Advanced Quantum Algorithms

Giulia Ferrini, Anton Frisk Kockum, Laura García-Álvarez, Pontus Vikstål

December 29, 2020

Contents

Chapter 1

The circuit model for quantum computation

In this course, we will give an overview of various approaches to quantum computation, reflecting many of the latest developments in the field. We will cover several different models of quantum computation, from the foundational circuit model through measurement-based and adiabatic quantum computation to boson sampling. We will discuss quantum computation with both discrete and continuous variables. When it comes to the algorithms that we study, they include both classics like Shor's algorithm and newer, heuristic approaches like the quantum approximate optimization algorithm (QAOA). We will also see how quantum computing can be combined with machine learning.

We assume that the students taking this course already have some familiarity with quantum physics (superposition, entanglement, etc.) and some basic concepts in quantum computation. We will repeat some of these basic concepts at the beginning of the course, but perhaps give a more thorough justification for why they can be used in quantum computation.

In this first chapter, we will study the circuit model of quantum computation. This introduces quantum bits, quantum gates, and other components in close similarity with concepts in classical computing and gives us the tools to begin investigating whether quantum computers can ever outperform classical computers. For this chapter, we have borrowed parts from Refs. [Nielsen and Chuang, 2000, Aaronson, 2018, Kockum and Nori, 2019].

1.1 Components of the circuit model

Loosely speaking, a computation requires a system that can represent data, a way to perform manipulation of that data, and a method for reading out the result of the computation. In the circuit model of quantum computation, we use:

- **Quantum bits (qubits)** to represent the data.
- **State preparation** to initialize the qubits in the input state we need to begin the computation.
- **Quantum gates** on the qubits to manipulate the data.
- **Measurements** on the qubits to read out the final result.

Below, we first say a few words about what qubits are. We then discuss various quantum gates, and what is required of such gates to allow us to perform any quantum computation we would like. We assume

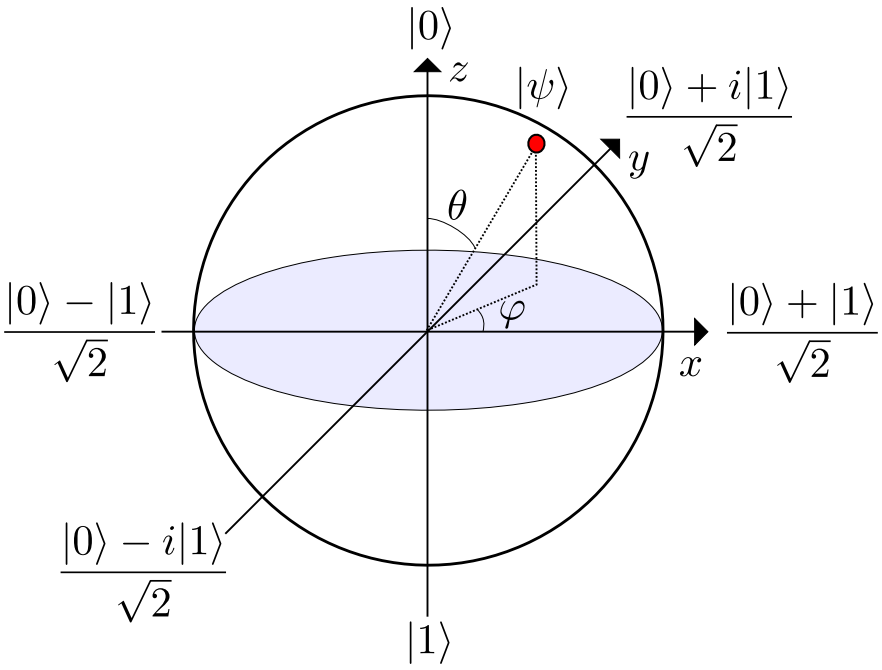


Figure 1.1: The Bloch-sphere representation of a qubit state. The north pole is the ground state $|0\rangle$ and the south pole is the excited state $|1\rangle$. To convert an arbitrary superposition of $|0\rangle$ and $|1\rangle$ to a point on the sphere, the parametrization $|\psi\rangle = \cos\frac{\vartheta}{2}|0\rangle + e^{i\phi}\sin\frac{\vartheta}{2}|1\rangle$ is used.

for now that it is possible to initialize our quantum computer in some simple state, and that we can read out the state of the qubits at the end of a computation.

1.2 Quantum bits

In a classical computer, the most basic unit of information is a *bit*, which can take two values: 0 and 1. In a quantum computer, the laws of quantum physics allow phenomena like superposition and entanglement. When discussing information processing in a quantum world, the most basic unit is therefore a *quantum bit*, usually called *qubit*, a two-level quantum system with a ground state $|0\rangle$ and an excited state $|1\rangle$. Unlike a classical bit, which only has two possible states, a quantum bit has infinitely many states: all superpositions of $|0\rangle$ and $|1\rangle$,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}, \tag{1.1}$$

where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. A measurement on this qubit state (in the basis of $|0\rangle$ and $|1\rangle$) gives the result 0 with probability $|\alpha|^2$ and the result 1 with probability $|\beta|^2$.

A useful tool for visualizing a qubit state is the *Bloch sphere* shown in Fig. 1.1. A state of the qubit is represented as a point on the surface of the sphere, which has radius 1. The two states of a classical bit correspond to the north and south poles on the sphere. The two states on opposite ends of the x axis are often denoted

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{1.2}$$

If there are N qubits in a system, the total state of that system can be a superposition of 2^N different states: $|000\ldots00\rangle$, $|100\ldots00\rangle$, $|010\ldots00\rangle$, ..., $|111\ldots10\rangle$, $|111\ldots11\rangle$. Note that N classical bits can be in *one* of these 2^N states, but not in a superposition of several of them. To store all the information about a general N-qubit state, one needs to keep track of the 2^N amplitudes in the superposition. This means that at least 2^N classical bits are required to represent the quantum system. This explains why it is hard for classical computers to simulate some classical systems, and gives a first hint that quantum computers can be more powerful than classical ones (at least when it comes to simulating quantum systems).

There are many physical implementations of qubits, e.g., superconducting qubits, trapped ions, natural atoms, etc. These implementations are a topic for another course. In the following, we assume that we have access to qubits, but do not care much about how they are made.

1.3 Single-qubit gates

Operations on a single qubit moves its state from $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ while preserving the norm $|\alpha|^2 + |\beta|^2 = 1 = |\alpha'|^2 + |\beta'|^2$. Such operations (gates) can be described by 2×2 unitary matrices. Here we list some of the most common ones. First are the Pauli matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{1.3}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{1.4}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1.5}$$

$$\tag{1.6}$$

The Pauli matrices generate rotations around the corresponding axes on the Bloch sphere when exponentiated, e.g.,

$$R_x(\vartheta) = \exp(-i\vartheta X/2) = \cos(\vartheta/2)I - i\sin(\vartheta/2)X = \begin{pmatrix} \cos(\vartheta/2) & -i\sin(\vartheta/2) \\ -i\sin(\vartheta/2) & \cos(\vartheta/2) \end{pmatrix}, \tag{1.7}$$

where I is the identity matrix.

The X gate is the quantum equivalent of the classical NOT gate:

$$X(\alpha|0\rangle + \beta|1\rangle) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \beta|0\rangle + \alpha|1\rangle. \tag{1.8}$$

By adding the X and Z gates, one obtains the Hadamard gate

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{X+Z}{\sqrt{2}}. \tag{1.9}$$

This gate transforms the qubit state from the $|0\rangle, |1\rangle$ basis to the $|+\rangle, |-\rangle$ basis. The Hadamard is often used to create superposition states at the beginning of a quantum algorithm.

The Z gate applies a phase factor -1 to the $|1\rangle$ part of the qubit state. Two gates that apply other phase factors are often given their own names. One is the T, or $\pi/8$, gate:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{pmatrix} = \exp(i\pi/8)\begin{pmatrix} \exp(-i\pi/8) & 0 \\ 0 & \exp(i\pi/8) \end{pmatrix}, \tag{1.10}$$

The other is the phase, or S, or P, gate

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = T^2. \tag{1.11}$$

1.4 Multi-qubit gates

To realize useful quantum algorithms, we also need to be able to make two or more qubits interact through multi-qubit gates. One way to achieve this is to let the state of one qubit control whether a certain single-qubit gate is applied to another qubit. One such gate is the controlled-NOT (CNOT) gate:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.12}$$

Note that the two-qubit Hilbert space is spanned by the basis vectors $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, in that order. This is the tensor product of the two single-qubit Hilbert spaces. The action of the CNOT gate is thus to do nothing if the first qubit is in state $|0\rangle$ ($|00\rangle$, $|01\rangle$ changes to $|00\rangle$, $|01\rangle$), and to apply NOT to the second qubit if the first qubit is in state $|1\rangle$ ($|10\rangle$, $|11\rangle$ changes to $|11\rangle$, $|10\rangle$).

