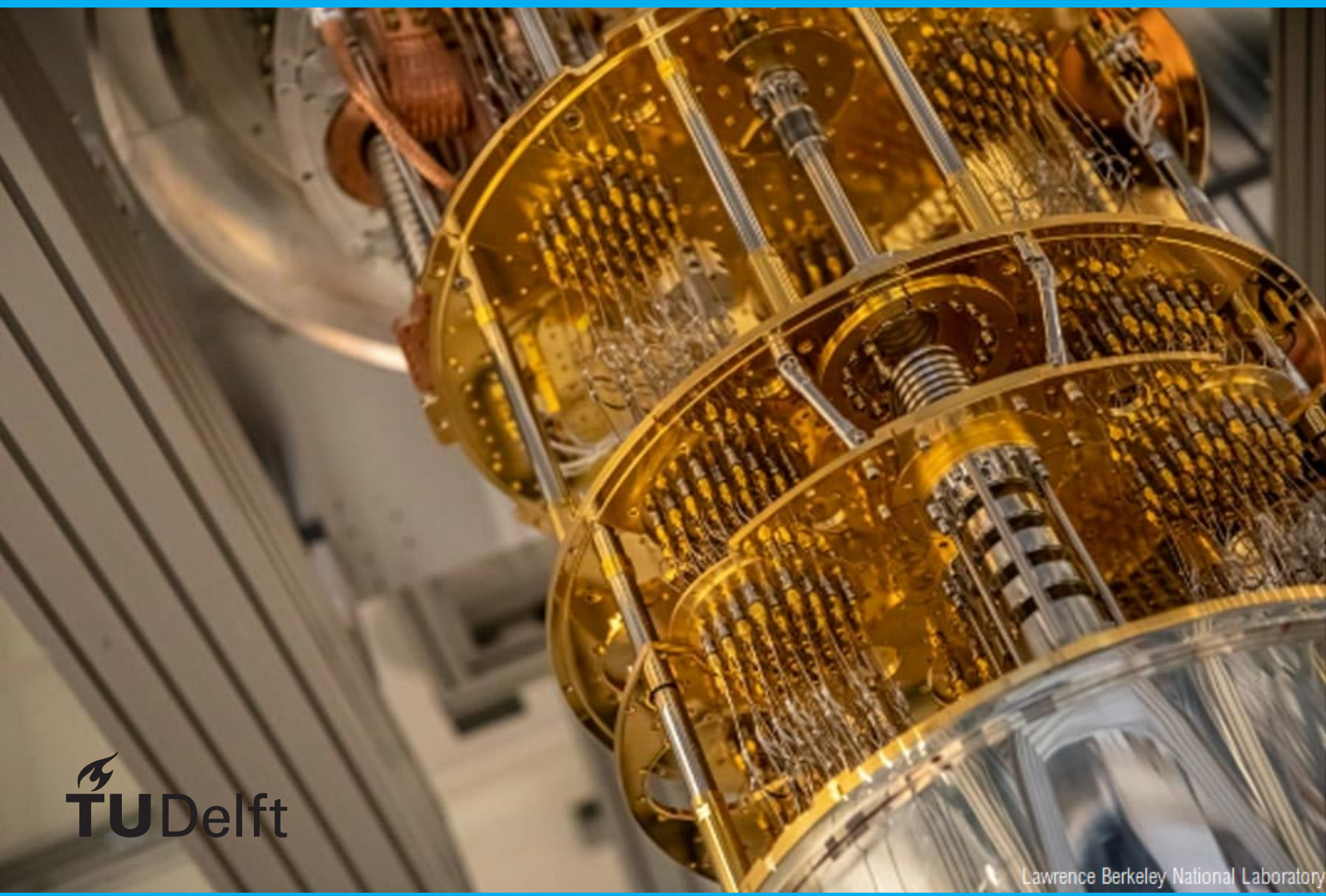
Variational Quantum Algorithms for Optimizing Probe States

Benjamin Oudejans

Supervised by Johannes Borregaard and David de Laat



TU Delft

Lawrence Berkeley National Laboratory

Variational Quantum Algorithms

for Optimizing Probe States

by

Benjamin Oudejans

to obtain the degree of Bachelor of Science in Applied Physics & Applied Mathematics
at the Delft University of Technology

Student number: 4910796
Project duration: March 2, 2021 – July 9, 2021
Thesis committee: Dr. J. Borregaard,
Dr. D. de Laat,
Dr. J. M. Thijssen,
Dr. W. G. M. Groenevelt

TU Delft

Abstract

As quantum computers are developing, they are beginning to become useful for practical applications, for example in the field of quantum metrology. In this work, a variational quantum algorithm is used to find an optimal probe state for measuring parameters in a noisy environment. This is achieved by optimizing a cost on a quantum computer, based on the Fisher information of the parameters to be estimated. These parameters are then estimated using maximum likelihood estimators. In a simulation, a probe state was found that performed better than the best possible state for noiseless measurements, although this could not be reproduced on an actual quantum computer.

Contents

1

Introduction

Quantum computers have been theorized since the 1980s, when Yuri Manin and Richard Feynman proposed using a quantum computer to simulate quantum systems, which Feynman showed was impossible on a classical computer. Later it was discovered that the uses of quantum computers extend beyond just quantum mechanical problems. Notably, in 1990, Peter Shor invented an algorithm that can factorize numbers into primes in polynomial time, while no known classical algorithms can do this. This has often been named the demise of encryption, as important encryption algorithms such as RSA on prime factorization being exponentially hard. More recently, quantum computers have been built, but encryption has held up so far, and in general quantum computers have not superseded classical computers for any everyday tasks. This is because making quantum computers is quite difficult, due to the fragility of quantum states. Various sources of noise disturb the qubits in the computer, causing inaccurate and unreliable outcomes. Scaling the amount of qubits up has proven difficult too. Hence, in this era of noisy intermediate scale quantum computers, the amount of practical applications is limited.

However, shorter circuits can be run reliably on quantum computers, and even these can take a long time to simulate on a classical computer for larger numbers of qubits. This leads to the idea of variational quantum algorithms, where quantum computers are combined with classical computers to perform an optimization algorithm. The cost function and its gradients can be computed on a quantum computer, and the classical computer optimizes the parameters using gradient descent. This is useful when evaluating the cost is infeasible on a classical computer but can be efficiently computed with the use of a quantum computer.

In this work, a variational quantum algorithm will be applied to the field of quantum metrology. An important result in quantum metrology is that measuring on a set of entangled particles can yield more precise results than independent measurements. The downside is that entangled particles are more sensitive to noise, since noise on any of the particles affects all of them. A variational quantum algorithm can be employed to find a state that optimizes measurements in the noisy environment of a quantum computer, by being entangled enough to reap the benefits of quantum metrology without being affected by noise too much.

This will be achieved by optimizing a cost function based on the Fisher information of the parameters to be measured. Then using maximum likelihood estimators, the parameters can be extracted from the probes. In this work, it will be investigated whether this method can find a probe state that can do better than the best probe state for noiseless measurements. This will be done both in simulators and on actual IBM quantum computer.

2

Quantum Computers

Quantum computers are computers that make use of quantum states for computation. Whereas classical computers make use of classical bits, quantum computers operate on the state of qubits. Qubits are quantum systems that are constrained to a two-dimensional state. By manipulating these states, the computer can perform some tasks very quickly compared to classical computers. The quantum computer can typically only measure in one basis, the *computational basis*, which are represented by $|0\rangle$ and $|1\rangle$. Qubit states will generally be written in this basis.

2.1. Quantum gates

In order to compute with these qubits, there are quantum gates which change their states. Since these gates bring about some evolution of the quantum system, they must be unitary operators. An example is the Hadamard gate. This gate acts on the basis states in the following manner

$$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

And is thus represented as $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ in the computational basis.

Something to note is that the $|0\rangle$ and $|1\rangle$ states will have different energy levels, so after time t the state $\alpha|0\rangle + \beta|1\rangle$ will become $\alpha|0\rangle + e^{-i\Delta E\, t/\hbar}\beta|1\rangle$. This is inconvenient, ideally the state should only change if the computer does something to it. Hence the state will be viewed in a rotating frame, where $|1\rangle$ actually represents $e^{-i\Delta E\, t/\hbar}\beta|1\rangle$.

If there are two qubits, the state of the system now lives in the tensor product of the two individual Hilbert spaces, which is spanned by the basis states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$.

In this multiple qubit case, gates can still be applied to each qubit individually. For example, applying the Hadamard gate to the first bit of the $|00\rangle$ state gives

$$(H|0\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

Another way to view this, is to apply an operator which acts as H on the first qubit and as I, the identity, on the second. The way to obtain the operator that acts on the 2-qubit state is by taking the tensor product $H_1 = H \otimes I$. In matrix representation, this is done by taking the Kronecker product.

This is a useful concept, because there are gates that work on 2-qubit states that do not act on the 2 qubits individually. This happens for controlled gates, which apply a gate on a *target qubit* depending on the state of a *control qubit*. If the control qubit is in the $|0\rangle$ state, the gate is not applied to the target qubit, if the control qubit is in the $|1\rangle$ state it is. Usually though, the control qubit is in a superposition. Then the target qubit is put in a superposition of the gate being applied and not being applied. More rigorously, if G

is some gate, then $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes G$ is the controlled-G gate, notated as CG, with the first qubit as control and the second as target. Then if the control qubit is in state $\alpha |0\rangle + \beta |1\rangle$, and the target qubit is in state $|\psi\rangle$, the resulting state after applying the gate is

$$CG((\alpha |0\rangle + \beta |1\rangle) \otimes |\psi\rangle) = \alpha |0\rangle |\psi\rangle + \beta |1\rangle (G |\psi\rangle)$$

Note that even though the qubits are referred to as the control and target qubits, the control qubit's state is affected by this operation too. In fact, depending on G, this operation will often entangle the two qubits [12].

