Quantum Computing

Heine Aabø og Stian Bilek

June 2019

Abstract

Our aim in this study was to implement a quantum algorithm for estimating the eigenvalues of a pairing hamiltonian and comparing the found eigenvalues with the benchmark provided by using Full Configuration Interaction theory (FCI). We found that the results provided by FCI was within two standard deviations of the eigenvalues found with the quantum algorithm. As can be seen in table 1, we got the eigenvalues -0.6 ± 0.1 and 1.61 ± 0.06 for one pair, and this result is consistent with the FCI eigenvalues -0.618 and 1.618. For two pairs, the quantum algorithm yielded 1.0 ± 0.2 which also is consistent with the FCI eigenvalue of 1.0.

1 Introduction

Feynman noted in 1982 that calculating properties of an arbitrary quantum system on a classical device is a seemingly very inefficient thing to do [5]. It is a problem which scales exponentially with the number of particles in the model being simulated, meaning that for complex molecules like for example the caffeine molecule, current modelling approaches is at their limits. He suggested that a quantum device might be able to calculate such properties efficiently, meaning that the problem instead scales polynomially with the number of particles. In 1985, David Deutch provided a description of such a quantum device, namely a universal quantum computer [3]. Moving forward to 1996, American physicist Seth Lloyd indeed showed that such a computer could simulate quantum systems efficiently [6].

Today, Quantum computers have moved beyond pen and paper and there are several big companies working on making these devices a reality, including IBM and Google. IBM even has a quantum computer consisting of 20 qubits (the quantum equivalent of classical bits), which can be accessed and programmed over the cloud. There are, however, still a lot of hurdles that has to be overcome in order to make these devices useful in practice.

In this study, our aim is to implement a quantum algorithm, described in [8], to approximate the eigenvalues of the pairing hamiltonian. We will make use of IBM's python library "qiskit" to write the code in python and

run the code by using IBM's quantum computer simulator. The found eigenvalues will be compared with the eigenvalues calculated with Full Configuration Interaction theory as these eigenvalues will be exact within the subspace spanned by our basis states. The theory section should contain the basic building blocks for understanding the quantum algorithms we utilized and also the theory behind FCI. [7] is a recommended read for a wider understanding of the quantum computing concepts. The code used to produce our results can be found on our github page:
https://github.com/stiandb/QuantumComputing

Contents

2 Theory

2.1 Configuration Interaction[1]

We want to solve the time-independent Schröedinger Equation

$$\hat{H}\left|\Phi_k\right\rangle = (\hat{H}_0 + \hat{H}_1)\left|\Phi_k\right\rangle = \epsilon_k \left|\Phi_k\right\rangle \tag{1}$$

The problem is that we do not know the solution when the particle-particle interaction is present. We do know however, that an arbitrary antisymmetric N-particle wave function can be written as a linear combination of Slater determinants. If we choose the 2K eigenfunctions ψ_k of $\hat{H}_0$ as our basis, the solutions to equation 1 are then given by

$$\left|\Phi_k\right\rangle = c_0^{(k)}\left|\Psi_0\right\rangle + \sum_{ia} c_i^{a(k)}\left|\Psi_i^a\right\rangle + \sum_{i<j,a<b} c_{ij}^{ab(k)}\left|\Psi_{ij}^{ab}\right\rangle + \cdots \tag{2}$$

We can write the Hamiltonian in the $\left|\Psi_i\right\rangle$ basis by utilizing the fact that $\sum_i \left|\Psi_i\right\rangle\left\langle\Psi_i\right| = I$:

$$\hat{H} = \sum_{ij}\left|\Psi_i\right\rangle\left\langle\Psi_i\right|\hat{H}\left|\Psi_j\right\rangle\left\langle\Psi_j\right| \tag{3}$$

By multiplying both sides with $\left\langle\Psi_l\right|$ and using equation 1 and 2, we see that

$$\left\langle\Psi_l\right|\hat{H}\left|\Phi_k\right\rangle = \sum_j \left\langle\Psi_l\right|\hat{H}\left|\Psi_j\right\rangle c_j = \epsilon_k c_l \tag{4}$$

This can be written in matrix notation as

$$\begin{bmatrix} \left\langle\Psi_0\right|\hat{H}\left|\Psi_0\right\rangle & \left\langle\Psi_0\right|\hat{H}\left|\Psi_1\right\rangle & \dots & \left\langle\Psi_0\right|\hat{H}\left|\Psi_{\binom{2K}{N}}\right\rangle \\ \left\langle\Psi_1\right|\hat{H}\left|\Psi_0\right\rangle & \left\langle\Psi_1\right|\hat{H}\left|\Psi_1\right\rangle & \dots & \left\langle\Psi_1\right|\hat{H}\left|\Psi_{\binom{2K}{N}}\right\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \left\langle\Psi_{\binom{2K}{N}}\right|\hat{H}\left|\Psi_0\right\rangle & \left\langle\Psi_{\binom{2K}{N}}\right|\hat{H}\left|\Psi_1\right\rangle & \dots & \left\langle\Psi_{\binom{2K}{N}}\right|\hat{H}\left|\Psi_{\binom{2K}{N}}\right\rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{\binom{2K}{N}} \end{bmatrix} = \epsilon_k \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{\binom{2K}{N}} \end{bmatrix} \tag{5}$$

Thus we can in principle solve equation 1 exactly by finding the eigenfunctions and eigenvalues of the above matrix. In practice however, we require an infinite number of single-particle wave functions to form a complete set, which means that the $\binom{2K}{N}$ Slater determinants will not form a basis for our N-particle functions. Nevertheless, the solutions will be exact within the N-particle subspace spanned by our Slater determinants.

2.2 Paring model

A common way to model the nuclear structure is to assume that nucleons form pairs. This is supported by the fact that even-even nuclei are slightly more bound than odd-even and odd-odd nuclei[1]. More energy is needed to excite the nuclei since a pair has to be broken.

[1] this section is copy-pasted from an earlier project, found at https://github.com/stiandb/ManyBodyFys4480/tree/master/PDF

Furthermore, the BCS theory of superconductivity assumes simultaneous creation and condensation of electron pairs, or Cooper pairs, due to electron-phonon interactions[2]. These realizations motivates the introduction of pairing interactions when studying certain many-body systems and their properties.

One such system is the simple pairing model of an ideal fermionic system where we assume no broken pairs, that is seniority quantum number $\mathcal{S} = 0$, [4] and spin degeneracy 2.