The controlled-Z (CZ) gate can be defined in the same manner:

$$\text{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{1.13}$$

More generally, the controlled application of a single-qubit unitary U to the second qubit takes the form

$$\begin{pmatrix} I_2 & 0_2 \\ 0_2 & U \end{pmatrix}. \tag{1.14}$$

There are also two-qubit gates that are not controlled operations. For example, the SWAP gate

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.15}$$

swaps the states $|01\rangle$, $|10\rangle$ to $|10\rangle$, $|01\rangle$.

It is also possible to define gates involving than two qubits. For three-qubit gates, the most well-known ones are the Toffoli and Fredking gates. The Toffoli gate is a controlled-controlled-NOT (CCNOT), i.e., the state of the third qubit is flipped if and only if both the first two qubits are in state $|1\rangle$:

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.16}$$

The Fredkin gate is a controlled-SWAP (CSWAP) gate, swapping the states of the second and third qubits if and only if the state of the first qubit is $|1\rangle$:

$$\text{Fredkin} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1.17}$$

Most practical implementations of quantum computing have limited connectivity between qubits, only allowing for pairwise interactions between nearest-neighbour qubits. This can prohibit direct implementations of multi-qubit gates with three or more qubits, but it turns out that any multi-qubit gate can be decomposed into a number of single- and multi-qubit gates.

1.5 Universal quantum computation

Are all the gates we specified above enough to carry out any quantum computation that we would like? Could we do any quantum computation using just a small subset of the gates above? These are questions about *universality*.

For classical computers, a set of gates is called universal if, by applying enough gates from this set in a sequence, it is possible to express any Boolean function on any number of bits. The classical NAND gate turns out to be universal all on its own, but there are other sets of gates that are not universal. For example, the set $\{\text{AND}, \text{OR}\}$ is not universal, because these gates cannot express non-monotone Boolean functions; changing an input bit from 0 to 1 in a circuit with these gates will never result in an output bit changing from 1 to 0.

For quantum computers, a gate set is called *universal* if the gates therein can be used to approximate any unitary transformation on any number of qubits to any desired precision. To understand what makes a gate set universal for quantum computing, let us see how a gate set can fail to be universal:

- **Inability to create superposition states**
  Some gates, e.g., X and CNOT, only change states in the computational basis into other states in the computational basis (e.g, $\text{CNOT}\,|11\rangle = |10\rangle$). These gates can maintain superpositions, but they cannot create new ones.

- **Inability to create entanglement**
  The Hadamard gate can create a superposition (e.g., $H\,|0\rangle = |+\rangle$), but it only acts on a single qubit, so it cannot create entanglement between two or more qubits. Clearly, no single-qubit gate can. Since an unentangled state of N qubits can be written $(\alpha_1\,|0\rangle + \beta_1\,|1\rangle) \otimes \ldots \otimes (\alpha_N\,|0\rangle + \beta_N\,|1\rangle)$, it can be specified using only $2N$ amplitudes, which a classical computer would be able to simulate efficiently.

- **Inability to create non-real amplitudes**
  A gate set like $\{\text{CNOT}, H\}$ would be able to create both entanglement and superposition states. However, the matrices specifying these gates only have real entries. Thus, they would never be able to create states with complex amplitudes.

- **The Gottesman-Knill theorem**
  If we take our gate set to be $\{\text{CNOT}, H, S\}$, we circumvent all the objections above. However, it turns out that this still is not enough to achieve universal quantum computation. This is the

  Gottesman-Knill theorem: A quantum circuit using only the following elements can be simulated efficiently on a classical computer:

  1. Preparation of qubits in computational basis states,
  2. Quantum gates from the Clifford group (H, CNOT, S), and
  3. Measurements in the computational basis.

  The Clifford group is the group of operations that map Pauli matrices onto (possibly different) Pauli matrices, and is generated by the gates (H, CNOT, S). Quantum circuits with only these gates are also called stabilizer circuits (we will return to this concept in Chapter 3, where we will give a proof of the Gottesman-Knill theorem and discuss quantum error correction). If one starts in the computational basis, the single-qubit states that can be reached using these circuits are the ± 1 points on the x, y, and z axes of the Bloch sphere. This restriction of the state space turns out to make the circuit classically simulatable even though the state space has states with both superposition, entanglement, and complex amplitudes.

What is then a universal gate set? Actually, replacing the Hadamard gate in the last gate set considered above, $\{\text{CNOT}, H, S\}$, with almost any other gate makes the set universal. We can also replace S with some other gate. One universal gate set is $\{\text{CNOT}, H, T\}$. Almost any two-qubit gate on its own is also universal.

We thus now know that we can implement any unitary operation on N qubits in a realistic experimental architecture that allows for a few single- and two-qubit gates. Does that mean that we can run quantum algorithms that are faster than classical algorithms for some problems? This will be discussed in the next chapter (and many other chapters after that).

Chapter 2

Comparing quantum and classical computers

Could a quantum computer find the answer to a problem that *cannot* be solved on a classical computer, even if there are no restrictions on the resources available to the classical computer? The answer is no. The classical computer could always simulate the quantum computation by storing, to enough precision, the 2^N amplitudes for the whole state of the quantum computer's N qubits, and changing these amplitudes according to the gates applied in the quantum algorithm. However, there is clearly a possibility that this classical simulation requires a lot more resources than the quantum computation itself. So a better question to ask is: are there problems that a quantum computer could solve *faster* than a classical computer? That question is the topic of this chapter.

2.1 The Solovay–Kitaev theorem

In the previous chapter, we saw that there are many gate sets that are universal for quantum computing, i.e., they can approximate any unitary transformation on any number of qubits to any desired precision ε. However, it is a different question whether that approximation is fast and efficient. For example, If the number of gates needed would scale exponentially in $1/\varepsilon$, there would probably not be any practical use for such a universal gate set. Luckily, there is a theorem telling us that the scaling for any universal gate set is much more benign:

Solovay–Kitaev theorem: Let G a finite subset of $SU(2)$ and $U \in SU(2)$. If the group generated by G is dense in $SU(2)$, then for any $\epsilon > 0$ it is possible to approximate U to precision ϵ using $O\big(\log^4(1/\epsilon)\big)$ gates from G.

The proof can be found in Ref. [Nielsen and Chuang, 2000]. What the theorem essentially says is that if we have a gate set that can approximate any single-qubit rotation, then we only need relatively few gates from that set to do the approximation, so we can do the approximation fast. By adding some two-qubit gate to make the gate set universal, it can be shown that the total number of gates needed to approximate U on N qubits is at most $O\big(4^N \log^4(1/\epsilon)\big)$. More recent results have shown that for certain gate sets, the number of gates can be brought down from $O\big(\log^4(1/\epsilon)\big)$ to $O(\log(1/\epsilon))$. There are also good algorithms for finding the gate sequences needed to achieve the Solovay-Kitaev bound.

This theorem, together with the observations in the previous chapter, is what gives us confidence that

quantum computing has potential. We now know how to find a universal gate set to approximate any unitary transformation, and we know that we can implement that approximation efficiently. This means that we are ready to compare how fast classical and quantum computers are at solving different types of problems: we can define quantum complexity classes.

2.2 Complexity classes

The time it takes to solve a computational problem can scale differently with the problem size, and can depend on what type of computational machine is used. This is the basis for defining computational complexity classes, which are the topic of this section. Most of the following is based on the supplementary material of Ref. [Douce et al., 2017] and section 2 in Ref. [Wendin, 2017].

For classical computers, the scaling is usually defined either for a deteriministic Turing machine (DTM) or a probabilistic Turing machine (PTM). The DTM is the model that corresponds to ordinary classical computers; it is a finite state machine that reads and writes on a finite tape.

2.2.1 Complexity classes for a deterministic Turing machine

The most basic and well-known complexity classes are those defined for a DTM:

- P (polynomial): The set of decision problems solvable in polynomial (poly) time by a DTM.
- NP (non-deterministic polynomial): The set of decision problems whose solutions can be verified in polynomial time by a DTM.
- NP-hard: The set of decision problems whose solutions allows solving all problems in NP.
- #P: The set of problems that count the number of solutions of NP problems.
- #P-hard: The set of problems whose solutions allows solving all other problems in #P

It is known that P $\subseteq$ NP, but the question whether the inclusion holds strictly (and hence ultimately P $\neq$ NP) stands as one of the most important open problems in the modern age of science.

A concept that is often mentioned in connection to P and NP is the polynomial hierarchy (PH). It is a hierarchy of complexity classes that generalizes the classes P, NP to the case in which oracles are accessible. An oracle is a black box that can output the solution of a decision problem contained in a given complexity class using a single call. For example, A^{B} is the set of decision problems in class A that are solvable in polynomial time by a DTM augmented by an oracle for some complete problem in class B.