Some frequently used gates are parametrized gates, such as the rotation gates $R_x(\theta) = e^{-i\theta\sigma_x/2}$, and similarly $R_y(\theta)$ and $R_z(\theta)$ with the corresponding Pauli matrix in the exponent.

In practice, a quantum computer will only be able to execute a certain set of gates. However, many gates can be decomposed into others. In general, there exists no finite set of gates that can be combined to perform every possible unitary operation, but it is possible to have a set of gates that can approximate any unitary arbitrarily well with a finite amount of gates. Such a set is called a universal gate set. The Solovay-Kitaev theorem shows that approximating gates can be done efficiently with any universal gate set: any 1-qubit gate can be approximated to error ε in operator norm by a sequence of gates from a universal gate set, where the length of the sequence grows as $\mathcal{O}(\log^c(1/\varepsilon))$. Here c is is a constant depending on the specific gate set, but is less than 4 for any universal gate set [8].

One particular task which quantum computers are more efficient at than classical computers, is evaluating expectation values of observables. The Hilbert space of states of n two level systems has dimension 2^n, meaning an observable is represented by a $2^n \times 2^n$ complex matrix. This quickly becomes infeasible to compute on a classical computer, a 30 particle state would take 10 GB of memory if stored as an array of floating point numbers, and 10^{18} multiplications would have to be performed. For a quantum computer, only 30 qubits are needed. This amount of qubits has already been achieved in quantum computers, although noise makes current quantum computers of this size too inaccurate for many purposes. The following chapter will discuss the workings of qubits and how this noise comes about.

3

Superconducting Qubits

In the previous chapter, it was discussed how quantum computers work in theory. Realising quantum computers has been difficult, however, due to the fragility of quantum states. There are several technologies in development, but the majority of currently operational quantum computers are based on superconducting qubits, specifically the transmon qubit. This chapter will describe how these qubits work.

3.1. Superconducting Qubit

To create a qubit, one needs to create a system with two available eigenstates. In a superconducting qubit, this system is a superconducting electrical circuit. In many ways such a circuit works similarly to a classical circuit, except for the fact that in some materials at low enough temperatures, electrons pair up. This is made possible by an attractive force between the electons, due to the electrons deforming the lattice of the material. Such a pair, called a Cooper pair, together forms a boson. Since the Pauli exclusion principle now no longer applies, all the Cooper pairs can occupy a shared ground state. In this state, there is a high energy gap to the next allowed energy level. This cannot be overcome by lattice interactions at low temperatures, hence eliminating any resistance. The mechanism of the formation of Cooper pairs as a result of the attractive force, and how the high energy gap is created is described by BCS theory [3].

To understand how a superconducting qubit is formed, we conider a circuit containing a capacitive and an inductive element, making an LC-circuit which has a resonant frequency at $\omega_r = \frac{1}{\sqrt{LC}}$, where L and C are the inductance and capacitance of the circuit elements, see figure 3.1.

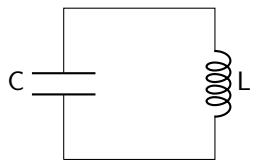


Figure 3.1: Superconducting LC-circuit

Such a circuit does not require a voltage or current source, since the lack of resistance means that any current will keep circulating indefinitely.

Classically, in a circuit with a capacitor of capacitance C and an inductor of inductance L such a circuit has energy

$$E = \frac{1}{2}CV^2 + \frac{1}{2}LI^2 \tag{3.1}$$

Where V is the voltage over the capacitor and I is the current through the inductor. To analyse the circuit on a quantum-mechanical level, it is more useful to express the energy in terms of the charge Q stored in the capacitor, and the branch flux Φ through the inductor. These quantities are defined as follows

$$Q(t) = \int_{-\infty}^{t} I(t')dt' \tag{3.2}$$

$$\Phi(t) = \int_{-\infty}^{t} V(t')dt' \tag{3.3}$$

Then the energy of the capacitor can be written as $E_C = \frac{1}{2}C\dot{\Phi}^2$, and using the relation $I = L\Phi$ for the inductor, its energy becomes $E_L = \frac{1}{2L}\Phi^2$. Since E_L only depends on Φ and E_C only on its derivative, E_L can be viewed as the potential and E_C as the kinetic energy. Then the Lagrangian of the system is

$$\mathcal{L} = \frac{1}{2}C\dot{\Phi}^2 - \frac{1}{2L}\Phi^2$$

The canonical conjugate of Φ can then be calculated as $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\dot{\Phi}} = C\dot{\Phi} = CV = Q$ by the known relation for capacitors. That results in the following Hamiltonian:

$$\begin{aligned}
\mathcal{H} &= \dot{\Phi}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\dot{\Phi}} - \mathcal{L} \\
&= \dot{\Phi}\cdot C\dot{\Phi} - \left(\frac{1}{2}C\dot{\Phi}^2 - \frac{1}{2L}\Phi^2\right) \\
&= \frac{1}{2}C\dot{\Phi}^2 + \frac{1}{2L}\Phi^2 \qquad (3.4)\\
&= \frac{Q^2}{2C} + \frac{\Phi^2}{2L} \qquad (3.5)
\end{aligned}$$

If we compare this to a harmonic oscillator of a spring-mass system with $E = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\omega_r^2 x^2$, comparing its coefficients to equation 3.4 gives $m = C$ and $\omega_r = \frac{1}{\sqrt{LC}}$. Thus such a circuit acts as a harmonic oscillator.

To go to a quantum-mechanical description, the classical Hamiltonian will be converted to a quantum Hamiltonian by replacing the classical variables in equation 3.5 by their quantum operator counterparts. Thus the Hamiltonian of the system is

$$\hat{H} = \frac{\hat{Q}^2}{2C} + \frac{\hat{\Phi}^2}{2L} \tag{3.6}$$

The quantization condition by Dirac [5] states that the commutator of quantum operators that correspond to a generalized coordinate and its canonical conjugate is $i\hbar$, hence

$$[\hat{\Phi}, \hat{Q}] = i\hbar \tag{3.7}$$

$\hat{Q}$ and $\hat{\Phi}$ can be reduced to unitless quantities: $\hat{n} = \frac{\hat{Q}}{2e}$ is the number of excess Cooper pairs in the inductor, and $\hat{\phi} = 2\pi\frac{\hat{\Phi}}{\Phi_0}$ is the reduced flux. These have a commutator of $[\hat{\phi}, \hat{n}] = i$. Then the Hamiltonian can be rewritten as [9]

$$\hat{H} = 4E_C\hat{n}^2 + \frac{1}{2}E_L\hat{\phi}^2$$

with $E_C = e^2/(2C)$ and $E_L = (\Phi_0/2\pi)^2/L$.

For a Cooper pair in the circuit, this means that it is in a harmonic oscillator potential with frequency ω_r. This creates the discrete energy levels $E_k = \hbar\omega_r(k + \frac{1}{2})$ that the Cooper pairs can occupy. The resonant frequency will be tuned so that energy transitions can be brought about by photons of microwave frequencies. However, since the energy levels are equally spaced, a photon that excites a Cooper pair from $|0\rangle$ to $|1\rangle$ might just as well excite a Cooper pair from $|1\rangle$ to $|2\rangle$. Since the Cooper pairs should only be able to access the two lowest state, a mechanism is necessary to differentiate the energy gaps .

Superconducting qubits achieve this using a Josephson junction. It consists of two superconducting islands, with a non-superconducting material in between. Electrons can tunnel across the gap from one

island to the other, but it turns out that Cooper pairs can tunnel together, which is called the Josephson effect. This allows a superconducting current through the junction, following the relation

$$I = I_c \sin(\phi) \tag{3.8}$$

where ϕ is the reduced flux through the junction.

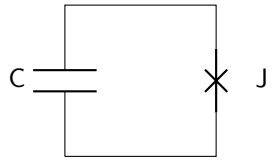


Figure 3.2: Superconducting qubit

If a Josephson junction is used instead of an inductor, the Hamiltonian becomes

$$H = 4E_C\hat{n}^2 - E_J \cos\left(\hat{\phi}\right) \tag{3.9}$$

with $E_J = I_c\Phi_0/2\pi$ [17].

This is still approximately a harmonic oscillator, which can be seen by Taylor-expanding the cosine, but has a perturbation of order $\mathcal{O}(\phi^4)$. This causes the eigenenergies to shift so that they are not equally spaced anymore. Now photons of frequency $\omega_{0\to1} = \frac{E_1-E_0}{\hbar}$ can be used to drive interactions between $|0\rangle$ and $|1\rangle$, while having only a small chance of affecting other energy transitions. The size of the anharmonicity depends on $\epsilon = \sqrt{2E_C/E_J}$ Using a perturbative solution of this Hamiltonian, the anharmonicities can be approximated as [10]

$$\frac{E_1 - E_0}{\omega_r} = 1 - \frac{1}{4}\epsilon - \frac{1}{16}\epsilon^2 + \mathcal{O}(\epsilon^3) \tag{3.10}$$

$$\frac{E_2 - E_1}{\omega_r} = 1 - \frac{1}{2}\epsilon - \frac{13}{64}\epsilon^2 + \mathcal{O}(\epsilon^3) \tag{3.11}$$

which results in a difference in energy gap of $\omega_r(\frac{1}{4}\epsilon - \frac{9}{64}\epsilon^2 + \mathcal{O}(\epsilon^3))$

3.2. Application of gates

Single qubit gates are applied by capacitively coupling a sinusoidal voltage source to the superconducting circuit, see figure . This voltage can be expressed as $V_d(t) = V_0 s(t)\sin(\omega_d t + \phi)$. This adds a term H_d to the Hamiltonian, which in the rotating frame is given by [9]

$$H_d = \Omega V_0 s(t) \cdot (\cos(\phi)\sin(\omega_d t) - \sin(\phi)\omega_d t)) \cdot (\cos(\omega_r t)\sigma_y - \sin(\omega_r t)\sigma_x) \tag{3.12}$$

Where $\Omega = C_d/(C_d + C)\sqrt{\hbar C/2L}$.

