source library composed by electronic structure modules written in Python and C.

Within this project, PYSCF was used to numerically calculate the electron interaction integrals of the molecular orbitals as described in section 1.2.4.

## 2.4 Basis Stack exchange

Basis stack exchange (BSE) [19] is an online collection of coefficients for atomic and molecular orbitals of various kinds.

BSE in the context of this thesis was being used to obtain the required STO-3G coefficients for the molecular orbitals of the $BeH_2$ molecule as shown in section 1.2.4

## 2.5 Jupyter notebook

Jupyter notebook is a web applications running in the browser for the creation of code and computational documents. Other than code execution, it enables Markdown and LaTeX execution and makes it possible to render images right next to code cells. In the scope of the thesis, Jupyter notebook served as frontend to put the pieces together for the final calculations.

# 3 Methodology

The full implementation source code is available in the corresponding Github repository `https://github.com/ChristophMoserGreil/BA_Thesis`. The implementation of both the VQE and the QPE were done in Python using the Qiskit framework. However, the internal handling and the level of abstraction vary between the two algorithms. This is due to the differing level of support for each of the algorithms by the existing libraries as well as due to the scope of the problem at hand.

This chapter briefly outlines the approach that was taken to implement the problem, highlights key parts of the sourcecode and discusses encountered challenges.

## 3.1 Implementation details

The implementation philosophy of the sourcecode was picked with the focus on the experimental nature of the thesis in mind. Therefore, generally, the ability to make quick changes and develop prototypes was regarded as more important than a high layer of abstraction.

Figure 3.1 gives an qualitative overview of how the different components were structured for the project. It should however be pointed out that the figure does not adhere to a standard such as UML or similar, and also that the pictured components can not in every case be translated to classes or software modules one to one.
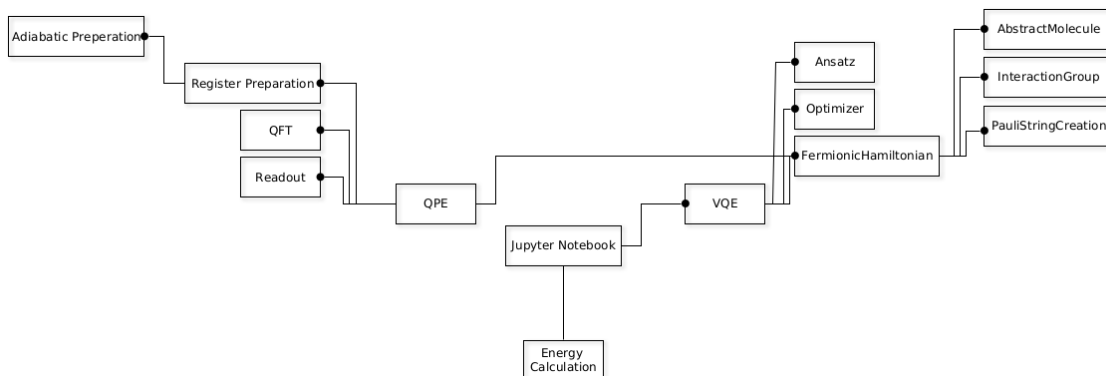
Figure 3.1: Qualitative overview of the Project structure

## 3.1.1 Abstraction

Nevertheless, basic abstraction guidelines of software engineering were employed in order to achieve reusability and testability. For example, an abstract molecule type has been introduced as interface between the custom software stack and the libraries handling the parsing of molecular orbital calculation. This decision improved testability by also allowing for the calculation of smaller molecules such as H2, which requires substantially lower runtime for a full calculation compared to the rather big BeH2 molecule. The concrete creation of the molecule was then done in a subclass. The listing 3.1.1 shows the configurations made for the concrete molecule. The files that are loaded files in the .gbs format storing the coefficients and exponents defining the molecular orbitals as described in 1.2.4.

```python
@staticmethod
def _pyscf_create_beh2():
    """Function to create the beh2 molecule as pyscf object
    using the MO basis defined in the gbs file"""
    molecule = gto.Mole()
    # geometry of the molecule
    molecule.atom = '''
        Be 1. 0. 0.
        H  0. 0. 0.
        H  2. 0. 0  '''
    # B for Bohr referring to Bohr radius
    molecule.unit = 'B'
    # custom sto-3g basis is being parsed here
    molecule.basis = {
        # absolute file paths used
        'Be': parse_gaussian.load(
        '/home/.../src/resources/BeH2_basis_new.gbs', 'Be'),
        'H': parse_gaussian.load(
        '/home/.../src/resources/BeH2_basis_new.gbs', 'H')
    }
    molecule.spin = 0
    molecule.build()
    return molecule
```