The first level of the PH is the class P; in symbols, $\Sigma_0 = \mathrm{P}$. Successive levels are refined recursively:

$$\Sigma_{k+1} = \mathrm{NP}^{\Sigma_k}. \tag{2.1}$$

A problem is in the polynomial hierarchy if it is in some Σ_k, i.e., the polynomial hierarchy is the union of all Σ_k.

Analogously to what was said above concerning the relation between P and NP, it is known that $\Sigma_k \subseteq \Sigma_{k+1}$, i.e., higher levels of the PH contain lower levels, and it is strongly believed that the inclusion is strict, namely that $\Sigma_k \neq \Sigma_{k+1}$. If there is a k for which $\Sigma_k = \Sigma_{k+1}$, the PH is said to collapse to level k. It can be shown that if a collapse occurs at level k then for all $k' > k$ it would hold that $\Sigma_{k'} = \Sigma_k$, which justifies the terminology "collapse".

2.2.2 Complexity classes for a probabilistic Turing machine

A PTM is much like a DTM, but it makes random choices of the state of the finite state machine when reading from the tape, and it traverses all states in a random sequence. This randomness means that a PTM will not get stuck away from a solution, which could happen for a DTM. This leads to the definition fo the following complexity classes:

- BPP (bounded probabilistic polynomial): The class of decision problems that a PTM solves in polynomial time with an error probability bounded away from (i.e., strictly less than) $1/3$ for all instances.
- PP (probabilistic polynomial): The class of decision problems that a PTM solves in polynomial time with an error probability bounded less than $1/2$ for all instances.

A PTM can be simulated by a DTM with only polynomial overhead. Therefore, it is believed that BPP = P.

2.2.3 Complexity classes for a quantum Turing machine

We can define a quantum Turing machine (QTM) in analogy with the classical Turing machines, using a quantum memory (tape) and a quantum processor. We then have a few new complexity classes:

- BQP (bounded quantum polynomial): This is the quantum analogue of BPP. Intuitively, BQP is the class of problems that can be solved using at most a polynomial number of gates, with at most $1/3$ probability of error.
- PostBQP (Post-selected bounded quantum polynomial): PostBQP is an extension of BQP where, during a polynomial time computation, one is allowed to abort and start all over again for free whenever the result on a specific conditioning qubit (or subset of qubits) is not satisfying.
- QMA (quantum Merlin–Arthur): Classically, Merlin–Arthur problems are a subclass of NP problems where the oracle is referred to as Merlin and Arthur is the one doing the verification in polynomial time. The QMA class is the one where Arthur can use a quantum computer to verify the solution in polynomial time.
- QMA-hard: The set of decision problems whose solutions allows solving all problems in QMA.

BQP it is the class we refer to when we talk about problems efficiently solved by a universal quantum computer. Note that we do not have to specify which gates the definition is based upon, as long as they constitute a universal set. Thanks to the Solovay-Kitaev theorem, using one universal set or another merely results in a polylogarithmic overhead; this cost is dominated by a polynomial function.

Quantum computing subsumes classical one. In terms of complexity classes, this is summarized by the statement BPP $\subseteq$ BQP.

More details on PostBQP

The post-selection procedure in PostBQP, which is not specific to quantum computing (one can define the classical complexity class PostBPP similarly), is highly unrealistic and brings in a lot of power to the model [Aaronson, 2005]. More formally, PostBQP is the class of problems solvable by a BQP machine such that:

- If the answer is yes, then the second qubit has at least $2/3$ probability of being measured 1, conditioned on the first qubit having been measured 1.

- If the answer is no, then the second qubit has at most $1/3$ probability of being measured 1, conditioned on the first qubit having been measured 1.
- On any input, the first qubit has a nonzero probability of being measured 1. This condition can actually be refined to an n-dependent probability.

Denoting q_o (q_c) the output (post-selected) qubit, the relevant mathematical object is the conditional probability which reads by definition:

$$P(q_o = 1/q_c = 1) = \frac{P(q_o = 1 \wedge q_c = 1)}{P(q_c = 1)}. \tag{2.2}$$

Intuitively, the power of PostBQP relies upon the denominator $P(q_c = 1)$: since it can be arbitrarily low, it may compensate for very unlikely events corresponding to the solution.

We now want to be more specific about the success probability $P(+_1)$. The Solovay-Kitaev theorem (see above) actually sets a lower bound on the acceptable probabilities: it lets us approximate any desired unitary within exponentially small error for only a polynomial increase in the circuit size. In other words, for an exponentially unlikely probability, the theorem still ensures that arbitrary universal gate sets can be used for polynomially long computations like BQP circuits—since a polynomial overhead remains in the BQP class. And indeed the class PostBQP is based upon BQP circuits. Thus it is well-defined only if the relevant output probabilities are at worst exponentially unlikely:

$$P(+_1) \gtrsim \frac{1}{2^n}. \tag{2.3}$$

It has been shown in [Kuperberg, 2015] that this condition was fulfilled whenever "reasonable" universal gate sets were considered.

Additionally, suppose now that there is a polynomial $p(n)$ such that $P(+_1) \geq 1/p(n)$. In that case $P(+_1)$ is polynomially unlikely. Then running the BQP circuit $p(n)$ more times would still correspond to a polynomial time computation and remain in BQP. On the other hand, such redundancy would enable recording enough statistics to simulate the quantum post-selection through classical postprocessing. Hence conditioning on an event which probability scales as $1/p(n)$ does not give any power to the post-selection. So $P(+_1)$ has to be worst than polynomially unlikely.

Following the discussion in [Aaronson, b], the definition of the class PostBQP could be refined to account for this feature: the conditioning probability $P(+_1)$ scales as the inverse of an exponential function,

$$P(+_1) \sim \frac{1}{2^n}, \tag{2.4}$$

up to some scaling factor irrelevant in terms of computational classes.

2.2.4 Summary of the complexity classes

The Venn diagram in Fig. 2.1 summarizes most of the complexity classes defined above and their relations. Another plot of the relationships between a few of the complexity classes is shown in Fig. 2.2.

Some of the definitions above, and a few more complexity classes, are given in 2.3. The definition of further complexity classes can be found in [Watrous, 2009, Aaronson, a].

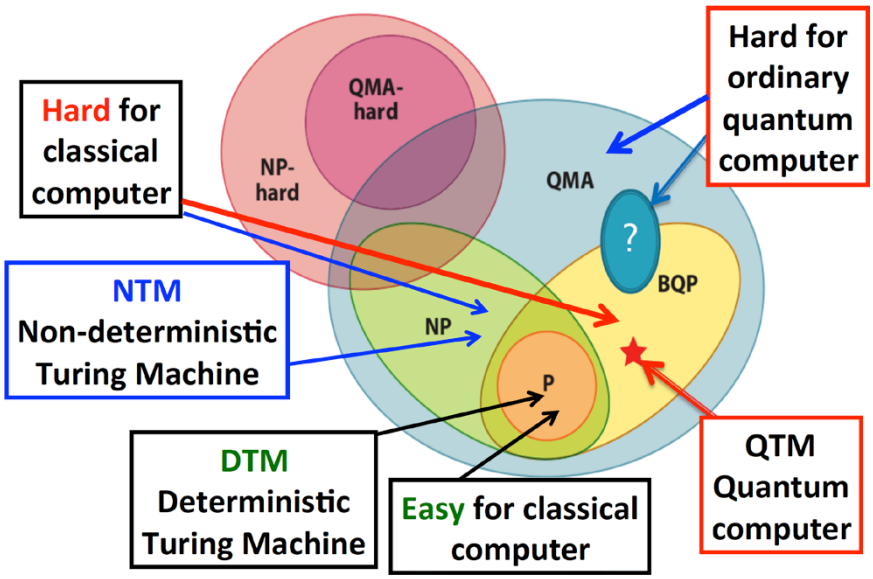


Figure 2.1: A Venn diagram of various classical and quantum complexity classes. From Ref. [Wendin, 2017].

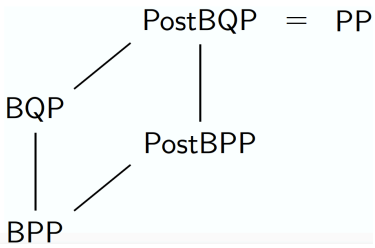


Figure 2.2: Main complexity classes useful for our purposes and the inclusion relationships (black line, inclusion from bottom to top).

Class	Type	Definition
P	*D*	Deterministic with polynomial runtime on a classical computer
EQP	*D*	Deterministic with polynomial runtime on a quantum computer
BPP	*D*	Random with classical statistics and an error probability less than 1/3
BQP	*D*	Random with quantum statistics and an error probability less than 1/3
NP	*D*	Outputs can be verified using an algorithm from P
PP	*D*	Random with classical statistics and an error probability less than 1/2
#P	*C*	Counts the number of 'accept' outputs for circuits from P
GapP	*Z*	Difference between the number of 'accept' and 'reject' outputs for circuits from P
PSPACE	*D*	Polynomial memory requirements on a classical computer

Note: The "Type" column describes the outputs generated by algorithms within the class. "*D*" denotes decision problems which output a single bit, whose values are often interpreted as 'accept' and 'reject'. "*C*" denotes counting problems which output a non-negative integer. "*Z*" denotes problems that generalise counting problems and allow negative integer outputs. The definitions give the properties algorithms are required to have within each class.

