QUANTUM COMPUTING AND MANY-PARTICLE PROBLEMS

by

Heine Olsson Aabø

THESIS

for the degree of

MASTER OF SCIENCE

Faculty of Mathematics and Natural Sciences
University of Oslo
May 2020

Abstract

In quantum physics, finding solutions to the Schrödinger equation is an exponentially complex task being heavily studied throughout the last century. The Quantum phase estimation algorithm has been a major motivation for the development of quantum computers, offering significantly better scaling than any classical counterpart. Since practical uses of the phase estimation algorithm will deem unrealistic in the near future, other quantum algorithms have been sought out. The variational quantum eigensolver (VQE) is a recently developed quantum algorithm for calculating eigenvalues of Hermitian operators. It is expected to be a potential candidate in providing quantum advantage on applications simulating many-body systems.

In this work we have implemented the QPE and VQE algorithms, used on three different systems. We demonstrate that for executions on current quantum computers, the QPE is dominated by noise and not realistic for practical uses. Furthermore, we demonstrate that our implementation of the VQE algorithm is executable on real quantum computers. We make use of efficient strategies, such as grouping commuting operators and ordering of exponential operators to reduce the number of gates. Additionally, we introduce two new problem specific and hardware efficient state preparations, simply called RY and RYRZ. This enables us to execute ground state simulations of the hydrogen molecule and one dimensional quantum dot on a 5-qubit quantum computer.

We show that the unitary coupled cluster (UCC) ansatz provides better results than traditional coupled cluster for strong correlations. This might indicate that a unitary coupled cluster approach on quantum computers are a more suitable choice when exact diagonalization is infeasible. However, comparing the UCC ansatz to the RYRZ ansatz under the influence of noise, we argue that explicit flipping of qubits as in the RYRZ ansatz is more robust to noise than state preparation by the exponential operators in the UCC.

Acknowledgement

Contents

Chapter 1

Introduction

Simulating quantum systems on a classical computer is exponentially costly with respect to its size, and quickly becomes intractable for large systems. As a result scientists have turned their attention to approximate methods such as the Hartree-Fock method and improvements of this; so called post-Hartree-Fock methods such as Configuration Interaction [1] and the Coupled Cluster method [2]. Configuration Interaction in its full limit provides the exact solutions within the space spanned by a finite set of basis functions. However, it scales exponentially with the size of the system. Coupled Cluster offers a strategy for truncating the search space, while maintaining a respectable accuracy being computationally less costly. Although, the traditional formulation of this method is not variational and by so we can not typically trust its results for systems with strong correlations. A variational formulation of this method has been sought out, among others giving the Unitary Coupled Cluster. Unfortunately this has no natural truncation of the so called cluster operator [3], thus it has to introduce even further approximations.

During the last few decades quantum computing has seen a growing interest and hype within many fields of science, particularly quantum physics from which the initial proposal by Feynman was targeted [4]. Since then a wide range of methods have been developed which scales exponentially better than their classical counterpart. Most famously Shor's algorithm [5] provides a method for factoring integers that are thought intractable for classical computers. An essential part of this algorithm is the so called quantum phase estimation (QPE) algorithm, which gathered significant attraction for its application on finding eigenvalues of Hermitian operators. This has a direct application on the many-body problem scaling polynomial in the size of the system. However, for practical use it demands sophisticated quantum hardware that we can not expect within near future. As a result, other methods have been studied that distribute its workload on both quantum and classical computers. While the development of fault-tolerant quantum computers continues, such quantum-classical hybrid algorithms are thought to provide a so called quantum advantage. That is, providing speedups to tasks with bad scaling on classical computers. One such algorithm is the Variational Quantum Eigensolver (VQE), that replaces long depth circuits (as in QPE) with short depth circuits and an increase in the number of measurements. The original paper by Peruzzo et al [6], proposed the additional use of a Unitary Coupled Cluster (UCC) ansätz on applications within quantum chemistry. Using transformations such as the Jordan-Wigner transformation, quantum computers offer a more natural scheme for implementing such a complex state preparation than classical computers. Thus, we

can improve Hartree-Fock calculations on a quantum computer using the variational UCC ansätz, that has no efficient classical implementation.

Throughout the last decade several companies have entered the race for good quantum computers. However, quantifying the "goodness" of a quantum computer has led the metric *quantum volume*. It takes into consideration not only the number of qubits, but the connectivity between them and their coherence times. IBM states that their quantum computers have doubled their quantum volume every year the last few years, and that they expect this trend to continue. If so, much is to be expected from quantum computing in the near future. *Quantum supremacy* is defined as the era where a controlled quantum system, such as a quantum computer, can perform tasks that go beyond the capabilities of classical computers [7]. In order to demonstrate the advantage of quantum computation, efforts have been made towards (in some sense) impractical algorithms for which classical simulations are infeasible. In 2019 Google scientists claimed to have achieved quantum supremacy producing the output of a pseudo-random quantum circuit [8]. Though this met some initial critique in their definition, and by that misuse, of the term quantum supremacy. IBM claimed that their classical benchmark was inefficient, and that time estimates of the classical simulation could be reduced from approximately 10000 years to two and a half days. Nevertheless, development of quantum computers have reached a point where quantum supremacy is thought to be within reach. Arguably, quantum supremacy is more about the state of the quantum hardware rather than the quantum algorithms.

In this thesis we will examine the VQE and QPE algortihms on many-body quantum systems, particularly how the VQE can be implemented on current and near-term quantum computers in order to yield sufficient results. Current quantum computers involves many errors and short coherence times, therefore executable quantum circuits need efficient implementation with respect to qubit architecture along with the number of operations in the circuit. The executions of the algorithms will be both simulated on a classical computer as well as executed on a real quantum computer from IBM's Quantum Experience, with cloud access to a range of devices. This will be a 5-qubit computer with limited connectivity, where we need an optimal implementation of the UCC ansätz or the use of another hardware-efficient state preparation. We will start by introducing the necessary theoretical foundations, starting with quantum mechanics and many-body theory in chapter (2), before moving on to quantum computing and quantum algorithms in chapter (3) and (4). In part (II) we will summarize our implementation, and introduce the systems that are simulated in the results in part (III). Lastly we will give some concluding remarks along with prospects on future work.

Part I

Theory

Chapter 2

Quantum mechanics

We start by introducing some fundamental concepts from quantum mechanics, before we dwelve into the theory of quantum many-body physics.

2.1 Wavefunctions and quantum states

In quantum mechanics the state of an isolated physical system is at all times described by a wavefunction Ψ. We define this by the abstract vector $|\Psi\rangle$ in the so called Hilbert space, a complex vector space with a scalar product. Here we make use of Dirac's *braket* notation where a *ket* represents the state vector, that can be expanded in any complete set of basis vectors $|\psi_i\rangle$

$$|\Psi\rangle = \sum_i c_i |\psi_i\rangle, \tag{2.1}$$

where c_k are complex numbers. A *bra* $\langle\Psi|$ is a vector in the dual vector space related to $|\Psi\rangle$ through the mapping

$$|\Psi\rangle \rightarrow \langle\Psi| = \sum_i c_i^* \langle\psi_i|. \tag{2.2}$$

The expansion coefficients is given by the scalar product

$$\langle\psi_k|\Psi\rangle = \sum_i c_i \langle\psi_k|\psi_i\rangle = \sum_i c_i \delta_{ik} = c_k, \tag{2.3}$$

where $\langle\psi_i|\psi_j\rangle = \delta_{ij}$ is the Kronecker delta

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \tag{2.4}$$

for orthogonal basis vectors. We interpret the expansion coefficients in terms of the probability to find the system in state the $|\psi_k\rangle$

$$|\langle\psi_k|\Psi\rangle|^2 = |c_k|^2. \tag{2.5}$$

As a consequence we must have that the state vector has unit norm

$$\langle\Psi|\Psi\rangle = \sum_k |c_k|^2 = 1, \tag{2.6}$$

in order for the system to exist in any state. By combining equation (2.1) and (2.3)

$$|\Psi\rangle = \sum_k \langle\psi_k|\Psi\rangle\,|\psi_k\rangle = \sum_k |\psi_k\rangle\,\langle\psi_k|\Psi\rangle\,, \tag{2.7}$$

we can define the completeness relation for a complete set of basis vectors

$$\sum_k |\psi_k\rangle\,\langle\psi_k| = \hat{\mathbb{1}}, \tag{2.8}$$

where $\hat{\mathbb{1}}$ is the identity operator.

2.2   Quantum operators

Performing an operation on a state vector $|\Psi\rangle$ changes it into another $|\Psi'\rangle$, with the transformation being described by an operator $\hat{A}$

$$\hat{A}|\Psi\rangle = \big|\Psi'\big\rangle. \tag{2.9}$$

The dual vector of $|\Psi'\rangle = \hat{A}|\Psi\rangle$ is written in terms of the adjoint operator

$$\big\langle\Psi'\big| = \langle\Psi|\hat{A}^\dagger. \tag{2.10}$$

being the conjugate transpose of $\hat{A}$. Using the completeness relation we can expand $\hat{A}$ in a complete set of basis vectors

$$\hat{A} = \sum_i |\psi_i\rangle\,\langle\psi_i|\hat{A}\sum_j \big|\psi_j\big\rangle\big\langle\psi_j\big| = \sum_{ij}\langle\psi_i|\hat{A}\big|\psi_j\big\rangle\,|\psi_i\rangle\big\langle\psi_j\big| = \sum_{ij}A_{ij}\,|\psi_i\rangle\big\langle\psi_j\big|, \tag{2.11}$$

where A_{ij} are elements in the matrix representation of $\hat{A}$. We can let $\hat{A}$ act on the state vector $|\Psi\rangle$ expanded in the same basis set

$$\begin{aligned}
\hat{A}|\Psi\rangle &= \sum_{ijk}A_{ij}\,|\psi_i\rangle\big\langle\psi_j\big|\,c_k\,|\psi_k\rangle = \sum_{ijk}A_{ij}c_k\,|\psi_i\rangle\,\delta_{jk}\\
&= \sum_{ij}A_{ij}c_j\,|\psi_i\rangle = \sum_{ij}b_j\,|\psi_i\rangle = \big|\Psi'\big\rangle.
\end{aligned} \tag{2.12}$$

A special case is when the new state vector is a multiple of the transformed state vector

$$\hat{A}|a\rangle = a\,|a\rangle\,, \tag{2.13}$$

where $|a\rangle$ is called an eigenstate of $\hat{A}$ with an associated eigenvalue a. We say that two operators $\hat{A}$ and $\hat{B}$ commute if

$$\hat{A}\hat{B}\,|\Psi\rangle = \hat{B}\hat{A}|\Psi\rangle\,, \tag{2.14}$$

so that the order of the operators are invariant, and we write this in terms of the operator commutator

$$[\hat{A},\hat{B}] = 0. \tag{2.15}$$

For commuting operators we have that there exist a common set of eigenstates, also called simultaneous eigenstates, that is

$$\hat{A}\hat{B}\,|ab\rangle = \hat{B}\hat{A}|ab\rangle = a\hat{B}\,|ab\rangle = ab\,|ab\rangle\,. \tag{2.16}$$

This will become useful later on, when we make measurements on a quantum computer. We can then measure multiple commuting operators simultaneously.

To acquire some information about a quantum system, we need to make measurements. Measurable quantities of quantum systems are called observables. The outcome of a measurement needs to be real valued, and so we associate a linear Hermitian operator with each observable. These are operators that are their own conjugate transpose

$$\hat{H} = \hat{H}^{\dagger}. \tag{2.17}$$

The expected value of a measurement of an observable on a state $|h\rangle$ is given by the expectation value

$$\langle h | \hat{H} | h \rangle = h \langle h | h \rangle = h, \tag{2.18}$$

so measuring an observable will result in one of the eigenvalues of the corresponding operator. One such operator of particular interest is the total energy operator also called the Hamiltonian operator

$$\hat{H} = \hat{t} + \hat{v}. \tag{2.19}$$

It is the sum of the kinetic energy operator and potential energy operator giving the respective energies of some system of interest. The non-relativistic time independent Schrödinger equation, being our main concern in this thesis, is defined as the eigenvalue problem

$$\hat{H} | \Psi \rangle = E | \Psi \rangle. \tag{2.20}$$

2.3   Measurements in quantum mechanics

As we mentioned in the last section, measurements are important when we want to acquire information about a quantum system. We will now outline some important concepts concerning measurements of quantum states, by introducing a common representation of such measurements.

2.3.1   Measurement of a quantum state

A quantum measurement is described by the operator $\hat{M}_m$ where the index m refers to the measurement outcome that occurs. The probability for the measurement of a quantum state $|\Phi\rangle$ to result in m is given by the expectation value

$$p(m) = \langle \Phi | \hat{M}_m^{\dagger} \hat{M}_m | \Phi \rangle. \tag{2.21}$$

The probability for all outcomes to occure must yield

$$\sum_m p(m) = 1, \tag{2.22}$$

from which we get the completeness of the measurement operator

$$\sum_m \hat{M}_m^{\dagger} \hat{M}_m = \mathbb{1}. \tag{2.23}$$

Immediately after a measurement with outcome m, the system is in the state

$$\frac{\hat{M}_m | \Phi \rangle}{\sqrt{\langle \Phi | \hat{M}_m^{\dagger} \hat{M}_m | \Phi \rangle}}. \tag{2.24}$$

2.3.2 Projective measurement

Next we consider a special case of the general measurement. A Hermitian operator gives $\hat{M}_m^\dagger = \hat{M}_m$ and so we have that $\hat{M}_m^\dagger \hat{M}_m = \hat{M}_m^2 = \hat{M}_m$. This is called a *projective measurement*. To distinguish between the two types of measurements we denote projective measurements with the projector $\hat{P}_m$ instead of $\hat{M}_m$. The probability is then given as

$$p(m) = \langle \Phi | \hat{P}_m | \Phi \rangle, \tag{2.25}$$

with the system immediately after the measurement in the state

$$\frac{\hat{P}_m | \phi \rangle}{\sqrt{p(m)}}, \tag{2.26}$$

given that the outcome was m. The projector $\hat{P}_m$ is part of a spectral decomposition of an observable M

$$M = \sum_m m \hat{P}_m. \tag{2.27}$$

We can regard the decomposition as several projections into the eigenspace of M along with the eigenvalues m. The expectation value of the measurement of M is then

$$\begin{aligned}
\langle M \rangle &= \langle \phi | M | \phi \rangle, \\
&= \langle \phi | \sum_m m \hat{P}_m | \phi \rangle, \\
&= \sum_m m \langle \phi | \hat{P}_m | \phi \rangle, \\
&= \sum_m m \, p(m).
\end{aligned} \tag{2.28}$$

2.3.3 Phase

Consider the state $e^{i\theta} | \phi \rangle$. We say that it is equal to the state vector $| \phi \rangle$ by a global phase factor $e^{i\theta}$. For measurements this has the effect that

$$\langle \phi | e^{-i\theta} \hat{M}_m^\dagger \hat{M}_m e^{i\theta} | \phi \rangle = \langle \phi | \hat{M}_m^\dagger \hat{M}_m | \phi \rangle$$

From an observational point of view these two states are identical. Consider then the two states

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.29}$$

Each state has an equal magnitude of amplitudes, but the amplitudes for $|1\rangle$ have different sign. We say that two states differ by a relative phase if $\alpha = e^{i\theta} \beta$ where α and β are the two different state amplitudes. Unlike a global phase, thr relative phase is a basis-dependent concept and we have to consider this when making measurements.

2.4 The variational method

We assume an arbitrary Hamiltonian $\hat{H}$ describing the energy of a many-body system. Next we assume it to have a discrete set of eigenvalues so that there exists a set of exact solutions to the Schrödinger equation

$$\hat{H} | \Phi_i \rangle = E_i | \Phi_i \rangle, \qquad i = 0, 1, \dots \tag{2.30}$$

where

$$E_0 \leq E_1 \leq E_2 \leq \ldots \tag{2.31}$$

Then for any normalized wave function $|\Psi\rangle$ the expectation value of $\hat{H}$ is an upper bound to E_0. That is,

$$E_0 \leq \langle\Psi|\hat{H}|\Psi\rangle. \tag{2.32}$$

To see this we first assume that the set of eigenstates $\{|\Phi_i\rangle\}$ of $\hat{H}$ is complete, with $\langle\Phi_i|\hat{H}|\Phi_j\rangle = E_i\delta_{ij}$. We can then write the normalization

$$
\begin{aligned}
\langle\Psi|\Psi\rangle &= \sum_{ij}\langle\Psi|\Phi_i\rangle\langle\Phi_i|\Phi_j\rangle\langle\Phi_j|\Psi\rangle = \sum_{ij}\langle\Psi|\Phi_i\rangle\,\delta_{ij}\,\langle\Phi_j|\Psi\rangle \\
&= \sum_i |\langle\Phi_i|\Psi\rangle|^2 = 1.
\end{aligned}
\tag{2.33}
$$

Similarly equation (2.32) becomes

$$
\begin{aligned}
\langle\Psi|\hat{H}|\Psi\rangle &= \langle\Psi|\Phi_i\rangle\langle\Phi_i|\hat{H}|\Phi_j\rangle\langle\Phi_j|\Psi\rangle = \sum_{ij}\langle\Psi|\Phi_i\rangle\,E_i\delta_{ij}\,\langle\Phi_j|\Psi\rangle \\
&= \sum_i E_i|\langle\Phi_i|\Psi\rangle|^2.
\end{aligned}
\tag{2.34}
$$

Since we have assumed that $E_i \geq E_0$, we can then write the expectation value as

$$\langle\Psi|\hat{H}|\Psi\rangle \geq \sum_i E_0|\langle\Phi_i|\Psi\rangle|^2 = E_0. \tag{2.35}$$

The variational method makes use of this principle by constructing a normalized trial wave function dependent on a set of variables which is optimized so that the expectation value of the Hamiltonian yields the lowest possible energy.

2.5   Many-body theory

This thesis will mainly be concerned with finding solutions to the time-independent Schrödinger equation

$$\hat{H}|\Psi\rangle = E|\Psi\rangle. \tag{2.36}$$

For simple systems as the hydrogen atom, solving this equation is quite straightforward, as done in most introductory courses in quantum mechanics. However, for more complex systems involving multiple particles the exact wave function becomes very complicated. A common practice is then to make an approximation and gradually improve this, e.g. using the variational method. Depending on the application and the system, several methods have been developed to make such approximations. Later on we will introduce the approximate wave function developed from the Hartree-Fock method as well as two popular improvements of this; the configuration interaction method [1] and the coupled cluster method [2].

2.5.1 The many-body Hamiltonian

We write the many-body Hamiltonian as the sum of a one-body operator and a two-body operator, assuming the interaction part can be limited to two-body interactions only

$$\hat{H} = \sum_i \hat{h}_0(\mathbf{x}_i) + \sum_{i<j} \hat{v}(\mathbf{r}_i, \mathbf{r}_j). \tag{2.37}$$

The one-body operator

$$\hat{h}_0(\mathbf{x}_i) = \hat{t}(\mathbf{x}_i) + \hat{u}_{\text{ext}}(\mathbf{x}_i), \tag{2.38}$$

is the sum of the kinetic energy and some external potential, representing the non-interacting part of the Hamiltonian. The two-body operator represents the particle-particle interaction, and for a many-electron system this can be the Coulomb interaction between the electron pairs

$$\hat{v}(\mathbf{r}_i, \mathbf{r}_j) = \frac{1}{\left|\mathbf{r}_i - \mathbf{r}_j\right|}, \tag{2.39}$$

written in natural units.

2.5.2 Systems of identical particles

For a system of identical particles we find all useful information contained in its wave function

$$\Psi \rightarrow \Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \tag{2.40}$$

where we limit the particle to be governed by its spatial coordinates $\mathbf{r}_i$ and spin coordinates $\mathbf{s}_i$, denoted as $\mathbf{x}_i = (\mathbf{r}_i, \mathbf{s}_i)$. As we deal we identical particles, there is no way to distinguish any two particles from each other. Thus, we must consider the effect of exchanging two particles as the probability of two different configurations is equivalent

$$\left|\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)\right|^2 = \left|\Psi(\mathbf{x}_2, \mathbf{x}_1, \ldots, \mathbf{x}_n)\right|^2. \tag{2.41}$$

We prove this by introducing the permutation operator $\hat{P}_{ij}$, with the action on any state that it exchange the particles i and j

$$\hat{P}_{12}\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \Psi(\mathbf{x}_2, \mathbf{x}_1, \ldots, \mathbf{x}_n). \tag{2.42}$$

We then have

$$\begin{aligned}
\left|\Psi(\mathbf{x}_2, \mathbf{x}_1, \ldots, \mathbf{x}_n)\right|^2 &= \langle\Psi(\mathbf{x}_2, \mathbf{x}_1, \ldots, \mathbf{x}_n)|\Psi(\mathbf{x}_2, \mathbf{x}_1, \ldots, \mathbf{x}_n)\rangle \\
&= \langle\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)|\hat{P}_{12}^{\dagger}\hat{P}_{12}|\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)\rangle \\
&= \langle\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)|\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)\rangle \\
&= \left|\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)\right|^2.
\end{aligned} \tag{2.43}$$

where we set $\hat{P}_{ij}^{\dagger}\hat{P}_{ij} = \mathbb{1}$. Next we make a generalization that the effect of exchanging the two particles back will not have any effect on the original wave function. We then clearly have

$$\hat{P}_{12}^2\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = p_{12}\hat{P}_{12}\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = p_{12}^2\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \tag{2.44}$$

where p_{ij} is the eigenvalue of $\hat{P}_{ij}$ with $p_{ij}^2 = 1$. Thus we are left with two possible sets of wave functions, all an eigenfunction of the permutation operator with eigenvalue 1 or -1. We call these symmetric and antisymmetric wave functions respectivly. It turns out that the symmetric wave functions correspond to a system of bosons, and the antisymmetric to fermions.

2.5.3   Slater determinants

A simple starting point for approximating the wave function of a many-body system is to assume that it can be written as the product of many one-particle wave functions. The wave function is then separable

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2)\ldots\phi_n(\mathbf{x}_n), \tag{2.45}$$

and there is no interaction between the electrons. This is known as the Hartree product, where $\phi(\mathbf{x})$ is an orbital. However, this fails to satisfy the antisymmetry principle. When dealing with a system of fermions, an interchange of the coordinates of any two particles would lead to a change in sign. To see this we look at the two-particle case

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) = \phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2), \tag{2.46}$$

$$\Psi(\mathbf{x}_2, \mathbf{x}_1) = \phi_1(\mathbf{x}_2)\phi_2(\mathbf{x}_1). \tag{2.47}$$

To satisfy the antisymmetry principle we must have

$$\phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2) = -\phi_1(\mathbf{x}_2)\phi_2(\mathbf{x}_1).$$

This is clearly not always fulfilled. Instead we ensure this by rewriting the wave function as follows

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{\sqrt{2}}\Big(\phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2) - \phi_1(\mathbf{x}_2)\phi_2(\mathbf{x}_1)\Big), \tag{2.48}$$

where we have

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) = -\Psi(\mathbf{x}_2, \mathbf{x}_1). \tag{2.49}$$

From this we also get that having any two particles in the same orbital is not allowed, that is if $\mathbf{x}_1 = \mathbf{x}_2$ then $\Psi(\mathbf{x}_1, \mathbf{x}_2) = 0$.

As we wish to generalize this to n particles, we notice that equation (2.48) can be written as the determinant

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \frac{1}{\sqrt{n!}}\begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_n(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_n) & \phi_2(\mathbf{x}_n) & \cdots & \phi_n(\mathbf{x}_n) \end{vmatrix}, \tag{2.50}$$

that we call the Slater determinant. Usually we write this in shorthand with only the diagonal elements in a ket:

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = |\phi_1\phi_2\ldots\phi_n\rangle = |12\ldots n\rangle. \tag{2.51}$$

We can either choose a single Slater determinant as our wave function, or a more complex linear combination. With this general form of an approximated wave function we only need to define the orbitals.

2.5.4   Orbitals

An *orbital* is defined as a single-particle wave function. The projected state

$$\psi(\mathbf{r}) = \langle \mathbf{r}|\psi \rangle \tag{2.52}$$

is called a *spatial orbital* and is a function of the position $\mathbf{r}$ of the particle. Generally we assume a set of spatial orbitals $\{\psi_i\}$ to be orthonormal, thus

$$\int d\mathbf{r}\psi_i^*(\mathbf{r})\psi_j(\mathbf{r}) = \delta_{ij}. \tag{2.53}$$

As we want to describe fermions we also need to include the spin. We introduce a new orthonormal set of functions $\{\sigma_+, \sigma_-\}$ specifying spin-up and spin-down respectively. A *spin orbital* is then defined as

$$\phi_i(\mathbf{x}) = \psi_i(\mathbf{r})\sigma, \tag{2.54}$$

with a set $\{\phi_i\}$ of spin orbitals being orthonormal as a consequence of all the spatial orbitals being orthonormal [1]. However, note that the spatial orbitals are not required to be orthonormal, by including the orthonormal spin functions any set of spatial orbitals will lead to a new set of orthonormal orbitals.

2.6   Second quantization

Second quantization is a formalism that often serves as the foundation of most many-body methods. States and operators are described in terms of creation and annihilation operators. As a result the often complicated usage of formal quantum mechanics is reduced to algebra [9].

2.6.1   Occupation number representation

Assume we can approximate the wave function of many-body system as the product of single-particle wave functions, that is

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2)\cdots\phi_n(\mathbf{x}_n), \tag{2.55}$$

where $\mathbf{x}_i$ describes the spin and spatial coordinates of particle i. Next we introduce the so called number operators $\hat{N}_i$ for each spatial orbital i, with eigenvalue equal to the number of particles in the orbital i

$$\hat{N}_i\Psi = n_i\Psi. \tag{2.56}$$

From this we can define the occupation number representation

$$\Psi \rightarrow |n_i n_j n_k \ldots\rangle \tag{2.57}$$

where $n_i, n_j, n_k, \cdots \in \mathbb{R}$ thus represents the number of particles in each orbital. For a system of bosons n_i can take any positive value. For a system of fermions only one particle can occupy a single spin orbital at a time, thus it can be represented as a state of ones and zeros. This will become very useful later on when we want to represent a system of qubits, that only can take one of these values.

2.6.2   Fock space

With every orbital ϕ_i there is an associated Hilbert space $\mathcal{H}_i$ describing the state space of a particle in this orbital. For an n-body system the state space is described by the tensor product

$$\mathcal{H}^n = \mathcal{H}_0 \otimes \cdots \otimes \mathcal{H}_n.$$

We define the Fock space as the direct sum of all possible n-body Hilbert spaces

$$\mathcal{F}_\mu = \bigoplus_{i=0}^{n} \hat{S}_\mu \mathcal{H}^i, \tag{2.58}$$

where $\hat{S}_\mu$ is an operator that symmetrizes or antisymmetrizes a tensor corresponding to the value of $\mu = \pm$ respectively, for a system of bosons or fermions. This then describes all possible configuration of particles, being useful to describe the state space further on in the context of creation and annihilation operators.

2.6.3   Creation and annihilation operators

We look at the Slater determinant with no orbitals

$$| \, \rangle,$$

that we call the vacuum state and assume it to be normalized $\langle \, | \, \rangle = 1$. Next we define a set of operators $\{\hat{a}_p, \hat{a}_p^\dagger\}$. The creation operator $\hat{a}_p^\dagger$ acts on a state by creating an orbital p as long at it is unoccupied

$$\hat{a}_p^\dagger | \, \rangle = \left| \phi_p \right\rangle, \qquad \hat{a}_p^\dagger \left| \phi_p \right\rangle = 0. \tag{2.59}$$

The annihilation operator $\hat{a}_p$ removes orbital p as long at it is occupied

$$\hat{a}_p | \, \rangle = 0, \qquad \hat{a}_p \left| \phi_p \right\rangle = | \, \rangle. \tag{2.60}$$

From this we can write an arbitrary Slater determinant as a product of creation operators on the vacuum state

$$|\phi_1 \phi_2 \ldots \phi_n\rangle = \hat{a}_1^\dagger \hat{a}_2^\dagger \ldots \hat{a}_n^\dagger | \, \rangle = \prod_{i=1}^{n} \hat{a}_i^\dagger | \, \rangle. \tag{2.61}$$

From section (2.5.3) we remember that Slater determinants are antisymmetric. To ensure this we need to develop the so called anticommutation relations for the creation and annihilation operators. First, we define the anticommutation relation between two operators A and B as

$$\{A, B\} = AB + BA. \tag{2.62}$$

Then we have that

$$\left| \phi_p \phi_q \phi_r \phi_s \right\rangle = -\left| \phi_q \phi_p \phi_r \phi_s \right\rangle \quad \Leftrightarrow \quad \hat{a}_p^\dagger \hat{a}_q^\dagger |\phi_r \phi_s\rangle = -\hat{a}_q^\dagger \hat{a}_p^\dagger |\phi_r \phi_s\rangle, \tag{2.63}$$

from which we derive the anticommutator

$$\{\hat{a}_p^\dagger, \hat{a}_q^\dagger\} = 0. \tag{2.64}$$

In the same manner we find the anticommutator for the annihilation operators from

$$|\phi_r\phi_s\rangle = -|\phi_r\phi_s\rangle \quad \Longleftrightarrow \quad \hat{a}_p\hat{a}_q\left|\phi_p\phi_q\phi_r\phi_s\right\rangle = -\hat{a}_q\hat{a}_p\left|\phi_p\phi_q\phi_r\phi_s\right\rangle, \tag{2.65}$$

to be

$$\{\hat{a}_p, \hat{a}_q\} = 0. \tag{2.66}$$

Next, we look at $\{\hat{a}_p, \hat{a}_q^\dagger\}$ and $\{\hat{a}_p^\dagger, \hat{a}_q\}$. Due to the symmetry of the anticommutator we clearly have that the two are equal. For all $q \neq p$ we have that

$$\left|\phi_p\phi_r\phi_s\right\rangle = \hat{a}_p^\dagger\hat{a}_q\left|\phi_q\phi_r\phi_s\right\rangle = -\hat{a}_q\hat{a}_p^\dagger\left|\phi_q\phi_r\phi_s\right\rangle. \tag{2.67}$$

For $q = p$ we get

$$\left|\phi_p\phi_r\phi_s\right\rangle = \hat{a}_p^\dagger\hat{a}_p\left|\phi_p\phi_r\phi_s\right\rangle \neq -\hat{a}_p\hat{a}_p^\dagger\left|\phi_p\phi_r\phi_s\right\rangle = 0. \tag{2.68}$$

That is, the anticommutator is equal to the Kronecker-delta

$$\{\hat{a}_p^\dagger, \hat{a}_q\} = \{\hat{a}_p, \hat{a}_q^\dagger\} = \delta_{pq}. \tag{2.69}$$

2.6.4   Particle-hole formalism

We assume that the wavefunction of a system of n fermions can be approximated by a single Slater determinant. In section (2.6.3) we defined this as a series of creation operators acting on the vacuum state

$$|\Phi_0\rangle = \prod_{i=1}^{n} a_i^\dagger |\,\rangle. \tag{2.70}$$

where $|\Phi_0\rangle$ now contains all orbitals below the Fermi level defined as the level under which all lowest possible orbitals are occupied. We denote these orbitals with labels i, j, k, and refer to $|\Phi_0\rangle$ as our reference vacuum. Acting on the reference vacuum with creation and annihilation operators then gives

$$\hat{a}_p^\dagger |\Phi_0\rangle = 0 \quad \text{and} \quad \hat{a}_p |\Phi_0\rangle = \left|\Phi_p\right\rangle \quad \text{for } p \in i, j, k, \ldots,$$
$$\hat{a}_p^\dagger |\Phi_0\rangle = |\Phi^p\rangle \quad \text{and} \quad \hat{a}_p |\Phi_0\rangle = 0 \qquad \text{for } p \notin i, j, k, \ldots,$$

where the subscript in $\left|\Phi_p\right\rangle$ denotes that orbital p is removed from $|\Phi_0\rangle$, while the superscript in $|\Phi^p\rangle$ denotes that orbital p is occupied in addition to $|\Phi_0\rangle$. Introducing new labels $a, b, c \cdots \notin i, j, k, \ldots$ for the so called virtual orbitals above the Fermi level, surely acting with $\hat{a}_a^\dagger$ on the reference vacuum creates a new particle. However, as we want to keep the number of particles in our system conserved we then also need to remove an orbital below the Fermi Level. This action is called a $1p - 1h$ excitation

$$a_a^\dagger a_i |\Phi_0\rangle = \left|\Phi_i^a\right\rangle, \tag{2.71}$$

and can be regared as creating both a hole in orbital i and a particle in orbital a. Thus both $a_a^\dagger$ and a_i are creation operators, while a_a and $a_i^\dagger$ are annihilation operators. Conviniently we call $i, j, k, \ldots$ hole states and $a, b, c, \ldots$ particle states, and a general $np - nh$ excitation can be written

$$\hat{a}_a^\dagger\hat{a}_b^\dagger \ldots \hat{a}_j\hat{a}_i |\Phi_0\rangle = \left|\Phi_{ij\ldots}^{ab\ldots}\right\rangle. \tag{2.72}$$

2.6.5   Second quantized Hamiltonian

A two-body Hamiltonian in second quantized form is

$$\hat{H} = \sum_{pq} \langle p | \hat{h} | q \rangle \, a_p^\dagger a_q + \frac{1}{4} \sum_{pqrs} \langle pq | \hat{v} | rs \rangle \, a_p^\dagger a_q^\dagger a_s a_r, \qquad (2.73)$$

where the indices run over all orbitals of interest. With this Hamiltonian we are dealing with non-commuting creation and annihilation operators, and so a common practice is to normal order these. Normal ordering means having all annihilation operators to the right of all creation operators. First we refresh our memory of the anticommutation rules of creation and annihilation operators

$$a_n a_m + a_m a_n = a_n^\dagger a_m^\dagger + a_m^\dagger a_n^\dagger = 0, \qquad a_n a_m^\dagger + a_m^\dagger a_n = \delta_{mn}. \qquad (2.74)$$

This means that for particle states we have

$$a_a a_b^\dagger + a_b^\dagger a_a = \delta_{ab},$$

and for hole states we have

$$a_i^\dagger a_j + a_j a_i^\dagger = \delta_{ij}.$$

We can then define the contraction between operators as

$$\overbrace{a_n a_m^\dagger} = a_n a_m^\dagger - \{a_n a_m^\dagger\}_N, \qquad (2.75)$$

where $\{\ \}_N$ means the normal ordered form. For particle and hole states the only non-zero contractions are

$$\overbrace{a_a a_b^\dagger} = a_a a_b^\dagger - \{a_a a_b^\dagger\}_N = a_a a_b^\dagger + a_b^\dagger a_a = \delta_{ab}, \qquad (2.76)$$

$$\overbrace{a_i^\dagger a_j} = a_i^\dagger a_j - \{a_i^\dagger a_j\}_N = a_i^\dagger a_j + a_j a_i^\dagger = \delta_{ij}. \qquad (2.77)$$

Looking at the first term in our Hamiltonian we need to normal order $\sum_{pq} a_p^\dagger a_q$. Using Wick's theorem, see appendix (A.1), we get

$$a_p^\dagger a_q = \{a_p^\dagger a_q\} + \{\overbrace{a_p^\dagger a_q}\}$$

For particle states the contraction is equal to zero while for hole states it is equal to δ_{pq}.

$$\sum_{pq} \langle p | \hat{h} | q \rangle \, a_p^\dagger a_q = \sum_{pq} \langle p | \hat{h} | q \rangle \, \{a_p^\dagger a_q\} + \sum_{i} \langle i | \hat{h} | i \rangle .$$

For the second term we need to normal order $\sum_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$. With Wick's theorem the non-zero contractions are

$$\begin{aligned}
a_p^\dagger a_q^\dagger a_s a_r = \ & \{a_p^\dagger a_q^\dagger a_s a_r\} + \{\overbrace{a_p^\dagger a_q^\dagger a_s a_r}\} + \{\overbrace{a_p^\dagger a_q^\dagger a_s a_r}\} + \{\overbrace{a_p^\dagger a_q^\dagger a_s a_r}\} \\
& + \{\overbrace{a_p^\dagger a_q^\dagger a_s a_r}\} + \{\overbrace{a_p^\dagger a_q^\dagger a_s a_r}\} + \{\overbrace{a_p^\dagger a_q^\dagger a_s a_r}\} \\
= \ & \{a_p^\dagger a_q^\dagger a_s a_r\} - \delta_{ips}\{a_q^\dagger a_r\} + \delta_{ipr}\{a_q^\dagger a_s\} \\
& + \delta_{iqs}\{a_p^\dagger a_r\} - \delta_{iqr}\{a_p^\dagger a_s\} - \delta_{ips}\delta_{jqr} + \delta_{ipr}\delta_{jqs}.
\end{aligned}$$

Again the contractions are equal to zero for particle states, but not for hole states. The interaction term then becomes

$$\sum_{pqrs} \langle pq|\hat{v}|rs\rangle\, a_p^\dagger a_q^\dagger a_s a_r = \sum_{pqrs} \langle pq|\hat{v}|rs\rangle\, \{a_p^\dagger a_q^\dagger a_s a_r\} - \sum_{ipr} \langle iq|\hat{v}|ri\rangle\, \{a_q^\dagger a_r\} + \sum_{ips} \langle iq|\hat{v}|is\rangle\, \{a_q^\dagger a_s\}$$

$$+ \sum_{iqr} \langle pi|\hat{v}|ri\rangle\, \{a_p^\dagger a_r\} - \sum_{iqs} \langle pi|\hat{v}|is\rangle\, \{a_p^\dagger a_s\}$$

$$- \sum_{ij} \langle ij|\hat{v}|ji\rangle + \sum_{ij} \langle ij|\hat{v}|ij\rangle.$$

Due to the anti symmetry $\langle pq|\hat{v}|rs\rangle = -\langle pq|\hat{v}|sr\rangle = -\langle qp|\hat{v}|rs\rangle = \langle qp|\hat{v}|sr\rangle$ this is

$$\sum_{pqrs} \langle pq|\hat{v}|rs\rangle\, a_p^\dagger a_q^\dagger a_s a_r = \sum_{pqrs} \langle pq|\hat{v}|rs\rangle\, \{a_p^\dagger a_q^\dagger a_s a_r\}$$

$$+ 4\sum_{ipq} \langle pi|\hat{v}|qi\rangle\, \{a_p^\dagger a_q\} + 2\sum_{ij} \langle ij|\hat{v}|ij\rangle.$$

We can now write the Hamiltonian as

$$\hat{H} = \sum_{pq} \langle p|\hat{h}|q\rangle\, \{a_p^\dagger a_q\} + \sum_{ipq} \langle pi|\hat{v}|qi\rangle\, \{a_p^\dagger a_q\} + \frac{1}{4}\sum_{pqrs} \langle pq|\hat{v}|rs\rangle\, \{a_p^\dagger a_q^\dagger a_s a_r\}$$

$$+ \sum_{i} \langle i|\hat{h}|i\rangle + \frac{1}{2}\sum_{ij} \langle ij|\hat{v}|ij\rangle, \tag{2.78}$$

or as

$$\hat{H} = \hat{H}_N + E_{ref}, \tag{2.79}$$

with the normal ordered Hamiltonian defined as

$$\hat{H}_N = \sum_{pq} f_{pq}\{a_p^\dagger a_q\} + \frac{1}{4}\sum_{pqrs} \langle pq|\hat{v}|rs\rangle\, \{a_p^\dagger a_q^\dagger a_s a_r\} = \hat{F}_N + \hat{V}_N, \tag{2.80}$$

and the reference energy

$$E_{ref} = \sum_{i} \langle i|\hat{h}|i\rangle + \frac{1}{2}\sum_{ij} \langle ij|\hat{v}|ij\rangle. \tag{2.81}$$

In equation (2.80) we have also introduced a new variable

$$f_{pq} = \langle p|\hat{h}|q\rangle + \sum_{i} \langle pi|\hat{v}|qi\rangle, \tag{2.82}$$

called the Fock matrix.