If the applied frequency is matched with the qubit frequency, so $\omega_d = \omega_r$, and high frequency terms are neglected, this reduces to

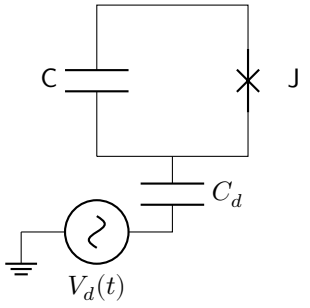


Figure 3.3: Superconducting qubit with driving voltage source

$$H_d = -\frac{\Omega}{2} V_0 s(t)(\cos(\phi)\sigma_x + \sin(\phi)\sigma_y) \tag{3.13}$$

Hence it is possible to choose the axis of rotation of the qubit state by choosing the corresponding initial phase ϕ. The angle of rotation is then

$$\Theta(t) = \Omega V_0 \int_0^t s(t')dt' \tag{3.14}$$

Any single qubit gate can be achieved by applying a sequence of these rotation gates.

This covers the single-qubit gates, but in order to make a universal quantum computer, there also need to be 2-qubit gates. There are several methods of doing this, but the cross-resonance gate is currently the most promising technology. It works by capactitively coupling two superconducting qubits of different resonant frequencies, and driving one qubit at the resonant frequency of the other. This results in a Hamiltonian term proportional to $\sigma_z \otimes \sigma_x$, the first qubit being the one that is driven [10]. Other 2-qubit gates can be decomposed into single-qubit gates and the cross-resonance gate.

3.3. Sources of noise

In practice, current quantum computers do not exactly reproduce the theoretical results, since they are affected by noise. Unlike a classical bit, the state of a qubit is continuous: each of the degrees of freedom in the general state of a qubit $\alpha|0\rangle + \beta|1\rangle$ is continuous, and therefore can deviate a small amount from the intended value.

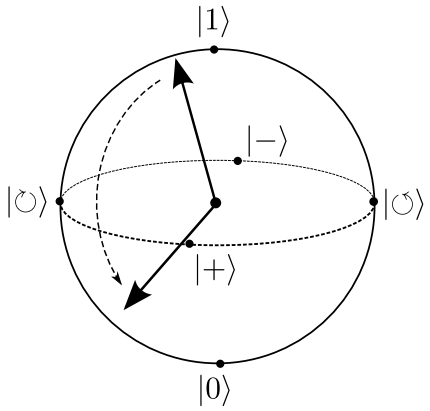
The noise will cause the qubit to become entangled with the environment, and thus ends up in a mixed state. Hence a density matrix can be used to describe the state of the qubit and how the noise affects it.

In its pure state, the qubit has a density matrix of $(\alpha|0\rangle + \beta|1\rangle)(\overline{\alpha}\langle 0| + \overline{\beta}\langle 1|)$ which becomes

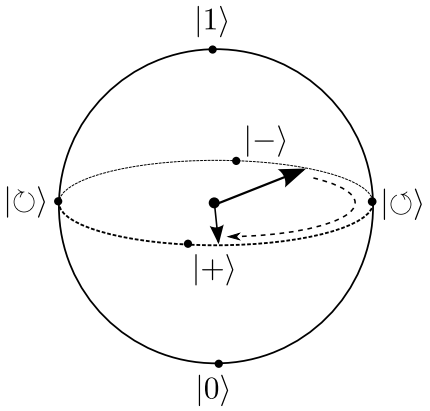
$$\begin{pmatrix} |\alpha|^2 & \alpha\overline{\beta} \\ \overline{\alpha}\beta & |\beta|^2 \end{pmatrix}$$

in matrix representation.

The domninant noise in superconducting qubits can be divided into two categories: latitudinal decay and longitudinal decay. These terms come from their effect on a state vector in the Bloch sphere.



(a) Latitudinal decay on the Bloch spere



(b) Longitudinal decay on the Bloch spere

Both these effects can be described using exponential distributions, with characteristic time T_1 for latitudinal decay and T_2 for longitudinal decay. This gives a new density matrix of

$$\begin{pmatrix} 1-(1-|\alpha|^2)e^{-t/T_1} & \alpha\overline{\beta}e^{-t/T_2} \\ \overline{\alpha}\beta e^{-t/T_2} & |\beta|^2 e^{-t/T_1} \end{pmatrix} \tag{3.15}$$

after time t. This is also known as the Bloch-Redfield model of decoherence.

Longitudinal decay is caused by amplitude dampening, which is due to energy exchange with the environment. Most likely, the qubit loses energy to the environment, so that a $|1\rangle$ state is reduced to a $|0\rangle$ state. Then T_1 is the time it takes for $1/e$ of qubits to decay.

Amplitude also causes longitudinal decay. There are also many other processes which induce longitudinal decay, together called dephasing. Dephasing happens due to the qubit becoming entangled to its environment, which may happen due to many effects such as interactions with the material that the qubit is made out of, or interactions with electromagnetic fields from the environment. The rate at which this happens is called the *pure dephasing rate* Γ_ϕ. Then $\frac{1}{T_2} = \frac{1}{2T_1} + \Gamma_\phi$.

In applying gates there can also occur errors. Firstly, there may be an error in the microwave frequency or the timing of the applied pulse, causing the phase applied to the qubit to be too small or large. These errors are systematic and can be found and corrected [9].

Multiple-qubit gates are especially sensitive to errors in the driving voltage, as an error in the phase can lead to the qubit states being rotated

Secondly, the microwave pulses applied to the qubit are short, and thus have a broad range of frequencies. This may accidentally drive a transition from $|1\rangle$ to $|2\rangle$, called a leakage error, which will cause an incorrect result at the readout. Longer pulses could be used to mitigate this effect, but then there is a higher chance of decay or decoherence during the pulse.

Errors can also occur if the process of measuring the qubit alters the state before the measurement, which is called a readout error [9].

3.4. Transmon Qubits

Qubits tend to be sensitive to charge noise. The effects of charge noise can be diminished by having the flux-dependent Josephson junction dominate over the charge-dependent capacitor, in other words, by designing the circuit such that $E_J >> E_C$. This can be achieved by making the capacitance of the capacitor much larger than the internal capacitance of the Josephson junction. A downside of this method is that it reduces the anharmonicity, as it depends on $\sqrt{E_C/E_J}$. This increases the chances of leakage errors. Nonetheless, the transmon qubit has been the most successful design as of now, notably all IBM quantum computers use this technology.

4

Quantum metrology

Metrology has become increasingly important in the current age of physics, as more and more accurate measurements are necessary to be able to corroborate hypotheses about the microscopic nature of the universe. However, any measurement device has some limit of measurement accuracy. In some cases though, it is possible to obtain a more accurate result by repeating the measurement many times. Then taking the average of these measurements gives a result with a smaller error. Treating the measurements as independent identically distributed random variables $X_1, \ldots, X_n$ whose expectation values are the true value to be measured, and with variance σ^2, the variance of the average $\bar{X}$ becomes

$$\mathrm{Var}(\bar{X}) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}(X_i) = \frac{\sigma^2}{n}$$

And thus, the standard deviation of the average becomes $\sigma/\sqrt{n}$. This shows that adding more measurements yields a more accurate result, but with diminishing returns due to the square root. This leads to the question if it is somehow possible to combine measurements in such a way that the variance decreases faster as a function of the amount of measurements.

If the measurements are independent, the answer is that this is not possible. So in order to achieve a better result, the measurements need to be dependent on each other. This cannot be achieved using measurements on classical states, since measuring one state does not affect other states. However, this is possible for entangled quantum states. If the value to be measured can somehow be encoded on a set of entangled probes, then measurements on these probes are dependent. Hence it could be possible for the error of the result to decrease faster than $1/\sqrt{n}$, where n in this case is the number of probes.

4.1. Measuring phase

One example where entangled states can produce more accurate measurements, is measuring the phase encoded in a quantum state.

Suppose that an unknown phase ϕ is added to a qubit system by applying the unitary $|0\rangle\langle 0| + e^{i\phi}|1\rangle\langle 1|$ to it, and we want to measure ϕ. This can be done by preparing the qubit in state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and applying the unitary to it, which yields $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\phi}|1\rangle)$. Measuring in the z-basis gives no information about ϕ, but measuring in the x-basis does, since in the x-basis the state is $\frac{1}{2}((1+e^{i\phi})|+\rangle + (1-e^{i\phi})|-\rangle)$. Then the probability of measuring $|+\rangle$ is $p_+ = |1+e^{i\phi}|^2 = \cos^2(\phi/2) = \frac{1}{2}(1+\cos(\phi))$. Then ϕ can be calculated as

$$\phi = \cos^{-1}(2p_+ - 1) \tag{4.1}$$

By repeating this experiment several times, p_+ can be estimated as the fraction of times that $|+\rangle$ was measured. If a more accurate result is needed, an obvious method is to increase the number of repetitions. However, this is exactly as in the classical situation described above, where in this case X_i is Bernoulli distributed with p_+ chance of $|+\rangle$. Hence the error only decreases with the square root of the number of

repetitions. The error in ϕ scales linearly with that of p_+, since, using error propagation, we have the relation $\sigma_\phi = \left|\frac{\mathrm{d}\phi}{\mathrm{d}p_+}\right|\sigma_{p_+}$ for any fixed value of p_+, Hence the error of ϕ also decreases with a square root.