Listing 1: Molecule encoding of BeH2 using PySCF

## 3.1.2 Testing

Testing was done in part by doing full system tests throughout the process of fully simulating the calculation. Due to growing complexity, especially within the creation of the representation matrices of the fermionic hamiltonian, additionally unit tests were employed for some especially complex parts of the program. This was done by Pythons native unittest environment, which proved powerful enough for the intended purposes. Listing 3.1.2 shows an excerpt of the unittests used to verify the correct creation of the fermionic interaction Hamiltonian.

```python
def test_diagonal_pauli_generator_2qubit(self):
    num_qubits = 2
    coeffs_sample = np.array([[0.3, 1], [3.4, 8.2]])
    # np.repeat(x, n) creates a new list, where each x is
    #   repeated n times
    correct_coeffs = np.repeat(np.array([coeffs_sample[k, k] * 1
    #   / 2 for k in range(num_qubits)]), 2)
    correct_op = SparsePauliOp(['II', 'ZI', 'II', 'IZ'],
    #   coeffs=correct_coeffs)
    result = diag(2, coeffs_sample)
    self.assertEqual(result, correct_op)


def test_diagonal_pauli_generator_3qubit(self):
    num_qubits = 3
    coeffs_sample = np.array([[2.2, 1.0, 3.2], [8.9, 4.1, 0.98],
    #   [0.63, 2.2, 0.82]])

    correct_coeffs = np.repeat(np.array([coeffs_sample[k, k] * 1
    #   / 2 for k in range(num_qubits)]), 2)
    correct_op = SparsePauliOp(['III', 'ZII', 'III', 'IZI',
    #   'III', 'IIZ'], coeffs=correct_coeffs)

    result = diag(3, coeffs_sample)
    self.assertEqual(result, correct_op)
```

Listing 2: Excerpt from the FermionicInteraction Testsuite

### 3.1.3 Other Challenges

The primary reoccurring challenge across the entire project was the handling of the high level of complexity resulting from in translating the mathematical formulae of the consulted papers into sourcecode entailing the same behaviour. To help achieve this, a number of techniques were employed, as follows:

- Typing

  Despite using Python, a weakly typed language, typing hints greatly improved the quality of the development process by making the intended behaviour of functions more clear. Subjectively, the saved time in this fashion far outweighed the needed effort to add type hints.

- Testing

  As already mentioned in section 3.1.2, unit testing proved vital in the more complex areas of the code to ensure the intended behaviour was properly reached in the code.

- Function decomposition

  A more lucid code base was also reached by ensuring that the Single responsibility principle was closely followed, and as a result the functions could be kept rather small, with necessary sub-calculations often relocated into dedicated classes or utility functions.

# 4 Results and Discussion

This chapter shows the results computed by the algorithm implementations and discusses potential deviations from experimental values.

## 4.1 VQE calculation on the BeH2 molecule

The VQE algorithms convergence plot for the parameters chosen as specified in chapter 3 is shown in the figure below. For a bond length of 1.0 Ångström ($10^{-10}m$) for each of the Be-H bonds, a final result of $-41.20632$ hartree has been computed. This result poses a deviation from the experimental results of around $-15.4$ hartree as published by Kandala et al. [10] and was therefore subject to an error. The source of the error could not be determined at the time of writing the thesis, however it is suspected to arise from an incorrect mapping of the molecular orbitals for larger systems, resulting in the calculation of erroneous interaction integrals.