2.2.1 Hamiltonian

The pairing model Hamiltonian can be written on the form

$$\begin{aligned}\hat{H} =& \hat{H}_0 + \hat{H}_1 \\ =& \sum_p \epsilon_p a_p^\dagger a_p - \frac{1}{2} g \sum_{pq} a_{p+}^\dagger a_{p-}^\dagger a_{q-} a_{q+}\end{aligned} \tag{6}$$

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 ξ we can write $\epsilon_p = (p-1)\xi$, and introduce $\sigma = \pm$ representing the possible spin values. Next we recognize the number operator $\hat{n}_p = a_p^\dagger a_p$. The interaction part $\hat{H}_1$ 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:

$$\hat{H} = \xi \sum_{p\sigma} (p-1)\hat{n}_p - \frac{1}{2} g \sum_{pq} P_p^+ P_q^- \tag{7}$$

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. The total number of basis states then becomes the binomial coefficient

$$\binom{p}{n} = \frac{p!}{n!(p-n)!} \tag{8}$$

where p is the number of hole states and n is the number of particle pairs. Since the particle pairs can not be broken we can define a general basis states in terms of pair creation and annihilation operators

$$\left| \Phi_{ij\ldots}^{ab\ldots} \right\rangle = P_a^+ P_b^+ \ldots P_k^- P_j^- P_i^- \left| \Phi_0 \right\rangle \tag{9}$$

with the ground state defined as

$$\left| \Phi_0 \right\rangle = \left(\prod_{i \leq F} P_i^\dagger \right) \left| 0 \right\rangle \tag{10}$$

acting on the vacuum state $|0\rangle$, where F denotes the Fermi level without degeneracy as we are exciting pairs. The matrix elements of eq:Hamiltonian is given by

$$H_{ij} = \langle\phi_i|\,\hat{H}\,|\phi_j\rangle = \langle\phi_i|\,\hat{H}_0\,|\phi_j\rangle + \langle\phi_i|\,\hat{H}_1\,|\phi_j\rangle \tag{11}$$

for the any basis state $|\phi_i\rangle$. Lets first deal with the one-body part, given as

$$\langle\phi_i|\,\hat{H}_0\,|\phi_j\rangle = \xi\sum_{p\sigma}(p-1)\,\langle\phi_i|\,\hat{n}_{p\sigma}\,|\phi_j\rangle \tag{12}$$

where $\langle\phi_i|\,\hat{n}_{p\sigma}\,|\phi_j\rangle = \delta_{ij}$. Since only pairs will occupy a state we can substitute the sum over σ with 2. As the number operator acting on a state gives zero if no particle occupies that state ($a\,|0\rangle = 0$), we get

$$\langle\phi_i|\,\hat{H}_0\,|\phi_i\rangle = 2\xi\sum_{p}(p-1)\delta_{p\in\phi_i} \tag{13}$$

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 deteminants. First we cover the diagonal elements:

$$\begin{aligned}\left\langle\Phi^{ab\cdots}_{ij\cdots}\right|\hat{H}_1\left|\Phi^{ab\cdots}_{ij\cdots}\right\rangle &= -\frac{1}{2}g\sum_{pq}\left\langle\Phi^{ab\cdots}_{ij\cdots}\right|P^+_pP^-_q\left|\Phi^{ab\cdots}_{ij\cdots}\right\rangle \\ &= -\frac{1}{2}g\left[\sum_{pq>F}\delta_{pq}\sum_{c>F}\delta_{qc}+\sum_{pq<F}\delta_{pq}\sum_{k<F}^{F}\delta_{kq}\right] \\ &= -\frac{1}{2}g\left[\sum_{a>F}1+\sum_{k<F}1\right] = -\frac{1}{4}g\cdot n_p\end{aligned} \tag{14}$$

where $c\in\Phi^{ab\cdots}_{ij\cdots}$ and $k\in\Phi^{ab\cdots}_{ij\cdots}$. The off-diagonal elements will be

$$\left\langle\Phi^{ab\cdots}_{ij\cdots}\right|\hat{H}_1\left|\Phi^{ac\cdots}_{ij\cdots}\right\rangle = -\frac{1}{2}g\sum_{pq}\delta_{pb}\delta_{qc} \tag{15}$$

for one different excited pair, and

$$\left\langle\Phi^{ab\cdots}_{ij\cdots}\right|\hat{H}_1\left|\Phi^{ab\cdots}_{ik\cdots}\right\rangle = -\frac{1}{2}g\sum_{pq}\delta_{pj}\delta_{qk} \tag{16}$$

for one pair different under the fermi-level.

We have included a python script to calculate the hamiltonian for arbitrary number of pairs, basis states, ξ and g on our github.

2.3 Quantum Computing

2.3.1 Basics

A bit is a basic unit of data in computation. In classical computers, a single bit has a binary value of either 1 or 0, and many bits can be assembled to represent complex information. For an n-bit classical computer,

we are able to represent one of 2^n possible numbers at a time. The bits are manipulated by what is called logic gates to perform logical operations and their operation on bits can be used to perform complex tasks. In quantum computing on the other hand, we represent each bit (called a qubit) by the quantum state

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{17}$$

where we know from quantum mechanics that

$$\alpha^2 + \beta^2 = 1 \tag{18}$$

A qubit in this state is in what we call a superposition, where the value/state of the qubit is not determined before we measure it. The qubit will collapse to either of the states $|0\rangle$ or $|1\rangle$ upon measurement and the probability of the qubit collapsing to the state $|0\rangle$ is given by α^2 while it is given by β^2 for the state $|1\rangle$. Some of the magic of quantum computing comes from the fact that an n-qubit quantum computer can represent 2^n values/states at the same time:

$$\begin{aligned}(\alpha_1\,|0\rangle + \beta_1\,|1\rangle)(\alpha_2\,|0\rangle + \beta_2\,|1\rangle)\cdots(\alpha_n\,|0\rangle + \beta_n\,|1\rangle)\\ = \alpha_1\alpha_2\cdots\alpha_n\,|000\cdots0\rangle + \alpha_1\alpha_2\cdots\beta_n\,|000\cdots1\rangle + \cdots\\ = \sum_{i=0}^{2^n-1} a_i\,|i\rangle\end{aligned}$$

We can represent the states $|0\rangle$ and $|1\rangle$ with vectors which constructs a basis for R^2;

$$\begin{aligned}|0\rangle &\equiv [1,0]^T\\ |1\rangle &\equiv [0,1]^T\end{aligned} \tag{19}$$

and use matrix algebra to perform manipulations on them. These matrices are a mathematical representation of what is called quantum gates and these gates have to preserve the probabilistic property given in eq. 18. One can show that this property is preserved when the matrices (which we will often refer to as operators from now on) are unitary. For an operator U, that is

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

where I is the identity operator. We will switch between the braket- and vector notation in eq. 19 when it suits our purpose.