However, in this case entangled states can be used to achieve a better result. Suppose the process was applied on n entangled two level systems in the GHZ state: $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n})$. Then after the process, the state is $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + e^{in\phi}|1\rangle^{\otimes n})$. In this case, $n\phi$ can be measured in the same way as ϕ before, except now p_+ is the probability of $|+\rangle^{\otimes n}$. This gives a better measurement result than the unentangled case: suppose N qubits can be used for each method. For the unentangled case, where each qubit is measured individually, this gives a standard deviation which scales with $1/\sqrt{N}$. For the entangled case, this means repeating the experiment N/n times since each measurement involves n qubits. Then since $n\phi$ is measured with the same accuracy as ϕ was in the unentangled case, the entangled measurement results in an error reduced by a factor n. Hence this error scales as $\frac{1}{n\sqrt{N/n}} = 1/\sqrt{nN}$, a factor $\sqrt{n}$ better than the unentangled case.

Something to take into account is that $n\phi$ is only determined modulo 2π. If ϕ is small, or if it is known that ϕ is close to some known value, this is not an issue. If this is not the case, measurements with smaller values of n can be made to get a rough estimate of ϕ, and then larger values of n can be used to increase precision. Furthermore, the above method only works for $n\phi \in [0, \pi]$ due to the inverse cosine. Measuring in the y-basis replaces the cosine by a sine, so the value of $n\phi \in [0, 2\pi]$ can be calculated with an arctangent. The errors still scale the same way in this method.

4.2. Heisenberg limit

Similarly to the classical case, there is a limit to how fast the error can decrease with n. This is known as the Heisenberg limit, and states that the error can at most decrease as $\frac{1}{n}$ [7]. Although it was not discovered by Heisenberg himself, the bound is derived from a generalized Heisenberg uncertainty principle.
In practice, the amount of probes is limited, as it is difficult to entangle many particles and leave their states undisturbed. Hence if the necessary precision is not achieved this way, the entangled measurement can be repeated N times to achieve a total bound of $\frac{1}{n\sqrt{N}}$

4.3. POVMs

As discussed in section 4.1, one can measure in different bases than the z-basis. While the quantum computer can only measure in the z-basis, gates can be used to change the coefficients of the state in the computational basis, so that they match the coefficients of the original state in the alternative basis. For example, a measurement in the x-basis can be realized by mapping the $|+\rangle$ state to $|0\rangle$ and $|-\rangle$ to $|1\rangle$, giving a transformation matrix of $|0\rangle\langle+| + |1\rangle\langle-|$, which turns out to be the Hadamard gate. In this way, it is possible to measure by projecting onto any orthogonal basis. However, more general, non-projective measurements are possible. In general, a measurement is any process where some quantity is extracted from a state, where the value of this quantity is drawn from a set of possible outcomes. For every possible outcome m, there is a measurement operator M_m, such that measuring $|\psi\rangle$ results in outcome m with probability $p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle$. Since the probability of getting any outcome must be 1, the completeness relation

$$\sum_m M_m^\dagger M_m = I \tag{4.2}$$

holds.

This leads to the definition of a positive operator valued measure: it is a set of positive operators $\{F_m\}$ that sum to I. Since every positive operator F_m can be factored as $M_m^\dagger M_m$ for some operator M_m, the POVM constitutes a complete set of measurements, with the probability of finding m being $p(m) = \langle\psi|F_m|\psi\rangle$. This characterizes the most general way to perform measurements on quantum states. [15]

4.4. Measurements in the Presence of Noise

So far, only ideal measurement processes have been considered. In the noiseless case the GHZ state is optimal, as it saturates the Heisenberg limit. But this state is also very sensitive to noise: only the sum of the phases applied on the $|1\rangle$ state is stored in the state, so if there is an outlier, it will offset the average phase and cannot be corrected. One way to potentially mitigate this is by choosing a probe state that

is less sensitive to this noise. Doing this theoretically will be a challenge, due to the intricacies of noise. However, since quantum computers exhibit noise themselves, it could be possible to optimize the amount of information about encoded parameters in a state, while automatically accounting for noise. In order to optimize over the possible probe states, a variational quantum algorithm can be used, which the next chapter will elaborate on.

5

Variational Quantum Algorithms

Variational quantum algorithms (VQAs) are algorithms that combine classical optimization algorithms with quantum computations that have low circuit depth and do not require many qubits. These algorithms seek to find a state that minimizes some cost function. The optimal state then encodes the solution. VQAs have many applications.

Many applications are directly related to quantum mechanics, such as the one that will be described in this work. Another example is the variational eigensolver, which can find the ground state and energy of a Hamiltonian, which is of use for systems which are too difficult to analyse analytically.

Other applications include quantum approximate optimization algorithms, which can find approximate solutions for combinatorial optimization problems such as the travelling salesman problem [14].

5.1. Overview of the algorithm

Variational quantum algorithms are useful when a problem can be solved by minimizing a cost function, and this cost function and its gradients can be computed efficiently on a quantum computer, while being impractical on a classical computer. In practice, this means that the cost function consists of expectation values of certain observables. The expectation values can be computed on the quantum computer, from which a classical computer can do calculations with these values to obtain the cost function if necessary. Then the parameter shift rule can be used to calculate the gradients, which will be explained in more detail in the next chapter. Firstly, one needs to design a parametrized circuit. On the one hand, it is desirable for the circuit to be able to produce as many states as possible, as this allows more freedom to reduce the cost function as much as possible. However, this comes at the cost of having many parameters, which will make optimization more costly.

Once a circuit has been designed, an initial state of the parameters has to be chosen. The cost function of this state and its gradients can be calculated on the quantum computer. Then a classical computer is used to calculate the next iteration of parameters, typically using some form of gradient descent. Then the cost and gradients at this new state can be calculated to to calculate the following iteration of parameters and so forth. Using this method, the cost function will approach a local minimum. It is not guaranteed to be a global minimum, so if something is known about where the global minimum could roughly be, it is best to choose the initial state in that neighborhood [4].

5.2. Optimizing Probe State with Fisher Information

As discussed in section 4.4, a variational quantum algorithm can be used to find an optimal probe state for measuring parameters in the presence of noise. The probe state will be generated by a parametrized circuit $U(\boldsymbol{\theta})$, giving a probe state $\left|\psi_{\mathsf{probe}}(\boldsymbol{\theta})\right\rangle$. This state will then undergo some process that encodes the parameters $\boldsymbol{\phi}$ on it, which are the parameters we are interested in measuring. The state after this process will be denoted $|\psi(\boldsymbol{\theta}, \boldsymbol{\phi})\rangle$ (or $|\psi(\boldsymbol{\phi})\rangle$ if $\boldsymbol{\theta}$ is fixed). In order to find the optimal state, we need a cost function that represents how well the $\boldsymbol{\phi}$ parameters can be measured based on the probe state.

The measurement outcomes are governed by some probability distribution which depends on the parameters to be measured. Thus in order to measure the parameters, we need an estimator that estimates them

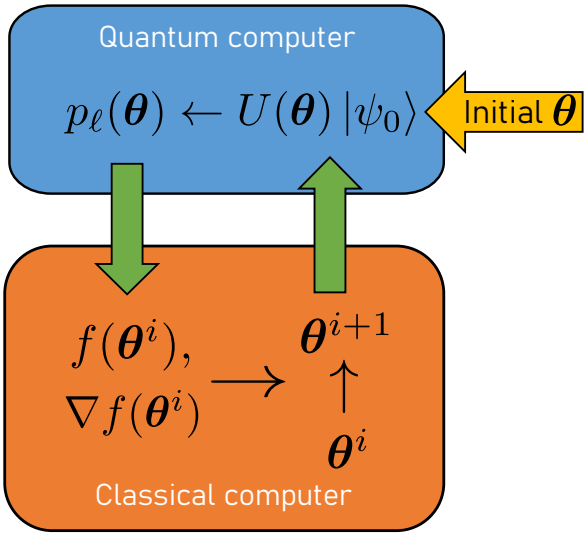


Figure 5.1: Overview of a VQA algorithm. f is the cost function, $U(\boldsymbol{\theta})$ is the parametrized circuit and p_ℓ are the POVM outcomes

from a set of samples. Then a measure of the variance of such an estimator could serve as a cost function. The Fisher information gives a method to achieve this.

The Fisher information is a way of quantifying how well some parameter in a probability distibution can be estimated. If there are multiple parameters to be estimated, say m parameters in $\boldsymbol{\phi}$, the Fisher information will be an $m \times m$ matrix. The matrix element at position (i, j) can be interpreted as a measure for how well parameter ϕ_i can be estimated based on the value of parameter ϕ_j, or vice versa. The probability distribution from which $\boldsymbol{\phi}$ is to be estimated, is that of the outcomes of the measurements from the quantum computer. These measurements can in general be represented by a POVM $\{F_\ell\}$, which results in a distribution

$$f(\ell; \boldsymbol{\phi}) = \langle \psi(\boldsymbol{\phi}) | F_\ell | \psi(\boldsymbol{\phi}) \rangle \tag{5.1}$$

The components of the fisher information are defined as [6]

$$[I_{\boldsymbol{\phi}}]_{jk} = \mathbb{E}\left[\left(\frac{\mathrm{d}}{\mathrm{d}\phi_j} \log f(X; \boldsymbol{\phi})\right)\left(\frac{\mathrm{d}}{\mathrm{d}\phi_k} \log f(X; \boldsymbol{\phi})\right)\right]$$

where X is a random variable for the value of ℓ with a distribution of $f(\ell; \boldsymbol{\phi})$.