Figure 2.3: Summary of the definition of the complexity classes used in this chapter. From Ref. [Lund et al., 2017]

Chapter 3

Quantum error correction

We have now seen that universal quantum computation can be performed with a rather small and simple set of quantum gates, and that there is good reason to believe that a universal quantum computer will be able to solve some problems faster than classical computers can. However, a crucial question remains to answer before we go ahead and invest large resources into actually building a quantum computer: can a quantum computer deal with errors?

If even a small error can wreck a quantum computation beyond repair, a practical quantum computer can never be realized. In today's classical computers, the probability p of an error occurring during a logical operation is amazingly low. Numbers on the order of $p \approx 10^{-18}$ are often quoted. For state-of-the-art quantum computers, however, p is rather on the order of 10^{-2} or, in the best case, 10^{-3}. In this chapter, we show that quantum error correction is indeed possible and feasible, even with error rates close to what we have today. For a more detailed account, se chapter 10 in Ref. [Nielsen and Chuang, 2000].

3.1 Challenges for quantum error correction

Correcting for errors on classical bits is quite straightforward. The basic idea is to encode the state of one bit redundantly using several bits such that an error on one of the latter does not change the encoded information. The simplest example is a three-bit code with majority voting. The state of one bit, 0 or 1, is encoded into three bits as 000 or 111. The encoded information is read out by measuring all three bits and going with the majority vote. For example, if the third bit is flipped, 000 changes into 001 (and 111 into 110), but the majority vote among 001 (110) still tells us that the encoded bit was 0 (1). For the error correction to fail, two bits need to be flipped. This means that while the error probability for the single unencoded bit is p, the error probability for the encoded bit is $\sim p^2$.

Correcting for errors on quantum bits is more complicated. Indeed, this was a major headache for researchers in the early days of quantum computing. Before Peter Shor showed in 1995 [Shor, 1995] how to achieve quantum error correction, it was hard to be optimistic about the prospects for quantum computing. The major obstacles thought to prevent quantum error correction were three:

- The no-cloning theorem [Wootters and Zurek, 1982, Dieks, 1982]. Quantum mechanics prohibits the existence of a unitary operation U that can change a known state $|\varphi\rangle$ into a copy of an unknown arbitrary quantum state $|\psi\rangle$ without perturbing the latter, i.e., $U|\varphi\rangle|\psi\rangle = |\psi\rangle|\psi\rangle$. For the classical error correction described above, such cloning was essential for encoding.

- Measuring a quantum state causes it to collapse into an eigenstate of the measured observable. How

to correct errors on an arbitrary superposition state $\alpha|0\rangle + \beta|1\rangle$ without disturbing the state when performing a measurement on it?

- In a classical bit, there is only one possible error: a bit flip, taking the bit from 0 to 1 or vice versa. For a quantum bit, there are infinitely many possible errors: any single-qubit operation, i.e., any rotation around any axis of the Bloch sphere by any angle, could possibly be induced by some external error source. How to make an error-correction procedure general enough to be able to deal with all these possible errors?

3.2 The three-qubit bit-flip code

In this section, we will look at the simplest quantum error-correction code, which demonstrates that all three obstacles above can be dealt with. The code is the three-qubit bit-flip code. The encoding and error-correction process is shown schematically in Fig. 3.1.

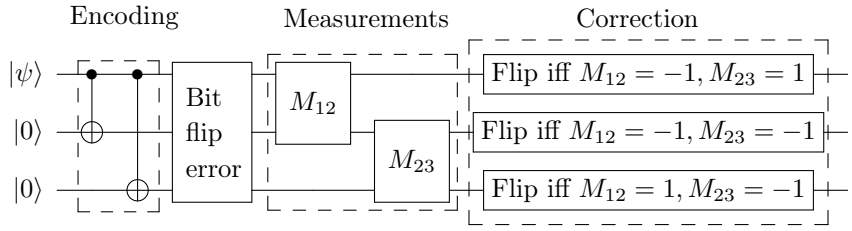


Figure 3.1: The three-qubit bit-flip error-correction code. In the first step, CNOT gates are applied to produce the state $|\psi_3\rangle$ from $|\psi\rangle$ [see Eq. (3.2)]. After a bit-flip error occurs, parity measurements are done on pairs of qubits and correcting bit flips are applied to the qubits depending on the measurement results (-1 means the qubits are in opposite states, +1 that they are in the same state). Figure from Ref. [Kockum, 2014].

We begin with an arbitrary one-qubit state

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

To protect it from bit-flip errors, i.e., from an unwanted application of the X gate, we encode it using three qubits as

$$|\psi_3\rangle = \alpha|000\rangle + \beta|111\rangle\,. \tag{3.2}$$

Note that this state can be created by application of CNOT gates, which is different from cloning the state; that would have resulted in $|\psi_3\rangle = (\alpha|0\rangle + \beta|1\rangle)^{\otimes 3}$ (if cloning had been possible).

Now, let us assume that the third qubit is flipped. This gives

$$|\psi_{3,\mathrm{err}}\rangle = \alpha|001\rangle + \beta|110\rangle\,. \tag{3.3}$$

If we first perform a parity measurement on qubits 1 and 2, i.e., measure the product Z_1Z_2, and then measure the parity of qubits 2 and 3, i.e., Z_2Z_3, we would not affect the state $|\psi_3\rangle$, since $Z_1Z_2|\psi_3\rangle = |\psi_3\rangle = Z_2Z_3|\psi_3\rangle$. Similarly, performing these measurements on $|\psi_{3,\mathrm{err}}\rangle$ does not change the coefficients α and β determining the superposition; it just adds a global phase factor. However, the result of the measurements lets us draw the conclusion that qubit 3 has been flipped (assuming that the probability of more than one qubit flipping is negligible). We can then apply an X gate to this qubit, flipping it back to its original state and recovering $|\psi_3\rangle$. As shown in Fig. 3.1, the same set of measurements also lets us correct the state if a bit-flip error

instead occurs on qubit 1 or 2. We have thus managed to circumvent the problem of quantum measurements being projective.

What happens if the error is an arbitrary X rotation $R_x(\vartheta)$ on the third qubit instead of just X? As we can see from Eq. (1.7), the resulting state would then be

$$\cos(\vartheta/2)\,|\psi_3\rangle - i\sin(\vartheta/2)\,|\psi_{3,\mathrm{err}}\rangle\,. \tag{3.4}$$

Measuring the observables Z_1Z_2 and Z_2Z_3 will either project this state into $|\psi_3\rangle$ or $|\psi_{3,\mathrm{err}}\rangle$ (modulo a global phase). In both cases, we will know what operation to apply to recover $|\psi_3\rangle$. This demonstrates how quantum error-correcting codes deal with the continuum of possible errors: measurements are used to project the perturbed state into a finite set of states from which we know how to recover the original state.

3.3 The three-qubit phase-flip code

The three-qubit bit-flip code can only correct for X errors on a single qubit. However, the idea of the code can be extended to instead deal with Z errors, i.e., phase flips. The crucial observation is that just as X flips $|0\rangle$ to $|1\rangle$ and vice versa, Z flips $|+\rangle$ to $|-\rangle$ and vice versa. So by encoding the one-qubit state we want to protect in the $\{|+\rangle\,,|-\rangle\}$ basis instead of the $\{|0\rangle\,,|1\rangle\}$ basis,

$$|\psi_3\rangle = \alpha\,|+++\rangle + \beta\,|---\rangle\,. \tag{3.5}$$

we achieve an encoded three-qubit state where a Z error flips one of the qubits from one of its basis states to the other. This encoding is implemented by adding a Hadamard gate on each qubit at the end of the encoding step in Fig. 3.1.

To detect a phase flip error in one of the three qubits, we measure the products $H^{\otimes 3}Z_1Z_2H^{\otimes 3} = X_1X_2$ and $H^{\otimes 3}Z_2Z_3H^{\otimes 3} = X_2X_3$. Just as above, these measurements do not change the coefficients α and β, but lets us conclude whether a qubit has suffered a phase flip (and which qubit it was), allowing us to apply a Z gate to correct. And just as before, this procedure also works for arbitrary Z rotations by projecting the state into either $|\psi_3\rangle$ or a state that can be corrected by applying a Z gate.