2.7   The Hartree-Fock approximation

The Hartree-Fock method gives a simple, but important approximation to the ground state of a many-body system and serves as the starting point for more accurate methods. We will not cover all the theory behind it, as we only implicitly make use of it,

see [1] or [9] for a more in-depth outline of the method. Its importance to this thesis is through the resulting single-particle basis $\{\phi_i\}$ so that the single Slater determinant

$$|\Phi\rangle = |\phi_1 \ldots \phi_n\rangle, \tag{2.83}$$

minimizes the energy functional

$$E[\Phi] = \langle\Phi|\hat{H}|\Phi\rangle. \tag{2.84}$$

The variational principle guarantees that $|\Phi\rangle$ gives the best approximation to the ground state of the system, and the corresponding energy gives an upper limit to the exact ground state energy. Due to the complex nature of two-body interactions, we want to replace it by a one-body interaction where each electron experiences an averaged interaction from the other electrons. This is defined in the Hartree-Fock potential $\hat{u}_{\mathrm{HF}}$ included in the so called Fock operator

$$\hat{f} = \hat{h} + \hat{u}_{\mathrm{HF}}. \tag{2.85}$$

We can derive this operator by minimizing equation (2.84) with the constraint that the single-particle basis remains orthonormal using Lagrange multipliers. This result in the canonical Hartree-Fock equations

$$\hat{f}|\phi_i\rangle = \epsilon_i|\phi_i\rangle. \tag{2.86}$$

However, $\hat{u}_{\mathrm{HF}}$ will depend on the other orbitals ϕ_j, making these a set of nonlinear equations that need to be solved iteratively. The optimized state $|\Phi\rangle$ is called the Hartree-Fock state and will be used as a reference state further on. The expectation value from equation (2.84) is equivalent to the reference energy discussed in section (2.6.5).

2.8   The configuration interaction method

Full configuration interaction (FCI) will be used as a benchmark for all calculations done in this thesis. The computational complexity of the method scales badly for a growing number of particles, which makes it unfavourable for realistic calculations. It provides an upper bound to the exact ground state energy and is exact within the subspace spanned by a given basis set. This makes it very useful when we want to examine the accuracy of other methods.

In the last section on Hartree-Fock theory we also found an upper bound to the exact energy using a single Slater determinant as our wave function, but this does not account for the correlation energy for all the electron-electron interactions. We thus seek to find the correlation energy

$$E = E_{\mathrm{HF}} + E_{\mathrm{corr}} \quad \Rightarrow \quad E_{\mathrm{corr}} = E - E_{\mathrm{HF}}, \tag{2.87}$$

where we must have $E_{\mathrm{corr}} \leq 0$ since the Hartree-Fock energy is an upper bound to the exact energy. Given a set of $2k$ spin orbitals the Hartree-Fock wave function can be excited into a set of possible N-electron determinants

$$\left|\Phi_i^a\right\rangle, \left|\Phi_{ij}^{ab}\right\rangle, \left|\Phi_{abc}^{ijk}\right\rangle, \cdots \tag{2.88}$$

By using these as a basis, we can expand the exact wave function

$$|\Psi\rangle = c_0 |\Phi_0\rangle + \sum_{ia} c_i^a |\Phi_i^a\rangle + \sum_{ijab} c_{ij}^{ab} |\Phi_{ij}^{ab}\rangle + \sum_{\substack{ijk \\ abc}} c_{ijk}^{abc} |\Phi_{ijk}^{abc}\rangle + \dots \tag{2.89}$$

and assume intermediate normalization

$$\langle\Phi|\Psi\rangle = 1 \tag{2.90}$$

where it follows that $c_0 = 1$. We can then rewrite the expansion of the FCI wave function in a more compact manner

$$|\Psi\rangle = (1 + \hat{C}) |\Phi_0\rangle, \tag{2.91}$$

using the so called correlation operator

$$\hat{C} = \sum_{ia} c_i^a a_a^\dagger a_i + \sum_{ijab} c_{ij}^{ab} a_a^\dagger a_b^\dagger a_j a_i + \dots \tag{2.92}$$

To solve the Schrödinger equation $\hat{H} |\Psi_p\rangle = E_p |\Psi_p\rangle$, we simply expand the Hamiltonian and the eigenstate in the determinant basis, that we for consistency write as $|\Phi_i\rangle$ with $\langle\Phi_j|\Phi_k\rangle = \delta_{jk}$, and left-project the resulting equation by $\langle\Phi_l|$

$$\langle\Phi_l|\Big(\sum_{ij} |\Phi_i\rangle \langle\Phi_i|\hat{H}|\Phi_j\rangle \langle\Phi_j|\Big)\Big(\sum_k c_{pk} |\Phi_k\rangle\Big) = E_p\Big(\sum_k c_{pk} \langle\Phi_l|\Phi_k\rangle\Big), \tag{2.93}$$

$$\Rightarrow \quad \sum_j \langle\Phi_l|\hat{H}|\Phi_j\rangle c_{pj} = E_p c_{pl}. \tag{2.94}$$

This can be rewritten as matrix products

$$HC = EC, \tag{2.95}$$

where E is a diagonal matrix containing the eigenvalues E_p, C is the matrix containing the expansion coefficients c_{pi} of each eigenstates $|\Phi_p\rangle$, and H contains the matrix elements $\langle\Phi_i|\hat{H}|\Phi_j\rangle$. We can solve this by diagonalizing the Hamiltonian matrix, giving all eigenvalues and eigenvectors. If we are only interested in the ground state energy, a faster approach is to minimize the Rayleigh quotient and rewrite it as a cost function for a recurrent neural network, as done by Feng et al [10].

2.9   The coupled cluster method

Another approach to improve the Hartree-Fock wave function is the coupled cluster method. As we will see it is equivalent to FCI in its full limit, but by introducing an exponential operator and a clever truncation scheme, the method scales significantly better with the number of particles in exchange of some accuracy. Most of this section is based on the excellent review on coupled cluster theory by Crawford and Schaefer [2].

2.9.1   Traditional coupled cluster

We start with the usual goal of approximating the exact solution of the time-independent Schrödinger equation for the ground state

$$\hat{H}\left|\Psi\right\rangle = E\left|\Psi\right\rangle. \tag{2.96}$$

To approximate this coupled cluster theory assumes a reference state $\left|\Phi_0\right\rangle$ along with an exponential ansatz

$$\left|\Psi\right\rangle = e^{T}\left|\Phi_0\right\rangle = \left(1 + T + \frac{T^2}{2} + \frac{T^3}{3!} + \dots\right)\left|\Phi_0\right\rangle. \tag{2.97}$$

An advantage of having an exponential ansatz is that it becomes size-extensive [2]. We call T the cluster operator, and seperate it in terms of the number of excitation operators

$$T = T_1 + T_2 + T_3 + \dots \tag{2.98}$$

where

$$T_1 = \sum_{ia} t_i^a a_a^\dagger a_i, \tag{2.99}$$

$$T_2 = \frac{1}{4}\sum_{ijab} t_{ij}^{ab} a_a^\dagger a_b^\dagger a_j a_i, \tag{2.100}$$

$$\dots$$

adopting the particle-hole formalism. Furthermore we assume intermediate normalization

$$\left\langle \Phi_0 | \Psi \right\rangle = 1, \tag{2.101}$$

with the realization that since $\left\langle \Phi_0 | \Phi_0 \right\rangle = 1$ all excited states are orthogonal to the reference state.

Combining equation (2.97) and (2.99) we can reorder the terms by recognizing that products of some operators are equivalent in terms of excitations

$$\begin{aligned}
e^T &= 1 + T + \frac{T^2}{2!} + \frac{T^3}{3!} + \dots \\
&= 1 + T_1 + (T_2 + \frac{1}{2}T_1^2) + (T_3 + T_1 T_2 + \frac{1}{6}T_1^3) + \dots \\
&= 1 + C_1 + C_2 + C_3 + \dots
\end{aligned} \tag{2.102}$$

Where we have grouped the operators exciting the same amount of orbitals. We see that this is equivalent to the CI operator, however inserting this into equation (2.97) we get the wave function

$$\begin{aligned}
\left|\Psi\right\rangle = \left|\Phi_0\right\rangle &+ \sum_{ia} t_i^a \left|\Phi_i^a\right\rangle + \sum_{ijab}\left(t_{ij}^{ab} + \frac{t_i^a t_j^b}{2}\right)\left|\Phi_{ij}^{ab}\right\rangle \\
&+ \sum_{ijkabc}\left(t_{ijk}^{abc} + t_{ij}^{ab}t_k^c + \frac{t_i^a t_j^b t_k^c}{6}\right)\left|\Phi_{ijk}^{abc}\right\rangle + \dots
\end{aligned} \tag{2.103}$$

The full cluster wave function just becomes a reparametrization of the full CI wave function. If we were to truncate the cluster operator we would effectively operate with

more degrees of freedom than truncated CI because products of lower-order excitation operators can make up higher-order excitations. The cluster operator is therefore truncated at some excitation level where we assume that the higher-order excitation amplitudes are small in comparison to the the non-truncated ones. In order to find these parameters we again look at the Schrödinger equation with the exponential ansatz

$$He^T |\Phi_0\rangle = Ee^T |\Phi_0\rangle, \tag{2.104}$$

and left-multiply this by the coupled cluster wave function

$$\langle\Phi_0| e^{-T}He^T |\Phi_0\rangle = E. \tag{2.105}$$

We call $e^{-T}He^T$ the similarity transformed Hamiltonian, that can be expanded using the Campbell-Baker-Hausdorff formula

$$e^{-T}He^T = H + [H,T] + \frac{1}{2!}[[H,T],T] + \ldots \tag{2.106}$$

Luckily this often truncate naturally after a given number of terms, and so we can compute $e^{-T}He^T$ exactly. We then find the cluster amplitudes by solving

$$\left\langle\Phi_i^a\middle| e^{-T}He^T |\Phi_0\rangle = 0, \tag{2.107}$$

$$\left\langle\Phi_{ij}^{ab}\middle| e^{-T}He^T |\Phi_0\rangle = 0, \tag{2.108}$$

$$\left\langle\Phi_{ijk}^{abc}\middle| e^{-T}He^T |\Phi_0\rangle = 0, \tag{2.109}$$

$$\ldots$$

2.9.2 Coupled cluster singles and doubles

A common truncation is the coupled cluster singles and doubles (CCSD)

$$T = T_1 + T_2. \tag{2.110}$$

We then need to solve

$$\left\langle\Phi_i^a\middle| e^{-T}He^T |\Phi_0\rangle = 0, \tag{2.111}$$

$$\left\langle\Phi_{ij}^{ab}\middle| e^{-T}He^T |\Phi_0\rangle = 0. \tag{2.112}$$

In order to solve these equations we define the two variables

$$D_i^a = f_{ii} - f_{aa}, \tag{2.113}$$

$$D_{ij}^{ab} = f_{ii} + f_{jj} - f_{aa} - f_{bb}, \tag{2.114}$$

and subtract $D_i^a t_i^a$ and $D_{ij}^{ab} t_{ij}^{ab}$ from both sides of equation (2.111) and equation (2.112) respectively

$$D_i^a t_i^a = D_i^a t_i^a + \left\langle\Phi_i^a\middle| \bar{H} |\Phi_0\rangle, \tag{2.115}$$

$$D_{ij}^{ab} t_{ij}^{ab} = D_{ij}^{ab} t_{ij}^{ab} + \left\langle\Phi_{ij}^{ab}\middle| \bar{H} |\Phi_0\rangle. \tag{2.116}$$

Finally we divide both sides by D_i^a and D_{ij}^{ab} and end up with an iterative equation for the cluster amplitudes

$$t_i^a = t_i^a + \frac{\left\langle \Phi_i^a \middle| \bar{H} \middle| \Phi_0 \right\rangle}{D_i^a}, \tag{2.117}$$

$$t_{ij}^{ab} = t_{ij}^{ab} + \frac{\left\langle \Phi_{ij}^{ab} \middle| \bar{H} \middle| \Phi_0 \right\rangle}{D_{ij}^{ab}}. \tag{2.118}$$

For canonical Hartree-Fock orbitals, the Fock matrix will be diagonal and the elements in $D_i^a t_i^a$ and $D_{ij}^{ab} t_{ij}^{ab}$ is just the orbital energies. It turns out that we then can make use of the first-order pertubed wave function parameters derived from Møller-Plesset theory [2] to define our initial guess for the cluster amplitudes

$$t_i^a = 0, \tag{2.119}$$

$$t_{ij}^{ab} = \frac{\langle ab| \hat{v} |ij\rangle - \langle ab| \hat{v} |ji\rangle}{D_{ij}^{ab}}. \tag{2.120}$$

2.9.3   A variational coupled cluster method

Since the similarity transformed Hamiltonian is not Hermitian, the variational method is not valid. We can solve this problem by introducing the unitary ansatz

$$|\Psi\rangle = e^{T - T^\dagger} |\Phi_0\rangle. \tag{2.121}$$

Restricting it to the singles and doubles excitations we get

$$\exp\left(T - T^\dagger\right) = \exp\left(\sum_{ia} t_i^a \left(a_a^\dagger a_i - a_i^\dagger a_a\right) + \sum_{ia} t_{ij}^{ab} \left(a_a^\dagger a_b^\dagger a_j a_i - a_i^\dagger a_j^\dagger a_b a_a\right)\right). \tag{2.122}$$

Unfortunately the expansion of the similarity transformed Hamiltonian does not truncate naturally, and so the method scales exponentially. This makes it very unfavorable on a classical computer.

Chapter 3

Quantum Computing

In classical computing all information is processed in terms of bits being turned on or off, expressed as a 1 or a 0. Along with a universal set of operations this provides a framework for solving all types of problems. This simple concept of processing information is valuable when we move on to quantum computation and quantum information theory. In this chapter we will introduce the quantum bit along with a model of quantum computation, the quantum circuit model, from which a set of operations known as quantum gates is said to be universal. Next we will show how problems from quantum mechanics can be solved with a quantum computer, that is how the many-body Hamiltonian can be transformed into a quantum computational problem. For more theory on quantum computing the excellent text book by Nielsen and Chuang [11] is highly recommend.

3.1 Qubits

We define a qubit as the superposition

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \tag{3.1}$$

where α and β are complex numbers given by the normalization

$$\langle \psi | \psi \rangle = |\alpha|^2 + |\beta|^2 = 1. \tag{3.2}$$

The numbers $|\alpha|^2$ and $|\beta|^2$ are interpreted as probabilities for the qubit to be measured in the state $|0\rangle$ or $|1\rangle$ respectively, given as

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{3.3}$$

These are called the computational basis states of a qubit. We recognize that this is the two-level system, being the simplest description of a quantum system.

From the normalization we can rewrite equation (3.1) as

$$|\psi\rangle = e^{i\gamma} \left(\cos\frac{\theta}{2} |0\rangle + e^{i\phi} \sin\frac{\theta}{2} |1\rangle \right), \tag{3.4}$$

where γ, θ and ϕ are real numbers. Here $e^{i\gamma}$ is a global phase factor that we can neglect as it will not have any effect on a measurement, see section (2.3.3).

From equation (3.4) we can think of a qubit state as a point on a sphere with unit radius, called the Bloch sphere, where θ is the angle with respect to the z-axis and ϕ the angle with respect to the x-axis. This view will be helpful later on when we introduce operations on qubits, being linear transformations on the Bloch sphere.

Measuring a qubit

To demonstrate the effect of a measurement operator, we can look at a measurement in the computational basis $\{|0\rangle, |1\rangle\}$. For a single qubit $|\phi\rangle$ the possible outcomes are 0 and 1. We then have the two measurement operators

$$\hat{M}_0 = |0\rangle \langle 0|, \qquad \hat{M}_1 = |1\rangle \langle 1|. \tag{3.5}$$

We see that these are projective measurements as $\hat{M}^\dagger \hat{M} = \hat{M}$, introduced in section (2.3.2), where the corresponding probabilities and outcome states are

$$p(0) = \langle \psi | \hat{M}_0 | \psi \rangle = |\alpha|^2, \qquad p(1) = \langle \psi | \hat{M}_1 | \psi \rangle = |\beta|^2,$$
$$\rightarrow \frac{\hat{M}_0 |\psi\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|} |0\rangle, \qquad \rightarrow \frac{\hat{M}_1 |\psi\rangle}{|\beta|} = \frac{\beta}{|\beta|} |1\rangle,$$

Since $\frac{\alpha}{|\alpha|}$ and $\frac{\beta}{|\beta|}$ can be substituted by a phase factor, the actual states immediately after the measurement are $|0\rangle$ or $|1\rangle$. This is important as we can not actually measure the amplitudes in equation (3.1), we need to measure the state multiple times to get a probability distribution.

3.2 Single-qubit gates
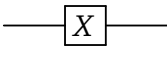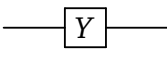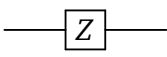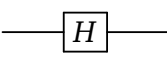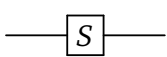
Now that we have defined the basics of qubits, we need to introduce a way to act on them with the quantum analog of logical gates. Naturally, unitary operations are the most prominent candidates, as they preserve norms. In theory any complex 2×2 unitary matrix $\hat{U}$ can be used as a *single-qubit gate*, an operation on a single qubit. We write this in terms of the quantum circuit

$$|\psi\rangle \quad \longrightarrow \boxed{U} \longrightarrow \quad \left| \hat{U}\psi \right\rangle.$$

However, not all unitary operators are suited for a physical implementation. We need to develop a set of gates that can be realized on an actual quantum computer. Of even more importance is developing a universal set of gates, that can represent an arbitrary qubit gate. We will come back to this later on when we introduce multi qubit gates.

First we need to motivate the use of single qubit gates. In table (3.1) we list some useful operations.

Table 3.1: Useful single-qubit gates

Gate	Matrix	Circuit	Action
Pauli-X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	X	Flips the qubit state.
Pauli-Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	Y	π-rotation around the y-axis.
Pauli-Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	Z	Sign change if $|1\rangle$.
Hadamard	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	H	Impose super-position of state
Phase	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	S	$\pi/2$-rotation around the z-axis.

The first three listed are the Pauli operators, or Pauli gates, that are particularly useful when we want to simulate a quantum system. As we will see in section (3.5), there are a direct mapping between these and the ladder operators introduced in section (2.6.3). The X-gate is the equivalent of the logical not-gate with the action on a qubit

$$\hat{X}|\psi\rangle = \hat{X}\big(\alpha|0\rangle + \beta|1\rangle\big) = \alpha|1\rangle + \beta|0\rangle, \tag{3.6}$$

flipping the probability amplitudes corresponding to each basis state. In terms of the Bloch-sphere we regard this as a π-rotation around the x-axis. Likewise the other Pauli-gates make a π-rotation around its respective axis. Consequently the Z-gate induces a relative phase factor

$$\hat{Z}|\psi\rangle = \hat{Z}\big(\alpha|0\rangle + \beta|1\rangle\big) = \alpha|0\rangle - \beta|1\rangle, \tag{3.7}$$

while the Y-gate can be regarded as combination of the two

$$\hat{Y}|\psi\rangle = \hat{Y}\big(\alpha|0\rangle + \beta|1\rangle\big) = i\alpha|1\rangle - i\beta|0\rangle. \tag{3.8}$$

Next we look at the Hadamard gate. Its action on a general qubit state is

$$\hat{H}|\psi\rangle = \hat{H}\big(\alpha|0\rangle + \beta|1\rangle\big) = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle. \tag{3.9}$$

However, assume we have a qubit in the $|0\rangle$ or $|1\rangle$ state we get

$$\hat{H}|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad \hat{H}|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{3.10}$$

being an equal superposition of the two basis states. That is, we will measure the qubit as either $|0\rangle$ or $|1\rangle$ with equal probability.

Lastly we introduce a more general, equally useful, operator: the rotation operator

$$\hat{R}_n(\theta) = \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)\big(n_x\hat{X} + n_y\hat{Y} + n_z\hat{Z}\big), \tag{3.11}$$

around an axis $\vec{n} = (n_x, n_y, n_z)$. Later on we will make use of the rotation operators around the the cartesian axes, so we will list them here along with their relation to the Pauli operators

$$\hat{R}_x(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix} \quad \Rightarrow \quad \hat{X} = i\hat{R}_x(\pi),$$

$$\hat{R}_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix} \quad \Rightarrow \quad \hat{Y} = i\hat{R}_y(\pi), \tag{3.12}$$

$$\hat{R}_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \quad \Rightarrow \quad \hat{Z} = i\hat{R}_z(\pi).$$

In [11] it is shown that for any real number α, β, γ and δ we can represent any single qubit unitary gate using the rotation operators as

$$\hat{U} = e^{i\alpha}\hat{R}_z(\beta)\hat{R}_y(\gamma)\hat{R}_z(\delta), \tag{3.13}$$

or

$$\hat{U} = e^{i\alpha}\hat{R}_x(\beta)\hat{R}_y(\gamma)\hat{R}_x(\delta). \tag{3.14}$$

3.3   System of qubits

We say that a system of qubits $|\Psi\rangle$ is defined as the tensor product of the n qubit states that compose the system

$$|\Psi\rangle = |\psi\rangle_1 \otimes |\psi\rangle_2 \otimes \cdots \otimes |\psi\rangle_n = \bigotimes_{i=1}^n |\psi\rangle_i \tag{3.15}$$

The corresponding computational basis states are the tensor product of the individual basis states. As an example we can look at a two-qubit system

$$\begin{aligned} |\psi\rangle_1 \otimes |\psi\rangle_2 &= \left(\alpha_1 |0\rangle + \beta_1 |1\rangle\right) \otimes \left(\alpha_2 |0\rangle + \beta_2 |1\rangle\right) \\ &= \alpha_1\alpha_2 |00\rangle + \alpha_1\beta_2 |01\rangle + \beta_1\alpha_2 |10\rangle + \beta_1\beta_2 |11\rangle, \end{aligned} \tag{3.16}$$

with the basis states

$$|00\rangle \equiv \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad |01\rangle \equiv \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad |10\rangle \equiv \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad |11\rangle \equiv \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3.17}$$

Clearly all two-qubit states can be written as a linear combination of these. We see that going from a single qubit state to a two-qubit state doubles the number of basis states. It turns out that a system of n qubits will have 2^n basis states, and in turn an equal amount of amplitudes, as in equation (3.16). Thus a system of qubits can represent exponentially more data than classical bits.

As the qubits are all described by the state $|\Psi\rangle$ in equation (3.15), a single qubit operation on the i-th qubit will be a tensor product

$$U \equiv I_1 \otimes \cdots \otimes I_{i-1} \otimes U_i \otimes I_{i+1} \otimes \cdots \otimes I_n, \tag{3.18}$$

acting on all other qubits with an identity operator.

3.4 Multi-qubit gates

In addition to the operations mentioned above, we have a second class of qubit operations we call multi-qubit gates. Below we list three important gates acting on two qubits.

Table 3.2: Useful two-qubit gates

Gate	Matrix	Circuit	Action
CNOT	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$		Pauli-X gate conditional to first qubit.
CU	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix}$		$\hat{U}$-operation conditional to first qubit.
Swap	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$		Swaps the two qubits.

We start by defining an important class of multi-qubit gates. Say we have two qubits that we label $|c\rangle$ and $|t\rangle$. Assume we want to do an operation

$$\hat{U} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \tag{3.19}$$

on $|t\rangle$ conditional to $|c\rangle$, e.g. that $|c\rangle = |1\rangle$, then we see from the basis states in equation (3.17) that this yields the following matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix} \rightarrow \tag{3.20}$$

along with its representation in a quantum circuit. We call this a *controlled operation* and denote it with an uppercase C in front of the operation, in the above case we have a *CU* operation. We call $|c\rangle$ the control qubit and $|t\rangle$ the target qubit. If we wanted a controlled operation conditional to $|c\rangle = |0\rangle$ we have the circuit identity

so for convinience we will refer to controlled operations as conditional to the control qubit being in state $|1\rangle$.

Perhaps the most important controlled operations is the CNOT, or CX, gate.
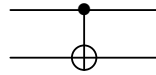
Its action on two qubits in the states $|\psi\rangle$ and $|\phi\rangle$ can be written in terms of modulo 2 addition

$$\text{CNOT}\,|\psi\rangle\,|\phi\rangle = |\psi\rangle\,|\phi \oplus \psi\rangle. \tag{3.21}$$

In [11] it is shown that all unitary matrices can be decomposed using two-level unitary matrices, thus two-level unitary gates are universal. In addition we can decompose all two-level unitary gates using CNOT gates and single-qubit gates. In other words all qubit gates, regardless of how many qubits they act on, can be decomposed into single-qubit gates and CNOTs.

For the SWAP gate, considering the states $|\psi\rangle$ and $|\phi\rangle$, we want the following action

$$\text{SWAP}\,|\psi\rangle\,|\phi\rangle = |\phi\rangle\,|\psi\rangle, \tag{3.22}$$

where ψ and ϕ only can takes values of 0 or 1. We see from equation (3.21) that we can construct the SWAP operation from three CNOTs

$$\text{CNOT}_{1,2}\,|\psi\rangle_1\,|\phi\rangle_2 = |\psi\rangle_1\,|\phi \oplus \psi\rangle_2, \tag{3.23}$$

$$\text{CNOT}_{2,1}\,|\psi\rangle_1\,|\phi \oplus \psi\rangle_2 = |\psi \oplus \phi \oplus \psi\rangle_1\,|\phi \oplus \psi\rangle_2 = |\phi\rangle_1\,|\phi \oplus \psi\rangle_2, \tag{3.24}$$

$$\text{CNOT}_{1,2}\,|\phi\rangle_1\,|\phi \oplus \psi\rangle_2 = |\phi\rangle_1\,|\phi \oplus \psi \oplus \phi\rangle_2 = |\phi\rangle_1\,|\psi\rangle_2. \tag{3.25}$$

This gives the circuit identity
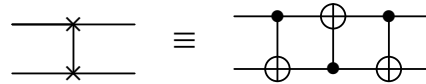
and so we have a very efficient way of swapping qubit states. Lastly we look at an important attribute with the CNOT gate. Acting with Hadamard gates before and after the CNOT on both qubits we have the circuit identity
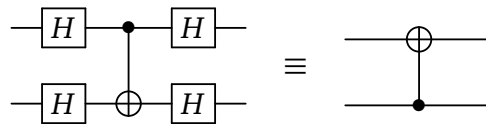
and so with two qubits in an arbitrary state we can not simply regard the CNOT as affecting one of the qubits, but in some way both. This can be shown if we consider the two qubits to be in either of the two eigenstates of the $\hat{X}$ operator

$$\begin{aligned}
\text{CNOT}\,|++\rangle &= |++\rangle, \\
\text{CNOT}\,|+-\rangle &= -|--\rangle, \\
\text{CNOT}\,|-+\rangle &= |-+\rangle, \\
\text{CNOT}\,|--\rangle &= -|+-\rangle,
\end{aligned} \tag{3.26}$$

where the control qubit is being "flipped", not the target qubit.

3.5   Fermionic mapping

In order to implement a general second quantized Hamiltonian given by equation (2.73) on a quantum computer, we need a mapping from creation and annihilation operators to quantum gates. For a fermionic system we have defined the commutation relations between the creation and annihilation operators in section (2.6.3) that the mapping needs to fulfill. From our wave function representation we have that

$$|\Phi\rangle = |\phi_1 \ldots \phi_n\rangle, \tag{3.27}$$

where $\phi_i \in \{0, 1\}$. For this thesis we choose 1 to represent an occupied orbital, and 0 for an unoccupied orbital. Choosing the opposite labeling is equivalent and only a matter of preference. We makes use of the convention

$$|0\rangle_i = \hat{a}_i^\dagger | \rangle, \tag{3.28}$$

$$|1\rangle_i = | \rangle. \tag{3.29}$$

In addition, assuming orbital $i > 1$ is occupied we must satisfy the antisymmetry

$$\hat{a}_i |\phi_1 \ldots \phi_i \ldots \phi_n\rangle = (-1)^{P_i} |\phi_1 \ldots \phi_n\rangle, \tag{3.30}$$

where P_i is the number of permutations needed.

3.5.1   The Jordan-Wigner transformation

A simple and intuitive mapping to quantum gates is the Jordan-Wigner (JW) transformation, where we end up with a sum of tensor products of Pauli operators. We start by defining the raising and lowering operators

$$\sigma^+ = \frac{1}{2}\left(\hat{X} - i\hat{Y}\right) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \tag{3.31}$$

$$\sigma^- = \frac{1}{2}\left(\hat{X} + i\hat{Y}\right) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \tag{3.32}$$

$$\tag{3.33}$$

and their action on the computational basis states

$$\begin{aligned} \sigma^+ |0\rangle = |1\rangle, &\quad \sigma^+ |1\rangle = 0, \\ \sigma^- |1\rangle = |0\rangle, &\quad \sigma^- |0\rangle = 0. \end{aligned} \tag{3.34}$$

To ensure the antisymmetry of fermions we have that the number of pertubations needed in equation (3.30) is equal to the number of occupied orbitals lower than i, that is

$$P_i = \sum_{j}^{i-1} \phi_j \tag{3.35}$$

We know that the eigenvalues of $\hat{Z}$ are ± 1, so we replace $(-1)^{P_i}$ with a product of $\hat{Z}$ operators giving the mapping

$$\hat{a}_i^\dagger = \left(\prod_{j}^{i-1} \hat{Z}_j\right) \sigma_i^+,$$

$$\hat{a}_i = \left(\prod_{j}^{i-1} \hat{Z}_j\right) \sigma_i^-, \tag{3.36}$$

where the sub script on the operators label the qubit they are applied to. Note that in this notation we have left out the identity operators acting on the remaining qubits, as well as replacing the tensor products with regular products.

3.6 Pauli strings

After the Jordan-Wigner transformation of the second quantized Hamiltonian we end up with a sum of tensor products of Pauli gates. It is more convenient to refer to these tensor products as single operators acting on a set of qubits, that we call Pauli strings. To define these we make use of the following notation for the Pauli gates and identity operator

$$\sigma_0 = \hat{\mathbb{1}}, \quad \sigma_1 = \hat{X}, \quad \sigma_2 = \hat{Y}, \quad \sigma_3 = \hat{Z}. \tag{3.37}$$

A Pauli string acting on N qubits is then defined as

$$\mathscr{P}^{q_1 q_2 \dots q_N}_{n_1 n_2 \dots n_N} = \bigotimes_{i=1}^{N} \sigma^{q_i}_{n_i}, \tag{3.38}$$

where q_i denotes a qubit and $n_i \in \{0, 1, 2, 3\}$ denotes a gate type. The only necessary information is the qubits that are not acted on by the identity gates, so we will not include these in any Pauli strings further on. As an example the Pauli string acting on 6 qubits with $\hat{X}$ on the first qubit, $\hat{Y}$ on the third qubit and $\hat{Z}$ on the fifth qubit, is written

$$\mathscr{P}^{135}_{xyz} = \hat{X}_1 \otimes \hat{\mathbb{1}}_2 \otimes \hat{Y}_3 \otimes \hat{\mathbb{1}}_4 \otimes \hat{Z}_5 \otimes \hat{\mathbb{1}}_6, \tag{3.39}$$

where we redefine $n_i \in \{x, y, z\}$ in equation (3.38).

3.6.1 Qubit-wise commutation of Pauli strings

Consider the two operators $\hat{A}$ and $\hat{B}$

$$\hat{A} = \bigotimes_{i}^{n} \hat{A}_i, \quad \hat{B} = \bigotimes_{i}^{n} \hat{B}_i, \tag{3.40}$$

as tensor product of n operators acting on individual qubits. We say that the $\hat{A}$ and $\hat{B}$ are qubit-wise commuting (QWC) operators if all corresponding operators acting on qubit i commute, that is

$$[\hat{A}, \hat{B}]_{QWC} = 0 \quad \Rightarrow \quad [\hat{A}_i, \hat{B}_i] = 0. \tag{3.41}$$

Since the different Pauli-operators do not commute, except with themselves, we have that two Pauli strings $\mathscr{P}^{a_1 a_2 a_3 a_4}_{n_1 n_2 n_3 n_4} \mathscr{P}^{b_1 b_2 b_3 b_4}_{m_1 m_2 m_3 m_4}$ are QWC operators if all $a_i \neq b_i$, or if for all $a_i = b_i$ we have that $n_i = m_i$. Recognizing QWC operators is very useful as they can be measured simultaneously in an efficient manner, since the shared eigenvectors are easy to find and require no extra gate actions. This will be shown later in the implementation part of this thesis.

3.6.2   General commutation of Pauli strings

The definition of QWC operators is just a special case of a more general definition of commutation relations between Pauli strings. If we again consider the two operators $\hat{A}$ and $\hat{B}$ we can have that $[\hat{A}, \hat{B}] = 0$ while $[\hat{A}, \hat{B}]_{QWC} \neq 0$. An example is the three Pauli strings on two qubits $\{\mathscr{P}_{xx}, \mathscr{P}_{yy}, \mathscr{P}_{zz}\}$. We have that

$$[\mathscr{P}_{xx}, \mathscr{P}_{yy}] = 0, \quad [\mathscr{P}_{xx}, \mathscr{P}_{zz}] = 0, \quad [\mathscr{P}_{yy}, \mathscr{P}_{zz}] = 0, \tag{3.42}$$

while the individual Pauli operators do not commute

$$[\hat{X}, \hat{Y}] \neq 0, \quad [\hat{X}, \hat{Z}] \neq 0, \quad [\hat{Y}, \hat{Z}] \neq 0. \tag{3.43}$$

From the anti-commutation relations of Pauli operators we have that the product of any two different Pauli operators σ_a and σ_b is equal to

$$\sigma_a \sigma_b = -\sigma_b \sigma_a, \tag{3.44}$$

this is easy to show and we will not provide any proof. This yields

$$\hat{A}\hat{B} = \bigotimes_i^n \begin{cases} \hat{B}_i \hat{A}_i & \text{if } [\hat{A}_i, \hat{B}_i] = 0, \\ -\hat{B}_i \hat{A}_i & \text{if } [\hat{A}_i, \hat{B}_i] \neq 0, \end{cases} = (-1)^k \hat{B}\hat{A}, \tag{3.45}$$

where k is the number of non-commuting terms. The operators thus commute if k is even.

3.7   Trotter approximation

Assume a unitary operator $U = -k(A+B)$ for some constant k. If we want to map this to a quantum circuit, we will in reality be left with two quantum circuits

$$\langle \Phi | U | \Phi \rangle = -k \Big(\langle \Phi | A | \Phi \rangle + \langle \Phi | B | \Phi \rangle \Big),$$

one for each term of U, requiring us to make $2n$ measurements. However, if the operator was in an exponent e^U (if its a propagator) there would be no straightforward way to implement it unless A and B commute. If they did we could write it in terms of the classical exponential law

$$e^{-k(A+B)} = e^{-kA} e^{-kB}. \tag{3.46}$$

This would then give a simple way to implement U as

$$e^{-kA} e^{-kB} | \Phi \rangle = e^{-kb} e^{-kA} | \Phi \rangle = e^{-kb} e^{-ka} | \Phi \rangle, \tag{3.47}$$

where $| \Phi \rangle$ is an eigenstate of both A and B with eigenvalues a and b. For the case where $[A, B] \neq 0$, the Baker-Campbell-Hausdorff formula yields

$$e^{-kA} e^{-kB} = e^{-k(A+B) + \frac{(-k)^2}{2}[A,B] + \frac{(-k)^3}{12}\left(\left[A,[A,B] \right] - \left[B,[A,B] \right] \right) + \cdots}. \tag{3.48}$$

As we would like exponential operators on the form of equation (3.46) we introduce a common truncation scheme called Trotter approximation. The Trotter product formula states that for arbitrary complex operators A and B we can write

$$e^{-k(A+B)} = \lim_{\rho \to \infty} \left(e^{-\frac{k}{\rho}A} e^{-\frac{k}{\rho}B} \right)^{\rho}. \tag{3.49}$$

From this we get the Trotter approximation

$$e^{-k(A+B)} = \left(e^{\frac{-k}{\rho}A} e^{\frac{-k}{\rho}B} \right)^{\rho} + \mathcal{O}\left(\frac{k^2}{\rho^2} \right). \tag{3.50}$$

Chapter 4

Quantum algorithms

Now that some basics of quantum computing has been presented we can look at how to actually solve problems on a quantum computer. In the following chapter we will in detail outline the mathematical motivations for the algorithms implemented in this thesis. Starting with quantum phase estimation and quantum Fourier transform that make up the Hamiltonian simulation algorithm, before we tackle the variational quantum eigensolver.