If we fix some value of $\boldsymbol{\phi}$, it is convenient to write $p_\ell = f(\ell; \boldsymbol{\phi})$, such that the Fisher information becomes

$$\begin{aligned}[I_{\boldsymbol{\phi}}]_{jk} = \mathbb{E}\left[\left(\frac{\mathrm{d}}{\mathrm{d}\phi_j} \log f(X; \boldsymbol{\phi})\right)\left(\frac{\mathrm{d}}{\mathrm{d}\phi_k} \log f(X; \boldsymbol{\phi})\right)\right] = \\ \sum_\ell \frac{\frac{\mathrm{d}}{\mathrm{d}\phi_j} p_\ell}{p_\ell} \cdot \frac{\frac{\mathrm{d}}{\mathrm{d}\phi_k} p_\ell}{p_\ell} \cdot p_\ell = \sum_\ell \frac{(\frac{\mathrm{d}}{\mathrm{d}\phi_j} p_\ell)(\frac{\mathrm{d}}{\mathrm{d}\phi_k} p_\ell)}{p_\ell}\end{aligned} \tag{5.2}$$

Usually, one is interested in some (vector-valued) function of the parameters $\mathbf{g}$. Then the Fisher information $I_{\mathbf{g}}$ of $\mathbf{g}$ can be computed using the Jacobian J with $[J]_{jk} = \frac{\partial g_j}{\partial \phi_l}$:

$$I_{\mathbf{g}} = J^T I_{\boldsymbol{\phi}} J \tag{5.3}$$

The intuition of the fisher information indicating how well parameters can be estimated is made rigorous by the Cramér-Rao bound. If $\hat{\mathbf{g}}$ is an unbiased estimator of $\mathbf{g}$, the Cramér-Rao bound states

$$\mathrm{Cov}_{\boldsymbol{\phi}}[\hat{\mathbf{g}}] \geq I_{\mathbf{g}}^{-1} \tag{5.4}$$

Where the matrix inequality $\geq$ is not element-wise, but indicates that the difference of the matrices is positive semi-definite. This theorem gives a lower bound on the variances of the parameter estimates. Hence, lowering the Fisher information allows for better estimates [16].

If g is a scalar function, $I_{\mathbf{g}}^{-1}$ is also just a scalar and can be used as a cost function. If g is vector valued, then one could use $\mathrm{Tr}\big(WI_{\mathbf{g}}^{-1}\big)$ as a cost function, where W is a positive semi-definite weighting matrix. This is a viable cost function since $\mathrm{Tr}\big(W\mathrm{Cov}_{\boldsymbol{\phi}}[\hat{\mathbf{g}}]\big) \geq \mathrm{Tr}\big(WI_{\mathbf{g}}^{-1}\big)$, so the cost still represents the lower bound of some function of the (co)variances. W can be chosen as all ones if all entries of the inverse Fisher matrix are to be weighted equally. Alternatively, one could choose to favor decreasing variances of the individual parameters over decreasing covariances between them by choosing the off-diagonal elements to be smaller [13].

5.3. Maximum Likelihood Estimation

Once the optimal probe state has been found, it can now be used to estimate the parameters. Maximum likelihood estimators use a set of samples from the distribution with the true parameters to estimate those parameters. This is done by calculating the parameters that maximize the probability of the samples being the outcomes of the distribution. In this case, each sample from the distribution is the measurement of one of the outcomes of the POVM $\{F_\ell\}$. The probability of finding outcome ℓ under the assumption that $\boldsymbol{\phi}$ are the parameters of the distribution is $\langle\psi(\boldsymbol{\phi})|F_\ell|\psi(\boldsymbol{\phi})\rangle =: p_\ell(\boldsymbol{\phi})$. If there are N samples $\{\ell_i\}_{i=1}^N$, the likelihood is then defined as

$$\mathcal{L}(\boldsymbol{\phi};\{\ell_i\}) = \prod_{i=1}^{N} p_{\ell_i}(\boldsymbol{\phi}) \tag{5.5}$$

The value of $\boldsymbol{\phi}$ that maximizes $\mathcal{L}$ given the samples is the maximum likelihood estimator. To find this maximum, it is easier to consider of the logarithm of the likelihood. It has its maxima at the same points and thus gives the same estimator, but it is easier to take the derivatives. Furthermore, since there is only a finite number of possible outcomes, this can be rewritten in a more convenient form. If we define $C(\ell)$ as the number of counts of ℓ out of the N samples of the true distribution, the log likelihood becomes

$$\log\mathcal{L}(\boldsymbol{\phi},\{\ell_i\}) = \sum_{\ell} C(\ell)\log(p_\ell(\boldsymbol{\phi})) \tag{5.6}$$

which has gradients

$$\frac{\partial}{\partial\phi_j}\log\mathcal{L}(\boldsymbol{\phi},\{\ell_i\}) = \sum_{\ell} C(\ell)\frac{\frac{\partial}{\partial\phi_j}p_\ell(\boldsymbol{\phi})}{p_\ell(\boldsymbol{\phi})} \tag{5.7}$$

Then using gradient ascent, the maximum likelihood estimator for $\boldsymbol{\phi}$ can be approximated.

5.4. Optimization

For the gradient descent and ascent, a backtracking line search algorithm can be used. In this method, the direction of the optimization step is determined first, which is usually minus the gradient of the cost function, although some methods also take into account the history of previous steps. Then an imaginary line is drawn from the current point in parameter space in the direction of the step, with a slope that is a factor c smaller than the actual slope of the cost function in that direction. Then the step size is repeatedly reduced by a factor of α until the new cost is below the imaginary line. Figure 5.2 depicts graphically how this works.

Formally, let $\mathbf{x}$ be the current parameters $\mathbf{p}$ be the direction of the next optimization step. Then n starting at 0 will be increased until

$$f(\mathbf{x}-\alpha^n\mathbf{p}) < f(\mathbf{x}) - c\alpha^n\langle\nabla f(\mathbf{x}), p\rangle$$

holds, which then makes $\mathbf{x}-\alpha^n\mathbf{p}$ the updated parameters [2].

Under normal circumstances, this method guarantees that the cost function decreases with every optimization step. However, due to the fact that the cost function can only be approximated, this is not the case anymore. Nonetheless, this method provides a good method of determining a suitable step size, and allows for various step sizes in different stages of optimization, as large jumps can be made at the start but smaller steps are required when approaching a minimum.

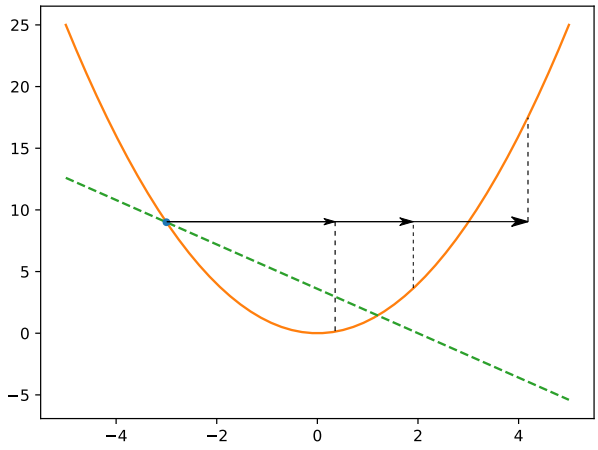


Figure 5.2: Diagram of backtracking line search in one dimension. The parabola is the cost function, and the dot indicates the current position. The dotted line is indicates the maximum cost of the next iteration. The arrows indicate potential step sizes, where only the smallest arrow satisfies the condition.

6

Parameter Shift Rule

In order to optimize a cost function using gradient descent, it is necessary to calculate the gradient of the cost function with respect to the optimization parameters. In general, there are several methods of doing this. The first is explicitly expressing the cost function as a function of the parameters and taking the partial derivatives, either by hand or using a computer algebra system. This becomes quite costly if the circuit size and amount of parameters becomes large, and is not flexible, as one needs to do the work again if a different circuit design is chosen. The second option is numerical differentiation. This approach only has limited success in quantum applications, since the difference in cost between two points nearby in parameter space may be dominated by noise, even in an ideal quantum computer there is statistical noise. To mitigate this, one could choose points further away from each other, thereby decreasing the accuracy of the numerical derivative, or increasing the amount of shots for calculating the cost function, which increases computation time.

There is, however, a third method that allows one to estimate the exact values of the gradients of the cost function. This method gives an exact formula for evaluating the gradient of the cost function with respect to a parameter in terms of the cost function evaluated at points with the parameter shifted, hence its name: the parameter shift rule.

6.1. Parameter Shift Rule

The cost function will depend on a parametrized state. This state is generated by a parametrized circuit: $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|\psi_0\rangle$.

For now, suppose the circuit depends on a single parameter. Parametrized quantum gates of the form $U(\theta) = e^{-i\theta G}$ will be considered, where the *generator* G is a hermitian matrix.