## 4.2 VQE Calculation on the H2 molecule

As a consequence of the error, the calculation was repeated for the case of a different molecule, the much smaller and simpler Hydrogen (H2) molecule. For a bond length of 0.735 Ångström, the final value of $-1.85728$ hartree was computed, which does match the values that can be found in the literature. Figure 4.1 shows the optimization convergence for the VQE approximation.
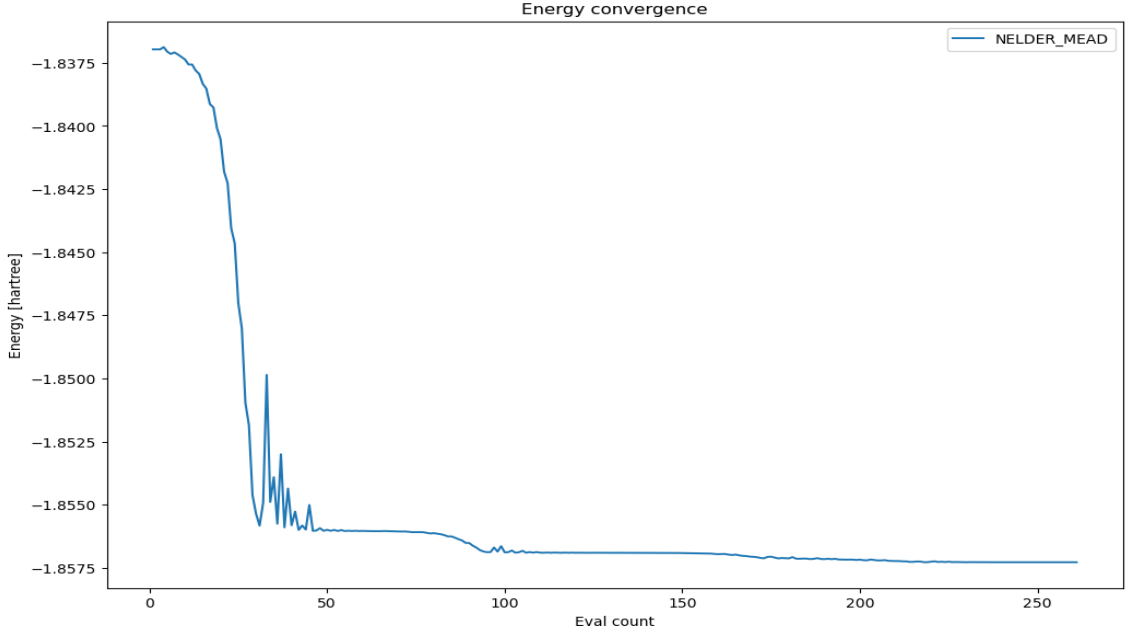
Figure 4.1: VQE ground state approximation for the H2 molecule

## 4.3 QPE calculation on the BeH2 molecule

The final result calculated via the QPE was $-42.85631$ hartree, which represents a deviation from the expected value, which for the specified bond length would be at around $-15.4$ hartree [10]. While the deviation is large, the erroneus value is somewhat consistent wit the result of the VQE calculation on BeH2. Therefore it is likely that the error stems from a bug within the Hamiltonian mapping that only arises for larger systems.

## 4.4 QPE calculation on the H2 molecule

Again, the simpler case of the H2 molecule was considered by the QPE algorithm as well. Again using the bond length of 0.735 Ångström for that calculation, an energy value of $-1.85713$ hartree was computed. Therefore, the results of the calculations of the groundstate energies of H2 do match each other closely and also agrees with similar calculations, for example the simulation done by Du et al. [6].

# 5 Conclusion and outlook

This thesis showed the pathway of encoding the simulation of a n-body system starting from the general Schrödinger equation via multiple steps (including the Born-Oppenheimer approximation, transformation into occupation representation via molecular orbitals, Second Quantization, Jordan-Wigner encoding) into a representation as unitary operator decomposed into a linear combination of weighted Pauli strings, which can be utilized by current quantum computer software. Additionally, two quantum algorithms for solving the Schrödinger Equation, i.e. calculating the ground state energy, the QPE and the VQE algorithm, were employed on a quantum computer simulation software for concrete examples on the molecules H2 and BeH2.

While the above mentioned main goal was achieved, an initial smaller subgoal of the thesis, additionally developing the algorithms from first principles simply on top of elementary algebra, had to be abandoned due to time and scope limitations.

## 5.1 Outlook

Both the VQE and the QPE algorithm are promising quantum algorithms with a plethora of potential applications in the context of quantum chemistry, material science, drug discovery and the likes. While QPE promises a bigger speedup and thus the feasible computation of large molecules, the current limitations of quantum hardware mean that the use for practical purposes is still some times in the future. On the other hand, VQE due to its relatively short circuit depth, will be

more readily available even within the current era of physical quantum computers, largely dominated by NISQ (Noisy intermediate scale quantum) devices.