4.1 Fourier transform

A useful way to solve problems in many fields of science, especially in physics and mathematics, is to transform it into some other (often simpler) problem for which a solution is known. The discrete fourier transform, which involves such a transformation, is currently one of a few known algorithms that can be computed much faster on a quantum computer than on a classical one. As the classical version of this algorithm should be well known to any undergraduate with a background in physics we will only state its action before elaborating more on the quantum analog. The parts of this chapter concerned with the quantum Fourier transform and quantum phase estimation is based on the book by Nielsen and Chuang [11].

4.1.1 Discrete Fourier transform

The classical discrete Fourier transform inputs a vector x of N complex numbers and outputs a new transformed vector y given by

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{n-1} x_j e^{2\pi ijk/N}, \tag{4.1}$$

where i is the imaginary unit.

4.1.2 Quantum Fourier transform

For the quantum Fourier transform we instead consider N orthonormal states $|0\rangle, \ldots, |N-1\rangle$ that generally form a basis set of a state of interest. We define the transformation as a

linear operator acting on the states with the action

$$|j\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle. \tag{4.2}$$

If we consider a general quantum state we see that

$$
\begin{aligned}
\sum_{j=0}^{N-1} x_j |j\rangle \longrightarrow \sum_{j=0}^{N-1} x_j \left[\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \right] \\
= \sum_{k=0}^{N-1} \left[\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N} \right] |k\rangle \\
= \sum_{k=0}^{N-1} y_k |k\rangle,
\end{aligned}
\tag{4.3}
$$

where we have written the amplitudes as in equation (4.1), giving a clear connection to the classical Fourier transform.

Binary representation of numbers

Any positive integer i can be written in terms of the *binary representation*

$$i = i_1 2^{n-1} + i_2 2^{n-2} + \cdots + i_n 2^0, \tag{4.4}$$

for binary values of $i_i, \ldots, i_n$ and given any integer $n > \frac{\log(j)}{\log(2)}$. It is then convenient to write this as $i = i_1 i_2 \ldots i_n$.

Furthermore any number $0 \leq j \leq 1$ can be written in terms of the *binary fraction*

$$j = \frac{j_1}{2^1} + \frac{j_2}{2^2} + \cdots + \frac{j_m}{2^m} + \mathcal{O}(m) \tag{4.5}$$

for some appropriate value of m. An error term is included as large values of m often will be necessary to get exact values. For our purpose, dealing with a limited number of qubits, we would like m to be sufficiently small. In the same manner as the binary representation we can write the binary fraction as $j = 0.j_1 j_2 \ldots j_m$.

Thus, any positive number k can be represented by a set of binary values as

$$k = k_1 \ldots k_n.k_{n+1} \ldots k_m, \tag{4.6}$$

where $n < m$. We note the following relations

$$2^{m-n}k = k_1 \ldots k_m \tag{4.7}$$

$$\frac{k}{2^n} = 0.k_1 \ldots k_m \tag{4.8}$$

Product representation of the quantum Fourier transform

We start by deriving the product representation of the quantum Fourier transform starting from equation (4.2). We assume n qubits, and so $N = 2^n$. In this case, where each qubit have two basis states, we can rewrite the sum

$$\sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n} |k\rangle. \tag{4.9}$$

First we use the binary representation of k

$$k = k_1 \ldots k_n = k_1 2^{n-1} + \ldots k_n 2^0 = \sum_{l=1}^{n} k_l 2^{n-l}, \tag{4.10}$$

so we can write the exponent as

$$\exp\left(\frac{k}{2^n}\right) = \exp\left(\sum_{l=1}^{n} \frac{k_l 2^{n-l}}{2^n}\right) = \exp\left(\sum_{l=1}^{n} k_l 2^{-l}\right), \tag{4.11}$$

that is the binary fraction of k. Combining this with equation (4.2) gives

$$
\begin{aligned}
|j\rangle &\longrightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle \\
&= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l / 2^l} |k_l\rangle \right] \\
&= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i j / 2^l} |1\rangle \right].
\end{aligned}
\tag{4.12}
$$

Next we use the binary representation of j as with k above. The exponent amplitude of $|1\rangle$ then becomes

$$\exp\left(\frac{2\pi i j}{2^l}\right) = \exp\left(\frac{2\pi i \sum_{m=1}^{n} j_m 2^{n-m}}{2^l}\right) = \exp\left(\sum_{m=1}^{n} 2\pi i j_m 2^{n-m-l}\right), \tag{4.13}$$

From our discussion on binary represenation we know that dividing by 2 moves the last fraction one step to the left. As $\exp(2\pi i x) = 1$ for all integers x, we will only be left with the binary fraction part of j. Writing the tensor product on regular product form thus yield the product representation:

$$|j\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle \right) \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle \right) \cdots \left(|0\rangle + e^{2\pi i 0.j_1 \ldots j_n} |1\rangle \right). \tag{4.14}$$

Algorithm

To implement the quantum Fourier transform algorithm we will make use of the Hadamard gates, controlled R_k gates and swap gates. The R_k gate is defined by the matrix

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k}, \end{bmatrix} \tag{4.15}$$

which up to a global phase is equivalent to the $\hat{R}_z$ operator from section (3.2), since

$$\begin{bmatrix} e^{i\theta/2} & 0 \\ 0 & e^{i\theta/2}, \end{bmatrix} \begin{bmatrix} e^{i\theta/2} & 0 \\ 0 & e^{i\theta/2}, \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta}, \end{bmatrix} \tag{4.16}$$

so that $R_k \equiv \hat{R}_z(2\pi / 2^k)$.

    We now remind ourselves of the action of the Hadamard gate on the state $|j\rangle = |j_1 \ldots j_n\rangle$. Since $e^{2\pi i} = 1$ and $e^{2\pi i / 2} = -1$, we get

$$H |j_1 \ldots j_n\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.j_1} |1\rangle \right) |j_2 \ldots j_n\rangle. \tag{4.17}$$

Acting on the first qubit with a R_k conditional to the second qubit and set $k = 2$ gate gives the state

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i 0.j_1 j_2} |1\rangle \Big) |j_2 \ldots j_n\rangle. \tag{4.18}$$

Applying an R_k gate conditional to all the remaining qubits we end up in the state

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i 0.j_1 \ldots j_n} |1\rangle \Big) |j_2 \ldots j_n\rangle. \tag{4.19}$$

By repeating this procedure on all the other qubits we get the final state

$$\frac{1}{\sqrt{2^n}}\Big(|0\rangle + e^{2\pi i 0.j_1 \ldots j_n} |1\rangle \Big)\Big(|0\rangle + e^{2\pi i 0.j_2 \ldots j_n} |1\rangle \Big) \cdots \Big(|0\rangle + e^{2\pi i 0.j_n} |1\rangle \Big). \tag{4.20}$$

Notice that for this to be equivalent to equation (4.12) we need to swap all k-th first and last qubits with each other. A single swap gate can be done with three CNOT gates, thus giving us a total gate count of $\mathcal{O}(n^2)$ for this algorithm. In comparison the discrete Fourier transform needs $\mathcal{O}(n2^n)$ gates, where n is the number of bits in both cases.

The circuit for the quantum Fourier transform can be seen in figure (4.1).
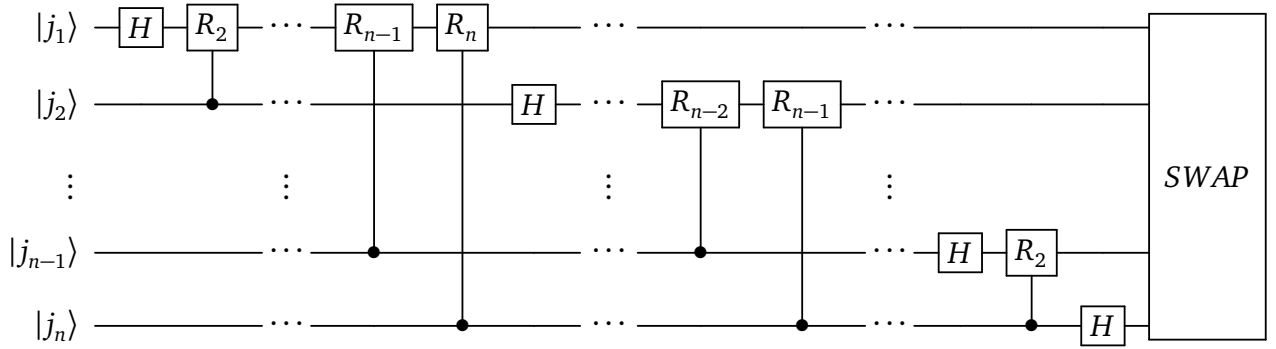
Figure 4.1: Quantum circuit for the quantum Fourier transform algorithm, involving controlled Hadamard gates and controlled R_k gates. The SWAP box represents swap gates between qubit i and qubit $n - i$.

4.2 Phase estimation

The motivation for the Phase estimation algorithm is quite simple; assume a unitary operator U with an eigenvector $|u\rangle$ and a corresponding eigenvalue $e^{2\pi i \lambda}$. As U can be implemented on a quantum computer where the qubits can be initialized to $|u\rangle$, there should be a way to measure λ. Some problems arise early on as we recognize that only some parts of λ can be measured on a quantum computer (from discussion on measurement). Writing $\lambda = n + \phi$, where $0 < \phi < 1$, we get

$$
\begin{aligned}
e^{2\pi i(n+\phi)} &= e^{2\pi i n} e^{2\pi i \phi} \\
&= \underbrace{\Big(\cos(2\pi n) + i \sin(2\pi n) \Big)}_{=1} \Big(\cos(2\pi \phi) + i \sin(2\pi \phi) \Big) \\
&= \cos(2\pi \phi) + i \sin(2\pi \phi) = e^{2\pi i \phi},
\end{aligned}
\tag{4.21}
$$

for any positive integer n. Thus, we can only measure ϕ which in the phase estimation algorithm is assumed to be estimated with the binary fraction given by equation (4.5).

Clearly we can encode the binary fraction on a set of t qubits. Large values of t will minimize the error of equation (4.5), but the restricted number of qubits on current quantum computers force us to favour smaller values. However, in many cases small values of t can yield appropriate accuracy. Luckily our aim for this project is in this category, where we want to estimate ground state energies of Hamiltonians measured in Hartrees.

4.2.1   Algorithm

We need to make use of two sets of qubits representing the phase to be estimated and the system to be simulated. The algorithm consists of several different parts and makes use of two sets of qubits we call the work register and simulation register. The work register consists of t qubits dependent of the desired accuracy of the estimation, and the simulation register consists of n qubits dependent on the operator U. Below we summarize the steps:

1. Initialize the qubits in the work register to all $|0\rangle$ and set up superposition.

2. Initialize the simulation register to the eigenvector $|u\rangle$ of U.

3. Apply the operator U to the simulation qubits conditional to the work qubits.

4. Apply the inverse quantum Fourier transform to the work register.

5. Measure the work register.

This is the same as can be shown in figure (4.2).
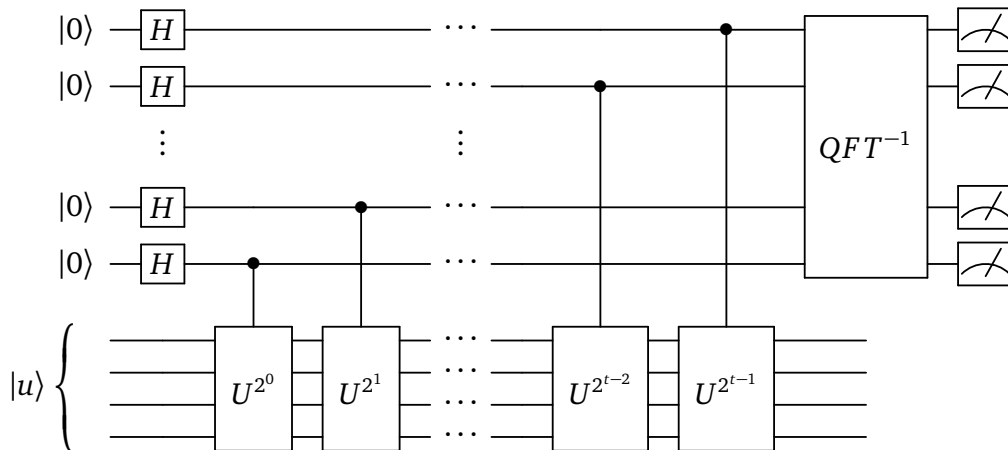
Figure 4.2: Quantum circuit for the phase estimation algorithm. The uppermost qubits make up the work register, initialized to the $|0\rangle$ state with a Hadamard gate. The lowermost qubits make up the simulation register, initialized to an eigenstate $|u\rangle$ of the Hamiltonian. The QFT box represents the inverse Quantum Fourier Transform performed on the work register before the measurements.

Here it is apparent that we need to act with U several times. To see why we start with the above assumptions that

$$U|u\rangle = e^{2\pi i \phi}|u\rangle, \tag{4.22}$$

where $\phi = 0.\phi_1 \ldots \phi_t$. Acting with U on the simulation register m times then have the action

$$U^m |u\rangle = e^{2\pi i \phi} U^{m-1} |u\rangle = \cdots = e^{2\pi i m \phi} |u\rangle. \tag{4.23}$$

From our discussion on the binary representation we know that multiplying a binary fraction with 2^m will move the fraction m times to the left. As an example we can do this 2^2 times giving

$$\begin{aligned}
U^{2^2} |u\rangle &= e^{2\pi i 2^2 \phi} |u\rangle \\
&= e^{2\pi i 2^2 0.\phi_1 \phi_2 \phi_3 \cdots \phi_t} |u\rangle \\
&= e^{2\pi i \phi_1 \phi_2 . \phi_3 \cdots \phi_t} |u\rangle \\
&= \underbrace{e^{2\pi i \phi_1 \phi_2}}_{=1} e^{2\pi i 0.\phi_3 \cdots \phi_t} |u\rangle \\
&= e^{2\pi i 0.\phi_3 \cdots \phi_t} |u\rangle
\end{aligned} \tag{4.24}$$

Then doing this operation conditional to one of the work qubits, that is in equal superposition after a Hadamard operation, we get

$$\frac{1}{\sqrt{2}} |u\rangle \left(|0\rangle + e^{2\pi i 0.\phi_3 \cdots \phi_t} |1\rangle \right). \tag{4.25}$$

Before applying the inverse QFT algorithm in figure (4.2) the two qubit registers will be in the state

$$\frac{1}{\sqrt{2^t}} |u\rangle \left(|0\rangle + e^{2\pi i 0.\phi_1 \phi_2 \cdots \phi_t} |1\rangle \right) \left(|0\rangle + e^{2\pi i 0.\phi_2 \cdots \phi_t} |1\rangle \right) \ldots \left(|0\rangle + e^{2\pi i 0.\phi_t} |1\rangle \right), \tag{4.26}$$

that we recognize as the product representation of the quantum Fourier transform. So applying the inverse quantum Fourier transform on the work register we will end up in the final state

$$|u\rangle |\phi_1 \phi_2 \ldots \phi_t\rangle, \tag{4.27}$$

and we can find an estimate for ϕ by measuring the work register.

4.2.2   Hamiltonian simulation

The problem of simulating a quantum system is often referred to as Hamiltonian simulation, as we simulate the evolution of an Hamiltonian operator corresponding to some quantum system. Hermitian operators, and thus Hamiltonians, cannot by definition be assumed to be unitary. This means we cannot just naively use the phase estimation algorithm with any Hamiltonian; we need to manipulate it so that it becomes unitary.

The time-dependent Schrödinger equation yields

$$i\hbar \frac{d}{dt} |\Psi\rangle = H |\Psi\rangle, \tag{4.28}$$

with the known solution

$$|\Psi(t)\rangle = e^{-iHt/\hbar} |\Psi(0)\rangle, \tag{4.29}$$

for a time-independent Hamiltonian, in what we call the Schrödinger picture of quantum mechanics. In this case the operator

$$U = e^{-iHt/\hbar}, \tag{4.30}$$

is the unitary time evolution operator. Furthermore we have that

$$H |\Psi\rangle = E |\Psi\rangle. \tag{4.31}$$

The time evolution operator thus yields

$$
\begin{aligned}
e^{-iHt/\hbar} |\Psi\rangle &= \sum_{j=0}^{\infty} \frac{(-iHt/\hbar)^j}{j!} |\Psi\rangle = \sum_{j=0}^{\infty} \frac{(-it/\hbar)^j}{j!} H^j |\Psi\rangle \\
&= \sum_{j=0}^{\infty} \frac{(-it/\hbar)^j}{j!} E^j |\Psi\rangle = \sum_{j=0}^{\infty} \frac{(-iEt/\hbar)^j}{j!} |\Psi\rangle \\
&= e^{-iEt/\hbar} |\Psi\rangle.
\end{aligned} \tag{4.32}
$$

In other words we should be able to estimate the eigenvalue of the Hamiltonian with the phase estimation algorithm. Combining the right-hand side of equation (4.22) and (4.32) we get

$$e^{-iEt/\hbar} |\Psi\rangle = e^{2\pi i \phi} |\Psi\rangle, \tag{4.33}$$

giving

$$
\begin{aligned}
&-iEt/\hbar = 2\pi i \phi \\
&\Rightarrow \phi = -\frac{Et}{2\pi\hbar}.
\end{aligned} \tag{4.34}
$$

The phase estimation will find $0 < \phi < 1$ and so we need to ensure that $E < 0$. We can overcome this by simply subtracting a value $E_{\max} > E$ from the Hamiltonian so that

$$H_{\max} = H - E_{\max}, \tag{4.35}$$

giving

$$e^{-iH_{\max}t/\hbar} |\Psi\rangle = e^{-i(E-E_{\max})t/\hbar} |\Psi\rangle. \tag{4.36}$$

$$\Rightarrow \phi = -\frac{(E-E_{\max})t}{2\pi\hbar}. \tag{4.37}$$

This is one of the disadvatages of the Hamiltonian simulation with phase estimation, since we need to set an upper limit for the eigenvalues of the Hamiltonian. Moreover we also need to ensure that we cover the whole eigenvalue spectrum so we are able to represent the lowest eigenvalue. We do this by setting $E = E_{\min}$ giving

$$\phi = -\frac{(E_{\min} - E_{\max})t}{2\pi\hbar} = \frac{(E_{\max} - E_{\min})t}{2\pi\hbar}, \tag{4.38}$$

where we know that $\phi < 1$, that is

$$\frac{(E_{\max} - E_{\min})t}{2\pi\hbar} < 1 \quad \Rightarrow \quad t < \frac{2\pi\hbar}{E_{\max} - E_{\min}} \tag{4.39}$$

Trotter truncation

However, in quantum many-body physics the Hamiltonian is usually assumed to be a sum of local interactions

$$H = \sum_{k} H_k, \tag{4.40}$$

with

$$E \left| \Psi \right\rangle = \sum_k H_k \left| \Psi \right\rangle = \sum_k E_k \left| \Psi \right\rangle. \tag{4.41}$$

From the Baker-Campbell-Hausdorff formula we know that

$$e^{H_1 + H_2} = e^{H_1} e^{H_2}, \tag{4.42}$$

only if $[H_1, H_2] = 0$. Equation (4.32) is true only if all individual terms of the Hamiltonian commute with each other, that is $[H_p, H_q] = 0$ for all $p \neq q$. Generally this is not the case, and so

$$e^{-i \sum_k H_k t/\hbar} \neq \prod_k e^{-iH_k t/\hbar}, \tag{4.43}$$

though this would be the desired form of the exponent as it would be equivalent to equation (4.32). We can make an approximation to this by using the Trotter product formula introduced in section (3.7). Truncating the time evolution operator at some value ρ and setting $\Delta t = t/\rho$ gives

$$e^{-i \sum_k H_k t/\hbar} = \left(\prod_k e^{-iH_k \Delta t/\hbar} \right)^\rho + \mathcal{O}(\Delta t^2), \tag{4.44}$$

where increasing ρ will decrease the error.

Pauli gates

An advantage of the Hamiltonian simulation with phase estimation is that the Hamiltonian often can be decomposed into Pauli gates. We have introduced the rotation gates which essentially are a generalization of the Pauli gates. Because of this we can replace the U^{2^k}-operation with a single U-operation where the 2^k phase is added with the rotation gates. This will greatly reduce the circuit depth of the algorithm.

To see why the above statement is true we first need to show how we implement the Pauli gates in an exponent, as this is not straight forward. Looking at the Pauli gates we see that we can define the following identities

$$\hat{X} = \hat{H}\hat{Z}\hat{H}, \quad Y = \hat{S}\hat{H}\hat{Z}\hat{H}\hat{S}^\dagger, \tag{4.45}$$

where $\hat{H}$ is the Hadamard gate and so all Pauli gates can be written in terms of the $\hat{Z}$-gate. Next we take notice of the relation

$$\begin{aligned}
\hat{U}^\dagger e^{-i\hat{O}t} \hat{U} &= \hat{U}^\dagger \left(\cos(t)\hat{\mathbb{1}} - i\sin(t)\hat{O} \right) \hat{U} \\
&= \cos(t)\hat{U}^\dagger \hat{\mathbb{1}} \hat{U} - i\sin(t)\hat{U}^\dagger \hat{O}\hat{U} \\
&= \cos(t)\hat{\mathbb{1}} - i\sin(t)\hat{U}^\dagger \hat{O}\hat{U} \\
&= e^{-i\hat{U}^\dagger \hat{O}\hat{U}t},
\end{aligned} \tag{4.46}$$

for any unitary operator $\hat{U}$ and operator $\hat{O}$. We can then define the identities

$$e^{-i\hat{X}t} = \hat{H}e^{-i\hat{Z}t}\hat{H}, \quad e^{-i\hat{Y}t} = \hat{S}\hat{H}e^{-i\hat{Z}t}\hat{H}\hat{S}^\dagger. \tag{4.47}$$

Then lastly, assume we want to implement the operator

$$e^{-it\hat{X}_1 \otimes \hat{Z}_2 \otimes X_3} = (\hat{H}_1 \otimes \hat{\mathbb{1}}_2 \otimes \hat{H}_3)e^{-it\hat{Z}_1 \otimes \hat{Z}_2 \otimes \hat{Z}_3}(\hat{H}_1 \otimes \hat{\mathbb{1}}_2 \otimes \hat{H}_3), \tag{4.48}$$
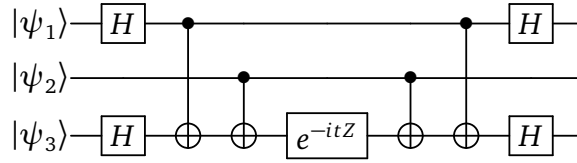
on a three qubit system. The eigenvalues of the $\hat{Z}$ gate measured in the computational basis are

$$\hat{Z}\left|\psi\right\rangle = \begin{cases} +1 & \text{if } \left|\psi\right\rangle = \left|0\right\rangle, \\ -1 & \text{if } \left|\psi\right\rangle = \left|1\right\rangle. \end{cases} \tag{4.49}$$

For the three qubit case we then get

$$\hat{Z}_1 \otimes \hat{Z}_2 \otimes \hat{Z}_3 \left|\psi_1 \psi_2 \psi_3\right\rangle = \begin{cases} +1 & \text{if } \sum_k \psi_k \text{ is even}, \\ -1 & \text{if } \sum_k \psi_k \text{ is odd}. \end{cases} \tag{4.50}$$

Clearly the operator is equivalent to the following circuit

since the CNOT gates store the parity in the last qubit. Looking at the Z-rotation gate

$$\hat{R}_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}, \tag{4.51}$$

we can replace the exponent gate in the circuit with $\hat{R}_z(2t)$.

4.3   Variational quantum eigensolver

In the previous sections we have disregarded some important aspects of current and near-term quantum computers:

1. *They do not have a large amount of qubits.*
   As a result we would like to make use of all the available qubits. In this sense the phase estimation algorithm is not ideal, as it makes use of two qubit registers.

2. *The coherence time of a qubit is short.*
   We have to keep the coherence time in mind as it affects the maximal circuit depth for which we can make reliable measurements. Again, the phase estimation algorithm is not ideal because of the multiple U operations on the simulation register.

3. *The available gates and measurements are noisy.*
   This will clearly affect all algorithms. However, the gate number will increase the error and so a good strategy is to act with few gates between each measurement.

The variational quantum eigensolver [6] is an algorithm that was developed with these points in mind. Its efficiency is restricted to a class of operators that can be decomposed into a sum of simple operators

$$\hat{O} = \sum_k \hat{O}_k, \tag{4.52}$$

Indeed, using e.g. the Jordan-Wigner transformation, an electronic Hamiltonian is such an operator, where the number of terms is polynomial in the size of the system

[12]. If we have an initial state of qubits $|\Phi\rangle$, we want to prepare our trial ground-state wave function $|\Psi(\theta)\rangle$, dependent on some set of variables θ, by an action of a suitable unitary operator $\hat{U}(\theta)$.

$$|\Psi(\theta)\rangle = \hat{U}(\theta)|\Phi\rangle \tag{4.53}$$

In this manner we can parametically explore the Hilbert space close to the reference state $|\Phi\rangle$. By variationally optimizing $\hat{U}$ we can find estimates to the ground state energy of the Hamiltonian according to the variational principle

$$\frac{\langle\Psi(\theta)|\hat{H}|\Psi(\theta)\rangle}{\langle\Psi(\theta)|\Psi(\theta)\rangle} \leq E_0 \tag{4.54}$$

To implement this, we divide our algorithm into two subroutines; one quantum and one classical. The quantum subroutine prepares our state followed by multiple measurements of equation (4.54) in an appropriate basis. The classical subroutine will optimize the expectation value, with gradient descent or such [13], by varying the parameters. The process is then repeated.

4.3.1 Quantum subroutine

Further on we assume the state $|\Psi(\theta)\rangle$ to be normalized, and equation (4.54) can be written

$$\langle E\rangle_\theta = \langle\Psi(\theta)|\hat{H}|\Psi(\theta)\rangle \leq E_0. \tag{4.55}$$

If we assume the Hamiltonian to be on second quantized form we can write it as a k-fold Hamiltonian, and the expectation value can be written

$$\langle E\rangle_\theta = \sum_k \langle\Psi(\theta)|\hat{H}_k|\Psi(\theta)\rangle, \tag{4.56}$$

where each k-fold operator will be a set of creation and annihilation operators and an interaction integral a_k. After the Jordan-Wigner transformation all $\hat{H}_k$ become a sum of Pauli strings $\sum_i \mathscr{P}_k^i$. The expectation value can thus be written

$$\langle E\rangle_\theta = \sum_k a_k \sum_i \langle\Psi(\theta)|\mathscr{P}_{k_i}|\Psi(\theta)\rangle. \tag{4.57}$$

Pauli operators all have eigenvalue $p_{k_i} = \pm 1$ in their respective bases, which yields

$$\langle\Psi(\theta)|\mathscr{P}_{k_i}|\Psi(\theta)\rangle = p_{k_i}\langle\Psi(\theta)|\Psi(\theta)\rangle. \tag{4.58}$$

Equation (4.57) then becomes

$$\langle E\rangle_\theta = \sum_k a_k \sum_i p_{k_i}\langle\Psi(\theta)|\Psi(\theta)\rangle. \tag{4.59}$$

4.3.2 Classical subroutine

After measuring a value for equation (4.59) we want to update the parameters θ so that we move towards the true ground state energy E_0. Since we can only make a finite number of measurements we need to redefine the expectation value as an objective function

$$L(\theta) = \langle E\rangle_\theta + \text{noise}, \tag{4.60}$$

where the noise also will include errors from the quantum hardware. This typically means we can not rely on gradient based optimization methods to minimize the objective function. Usually we turn to classical non-linear optimization methods [12]. Some of these methods can be robust to small amounts of noise in the objective function, even for a search in high-dimensional spaces.

4.3.3   Unitary coupled cluster ansatz

When we prepare our trial wave function $|\Psi(\theta)\rangle$ from an initial reference state $|\Phi_0\rangle$, we need to ensure that the particle number is conserved. That is, we need the unitary operation in equation (4.53) to commute with the number operator $\hat{N} = \sum_p \hat{a}_p^\dagger \hat{a}_p$. This is a useful property of the unitary coupled cluster ansatz introduced in section (2.9.3)

$$|\Psi(\theta)\rangle = e^{\hat{T}-\hat{T}^\dagger} |\Phi_0\rangle. \tag{4.61}$$

A quantum state prepared by this operator has no efficient classical implementation. Unitary state preparation is a natural operation on quantum computers so we are motivated to use this as an ansatz in the VQE algorithm. As with traditional coupled cluster we truncate the operator at singles and doubles terms

$$\hat{T}_1 - \hat{T}_1^\dagger = \sum_{ia} t_i^a \left(\hat{a}_a^\dagger \hat{a}_i - \hat{a}_i^\dagger \hat{a}_a \right), \tag{4.62}$$

$$\hat{T}_2 - \hat{T}_2^\dagger = \sum_{ijab} t_{ij}^{ab} \left(\hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i - \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_b \hat{a}_a \right). \tag{4.63}$$

Using the anti-commutation relation

$$\hat{a}_p \hat{a}_q^\dagger = \hat{a}_q^\dagger \hat{a}_p + \delta_{pq}, \tag{4.64}$$

we can see that both operators commute with the number operator since

$$\left[\hat{a}_a^\dagger \hat{a}_i, \hat{N} \right] = \sum_p \left(\hat{a}_a^\dagger \hat{a}_i \hat{a}_p^\dagger \hat{a}_p - \hat{a}_p^\dagger \hat{a}_p \hat{a}_a^\dagger \hat{a}_i \right) = \sum_p \left(\hat{a}_a^\dagger \delta_{ip} \hat{a}_p - \hat{a}_p^\dagger \delta_{ap} \hat{a}_i \right) = 0 \tag{4.65}$$

$$\left[\hat{a}_i^\dagger \hat{a}_a, \hat{N} \right] = \sum_p \left(\hat{a}_i^\dagger \hat{a}_a \hat{a}_p^\dagger \hat{a}_p - \hat{a}_p^\dagger \hat{a}_p \hat{a}_i^\dagger \hat{a}_a \right) = \sum_p \left(\hat{a}_i^\dagger \delta_{ap} \hat{a}_p - \hat{a}_p^\dagger \delta_{ip} \hat{a}_a \right) = 0 \tag{4.66}$$

and

$$\left[\hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i, \hat{N} \right] = \sum_p \left(\hat{a}_a^\dagger \hat{a}_b^\dagger \delta_{jp} \hat{a}_i \hat{a}_p + \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \delta_{pi} \hat{a}_p - \hat{a}_p^\dagger \hat{a}_a^\dagger \delta_{pb} \hat{a}_j \hat{a}_i - \hat{a}_p^\dagger \delta_{pa} \hat{a}_b^\dagger \hat{a}_j \hat{a}_i \right)$$

$$= 0 \tag{4.67}$$

$$\left[\hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_b \hat{a}_a, \hat{N} \right] = \sum_p \left(\hat{a}_i^\dagger \hat{a}_j^\dagger \delta_{bp} \hat{a}_a \hat{a}_p + \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_b \delta_{pa} \hat{a}_p - \hat{a}_p^\dagger \hat{a}_i^\dagger \delta_{pj} \hat{a}_b \hat{a}_a - \hat{a}_p^\dagger \delta_{pi} \hat{a}_j^\dagger \hat{a}_b \hat{a}_a \right)$$

$$= 0 \tag{4.68}$$

Since $[(\hat{a}_a^\dagger \hat{a}_i)^2, \hat{a}_p^\dagger \hat{a}_p] = \left[\hat{a}_a^\dagger \hat{a}_i, [\hat{a}_a^\dagger \hat{a}_i, \hat{a}_p^\dagger \hat{a}_p] \right] = 0$, with the same for the two-body operator, all terms in the exponent expansion of the cluster operator will commute with the number operator

$$\left[e^{\hat{T}-\hat{T}^\dagger}, \hat{N} \right] = 0. \tag{4.69}$$

Next we can map equation (4.62) and 4.63 to quantum gates using the Jordan-Wigner transformation. In appendix (B.2) we show that we get

$$t_i^a(a_a^\dagger a_i - a_i^\dagger a_a) \equiv \frac{it_i^a}{2}\left(\mathscr{P}_{yx}^{ia} - \mathscr{P}_{xy}^{ia}\right)\left[\bigotimes_{k=i+1}^{a-1} \sigma_z^k\right] \tag{4.70}$$

$$
\begin{aligned}
t_{ij}^{ab}(a_a^\dagger a_b^\dagger a_j a_i - a_i^\dagger a_j^\dagger a_b a_a) \equiv \frac{it_{ij}^{ab}}{8} &\left[\bigotimes_{k=i+1}^{j-1} \sigma_z^k\right]\left[\bigotimes_{k=a+1}^{b-1} \sigma_z^k\right] \\
&\left(\mathscr{P}_{xxxy}^{ijab} + \mathscr{P}_{xxyx}^{ijab} - \mathscr{P}_{xyxx}^{ijab} + \mathscr{P}_{xyyy}^{ijab}\right. \\
&\left.- \mathscr{P}_{yxxx}^{ijab} + \mathscr{P}_{yxyy}^{ijab} - \mathscr{P}_{yyxy}^{ijab} - \mathscr{P}_{yyyx}^{ijab}\right).
\end{aligned}
\tag{4.71}
$$

However, as discussed in context with the phase estimation algorithm, an exponential operator including non-commuting terms must be expanded using the Campbell-Baker-Hausdorff formula. Unlike traditional coupled cluster this does not truncate naturally, so we have to make use of the Trotter truncation. We then end up with a product of exponents for each term. This allows us to reorder them in our circuit in an optimal manner to reduce the number of gates, as shown in appendix (B.2).

The cluster parameters can be initialized using the amplitudes obtained from second order Møller-Plesset perturbation theory. In comparison to other state preparations relying on random initialization, the UCC ansatz thus gains an advantage with a good initial guess for the parameters

$$t_i^a = 0, \tag{4.72}$$

$$t_{ij}^{ab} = \frac{\langle ab|\,\hat{v}\,|ij\rangle_{AS}}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}. \tag{4.73}$$

4.3.4   Reducing the number of state preparations

One disadvantage of using the UCC ansatz is that it requires a relatively large number of gates to prepare the state, compared to the number of gates that will be measured from the Jordan-Wigner transformed Hamiltonian. This makes the state preparation the bottleneck of the algorithm. For a Jordan-Wigner transformed two-body operator we need to measure maximum 16 Pauli-strings. Each measurement needs its own state preparation by the UCC ansatz. To get a good approximation to the state coefficients we also need to measure the same Pauli-string multiple times. Thus, for large sets of two-body operators we will need a large number of state preparations. A naive implementation of the UCC ansatz scales as $\mathcal{O}(n^4)$, thus we are interested in reducing the number of state preparations. In section (2.2) we showed that commuting operators have simultaneous eigenstates, and in section (3.6) we showed how to recognize commuting Pauli-strings. Thus, we can group Pauli-strings into commuting groups that can be measured simultaneously to reduce the number of state preparations. We then need to consider how this will affect our estimate of the expectation value of the Hamiltonian.

Covariance of commuting sets of Pauli-strings

If M is the number of terms in the transformed Hamiltonian, then to achieve an accuracy of ϵ in our estimate of the expectation value

$$\langle \hat{H} \rangle = \sum_k \langle \hat{H}_k \rangle, \tag{4.74}$$

the expected number of measurements is given by

$$n_{\text{expect}} = M \sum_k \frac{\text{Var}[\hat{H}_k]}{\epsilon^2}. \tag{4.75}$$

For a finite set of n of measurements $\{m_i\}$ we can approximate equation (4.74) as the estimator

$$\mathbb{E}[\hat{H}] = \sum_k \mathbb{E}[\hat{H}_k] = \sum_k \frac{1}{n} \sum_i^n m_i^{(k)}. \tag{4.76}$$

Since each measurement need an independent state preparation, the covariance between each measurement have to be $\text{Cov}[\mathbb{E}[\hat{H}_i], \mathbb{E}[\hat{H}_j]] = 0$ for $i \neq j$. The variance of the total estimator then becomes

$$\text{Var}[\mathbb{E}[\hat{H}]] = \sum_k \text{Var}[\mathbb{E}[\hat{H}_k]]. \tag{4.77}$$

In other words, by measuring all the terms independently the variance of the total estimator is not affected by the covariance between any of the terms. If we were to group commuting terms together and measure them in a simultaneous eigenbasis, we need to ensure that the total variance is not affected by the covariance of terms within these groups.

Suppose we group the terms into G commuting groups C_k, then the expected number of measurements becomes

$$n_{\text{expect}} = G \sum_k \frac{\text{Var}[C_k]}{\epsilon^2}. \tag{4.78}$$

Each group is a sum of n Pauli-strings, with the variance defined as

$$\text{Var}\left[\sum_k^n \mathscr{P}_k\right] = \sum_k^n \text{Var}[\mathscr{P}_k] + 2 \sum_{k \neq l}^n \text{Cov}[\mathscr{P}_k, \mathscr{P}_l]. \tag{4.79}$$

McClean et al. [12] argue that, given a quantum state $|\Psi\rangle$, there can exist two different sets of commuting groups where the set with the most groups give the lowest variance. This is not desirable, as we then need to be careful when we group Pauli-strings together and we can not just choose the set with the least groups. However, Gokhale et al. [14] show that since we have no prior on $|\Psi\rangle$ the expectation of the covariance will be zero when it is taken over a uniform distribution over all possible state vectors. The best way to reduce equation (4.78) is then to choose the set with the least amount of groups. Note that we can only assume this to be true for most states since there do exist some states where it is not (as shown in both [12] and [14]), and the covariance can be measured during the VQE minimization to ensure that the group set is the best or changed into another.