The parameter shift rule works for cost functions in the form of the expectation value of some observable H, so the cost function f can be expressed as

$$f(\theta) = \langle\psi_0|U^\dagger(\theta)HU(\theta)|\psi_0\rangle \tag{6.1}$$

Since $\frac{\mathrm{d}}{\mathrm{d}\theta}U(\theta) = -iGU(\theta)$ and $U(0) = I$, G can be calculated as

$$G = i\,\frac{\mathrm{d}}{\mathrm{d}\theta}U(\theta)\Big|_{\theta=0} \tag{6.2}$$

Depending on the properties of G, it is possible in many cases to derive a parameter shift rule. For example, if $G^2 = aI$ for some $a \in \mathbb{R}$, as is the case for the rotation gates which have the Pauli matrices divided by 2 as generators, we have the following rule [11]:

$$\frac{\mathrm{d}}{\mathrm{d}\theta}f(\theta) = \frac{1}{2\sin(\alpha)}(f(\theta+\alpha) - f(\theta-\alpha)) \tag{6.3}$$

for any $\alpha \in \mathbb{R}$ except multiples of π. The specific case of $\alpha = \frac{\pi}{2}$ turns out to be the optimal value of α to minimize variance [1]. However, it may also be advantageous to use the freedom of choosing α to reduce

the amount of evaluations of $f(\theta)$ necessary. For instance, choosing $\alpha = -\theta$ always reduces the first term to $f(0)$. This value could be computed once beforehand, hence if many gradients need to be calculated, this method halves the amount of evaluations. This does not work if θ is a multiple of π, as the parameter shift formula is not valid for the corresponding value of α, but in this case $\alpha = \frac{\pi}{2} - \theta$ can be chosen, reducing the first term to $f\left(\frac{\pi}{2}\right)$. In general, the variance of this method will be larger the closer α is to a multiple of π, so the choice of α should be such that it is closest to $\frac{\pi}{2}$. If the number of gradients to be computed is very large, it could be worthwhile to precompute several values of $f(\theta)$, to be able to choose values of α closer to $\frac{\pi}{2}$. This method of precomputing values could introduce systematic errors: the precomputed values of $f(\theta)$ are not exact due to being evaluated on a quantum computer, but are reused many times. Hence it is advisable to compute these values with more shots, reducing the chance that the evaluation is significantly different from the true value. Furthermore, if one can calculate multiple values of interest from a single quantum computation, such as the outcome probabilities of orthogonal projectors, it still requires fewer quantum computations to compute their gradients using a fixed α, in which case $\frac{\pi}{2}$ is the preferable choice.

For many 2-qubit gates the generator is not involutory, but it does have the property $G^3 = aG$ for some $a \in \mathbb{R}$, notably controlled rotation gates. In this case, there is the following rule [1]:

$$\frac{\mathrm{d}}{\mathrm{d}\theta} f(\theta) = d_1(f(\theta+\alpha) - f(\theta-\alpha)) + d_2(f(\theta+\beta) - f(\theta-\beta))) \tag{6.4}$$

with $d_1 \sin(\frac{\alpha}{2}) - d_2 \sin\left(\frac{\beta}{2}\right) = \frac{1}{4}$ and $d_1 \sin(\alpha) - d_2 \sin(\beta) = \frac{1}{2}$.
For this parameter shift rule, the minimum variance is achieved for $d_1 = \frac{\sqrt{2}+1}{4\sqrt{2}}$, $d_1 = \frac{\sqrt{2}-1}{4\sqrt{2}}$, $\alpha = \frac{\pi}{2}$ and $\beta = \frac{3\pi}{2}$

This parameter shift rule suffices for most applications: the most common single parameter gates all have a generator that satisfies $G^3 = aG$. Furthermore, if a gate satisfies this property, then so does a controlled version of it. If $M(\theta) = e^{-i\theta G}$ is some gate with $G^3 = aG$, then the controlled gate is $CM(\theta) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes M(\theta)$. This has generator $G_{CM} = i\frac{\mathrm{d}}{\mathrm{d}\theta} CM(\theta)\big|_{\theta=0} = |1\rangle\langle 1| \otimes i\frac{\mathrm{d}}{\mathrm{d}\theta} M(\theta)\big|_{\theta=0} = |1\rangle\langle 1| \otimes G$, so $CM^3 = |1\rangle\langle 1|^3 \otimes G^3 = |1\rangle\langle 1| \otimes aG = aG_{CM}$. Inductively, this also applies to controlled-controlled gates and so forth.

6.2. Multiple Gates

If the circuit consists of multiple parametrized gates, and we want to take the partial derivative of the cost with respect to one of these parameters, the situation isn't much more complicated. Then the new circuit becomes

$$U(\boldsymbol{\theta}) = U_N(\theta_N) \cdot \ldots \cdot U_1(\theta_1) \tag{6.5}$$

with cost function

$$f(\boldsymbol{\theta}) = \langle \psi_0 | U(\boldsymbol{\theta})^\dagger H U(\boldsymbol{\theta}) | \psi_0 \rangle \tag{6.6}$$

If we want to take the derivative with respect to θ_i, we can set $|\psi_{i-1}\rangle = U_{i-1}(\theta_{i-1}) \cdot \ldots \cdot U_1(\theta_1) |\psi_0\rangle$ and $H' = (U_N(\theta_N) \cdot \ldots \cdot U_{i+1}(\theta_{i+1}))^\dagger H (U_N(\theta_N) \cdot \ldots \cdot U_{i+1}(\theta_{i+1}))$. Then f can be rewritten as

$$f(\boldsymbol{\theta}) = \langle \psi_{i-1} | U_i(\theta_i)^\dagger H' U_i(\theta_i) | \psi_{i-1} \rangle \tag{6.7}$$

Neither H' nor $|\psi_{i-1}\rangle$ depends on θ_i, so the parameter shift rule can simply be applied by only shifting the θ_i component of $\boldsymbol{\theta}$ according to the parameter shift rule that applies to $U_i(\theta_i)$.

7

Experimental Setup

As discussed in chapter 5, the Fisher information can be used to quantify how well a function of parameters can be estimated from the outcomes of measuring a certain state. Hence if a certain experiment involves measuring some parameters that will be encoded on the quantum state of some probes, it could be worthwhile to investigate what state these probes need to be prepared in to make an optimally precise measurement.

The parametrized circuit used to prepare the probe state $|\psi(\boldsymbol{\theta})\rangle$ has 4 qubits and depends on 7 parameters.

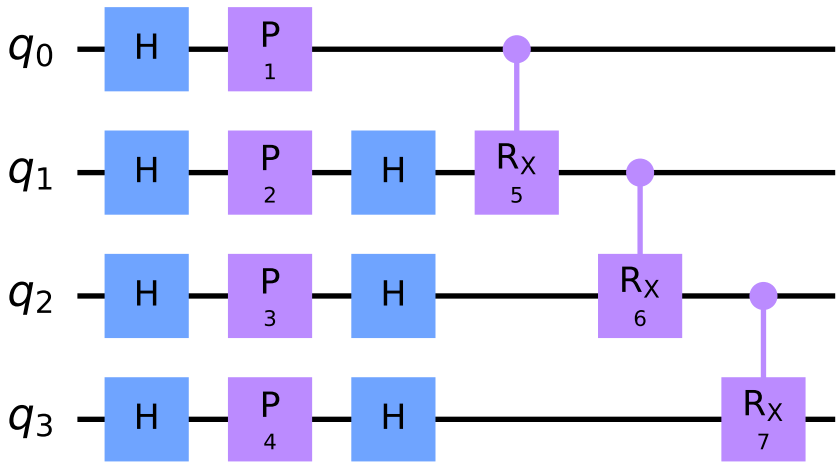


Figure 7.1: The circuit used for generating the probe state.

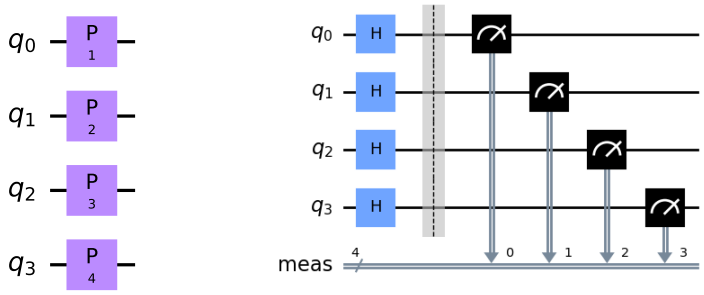
The layout can be seen in figure 7.1. The blue H gates are Hadamard gates. The purple gates are parametrized gates, with the number indicating which index in $\boldsymbol{\theta}$ controls that gate. The P gates are $|0\rangle\langle 0| + e^{i\theta}\,|1\rangle\langle 1|$, applying a phase of θ to the $|1\rangle$ state. The Rx gates with attached dots are controlled Rx gates, the dot representing the control qubit. These act as $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \mathrm{Rx}$, with $\mathrm{Rx}(\theta) = e^{i\theta\sigma_x/2}$, or

$$\begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$$

in matrix representation.

This circuit is designed such that it can generate the GHZ state for parameters $(0, 0, 0, 0, \pi, \pi, \pi)$, in order of the labels of the gates. It can also produce completely unentangled states whenever the parameters of the Rx gates are 0. The first Hadamard gates are there such that the P gates have effect, since all qubits start in the $|0\rangle$ state. The second layer of Hadamards make it such that the coefficients can also attain different amplitudes. The controlled Rx gates make sure that the qubits can be entangled.

For the encoding of the parameters, P gates are used to encode the parameters in the phases of the $|1\rangle$ states. Measurements are made in the x-basis, which is achieved using Hadamard gates as explained in section 4.3.



(a) Encoding circuit

(b) Measurement circuit

The parameters are encoded on the probe states using P gates on each qubit, see figure 7.2a . Figure 7.2b shows the measurement circuit. The Hadamard gates are used to transform to state into the x-basis, after which all qubits are measured.

The variational quantum algorithm described in chapter 5 will be run a IBM 5-qubit quantum computer by the name of Bogota, as well as on a simulator thereof. The true value of $\boldsymbol{\phi}$ will be $\pi/5 \cdot (1, 1, 1, 1)$, and the VQA will attempt to find the optimal state to measure the average of these parameters, so $g(\boldsymbol{\phi}) = \sum_{i=1}^{4} \phi_i/4$.

The algorithm was implemented in Python using the Qiskit package by IBM, which allows one to run circuits on simulators as well as their actual quantum computers. The simulators include ones that match their individual quantum computers as closely as possible, as well as ideal simulators. The used code can be found at `https://github.com/benc2/BachelorProjectVQA`.