3.4 The nine-qubit Shor code

The breakthrough of Shor in 1995, which showed that quantum error correction could correct arbitrary single-qubit errors, was a nine-qubit code [Shor, 1995] that combines the three-qubit bit-flip and phase-flip codes into one. The two codes are concatenated such that the single-qubit state $|psi\rangle$ first is encoded with the three-qubit phase-flip code and then each of these three qubits are encoded with the three-qubit bit-flip code into three more qubits each. The resulting state is

$$|\psi_9\rangle = \alpha\frac{(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)}{2\sqrt{2}} + \beta\frac{(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)}{2\sqrt{2}}. \tag{3.6}$$

To find whether a bit flip has occurred, we can treat each of the three group of three qubits separately, measuring the product of Zs on the first two and the last two qubits in each group. For phase-flip errors, the procedure is slightly more complicated. We first determine in which of the three groups of three qubits that a phase flip has occurred. This is done by measuring the product of Xs on all qubits in two groups of qubits at a time, i.e., measuring $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$. The results of these measurements will tell us if a phase flip has occurred in one of the three groups, but it will not tell us which of these three qubits has been flipped. However, we do not need to know which qubit it was, because we can correct the error to the encoded state by applying Zs to all three qubits in the group.

Studying the procedure for error-correction in the nine-qubit code, it becomes clear that the code also is able to correct a combined Z and X error on one qubit. The check for bit flip errors will locate the bit flip without being affected by the presence of the phase flip error. Once the bit flip has been corrected, only the phase-flip error remains and will thus be detected and corrected as before.

Since the nine-qubit code thus can correct for both X, Z, and XZ errors, it is able to correct for any single-qubit error. This is seen by noting that any rotation of a qubit on the Bloch sphere can be decomposed into $R_z(\gamma)R_x(\beta)R_z(\alpha)$. Applying this operation to a qubit gives

$$R_z(\gamma)R_x(\beta)R_z(\alpha)\left|\psi\right\rangle = (\cos(\gamma/2) - i\sin(\gamma/2)Z)(\cos(\beta/2) - i\sin(\beta/2)X)(\cos(\alpha/2) - i\sin(\alpha/2)Z)\left|\psi\right\rangle, \tag{3.7}$$

which results in terms where either the identity operator, X, Z, or XZ has been applied to $|\psi\rangle$ (note that $XZX = -Z$). Measuring the observables for error correction will thus project any error to identity, a bit-flip error, a phase-flip error, or a combined bit- and phase-flip error. Since each of these can be corrected, any single-qubit error can be corrected.

3.5 Stabilizers

The error-correction codes above, and many others, can be understood in terms of *stabilizers*. If a state $|\psi\rangle$ is unchanged under the action of a unitary operator U, i.e., $U|\psi\rangle = |\psi\rangle$, we say that the state $|\psi\rangle$ is stabilized by U. We can extend this to a collection of operators and states. Let S be a group of N-qubit operators and V_S the set of N-qubit states that are fixed by (unchanged under the action of) every element in S. Then we say that S is the stabilizer of the vector space V_S. For V_S to be non-trivial (i.e., not just zero), it can be seen quite easily that the elements of S need to commute and that $-I$ cannot be in S.

Recall that for the three-qubit bit flip code in Sec. 3.2, we measured the operators Z_1Z_2 and Z_2Z_3, noting that they left the encoded state, a superposition of $|000\rangle$ and $|111\rangle$, unchanged. Here, the stabilized vector space V_S is spanned by $|000\rangle$ and $|111\rangle$ and the stabilizer S is the group generated by Z_1Z_2 and Z_2Z_3, i.e., the group $\{I, Z_1Z_2, Z_2Z_3, Z_1Z_3\}$ (products of the elements in the group are elements in the group; the elements I and Z_1Z_3 can be constructed by multiplying together the generators Z_1Z_2 and Z_2Z_3 in various ways).

So by measuring the generators of S, we were able to correct certain errors on the stabilized states. We can imagine constructing other error-correction codes by finding a stabilizer, its generators, and the corresponding stabilized states. But how do we know which errors the code protects from? If we work with a stabilizer S and errors $\{E_j\}$ that are subgroups of the N-qubit Pauli group (products of single-qubit Pauli matrices and factors $\pm 1, \pm i$), there is a theorem (Theorem 10.8 in Ref. [Nielsen and Chuang, 2000]) that tells us that these errors can be corrected if $E_j^\dagger E_k \notin N(S) - S \ \forall j, k$. Here, the normalizer of S, $N(S)$, consists of all elements $A \notin S$ such that A commutes with all elements in S. For our example with the three-qubit bit-flip code, it is quite straightforward to see that any product of two elements in the error set $\{I, X_1, X_2, X_3\}$ anti-commutes with the generators Z_1Z_2 and Z_2Z_3 of the stabilizer (except I, but $I \in S$, so $I \notin N(S) - S$), so all errors in the set can be corrected.

3.6 Proving the Gottesman–Knill theorem

Here we outline how to prove the Gottesman–Knill theorem using stabilizers. For a more detailed description of the proof, see Sections 10.5.2–10.5.4 in Ref. [Nielsen and Chuang, 2000].

If we have a vector space V_S stabilized by S, the action of any unitary U on a state $|\psi\rangle \in V_S$ can, for any element $g \in S$, be written

$$U\left|\psi\right\rangle = Ug\left|\psi\right\rangle = UgU^\dagger U\left|\psi\right\rangle. \tag{3.8}$$

This means that $U|\psi\rangle$ is stabilized by $UgU^\dagger$, and thus UV_S is stabilized by $USU^\dagger$, which is generated by the operators $Ug_1U^\dagger, ..., Ug_n$ if $g_1, ..., g_n$ are the generators of S.

The point of this is that it sometimes allows a compact representation of qubit states under transformations. Consider, for example, the state $|0\rangle^{\otimes N}$. Applying a Hadamard gate to each qubit in this state transforms it to $|+\rangle^{\otimes N}$, which requires 2^N amplitudes to represent in the computational basis. However, the state $|0\rangle^{\otimes N}$ is the only state (up to a global phase) that is stabilized by the stabilizer generated by $\{Z_1, Z_2, ..., Z_N\}$. After the transformation, $|+\rangle^{\otimes N}$ is uniquely determined by $H\{Z_1, Z_2, ..., Z_N\}H^\dagger = \{X_1, X_2, ..., X_N\}$, which only is N generators.

It turns out that the gate set in the Gottesman–Knill theorem, $\{\text{CNOT}, H, S\}$, is such that any unitary that transforms elements of the N-qubit Pauli group to other elements of the N-qubit Pauli group can be composed from $\mathcal{O}(N^2)$ gates in that set. Thus, starting in a computational basis state, specified by N generators in the N-qubit Pauli group, we can keep track of changes to the state by keeping track of how these generators change under the action of $\{\text{CNOT}, H, S\}$. This only requires $\mathcal{O}(N^2)$ steps on a classical computer. The same goes for measurements of these states, so the quantum circuit is classically simulatable.

3.7 Fault-tolerant quantum computing

The error-correction codes we have looked at so far have helped us reduce the probability that a single-qubit error will result in an actual error in an encoded single-logical-qubit state formed by several physical qubits. However, this is just the first step towards quantum computing that works in the presence of errors. Such *fault-tolerant quantum computing* requires schemes to perform logical operations and measurements on logical qubits in a way that itself is robust against errors. Explaining how this is done is beyond the scope of this course, but it can be done.

The most important concept in fault-tolerant quantum computing is the *error threshold*. This is the single-qubit error probability which can be tolerated in practice and still allow fault-tolerant quantum computing. The concept can be intuitively understood already from the three-qubit bit-flip code. For an unencoded qubit, the error probability is p. For the encoded three-qubit state, we can correct for one single-qubit error, but two single-qubit errors will result in an error for the logical qubit. This occurs with a probability cp^2 for some c that is determined by how the code works (in this case, $c = 3$). For it to make sense to invest qubit resources into the three-qubit encoding, we must have $cp^2 < p$, i.e., $p < 1/c$. The error threshold is thus $1/c$.

If we are below the error threshold, we can reduce the logical error rate further by concatenating the code at more levels. For the bit-flip case, each of the three qubits making up the logical qubit could in turn be encoded into three qubits each to protect against bit-flip errors. And each of those qubits could be encoded into three more, and so on. But will this not lead to having to invest too many resources to lower the logical error rate enough? Fortunately, the answer is no. This is the *threshold theorem for quantum computation*: A quantum circuit containing s gates may be simulated with probability error of at most ε using $\mathcal{O}(\text{poly}(\log s/\varepsilon)s)$ gates on hardware whose components fail with probability at most p, provided p is below some constant threshold, $p > p_{\text{th}}$, and given reasonable assumptions about the noise in the underlying hardware. Here, "poly" is a polynomial of fixed degree.

3.8 The surface code

In practice, only small examples of error-correcting codes have been demonstrated in experiments so far. There are several practical challenges for experimental implementations of error correction at larger scale. One is that some concatenated codes may require high qubit connectivity. Recall, as a simple example, that the nine-qubit Shor code required measuring two six-qubit stabilizers to identify the phase-flip errors.