Some of the most important single-qubit operators in quantum computing, and the ones most relevant to our project, are the Pauli gates;

$$\sigma_x = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \qquad \sigma_y = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} \qquad \sigma_z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \tag{20}$$

the Hadamard gate;

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \tag{21}$$

and the rotation operator;

$$R_z(\theta) = e^{-i\theta\sigma_z/2} = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)\sigma_z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \quad (22)$$

The Pauli X gate for example flips the qubit, that is

$$\sigma_x |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \qquad \sigma_x |1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

while the Hadamard gate creates a superposition:

$$H |0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$H |1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

A gate that will become useful when we get into quantum fourier transform is the R_k gate:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} \quad (23)$$

A system of multiple qubits are represented by tensor products. For example, for an n-qubit state we may write

$$|\psi_1\rangle |\psi_2\rangle \cdots |\psi_n\rangle \equiv |\psi_1\psi_2 \cdots \psi_n\rangle \equiv |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle \quad (24)$$

We can also represent a manipulations on multiple qubits by a tensor product of operators;

$$(A \otimes B \otimes \cdots \otimes N) |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle = A |\psi_1\rangle \otimes B |\psi_2\rangle \otimes \cdots \otimes N |\psi_n\rangle$$
$$or$$
$$A^1 B^2 \cdots N^n |\psi_1\rangle |\psi_2\rangle \cdots |\psi_n\rangle = A^1 |\psi_1\rangle B^2 |\psi_2\rangle \cdots N^n |\psi_n\rangle \quad (25)$$

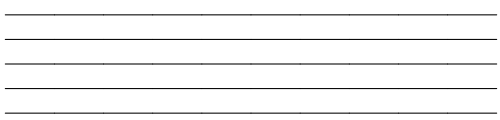
where the superscript denotes which qubit the operator acts on. An important two-qubit gate is the CNOT-gate. Its matrix representation is given by

$$CNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (26)$$

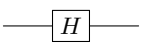
Operating on two qubits $|c\rangle |t\rangle$, its task is to flip the target qubit $|t\rangle$ if the control qubit $|c\rangle$ is in the $|1\rangle$-state. If it is in the $|0\rangle$-state it should act as the identity operator. This kind of gate is called a controlled gate. In the case of the CNOT gate we perform a pauli-x gate on the target qubit conditioned on the control qubit, but we can of course apply any of the other gates mentioned.

2.3.2 Quantum Circuits

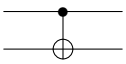
A convenient way to illustrate a quantum algorithm is through quantum circuits. A quantum cirquit consists of wires, where each wire represents a qubit:



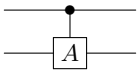
These wires are not physical wires, but you can think of them as representing the passage of time. When reading a quantum circuit, the leftmost operations are performed first, so the circuits are read from left to right. The qubits are usually initialized in the $|0\rangle$ state unless else is specified. To illustrate that an operation is perfomed on a qubit, we draw a gate on the wire corresponding to this qubit:



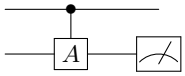
In the above circuit, we only have one qubit which is put in superposition by applying the hadamard gate (eq. 21). To illustrate conditional operations, like the CNOT gate, we write the circuit as follows:



The black dot indicates that we put the condition on the first qubit, while the plus sign surrounded by a circle indicates which qubit to apply the pauli-X gate on. We can also indicate conditional operations with an arbitrary gate A as follows:



If we want to indicate that we perform a measurement on the bottom qubit in the circuit above, we can use a meter symbol on the wire corresponding to the qubit we measure:



2.3.3 Quantum Fourier Transform

The Quantum Fourier Transform (QFT) is a linear transformation on qubits which is used quite frequently in quantum algorithms. The QFT performed on an orthonormal basis yields the following state:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \tag{27}$$

In order to see how we can implement this transformation on a quantum computer, we write the n-qubit state $|j\rangle$ using the binary representation $j = j_1 j_2 \cdots j_n$. Or

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0 \tag{28}$$

It is also useful to denote

$$0.j_1 j_2 \cdots j_n = j_1/2 + j_2/2^2 + \cdots + j_n/2^n, \tag{29}$$

as the binary fraction $0.j$. We can now rewrite equation 27:

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/2^n} |k\rangle$$

Now use binary representation of k

$$= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{n-l})/2^n)} |k_1 \cdots k_n\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} |k_1 \cdots k_n\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle$$

$$= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle$$

Insert $k_l = 0$ and $k_l = 1$

$$= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} [|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle]$$

Now use the binary representation of j (eq. 28)

$$= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} [|0\rangle + e^{2\pi i \sum_{i=1}^{n} j_i 2^{n-l-i}} |1\rangle]$$

Observe that when $k \geq 0$ for the terms $j_i 2^k$ in the exponent, we will just be multiplying $|1\rangle$ with 1. Therefore, we have

$$= \frac{1}{2^{n/2}} \left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle\right) \cdots \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n} |1\rangle\right) \tag{30}$$

Let's see how we set up a circuit to perform a QFT on the state $|j_1 j_2 \cdots j_n\rangle$. First we apply a Hadamard gate to the first state:

$$H^1 |j_1 j_2 \cdots j_n\rangle = (|0\rangle + e^{2\pi i 0.j_1} |1\rangle) |j_2 \cdots j_n\rangle$$

We see that this expression is correct since $e^{2\pi i 0.j_1} = -1$ if $j_1 = 1$ and its 1 if $j_1 = 0$. For the next step, we need to apply a controlled R_2 gate to the first qubit, (see equation 23 for the R_k gate, with $k = 2$) conditioned on the second qubit. This action results in

$$(|0\rangle + e^{2\pi i 0.j_1 + 2\pi i j_2/2^2} |1\rangle) |j_2 \cdots j_n\rangle = (|0\rangle + e^{2\pi i 0.j_1 j_2} |1\rangle) |j_2 \cdots j_n\rangle$$

Continuing with applying a controlled R_k gate on the first qubit conditioned on qubit k for $k = 3, 4, ..., n$, we will end up with

$$(|0\rangle + e^{2\pi i 0.j_1 \cdots j_n} |1\rangle) |j_2 \cdots j_n\rangle$$

We can now apply the Hadamard gate to the second qubit and apply the controlled R_k gate on the preceeding qubits in the same manner and we will end up with equation 30. The complete circuit is illustrated below:

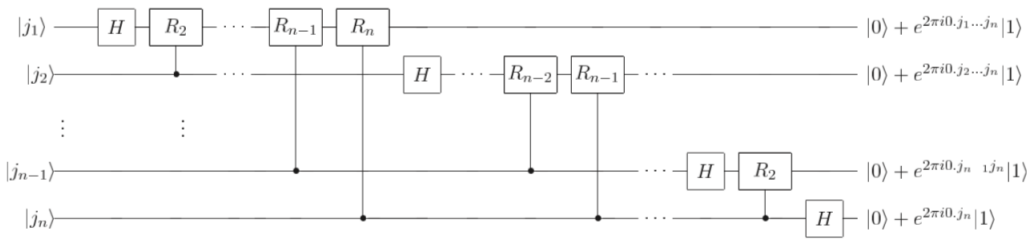


Figure 1: QFT circuit. Figure is taken from the book "Quantum computation and Quantum information" by Michael A. Nielsen and Isaac L. Chuang.