Part II

Implementation

Chapter 5

Structure of the implementation

The main goal of this thesis is to transform a general second quantized Hamiltonian into a quantum circuit and execute it on a quantum computer. The limitations of near-term quantum computers discussed in section (4.3), restricts us to be conscious of efficient implementations of quantum circuits. Particularly, qubits can only be assumed to exist in a certain state for a short period of time, and so we need to shorten the depth of the circuit where it is possible. In order to do this in a simple manner we have implemented the `quantum_circuit` module which handles gates and circuits, including transformations and simple optimization. It includes the `QuantumCircuit` class from which we can create quantum circuits by applying gates. The Jordan-Wigner transformation of creation and annihilation operators is handled here. This turned out somewhat inefficient when it considers a general Hamiltonian, but it serves as a reasonable approach when we do not know anything about the symmetries of the interaction integrals. A more efficient approach is included in the `Hamiltonian` class for the special case of a fermionic Hamiltonian with diagonal one-body matrix. The transformed circuits is included as an instance of the `CircuitList` class which is an input to the VQE and QPE class in the `quantum_algorithms` module. This module is concerned with two quantum algorithms; Quantum Phase Estimation (QPE) and Variational Quantum Eigensolver (VQE) discussed in section (4.2) and (4.3) respectively.

The VQE is dependent on an optimization method and an ansatz for the initial state of the qubits. The optimization methods are implemented as a set of callable classes, so that they act the same in the VQE algorithm. These include classical optimization methods as well as a method taking advantage of the nature of the ansatz. We have implemented three different ansätze, the Unitary Coupled Cluster (UCC) along with more efficient and problem specific state preparations. These are constructed in a general manner so that other ansätze can easily be included. To be able to execute any of the algorithms on a quantum computer we have made them into subclasses of a `QuantumAlgorithm` superclass that handles all specifications related to simulations of the quantum circuits. IBM offers several quantum computers to execute quantum ciruits on. In order to prepare for these execution we can access the architecture of the qubits and noise models related to the hardware. These are accessed with the `QuantumComputer` class as an input to the `QuantumAlgorithm` class.

The structure of our implementation is shown in figure (5.1), where a line between nodes represents a dependency. Further on we will explain in more detail how the code is structured and how all the parts connect to one another. We will start by introducing a simple system called the Pairing model. This will be used as an example through-

out this part, to demonstrate how the code works and how some decisions affect the performance of the algorithms. Then we will give a brief review of IBM's `Qiskit` package, handling the circuit simulations and executions on quantum computers. Before we explain our implementation of the quantum algorithms and everything connected to them. At the end we will introduce two new systems that we will examine further in the results.

Lastly, all code is available in the two GitHub repositories https://github.com/heineaabo/Quantum and https://github.com/heineaabo/QuantumCircuit.
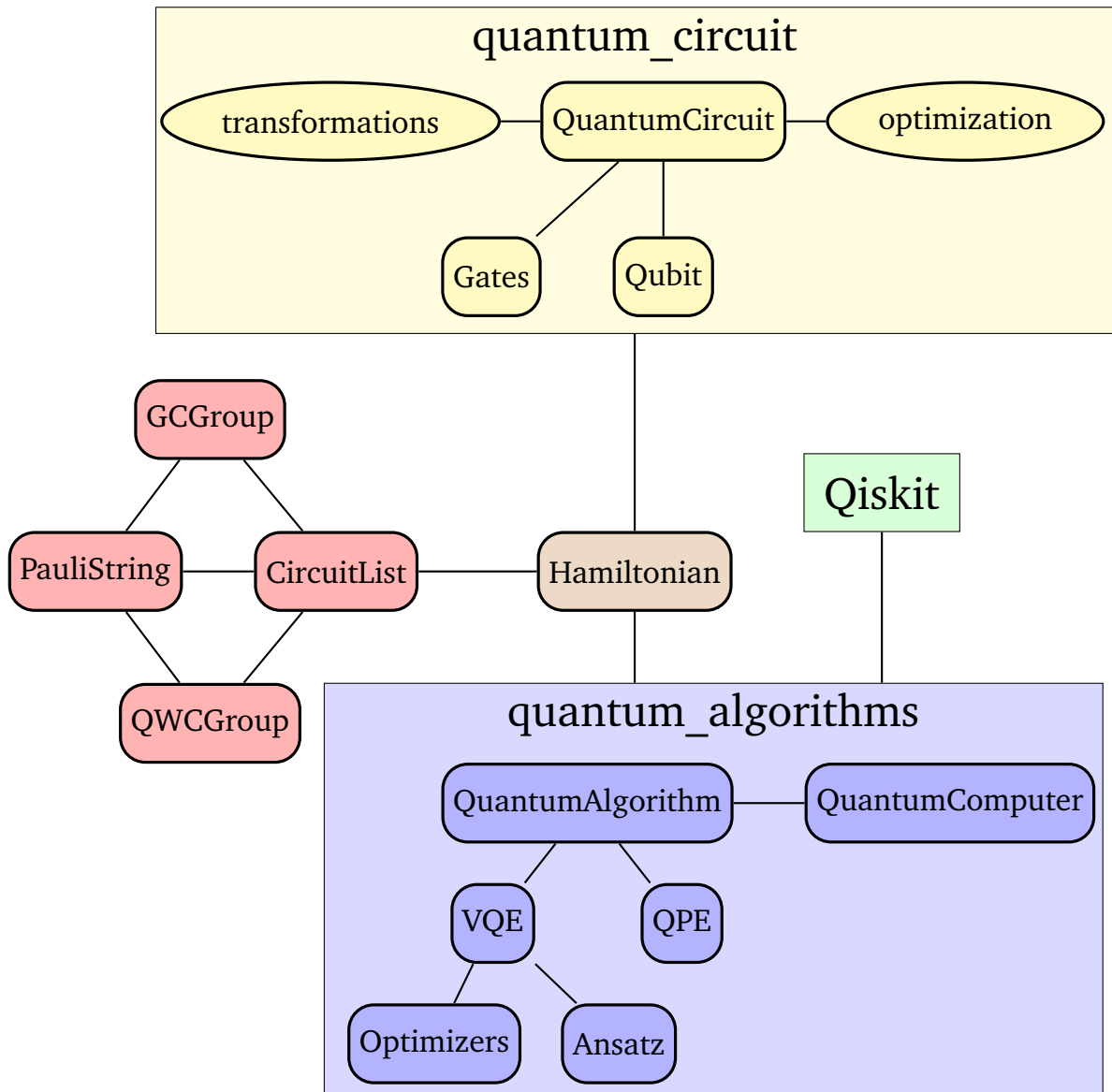
Figure 5.1: Flow chart of code implemented for this thesis. The rounded rectangles represents classes, where some belong to a module being the rectangles surrounding them. The ellipses represents sets of functions with a certain purpose. The lines indicate the dependencies between two instances.

5.1   The pairing model

To demonstrate our implementation we will make use of the simple pairing model. This models an ideal fermionic system where particles form pairs, and no pairs can be broken. That is, only pair excitations are allowed. The pairing model Hamiltonian can be written on the form

$$\hat{H} = \sum_p \epsilon_p a_p^\dagger a_p - \frac{1}{2} g \sum_{pq} a_{p+}^\dagger a_{p-}^\dagger a_{q-} a_{q+} \tag{5.1}$$

where ϵ_p is the single-particle energy of level $p = 1, 2, \ldots$ and g is the pairing strength. Ideally the pairing strength can be treated as a constant, however for a realistic atomic nucleus this is not the case. Still it serves as a useful tool to study the pairing interaction. Assuming equally spaced single-particle orbitals by a constant δ we can write $\epsilon_p = (p-1)\delta$, and introduce $\sigma = \pm$ representing the possible spin values. Next we recognize the number operator $\hat{n}_p = a_p^\dagger a_p$. The two-body interaction can only excite a pair of particles at a time, so it is convenient to introduce the pair creation and annihilation operators $P_p^+ = a_{p+}^\dagger a_{p-}^\dagger$ and $P_p^- = a_{p-} a_{p+}$. This yields the simple expression

$$\hat{H} = \delta \sum_{p\sigma} (p-1)\hat{n}_p - \frac{1}{2} g \sum_{pq} P_p^+ P_q^- \tag{5.2}$$

Calculations of the matrix elements for the Full Configuration Interaction method is shown in appendix (B.3).
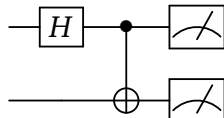
5.2   Qiskit

Several companies developing quantum computers have in addition developed their own framework for quantum programming. In this chapter we will go through the basics of how the quantum algorithms can be implemented using a popular python package for quantum computing. As well as some of the difficulties to overcome in order to run a quantum circuit on an actual quantum computer. We will make use of the IBM's `Qiskit` package when we develop our quantum algorithms. It provides a simple programming language with both direct access to simulators of quantum circuits and cloud access to physical quantum computers. This way we can write our algorithms in terms of the quantum circuit model, without having to worry too much about problems that arise when a circuit is to be compiled to quantum hardware.

The quantum circuits will be simulated with `Qiskit`'s QASM simulator. This is a noisy simulator of a quantum circuit. Quantum Assembly (QASM) is a language for intermediate representation of quantum circuits, but not of our concern since this is dealt with by `Qiskit`. A noisy simulator is useful since we can not expect quantum circuits and measurements to be error free, especially when we deal with a finite number of measurements. We can then get an apprehension of the possible outcome of a circuit on a quantum computer.

Currently there are ten available quantum computers from IBM that can be accessed using `Qiskit`: a 1-qubit computer, a 15-qubit computer and eight 5-qubit computers. We will focus on the 5-qubit computers for our implementation on physical quantum computers. In order to prepare our algorithms for implementations on these

computers we have to be aware of the limitations they include; the limited connectivity between the qubits and the noise from gate actions and measurements. In `Qiskit` each quantum computer is associated with a coupling map and a noise model so we can simulate the expected behaviour of a quantum circuit before running it on the physical quantum computer.

The creation and execution of quantum circuits in `Qiskit` are handled in the VQE and QPE classes. Below we show an example of how this can be done for the following circuit,

and get the number of times each state is measured using the QASM simulator.

```python
from qiskit import QuantumCircuit, QuantumRegister,
                    ClassicalRegister, Aer

qasm = Aer.get_backend('qasm_simulator')

qb = QuantumRegister(2)
cb = ClassicalRegister(2)
qc = QuantumCircuit(qb,cb)

qc.h(qb[0])
qc.cx(qb[0],qb[1])

qc.measure(qb,cb)
measurement = execute(qc,qasm,shots=1024).result().get_counts()
```

5.2.1 Basis gates

If we wish to execute a circuit on a given quantum computer, the circuit should be compiled down to a universal set of basis gates corresponding to a quantum computer. Different types of quantum computers make use of different gate sets that can be utilized for more efficient circuits, as demonstrated by Nam et al[15] for a trapped-ion quantum computer. IBM's quantum computers make use of a more general gate set involving three new gates $\hat{U}_1$, $\hat{U}_2$ and $\hat{U}_3$. Togheter with the CNOT gate, this is a universal gate set. They are represented by the matrices

$$\hat{U}_1(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}, \tag{5.3}$$

$$\hat{U}_2(\phi,\lambda) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{bmatrix}, \tag{5.4}$$

$$\hat{U}_3(\theta,\phi,\lambda) = \begin{bmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{bmatrix}. \tag{5.5}$$

We see that the $\hat{U}_1$ is a phase gate, only inducing a phase to a qubit. The $\hat{U}_2$ can create superpositions, while the $\hat{U}_3$ is a generalization of both. In reality $\hat{U}_1$ and $\hat{U}_2$ are just special cases of the $\hat{U}_3$ gate where we have

$$\hat{U}_3(0,0,\lambda) = \hat{U}_1(\lambda), \tag{5.6}$$

$$\hat{U}_3(\frac{\pi}{2},\phi,\lambda) = \hat{U}_2(\phi,\lambda). \tag{5.7}$$

Thus the $\hat{U}_3$ can approximate any single qubit gate. We will not refer to $\hat{U}_1$, $\hat{U}_2$ and $\hat{U}_3$ further on, but in our implementation we will use them exclusively. Below we list the gate identities with the gates discussed in section (3.2).

$$
\begin{aligned}
\hat{X} &= \hat{U}_3(\pi, 0, \pi), & \hat{R}_x(\theta) &= \hat{U}_3(\theta, -\frac{\pi}{2}, \frac{\pi}{2}), \\
\hat{Y} &= \hat{U}_3(\pi, \frac{\pi}{2}, \frac{\pi}{2}), & \hat{R}_y(\theta) &= \hat{U}_3(\theta, 0, 0), \\
\hat{Z} &= \hat{U}_1(\pi), & \hat{R}_z(\theta) &= \hat{U}_1(\theta), \\
\hat{S} &= \hat{U}_1(\frac{\pi}{2}), & \hat{S}^\dagger &= \hat{U}_1(-\frac{\pi}{2}), \\
\hat{H} &= \hat{U}_2(0, \pi),
\end{aligned}
$$

5.2.2 Coupling map

All 5-qubit computers have the same so called coupling map, shown in figure (5.2). Each node represents a physical qubit. The arrows describe the connection between the qubits, showing the qubits that can be acted on with a CNOT gate. Clearly not all qubits are connected, but all connections are in both directions. In Qiskit qubits contained in a `QuantumRegister` instance are called virtual qubits. We can map the virtual qubits to the physical qubits. Later we will make use of the connectivity between
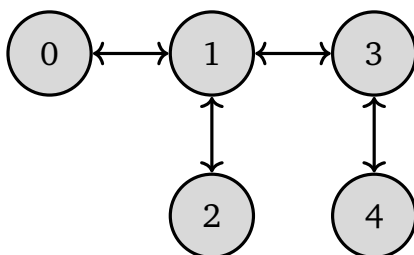

Figure 5.2: Coupling map of IBM's 5-qubit computers. Each index label a physical qubit. Each directional arrow represents the connectivity between the qubits.

the nodes 0-3 in figure (5.2). This can be done in a simple manner in Qiskit

```
from qiskit import QuantumRegister

qb = QuantumRegister(4)

layout = {0: qb[0],1:qb[3],2:qb[1],3:qb[2]}
```

5.2.3 Noise model

Executing a quantum circuit on a quantum computer is associated with many errors. The study of these errors is an active area of research, and so we can only include approximation to real errors when we do simulations. Qiskit contains noise models for all the IBM quantum computers, taking into account the following errors

- The error associated with the basis gates on each qubit.

- The gate length of the basis gate on each qubit.

- The relaxation time constants related to the the decay and dephasing of each qubit.

- The readout error of each qubit.

A noise model can be passed on to the QASM simulator in order to simulate a quantum circuit executed on a real quantum computer.

Error mitigation of measurement noise

When we execute a circuit on a real quantum computer we must assume measurement noise because of the readout error of the qubits. `Qiskit` provide an error mitigation scheme that inputs the result of a measurement. The measurement statistics are calibrated in terms of errors from measurements of the computational basis states, where we know the expected outcome of an ideal measurement. We then make noisy measurements of all prepared computational basis states. For a system of two qubits these are the four circuits given in figure (5.3).
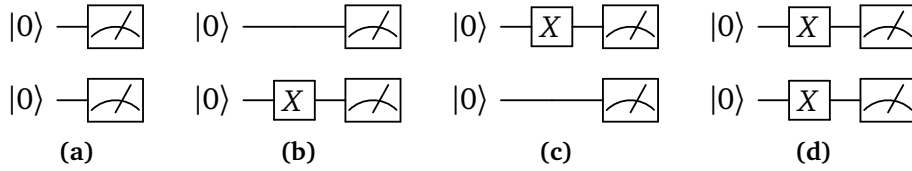
Figure 5.3: Measured circuits for two qubits in Qiskit's measurement noise mitigation scheme. State preparation of the states (a) $|00\rangle$, (b) $|01\rangle$, (c) $|10\rangle$ and (d) $|11\rangle$.

5.3   Quantum circuit

In order to solve the time-independent Schrödinger equation on a quantum computer we need to map the Hamiltonian to a quantum circuit. A second quantized Hamiltonian can be written as sum of k interaction terms

$$\hat{H} = \sum_k \hat{H}_k^{n_k}, \tag{5.8}$$

where n_k denotes the number of creation and annihilation operators in $\hat{H}_k$. Using the Jordan-Wigner transformation each $\hat{H}_k$ will be transformed into 2^{n_k} circuits. These circuits can either be used as a direct input to the VQE algorithm, or exponentiated as discussed in section (4.2) as an input for the QPE algorithm. This motivates the use of an independent `quantum_circuits` module for calculating the transformation of the Hamiltonian into quantum circuits.

    To handle the implementation of gates we have included the `QuantumCircuit` class, with an additional simple gate reduction strategy. All gates are stored in instances of a `Qubit` class. A quantum circuit can then be created by applying gates, and subsequently optimized to reduce the number of gates. It was also intended that certain transformations could be done in terms of circuit identities, to reduce the gate number even more where it is possible. This way it could serve as a framework for designing more efficient circuits. However, because the circuits resulting from the Hamiltonian transformation are small in terms of gate count, it would only prove useful for

exponential operators where we seek to remove CNOT gates. Therefore this has not been further implemented, since even well optimized QPE circuits must be consider unrealistic for executions on near-term quantum computers.

5.3.1   Gates and gate reduction

We have implemented a set of gates that will be used in the QPE and VQE algorithms. In order to optimize the circuits, hard-coded matrix products have been implemented. Figure (5.4) shows a circuit and how the gate reduction works in three different scenarios: where the product result in an identity gate, where the product is equivalent to another gate (including a complex factor), and where the product is not equivalent to another gate. All gates are mapped to a class function of the `QuantumCircuit`
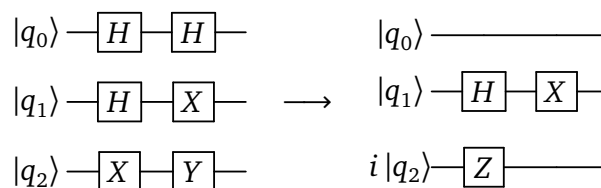

Figure 5.4: Example circuit before and after optimization in the `quantum_circuit` module.

class so that applying gates to a circuit is simple, similar to the syntax used in `Qiskit`. Below we show examples of how to act with different gates.

```python
from math import pi
from quantum_circuit import QuantumCircuit

qc = QuantumCircuit(n_qubits)

qc.x(q)        # Apply Pauli-X gate to qubit q
qc.h(q)        # Apply Hadamard gate to qubit q
qc.rx(q, pi/2) # Apply pi/2-rotation to qubit q
qc.cx(q1, q2)  # Apply CNOT gate with qubit q1 as control
               #             and qubit q2 as target
```

where the syntax is the same for the other Pauli and rotation gates. Since some matrix products, as the $\hat{X}\hat{Y} \to i\hat{Z}$ in figure (5.4), result in a different factor than the previous gates, all gate classes have a factor included. For a set of gates acting on a single qubit, all the factors from the gates can be replaced with a factor for the qubit. The qubits in a circuit exist in a tensor product and so the factor of all qubits can be replaced by a factor for the circuit. This can be extracted with the `QuantumCircuit.defactor()` function.

5.3.2   Hamiltonian transformation

As discussed in section (3.5) we can map the Hamiltonian operator to a quantum circuit using the Jordan-Wigner transformation. Doing this by hand can be a tedious process, and so we have developed a general solver that inputs the one-body and two-body matrix elements and constructs a circuit list. The circuit list includes all necessary

information about the gates so that the VQE and QPE classes can implement the circuit into qiskit. We will elaborate more on this later when we discuss the implementation of these algorithms.

Next we need to transform the general second quantized Hamiltonian for a fermionic system, written on the form

$$\hat{H} = \sum_{pq} \langle p|\hat{h}|q \rangle \, \hat{a}_p^\dagger \hat{a}_q + \frac{1}{4} \sum_{\substack{p<q \\ r<s}} \langle pq|\hat{v}|rs \rangle \, \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r, \tag{5.9}$$

where we take advantage of the anti-symmetry of the two-body matrix elements

$$\langle pq|\hat{v}|rs \rangle = -\langle pq|\hat{v}|sr \rangle = -\langle qp|\hat{v}|rs \rangle = \langle qp|\hat{v}|sr \rangle.$$

In section (3.5) we discussed the Jordan-Wigner transformation of second quantized operators. If we let the $|1\rangle$ state of a qubit denote an occupied orbital, the Jordan-Wigner transformation transform all creation and annihilation operators into either $\sigma_+ = (\hat{X} - i\hat{Y})/2$ and $\sigma_- = (\hat{X} + i\hat{Y})/2$, along with $\hat{Z}$ gates to ensure the antisymmetry. An n-body second quantized operator will then give maximum 2^{2n} circuits, see appendix (B.1), as can be seen with the one-body operator $\hat{a}_p^\dagger \hat{a}_q$

$$\hat{a}_p^\dagger \hat{a}_q \longrightarrow \frac{1}{4} \left(\hat{X}_p \hat{X}_q + \hat{Y}_p \hat{Y}_q + i\hat{X}_p \hat{Y}_q - i\hat{Y}_p \hat{X}_q \right) \left[\prod_{k=p}^{q} \hat{Z}_k \right], \tag{5.10}$$

and the two-body operator $\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r$

$$
\begin{aligned}
\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r \longrightarrow \frac{1}{16} \Big(& \hat{X}_p \hat{X}_q \hat{X}_s \hat{X}_r + \hat{Y}_p \hat{Y}_q \hat{Y}_s \hat{Y}_r - \hat{X}_p \hat{X}_q \hat{Y}_s \hat{Y}_r - \hat{Y}_p \hat{Y}_q \hat{X}_s \hat{X}_r \\
& + i\hat{X}_p \hat{X}_q \hat{X}_s \hat{Y}_r + i\hat{X}_p \hat{X}_q \hat{Y}_s \hat{X}_r - i\hat{X}_p \hat{Y}_q \hat{X}_s \hat{X}_r - i\hat{Y}_p \hat{X}_q \hat{X}_s \hat{X}_r \\
& + \hat{Y}_p \hat{X}_q \hat{X}_s \hat{Y}_r + \hat{X}_p \hat{Y}_q \hat{Y}_s \hat{X}_r + \hat{X}_p \hat{Y}_q \hat{X}_s \hat{Y}_r + \hat{Y}_p \hat{X}_q \hat{Y}_s \hat{X}_r \\
& + i\hat{X}_p \hat{Y}_q \hat{Y}_s \hat{Y}_r - i\hat{Y}_p \hat{Y}_q \hat{Y}_s \hat{X}_r - i\hat{Y}_p \hat{Y}_q \hat{X}_s \hat{Y}_r + i\hat{Y}_p \hat{X}_q \hat{Y}_s \hat{Y}_r \Big) \\
& \times \left[\prod_{k=p}^{q} \hat{Z}_k \right] \left[\prod_{k=r}^{s} \hat{Z}_k \right],
\end{aligned}
\tag{5.11}
$$

for no equal indices p, q, r and s. For a general set of indices, we get more complicated ciruits, so we have implemented a `SecondQuantizedHamiltonian` class to solve the Jordan-Wigner transformation. For the general case where we do not know much about the interaction integrals the `SecondQuantizedHamiltonian` class will calculate the circuit. Here the sums in equation (5.9) become

```
# l -> Number of orbitals
# h and v -> one- and two-body integrals

circuits = [] # Resulting circuits
for p in range(l):
    for q in range(l):
        qc = QuantumCircuit(l)
        circuits += qc.insert_one_body_operator(h[p,q],p,q)
for p in range(l):
    for q in range(l):
        for r in range(l):
```

```
for s in range(l):
    qc = QuantumCircuit(l)
    circuits += qc.insert_two_body_operator(v[p,q,s
                                            ,r],p,q,r,s)
```

The `insert_one_body_operator` and `insert_two_body_operator` functions insert raising and lowering operators along with $\hat{Z}$ gates to the labeled qubits. These are then transformed into 2^n circuits with the n raising and lowering operators transformed into $\hat{X}$ and $\hat{Y}$ gates. The circuit list can easily be extracted for n and l orbitals from the one-body and two-body matrix elements,

```
H = SecondQuantizedHamiltonian(n,l,one_body, two_body)
circuit_list = H.circuit_list('vqe')
```

where 'vqe' can be replaced with 'qpe' in order to get the exponential circuit.

5.3.3   Exploiting symmetries

For systems like the paring model, we can reduce the number of circuits by exploiting some symmetries. First we know that the one-body matrix elements will be diagonal, as with all canonical Hartree-Fock orbitals, so we only need to transform the number operators $\hat{a}_p^\dagger \hat{a}_p$

$$\hat{a}_p^\dagger \hat{a}_p \longrightarrow \frac{1}{2}\left(\hat{\mathbb{1}}_p - \hat{Z}_p\right). \tag{5.12}$$

Since all measurements of the identity operator will yield the same result, we can disregard it as a circuit and add it to the final result. This leaves us with a single circuit per one-body operator.

For the two-body operator $\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r$ in the pairing model we can not break any pairs. All non-zero matrix elements correspond to excitations of particle pairs in the same spatial orbital with opposite spin. We introduced the pair creation and annihilation operators $\hat{P}^+$ and $\hat{P}^-$ which sums over all spatial orbitals

$$\sum_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r \equiv \sum_{pq} \hat{a}_{p+}^\dagger \hat{a}_{p-}^\dagger \hat{a}_{q-} \hat{a}_{q+} = \sum_{pq} \hat{P}_p^+ \hat{P}_q^-, \tag{5.13}$$

where the orbitals are ordered so that $p_- = p_+ + 1$. All $\hat{Z}$ operators in the JW transformation are then canceled out, except on p_+. Since we have $\sigma_+ \hat{Z} = \sigma_+$ and $\hat{Z}\sigma_- = \sigma_-$ we can disregard the $\hat{Z}$ operators in the JW transformation. The transformed pair creation and annihilation operators are then given by

$$\hat{P}_p^+ \rightarrow \sigma_+^{p+} \sigma_+^{p-}, \tag{5.14}$$

$$\hat{P}_q^- \rightarrow \sigma_-^{q-} \sigma_-^{q+}. \tag{5.15}$$

For the two-body operators we have the special case where $p = q$

$$\hat{P}_p^+ \hat{P}_p^- \longrightarrow \frac{1}{4}\left(\hat{\mathbb{1}}_{p+}\hat{\mathbb{1}}_{p-} - \hat{\mathbb{1}}_{p+}\hat{Z}_{p-} - \hat{Z}_{p+}\hat{\mathbb{1}}_{p-} + \hat{Z}_{p+}\hat{Z}_{p-}\right). \tag{5.16}$$

For all other indices $p \neq q$ we recognize that $\hat{P}_p^+ \hat{P}_q^- = (\hat{P}_q^+ \hat{P}_p^-)^\dagger$. Suppose $|\Phi\rangle$ is a set of qubits then

$$\left(\hat{P}_p^+ \hat{P}_q^- + \hat{P}_q^+ \hat{P}_p^-\right)|\Phi\rangle = 2\,\mathrm{Re}\left(\hat{P}_p^+ \hat{P}_q^- |\Phi\rangle\right), \tag{5.17}$$

since $z + z^\dagger = (\text{Re}(z) + \text{Im}(z)) + (\text{Re}(z) - \text{Im}(z)) = 2\,\text{Re}(z)$. We can then restrict the sum to $p < q$ giving the transformation

$$
\hat{P}_p^+ \hat{P}_q^- \to \frac{1}{8}\Big(\mathscr{P}_{xxxx}^{p_+p_-q_+q_-} + \mathscr{P}_{yyyy}^{p_+p_-q_+q_-} - \mathscr{P}_{xxyy}^{p_+p_-q_+q_-} - \mathscr{P}_{yyxx}^{p_+p_-q_+q_-} \\
+ \mathscr{P}_{xyyx}^{p_+p_-q_+q_-} + \mathscr{P}_{yxxy}^{p_+p_-q_+q_-} + \mathscr{P}_{xyxy}^{p_+p_-q_+q_-} + \mathscr{P}_{yxyx}^{p_+p_-q_+q_-} \Big).
\tag{5.18}
$$

The two-body sum now scales as $\mathcal{O}(n^2)$, in comparison to the naive implementation as $\mathcal{O}(n^4)$. This more efficient implementation of the pairing model is included in the `PairingHamiltonian` class.

5.3.4   Exponential operators

Now that we have the transformed Hamiltonian circuits, we need an efficient way to turn them into the exponential operator $\exp(-i\hat{H}t)$. Using the Trotter approximation the exponential operator can be written as a product of all the terms from equation (5.12), (5.16) and (5.18)

$$
e^{-\frac{i}{\hbar}\hat{H}t} \approx \prod_k e^{-\frac{i}{\hbar}\cdot\mathscr{P}_k t},
\tag{5.19}
$$

where $\mathscr{P}_k$ are Pauli strings. In section (4.2) we showed that all Pauli operators can be written in terms of a unitary transformation of $\hat{Z}$, and that the exponent of a Pauli string can be decomposed into

$$
e^{\mathscr{P}_k} = \hat{U}_k \big(\hat{\mathbb{1}} \otimes \cdots \otimes \hat{\mathbb{1}} \otimes \hat{R}_z(\theta) \big) \hat{U}_k^\dagger,
\tag{5.20}
$$

where $\hat{U}_k$ is the basis transformation of all Pauli gates into $\hat{Z}$-basis along with CNOT gates from all qubits with the last qubit as target. This way the parity $p_k = \pm 1$ of the qubits is stored in the last qubit, and the $\hat{R}_z$ gate adds the exponential phase e^{-itp_k}.

If there are any qubit-wise equal operators in $\hat{U}_k$ and $\hat{U}_{k+1}^\dagger$, then these can be removed, and in turn some CNOT gates might be removed. As an example we see that in the figure (5.5) two Hadamard gates and two CNOT gates can be removed, giving the reduced circuit in figure (5.6). We thus seek to find an order in which a maximum
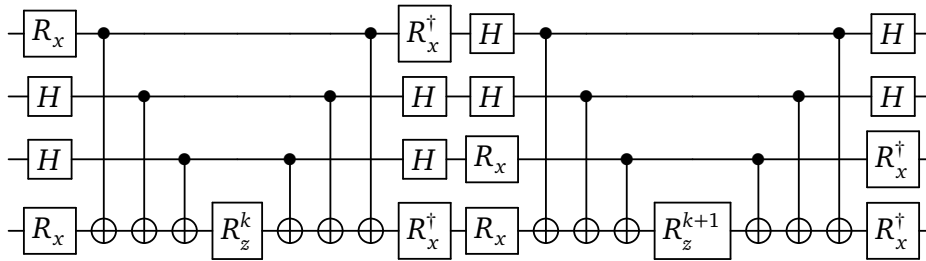

Figure 5.5: Quantum circuit of the exponential operators $\mathscr{P}_{yxxy}$ with basis transformation $\hat{U}_k$ and $\mathscr{P}_{xxyy}$ with basis transformation $\hat{U}_{k+1}$.

number of gates are removed. Looking at the possible Pauli strings in equation (5.18) we can order the terms so that at least two operators and two CNOT gates are removed between each Pauli string. We also note that having all control operations on
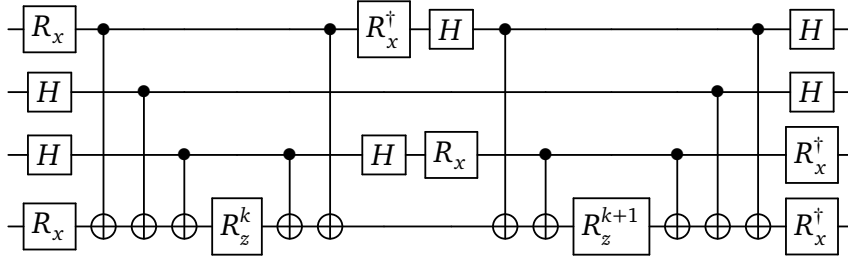
Figure 5.6: Reduced quantum circuit of the exponential operators $\mathscr{P}_{yxxy}$ with basis transformation $\hat{U}_k$ and $\mathscr{P}_{xxyy}$ with basis transformation $\hat{U}_{k+1}$.

a single qubit is the most efficient implementation considering the coupling map in figure (5.2). Then we can map th qubit to the one with connectivity to the remaining qubits, without having to swap qubits in order to use controlled operations.

In the `Hamiltonian` class the circuit is turned into an exponential operator by setting `exp=True` in the constructor. Each term is transformed into an exponential operator using the `QuantumCircuit` class function `to_exponent()` and then appended to the final circuit. The `to_exponent()` makes use of another function `insert_pauli_string(exp=True)`, where the operators in the pauli string is replace by its unitary transformation along with CNOT gates. This final circuit is then reduced by removing identical gate actions. The following code will implement the Pauli strings $\mathscr{P}_{yxxy}$ and $\mathscr{P}_{xxyy}$ from figure (5.5) and reduce it to figure (5.6).

```
from quantum_circuit import QuantumCircuit

qc = QuantumCircuit(4)
qc.insert_pauli_String(['YXXY',[0,1,2,3],factor=1],exp=True)
qc.insert_pauli_String(['XXYY',[0,1,2,3],factor=1],exp=True)
qc.gate_optimization()
```

5.4   Phase estimation

The quantum phase estimation algorithm is implemented as the QPE class in the `QuantumAlgorithms` module. It inputs an instance of the `Hamiltonian` class, an ansatz function for the simulation register, the number of work qubits and an upper limit to the energy estimate E_{max}. Additional input arguments include the number of Trotter steps, the step length and options for the execution of the circuit. It includes the inverse Fourier transform with optional qubit swapping at the end, as this is not strictly necessary.

5.4.1   State preparation

In the phase estimation we need to ensure that the simulation register is in an eigenstate of the Hamiltonian. For the pairing model no pairs can be broken and we initialize the eigenstate as the Bell state

$$\longrightarrow \quad \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right).$$

If we set up this Bell state for all pairs, then the qubits will be in an equal superposition of all possible configurations of pairs. The result of the QPE algorithm will then yield the energies for all these configurations.

5.4.2   Efficient implementation of controlled operations

In section (5.3.4) we discussed how the exponential operator is made into a circuit list. This circuit is then implemented to the simulation register in the phase estimation algorithm as a controlled circuit 2^j times for each work qubit j. We want to reduce the number of controlled operation, as they are more difficult to perform than singel qubit operations. Since the individual operations in $e^{\frac{i}{\hbar}\hat{H}_k t}$ from equation (5.19) are symmetric except for the $\hat{R}_z$ operations, we only need one controlled operation for each exponent operator. We can see this from the following circuit with $R_x = \hat{R}_x(-\pi/2)$
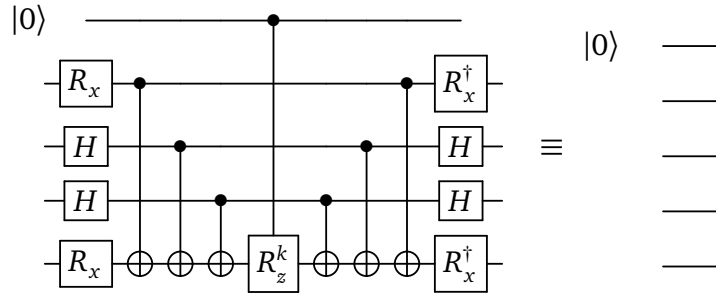

Figure 5.7: Controlled operation of an exponential Pauli string when the control qubit is in the $|0\rangle$ state is equivalent to an empty circuit. Here $\hat{R}_x = \hat{R}_x(-\pi/2)$.

since the control qubit is in the $|0\rangle$ state the $\hat{R}_z$ is not applied and all other gates get contracted. In addition, the target gate $\hat{R}_z(\theta)$ can take any parameter θ and only adds a phase to the state. The matrix representation of the R_z gate is

$$\hat{R}_z(\theta) = \begin{bmatrix} e^{-\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}, \tag{5.21}$$

up to an unimportant global phase. For each work qubit j we want to end up in the state

$$|0\rangle + e^{2\pi i 2^j \theta} |1\rangle. \tag{5.22}$$

Normally we would have to apply the circuit 2^j times to achieve this. If $e^{\hat{U}}$ is an operator whose eigenstate is $|\psi\rangle$ with eigenvalue $e^{2\pi i \theta}$ then, the 2^n controlled actions on $|\psi\rangle$ conditional on a work qubit $|w\rangle$ gives

$$\left(e^{\hat{U}}\right)^{2^n} |w\rangle |\psi\rangle = \left(|0\rangle + e^{2\pi i 2^n \theta} |1\rangle\right) |\psi\rangle. \tag{5.23}$$

However if $\hat{U}$ could be decomposed to a $\hat{Z}$ gate then the exponential operator is equivalent to a controlled $\hat{R}_z(2\pi\theta)$ operation. This means we can replace equation (5.23) by

$$\hat{R}_z\left(2^{n+1}\pi\theta\right) |w\rangle |\psi\rangle = \left(|0\rangle + e^{2\pi i 2^n \theta} |1\rangle\right) |\psi\rangle. \tag{5.24}$$

So instead of acting with the whole circuit 2^j times, we multiply θ with 2^j to replace the multiple operations.

Table 5.1: FCI energies for all configurations of the pairing model with four orbitals.