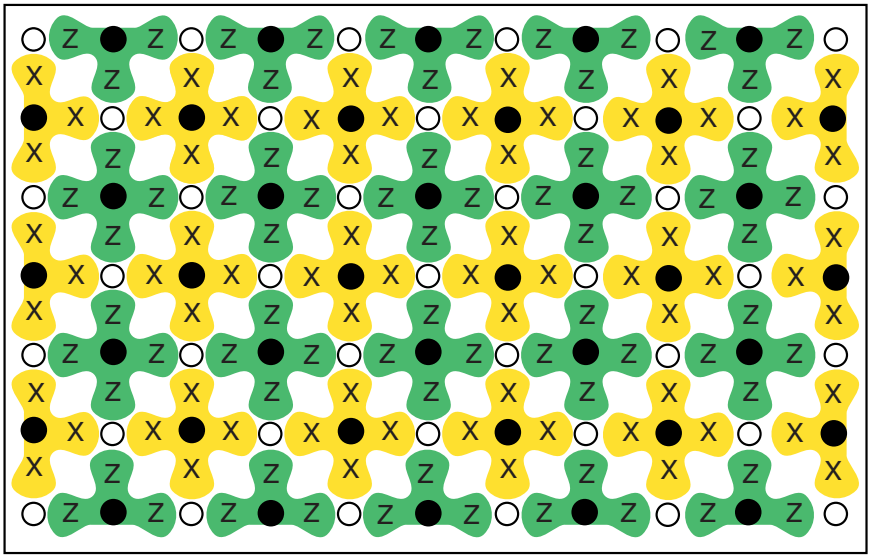


Figure 3.2: The layout for the surface code. The open circles represent qubits that are part of the encoded logical state. The solid circles are qubits that are not part of the logical state, but are used for measurements of the four-qubit $XXXX$ and $ZZZZ$ stabilizers of the code. From Ref. [Fowler et al., 2012].

The error-correction code attracting most attention for large-scale implementation today is the surface code. For a detailed explanation of how the surface code works, we refer to Ref. [Fowler et al., 2012]. Here, we will only explain some basic points of the code.

The surface code is adapted to a square grid of qubits, where each qubit can interact (perform two-qubit gates) with its four nearest neighbours. The layout is depicted in Fig. 3.2. The surface code only uses four-qubit stabilizers on the form $XXXX$ and $ZZZZ$, which can be measured using the nearest-neighbour interactions on the grid to perform sequential CNOT or CZ gates between the four "data qubits" to be measured and a "measurement qubit" connected to them, as shown in the figure.

Once the system state has been projected into a state that is stabilized by the four-qubit measurements, a Z error on a data qubit anti-commutes with the $XXXX$ stabilizer measurements that involve one of the two X-measurement qubits connected to this data qubit. Similarly, an X error on a data qubit anti-commutes with the $ZZZZ$ stabilizer measurements that involve one of the two Z-measurement qubits connected to this data qubit. In this way, it is possible to identify when single-qubit errors (also XZ errors) occur on data qubits. However, if the error probability increases such that several data qubits close to each other experience errors, it can be hard to deduce which error configuration actually caused the measurement results, even if a logical error has not occurred.

To flip the state of the encoded qubit, a chain of X or Z operations, stretching from one side to the other of the square grid, is required, as shown in Fig. 3.3. The larger the distance d across the grid of data qubits, the better protected the encoded qubit is, provided the error probability is below the threshold. For the surface code, the threshold is on the order of 1 %, which is better than many other error-correcting codes. This contributes to the great interest in implementing the surface code.

It is possible to perform two-qubit operations on encoded qubits in the surface code. However, explaining how this works is beyond the scope of these notes. See Ref. [Fowler et al., 2012] for details.

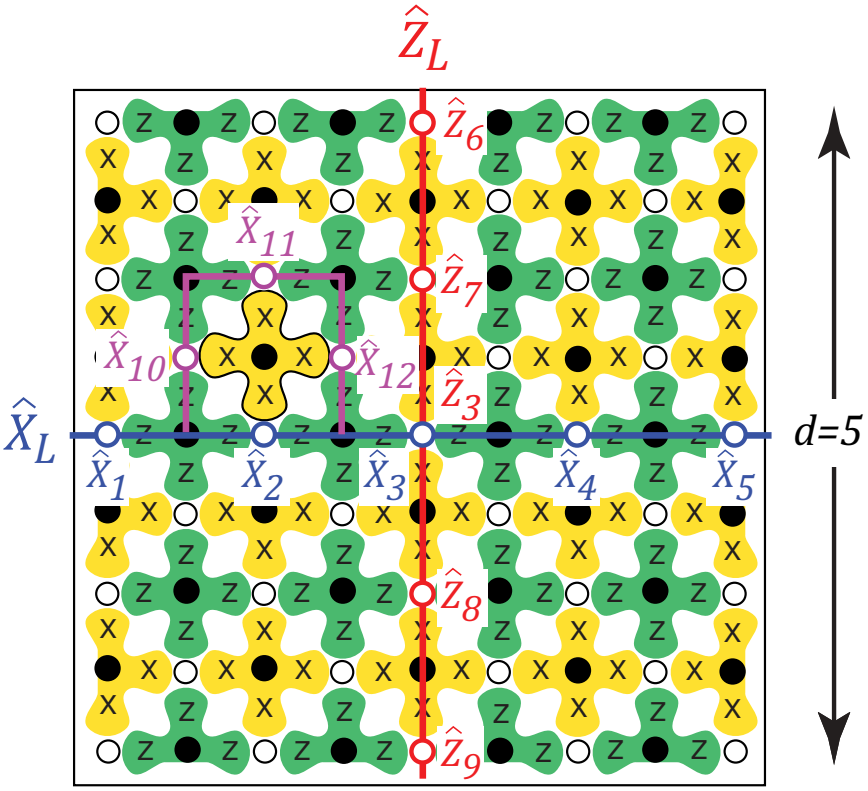


Figure 3.3: Logical operations on the surface code. The depicted setup has distance 5, i.e., $d = 5$ data qubits along each side of the square. There are 41 data qubits in the setup and together with 40 measurement qubits for X and Z stabilizers. The difference leaves room for encoding one logical qubit. To perform logical operations on this encoded qubit, a chain of operators must be applied across the square grid, as shown. A chain of X operators connecting the two opposite sides with X measurements at the boundaries implement a logical X operation. Likewise, a chain of Z operators connecting the two opposite sides with X measurements at the boundaries implement a logical Z operation. From Ref. [Fowler et al., 2012].

Chapter 4

Fast Quantum Algorithms

For this part of the notes, we follow Ref. [Nielsen and Chuang, 2011].

4.1 The Quantum Fourier Transform

You should all have seen the discrete Fourier transform (DFT) where a set of N complex numbers $\{x_0, \ldots, x_{N-1}\}$ are Fourier transformed into N new complex numbers $\{y_0, \ldots, y_{N-1}\}$ according to

$$y_k = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} e^{i\frac{2\pi mk}{N}} x_m.$$

The Fourier transformation is extremely useful, e.g. to detect periods in a signal where $\{x_0, \ldots, x_{N-1}\}$ could be the amplitude of some signal as a function of discretized time. The Fourier transformed signal $\{y_0, \ldots, y_{N-1}\}$ then describes the frequency content. Solving the Schrödinger equation on a lattice, DFT is what takes you between real space and momentum (k) space.

The *quantum* Fourier transform (QFT) is a unitary n-qubit operation, transforming the initial $N = 2^n$ basis states $\{|0\rangle, \ldots, |N-1\rangle\}$ into a new basis in a way which looks mathematically identical to the DFT,

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i\frac{2\pi jk}{N}} |k\rangle. \tag{4.1}$$

The action on an arbitrary state is

$$\sum_{j=0}^{N-1} x_j |j\rangle \to \sum_{k=0}^{N-1} y_k |k\rangle,$$

where the amplitudes y_k are the DFT transforms of the amplitudes x_m. One may easily verify that the new states are normalized and form an orthogonal set, and thus that the QFT is a unitary transform.

The QFT can be used to find periods and also to extract eigenvalues of unitary operators to a high precision. But before discussing these issues in more detail, let's see if we can find an effective implementation of the QFT. Remember that there are operators that need exponentially many single- and two-qubit gates for implementation, so what about the QFT?

4.1.1 Another definition

We'll now rewrite the definition of the QFT in a way that is more transparent for constructing a circuit. First we need a simple way to number the basis states. We simply number the n-qubit state $|j\rangle$ using the

binary n-bit representation of $j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_{n-1} 2^1 + j_n 2^0$. E.g. in the 4 qubit case the state $|5\rangle = |0101\rangle = |0_1\rangle|1_2\rangle|0_3\rangle|1_4\rangle$.

Alternative notation: the binary fraction

Before starting, we introduce a notation that is used in NC to perform calculations on the quantum Fourier transform and the phase estimation algorithms. However, in these notes we will make use of both notations - the standard one and the one based on the binary fraction - leaving to the reader the choice of which one is preferable.