## 5.2  Future Work

Albeit these aspects were not included in the thesis, it should be noted that the most compelling challenges for quantum computation for now remain on the side of the physical implementations of quantum computers. Future practical applications still require substantial improvements in the number of qubits, decoherence times, error resilience and gate fidelities, depending on the physical implementation paradigm. Aside from these goals concerning the physical realization, a number of shorter term goals have been identified:

- Due to the hardware requirements of QPE exceeding the momentarily available state, best practices for application of QPE onto molecular structure problems are rare so far. In the medium-term, an overview of best practices analogous to [27] for the VQE would benefit future research greatly.

- While the asymptotic runtime for the most frequently used VQE techniques are known [27], a more precise classification via benchmarking processes is desirable. First steps in this direction have already been taken, for example with the QuantMark framework [14].

- For both the QPE (although to a lesser degree) the VQE approximation technique, there is currently a lack of rigorous error analysis.

# Bibliography

[1] Microsoft Azure second quantization. `https://learn.microsoft.com/en-us/azure/quantum/user-guide/libraries/chemistry/concepts/second-quantization`. Accessed: 2023-05-24.

[2] ASPURU-GUZIK, A., DUTOI, A. D., LOVE, P. J., AND HEAD-GORDON, M. Simulated Quantum Computation of Molecular Energies. *Science 309*, 5741 (Sept. 2005), 1704–1707. Publisher: American Association for the Advancement of Science.

[3] CAO, Y., ROMERO, J., AND ASPURU-GUZIK, A. Potential of quantum computing for drug discovery. *IBM Journal of Research and Development 62*, 6 (Nov. 2018), 6:1–6:20. Conference Name: IBM Journal of Research and Development.

[4] CAO, Y., ROMERO, J., OLSON, J. P., DEGROOTE, M., JOHNSON, P. D., KIEFEROVÁ, M., KIVLICHAN, I. D., MENKE, T., PEROPADRE, B., SAWAYA, N. P. D., SIM, S., VEIS, L., AND ASPURU-GUZIK, A. Quantum Chemistry in the Age of Quantum Computing. *Chemical Reviews 119*, 19 (Oct. 2019), 10856–10915. Publisher: American Chemical Society.

[5] DEMTRÖDER, W. *Molecular Physics: Theoretical Principles and Experimental Methods*. Physics textbook. Wiley, 2008.

[6] DU, J., XU, N., PENG, X., WANG, P., WU, S., AND LU, D. NMR Implementation of a Molecular Hydrogen Quantum Simulation with Adiabatic State

Preparation. *Physical Review Letters 104*, 3 (Jan. 2010), 030502. Publisher: American Physical Society.

[7] EUROPEAN COMMISSION. DIRECTORATE GENERAL FOR RESEARCH AND INNOVATION. *Recognising the importance of software in research: Research Software Engineers (RSEs), a UK example.* Publications Office, LU, 2019.

[8] HELGAKER, T., JORGENSEN, P., AND OLSEN, J. *Molecular Electronic-Structure Theory.* John Wiley & Sons, Aug. 2014. Google-Books-ID: lNVL-BAAAQBAJ.

[9] HIDARY, J. D. *Quantum Computing: An Applied Approach.* Springer International Publishing, Cham, 2019.

[10] KANDALA, A., MEZZACAPO, A., TEMME, K., TAKITA, M., BRINK, M., CHOW, J. M., AND GAMBETTA, J. M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature 549*, 7671 (Sept. 2017), 242–246.

[11] KÜHN, M., ZANKER, S., DEGLMANN, P., MARTHALER, M., AND WEISS, H. Accuracy and Resource Estimations for Quantum Chemistry on a Near-Term Quantum Computer. *Journal of Chemical Theory and Computation 15*, 9 (Sept. 2019), 4764–4780.

[12] LORDI, V., AND NICHOL, J. M. Advances and opportunities in materials science for scalable quantum computing. *MRS Bulletin 46*, 7 (July 2021), 589–595.

[13] MCARDLE, S., ENDO, S., ASPURU-GUZIK, A., BENJAMIN, S. C., AND YUAN, X. Quantum computational chemistry. *Reviews of Modern Physics 92*, 1 (Mar. 2020), 015003. Publisher: American Physical Society.