N_{pairs}	E_{FCI}
0	0
1	-0.618, 1.618
2	1

5.4.3   Trotter steps

From the Trotter truncation given by equation (4.44) we write the unitary operator

$$\hat{U} = \left(\prod_k e^{-i\hat{H}_k \Delta t} \right)^{\rho}, \tag{5.25}$$

where increasing the number of Trotters steps ρ gives a better approximations to the exact time evolution operator. Figure (5.8) shows three simulations of the pairing model with four simulation qubits and six work qubits. There are four possible energies corresponding to different states for this simulation register shown in the table below.
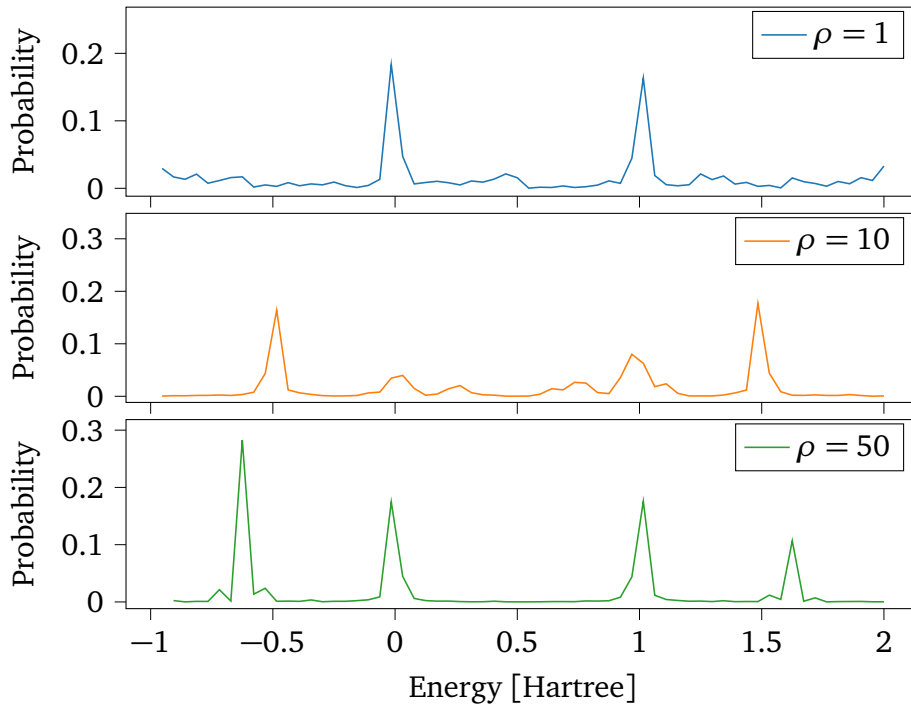

Figure 5.8: Probability distribution of the energy spectrum for the pairing model using QPE with four simulation qubits and six work qubits. Plotted for different numbers of Trotter steps ρ. QPE needs a given number of Trotter steps in order to yield the correct eigenvalues, given in table (5.1).

It is apparent that a single Trotter step is not sufficient and gives the wrong energies, while 10 and 100 Trotter steps give energy peaks close to the exact values. Increasing ρ, after a given value for which we get the correct energy peaks, will narrow the peaks and by that reduce the variance of the results.

5.4.4   Number of work qubits

The finite number of work qubits t used in the QPE leads to the error

$$\delta = \phi + \frac{b}{2^t}, \tag{5.26}$$

where ϕ is the true phase and b is an integer giving the best approximation to ϕ in terms of the binary fraction

$$\frac{b}{2^t} = 0.b_1 \ldots b_t. \tag{5.27}$$

Since $0 \leq b < 2^t$, we have that $0 \leq \delta \leq 2^{-t}$. An accurate estimation of ϕ requires the QPE to produce a result m close to b with high probability. Suppose we want an accuracy of 2^{-n} to our estimate of ϕ, if we choose $e = 2^{t-n} - 1$ then the probability of failure can be defined as [11]

$$p(|m - b| > e) \leq \frac{1}{2(e-1)}. \tag{5.28}$$

To achieve this accuracy we must choose a number of work qubits

$$t = n + \left\lceil \log_2 \left(2 + \frac{1}{2\epsilon} \right) \right\rceil, \tag{5.29}$$

for a probability of success at least $1 - \epsilon$. In figure (5.9) we show how the energy spikes corresponding to a possible eigenvalue becomes narrower for an increasing number of qubits.
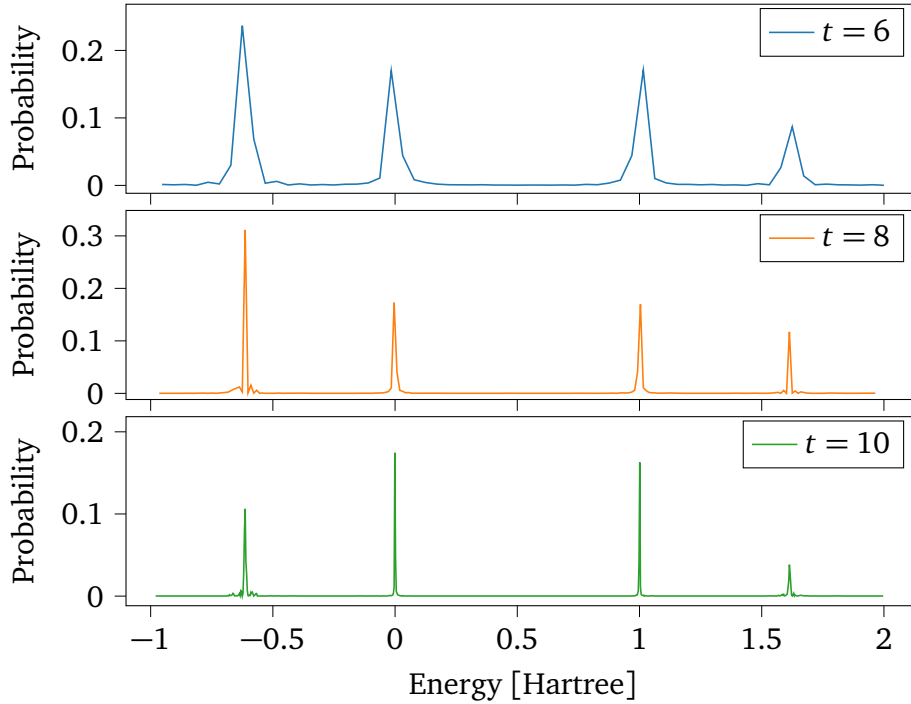
Figure 5.9: Probability distribution of the energy spectrum for the pairing model using QPE with four simulation qubits and $\rho = 125$ Trotter steps. Plotted for different numbers of work qubits t. As the number of work qubits increase, the spikes become narrower, thus decreasing the variance.

5.5 Variational quantum eigensolver

The main components of the VQE algorithm is the state ansätz with parameter optimization, and the measurements. For this purpose we have implemented a set ansätz classes, including Unitary Coupled Cluster, along with several classes for the parameter optimization. We have shown that a second quantized Hamiltonian can be transformed into quantum circuits composed of Pauli strings. To simplify the measurements of the Pauli gates, we have implemented a `PauliString` class that performs the basis transformation into the eigenbasis of a single Pauli string. After a measurement, this also inputs the statistics from which it outputs the expectation value. In addition we have implemented two classes `QWCGroup` and `GCGroup` for commuting Pauli strings that performs the basis transformation into the simultanous eigenstates of a commuting group. These are included in a `CircuitList` class which the VQE algorithm iterates through to calculate the expectation value of the Hamiltonian.

5.5.1 Pauli measurements

In section (3.6) we defined the Pauli string as a tensor product of Pauli operators acting on individual qubits. When we measure the expectation of the Hamiltonian we need to measure each resulting Pauli string from the Jordan-Wigner transformation

$$\langle E \rangle_\theta = \sum_k a_k \langle \Psi(\theta) | \mathscr{P}_k | \Psi(\theta) \rangle, \tag{5.30}$$

66

where we require $|\Psi(\theta)\rangle$ to be in an eigenstate of $\mathscr{P}_k$ in order to yield

$$\langle \Psi(\theta)| \mathscr{P}_k |\Psi(\theta)\rangle = p_k \langle \Psi(\theta)|\Psi(\theta)\rangle = p_k. \tag{5.31}$$

A quantum computer will naively measure this in the computational basis. Unfortunately these are only eigenvectors for a Pauli string with $\hat{Z}$ operations exclusively. For a general Pauli string we need to make a change of basis, so that the state $|\Psi\rangle$ becomes an eigenvector of the Pauli string.

Pauli-X basis

Consider a single qubit for which we want to measure the eigenvalue of the $\hat{X}$ operation. The computational basis yields

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{5.32}$$

If we apply a Hadamard-gate on these states we get the basis states of $\hat{X}$

$$\hat{H}|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \hat{H}|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{5.33}$$

By applying a Hadamard gate before $\hat{X}$ we can thus measure the eigenvalues of $\hat{X}$, resulting in $+1$ if we measure the state to be $|0\rangle$ and -1 if we measure $|1\rangle$.

Pauli-Y basis

Next we want to do the same for a $\hat{Y}$ operation. We can make two different transformations of the computational basis; either a Hadamard gate followed by a phase gate

$$\hat{S}\hat{H}|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ i \end{bmatrix}, \quad \hat{S}\hat{H}|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -i \end{bmatrix}, \tag{5.34}$$

or $\hat{R}_x(-\frac{\pi}{2})$

$$\hat{R}_x\left(-\frac{\pi}{2}\right)|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ i \end{bmatrix}, \quad \hat{R}_x\left(-\frac{\pi}{2}\right)|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} i \\ 1 \end{bmatrix}, \tag{5.35}$$

where both sets are eigenvectors of $\hat{Y}$.

Pauli string basis

If we consider multiple qubits and want to measure the eigenvalue of a Pauli string, we see that the transformation operator is the tensor product of the Pauli gate transformations above. The eigenvalues will only depend on the parity of the qubits. Consider the Pauli string

$$\mathscr{P}_{xy} = \hat{X} \otimes \hat{Y} = \begin{bmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{bmatrix}, \tag{5.36}$$

and the transformation operator as the tensor product of each Pauli operator in the Pauli string

$$\hat{T} = \hat{H} \otimes \hat{R}_x\left(-\frac{\pi}{2}\right) = \frac{1}{2}\begin{bmatrix} 1 & i & 1 & i \\ i & 1 & i & 1 \\ 1 & i & -1 & -i \\ i & 1 & -i & -1 \end{bmatrix}, \tag{5.37}$$

Acting on the computational basis with the transformation operator yields

$$\hat{T}\ket{00} = \frac{1}{2}\begin{bmatrix} 1 \\ i \\ 1 \\ i \end{bmatrix}, \quad \hat{T}\ket{01} = \frac{1}{2}\begin{bmatrix} i \\ 1 \\ i \\ 1 \end{bmatrix}, \quad \hat{T}\ket{10} = \frac{1}{2}\begin{bmatrix} 1 \\ i \\ -1 \\ -i \end{bmatrix}, \quad \hat{T}\ket{11} = \frac{1}{2}\begin{bmatrix} i \\ 1 \\ -i \\ -1 \end{bmatrix}, \tag{5.38}$$

being eigenvectors of the Pauli string with

$$\mathscr{P}_{xy}\hat{T}\ket{00} = \hat{T}\ket{00}, \quad \mathscr{P}_{xy}\hat{T}\ket{01} = -\hat{T}\ket{01}, \quad \mathscr{P}_{xy}\hat{T}\ket{10} = -\hat{T}\ket{10}, \quad \mathscr{P}_{xy}\hat{T}\ket{11} = \hat{T}\ket{11}. \tag{5.39}$$

Thus the eigenvalues of $\mathscr{P}_{xy}$ are only dependent on the parity of the qubit. This means we do not have to act with the Pauli string on the qubits, only the transformation operator $\hat{T}$ before we measure the qubits. It is straight forward to show that this holds for all Pauli strings. We note however, that for a general qubit state

$$\ket{\phi_1 \ldots \phi_n},$$

and a Pauli string including identity operators

$$\mathscr{P}_{xzx}^{ijk} = \hat{\mathbb{1}}_1 \otimes \cdots \otimes \hat{\mathbb{1}}_{i-1} \otimes \hat{X}_i \otimes \hat{\mathbb{1}}_{i+1} \otimes \cdots \otimes \hat{\mathbb{1}}_{j-1} \otimes \hat{Z}_j \otimes \hat{\mathbb{1}}_{j+1} \otimes \cdots \otimes \hat{\mathbb{1}}_{k-1} \otimes \hat{X}_k \otimes \hat{\mathbb{1}}_{k+1} \otimes \cdots \otimes \hat{\mathbb{1}}_n,$$

with $i < j < k < n$, we only care for the parity of qubits i, j and k as these are the only ones affecting the eigenvalue of the Pauli string.

5.5.2   Circuit construction of simultaneous eigenstates

Recognizing commuting operators is very useful, since it can greatly reduce the number of state preparations needed to calculate the expectation value of the Hamiltonian. A group of commuting operators have a set of simultaneous eigenstates, and can thus be measured simultaneously. In section (3.6) we discussed two different types of commuting operators; qubit-wise commuting (QWC) and general commuting (GC) operators. As we will see, finding the transformation to the eigenbasis of a group of QWC operators is much easier than for a group of GC operators.

Qubit-wise commuting operators

For a group of QWC operators we make use of the realization from the last section where we showed that the eigenvalues of a Pauli string with more than one Pauli gate is only dependent on the pairity of the qubits it acts on. Since all qubit-wise gates in the Pauli strings commute they either have the same gate on the same qubit, or they do not act on the same qubit. As a result we only transform the qubits where the Pauli strings share a gate action. For the pairing Hamiltonian we have a set of QWC

commuting operators as all Pauli strings with only $\hat{Z}$ gates, where no transformation is needed.

We have implemented a `QWCGroup` class where `PauliString` instances can be appended. It finds the qubits to transform and perform the transformation in the VQE simulation with the `QWCGroup.prepare()` function.

General commuting operators

Following Gokhale et al [14] we make use of stabilizer matrices, a concept connected to the development of quantum error-correcting codes. Quantum error-correction is outside the scope of this thesis, so we will not outline the full theory of this. However, stabilizer matrices can be used to construct circuits for simultaneous eigenstates of Pauli strings. For an N qubit Pauli string there is associated a $2N$-entry column vector where the Pauli operations are represented as follows:

- A $\hat{Z}$ operation on the ith qubit is represented by a 1 in the ith entry.

- A $\hat{X}$ operation on the ith qubit is represented by a 1 in the $(N+i)$th entry.

- A $\hat{Y}$ operation on the ith qubit is represented by a 1 in the ith and the $(N+i)$th entry.

All entries are 0 otherwise. As an example we have the column vector associated with the Pauli string $\mathscr{P}_{xyz}$

$$\begin{pmatrix} 0 \\ 1 \\ 1 \\ \hline 1 \\ 1 \\ 0 \end{pmatrix}. \tag{5.40}$$

A set of M Pauli strings is then associated with an $2N \times M$ matrix, being the stabilizer matrix. The computational basis stabilizer matrix $\mathscr{S}_C$ is the matrix associated with the set

$$\{\hat{Z}\hat{\mathbb{1}}\dots\hat{\mathbb{1}}, \hat{\mathbb{1}}\hat{Z}\dots\hat{\mathbb{1}}, \dots, \hat{\mathbb{1}}\hat{\mathbb{1}}\dots\hat{Z}\}.$$

For a set of Pauli strings, with stabilizer matrix $\mathscr{S}_P$, we can construct the circuit for the simultaneous eigenstates by finding the operators that transforms $\mathscr{S}_P$ into $\mathscr{S}_C$. To do this we have the following rules:

- **Hadamard gate on the ith qubit.**
  Swaps the ith and $(N+i)$th row of the matrix.

- **Phase gate ($\hat{S}$) on the ith qubit.**
  Sets the (i, i) entry of the matrix to 0.

- **Controlled $\hat{Z}$ gate with control on the ith qubit and target on the jth qubit.**
  Sets the (i, j) and (j, i) entries to 0.

- **CNOT gate with control on the jth qubit and target on the ith qubit.**
  With modulo 2 addition. Adds the ith row to the jth row, and the $(N+i)$th row to the $(N+j)$th row of the matrix.

- **SWAP gate between the ith qubit and jth qubit.**
  Swaps the ith row and jth row, and the $(N+i)$th and $(N+j)$th row of the matrix.

As an example we show in appendix (B.4) how to make such a circuit for the transformed two-body operators of the pairing model in equation (5.18), given in figure (5.10). Gokhale et al. [14] finds a similar circuit, however recognizing that we can ignore SWAP gates our circuit has 5 CNOT gates less. We can reduce this circuit
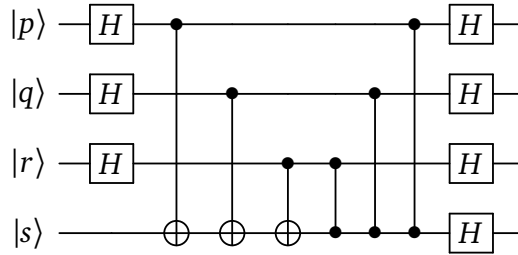

Figure 5.10: Quantum circuit for transforming the computational basis into the simultaneous eigenstates of the Pauli strings in equation (5.18).

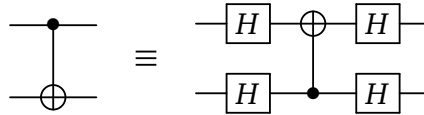even further by using the circuit identities for a CNOT gate and a CZ gate

Figure 5.11: CNOT gate identity using Hadamard gates.
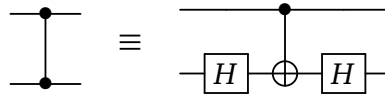

Figure 5.12: CZ identity using Hadmard gates and a CNOT gate.

Replacing all CNOT and CZ gates by these we end up with the reduced circuit shown in figure (5.13). From the construction of the circuit we made use of the four Pauli strings

$$\mathscr{P}_{xxxx}^{p+p-q+q-}, \quad \mathscr{P}_{xxyy}^{p+p-q+q-}, \quad \mathscr{P}_{xyxy}^{p+p-q+q-}, \quad \mathscr{P}_{yxxy}^{p+p-q+q-}.$$
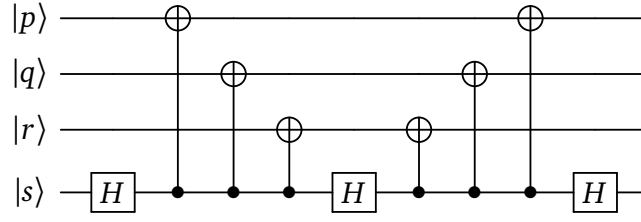
Figure 5.13: Reduced quantum circuit for transforming the computational basis into the simultaneous eigenstates of the Pauli strings in equation (5.18).

After measuring the circuit the eigenvalues of these Pauli strings are determined from the value of each qubit, that is

$$\left\langle \mathscr{P}_{xxxx}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if } |q_-\rangle = 0, \\ -1 & \text{if } |q_-\rangle = 1, \end{cases} \tag{5.41}$$

$$\left\langle \mathscr{P}_{xxyy}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if } |q_+\rangle = 0, \\ -1 & \text{if } |q_+\rangle = 1, \end{cases} \tag{5.42}$$

$$\left\langle \mathscr{P}_{xyxy}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if } |p_-\rangle = 0, \\ -1 & \text{if } |p_-\rangle = 1, \end{cases} \tag{5.43}$$

$$\left\langle \mathscr{P}_{yxxy}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if } |p_+\rangle = 0, \\ -1 & \text{if } |p_+\rangle = 1, \end{cases} \tag{5.44}$$

where N is the number of measurements. To find the eigenvalues of the remaining terms we first find the matrix representation of both the circuit and each Pauli string through tensor and matrix products. The eigenvalues can then be found by multiplying the Pauli string matrix with each column of the circuit matrix representing the transformation of each computational basis state. Table (5.2) shows the eigenvalues of all the Pauli strings. Here we see that for a set of N measurements the expectation value of the remaining Pauli strings are given as

$$\left\langle \mathscr{P}_{yyyy}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if parity of } |p_+p_-q_+\rangle = \text{odd}, \\ -1 & \text{if parity of } |p_+p_-q_+\rangle = \text{even}, \end{cases} \tag{5.45}$$

$$\left\langle \mathscr{P}_{yyxx}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if parity of } |p_+p_-q_-\rangle = \text{odd}, \\ -1 & \text{if parity of } |p_+q_+q_-\rangle = \text{even}, \end{cases} \tag{5.46}$$

$$\left\langle \mathscr{P}_{yxyx}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if parity of } |p_+q_+q_-\rangle = \text{odd}, \\ -1 & \text{if parity of } |p_+q_+q_-\rangle = \text{even}, \end{cases} \tag{5.47}$$

$$\left\langle \mathscr{P}_{xyyx}^{p_+p_-q_+q_-} \right\rangle = \frac{1}{N} \sum^{N} \begin{cases} +1 & \text{if parity of } |p_-q_+q_-\rangle = \text{odd}, \\ -1 & \text{if parity of } |p_-q_+q_-\rangle = \text{even}, \end{cases} \tag{5.48}$$

where the parity of a state $|abc\rangle$ is given by the sum $a + b + c$.

We have implemented a `GCGroup` class where `PauliString` instances can be appended. The circuit to prepare the simultaneous eigenstates must be implemented as an ansatz function `f()`, and set to the class instance with `GCGroup.set_ansatz(f)`. In the VQE simulation `f()` is called with the `GCGroup.prepare()` function.

Table 5.2: Eigenvalues of the Pauli strings in equation (5.18) in their simultaneous eigenbasis. The leftmost column shows the outcome of a measurement after the transformation of the computational basis.

$\lvert p_+p_-q_+q_-\rangle$	XXXX	XXYY	XYXY	YXXY	YYYY	YYXX	YXYX	XYYX
$\lvert 0000\rangle$	+1	+1	+1	+1	-1	-1	-1	-1
$\lvert 0001\rangle$	-1	+1	+1	+1	-1	+1	+1	+1
$\lvert 0010\rangle$	+1	-1	+1	+1	+1	-1	+1	+1
$\lvert 0011\rangle$	-1	-1	+1	+1	+1	+1	-1	-1
$\lvert 0100\rangle$	+1	+1	-1	+1	+1	+1	-1	+1
$\lvert 0101\rangle$	-1	+1	-1	+1	+1	-1	+1	-1
$\lvert 0110\rangle$	+1	-1	-1	+1	-1	+1	+1	-1
$\lvert 0111\rangle$	-1	-1	-1	+1	-1	-1	-1	+1
$\lvert 1000\rangle$	+1	+1	+1	-1	+1	+1	+1	-1
$\lvert 1001\rangle$	-1	+1	+1	-1	+1	-1	-1	+1
$\lvert 1010\rangle$	+1	-1	+1	-1	-1	+1	-1	+1
$\lvert 1011\rangle$	-1	-1	+1	-1	-1	-1	+1	-1
$\lvert 1100\rangle$	+1	+1	-1	-1	-1	-1	+1	+1
$\lvert 1101\rangle$	-1	+1	-1	-1	-1	+1	-1	-1
$\lvert 1110\rangle$	+1	-1	-1	-1	+1	-1	-1	-1
$\lvert 1111\rangle$	-1	-1	-1	-1	+1	+1	+1	+1

5.5.3   Circuit list class

The Pauli strings and commuting groups are appended to an instance of the simple `CircuitList` class. This only contains as set of functions for grouping individual Pauli strings appended to it. The `find_qwc()` function searches through all Pauli strings and finds all groups of QWC operators. For the special case, like the pairing model with $n = 2$ and $l = 4$, a more efficient approach is to only group operators containing $\hat{Z}$ gates, since these are the only QWC operators. This can be done with the `groupz()` function. For the remaining Pauli strings from the two-body operators, we need to define both the measurement preparation circuit and what operators map to the different measurement outcomes. The circuit previously discussed, see figure (5.13), prepares the eigenstates of a general group of Pauli strings given by equation (5.18). That is, this circuit can be used if the Pauli strings act on the same qubits. Thus, we have included a `group_two_body()` function that groups Pauli strings in a circuit list that make up such a group. Below we show have to the group the Pauli strings in the transformed Hamiltonian for the pairing model with two particles and four orbitals, found in section (5.3.3):

```
from quantum_circuit import PairingHamiltonian

n,l = 2,4
delta, g = 1,1
H = PairingHamiltonian(n,l,delta,g)
circuit_list = H.circuit_list('vqe')
circuit_list.groupz()
circuit_list.group_two_body()
```

5.5.4 Number of measurements

The number of measurements per Pauli string or commuting group will affect the results. In figure (5.14) brute force calculations are shown near the optimal value for the pairing model with $N = 1000$ and $N = 10000$ measurements. We see that increas-
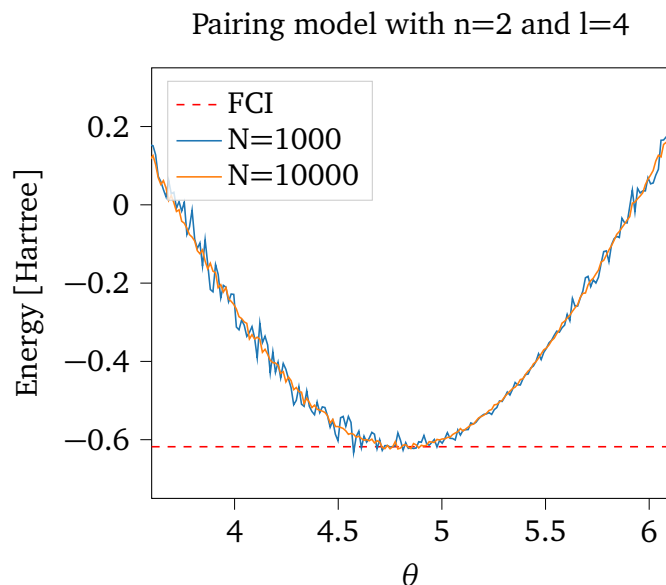

Figure 5.14: Brute force calculations of expectation value of the pairing Hamiltonian with 2 particles and 4 orbitals, with the RY ansätz (see section (5.8)) giving a single parameter θ to optimize. N is the number of measurements per commuting group.

ing the number of measurements N leads to less variance in our results, however a large number of measurements is not desired. For $N = 1000$ the energies calculated are at some points below the exact energy. This is not in violation with the variational principle, but merely a consequence of the statistical errors due to a finite set of measurements.

5.5.5 Classical optimization

Optimizing the VQE parameters with classical optimization methods are done with the `minimize` method from `scipy.optimize`. It includes a range of different methods such as Nelder-Mead [16], Cobyla [17] and Powell [18]. It inputs a scalar function being `VQE.expval()`, and outputs the optimized parameters being the UCC amplitudes. Since these methods use different names for the same inputs, we include all in a `Minimizer` class where the input is the name of the method, the maximum number of iterations or evaluations, and the convergence tolerance. In addition we will make use of the Simultaneous Perturbation Stochastic Approximation (SPSA) [19], with our own implementation. We will in short describe the characteristics of each method.

SPSA

In some sense the SPSA method [19] can be said to expect noisy measurements. It seeks to minimize a loss function $L(\theta)$ for an n-dimensional vector of parameters θ,

with some minimum $L(\theta^*)$ for an optimal set of parameters. For any value of θ we assume measurements of the loss function

$$y(\theta) = L(\theta) + \text{noise}. \tag{5.49}$$

If θ_k is the approximation to the minimum parameters θ^*, the gradient is defined as

$$g_k(\theta_k) = \frac{y(\theta_k + c_k\Delta_k) - y(\theta - c_k\Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix}, \tag{5.50}$$

where c_k is some variable assosiated with the noise, and Δ_k is a vector with Bernoulli distributed variables $\Delta_{ki} = \pm 1$. The parameters is updated recursively by

$$\theta_{k+1} = \theta_k - a_k g_k(\theta_k), \tag{5.51}$$

where a_k is a variable dependent on the step size. We define a_k and c_k at each given iteration by

$$a_k = \frac{a}{(A + k + 1)^\alpha}, \tag{5.52}$$

$$c_k = \frac{c}{(k + 1)^\gamma}, \tag{5.53}$$

where a is the step length, c is assumed to be the standard deviation of the measurement noise, A is usually set equal to 10% of the maximum number of iterations. Good values of a and c are crucial for the performance of the SPSA algorithm.

Nelder-Mead

The Nelder-Mead method is a so-called simplex method. It is popular heuristic non-derivative method and is widely used when the function values are subject to noise, however it is not guaranteed to find the optimal parameters and can get stuck in non-stationary points. Assuming n parameters, $(n + 1)$ points $P_1, \ldots, P_n$ define the simplex in an n-dimensional space. Each point is ordered in terms of its function value in order to update the worst point P_h with respect to the best point P_l and the centroid P^* of the simplex. The algorithm is terminated when all points get sufficiently close, or the number of iterations exceed some prescribed value.

   The `SciPy` implementation is based upon recent developments [16] improving the inefficiency of standard Nelder-Mead in high-dimensional space.

Powell

Powell's method [**?**] is a conjugate direction method. Thus it finds a set of conjugate vectors and minimizes the parameters along each of them separately. This way minimization along one direction does not ruin the previous minimization along another direction. It does not depend on any calculations of the gradient.

COBYLA

Constrained Optimization BY Linear Approximation (COBYLA) [17] is another derivative-free method minimizing a linear approximation of the objective functions on the vertices of a simplex. It operates within a trust region where the radius is adjusted as the method moves towards an optimal set of parameters.

5.5.6　Tolerance

Tolerance can be set to $1/N$ as the variance of measurements of the same parameters tend to be of this value.

5.6　The single-particle basis

In order to represent a quantum system on a quantum computer, we need to look at how the single-particle basis is represented. At the beginning of a simulation all qubits are assumed to be initialized to the $|0\rangle$ state. For an N-qubit register we then have the initial state

$$\bigotimes_{i=1}^{N} |0\rangle_i = |0_1 \ldots 0_N\rangle. \tag{5.54}$$

Making use of the occupation number representation from section (2.6), we have that for a system of fermions the state of each qubit can represent the occupation number of a given orbital. Thus, a system of qubits can naturally represent Slater determinants.

We add particles to equation (5.54) by flipping the respective qubit states. In section (3.2) we show that this is equivalent to applying a $\hat{X}$ gate. The Hartree-Fock state, where all p lowest energy orbitals are occupied is then given by

$$\left[\bigotimes_{k=1}^{p} \hat{X}_k\right] \bigotimes_{i=1}^{N} |0\rangle_i = \left|1_1 \ldots 1_p 0_{p+1} \ldots 0_N\right\rangle. \tag{5.55}$$

Following this we can initialize the Hartree-Fock state in `Qiskit` with the following code,

```python
from qiskit import QuantumCircuit

qc = QuantumCircuit(n_qubits)

for i in range(p):
    qc.x(i)
```

This will be done in all the ansätz classes, before implementing the ansätz circuits.

5.7　The unitary coupled cluster ansatz

To perform the UCC ansatz we have implemented a `UnitaryCoupledCluster` class that inputs the number of particles and orbitals, the truncation labels being either 'D' or 'SD', as well as the one-body and two-body matrix elements. From the matrix elements we can calculate a good initial guess for the parameters from Møller-Plesset perutrbation theory, see equation (4.72) and (4.73).

The `UnitaryCoupledCluster` class performs the implementation of the unitary coupled cluster ansatz as derived in appendix (B.2). Though we can reduce the gate count by utilizing the term ordering strategy from section (5.3.4).

5.7.1　Optimization of UCC circuits

It is desirable to miminize the gate count and circuit depth. Both in the case where we run the algorithm on an actual quantum computer and when we simulate one. For

the simulation part this is because it will lead to less matrix products and thus when we have larger numbers of qubits it will make a big time difference.

Singles terms

We start with the singles terms, where we ignore the $\hat{Z}$ gates. The Pauli strings then only act on two qubits at the time so we represent the whole circuit with this, where it obviously is repeated to complete the whole circuit.

We know that we can write the $\hat{X}$ and $\hat{Y}$ gates in the computational basis as

$$\hat{X} = \hat{H}\hat{Z}\hat{H}, \quad \hat{Y} = \hat{H}\hat{R}_x\left(\frac{\pi}{2}\right)\hat{Z}\hat{R}_x\left(-\frac{\pi}{2}\right)\hat{H}. \tag{5.56}$$

Here the H can potentially cancel each other out. We want to implement the following equation

$$\exp\left(\sum_{ia}\frac{it_i^a}{2}\left(\mathscr{P}_{yx}^{ia} - \mathscr{P}_{xy}^{ia}\right)\right) \approx \prod_{ia}\exp\left(\frac{it_i^a}{2}\mathscr{P}_{yx}^{ia}\right)\exp\left(-\frac{it_i^a}{2}\mathscr{P}_{xy}^{ia}\right). \tag{5.57}$$

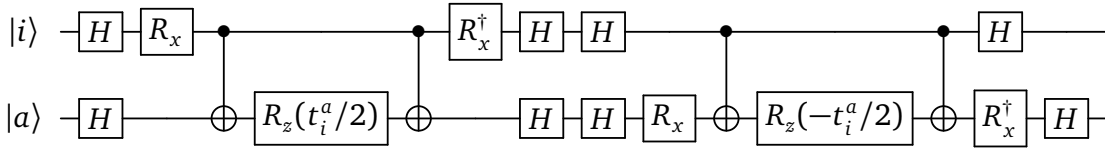Each term can be represented in a circuit for qubit i and a as in figure (5.15).

Figure 5.15: Unoptimized circuit for UCC singles term.

This circuit includes 14 single qubit gates, 4 CNOT gates and a circuit depth of 14. We see that we can remove four H gates reducing the single qubit count to 10 and the circuit depth to 11. The circuit is then as given by figure (5.16). Though this is not a significant reduction, this simple strategy will prove more useful for the doubles terms.
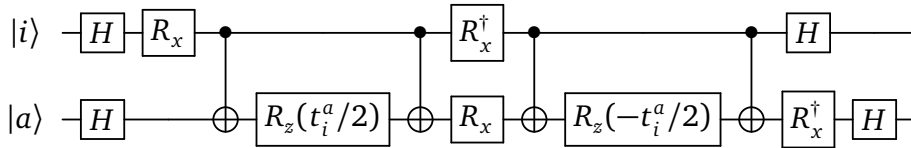

Figure 5.16: Optimized circuit for UCC singles term.

Doubles terms

The doubles terms can be rewritten in a similar manner as the singles terms using the Trotter truncation and ignoring the $\hat{Z}$ gates. In addition we make some smart reordering of the terms. We see that all the Pauli string in equation (B.18) share at least two operations on the same qubits. As an example $\mathscr{P}_{xxxy}^{ijab}$ and $\mathscr{P}_{xxyx}^{ijab}$ both act with $\hat{X}$ on qubit i and j. If these Pauli strings are ordered next to each other the gates between them will cancel each other. For the qubits where different Pauli gates act, we will get two transformation gates in between, we can call this a gate change. However, since all Pauli strings share two gates with at least two other Pauli strings

we can clearly get rid of a lot of $\hat{H}$ gates. We choose to store the parity of the qubits in the qubit that has a minimal number of gate changes after reordering the Pauli strings. Potentially this can lead to reduction of CNOT gates as well. We choose an order of the Pauli strings with only one gate change on the target qubit and a total gate change of 14:

$$
\begin{aligned}
&\mathscr{P}^{ijab}_{xxyx}, \\
&\mathscr{P}^{ijab}_{xyxx}, \\
&\mathscr{P}^{ijab}_{yxxx}, \\
&\mathscr{P}^{ijab}_{yyyx}, \\
&\mathscr{P}^{ijab}_{xyyy}, \\
&\mathscr{P}^{ijab}_{yxyy}, \\
&\mathscr{P}^{ijab}_{yyxy}, \\
&\mathscr{P}^{ijab}_{xxxy}.
\end{aligned}
\tag{5.58}
$$

This lead to a significant gate reduction, with the gate counts and circuit depths summarized in table (5.3). The final circuit can be seen in figure (5.17).

Table 5.3: Circuit summary for UCC doubles.