2.3.4 Quantum Phase Estimation

Now that we have shown how to do a Quantum Fourier transform, the next question one may ask is how to make use of it. A central procedure in many quantum algorithms is known as Phase Estimation. If we have a unitary operator U which has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i \lambda}$, where λ is unknown, that is

$$U |u\rangle = e^{2\pi i \lambda} |u\rangle ,$$

the purpose of phase estimation is to estimate λ. The first stage of the phase estimation algorithm is shown in the figure below:

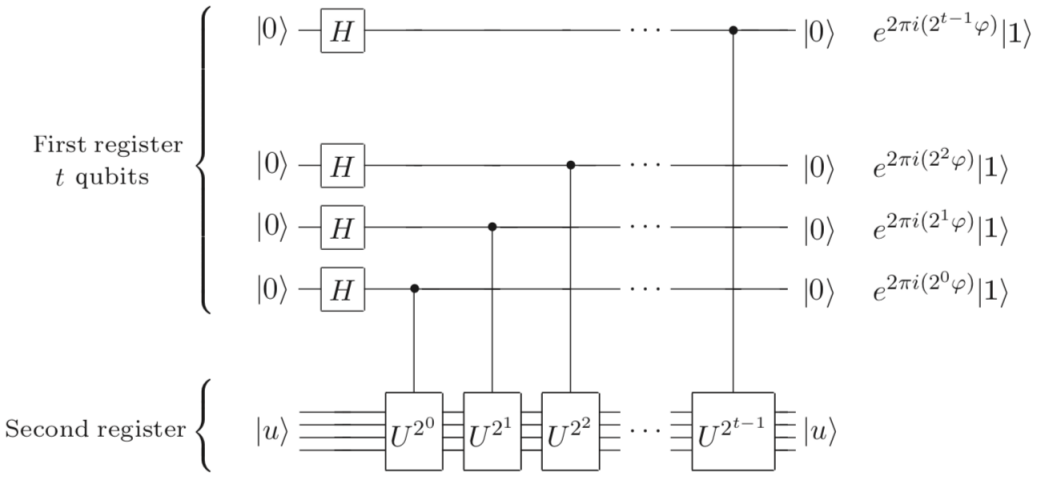


Figure 2: Phase estimation circuit. Figure is taken from the book "Quantum computation and Quantum information" by Michael A. Nielsen and Isaac L. Chuang.

What we see in figure 2 is that we have prepared two quantum registers. One with t qubits, all initialised in state $|0\rangle$ and the second register is initialized as the eigenvector $|u\rangle$. All the qubits in the first register is then put in a superposition by applying the Hadamard gate. We then apply the U^{2^0} operator to the second register, conditional on a qubit in the t-register. This yields

$$|u\rangle \frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i 2^0 \lambda} |1\rangle\right) (|0\rangle + |1\rangle) \cdots (|0\rangle + |1\rangle)$$

Applying the U^{2^1} operator to the second register conditional on the next qubit in the t-register gives

$$|u\rangle \frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i 2^0 \lambda} |1\rangle\right) \left(|0\rangle + e^{2\pi i 2^1 \lambda} |1\rangle\right) (|0\rangle + |1\rangle) \cdots (|0\rangle + |1\rangle).$$

Continuing as shown in the circuit gives us finally

$$|u\rangle \frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i 2^0 \lambda} |1\rangle\right) \left(|0\rangle + e^{2\pi i 2^1 \lambda} |1\rangle\right) \left(|0\rangle + e^{2\pi i 2^2 \lambda} |1\rangle\right) \cdots \left(|0\rangle + e^{2\pi i 2^{t-1} \lambda} |1\rangle\right).$$

Now suppose that we can write the phase exactly as the binary fraction

$$0.\lambda_1 \lambda_2 \cdots \lambda_t = \lambda_1/2 + \lambda_2/2^2 + \cdots + \lambda_t/2^t.$$

The first qubit in the t-register can then be written as

$$\begin{aligned} |0\rangle + e^{2\pi i 2^0 \lambda} |1\rangle &= |0\rangle + e^{2\pi i(\lambda_1/2 + \lambda_2/2^2 + \cdots + \lambda_t/2^t)} |1\rangle \\ &= |0\rangle + e^{2\pi i 0.\lambda_1 \lambda_2 \cdots \lambda_t} |1\rangle \end{aligned}$$

For the second qubit in the t-register, we can write

$$|0\rangle + e^{2\pi i 2^1 \lambda} |1\rangle = |0\rangle + e^{2\pi i 2(\lambda_1/2 + \lambda_2/2^2 + \cdots + \lambda_t/2^t)} |1\rangle$$

Since λ_1 has to be either 1 or 0 we are then left with

$$|0\rangle + e^{2\pi i 2(\lambda_1/2 + \lambda_2/2^2 + \cdots + \lambda_t/2^t)} |1\rangle = |0\rangle + e^{2\pi i \lambda_1} e^{2\pi i(\lambda_2/2^1 + \cdots + \lambda_t/2^{t-1})} |1\rangle$$

$$= |0\rangle + e^{2\pi i 0.\lambda_2 \lambda_3 \cdots \lambda_t} |1\rangle$$

Doing this for all the qubits in the t-register, we get

$$\frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i 0.\lambda_1 \cdots \lambda_t}\right) \left(|0\rangle + e^{2\pi i 0.\lambda_2 \cdots \lambda_t}\right) \cdots \left(|0\rangle + e^{2\pi i 0.\lambda_t}\right) \tag{31}$$

Comparing this with the QFT equation (eq. 30), we see that they are of the same form. Since all operations on qubits are unitary, we can complex conjugate all the operations in the QFT circuit (figure 1) to yield the inverse fourier transform. We can show this by considering some arbitrary unitary operations on a state $|j\rangle$:

$$ABCD \cdots N |j\rangle = |k\rangle$$

$$(ABCD \cdots N)^\dagger |k\rangle = (ABCD \cdots N)^\dagger ABCD \cdots N |j\rangle$$
$$= N^\dagger \cdots D^\dagger C^\dagger B^\dagger A^\dagger ABCD \cdots N |j\rangle = |j\rangle$$

By performing the inverse quantum fourier transform on the state in equation 31, we will end up with the state

$$|\lambda_1 \lambda_2 \cdots \lambda_t\rangle |u\rangle .$$

In other words, we get the exact phase. In reality though, we wont necessarily know the eigenvector $|u\rangle$. To deal with this, we can prepare the second register in a state $|\psi\rangle = \sum_{i=1}^{n} c_i |u_i\rangle$ which is a linear combination of the eigenstates. Repeated applications of the phase estimation algorithm followed by measurements will then yield a specter of eigenvalues and eigenvectors. We also cant always express the phase exactly as a t-bit binary fraction. It can be shown that we will with high probability produce a pretty good estimation to λ nevertheless ([7] section 5.2.1).