8

Results

8.1. Measuring phase

Firstly, a verification of the method described in section 4.1 was performed, including a simulation with noise. This was performed with $N = 1024$, where for the entangled measurement method, the amount of entangled qubits n ranged from 2 to 10. The error was estimated by repeating the experiment 100 times for each n and calculating the sample standard deviation. The simulation of the noisy case was performed with the simulator of IBM's 27-qubit Paris quantum computer

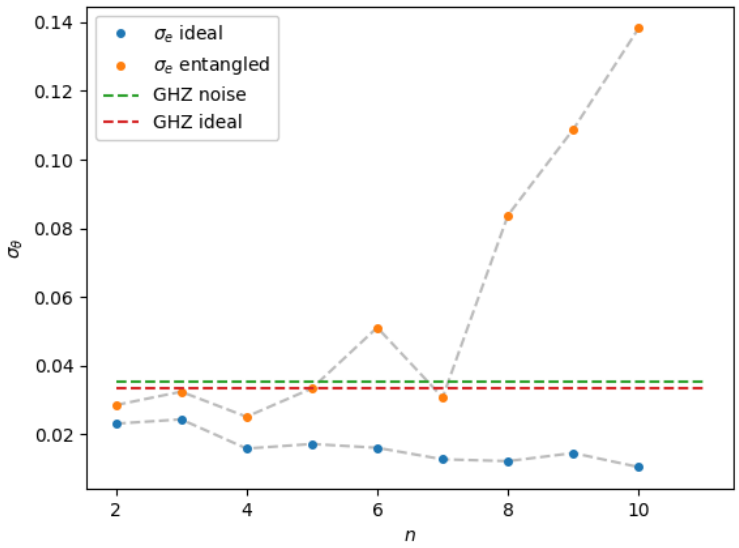


Figure 8.1: Estimates of the standard deviation for different amounts of probes n, for both ideal (decreasing line) and noisy (increasing line) simulations. The dotted lines are the estimated standard deviations for a single probe without entanglement, where the lower one is from an ideal simulation and the upper one from a noisy simulation

From 8.1 it can be seen that the noisy simulation yields high standard deviations. The noise, in this case probably mostly caused by the multiple-qubit gates, dominates over the beneficial effect of entangled measurements. However, for the noiseless case, the standard deviation does decrease with n. Whether it decreases at the same rate as in theory can be seen in figure 8.2

It can be seen that the ratio of the errors indeed follows a $1/\sqrt{n}$ as predicted in theory. There are deviations from the line, and the errors do not decrease monotonically. This can, however, be attributed to the variability in the estimate of the standard deviation.

The noisy case is quite different, as can be seen in more detail in figure 8.3, since the errors increase with n much faster than the inverse square root decreases. Hence this method of preparing a GHZ probe state

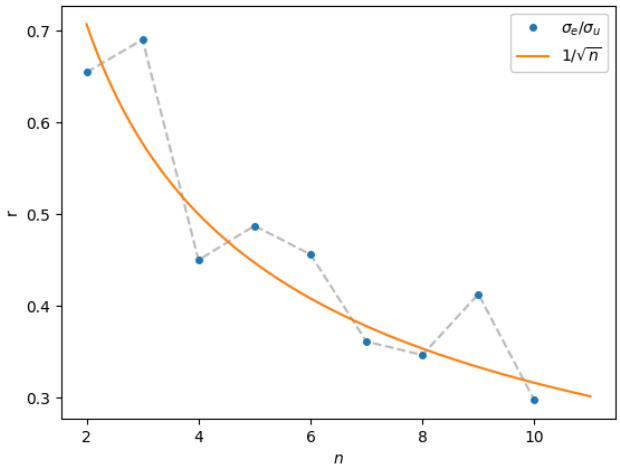


Figure 8.2: The ratio r of the errors of a measurement on n entangled measurements σ_e and an unentangled measurement σ_u, compared to a graph of $1/\sqrt{n}$

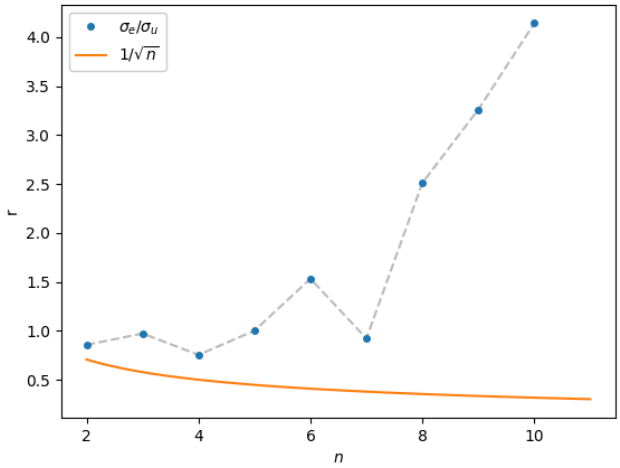


Figure 8.3: The same graph as the previous, but for the noisy simulations

on a quantum computer is not an effective method of improving measurement accuracy.

8.2. Optimal probe state

Firstly, the VQA was run on a simulator of the Bogota system. In figure 8.4, the cost function is plotted for each iteration of the gradient descent algorithm.

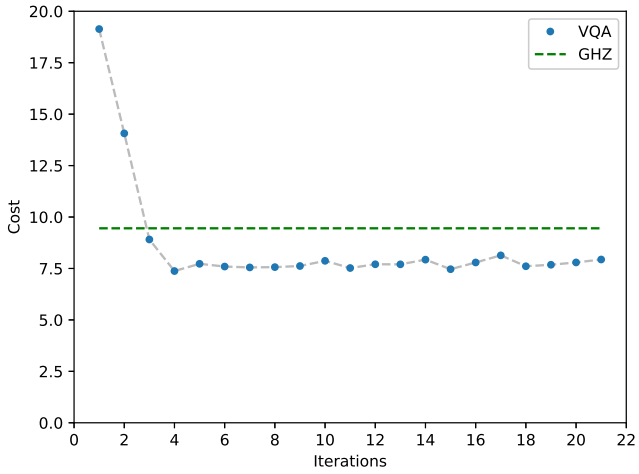
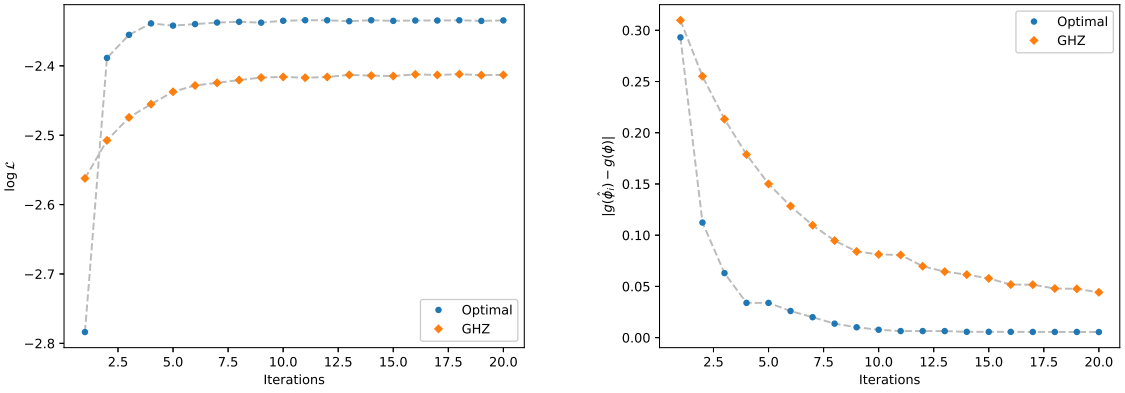


Figure 8.4: The cost function for each iteration. The dotted line indicates the cost of the GHZ state

Hence, in the noisy simulation, a better probe state than the GHZ state can be found using the VQA.

This shows that the probe state has a lower Fisher information, but in order to see whether it actually gives a better estimate, the results of the maximum likelihood estimation have to be compared.



(a) (b)

Figure 8.5: Results of the maximum likelihood maximization using a simulator. (a) shows the log likelihood of the estimates over the iterations for both the optimal probe state and the GHZ state, and (b) the difference between the true average value of the parameters and the estimated average value

From figure 8.5a it can be seen that the optimized state achieves a higher likelihood than the GHZ state, indicating that the optimized state indeed does a better job of estimating the parameters. Note that the log likelihood has been divided by the number of samples squared, this way the value does does not increase with the amount of samples. 8.5b shows that the optimized state actually achieves a better estimate of the average of the parameters than the GHZ state can.