	Single gates	CNOTs	Circuit depth
Unoptimized	105	48	88
Optimized	32	36	55
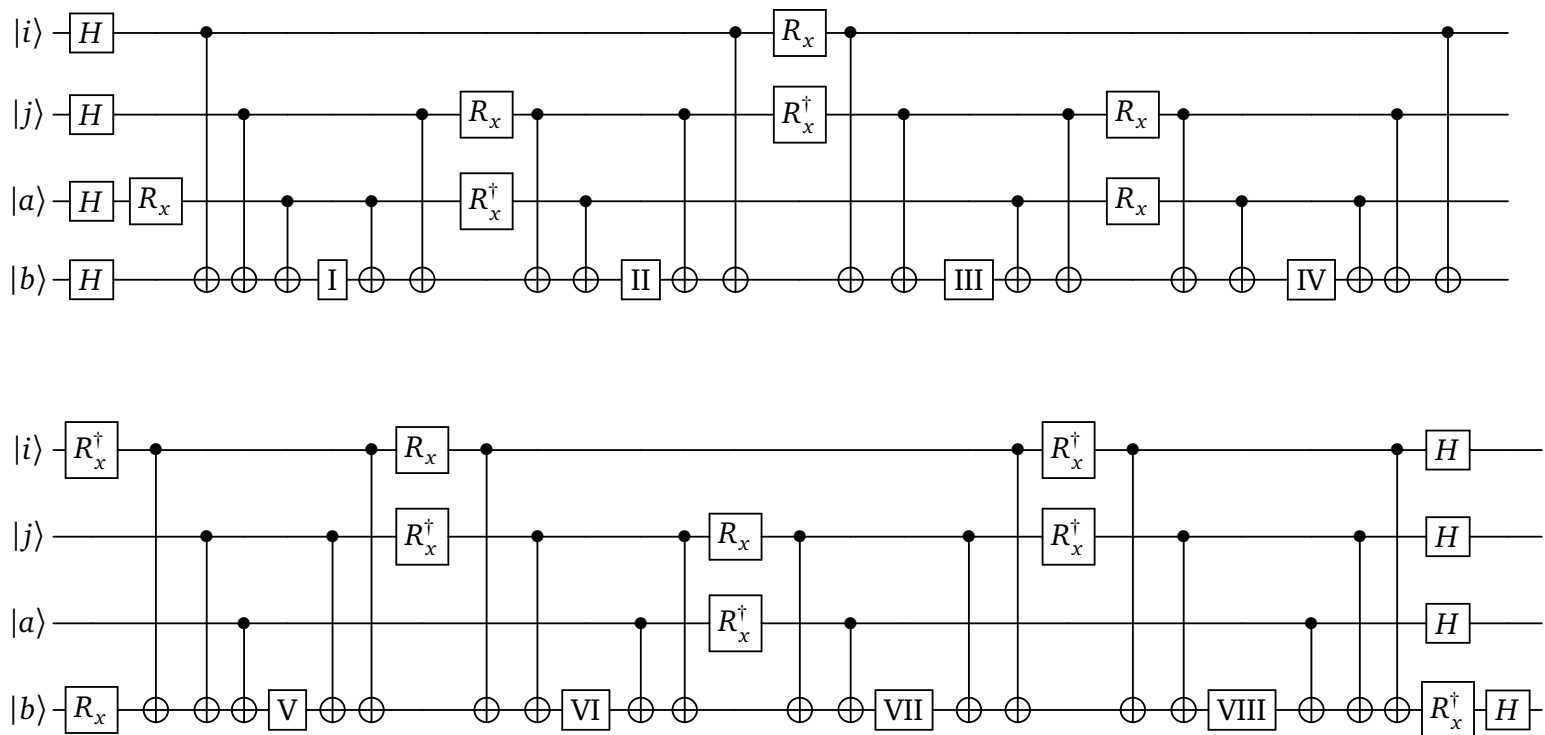
Figure 5.17: Optimized circuit for UCC doubles term.

$$\mathrm{I} \equiv R_z\left(i t^{ab}_{ij}/8\right), \qquad \mathrm{II} \equiv R_z\left(-i t^{ab}_{ij}/8\right),$$

$$\mathrm{III} \equiv R_z\left(-i t^{ab}_{ij}/8\right), \qquad \mathrm{IV} \equiv R_z\left(-i t^{ab}_{ij}/8\right),$$

$$\mathrm{V} \equiv R_z\left(i t^{ab}_{ij}/8\right), \qquad \mathrm{VI} \equiv R_z\left(i t^{ab}_{ij}/8\right),$$

$$\mathrm{VII} \equiv R_z\left(-i t^{ab}_{ij}/8\right), \quad \mathrm{VIII} \equiv R_z\left(i t^{ab}_{ij}/8\right).$$

5.7.2 Symmetry reduction of the doubles terms

For the special case of a system with two particles and four orbitals, the UCCD circuit can be reduced by recognizing that the eight Pauli strings in equation (5.58) act identically on the Hartree-Fock state. Thus, only one term is sufficient to approximate the operator, as shown by McCaskey et al [20]. We follow their example and use the Pauli string $\mathscr{P}_{yxxx}^{ijab}$, shown in figure (5.18). In the the VQE class constructor, the doubles re-
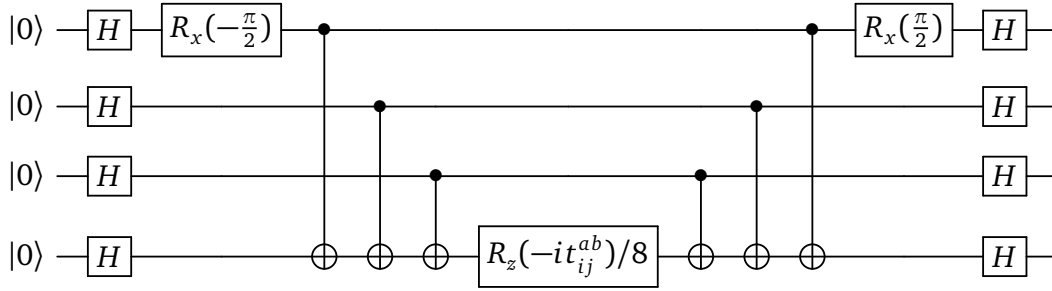
Figure 5.18: Reduced UCC doubles circuit, using only the Pauli string $\mathscr{P}_{yxxx}^{ijab}$.

duction is used by including an 'r' in the ansatz string, e.g. 'UCCDr' for the reduced UCCD circuit.

5.8 More efficient state preparation

Recent research on the Variational Quantum Eigensolver for quantum chemistry focus on finding more efficient state preparations than the UCC ansatz. To avoid searches in unimportant parts of the Hilbert space, we need to ensure that certain symmetries are conserved. Most importantly the particle number, but in some cases this could include other such as spin or the seniority number, as in the pairing model.

5.8.1 Pairing ansatz

First we see that a combination of a z-rotation and y-rotation can represent all points on the Bloch sphere. Drawing inspiration from the preparation of the Bell state in section (5.4.1), we can prepare a pair with two parameters using the following circuit
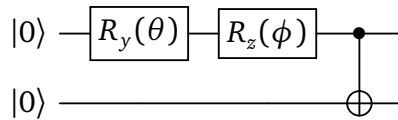
Figure 5.19: State preparation of pair with two parameters θ and ϕ.

Note that for $\theta = \frac{\pi}{2}$ and $\phi = \pi$ this becomes the Bell state, since $\hat{H} = \hat{R}_y(\frac{\pi}{2})\hat{Z}$. Instead of creating an equal superposition of the two pair states $|00\rangle$ and $|11\rangle$, we get a weighted superposition. Furthermore we assume that the inital state of the qubits is the Hartree-Fock state. For two particles and four orbitals this is the state $|1100\rangle$, with

the possible excited state $|0011\rangle$. We then seek a transformation $\hat{U}$ of the qubit state into a normalized superposition of the two states

$$\hat{U}(\omega)|1100\rangle = c_1|1100\rangle + c_2|0011\rangle,$$

where the coefficients c_1 and c_2 depend on a set of parameters ω. When the qubits in figure (5.19) are initialized to the $|11\rangle$ state, we need to make a small adjustment in order to conserve the particle number. By acting with a $\hat{X}$ gate on the control qubit before and after a CNOT gate, the target qubit will be flipped conditional to the control qubit being in the $|0\rangle$ state. To conserve the particle number for the whole state $\hat{U}(\omega)|1100\rangle$, we also need to flip the two last qubits if the first qubit is in the $|0\rangle$ state. We then have the final circuit for the state preparation shown in figure (5.20). This
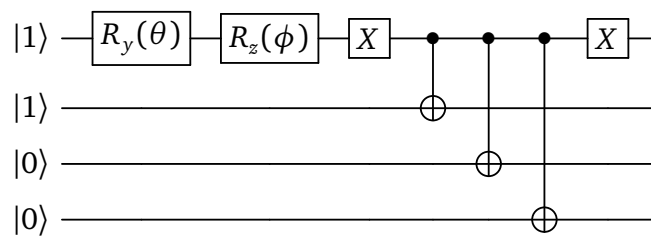

Figure 5.20: State preparation of the pairing model with 2 particles and 4 orbitals. Dependent of two parameters θ and ϕ.

ansatz depend on two parameters compared to one parameter in the UCCD ansatz for the same system. However, the $\hat{R}_y$ gate alone, should be suffiecient to create are superposed state, and so we leave out $\hat{R}_z$ gate. The resulting ansatz is implemented in the RY class with a circuit depth of 6 compared to 55 for the optimized UCCD circuit or 11 for the reduced UCCD, being a huge advantage when we want to implement this on a noisy quantum computer.

5.8.2 A more general ansatz

We can generalize this for a system without the pairing constriction. First we introduce some useful ciruits. The Toffoli gate [11] is the quantum equivalent of the classical AND gate. It is a CNOT gate conditional to an additional qubit, as seen in figure (5.21) with its two-qubit decomposition. The Toffoli gate flips the target qubit if both control
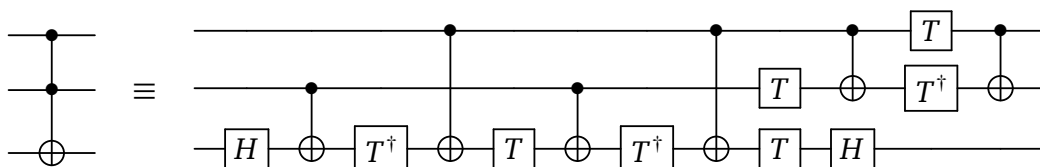

Figure 5.21: The Toffoli gate and its two-qubit decomposition.

qubits are in the $|1\rangle$ state. Acting with $\hat{X}$ gates before and after both control qubits, the target qubit is flipped conditional on both control qubits being in the $|0\rangle$ state. We can
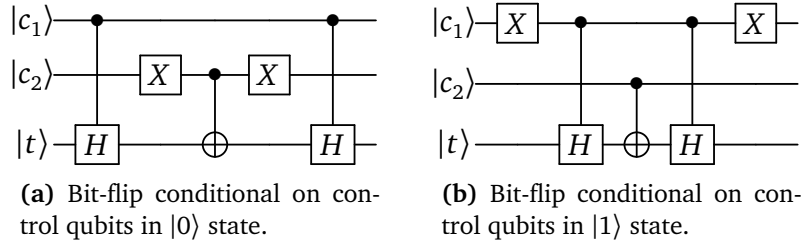
(a) Bit-flip conditional on control qubits in $|0\rangle$ state.

(b) Bit-flip conditional on control qubits in $|1\rangle$ state.

Figure 5.22: Quantum circuits for equivalent of the classical AND gate. **(a)** $|t\rangle$ qubit is flipped conditional to $|c_1 c_2\rangle = |00\rangle$. **(b)** $|t\rangle$ qubit is flipped conditional to $|c_1 c_2\rangle = |11\rangle$

achieve the same action with a reduced circuit as shown in figure (5.22), where both circuits include controlled Hadamard gates. To implement the controlled Hadamard gate we can make use of the circuit identity shown in figure (5.23). We see that if the the control qubit is $|0\rangle$ we get

$$R_y(-\frac{\pi}{4})R_y(\frac{\pi}{4}) = \hat{\mathbb{1}},$$

while we get the Hadamard gate if the target qubit is flipped in between the rotation gates. When we have the qubits in the Hartree-Fock state we need to create a super-
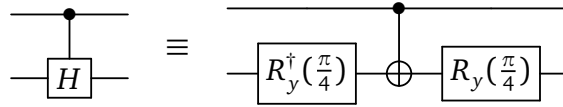

Figure 5.23: Decomposition of the controlled Hadamard gate.

position of the occupied orbitals that represent the possible excitaions. For the case of two particles that is

$$|q_1 q_2\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle. \tag{5.59}$$

This can be done just using rotations gates on the respective qubits, but in addition we include a CNOT gate to entangle the qubits. This gives the gate in figure (5.24), which can be repeated d times for a more complex ansatz. Constructing the ansatz we
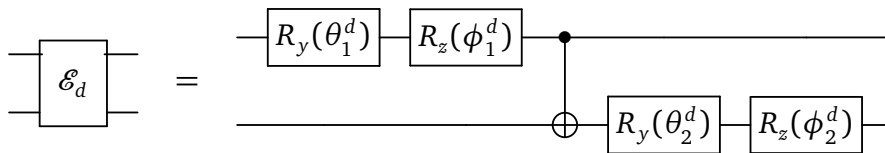

Figure 5.24: Circuit for possible excitations of the occupied orbitals in the Hartree-Fock state, using y and z rotations.

need to ensure that the particle number is conserved. After the R_y and R_z rotations on the occupied orbitals, they can be all possible states. Thus we have two singles excitations, a doubles excitation or no excitation. The singles excitations will leave the two first qubits in either $|01\rangle$ or $|10\rangle$. By storing the parity in the the unoccupied qubits using CNOT gates we can induce a possible single excitation. As with the controlled

Hadamard gate, we can insert rotation gates before and after the CNOT gates on the first unoccupied orbital followed by a CNOT targeted on the second unoccupied orbital to represent the possible singles excitations. So far the ansatz conserves the particle number for all states unless both occupied orbitals are excited, that is that the first two qubits are in the $|00\rangle$ state. We make use of the quantum equivalent of the AND gate given in figure (5.22a) in order to excite the first unoccupied orbital if this is the case. To excite the second qubit as well we include a CNOT gate on the two unoccupied orbitals before and after the AND gate. Thus we have a possible doubles excitation, and the particle number is conserved. This circuit is shown in figure (5.25). We see
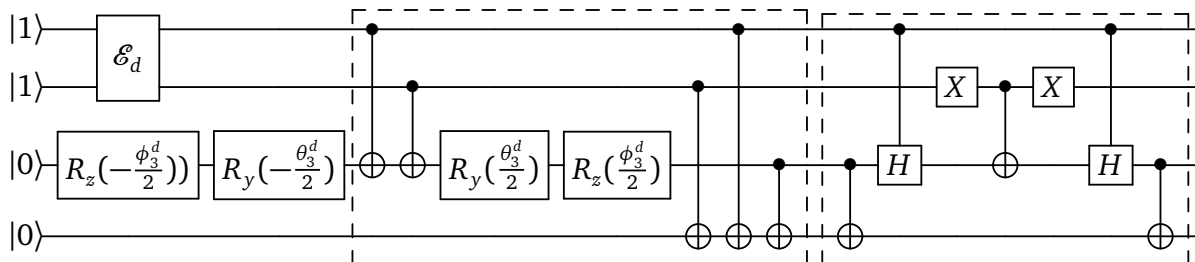

Figure 5.25: General RYRZ ansatz for 2 particles and 4 orbitals. The first box takes account for singles excitations, while the second box takes account for the possible doubles excitation if the two first qubits are in the $|0\rangle$ state. The qubits are initialized in the Hartree-Fock state.

that we can remove two CNOT gates in between the singles and doubles box, giving a circuit depth of $16 + 3d$ if we make use of the controlled Hadamard decomposition in figure (5.23). Typically a a depth of $d = 2$ is sufficient. Compared to the UCCSD, with a circiut depth of at least 99, we get a greatly reduced ansatz possibly spanning the same space.

    This ansatz is implemented in the RYRZ class, which inputs the number of particles and orbitals. However, we have not expanded this strategy further than for 2 particles and 4 orbitals, since this is the system we will run our simulations with. This leads to $6d$ parameters, compared to 5 for UCCSD. By initializing all parameters to 0 we get a good initial guess, being the Hartree-Fock state, since a rotation of 0 radians about an axis is equivalent no rotation e.g. the identity operator.

5.9   Running noisy simulations

In order to prepare an execution of a circuit on a real quantum computer we can simulate its expected behaviour using noise models as explained in section (5.2). Both the QPE and VQE classes input an options dictionary. The following code will run the VQE algorithm calculating the ground state energy of the pairing model with the RY ansatz and the Powell minimization method, simulated with a noise model for IBM's London quantum computer.

```
from quantum_circuits import PairingHamiltonian
from quantum_algorithms import VQE, Minimizer,
                               get_pairing_matrix

n, l = 2, 4
h, v = get_pairing_matrix(n,l)
```

```
pairing = PairingHamiltonian(n,l,h,v)

options = {'device':'ibmq_london',
           'noise_model':True,
           'coupling_map':True,
           'layout':[1,0,2,3],
           'meas_fit':True}
vqe = VQE(pairing,Minimizer('Powell'),'RY',options=options)
theta,E = vqe.optimize()
```

Here we have also included the `meas_fit` option, so that qiskit performs the measurement error mitigation explained in section (5.2). In figure (5.26) we can see the results of running a simulation with and without this option, compared with an ideal simulation.
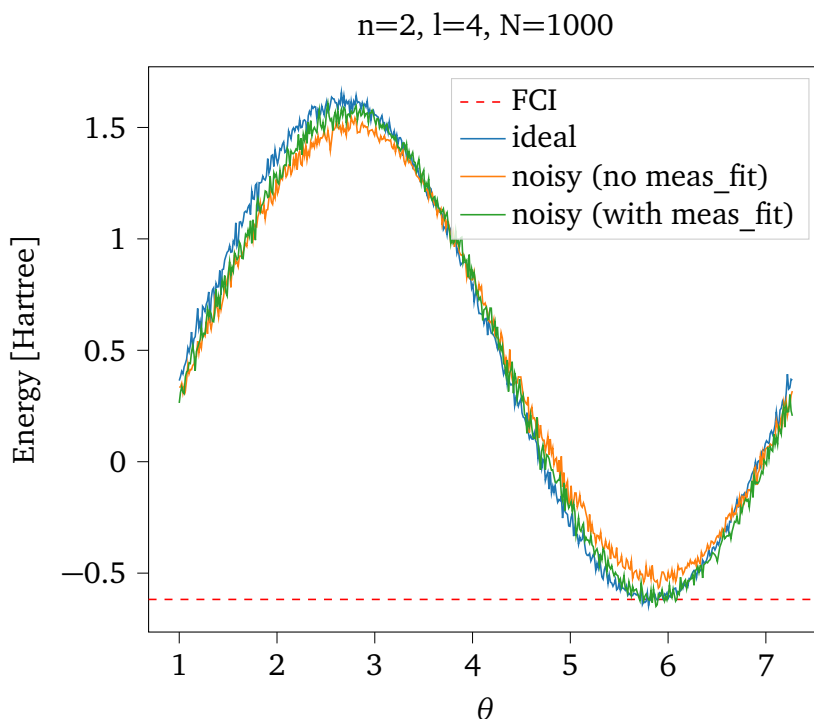

Figure 5.26: Brute force calculations of pairing model with the RY ansätz with θ as the rotation angle in the $\hat{R}_y$ gate in figure (5.20). The noisy plots are simulations done with a noise model from IBM's London quantum computer.

5.10 Executing circuits on an IBM quantum computer

Lastly, we can execute the a transformed Hamiltonian on one of IBM's quantum computers with our VQE class in a simple manner. First we need to define the backend, being the quantum computer, for the execution. In the following code we show both how we can define the specific "London" quantum computer as backend, or the least busy 5 qubit quantum computer.

```
import qiskit as qk
from qiskit.provider.ibmq import least_busy
```

```python
qk.IBMQ.load_account()
provider = qk.IBMQ.get_provider('ibm-q')
# For manual selection of computer
backend1 = provider.get_backend('ibmq_london')
# For selecting least busy computer
qbit5 = provider.backends(filters=lambda x: x.configuration().
                                 n_qubits == 5 and not x.
                                 configuration().simulator)
backend2 = least_busy(qbit5)
```

This is then used as an input to the VQE class through the options dictionary given below.

```python
from quantum_algorithms import VQE

options = {'ibmq':True,'backend':backend1}
vqe = VQE(H,optimizer,ansatz,options=options)
```

Chapter 6

Systems

The main focus of this thesis has been on how quantum algorithms can be efficiently implemented on quantum computers, particularly the VQE algorithm on near-term devices. To demonstrate the functionality of the methods we will look at two different systems in addition to the pairing model introduced in section (5.1). Our algorithms need a set of one-body and two-body matrix elements in order to calculate the ground state energy of some Hamiltonian. For quantum chemistry calculations the `Psi4` package can be used to retrieve the interaction integrals for the matrix elements, in addition to the FCI and HF energies. For a 5-qubit quantum computer using the Jordan-Wigner transformation, we cannot freely choose a molecule to look at due to the limited number of qubits that can represent orbitals. Molecular hydrogen in a minimal basis set result in four orbitals, and can be implemented on the quantum computer. Though this will suffer from poor accuracy compared to other basis sets, we are only interested in the accuracy between the VQE method compared to FCI, not so much the quality of the integrals.

Next we also want a system where the correlation energy clearly deviates for different scenarios. The one-dimensional quantum dot confined by a harmonic oscillator potential with angular frequency ω, will have higher correlation energies for smaller values of ω. As will be clear later, this is due to the expectation value of the Coulomb interaction being the dominant energy contribution.

6.1 Hydrogen molecule

The first system we will look at is molecular hydrogen. Common in quantum chemistry we define the Hamiltonian using the Born-Oppenheimer. The Hamiltonian of the system is written as the kinetic energy of the particles and the potential energy for the Coulomb interactions between them

$$H = T_e + T_n + V_{ee} + V_{nn} + V_{ne}, \tag{6.1}$$

and the wavefunction can be written as the product of an electron and nucleus wave function

$$\Psi = \Phi_{elec}(\{\mathbf{r}_i,,\}\{\mathbf{R}_A,)\}\Phi_{nuc}(\{\mathbf{R}_A,)\}, \tag{6.2}$$

where the electronic wave function only depends explicitly [1] on the nuclear coordinates. Here it is assumed that the nucleus can be treated as stationary, since it will move more slowly than the electrons. Thus the electrons can be thought of as moving

in the field of a fixed nucleus. Here it is assumed that because of the large mass ratio between an electron and a nucleon, the two move at vast different speeds, and so the nucleon can be treated as stationary. The electron can then be treated separately from the nucleon. Consequently we set $T_n = V_{nn} = 0$, since these terms only will scale the total energy by a fixed amount. We are then left with the electronic Hamiltonian

$$H = T_e + V_{ee} + V_{ne}, \tag{6.3}$$

and the electronic wavefunction

$$\Psi = \Phi_{elec}. \tag{6.4}$$

On second quantized form this can be written

$$\hat{H} = E_{nuc} + \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} v_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r, \tag{6.5}$$

where h_{pq} and v_{pqrs} are the one-body and two-body integrals, and E_{nuc} is the electrostatic repulsion between the nuclei.

$$E_{nuc} = \frac{1}{2} \sum_{i \neq j} \frac{Z_i Z_j}{\left| \mathbf{R}_i - \mathbf{R}_j \right|}, \tag{6.6}$$

$$h_{pq} = \int d\mathbf{x} \phi_p^*(\mathbf{x}) \left(-\frac{\nabla_\mathbf{r}}{2} - \sum_i \frac{Z_i}{|\mathbf{R}_i - \mathbf{r}|} \right) \phi_q(\mathbf{x}), \tag{6.7}$$

$$h_{pqrs} = \iint d\mathbf{x}_1 d\mathbf{x}_2 \frac{\phi_p^*(\mathbf{x}_1) \phi_q^*(\mathbf{x}_2) \phi_s(\mathbf{x}_1) \phi_r(\mathbf{x}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}, \tag{6.8}$$

where Z is the nuclear charge, $\mathbf{R}$ are the nuclear spatial coordinates, $\mathbf{r}$ is the electron spatial coordinates and $\mathbf{x}$ is the spatial and spin coordinates of the electron. The matrix elements will be calculated with the `Psi4` package, with a minimal basis set STO-3G.

6.1.1 Transformed Hamiltonian

We transform the Hamiltonian using the Jordan-Wigner transformation with the `SecondQuantizedHamiltonian` class in the `quantum_circuit` module. We show in table (6.2) the resulting Pauli-strings when the bond bond length is equal to $R = 1$. The matrix elements are gathered using the `get_h2_matrix` function from `quantum_algorithm` module, for the molecular orbitals. We need to transform them into matrix elements for spin orbitals. This is done in the Hamiltonian class by setting `add_spin=True` in the constructor as shown below for $R = 0.74$.

```
from quantum_circuit import SecondQuantizedHamiltonian
from quantum_algorithms.systems import get_h2_matrix

R = 0.74
h,v,_ = get_h2_matrix(R)
qdot = SecondQuantizedHamiltonian(2,4,h,v,add_spin=True)
qdot.group_paulis(qwc=True,gc=True)
```

This result in the Pauli strings given in table (6.1).

Table 6.1: Jordan-Wigner transformed Hamiltonian for the Hydrogen molecule in a minimal basis, with bond length $R = 0.74$. The Pauli-strings are grouped into a qubit-wise commuting (QWC) group and a general commuting (GC) group.

QWC group		GC group	
Factor	Operator	Factor	Operator
-0.09707	IIII	-0.04530	XXYY
0.17141	ZIII	0.04530	YXXY
0.17141	IZII	-0.04530	YYXX
-0.22343	IIZI	0.04530	XYYX
-0.22343	IIIZ		
0.16869	ZZII		
0.16593	IZZI		
0.17441	IIZZ		
0.12063	ZIZI		
0.16593	ZIIZ		
0.12063	IZIZ		

6.2   The one-dimensional quantum dot

A quantum dot consist of bounded particles confined in a small area. The confining potential defines the different types of quantum dots. For the one-dimensional quantum dot we have the one-body operator

$$\hat{h}(x) = -\frac{1}{2}\frac{d^2}{dx^2} + \hat{v}(x), \tag{6.9}$$

where the confining potential $\hat{v}$ usually is chosen to be the harmonic oscillator potential in one dimension

$$\hat{v}(x) = \frac{1}{2}\omega^2 x^2. \tag{6.10}$$

The two-body operator represents the interaction between the particles and can be the Coulomb interaction. As we later on will solve the matrix elements using a grid, we instead choose the shielded Coulomb interaction

$$\hat{V}(x_1, x_2) = \frac{\alpha}{\sqrt{(x_1 - x_2)^2 + \lambda^2}}, \tag{6.11}$$

where α is a dimensionless constant and λ is the screening parameter. With λ we avoid the singularity at $x_1 = x_2$. For all calculation in this thesis we make these parameters to $\lambda = 0.25$ and $\alpha = 1$. In terms of a single-particle basis $\{\psi_p\}$ we can write the Hamiltonian on second quantized form

$$\hat{H} = \sum_{pq}\langle\psi_p|\hat{h}(x)|\psi_q\rangle\hat{a}_p^\dagger\hat{a}_q + \sum_{pqrs}\langle\psi_p\psi_q|\hat{V}(x_1, x_2)|\psi_r\psi_s\rangle\hat{a}_p^\dagger\hat{a}_q^\dagger\hat{a}_s\hat{a}_r. \tag{6.12}$$

6.2.1   Matrix elements

To calculate the matrix elements have made use of a quantum dot solver developed by former master students [21, 22]. We summarize the their approach below.

We find the single-particle functions by solving

$$\hat{h}(x)\phi_n(x) = \epsilon_n \phi_n(x). \tag{6.13}$$

For the harmonic oscillator potential we know from introductory quantum mechanics, see e.g. [23], that the eigenfunctions in natural units are given by

$$\phi_n(x) = \left(\frac{\omega}{\pi}\right)^{1/4} \frac{1}{\sqrt{2^n n!}} H_n(\sqrt{\omega}x)e^{-\omega x^2/2}, \tag{6.14}$$

where $H_n(x)$ are the Hermite polynomials

$$H_n(x) = (-1)^n e^{x^2} \left(\frac{d}{dx}\right)^n e^{-x^2}. \tag{6.15}$$

The single-particle energies yield

$$\epsilon_n = \left(n + \frac{1}{2}\right)\omega. \tag{6.16}$$

From equation (6.14) we define the spin orbital basis as

$$\psi_{p\sigma} = \phi_n(x)\sigma_{\pm}, \tag{6.17}$$

from which we can calculate the two-body matrix elements given by the integral

$$\langle \psi_p \psi_q | \hat{V}(x_1, x_2) | \psi_r \psi_s \rangle = \iint dx_1 dx_2 \psi_p^*(x_1)\psi_q^*(x_2)\hat{V}(x_1, x_2)\psi_r(x_1)\psi_s(x_2). \tag{6.18}$$

This can be rewritten by introducing the inner integral

$$W_{qs}(x_1) = \int dx_2 \psi_q^*(x_2)\hat{V}(x_1, x_2)\psi_s(x_2), \tag{6.19}$$

giving

$$\langle \psi_p \psi_q | \hat{V}(x_1, x_2) | \psi_r \psi_s \rangle = \int dx_1 \psi_p^*(x_1) W_{qs}(x_1)\psi_r(x_1). \tag{6.20}$$

The integrals can be approximated numerically be defining a grid $x \in [a, b]$ of N equally spaced points and make use of the trapezoidal rule

$$\int_a^b f(x)dx \approx \Delta x \sum_k^N \frac{f(x_k + \Delta x) + f(x_k)}{2}. \tag{6.21}$$

We then have

$$W_{qs}(x_1) \approx \Delta x \sum_k^N \frac{\psi_q^*(x_k)\hat{V}(x_1, x_k + \Delta x)\psi_s(x_l) + \psi_q^*(x_k)\hat{V}(x_1, x_k)\psi_s(x_k)}{2}, \tag{6.22}$$

$$\langle \psi_p \psi_q | \hat{V}(x_1, x_2) | \psi_r \psi_s \rangle \approx \Delta x \sum_k^N \frac{\psi_p^*(x_k)W_{qs}(x_k + \Delta x)\psi_r(x_k) + \psi_p^*(x_k)W_{qs}(x_k)\psi_r(x_k)}{2}.$$

$$\tag{6.23}$$

6.2.2 Transformed Hamiltonian

We transform the Hamiltonian using the Jordan-Wigner transformation with the `Sec-ondQuantizedHamiltonian` class in the `quantum_circuit` module. As an example we show in table (6.2) the resulting Pauli-strings when $\omega = 1$ for a system of two particles and four orbitals. The matrix elements are calculated using the `get_qdot_matrix` function from `quantum_algorithm` module, they are anti-symmetric so we need to divide them by 4 which can be done in the Hamiltonian class by setting `anti_symmetric=True` in the constructor as shown below.

Table 6.2: Jordan-Wigner transformed Hamiltonian for a one-dimensional quantum dot with two particles, four orbitals and $\omega = 1$. The Pauli-strings are grouped into a qubit-wise commuting (QWC) group and a general commuting (GC) group.

QWC group		GC group	
Factor	Operator	Factor	Operator
3.76171	IIII	-0.12207	XXYY
-1.18309	ZIII	0.12207	YXXY
-1.18309	IZII	-0.12207	YYXX
-1.57862	IIZI	0.12207	XYYX
-1.57862	IIIZ		
0.43310	ZZII		
0.31103	IZZI		
0.32864	IIZZ		
0.18896	ZIZI		
0.31103	ZIIZ		
0.18896	IZIZ		

```
from quantum_circuit import SecondQuantizedHamiltonian
from quantum_algorithms.systems import get_qdot_matrix

h,v,_ = get_qdot_matrix(2,4,omega=1)
qdot = SecondQuantizedHamiltonian(2,4,h,v,anti_symmetric=True)
qdot.group_paulis(qwc=True,gc=True)
```

Part III

Results

Chapter 7

Phase Estimation

For our aim in this thesis, as we will show, we will not spend much time examining the results of the quantum phase estimation algorithm (QPE). Previously we showed how it can find the ground state and excited states for the pairing model. We continue this by showing how the algorithm performs under the influence of noise. In figure (7.1) we plot the results of a noisy simulation of the pairing model for four simulation qubits and six work qubits. Having a total of ten qubits, we make use of the noise model from IBM's 16-qubit quantum computer "Melbourne". We make use of a total of five Trotter steps since this is sufficient to show that the noise quickly becomes intractable. We see that can not extract any information from these results, as no clear peaks are present. Thus showing that the use of QPE not realistic for use on near-term quantum computers. In addition it requires multiple controlled operations from the work register to the control register, as well as within both register. Since the number of Trotter step have to be large to yield good results, the resulting circuit depth becomes very long.
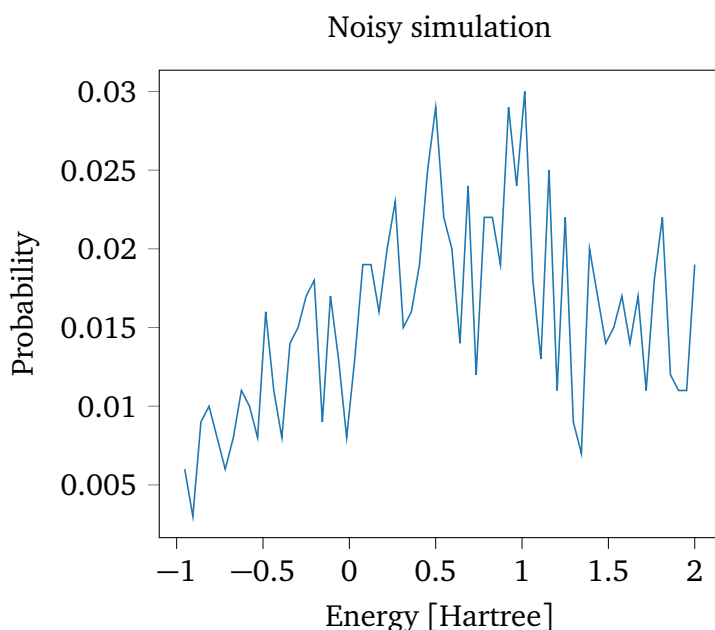

Figure 7.1: Quantum phase estimation on the pairing model with four simulation qubits and six work qubits for level spacing $\delta = 1$ and interaction strength $g = 1$. Simulated with a noise model from IBM's "Melbourne" quantum computer.

Chapter 8

Variational Quantum Eigensolver

The main focus of our results will be on the Variational Quantum Eigensolver (VQE). We verify the functionality of our implementation by first looking at the its performance on the hydrogen molecule and the one-dimensional quantum dot, running both simulations of the expected outcome of our circuits as well as executing them on a real quantum computer. Then we run simulations for larger systems in order to see the advantage of using a unitary coupled cluster (UCC) ansätz instead of traditional coupled cluster. Lastly we will discuss the impact of noise in our simulations with respect to the different state preparations.

8.1 Hydrogen molecule

To validate the functionality of the different state preparations and optimization methods for the VQE algorithm we will start by looking at the simplest molecule available; the Hydrogen molecule. All simulations will be run with $N = 8192$ shots if not stated otherwise. For all bond lengths the Jordan-Wigner transformed Hamiltonian involve the same Pauli strings, listed in table (6.1). We group all Pauli strings with only $\hat{Z}$ operators into a qubit-wise commuting (QWC) group that are measured simultaneously. The remaining Pauli strings are contained in the general commuting group discussed in section (5.5.2) and (B.4), that are measured simultaneously using the additional circuit from the above sections, increasing the circuit depth by 9. This result in two state preparations per expectation value, compared to the naive 14 state preparations without grouping.

8.1.1 Ideal simulations

Before we can run the circuit on a real quantum computer we need ensure that the state preparations are able to represent the ground state. Here we present the results of ideal simulations of the Hydrogen molecule.

Optimization methods

Choosing the right optimization method for specific problems is important, as some methods need significantly less function evaluations to converge to good values. In table (8.1) we show the results of running the VQE algorithm for the H_2 molecule

Table 8.1: The ground state energy (E) and number of function evaluations (N_{eval}) after ideal simulations of the H_2 molecule with bond length $R = 0.74\text{Å}$ using the VQE algorithm for different classical optimization methods and state preparations. The FCI energy is by comparison equal to $E_{FCI} = -1.13728$.

	RYRZ		UCCSD	
	E [Hartree]	N_{eval}	E [Hartree]	N_{eval}
Cobyla	-1.13479	193	-1.13504	104
Powell	-1.12729	301	-1.13037	309
Nelder	-1.11572	210	-1.12498	213
SPSA	-1.13608	451	-1.13667	451

Table 8.2: Circuit information of the UCCSD and RYRZ state preparations.

Ansatz	Parameters	Circuit depth	CNOT gates	Single qubit gates
RYRZ	12	23	10	28
UCCSDr	5	59	30	55
UCCSD	5	104	60	78

with bond length $R = 0.74\text{Å}$ using the classical optimization methods discussed in section (5.5.5), for both the UCCSD and RYRZ state preparations.

The FCI energy is -1.13728 Hartree, and so we see that the Cobyla method performs well compared to the other methods in terms of both accuracy and number of function evaluations for all ansätze. The SPSA method appear to perform well in reaching close to the optimal parameter, but its convergence criteria seems unable to tackle the statistical errors from a finite number of measurements. For the results further on we therefore choose to use the Cobyla method exclusively.

State preparations

We run VQE simulations of the ground state of the Hydrogen molecule for a range of bond lengths. All simulations make $N = 8192$ measurements, being the maximum amount allowed when running on the IMB computers. First we show the results for the UCCSD and RYRZ ansätze, shown in figure (8.1). Here we plot the energies for a set of bond length between $R = 0.3\text{Å}$ to $R = 3.0\text{Å}$ with FCI and HF energies to compare, along with the absolute error $\left|E_{VQE} - E_{FCI}\right|$ between the VQE and FCI energies, that is compared to chemical accuracy defined as ≈ 0.0016 Hartree [9]. The parameter optimization was done with the Cobyla method and a convergence tolerance of 10^{-6}, being a lower bound to the expected measurement error. However, we note that a lower convergence tolerance potentially can give better results.

For the UCCSD ansatz we make use of the reduced circuit shown in section (5.7.2), giving five parameters for a hydrogenic problem. For the RYRZ ansatz we need to use a depth of 2 in order to setup the correct state, giving 12 parameters. In average the RYRZ ansatz had to make twice as many function evaluations as UCCSD to converge. To compare the two state preparations we summarize the circuit information in table (8.2), where we list the resulting circuit depths and gate counts. It is clear that with RYRZ we trade a more efficient circuit for a larger parameter set and more function evaluations, compared to UCCSD.
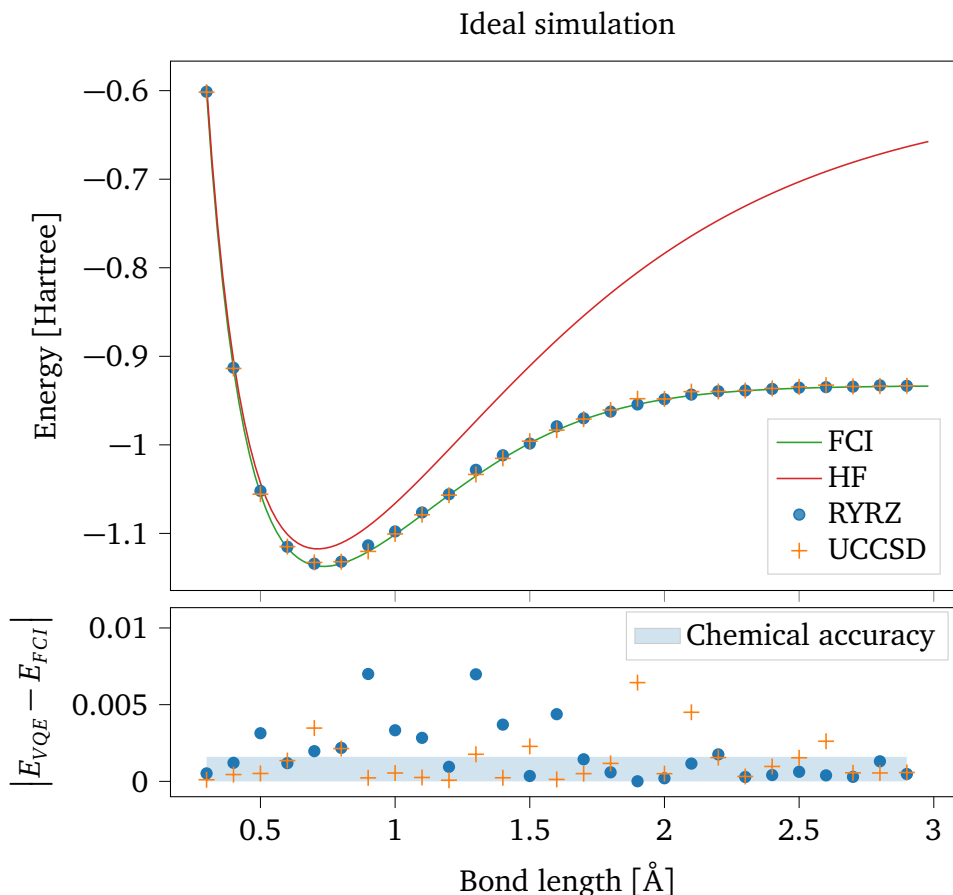
Figure 8.1: Ground state energy of the Hydrogen molecule for bond length between $R = 0.3$Å to $R = 2.3$Å, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with the reduced UCCSD ansatz and the RYRZ ansatz, using the Cobyla method and $N = 8192$ measurements.