The definition of binary fractions is the following:

$$0.j_1 j_2 j_3 \dots j_n = j_1/2 + j_2/2^2 + j_3/2^3 \cdots + j_n/2^n,$$

e.g. $0.101 = 0.5 + 0.125 = 0.625$, and more generally

$$0.j_l j_{l+1} \dots j_m = j_l/2 + j_{l+1}/2^2 + \cdots + j_m/2^{m-l+1}.$$

Using this notation we can write the QFT in Eq. (4.1) as

$$|j\rangle = |j_1, j_2, \dots, j_n\rangle \rightarrow \frac{\left(|0\rangle + e^{i2\pi 0.j_n}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_{n-1}j_n}|1\rangle\right)\dots\left(|0\rangle + e^{i2\pi 0.j_1 j_2 \dots j_n}|1\rangle\right)}{2^{n/2}}. \tag{4.2}$$

Rewriting the output state of the quantum Fourier transform

The algebraic manipulations connecting the two expressions are straightforward, but need some afterthought. Observing that

$$\frac{k}{2^n} = \frac{k_1 2^{n-1}}{2^n} + \cdots + \frac{k_n 2^0}{2^n} = k_1 2^{-1} + \cdots + k_n 2^{-n} = \sum_{l=0}^{n} k_l 2^{-l}, \tag{4.3}$$

from Eq.(4.1) we have:

$$\begin{aligned}
|j\rangle &\rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i\frac{2\pi jk}{N}} |k\rangle \qquad (4.4)\\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{i2\pi j \sum_{l=1}^{n} k_l 2^{-l}} |k_1 \dots k_n\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{i2\pi j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \sum_{k_l=0}^{1} e^{i2\pi j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} (|0\rangle + e^{i2\pi j 2^{-l}}|1\rangle) \\
&= \frac{1}{\sqrt{N}} (|0\rangle + e^{i2\pi j 2^{-1}}|1\rangle)(|0\rangle + e^{i2\pi j 2^{-2}}|1\rangle) \dots (|0\rangle + e^{i2\pi j 2^{-n}}|1\rangle) \\
&= \frac{1}{\sqrt{N}} (|0\rangle + e^{i2\pi 0.j_n}|1\rangle)(|0\rangle + e^{i2\pi 0.j_{n-1}0.j_n}|1\rangle) \dots (|0\rangle + e^{i2\pi 0.j_1 \dots j_n}|1\rangle)
\end{aligned}$$

where in the last step we have used that

$$\begin{aligned} j2^{-n} &= j_1/2 + j_2/4 + \ldots j_n/2^n = 0.j_1 \ldots j_n \\ &\ldots \\ j2^{-1} &= j_1 2^{n-2} + j_2 2^{n-3} + \ldots j_n/2 = j_1 2^{n-2} + j_2 2^{n-3} + \cdots + 0.j_n \end{aligned} \tag{4.5}$$

and that the integer part of $j \cdot 2^l$ disappears in the exponent since it is multiplied by 2π.

4.1.2 An efficient implementation

Using the form of the QFT in Eq. (4.2) it is straightforward to implement the desired transformation with a quantum a circuit. We realize that we have to implement conditional phase shifts on each qubit, therefore we define the single qubit operator

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{bmatrix}.$$

Now let's see what happens an input state $|j_1, j_2, \ldots, j_n\rangle$ when it passes through the circuit in Fig. 4.1. The first Hadamard gate produces the state $(|0\rangle + |1\rangle)/\sqrt{2}$ if $j_1 = 0$ and $(|0\rangle - |1\rangle)/\sqrt{2}$ if $j_1 = 1$, i.e.

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi j_1/2}|1\rangle\right)|j_2, \ldots, j_n\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi 0.j_1}|1\rangle\right)|j_2, \ldots, j_n\rangle,$$

since $e^{i2\pi 0.j_1} = e^{i2\pi j_1/2} = -1$ for $j_1 = 1$ and $+1$ otherwise. The controlled-R_2 gate rotates the component $|1\rangle$ of the first qubit by $e^{i2\pi/2^2}$ if $j_2 = 1$, i.e. it applies the phase $e^{i2\pi j_2/2^2}$. Therefore, it produces the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi j_2/2^2} e^{i2\pi j_1/2}|1\rangle\right)|j_2, \ldots, j_n\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi 0.j_1 j_2}|1\rangle\right)|j_2, \ldots, j_n\rangle.$$

After all the controlled-R_k operations on the first qubit, the state is

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi\left(\frac{j_1}{2} + \ldots + \frac{j_n}{2^n}\right)}|1\rangle\right)|j_2, \ldots, j_n\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi 0.j_1 j_2 \ldots j_n}|1\rangle\right)|j_2, \ldots, j_n\rangle.$$

The Hadamard on the second qubit produces

$$\begin{aligned} &\tfrac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi\left(\frac{j_1}{2} + \ldots + \frac{j_n}{2^n}\right)}|1\rangle\right)\left(|0\rangle + e^{i2\pi\frac{j_2}{2}}|1\rangle\right)|j_3, \ldots, j_n\rangle \\ &= \tfrac{1}{\sqrt{2^2}}\left(|0\rangle + e^{i2\pi 0.j_1 j_2 \ldots j_n}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_2}|1\rangle\right)|j_3, \ldots, j_n\rangle, \end{aligned} \tag{4.6}$$

and the controlled R_2 to R_{n-1} gates yield the state

$$\begin{aligned} &\tfrac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi\left(\frac{j_1}{2} + \ldots + \frac{j_n}{2^n}\right)}|1\rangle\right)\left(|0\rangle + e^{i2\pi\left(\frac{j_2}{2} + \ldots + \frac{j_n}{2^{n-1}}\right)}|1\rangle\right)|j_3, \ldots, j_n\rangle \\ &= \tfrac{1}{\sqrt{2^2}}\left(|0\rangle + e^{i2\pi 0.j_1 j_2 \ldots j_n}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_2 \ldots j_n}|1\rangle\right)|j_3, \ldots, j_n\rangle. \end{aligned} \tag{4.7}$$

We continue in this fashion for all qubits, obtaining the final state

$$\begin{aligned} &\tfrac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi j 2^{-n}}|1\rangle\right)\left(|0\rangle + e^{i2\pi j 2^{-(n-1)}}|1\rangle\right) \ldots \left(|0\rangle + e^{i2\pi j 2^{-1}}|1\rangle\right) \\ &= \tfrac{1}{\sqrt{2^n}}\left(|0\rangle + e^{i2\pi 0.j_1 j_2 \ldots j_n}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_2 \ldots j_n}|1\rangle\right) \ldots \left(|0\rangle + e^{i2\pi 0.j_n}|1\rangle\right). \end{aligned} \tag{4.8}$$

We now need to reverse the order of the qubits, which can be achieved using a series of SWAP gates. The number of gates needed are n on the first qubit plus $n-1$ on the second and so on, adding up to $n(n+1)/2 = O(n^2)$ gates. Then we need on the order of n SWAP gates, not changing the scaling. Thus

we can implement the QFT for n qubits using on the order of $O(n^2)$ gates. The best classical algorithm (FFT) needs $O(n2^n)$ gates, indicating why the QFT could be used for speedup. This does not translate in an immediate speed-up for computing classical FFT, because we cannot access the amplitudes when measuring the Fourier-transformed quantum state, and we don't even know how to efficiently prepare the input state to be transformed. However, in the next section we'll see one problem where the quantum Fourier transform is useful.

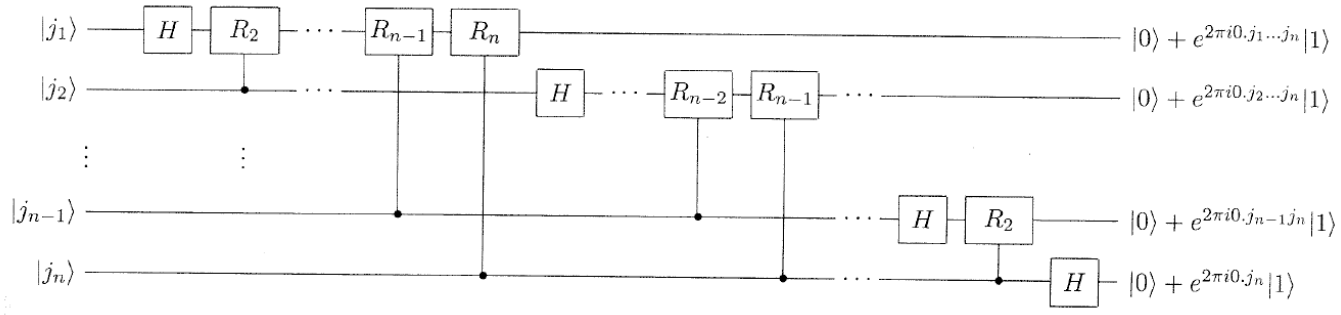


Figure 4.1: An efficient circuit to perform the quantum Fourier transform. (From Nielsen & Chuang), Fig. 5.1. Here $0.j_1 \ldots j_n = j2^{-n}, 0.j_2 \ldots j_n = j2^{-(n-1)}, \ldots, 0.j_{n-1} j_n = j/2^{-2}$ and $0.j_n = j/2^{-1}$.