The variational algorithm on the actual quantum computer was not able to improve upon the GHZ state, see figure 8.6

Due to limited allowed compute time on the IBM system, only a few iterations could be performed. In these iterations, the cost was not reduced below the cost of the GHZ probe state. The GHZ state has a

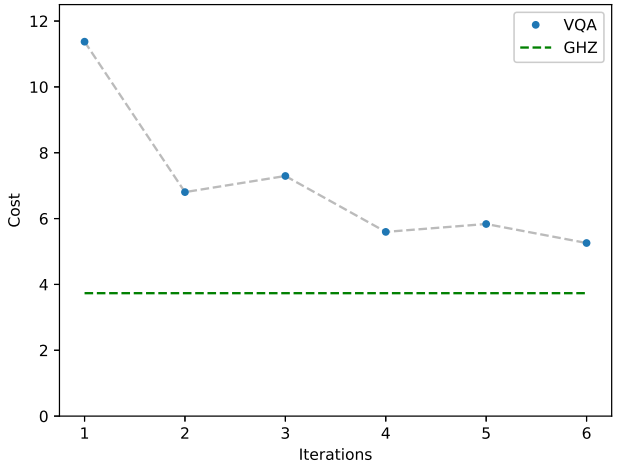
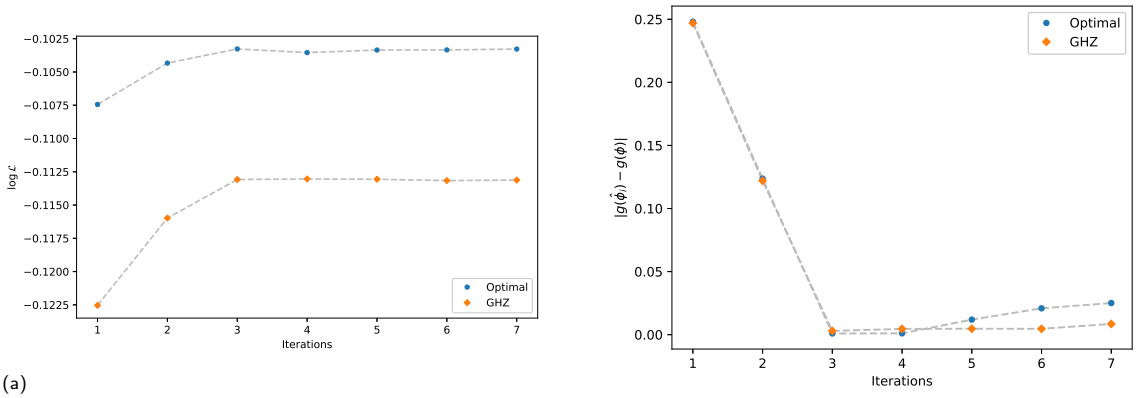


Figure 8.6: The cost function over the iterations for the VQA run on the actual IBM Bogota system. The dotted line indicates the cost of the GHZ state.

much lower cost in the quantum computer than in the simulation. This indicates that the simulation does not simulate the real system completely accurately, in fact, it overestimates the noise since the higher cost indicates less information about the parameters. Since there is less noise in this environment, it is more difficult to find a better state than the GHZ state.

The maximum likelihood estimators were also approximated on the quantum computer.



(a)

(b)

Figure 8.7: Results of the maximum likelihood maximization on the IBM Bogota system, the graphs indicate the same values as in the figures 8.5a and 8.5b.

In this case, the likelihood of the optimal state is higher than that of the GHZ state, despite having a lower cost, see figure 8.7a. There are a few factors that could contribute to this. The first is that the cost of the last state could be lower than indicated, and is higher due to noise. Noise makes it difficult to estimate the error on the cost, as the causes of noise are complex and it is not evident how the noise affects the outcome probabilities of any particular circuit. However, as an indication, a sample of 20 costs for the GHZ state results in a standard deviation of 0.35, which is much smaller than the gap, so this is not likely the cause. The cost of the Fisher information does not have a direct relationship with the likelihood though, so despite the lower likelihood, the GHZ state could still produce results that are more accurate.

The outcome estimates seem to be similarly accurate, as can be seen in figure 8.7b. Since the likelihoods of the optimal and GHZ states are closer to each other than in the simulation, the estimators are potentially not significantly different. In this case, the bigger factor might be the accuracy of the gradient ascent, which

for the optimal case worsens the estimate in later iterations.

Nonetheless, the fact that the cost function can be optimized on a real quantum computer is a promising result. Evaluating the cost function requires processing 9 circuits, and the gradient requires 109 circuits to be processed. Despite the fact that quantum computers are quite noisy and the inaccuracies in all the results from the circuits could add up to obtaining a result that is too inaccurate to use, the cost and gradient can still be evaluated accurately enough to successfully perform gradient descent. Hence variational quantum algorithms could potentially prove the be a useful tool in this era of noisy quantum computers.

9 Discussion

One major point of discussion is how applicable these results are to actual quantum measurements. Since the variational algorithm is run on the quantum computer, the probe state is optimized for the type of noise present in the quantum computer. This is effective if the probes experience the same type of noise, hence if the probes are also superconducting qubits. However, in real applications these probes are more likely to be particles such as photons. These could still experience the same types of noise as the qubits do, for example energy decay and dephasing due to interaction with the environment. However, the degree in which different causes of noise affect the photon will be different from the qubit, and hence it is not certain that this method will also generate an effective probe state for photons. Further research could indicate whether this method is effective under realistic circumstances.

Another issue about practical application is the effectiveness of the maximum likelihood estimators. In the presented method, the way to calculate the estimator is by gradient ascent, which requires many measurements of the system. Hence it might just be more efficient to perform many single qubit measurements, defeating the point of using the presented method. However, using entangled probes becomes relatively more effective with larger numbers of qubits. Thus for a large enough number of qubits, this process could be advantageous overall. Furthermore, with larger numbers of qubits, noise becomes a more significant factor, so finding a good probe state is more important. This could be relevant for the future when quantum computers are scaled up, where even with smaller error rates than current technology, the noise in for many entangled qubits causes the GHZ state to be suboptimal.

There is also a practical issue considering the Fisher information. It depends on the values of ϕ which are unknown, as they are the parameters to be measured. A possible solution is to use the GHZ state to find approximate value of ϕ, and then use this to compute the Fisher information. Assuming it is not too sensitive to changes in ϕ, a good probe state can still be found this way.

Furthermore, one could also consider optimizing the POVMs used to measure. This could be done by applying another parametrized circuit on the state after the parameters have been encoded. This has not been done here due to computation times. Potentially, choosing a more optimal POVM could increase measurement accuracy even further.

10

Conclusion

In this work, it is shown that variational quantum algorithms can be used to optimize a probe state for measuring parameters in a quantum computer simulation that performs better than the GHZ state, although this result could not be reproduced on an actual quantum computer. Maximum likelihood estimators were shown to be effective in obtaining the parameter values, where in the simulation the optimized probe state produced a more accurate result than the GHZ state did. This indicates that in sufficiently noisy environments, the presented variational quantum algorithm provides a method for obtaining more accurate measurements than the GHZ state. Furthermore, current quantum computers are shown to be accurate enough to perform optimization with variational quantum algorithms, hence they can provide a useful tool for practical applications of noisy quantum computers. Further research could indicate if optimizing over POVMs provides an even more accurate result, and if these results are also applicable to more realistic measurement scenarios.

Bibliography

[1] Gian-Luca R. Anselmetti et al. *Local, Expressive, Quantum-Number-Preserving VQE Ansatze for Fermionic Systems*. 2021. arXiv: 2104.05695 [quant-ph].

[2] Larry Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives." In: *Pacific Journal of Mathematics* 16.1 (1966), pp. 1–3. DOI: pjm/1102995080. URL: https://doi.org/.

[3] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. "Microscopic Theory of Superconductivity". In: *Phys. Rev.* 106 (1 Apr. 1957), pp. 162–164. DOI: 10.1103/PhysRev.106.162. URL: https://link.aps.org/doi/10.1103/PhysRev.106.162.

[4] M. Cerezo et al. *Variational Quantum Algorithms*. 2020. arXiv: 2012.09265 [quant-ph].

[5] P. A. M. Dirac. *The principles of quantum mechanics*. Oxford England: Clarendon Press, 1981. ISBN: 0-19-852011-5.

[6] R. A. Fisher. *On the Mathematical Foundations of Theoretical Statistics*. 1922. DOI: 10.1098/rsta.1922.0009. URL: https://doi.org/.

[7] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Quantum Metrology". In: *Phys. Rev. Lett.* 96 (1 Jan. 2006), p. 010401. DOI: 10.1103/PhysRevLett.96.010401. URL: https://link.aps.org/doi/10.1103/PhysRevLett.96.010401.

[8] A Yu Kitaev. "Quantum computations: algorithms and error correction". In: *Russian Mathematical Surveys* 52.6 (Dec. 1997), pp. 1191–1249. DOI: 10.1070/rm1997v052n06abeh002155. URL: https://doi.org/10.1070/rm1997v052n06abeh002155.

[9] P. Krantz et al. "A quantum engineer's guide to superconducting qubits". In: *Applied Physics Reviews* 6.2 (June 2019), p. 021318. ISSN: 1931-9401. DOI: 10.1063/1.5089550. URL: http://dx.doi.org/10.1063/1.5089550.

[10] Moein Malekakhlagh, Easwar Magesan, and David C. McKay. "First-principles analysis of cross-resonance gate operation". In: *Phys. Rev. A* 102 (4 Oct. 2020), p. 042605. DOI: 10.1103/PhysRevA.102.042605. URL: https://link.aps.org/doi/10.1103/PhysRevA.102.042605.

[11] Andrea Mari, Thomas R. Bromley, and Nathan Killoran. "Estimating the gradient and higher-order derivatives on quantum hardware". In: *Phys. Rev. A* 103 (1 Jan. 2021), p. 012405. DOI: 10.1103/PhysRevA.103.012405. URL: https://link.aps.org/doi/10.1103/PhysRevA.103.012405.

[12] N Mermin. *Quantum computer science : an introduction*. Cambridge, UK New York: Cambridge University Press, 2007. ISBN: 978-0521876582.

[13] Johannes Jakob Meyer, Johannes Borregaard, and Jens Eisert. *A variational toolbox for quantum multi-parameter estimation*. 2020. arXiv: 2006.06303 [quant-ph].

[14] Nikolaj Moll et al. "Quantum optimization using variational algorithms on near-term quantum devices". In: *Quantum Science and Technology* 3.3 (June 2018), p. 030503. DOI: 10.1088/2058-9565/aab822. URL: https://doi.org/10.1088/2058-9565/aab822.

[15] Michael Nielsen. *Quantum computation and quantum information*. Cambridge New York: Cambridge University Press, 2010. ISBN: 978-1-107-00217-3.

[16] C.R. Rao. *Information and the Accuracy Attainable in the Estimation of Statistical Parameters*. 1945.

[17] S. E. Rasmussen et al. *The superconducting circuit companion – an introduction with worked examples*. 2021. arXiv: 2103.01225 [quant-ph].