2.3.5 Hamiltonian Simulation

We know that a quantum state evolves according to the time evolution operator

$$|\psi(t)\rangle = e^{-i\hat{H}t/\hbar} |\psi(0)\rangle = \hat{U} |\psi(0)\rangle \tag{32}$$

We also know that an eigenstate of the Hamiltonian is an eigenstate of the time evolution operator. Its eigenvalue is given by

$$e^{-i\hat{H}t/\hbar} |\psi_k\rangle = e^{-iE_k t/\hbar} |\psi_k\rangle \tag{33}$$

where E_k is an eigenvalue of $\hat{H}$. On a quantum computer, the phase estimation algorithm finds the phase λ_k of a unitary operator with eigenvalue $e^{-i\lambda_k 2\pi}$. We know that the time evolution operator is unitary so if

we can approximate this efficiently on a quantum computer, we can also approximate the eigenvalue of the Hamiltonian with the phase estimation algorithm. If our Hamiltonian $\hat{H}$ consists of several terms $\sum_k \hat{H}_k$, we can use the trotter approximation to rewrite $\hat{U}$ in a way that, as will be seen, is convenient to implement on a quantum computer. The trotter approximation is given by

$$e^{-i\sum_k \hat{H}_k t/\hbar} = \prod_k e^{-iH_k t/\hbar} + \mathcal{O}(t^2) \tag{34}$$

We first need to rewrite the pairing hamiltonian in terms of quantum logic gates. Some of the most notable are the Pauli gates, which are represented by the unitary matrices in eq. 20. If we chose to represent a qubit in state $|0\rangle$ as a state with a fermion and $|1\rangle$ as a state with no fermion, we see that the operators

$$\begin{aligned} \sigma_+ &= \frac{1}{2}(\sigma_x + i\sigma_y) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ \sigma_- &= \frac{1}{2}(\sigma_x - i\sigma_y) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \end{aligned} \tag{35}$$

has the following effect on qubits

$$\sigma_+ |1\rangle = |0\rangle \qquad \sigma_- |0\rangle = |1\rangle .$$

Hence σ_+ acts as a creation operator and σ_- acts as an annihilation operator. However, since fermionic states are anti-symmetric, $a_a^\dagger a_b^\dagger |c\rangle = -a_b^\dagger a_a^\dagger |c\rangle$, we need our quantum gate representation of the creation/annihilation operators to preserve this property. This can be achieved by multiplying the σ_z matrix on all the occupied states leading up to the one we operate on. The complete creation and annihilation operators can then be represented as

$$a_n^\dagger = \left(\prod_{k=1}^{n-1} \sigma_z^k\right) \sigma_+^n \qquad a_n = \left(\prod_{k=1}^{n-1} \sigma_z^k\right) \sigma_-^n \tag{36}$$

where the superscript tells us which qubit the operator acts on. For convenience, we chose that odd qubits are in a spin up state, while even qubits are in spin down state. For example for the following state

$$\begin{aligned} \textit{Qubit state:} &|0 \quad 0 \quad 1 \quad 1\rangle \\ \textit{Spin state:} &+ \quad - \quad + \quad - \\ \textit{Spacial state:} &\ 1 \quad 1 \quad 2 \quad 2 \end{aligned}$$

the first spacial basis state is occupied with an electron pair with opposite spin, while the second spacial state is not occupied. Our next job is to write our hamiltonian (eq. 7) in terms of the Pauli matrices and a detailed derivation is provided in the appendix section. Here we state the result: For the one body part of the hamiltonian we have the terms

$$\hat{H}_{0p} = \frac{1}{2}\delta(p - 1 - I[p\%2 = 0])(I^p + \sigma_z^p). \tag{37}$$

where we sum this term over each qubit p and % is the modulo operator. $I[f(x) = y] = 1$ if $f(x) = y$ and zero otherwise. For the interaction term we have two possibilities. First for p = q we have:

$$\begin{aligned}\hat{V}_p = -\frac{1}{8}g\Big[&I^{\otimes^{2p-2}} \otimes I \otimes I \otimes I^{\otimes^{n-2p}} \\ &+I^{\otimes^{2p-2}} \otimes I \otimes \sigma_z \otimes I^{\otimes^{n-2p}} \\ &+I^{\otimes^{2p-2}} \otimes \sigma_z \otimes I \otimes I^{\otimes^{n-2p}} \\ &+I^{\otimes^{2p-2}} \otimes \sigma_z \otimes \sigma_z \otimes I^{\otimes^{n-2p}}\Big]\end{aligned} \tag{38}$$

and for $q - p \geq 1$ we get:

$$\begin{aligned}\hat{V}_{pq} = -\frac{1}{16}g[&I^{\otimes^{2p-2}} \otimes \sigma_x \otimes \sigma_x \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_x \otimes \sigma_x \otimes I^{\otimes^{n-2q}} \\ &- I^{\otimes^{2p-2}} \otimes \sigma_x \otimes \sigma_x \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_y \otimes \sigma_y \otimes I^{\otimes^{n-2q}} \\ &+ I^{\otimes^{2p-2}} \otimes \sigma_x \otimes \sigma_y \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_x \otimes \sigma_y \otimes I^{\otimes^{n-2q}} \\ &+ I^{\otimes^{2p-2}} \otimes \sigma_x \otimes \sigma_y \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_y \otimes \sigma_x \otimes I^{\otimes^{n-2q}} \\ &+ I^{\otimes^{2p-2}} \otimes \sigma_y \otimes \sigma_x \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_x \otimes \sigma_y \otimes I^{\otimes^{n-2q}} \\ &+ I^{\otimes^{2p-2}} \otimes \sigma_y \otimes \sigma_x \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_y \otimes \sigma_x \otimes I^{\otimes^{n-2q}} \\ &- I^{\otimes^{2p-2}} \otimes \sigma_y \otimes \sigma_y \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_x \otimes \sigma_x \otimes I^{\otimes^{n-2q}} \\ &+ I^{\otimes^{2p-2}} \otimes \sigma_y \otimes \sigma_y \otimes I^{\otimes^{2(q-p-1)}} \otimes \sigma_y \otimes \sigma_y \otimes I^{\otimes^{n-2q}}]\end{aligned} \tag{39}$$

We have included a factor of two so the sum over p and q here can be restricted to $q > p$.