8.1.2   Noisy simulations

Next we run the same simulations with a noise model from IBM's "London" quantum computer. In addition we make use of the coupling map given in section (5.2). The results are plotted in figure (8.2). We see that both ansätze perform significantly worse when noise is present. Energies calculated with the RYRZ ansatz yield somewhat accurate results, giving lower energies than HF for most bond lengths, though we desire better accuracy for such a simple system. The UCCSD ansatz have a much longer circuit depth, and as one would expect it gives poor results in the presence of noise. It seems like it follows the shape of the curve, but this is most likely due to the constant term in the circuit list for the identity operator which is by far the most dominant term. We will elaborate this further on when we discuss the quantum dot results.

8.1.3   More efficient state preparation

In a minimal basis, the two electrons each have one possible excitation. Using the `get_state_coeff()` function from the VQE class on the previous results, it turns out that the $|1100\rangle$ and $|0011\rangle$ states are dominating for all bond lengths. That is, both electrons are either excited or not excited at the same time. This motivates the use
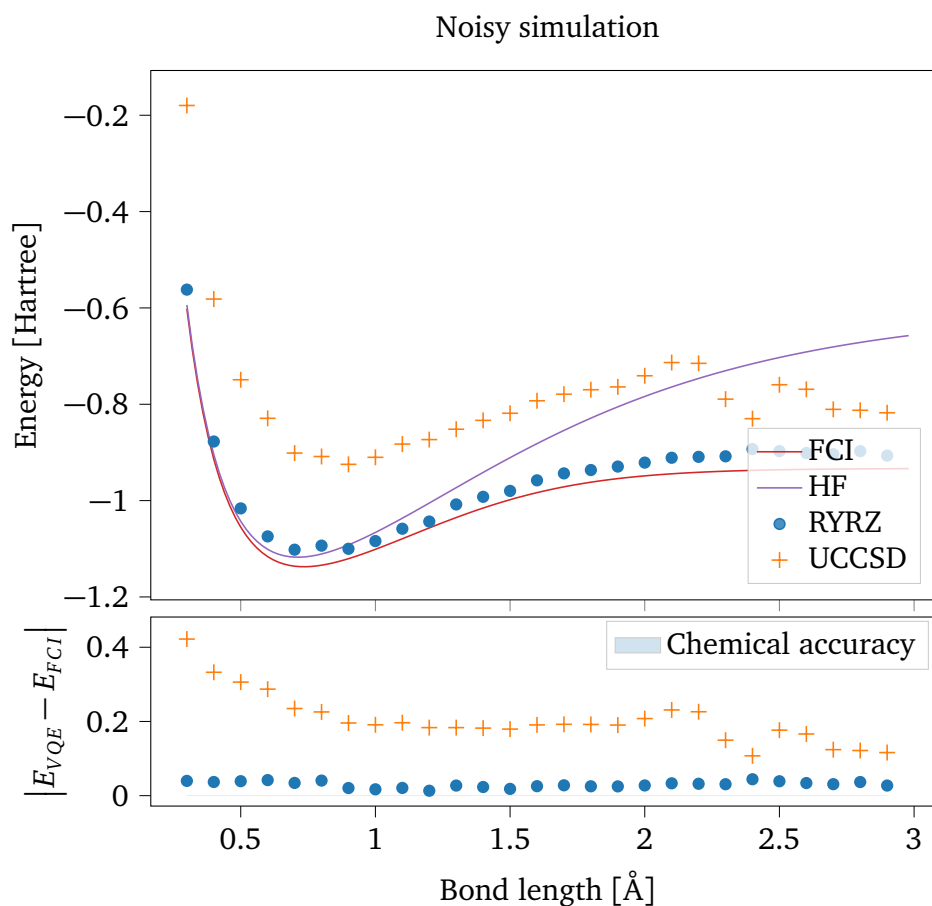
Figure 8.2: Ground state energy of the Hydrogen molecule for bond length between $R = 0.3\text{Å}$ to $R = 2.3\text{Å}$, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with UCCSD and RYRZ ansatz, using the Cobyla method and $N = 8192$ measurements. Simulated using a noise model from the "London" quantum computer.

of the UCCD and RY ansätze, which only make doubles excitations from the Hartree-Fock state. As with the former state preparations we show the circuit information in table (8.3), both having much shorter circuit depth. In addition we check how the

Table 8.3: Circuit information of the UCCSD and RYRZ state preparations.

Ansatz	Parameters	Circuit depth	CNOT gates	Single qubit gates
RY	1	8	3	9
UCCDr	1	11	6	15
UCCD	1	56	36	38

optimization methods perform with these state preparations, given in table (8.4), for a bond length $R = 074$Å. Again we run ideal simulations for the same bond lengths,

Table 8.4: The ground state energy (E) and number of function evaluations (N_{eval}) after ideal simulations of the H_2 molecule with bond length $R = 0.74$Å using the VQE algorithm for different classical optimization methods and state preparations.

	RY		UCCD	
	E [Hartree]	N_{eval}	E [Hartree]	N_{eval}
Cobyla	-1.13731	25	-1.13544	25
Powell	-1.13611	63	-1.13368	80
Nelder	-1.13722	201	-1.12800	149
SPSA	-1.13672	451	-1.13763	451

number of measurements, optimization method and convergence tolerance. The resulting energies and absolute errors are shown in figure (8.3). Because of the shorter circuit depths and gate counts we would expect much better results for simulations in presence of noise. We plot the result for the noisy simulations using the same noise model as before. This is shown in figure (8.4), along with the absolute error. Clearly the RY ansatz performs better than the reduced UCCD ansatz, regardless of the small difference in circuit depth. The noise associated with CNOT gates is far greater than that of the single qubit gates. So as the UCCD makes use of more CNOT gates, in addition to the CNOT gates included from the GC group measurement preparation, the noise is more prominent in simulations with UCCD than with RY. However, the resulting accuracy using RY seems to get worse as the bond length gets larger. Its inability to reach chemical accuracy might be explained using figure (5.26) in section (**??**). Here the optimal parameter for a noisy simulation is unable to reach the exact energy, and so in the presence of noise we cannot expect better results.

8.1.4   Execution on a quantum computer

From the above results we find that the best possible execution will involve the RY ansatz optimized by the Cobyla method. We run the calculations on IBM's "London" quantum computer, with $N = 4000$ measurements. The results are shown in figure (8.5), for the mean value of the last five function evaluations of the expectation value. We see that the results are as expected from the noisy simulations, however it is clear that the absolute error gets worse for lower bond lengths. These are low correlated areas, where one might expect the results to be more accurate. Looking at the
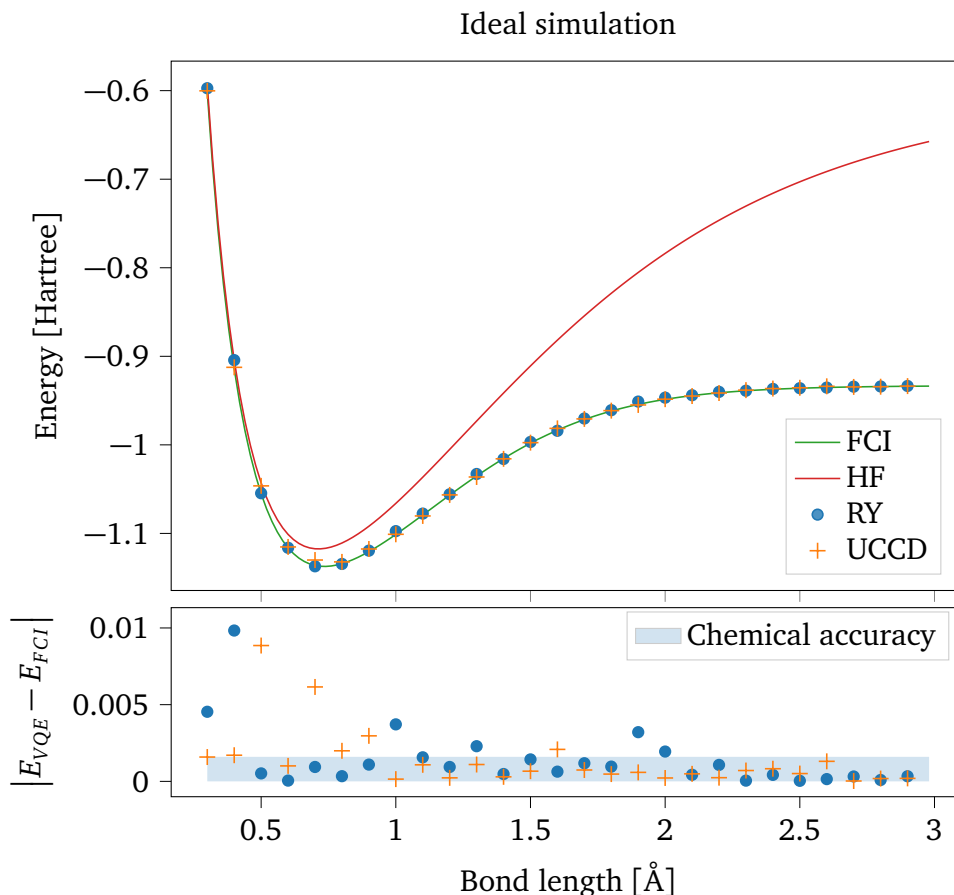
Figure 8.3: Ground state energy of the Hydrogen molecule for bond length between $R = 0.3$Å to $R = 2.3$Å, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with UCCD ansatz along with the simple RY pair ansatz, using the Cobyla method and $N = 8192$ measurements.

noisy simulations of the UCCSD in figure (8.2), the same behaviour is apparent. Thus, one could conclude that this is an area more sensitive to noise. This could be explained by the fact that the constant term in the transformed Hamiltonian, being simply the Pauli string with identity operators on all qubits, is the dominating term. For small bond length the deviation from the FCI energy is much bigger than for larger bond lengths, this is shown in figure (8.6). Looking at the other Pauli strings, the ones corresponding to one-body operators also increase in factor, being the constant associated with each Pauli string, as the bond length gets shorter. The Pauli strings corresponding to two-body operators stay approximately in the same order of magnitude. We discuss this more in detail later on.
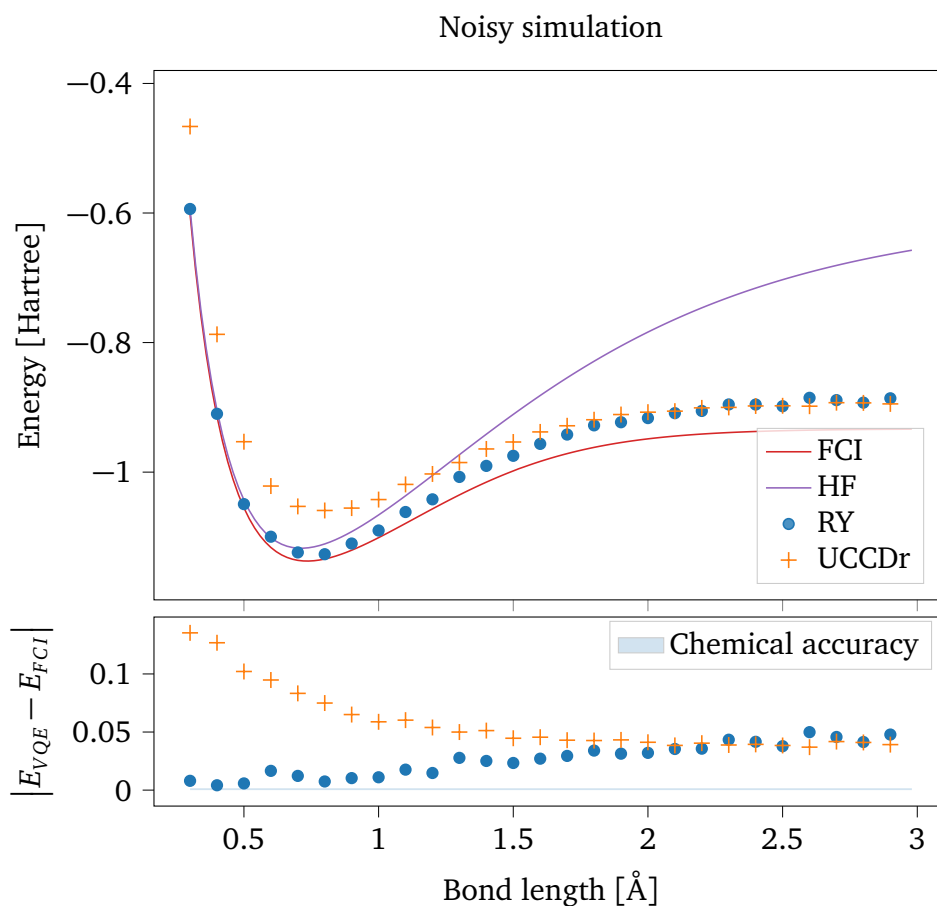
Figure 8.4: Ground state energy of the Hydrogen molecule for bond length between $R = 0.3$Å to $R = 2.3$Å, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with UCCD ansatz along with the simple RY pair ansatz, using the Cobyla method and $N = 8192$ measurements. Simulated using a noise model from the "London" quantum computer.
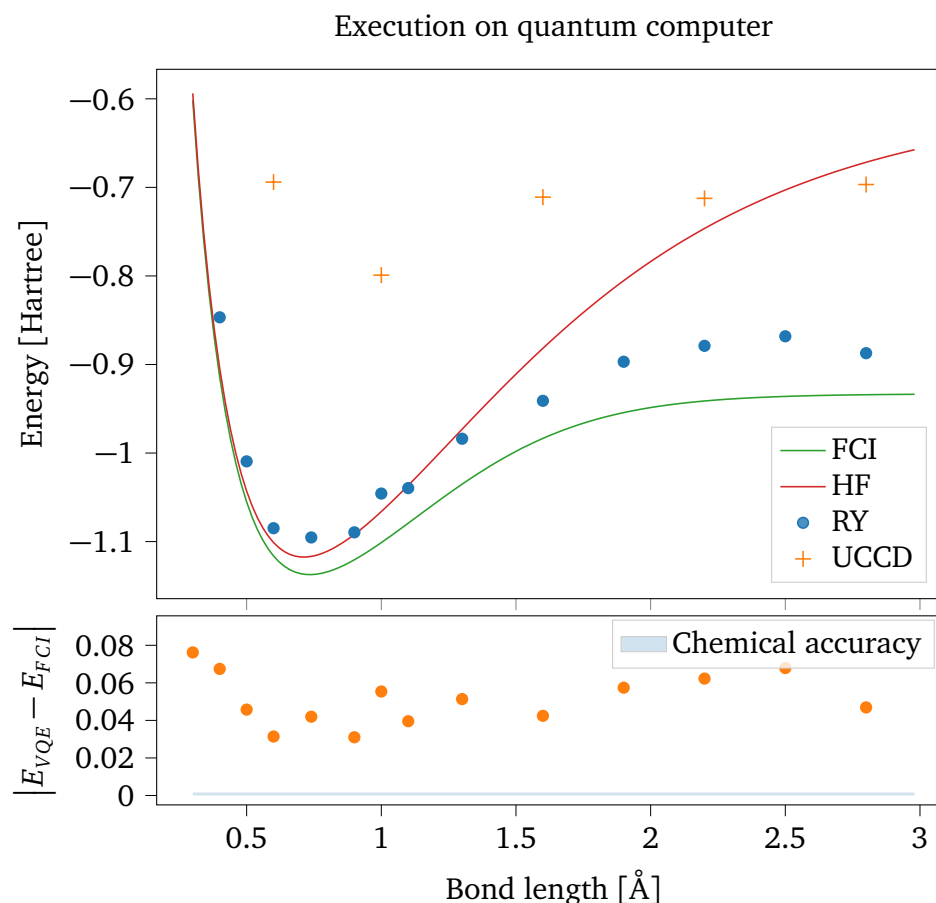
Figure 8.5: Ground state energy of the Hydrogen molecule, executed on the 5-qubit "London" quantum computer, for bond length between $R = 0.3$Å to $R = 2.3$Å, with the absolute error between the VQE and FCI calculations plotted below, for RY only since the UCCD error is so large in comparison. VQE calculations done with RY and UCCD, using the Cobyla method and $N = 4000$ measurements.
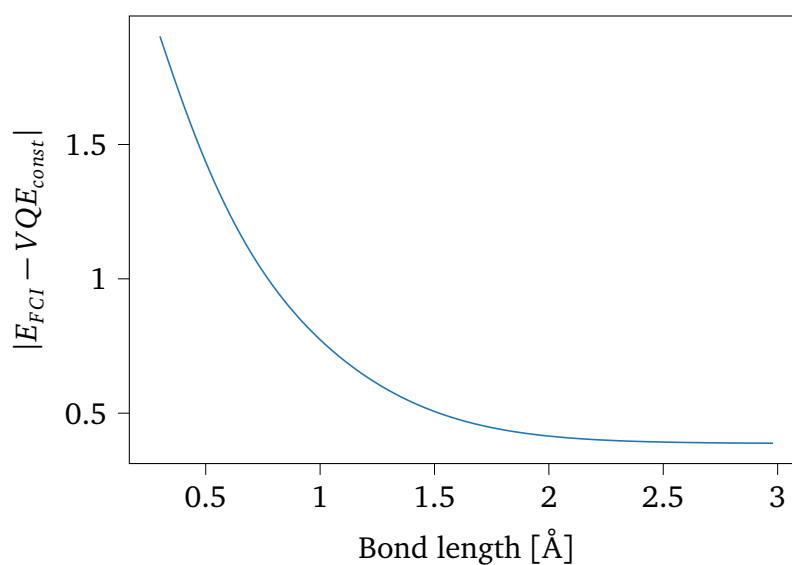

Figure 8.6: The constant term from the Jordan-Wigner transformation of the H_2 Hamiltonian, compared to the FCI energy for bond lengths between $R = 0.3$Å to $R = 2.3$Å. The deviation decrease for larger bond lengths.

8.2   One-dimensional quantum dots

We run similar simulations for the one-dimensional quantum dots. From table (6.1) and (6.2) we see that for two particles and four orbitals the Pauli strings operators for quantum dots and molecular hydrogen are the same. We might suspect that the results will reflect this. All simulations are done using $N = 8192$ measurements, when not stated otherwise.

8.2.1   Simulations

In figure (8.7) we plot resulting energies and absolute errors between the FCI and VQE energies for a range of angular frequencies between $\omega = 0.1$ to $\omega = 2.9$, using the RYRZ and UCCSD state preparations. It is apparent that RYRZ is sufficient for
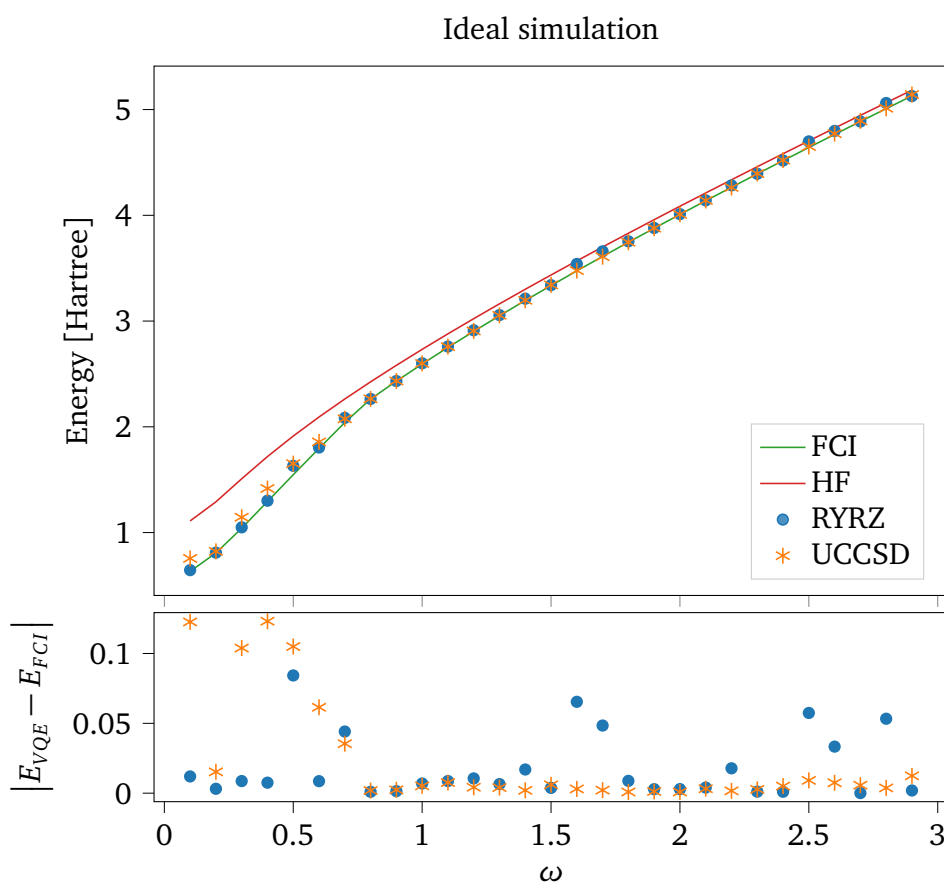

Figure 8.7: Ground state energy of the one-dimensional quantum dot for $\omega = 0.1$ to $\omega = 2.9$, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with UCCSD and RYRZ ansätze, using the Cobyla method and $N = 8192$ measurements.

preparing the qubit state when correlations are strong for small values of ω, as well as larger values where the correlations are not as strong. We see that for small values of ω, the UCCSD perform worse. Next we run noisy simulations using the noise model from IBM's "London" quantum computer, shown in figure (8.8). We see that RYRZ manages to converge to energies close to FCI, and find better energies than HF for most values, with some exceptions for large values of ω. For these values the error between FCI
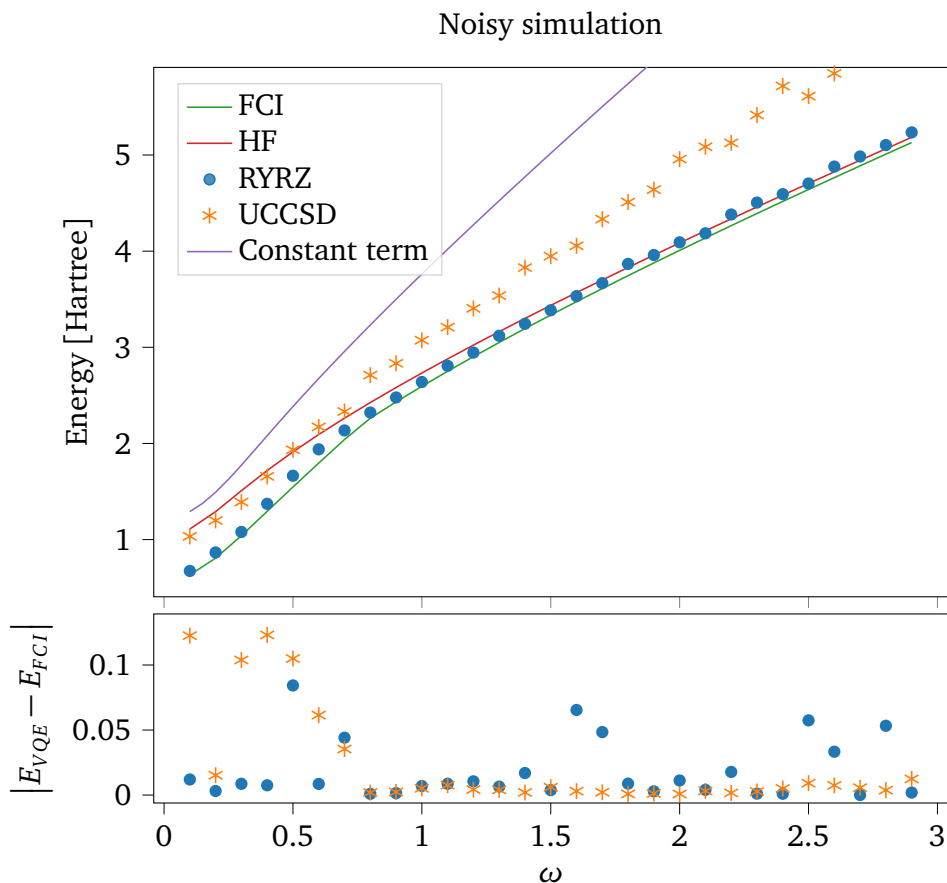
Figure 8.8: Ground state energy of the one-dimensional quantum dot for $\omega = 0.1$ to $\omega = 2.9$, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with UCCSD and RYRZ ansätze, using the Cobyla method and $N = 8192$ measurements. Simulated using a noise model from the "London" quantum computer. The constant term from the Paulis string with only identity operators is plotted s well.

and HF, and thus the correlation energies, are smaller. As a result the noise becomes more apparent. This is due to the expectation values of the Pauli strings corresponding to one-body operators being larger, along with deviations between the constant term from the Jordan-Wigner transformation and FCI growing rapidly. This can be seen in figure (8.8), where the constant term is plotted. We will discuss the impact of this later on.

Lastly we include calculations using the RY and UCCD state preparations as well, given in figure (8.9) and (8.11). This is because it will become useful when we discuss the different state preparations later on. We see that they perform poorly for small values of ω, where clearly single excited Slater determinants are more important. Looking at the state probabilities from the previous simulations, found with `get_state_coeff()` in the VQE, we see that for larger values of ω, the $|1100\rangle$ and $|0011\rangle$ are dominating, but not for smaller values.
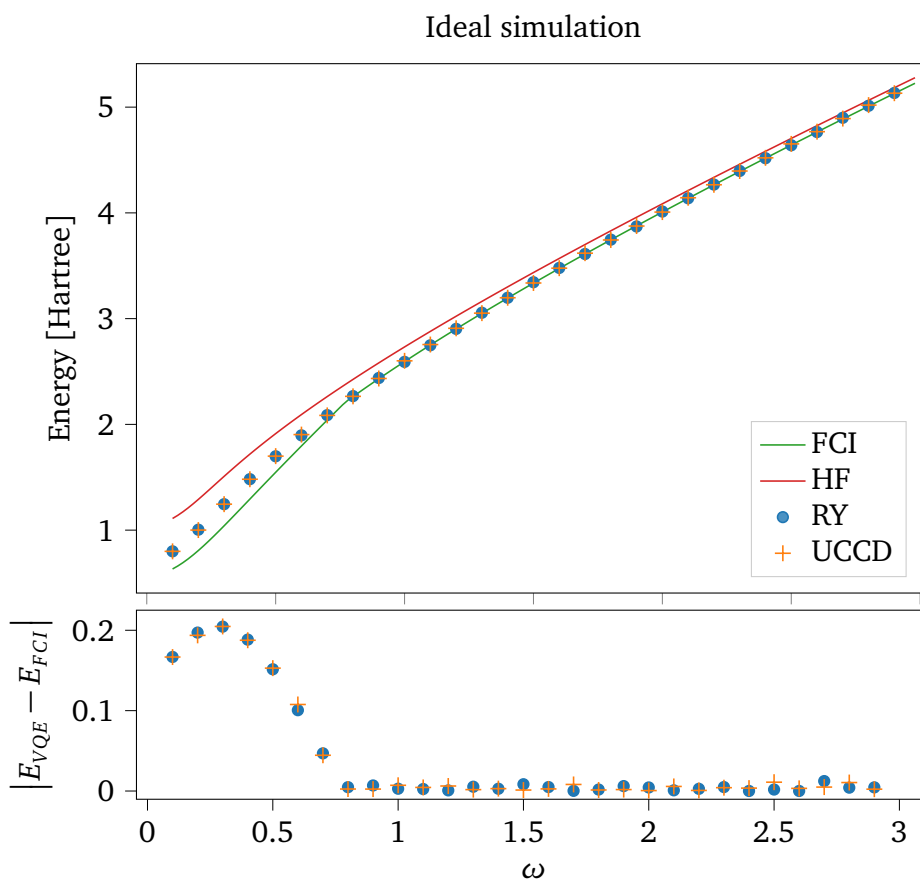

Figure 8.9: Ground state energy of the one-dimensional quantum dot for $\omega = 0.1$ to $\omega = 2.9$, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with UCCD and RY ansätze, using the Cobyla method and $N = 8192$ measurements.

8.2.2   Execution on a quantum computer

In figure (8.8) we see that RYRZ also yields good results when noise is present, though it needed a depth of 2 in order to do so. Thus we would need additional CNOT gates in order to swap the qubits to get the connectivity of the 5-qubit quantum computers.
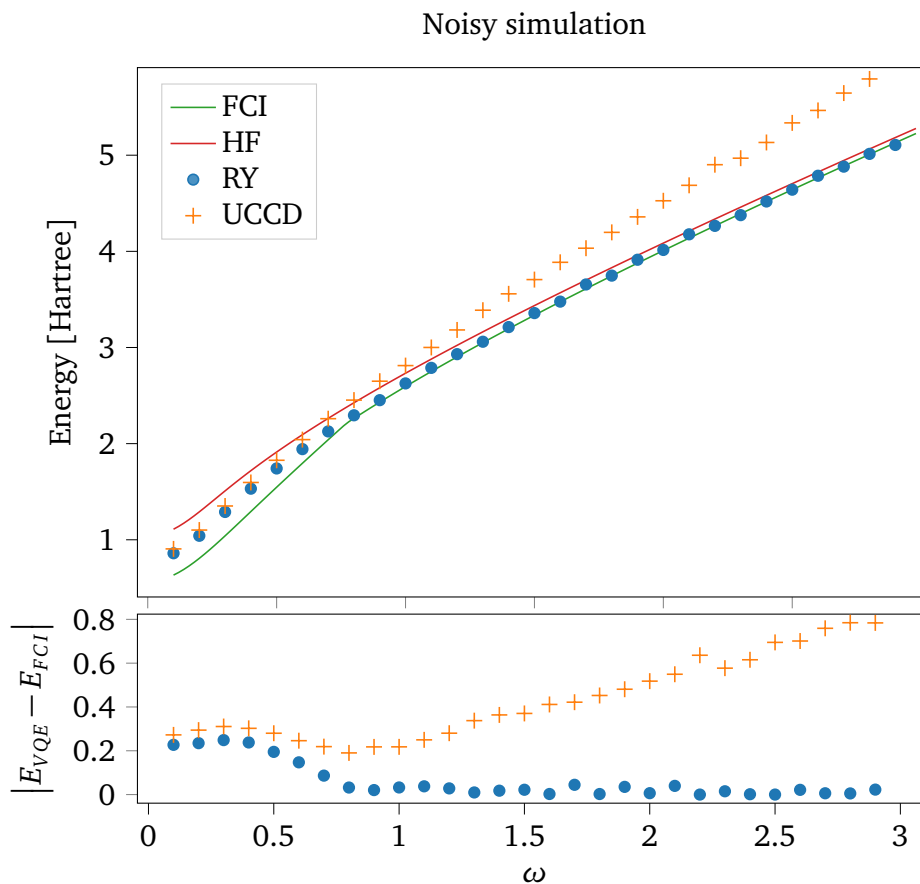
Figure 8.10: Ground state energy of the one-dimensional quantum dot for $\omega = 0.1$ to $\omega = 2.9$, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with UCCD and RY ansätze, using the Cobyla method and $N = 8192$ measurements. Simulated using a noise model from the "London" quantum computer.

From the previous results on the hydrogen molecule, we found that only the RY ansätz was sufficient in order for the quantum computer to give results not dominated by noise. Thus, to execute the circuit for a one-dimensional quantum dot on the same quantum computer we can only use this ansätz. We include the results of using the RY ansätz, and executing it on IBM's "London" quantum computer.
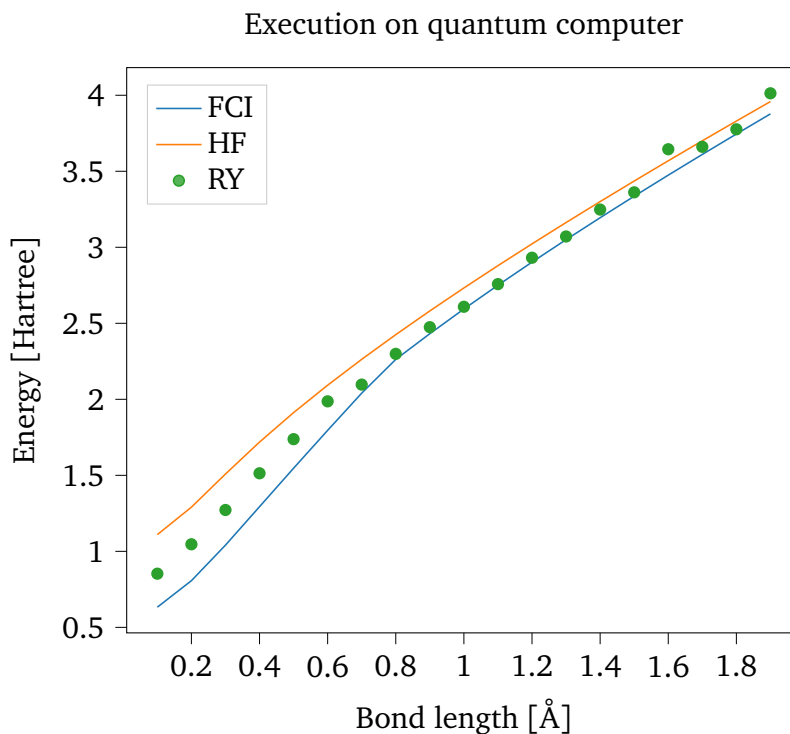
Execution on quantum computer

Figure 8.11: Ground state energy of the one-dimensional quantum dot for $\omega = 0.1$ to $\omega = 2.9$, with the absolute error between the VQE and FCI calculations plotted below. VQE calculations done with the RY ansätz, using the Cobyla method and $N = 4000$ measurements.Executed on IBM's "London" quantum computer.

8.3   Running simulations for larger systems

Lastly we want to see how the VQE algorithm performs for larger systems, where traditional coupled cluster is not equivalent to FCI. We make use of both the RYRZ and UCCD ansätze. We can extend the RYRZ ansätz in figure (5.25) into a state preparation for the pairing model, by including four additional qubits at the end, between each original qubit, and use a CNOT gate from the original qubits to the new qubits. Thus we have an ansätz for four particles and eight orbitals with a pairing constriction, as can be seen in figure (8.12) where the box labeled RYRZ is the same ansatz as for two particels and four orbitals. In figure (8.13) we plot the resulting energies for interaction strengths between $g = 0.2$ and $g = 10$. In addition we include the resulting energies from FCI, HF and CCD. For the transformed Hamiltonian we get 61 Pauli strings. We can first group all qubit-wise commuting (QWC) operators, leading to 12 groups, where the remaining Pauli strings belong to e general commuting (GC) group, giving 13 state preparations per expectation value of the Hamiltonian compared to the naive 61 state preparations. Another and more efficient method is to first group all Pauli strings belonging to a GC group, giving 6 groups, with the remaining Pauli strings belonging to a QWC group. This gives 7 state preparations, a significant reduction, and is the partitioning we use in the results. We see that as the interaction strength increases the correlations get stronger, and the energies calculated by the different methods starts to deviate from one another. Since traditional coupled cluster is not variational, it can find values lower than that of FCI. We see that for strong correlations the CCD energy diverges from the FCI energy. To comparison the UCCD energy remains
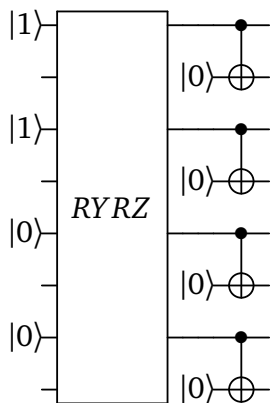
Figure 8.12: RYRZ ansätz for four particles and eight orbitals, using the original RYRZ for two paricles and four orbitals with CNOT gates on the additional qubits, imposing the pairing restriction.
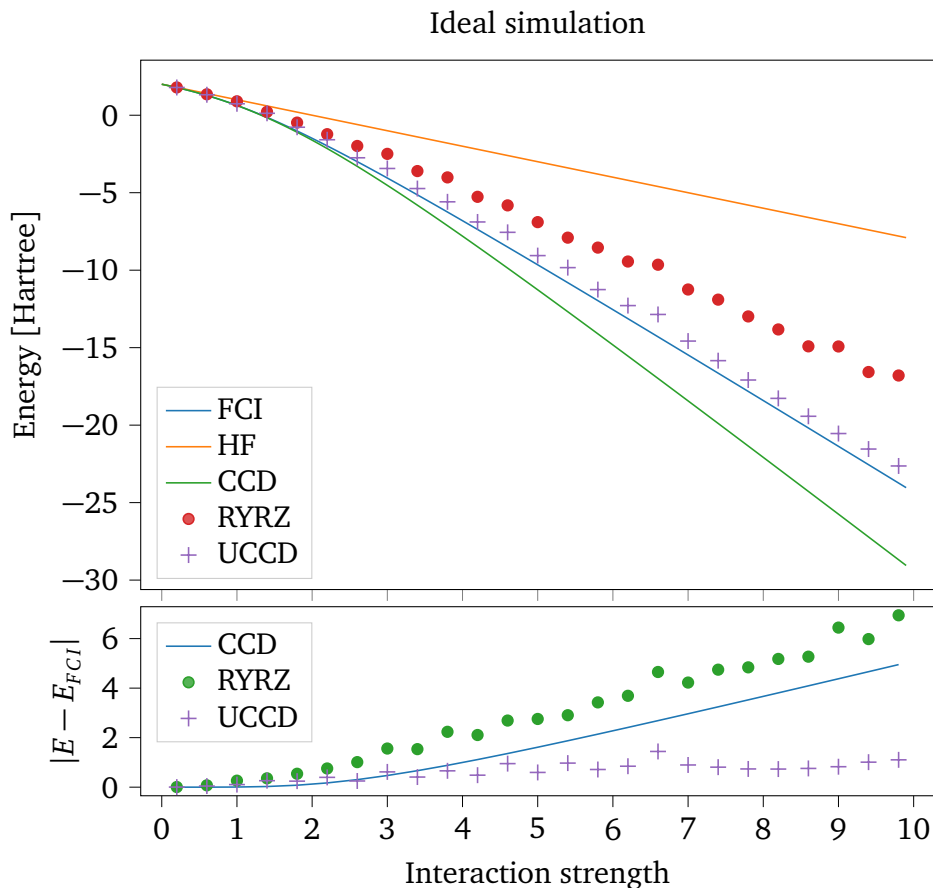

Figure 8.13: VQE simulations of pairing model with four particles and eight orbitals, for an increasing interaction strength g. FCI, CCD and HF energies are included for comparison.

an upper bound the exact energy, and do not diverge in the same manner with a potential trend of a less steep increase in absolute error. The small deviations from this trend at some points probably result from the convergence tolerance being too large, in combination with the errors from the finite set of measurements made. RYRZ fails to perform well when correlation gets high. This minimal ansätz might fail to properly represent the complete search space. The $\hat{R}_y$ and $\hat{R}_z$ rotations along with a CNOT that entangle the qubits from the Hartree-Fock state clearly is not sufficient in order to represent the exact state, when correlations get strong. Additional entanglement between the qubits could solve this.