[14] MEIJER VAN DE GRIEND, A., AND NURMINEN, J. K. QuantMark: A Benchmarking API for VQE Algorithms. *IEEE Transactions on Quantum Engi-*

*neering 3* (2022), 1–6. Conference Name: IEEE Transactions on Quantum Engineering.

[15] NIELSEN, M. A., AND CHUANG, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.

[16] ORTIZ, G., GUBERNATIS, J. E., KNILL, E., AND LAFLAMME, R. Quantum algorithms for fermionic simulations. *Physical Review A 64*, 2 (July 2001), 022319. Publisher: American Physical Society.

[17] OVRUM, E., AND HJORTH-JENSEN, M. Quantum computation algorithm for many-body studies, May 2007. arXiv:0705.1928 [quant-ph].

[18] PERUZZO, A., MCCLEAN, J., SHADBOLT, P., YUNG, M.-H., ZHOU, X.-Q., LOVE, P. J., ASPURU-GUZIK, A., AND O'BRIEN, J. L. A variational eigenvalue solver on a quantum processor. *Nature Communications 5*, 1 (Sept. 2014), 4213. arXiv:1304.3061 [physics, physics:quant-ph].

[19] PRITCHARD, B. P., ALTARAWY, D., DIDIER, B., GIBSON, T. D., AND WINDUS, T. L. New Basis Set Exchange: An Open, Up-to-Date Resource for the Molecular Sciences Community. *Journal of Chemical Information and Modeling 59*, 11 (Nov. 2019), 4814–4820. Publisher: American Chemical Society.

[20] QISKIT CONTRIBUTORS. Qiskit: An open-source framework for quantum computing, 2023.

[21] SAKURAI, J., AND NAPOLITANO, J. *Modern Quantum Mechanics.* Cambridge University Press, 2020.

[22] SETIA, K., CHEN, R., RICE, J. E., MEZZACAPO, A., PISTOIA, M., AND WHITFIELD, J. D. Reducing Qubit Requirements for Quantum Simulations Using Molecular Point Group Symmetries. *Journal of Chemical Theory and Computation 16*, 10 (Oct. 2020), 6091–6097. Publisher: American Chemical Society.

*Bibliography*

[23] STAIR, N. H., AND EVANGELISTA, F. A. Simulating Many-Body Systems with a Projective Quantum Eigensolver. *PRX Quantum 2*, 3 (July 2021), 030301. Publisher: American Physical Society.

[24] SUGISAKI, K., TOYOTA, K., SATO, K., SHIOMI, D., AND TAKUI, T. Adiabatic state preparation of correlated wave functions with nonlinear scheduling functions and broken-symmetry wave functions. *Communications Chemistry 5*, 1 (July 2022), 1–13. Number: 1 Publisher: Nature Publishing Group.

[25] SUN, Q., BERKELBACH, T. C., BLUNT, N. S., BOOTH, G. H., GUO, S., LI, Z., LIU, J., MCCLAIN, J. D., SAYFUTYAROVA, E. R., SHARMA, S., WOUTERS, S., AND CHAN, G. K.-L. PySCF: the Python-based simulations of chemistry framework. *WIREs Computational Molecular Science 8*, 1 (2018), e1340. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1340.

[26] SUSSKIND, L., FRIEDMAN, A., AND SUSSKIND, L. *Quantum mechanics: the theoretical minimum.* No. second volume in Theoretical minimum. Basic Books, a member of the Perseus Group, New York, 2014. OCLC: ocn853310551.

[27] TILLY, J., CHEN, H., CAO, S., PICOZZI, D., SETIA, K., LI, Y., GRANT, E., WOSSNIG, L., RUNGGER, I., BOOTH, G. H., AND TENNYSON, J. The Variational Quantum Eigensolver: A review of methods and best practices. *Physics Reports 986* (Nov. 2022), 1–128.

[28] TOWNSEND, J. S. *A modern approach to quantum mechanics*, 2nd ed ed. University Science Books, Sausalito, Calif, 2012.

[29] TROTTER, H. F. ON THE PRODUCT OF SEMI-GROUPS OF OPERATORS. 7.

[30] WHITFIELD, J. D., BIAMONTE, J., AND ASPURU-GUZIK, A. Simulation of electronic structure Hamiltonians using quantum computers. *Molecular Physics 109*, 5 (Mar. 2011), 735–750.