We then see that our complete pairing hamiltonian can be written as

$$\hat{H} = \sum_p \hat{H}_{0p} + \sum_p \hat{V}_p + \sum_{q>p} \hat{V}_{pq} \tag{40}$$

where $\hat{H}_{0p}$, $\hat{V}_p$ and $\hat{V}_{pq}$ is given by equation 37, 38 and 39 respectively. The time evolution operator is then given by

$$\hat{U}(t) = e^{-i(\sum_p \hat{H}_{0p} + \sum_p \hat{V}_p + \sum_{q>p} \hat{V}_{pq})t} \tag{41}$$

The trotter approximation (eq. 34) can now be used to approximate this operator. We get that

$$\hat{U}(t) = \prod_p e^{-i\hat{H}_{0p}t} \prod_p e^{-i\hat{V}_p t} \prod_{q>p} e^{-i\hat{V}_{pq}t} + \mathcal{O}(t^2) \tag{42}$$

We need a way to implement the above product on a quantum computer. To do this we utilize that a rotation of angle θ about axis $a \in \{x, y, z\}$ in the block sphere is given by

$$R_a(\theta) = e^{-i\frac{\theta}{2}\sigma_a} = cos(\frac{\theta}{2})I - isin(\frac{\theta}{2})\sigma_a. \tag{43}$$

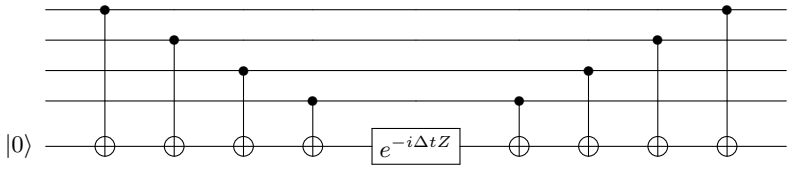
Further we have that

$$\sigma_x = H\sigma_z H \tag{44}$$

and

$$\sigma_y = R_z(\pi/2)H\sigma_z H R_z(-\pi/2) \tag{45}$$

and also that the following circuit implements the time evolution $e^{-i\sigma_z\otimes\sigma_z\otimes\sigma_z\otimes\sigma_z\Delta t} =$



We see that equation 43, 44, 45 and this circuit is all we need to implement the products in the pairing time evolution operator. For example, for the tensor product $H = \sigma_x \otimes \sigma_y \otimes \sigma_x \otimes \sigma_y$ we have:

$$\begin{aligned}
&H \otimes R_z(\pi/2)H \otimes H \otimes R_z(\pi/2)H \\
&\times e^{-i\Delta t\sigma_z\otimes\sigma_z\otimes\sigma_z\otimes\sigma_z} \\
&\times H \otimes HR_z(-\pi/2) \otimes H \otimes HR_z(-\pi/2) \\
&= U e^{-i\Delta t\sigma_z\otimes\sigma_z\otimes\sigma_z\otimes\sigma_z} U^\dagger \\
&= U[cos(\Delta t)I - isin(\Delta t)\sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_z]U^\dagger \\
&= cos(\Delta t)I - isin(\Delta t)\sigma_x \otimes \sigma_y \otimes \sigma_x \otimes \sigma_y \\
&= e^{-i\Delta t\sigma_x\otimes\sigma_y\otimes\sigma_x\otimes\sigma_y}
\end{aligned}$$

since $UU^\dagger = I$ and

$$\begin{aligned}
&U \times (\sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_z) \times U^\dagger \\
&= (H \otimes R_z(\pi/2)H \otimes H \otimes R_z(\pi/2)H) \\
&\times(\sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_z) \\
&\times(H \otimes HR_z(-\pi/2) \otimes H \otimes HR_z(-\pi/2)) \\
&= H\sigma_z H \otimes R_z(\pi/2)H\sigma_z HR_z(-\pi/2) \otimes H\sigma_z H \otimes R_z(\pi/2)H\sigma_z HR_z(-\pi/2) \\
&= \sigma_x \otimes \sigma_y \otimes \sigma_x \otimes \sigma_y
\end{aligned}$$

2.3.6 Practical information on the Phase Estimation implementation

The phase estimation algorithm is aimed at finding λ_k for a unitary operator $\hat{U}$ with eigenvalue $e^{i2\pi\lambda_k}$, such that

$$\hat{U}\left|\psi_k\right\rangle = e^{i2\pi\lambda_k}\left|\psi_k\right\rangle,$$

In our case, the time evolution operator has the eigenvalues $e^{-iE_k\Delta t}$ which means that the value we read from the phase estimation algorithm is

$$\lambda_k = -E_k\Delta t/2\pi$$

An assumption with the phase estimation algorithm is that we can write the eigenvalue $\lambda = 0.\lambda_1\lambda_2\cdots\lambda_t$ as a binary fraction. Since this is a positive number less than one, we need our eigenvalues to be negative for the phase estimation algorithm to work according to the above equation for λ_k. We can force this by subtracting a large enough constant value E_{max} from the Hamiltonian since

$$(\hat{H} - E_{max})\left|\phi_k\right\rangle = (E_k - E_{max})\left|\phi_k\right\rangle.$$

This gives us

$$\lambda_k = -(E_k - E_{max})\Delta t/2\pi$$
$$\lambda_k 2\pi/\Delta t = E_{max} - E_k$$
$$E_k = E_{max} - \lambda_k 2\pi/\Delta t$$

Further, since the phase estimation algorithm yields the following state for the work qubits

$$\left|\lambda_1\lambda_2\cdots\lambda_t\right\rangle = \left|\lambda 2^t\right\rangle$$

we transform the measured binary number to a binary fraction before plugging it into the equation for E_k. We can also see that if we have $\lambda = \lambda^{'} + n > 1$ where $0 < \lambda^{'} < 1$ and n is a positive integer, we get from the phase estimation algorithm

$$e^{i2\pi(\lambda^{'}+n)} = e^{i2\pi n}e^{i2\pi\lambda^{'}} = e^{i2\pi\lambda^{'}}$$

or written in binary form

$$e^{i2\pi(\lambda^{'}+n)} = e^{i2\pi\lambda_1\cdots\lambda_k.\lambda_{k+1}\cdots\lambda_n} = e^{i2\pi 0.\lambda_{k+1}\cdots\lambda_n}$$