8.4   Comparing ideal and noisy simulations

In the influence of noise certain Pauli strings might be more important than others. Considering the hydrogen molecule, for shorter bond lengths the Pauli string factors corresponding to one-body operators get much larger compared to the same Pauli strings for longer bond lengths. This is due to the increase in potential energy as the electrons get closer. In addition, the Pauli string composed of identity operators increase rapidly for smaller bond lengths, which we can refer to as the constant term since the factor is only added to the total energy. As a consequence, calculating the correct eigenvalues of the largest Pauli strings is more important for shorter bond lengths.

We see that there is a clear difference in how the noise affects the UCC ansätz compared to RY and RYRZ. Looking at how the state of the qubits is prepared using the different state preparations, we see that RY and RYRZ directly flips the qubits in relation to each other, while UCCD and UCCSD more indirectly flips them adding phases with $\hat{R}_z$ rotations. In table (8.5) and (8.6) we list the probabilities for measuring the different state configurations of a one-dimensional quantum dot after the different state preparations, for $\omega = 0.3$, $\omega = 0.8$ and $\omega = 2.0$, for ideal and noisy simulations respectively. Note that for the noisy simulations the measurement also include noise, so this is not straight forward to discuss. In addition, the noisy and ideal simulation find different parameters, thus the numbers should not be compared to one another. However, it gives an indication of how the state preparations work in presence of noise. We know that RY and UCCD finds the wrong state when ω is small, since they only include the doubles excitations. However, comparing the ideal and noisy simulations for these it is apparent how the noise impact the state preparation and measurements. For the noisy simulations we get contributions from states we ideally should not. Looking at UCCSD and RYRZ, we see that the singles excitation do not contribute with the same amount when noise is present. For $\omega = 0.1$, when the correlations are strong, RYRZ performs better than UCCSD. The state probabilities reflect this as the deviation between the major contributors in noisy and ideal simulations for UCCSD are greater than for RYRZ. As a result we might suspect that the exponential state preparation of UCC fails to properly setup the state with noise present, where additional Trotter steps could be required, however this would greatly increase the circuit depth and thus the noise. To comparison the more explicit bit flips of the RY and RYRZ state preparations performs better, though as we see from the simulations for larger systems, the RYRZ can not properly prepare more complex superpositions and most likely require additional entanlgement.

Table 8.5: Probability for measuring the different state configurations of a one-dimensional quantum dot with two particles and four orbitals after different state preparations in an ideal simulation. Listed for different angular frequencies ω.

Ansätz	ω	$|1100\rangle$	$|0101\rangle$	$|0110\rangle$	$|1001\rangle$	$|1010\rangle$	$|0011\rangle$
RY	0.1	0.6283	0	0	0	0	0.3717
	1.0	0.68359	0	0	0	0	0.31641
	2.5	0.8844	0	0	0	0	0.1156
UCCD	0.1	0.61462	0	0	0	0	0.38538
	1.0	0.7251	0	0	0	0	0.2749
	2.5	0.82324	0	0	0	0	0.17676
RYRZ	0.1	0.04895	0.12463	0.16968	0.25598	0.36365	0.03711
	0.8	0.92773	0.01257	0	0.00232	0	0.05737
	2.0	0.97974	0.00183	0	0.00012	0	0.01831
UCCSD	0.1	0.44763	0.2666	0.00012	0	0	0.28564
	1.0	0.91821	0.00012	0.00452	0	0	0.07715
	2.5	0.96985	0.00391	0.00024	0	0	0.026

Table 8.6: Probability for measuring the different state configurations of a one-dimensional quantum dot with two particles and four orbitals after different state preparations in a noisy simulation. Listed for different angular frequencies ω.

Ansätz	ω	$|1100\rangle$	$|0101\rangle$	$|0110\rangle$	$|1001\rangle$	$|1010\rangle$	$|0011\rangle$
RY	0.1	0.5061	0.00024	0.00073	0.00085	0.00049	0.36963
	1.0	0.72876	0.00024	0.00061	0.00061	0.00085	0.16321
	2.5	0.81958	0.00049	0.00073	0.00037	0.00073	0.07776
UCCD	0.1	0.43115	0.01343	0.00415	0.00342	0.04199	0.23035
	1.0	0.50037	0.01123	0.00293	0.00256	0.03723	0.17151
	2.5	0.58569	0.00928	0.00378	0.00342	0.04504	0.07947
RYRZ	0.1	0.04260	0.01990	0.11572	0.09778	0.57202	0.02246
	0.8	0.87085	0.00061	0.00244	0.0011	0.00366	0.02405
	2.0	0.87573	0.0022	0.0011	0.00354	0.00134	0.02112
UCCSD	0.1	0.43042	0.02356	0.06055	0.01501	0.04675	0.05603
	1.0	0.48279	0.02356	0.02087	0.01697	0.04016	0.03662
	2.5	0.53125	0.03174	0.01306	0.01306	0.03674	0.04736

Chapter 9

Conclusions and future work

9.1 Conclusions

The goal of this thesis has been to implement two promising quantum algorithms for finding the ground state and corresponding energy of quantum many-body systems. Specifically our aim was to study their usage on current and near-term quantum computers, with different strategies for more efficient implementations.

Our implementation includes the quantum phase estimation (QPE) and the variational quantum eigensolver (VQE). For QPE we have shown that near-term applications are unrealistic, because of the dominating noise in current quantum hardware. Within the near future this is not expected to improve sufficiently for any practical uses of the algorithm. This is mainly due to the number of qubits required, as well as the number of controlled operations necessary in order to yield good results. The restricted connectivity between qubits in current quantum computers limits even further the prospects of this algorithm since qubits need to be swapped in order to act with controlled operations, thus increasing the circuit depth. Finding good architectures for the qubits in order to maximize the connectivity is one of the major challenges in developing efficient quantum computers.

For the VQE we have mainly been concerned with how we can implement it on current and near-term quantum computers. We have based our implementation on the connectivity between the qubits on IBM's 5-qubit quantum computers. With that, we have made use of an efficient way of implementing exponential operators by exploiting the Trotter truncation, leading to potential reduction of single qubit gates as well as CNOT gates. While working on our implementation we became aware of a recent more in-depth review [24] utilizing similar insights in potential implementations of exponential operators. Looking further into this would be a good extension. Our implementation has resulted in a significantly reduced circuit for the unitary coupled cluster (UCC) doubles. In addition, we have found an efficient circuit for preparing the measurement of a general commuting group. This group is contained in most Jordan-Wigner transformed Hamiltonians for a system with two particles and four orbitals. The grouping strategy implemented is rather naive,though for the simple systems we have looked at, this gives no immediate drawback. As a consequence, the number of state preparations have been significantly reduced.

Regarding the state preparations used in the VQE algorithm we have, as mentioned, implemented the unitary coupled cluster ansatz. In addition, we have developed two new problem specific state preparations, the RY and RYRZ ansätze, to some

success for systems composed of two particles and four orbitals. We have successfully simulated two simple quantum systems on a real quantum computer, using the RY ansatz, giving desirable results with respect to the expected outcome from noisy simulations of the quantum circuit. Our work has shown that problem specific state preparations might be preferable on noisy quantum hardware.

Through our results we have discussed the potential limitations of the UCC ansatz in the presence of noise. The implicit flipping of qubit states in an exponential operator might suffer more from noisy gate executions compared to state preparations with more explicit qubit state flipping, as RY and RYRZ. However, for ideal simulations the UCC ansatz seems to be superior for more complex quantum state than two particles and four orbitals. In addition we have shown the advantage of UCC compared to traditional coupled cluster. Since the traditional coupled cluster method makes use of a non-Hermitian operator, it is not variational. For systems where exact diagonalization of the Hamiltonian is infeasible, we then cannot generally trust the results of a traditional coupled cluster approach. Clearly this is a major disadvantage. On the other hand UCC is variational, and as we have shown it gives desirable results when a quantum system exhibits stronger correlations. This might indicate that the UCC is a favorable choice when we want to look at system where we do not know the exact solution.

9.2   Future work

The Jordan-Wigner transformation is not the only transformation of creation and annihilation operators to quantum circuits. Another much used transformation is the Bravyi-Kitaev transformation [25, 26]. The Bravyi-Kitaev transformation requires only $\mathcal{O}(log\ n)$ qubit operations to represent a single fermionic operator. In contrast, the Jordan-Wigner transformation requires $\mathcal{O}(n)$ qubit operations. Among other things, it does not have to operate with Pauli-$\hat{Z}$ gates in between qubits corresponding to the creation and annihilation operators, leading to smaller Pauli strings when Hamiltonians of larger systems are transformed. It involves a change of basis, to the so-called parity basis, and is by that a little more complicated to use than the Jordan-Wigner transformation. However, it can lead to what is known as qubit tapering [27]. That is, in some scenarios the transformed Hamiltonian will not act on some qubits at all. This can lead to the removal of these qubits. An example is for the hydrogen molecule, where the number of qubits can be lowered to two qubits instead of four. In terms of groupings of commuting operators, it would be interesting to examine the differences between a Jordan-Wigner transformed Hamiltonian and a Bravyi-Kitaev transformed one.

For the actual strategy of finding commuting operators, our method has not been the most efficient. We naively find and group all Pauli-strings that qubit-wise commute. Furthermore, we only found one general commuting group (though it was the only sensible one). If we were to look at larger systems, other general commuting groups would appear. Recent research has been done on the topic of optimal partitioning of operators, see e.g. [14, 28, 29]. Looking more into this would be an exciting topic.

Implementing the UCC or other state preparations for a larger number of qubits, requires quantum hardware with sophisticated connectivity between qubits. Current quantum computers have limited connectivity and thus qubits need to be swapped

in order to act with control operations on arbitrary pairs of qubits. Implementing so-called fermionic swap networks [30] into our VQE class would serve as a good extension when larger systems are to be simulated, such as on IBM's 16 qubit-computer. Especially as the number of qubits on current quantum computer increase drastically, as we mentioned in the introduction.

Part IV

Appendix

Appendix A

Quantum Mechanics

A.1 Wick's theorem

We assume $ABCD\ldots$ to be a string of creation and annihilation operators. Writing this in normal-ordered form puts all annihilation operators to the right of all creation operators. That is

$$\{ABCD\ldots\} = (-1)^p [\text{creation operators}] \cdot [\text{annihilation operators}] \tag{A.1}$$

with p being the number of permutations needed to put the string into normal-ordered form. Next, we define the contraction between two operators as its expectation value on the vacuum state

$$\overset{\sqcap}{AB} = \langle| AB |\rangle \tag{A.2}$$

Wick's theorem states that the operator string $ABCD\ldots$ can be written as the sum of normal-ordered products for all possible contractions.

$$\begin{aligned}
ABCD\cdots = &\{ABCD\ldots\} \\
&+ \sum_{\text{singles}} \{\overset{\sqcap}{AB}CD\ldots\} \\
&+ \sum_{\text{doubles}} \{\overset{\sqcap\sqcap}{ABCD}\ldots\} \\
&+ \ldots
\end{aligned} \tag{A.3}$$

Appendix B

Quantum computing

B.1 Jordan-Wigner transformation

B.1.1 One-body operators

Next we will show the transformation of the one-body operators in equation (2.73), given by

$$\sum_{pq} \hat{a}_p^\dagger \hat{a}_q. \tag{B.1}$$

When $p = q$ all the $\hat{Z}$ operators will cancel each other out since $\hat{Z}^2 = \hat{\mathbb{1}}$, and we are left with

$$\hat{a}_p^\dagger \hat{a}_p \rightarrow \left(\prod_k^{p-1} \hat{Z}_k\right)\sigma_+^p\left(\prod_k^{p-1} \hat{Z}_k\right)\sigma_-^p = \frac{1}{2}(\hat{\mathbb{1}}_p - \hat{Z}_p), \tag{B.2}$$

note that the product $\sigma_+\sigma_- = \frac{1}{2}(\hat{\mathbb{1}} - \hat{Z})$ since σ_- first act on the qubit and then σ_+.

For all other indices $p \neq q$ the $\hat{Z}$ operators will remain for the qubits between p and q, including p if $p < q$. Since the raising and lowering operators act on different qubits we make use of the transformations in equation (3.31) and (3.32)

$$\hat{a}_p^\dagger \hat{a}_q \rightarrow \left(\prod_k^{p-1} \hat{Z}_k\right)\sigma_+^p\left(\prod_k^{q-1} \hat{Z}_k\right)\sigma_-^q = \frac{1}{4}\left(\prod_{k=p}^{q-1} \hat{Z}_k\right)\left(\hat{X}_p - i\hat{Y}_p\right)\left(\hat{X}_q + i\hat{Y}_q\right)$$

$$= \frac{1}{4}\left(\prod_{k=p}^{q-1} \hat{Z}_k\right)\left(\hat{X}_p\hat{X}_q - i\hat{X}_p\hat{Y}_q - i\hat{Y}_p\hat{X}_q + \hat{Y}_p\hat{Y}_q\right). \tag{B.3}$$

B.1.2 Two-body operators

Furthermore we will show the same for the two-body operators in equation (2.73)

$$\sum_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_t. \tag{B.4}$$

For a fermionic system we have that $p < q$ and $r < s$ for anti-symmetric matrix elements. Again, this leaves us with two general scenarios; where $p = s$ and $q = r$, and all other indices. First when $p = s$ and $q = r$, all $\hat{Z}$ operators cancel out since they act

an even amount of times on each qubit

$$\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_t \rightarrow \left(\prod_k^{p-1} \hat{Z}_k\right) \sigma_+^p \left(\prod_k^{q-1} \hat{Z}_k\right) \sigma_+^q \left(\prod_k^{p-1} \hat{Z}_k\right) \sigma_-^p \left(\prod_k^{q-1} \hat{Z}_k\right) \sigma_-^q$$
$$= \frac{1}{4}(\hat{1}_p - \hat{Z}_p)(\hat{1}_q - \hat{Z}_q) = \frac{1}{4}\left(\hat{1}_p \hat{1}_q - \hat{1}_p \hat{Z}_q - \hat{Z}_p \hat{1}_q + \hat{Z}_p \hat{Z}_q\right). \tag{B.5}$$

B.2 Derivation of the UCC equations

B.2.1 Singles

Using the Jordan-Wigner transformation we want to transform the following equation into pauli strings

$$\sum_{i<a} t_i^a (a_a^\dagger a_i - a_i^\dagger a_a). \tag{B.6}$$

Since $a_i^\dagger a_a$ is the hermitian conjugate of $a_a^\dagger a_i$ we only need to find an expression for one of the terms. The difference between the two is thus

$$(a + ib) - (a - ib) = 2ib,$$

where a is the real part and b is the imaginary part. The JW transformation gives

$$a_a^\dagger a_i = 1^{\otimes i-1} \otimes \hat{Z}_i \sigma_i^- \otimes \hat{Z}_{i+1} \otimes \cdots \otimes \hat{Z}_{a-1} \otimes \sigma_a^+ \otimes 1^{\otimes n-a}, \tag{B.7}$$

Moreover we have the following relation

$$\hat{Z}\sigma^- = -\sigma^- = -\frac{1}{2}(\hat{X} - i\hat{Y}). \tag{B.8}$$

Combining the two equations yield

$$a_a^\dagger a_i = -\frac{1}{4}\left(1^{\otimes i-1} \otimes (\hat{X}_i - i\hat{Y}_i) \otimes \hat{Z}_{i+1} \otimes \cdots \otimes \hat{Z}_{a-1} \otimes (\hat{X}_a + i\hat{Y}_a) \otimes 1^{\otimes n-a}\right), \tag{B.9}$$

Disregarding the identity operators and adopting the pauli string notation gives

$$a_a^\dagger a_i = -\frac{1}{4}\left(\hat{X}_i - i\hat{Y}_i\right) \otimes \left(\hat{X}_a + i\hat{Y}_a\right) \left[\bigotimes_{k=i+1}^{a-1} \hat{Z}_k\right]$$
$$= -\frac{1}{4}\left(\mathscr{P}_{xx}^{ia} + i\mathscr{P}_{xy}^{ia} - i\mathscr{P}_{yx}^{ia} + \mathscr{P}_{yy}^{ia}\right) \left[\bigotimes_{k=i+1}^{a-1} \hat{Z}_k\right], \tag{B.10}$$

We have seen that only the imaginary parts of equation (B.10) giving the final equation

$$t_i^a (a_a^\dagger a_i - a_i^\dagger a_a) \equiv \frac{it_i^a}{2}\left(\mathscr{P}_{yx}^{ia} - \mathscr{P}_{xy}^{ia}\right) \left[\bigotimes_{k=i+1}^{a-1} \hat{Z}_k\right] \tag{B.11}$$

B.2.2   Doubles

The doubles equation reads

$$\sum_{ijab} t_{ij}^{ab}(a_a^\dagger a_b^\dagger a_j a_i - a_i^\dagger a_j^\dagger a_b a_a) \tag{B.12}$$

First we can divide the sum into four parts:

$$a < b \text{ and } i < j,$$
$$a > b \text{ and } i < j,$$
$$a < b \text{ and } i > j,$$
$$a > b \text{ and } i > j.$$

Together with the antisymmetry of the cluster amplitudes we can restrict the sums to $a < b$ and $i < j$ giving

$$\sum_{ijab} t_{ij}^{ab} = 4 \sum_{i<j} \sum_{a<b} t_{ij}^{ab}. \tag{B.13}$$

Next the Jordan-Wigner transformation gives

$$a_a^\dagger a_b^\dagger a_j a_i = \mathbb{1}^{\otimes i-1} \otimes \hat{Z}_i \sigma_i^- \otimes \hat{Z}^{\otimes j-i-1} \otimes \sigma_j^- \otimes \hat{Z}^{\otimes a-j-1}$$
$$\otimes \sigma_a^+ \hat{Z}_a \otimes \hat{Z}^{\otimes b-a-1} \otimes \sigma_b^+ \otimes \mathbb{1}^{\otimes n-b-1}. \tag{B.14}$$

Including equation (B.8) we have the relation

$$\sigma^+ \hat{Z} = -\sigma^+ = -\frac{1}{2}(\hat{X} + i\hat{Y}). \tag{B.15}$$

Combining these with equation (B.14) gives

$$a_a^\dagger a_b^\dagger a_j a_i = \frac{1}{16}\Big(\mathbb{1}^{\otimes i-1} \otimes (\hat{X}_i - i\hat{Y}_i) \otimes \hat{Z}^{\otimes j-i-1} \otimes (\hat{X}_j - i\hat{Y}_j) \otimes \hat{Z}^{\otimes a-j-1}$$
$$\otimes (\hat{X}_a + i\hat{Y}_a) \otimes \hat{Z}_{\otimes b-a-1} \otimes (\hat{X}_b + i\hat{Y}_b) \otimes \mathbb{1}^{\otimes n-b-1}\Big). \tag{B.16}$$

Again we disregard the identity operators and make use of the Pauli string notation

$$a_a^\dagger a_b^\dagger a_j a_i = \frac{1}{16}(\hat{X}_i - i\hat{Y}_i)(\hat{X}_j - i\hat{Y}_j)(\hat{X}_a - i\hat{Y}_a)(\hat{X}_b - i\hat{Y}_b)\left[\bigotimes_{k=i+1}^{j-1} \hat{Z}_k\right]\left[\bigotimes_{k=a+1}^{b-1} \hat{Z}_k\right] \tag{B.17}$$

As with the singles terms we only get the imaginary part of equation (B.17), and so we are left with the final equation

$$t_{ij}^{ab}(a_a^\dagger a_b^\dagger a_j a_i - a_i^\dagger a_j^\dagger a_b a_a) \equiv \frac{i t_{ij}^{ab}}{8}\left[\bigotimes_{k=i+1}^{j-1} \hat{Z}_k\right]\left[\bigotimes_{k=a+1}^{b-1} \hat{Z}_k\right]$$
$$\Big(\mathscr{P}_{xxxy}^{ijab} + \mathscr{P}_{xxyx}^{ijab} - \mathscr{P}_{xyxx}^{ijab} + \mathscr{P}_{xyyy}^{ijab}$$
$$- \mathscr{P}_{yxxx}^{ijab} + \mathscr{P}_{yxyy}^{ijab} - \mathscr{P}_{yyxy}^{ijab} - \mathscr{P}_{yyyx}^{ijab}\Big). \tag{B.18}$$

B.3   Full configuration interaction for pairing model

The pairing Hamiltonian yields

$$\hat{H} = \hat{H}_0 + \hat{H}_1 = \xi \sum_{p\sigma}(p-1)\hat{n}_p - \frac{1}{2}g \sum_{pq} P_p^+ P_q^-. \tag{B.19}$$

Since we soon will make use of the Kronecker delta, we have replaced the notation of the single particle spacing with ξ instead of δ as in section (5.1) and our implemented code, to avoid any confusion.

Since we are only dealing with pairs of particles occupying different energy levels, we can turn to occupation representation to calculate the matrix with Slater determinants as basis states. The basis will be composed of all possible variations of particle-hole excitations, where the total number of basis states then becomes the binomial coefficient

$$\binom{p}{n} = \frac{p!}{n!(p-n)!}, \tag{B.20}$$

where p is the number of hole states and n is the number of particle pairs. The pair creation and annihilation operators ensure that no pairs are broken, and can be used to define basis states

$$\left|\Phi_{ij\dots}^{ab\dots}\right\rangle = P_a^+ P_b^+ \dots P_k^- P_j^- P_i^- |\Phi_0\rangle, \tag{B.21}$$

with the ground state defined as all pair creations acting on the vacuum state $|\rangle$ below the Fermi level

$$|\Phi_0\rangle = \left(\prod_{i\leq F} P_i^\dagger\right)|\rangle. \tag{B.22}$$

The Hamiltonian matrix elements are given by

$$H_{ij} = \langle\phi_i|\hat{H}\left|\phi_j\right\rangle = \langle\phi_i|\hat{H}_0\left|\phi_j\right\rangle + \langle\phi_i|\hat{H}_1\left|\phi_j\right\rangle, \tag{B.23}$$

where $|\phi_i\rangle$ is a basis state. We begin with the one-body part, given as

$$\langle\phi_i|\hat{H}_0\left|\phi_j\right\rangle = \xi \sum_{p\sigma}(p-1)\langle\phi_i|\hat{n}_{p\sigma}\left|\phi_j\right\rangle, \tag{B.24}$$

where $\langle\phi_i|\hat{n}_{p\sigma}\left|\phi_j\right\rangle = \delta_{ij}$. We only get non-zero contributions from the number operator when an orbital is occupied by a particle. The sum over σ can be replaced with a factor of 2 since the contributions are the same for opposite-spin particles in the same spatial orbitals. This gives

$$\langle\phi_i|\hat{H}_0\left|\phi_i\right\rangle = 2\xi \sum_{p}(p-1)\delta_{p\in\phi_i} \tag{B.25}$$

where the Kronecker delta $\delta_{p\in\phi_i}$ is zero unless state p is occupied in ϕ_i.

For the two-body part, we only get a contribution if there is a maximum of one

pair different in the two Slater determinants. The diagonal elements become

$$
\begin{aligned}
\left\langle \Phi_{ij\dots}^{ab\dots} \middle| \hat{H}_1 \middle| \Phi_{ij\dots}^{ab\dots} \right\rangle &= -\frac{1}{2} g \sum_{pq} \left\langle \Phi_{ij\dots}^{ab\dots} \middle| P_p^+ P_q^- \middle| \Phi_{ij\dots}^{ab\dots} \right\rangle \\
&= -\frac{1}{2} g \left[\sum_{pq>F} \delta_{pq} \sum_{c>F} \delta_{qc} + \sum_{pq<F} \delta_{pq} \sum_{k<F}^{F} \delta_{kq} \right] \\
&= -\frac{1}{2} g \left[\sum_{a>F} 1 + \sum_{k<F} 1 \right] = -\frac{1}{4} g \cdot n_p
\end{aligned}
\tag{B.26}
$$

where $c \in \Phi_{ij\dots}^{ab\dots}$ and $k \in \Phi_{ij\dots}^{ab\dots}$. The off-diagonal elements become

$$
\left\langle \Phi_{ij\dots}^{ab\dots} \middle| \hat{H}_1 \middle| \Phi_{ij\dots}^{ac\dots} \right\rangle = -\frac{1}{2} g \sum_{pq} \delta_{pb} \delta_{qc}
\tag{B.27}
$$

for one different excited pair, and

$$
\left\langle \Phi_{ij\dots}^{ab\dots} \middle| \hat{H}_1 \middle| \Phi_{ik\dots}^{ab\dots} \right\rangle = -\frac{1}{2} g \sum_{pq} \delta_{pj} \delta_{qk}
\tag{B.28}
$$

for one pair different below the fermi-level. The resulting matrix can be diagonalized using a standard eigenvalue solver such as, e.g. numpy's `eigh` function.

B.4   Circuit contruction for simultaneous eigenstates

We want to find the circuit that transform the computational basis into the simultaneous eigenstates of the eight Pauli-strings

$$
\mathscr{P}_{xxxx}^{pqrs}, \quad \mathscr{P}_{xxyy}^{pqrs}, \quad \mathscr{P}_{xyxy}^{pqrs}, \quad \mathscr{P}_{yxxy}^{pqrs}, \quad \mathscr{P}_{yyyy}^{pqrs}, \quad \mathscr{P}_{yyxx}^{pqrs}, \quad \mathscr{P}_{yxyx}^{pqrs}, \quad \mathscr{P}_{xyyx}^{pqrs}.
\tag{B.29}
$$

Following the stabilizer formalism introduced in section (5.5.2) we use the first four Pauli-strings above as a basis, resulting in the stabilizer matrix

$$
\begin{array}{c}
|p\rangle \;\text{—} \\
|q\rangle \;\text{—} \\
|r\rangle \;\text{—} \\
|s\rangle \;\text{—}
\end{array}
\quad \rightarrow \quad
\left(
\begin{array}{cccc}
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 \\
\hline
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1
\end{array}
\right).
\tag{B.30}
$$

We separate the matrix into the the top (Z) and bottom (X) square matrices. Neither matrices have full rank, so the first step is to give the X matrix full rank. The first three rows of the Z matrix along with the last row in the X matrix can provide a full rank matrix, thus we make use of Hadamard gates to swap rows from the first three rows of the Z matrix with the first three rows of the X matrix. The stabilizer matrix then

becomes

$$
\begin{array}{c}
|p\rangle - \boxed{H} - \\[4pt]
|q\rangle - \boxed{H} - \\[4pt]
|r\rangle - \boxed{H} - \\[4pt]
|s\rangle \rule[0.5ex]{2em}{0.4pt}
\end{array}
\quad \rightarrow \quad
\left(
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 \\
\hline
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1
\end{array}
\right).
\tag{B.31}
$$

The X matrix should end up as a diagonal matrix, so we can swap the first and last qubit, along with the second and third qubit, which gives

$$
\rightarrow
\left(
\begin{array}{cccc}
0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\hline
1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}
\right).
\tag{B.32}
$$

Next we need to remove the 1-entries in the Z matrix, where we make use of CNOT gates for modulo 2 addition between two rows in both the Z and X matrix. CNOT gates targeted on the first qubit conditional on the other latter qubits give

$$
\rightarrow
\left(
\begin{array}{cccc}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}
\right).
\tag{B.33}
$$

To remove the remaining, non-diagonal, 1-entries in the Z matrix we make use of the CZ gates on targeted on the first qubit conditional on all latter qubits, giving the stabilizer matrix

$$
\rightarrow
\left(
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}
\right).
\tag{B.34}
$$

Finally we only need to swap the X matrix with the Z matrix, by applying Hadamard

gates on all qubits

$$
\rightarrow
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}.
\tag{B.35}
$$
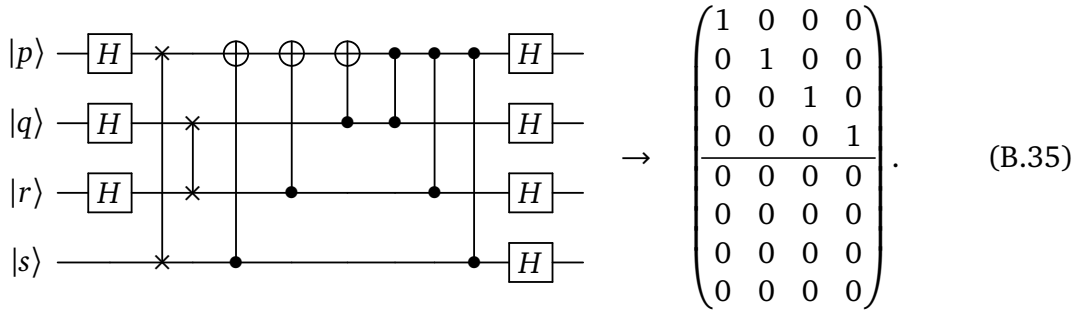
We then have a circuit for the simultaneous eigenstates of the Pauli-strings in equation (B.29). A measurement of the first four Pauli-strings is after the circuit the same as measuring the following Pauli-strings

$$
\begin{aligned}
\mathscr{P}^{pqrs}_{xxxx} &\rightarrow \mathscr{P}^{p}_{z}, \\
\mathscr{P}^{pqrs}_{xxyy} &\rightarrow \mathscr{P}^{q}_{z}, \\
\mathscr{P}^{pqrs}_{xyxy} &\rightarrow \mathscr{P}^{r}_{z}, \\
\mathscr{P}^{pqrs}_{yxxy} &\rightarrow \mathscr{P}^{s}_{z},
\end{aligned}
$$

the remaining Pauli-strings will depend on the parity of the different qubits. Before we find the eigenvalues of these Pauli-strings, we see that the SWAP gates are not necessary, giving the reduced circuit in figure (B.1).
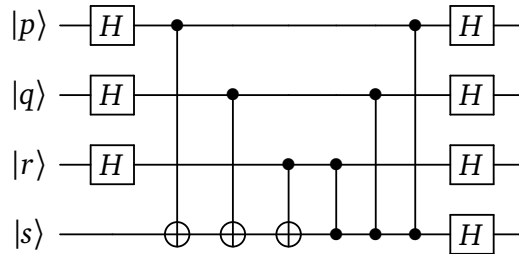
Figure B.1: Final circuit for measurement preparation of the simultaneous eigenstates of the Pauli-strings in equation (B.29).

This result in a swap of the Pauli-string measurements above

$$
\begin{aligned}
\mathscr{P}^{pqrs}_{xxxx} &\rightarrow \mathscr{P}^{s}_{z}, \\
\mathscr{P}^{pqrs}_{xxyy} &\rightarrow \mathscr{P}^{r}_{z}, \\
\mathscr{P}^{pqrs}_{xyxy} &\rightarrow \mathscr{P}^{q}_{z}, \\
\mathscr{P}^{pqrs}_{yxxy} &\rightarrow \mathscr{P}^{p}_{z}.
\end{aligned}
$$

The eigenvalues of the remaining Pauli-strings is listed in section (5.5.2).

Bibliography

[1] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Mineola: Dover Publications, Inc., first ed., 1996.

[2] T. D. Crawford and H. F. Schaefer III, *An Introduction to Coupled Cluster Theory for Computational Chemists*, pp. 33–136. John Wiley Sons, Ltd, 2007.

[3] M. R. Hoffmann and J. Simons, "A unitary multiconfigurational coupled-cluster method: Theory and applications," *The Journal of Chemical Physics*, vol. 88, no. 2, pp. 993–1002, 1988.

[4] R. P. Feynman, "Simulating physics with computers," *International journal of theoretical physics*, vol. 21, no. 6/7, pp. 467–488, 1982.

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, p. 1484–1509, Oct 1997.

[6] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, no. 1, p. 4213, 2014.

[7] J. Preskill, "Quantum computing and the entanglement frontier," 2012.

[8] F. Arute, K. Arya, R. Babbush, D. Bacon, J. Bardin, R. Barends, R. Biswas, S. Boixo, F. Brandao, D. Buell, B. Burkett, Y. Chen, J. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. M. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. Harrigan, M. Hartmann, A. Ho, M. R. Hoffmann, T. Huang, T. Humble, S. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, p. 505–510, 2019.

[9] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic Structure Theory*. Chichester: John Wiley & Sons, LTD, 2000.

[10] F. Feng, Q. Zhang, and H. Liu, "A recurrent neural network for extreme eigenvalue problem," in *Advances in Intelligent Computing* (D.-S. Huang, X.-P. Zhang, and G.-B. Huang, eds.), (Berlin, Heidelberg), pp. 787–796, Springer Berlin Heidelberg, 2005.

[11] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[12] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, p. 023023, feb 2016.

[13] J. Romero, R. Babbush, J. Mcclean, C. Hempel, P. Love, and A. Aspuru-Guzik, "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz," *Quantum Science and Technology*, vol. 4, jan 2017.

[14] P. Gokhale, O. Angiuli, Y. Ding, K. Gui, T. Tomesh, M. Suchara, M. Martonosi, and F. T. Chong, "Minimizing state preparations in variational quantum eigensolver by partitioning into commuting families," *arXiv: Quantum Physics*, 2019.

[15] Y. Nam, J.-S. Chen, N. C. Pisenti, K. Wright, C. Delaney, D. Maslov, K. R. Brown, S. Allen, J. M. Amini, J. Apisdorf, K. M. Beck, A. Blinov, V. Chaplin, M. Chmielewski, C. Collins, S. Debnath, K. M. Hudek, A. M. Ducore, M. Keesan, S. M. Kreikemeier, J. Mizrahi, P. Solomon, M. Williams, J. D. Wong-Campos, D. Moehring, C. Monroe, and J. Kim, "Ground-state energy estimation of the water molecule on a trapped-ion quantum computer," *npj Quantum Information*, vol. 6, no. 1, p. 33, 2020.

[16] F. Gao and L. Han, "Implementing the nelder-mead simplex algorithm with adaptive parameters," *Computational Optimization and Applications*, vol. 51, no. 1, pp. 259–277, 2012.

[17] M. Powell, "A view of algorithms for optimization without derivatives," *Mathematics TODAY*, vol. 43, 01 2007.

[18] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The Computer Journal*, vol. 7, pp. 155–162, 01 1964.

[19] J. C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 817–823, 1998.

[20] A. McCaskey, Z. Parks, J. Jakowski, S. Moore, T. Morris, T. Humble, and R. Pooser, "Quantum chemistry as a benchmark for near-term quantum computers," *npj Quantum Information*, vol. 5, 12 2019.

[21] S. Schøyen, "Real-time quantum many-body dynamics," Master's thesis, University of Oslo, 2019.

[22] S. Winther-Larsen, "Aquaduct ab initio quantum dynamics using coupled cluster in time," Master's thesis, University of Oslo, 2019.

[23] D. J. Griffiths, *Introduction to Quantum Mechanics (2nd Edition)*, pp. 51–57. Pearson Prentice Hall, 2nd ed., Apr. 2004.

[24] K. Gui, T. Tomesh, P. Gokhale, Y. Shi, F. T. Chong, M. Martonosi, and M. Suchara, "Term grouping and travelling salesperson for digital quantum simulation," *arXiv: Quantum Physics*, 2020.

[25] S. B. Bravyi and A. Y. Kitaev, "Fermionic quantum computation," *Annals of Physics*, vol. 298, p. 210–226, May 2002.

[26] A. Tranter, S. Sofia, J. Seeley, M. Kaicher, J. McClean, R. Babbush, P. V. Coveney, F. Mintert, F. Wilhelm, and P. J. Love, "The bravyi–kitaev transformation: Properties and applications," *International Journal of Quantum Chemistry*, vol. 115, no. 19, pp. 1431–1441, 2015.

[27] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme, "Tapering off qubits to simulate fermionic hamiltonians," *arXiv: Quantum Physics*, 2017.

[28] O. Crawford, B. van Straaten, D. Wang, T. Parks, E. Campbell, and S. Brierley, "Efficient quantum measurement of pauli operators in the presence of finite sampling error," *arXiv: Quantum Physics*, 2019.

[29] V. Verteletskyi, T.-C. Yen, and A. F. Izmaylov, "Measurement optimization in the variational quantum eigensolver using a minimum clique cover," *The Journal of Chemical Physics*, vol. 152, p. 124114, Mar 2020.

[30] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, "Low-depth quantum simulation of materials," *Physical Review X*, vol. 8, p. 011044, Mar 2018.