In other words; for eigenvalues greater than one, we lose information. A restriction on $\lambda_k < 1$ gives

$$-E_k\Delta t/2\pi < 1$$
$$\implies -E_k\Delta t < 2\pi$$

or

$$-(E_k - E_{max})\Delta t < 2\pi$$

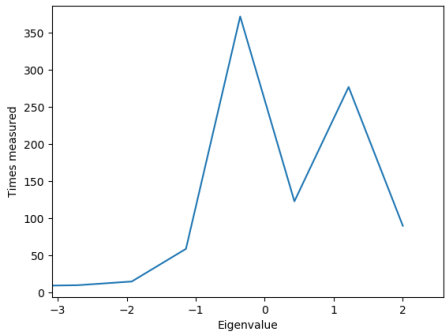
Substituting E_k with E_{min} (the lowest eigenvalue of $\hat{H}$) gives the upper bound on Δt in order to yield the whole eigenvalue spectrum

$$\Delta t < \frac{2\pi}{E_{max} - E_{min}}$$

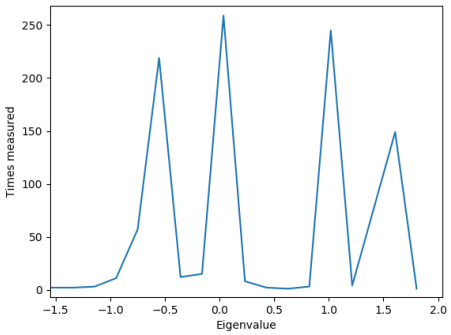
We also have to keep in mind the number of work qubits to use. k work qubits gives us the ability to represent 2^k binary fractions. A quantum state represented by s simulation qubits potentially has 2^s eigenvalues. This means that we will have $\frac{2^t}{2^s} = 2^{t-s}$ points for each eigenvalue. Section 5.4.2 in the article in ref [8] illustrates that a surplus of around 5 work qubits are usually sufficient to calculate the eigenvalue-spectra.

3 Results

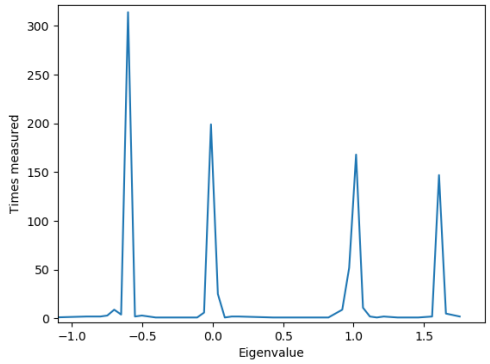
We run the phase estimation algorithm on the unitary operator given by eq. 42. We did this with four simulation qubits and for varying amount of work qubits. The simulation qubits are initialised to superpositions of bell states since electron pairs are not allowed to be separated. We perform 1000 measurements on the system. ξ and g from eq. 7 was both set to 1. The results can be seen in the plots in figure 3 below:



(a) 4 simulation and 4 work qubits



(b) 4 simulation and 6 work qubits



(c) 4 simulation and 8 work qubits

Figure 3: Plots of the amount of times an eigenvalue estimate is measured against the eigenvalue estimate. The quantum circuit is run and measured 1000 times in total.

We see that as we increase the amount of work qubits, we get well defined peaks in the distribution of the measured eigenvalues. We can find the eigenvalues from the plots by approximating eigenvalue j with

the mean of peak j as

$$E[\lambda^{(j)}] = \frac{\sum_i \lambda_i^{(j)} N_i^{(j)}}{\sum_i N_i^{(j)}}$$

where $\lambda_i^{(j)}$ is the i'th measurement in peak j and $N_i^{(j)}$ is the number of times its measured. The variance is then given by

$$\frac{1}{\sum_i N_i^{(j)}} \sum_i (E[\lambda^{(j)}] - \lambda_i^{(j)})^2.$$

In order to do this, we need to define a minimum amount of measurements before the algorithm considers that an eigenvalue to belongs to a peak. We set this value to 4 measurements and the results for 8 work qubits can be can be seen in table 1 below:

Number of pairs	FCI Eigenvalue(s)	Quantum Eigenvalue(s)
0	0	0.0 ± 0.2
1	-0.61803399, 1.61803399	-0.6 ± 0.1, 1.61 ± 0.06
2	1.00000000	1.0 ± 0.2

Table 1: FCI and quantum computer energies for 4 basis states for one and two pairs. $\xi = 1$, $g = 1$. We also provide two standard deviations with the quantum phase estimation algorithm.

We see that the FCI and Quantum phase estimation eigenvalues correspond well.

4 Discussion

As we can see from figure 3, we get defined peaks in the distribution of measurements as we increase the number of work qubits. This is expected as t work qubits can represent 2^t binary numbers, yielding a doubling of the resolution with a unit increase of t. Further we found, as can be seen in table 1, that the quantum phase estimation algorithm yielded eigenvalues within two standard deviations of the eigenvalues found with configuration interaction theory. We used the same amount of spacial basis states in both methods and since FCI yields an exact result within the basis spanned by these states, it serves as a good benchmark for the quantum algorithm. Further we see that while we need to specify the amount of pairs for the FCI algorithm, the quantum algorithm provides the eigenvalues for all possible amount of pairs (in this case 0, 1 and 2 pairs). We used the trotter approximation to order $\mathcal{O}(t^2)$, but it would be interesting to compare this with the approximation to order $\mathcal{O}(t^3)$. Research have indicated that while the $\mathcal{O}(t^3)$ approximation requires roughly twice as many operations per iteration, the decrease in the required amount of iterations provides a reduction in computational time when compared with the $\mathcal{O}(t^2)$ approximation ([8] pg. 83).

5 Conclusion

The quantum algorithm provided results which showed good correspondence with the eigenvalues provided with FCI. As can be seen in table 1, we got the eigenvalues -0.6 ± 0.1 and 1.61 ± 0.06 for one pair, and this result is consistent with the FCI eigenvalues -0.618 and 1.618. For two pairs, the quantum algorithm yielded 1.0 ± 0.2 which also is consistent with the FCI eigenvalue of 1.0. We also found that an increase in the amount of work qubits provided a higher resolution in the estimate of the eigenvalues as expected. We utilized the trotter approximation to order $\mathcal{O}(t^2)$ and it would be interesting to compare this approximation with the approximation to order $\mathcal{O}(t^3)$.

References

[1] A. Bohr, B. R. Mottelson, and D. Pines. Possible analogy between the excitation spectra of nuclei and those of the superconducting metallic state. *Phys. Rev.*, 110:936–938, May 1958.

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

[3] David Deutch. Quantum theory, the church-turing principle and the universal quantum computer. 1985.

[4] Morten H. Jensen and D. J. Dean. Pairing in nuclear systems: from neutron stars to finite nuclei. *journals.aps.org*, April 2003.

[5] Tomi H Johnson. What is a quantum simulator? 2014.

[6] Seth Lloyd. Universal quantum simulators. 1996.

[7] Michael A. Nielsen and Isaac L. Chuang. Quantum computation and quantum information. Mai 2003.

[8] Eirik Ovrum. Quantum computing and many-body physics. Mai 2003.

6 Appendix

Here we show how we write our hamiltonian (eq. 7) in terms of the Pauli matrices. First, we observe that

$$a_i^\dagger a_i = \left(\prod_{k=1}^{i-1} \sigma_z^k\right) \sigma_+^i \left(\prod_{k=1}^{i-1} \sigma_z^k\right) \sigma_-^i$$

$$= \sigma_+^i \sigma_-^i = \frac{1}{2}(I^i + \sigma_z^i)$$

and see that the one-body part of the hamiltonian can be written as

$$\hat{H}_0 = \frac{1}{2}\xi \sum_p (p - 1 - I[p\%2 = 0])(I^p + \sigma_z^p).$$

where p now runs over each qubit and % is the modulo operator. The term $I(f(x) = y) = 1$ if $f(x) = y$ and zero otherwise and comes from the

fact that we sum over both spin states for each spacial state. Now for the interaction term:

$$\hat{V} = -\frac{1}{2}g\sum_{pq}\left(\prod_{k=1}^{2p-2}\sigma_z^k\right)\sigma_+^{2p-1}\left(\prod_{k=1}^{2p-1}\sigma_z^k\right)\sigma_+^{2p}\left(\prod_{k=1}^{2q-1}\sigma_z^k\right)\sigma_-^{2q}\left(\prod_{k=1}^{2q-2}\sigma_z^k\right)\sigma_-^{2q-1} \tag{46}$$

We can see that

$$\left(\prod_{k=1}^{2p-2}\sigma_z^k\right)\sigma_+^{2p-1}\left(\prod_{k=1}^{2p-1}\sigma_z^k\right)\sigma_+^{2p} = \sigma_z^{\otimes^{2p-2}}\otimes\sigma_+\otimes I^{\otimes^{n-2p-1}}\times\sigma_z^{\otimes^{2p-1}}\otimes\sigma_+\otimes I^{\otimes^{n-2p}} = I^{\otimes^{2p-2}}\otimes\sigma_+\sigma_z\otimes\sigma_+\otimes I^{\otimes^{n-2p}}$$

so we can rewrite equation 46 as

$$\hat{V} = -\frac{1}{2}g\sum_{pq}\left[I^{\otimes^{2p-2}}\otimes\sigma_+\otimes\sigma_+\otimes I^{\otimes^{n-2p}}\right]\left[I^{\otimes^{2q-2}}\otimes\sigma_-\otimes\sigma_-\otimes I^{\otimes^{n-2q}}\right]$$

We have two possibilities here. First for $p = q$:

$$-\frac{1}{2}g\left[I^{\otimes^{2p-2}}\otimes\sigma_+\sigma_-\otimes\sigma_+\sigma_-\otimes I^{\otimes^{n-2p}}\right] \tag{47}$$

and for $q - p \geq 1$ we have

$$-\frac{1}{2}g\left[I^{\otimes^{2p-2}}\otimes\sigma_+\otimes\sigma_+\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_-\otimes\sigma_-\otimes I^{\otimes^{n-2q}}\right] \tag{48}$$

For $p - q \geq 1$ we simply exchange p with q and σ_+ with σ_-. To continue, we insert the expressions for $\sigma_\pm$ (eq. 35) into these equations. For p = q (eq. 47) we then have:

$$\begin{aligned} V_{p=q} &= -\frac{1}{8}g\Big[I^{\otimes^{2p-2}}\otimes(I+\sigma_z)\otimes(I+\sigma_z)\otimes I^{\otimes^{n-2p}}\Big] \\ &= -\frac{1}{8}g\Big[I^{\otimes^{2p-2}}\otimes I\otimes I\otimes I^{\otimes^{n-2p}} \\ &\quad + I^{\otimes^{2p-2}}\otimes I\otimes\sigma_z\otimes I^{\otimes^{n-2p}} \\ &\quad + I^{\otimes^{2p-2}}\otimes\sigma_z\otimes I\otimes I^{\otimes^{n-2p}} \\ &\quad + I^{\otimes^{2p-2}}\otimes\sigma_z\otimes\sigma_z\otimes I^{\otimes^{n-2p}}\Big] \end{aligned}$$

and for $q - p \geq 1$ (eq. 48) we get:

$$-\frac{1}{32}g\left[I^{\otimes^{2p-2}}\otimes(\sigma_x+i\sigma_y)\otimes(\sigma_x+i\sigma_y)\otimes I^{\otimes^{2(q-p-1)}}\otimes(\sigma_x-i\sigma_y)\otimes(\sigma_x-i\sigma_y)\otimes I^{\otimes^{n-2q}}\right]$$

This can be rewritten as sixteen four qubit operations

$$
\begin{aligned}
-\frac{1}{32}g[&I^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&\mp iI^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}\\
&\mp iI^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&- I^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}\\
&\pm iI^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\sigma_x\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&+ I^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}\\
&+ I^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&\mp iI^{\otimes^{2p_q-2}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}\\
&\pm iI^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&+ I^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}\\
&+ I^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&\mp iI^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}\\
&- I^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&\pm iI^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}\\
&\pm iI^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{n-2q_p}}\\
&+ I^{\otimes^{2p_q-2}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{2(q_p-p_q-1)}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{n-2q_p}}]
\end{aligned}
$$

where the subscript is used if $p > q$. We can easily see that when running over the sum over p and q, the terms with $\pm$ and $\mp$ sign in front will cancel out. Also, we can include a factor 2 and limit the sum to $p < q$. Therefore:

$$
\begin{aligned}
V_{p\neq q} = -\frac{1}{16}g[&I^{\otimes^{2p-2}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{n-2q}}\\
&- I^{\otimes^{2p-2}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{n-2q}}\\
&+ I^{\otimes^{2p-2}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{n-2q}}\\
&+ I^{\otimes^{2p-2}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{n-2q}}\\
&+ I^{\otimes^{2p-2}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_x\otimes\sigma_y\otimes I^{\otimes^{n-2q}}\\
&+ I^{\otimes^{2p-2}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_y\otimes\sigma_x\otimes I^{\otimes^{n-2q}}\\
&- I^{\otimes^{2p-2}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_x\otimes\sigma_x\otimes I^{\otimes^{n-2q}}\\
&+ I^{\otimes^{2p-2}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{2(q-p-1)}}\otimes\sigma_y\otimes\sigma_y\otimes I^{\otimes^{n-2q}}]
\end{aligned}
$$