4.2 Phase estimation

The aim of this algorithm is to estimate the angle in the eigenvalue $e^{i2\pi\phi}$ corresponding to an eigenvector $|u\rangle$ of a unitary operator U. The vector $|u\rangle$ is given, as well as a circuit (black box, oracle) effectively implementing controlled-U^n operations. A circuit solving a first stage of this problem is shown in Fig. 4.2 and an overview circuit showing the whole algorithm is given in Fig. 4.3. Two qubit registers are needed, the first has t qubits which are initialized to zero. The number of qubits t is determined by the required accuracy in the estimate of ϕ. The second register is large enough to represent the eigenvector $|u\rangle$, and it is also initialized to $|u\rangle$ and remains in this state throughout the computation. The initial set of Hadamard gates puts all qubits of register 1 in an equal superposition of $|0\rangle$ and $|1\rangle$. If the k-th control qubit is in the state $|1\rangle$ a unitary operation U^{2^k} will be performed on the second register, picking up a phase $\left(e^{i2\pi\phi}\right)^{2^k} = e^{i2\pi\phi 2^k}$. The first step ($k = 0$) gives for example:

$$C_U \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right) |u\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi\phi}|1\rangle\right) |u\rangle. \tag{4.9}$$

The final state of the first register in this first step is

$$\frac{1}{\sqrt{2^t}} \left(|0\rangle + e^{i2\pi 2^{t-1}\phi}|1\rangle\right) \left(|0\rangle + e^{i2\pi 2^{t-2}\phi}|1\rangle\right) \ldots \left(|0\rangle + e^{i2\pi 2^1\phi}|1\rangle\right) \left(|0\rangle + e^{i2\pi 2^0\phi}|1\rangle\right) = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{i2\pi k\phi}|k\rangle. \tag{4.10}$$

By comparison with (4.4) we see that this state is nothing else than the Fourier transform of the state $|2^t\phi\rangle = |\phi_1\phi_2 \ldots \phi_t\rangle$, where in the last step we have assumed that the phase ϕ has an exact representation in t bits as $\phi = 0.\phi_1\phi_2 \ldots \phi_t$ (with a slight abuse of notation). The final step is hence to make an Inverse quantum Fourier transform of the first register. This allows recovering the latter state. Register 1 is then read out and ϕ is recovered. If the phase is not an exact binary fraction in t qubits there will be some finite

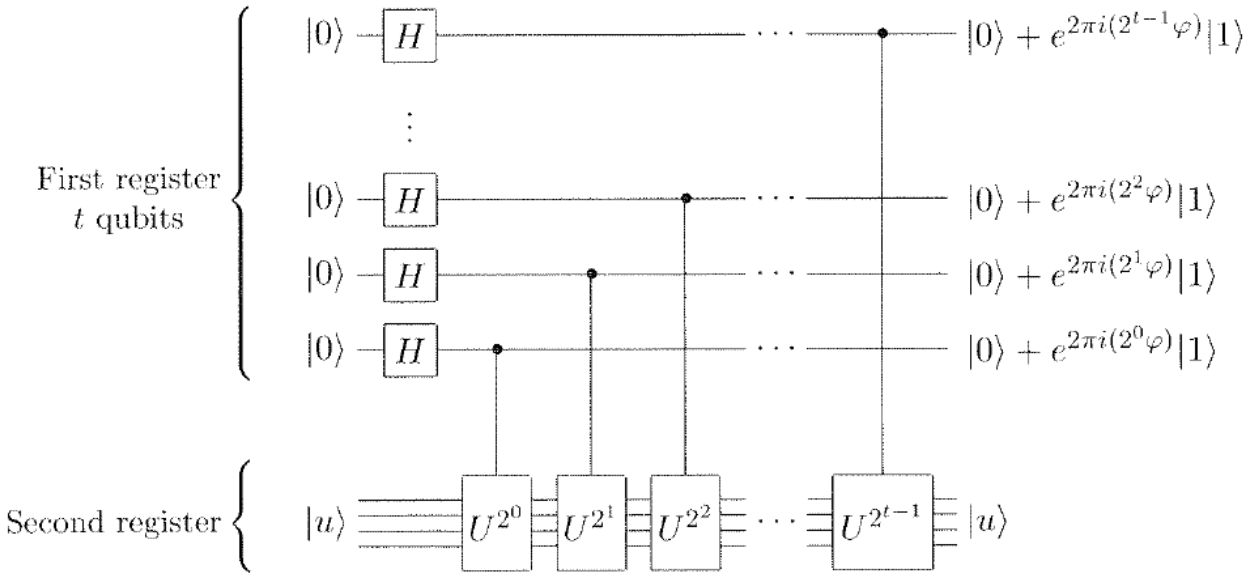


Figure 4.2: A circuit performing the first step of the phase estimation algorithm. (From Nielsen & Chuang), Fig. 5.2.

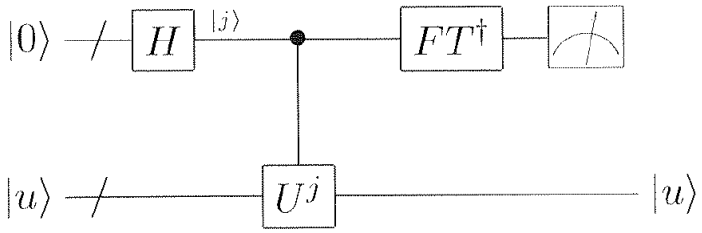


Figure 4.3: An overview circuit figure of the phase estimation circuit. (From Nielsen & Chuang), Fig. 5.3.

probability of reading out some other state "close" to the best approximation. A careful analysis gives that if we want n bits precision, with a failure probability less than ε, we need a register of size

$$t = n + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil.$$

Phase estimation is interesting in its own right and may be used in quantum simulations. We will see how it enters Shor's algorithm for factoring.

4.3 Factoring - Shor's algorithm

We now know how to efficiently determine the phase of an eigenvalue to a unitary operator. In this section we'll see how this enables us to efficiently solve a number theoretical problem which is considered hard on classical computers: order finding. Finally we show how factoring can be reduced to order finding.

4.3.1 Modular arithmetics

Order finding is defined in *modular arithmetics*. Modular arithmetics is based on the fact that, given any two positive integers x and n, x can uniquely be written as

$$x = k \cdot n + r,$$

where k is a non-negative integer and $0 \leq r < n$ is the reminder

$$x = r \mod n,$$

as an example

$$2 = 5 = 8 = 11 \mod 3.$$

The *greatest common divisor* $gcd(a, b)$ of two integers a and b is the largest integer dividing both a and b. If $gcd(a, b) = 1$ then a and b are called *co-prime*.

Multiplicative inverse

Now let's look at modular multiplication by looking at the series

$$m_k = k \cdot a \mod n,\ 0 < k < n.$$

As an example, take $a = 6$ and $n = 15$ with $gcd(a, n) = 3$ giving

$$m_k = \{6, 12, 3, 9, 0, 6, 12, 3, 9, 0, 6, 12, 3, 9\},$$

showing that the equation $x \cdot 6 = y \mod 15$ has no solution for $y \in \{1, 2, 4, 5, 7, 8, 10, 11\}$. Note that in particular it has no solution for $y = 1$. Then take $a = 7$ and $n = 15$ which are co-prime giving

$$m_k = \{7, 14, 6, 13, 5, 12, 4, 11, 3, 10, 2, 9, 1, 8\},$$

showing that the equation $x \cdot 7 = y \mod 15$ may be solved for all y.

The *multiplicative inverse* a^{-1} of an integer a modulus n is another integer which fulfils

$$a^{-1} \cdot a = 1 \mod n,$$

and it exists if and only if a and n are co-prime. If the inverse exists we can solve the equation

$$x \cdot a = c \mod n$$

for all integers a, b and c through

$$x = c \cdot a^{-1} \mod n.$$

Another way of formulating this is the following: all the integers between 1 and $(n - 1)$ appear once and only once in $\{m_k\}$ if and only if a and n are co-prime. If not we can write $a = x \cdot gcd(a, n)$ and $n = y \cdot gcd(a, n)$, where $0 < y < n$. So for $k = y$ we get

$$m_y = y \cdot x \cdot gcd(a, n) \mod (y \cdot gcd(a, n)) = 0,$$

and then the series $\{m_k\}$ will just repeat from the start.

4.3.2 Order finding

Consider the equation

$$x^r = 1 \mod N,$$

which has solutions for the integers x and N being co-prime, and $x < N$. The lowest positive integer r solving the equation is called the *order* of x modulo N. One straightforward method to calculate r is to evaluate the series $m_k = x^k \mod N$ for $0 < k < N$, then it's clear that the series $\{m_k\}$ is periodic with period r since $x^{r+a} = x^r \cdot x^a = x^a \mod N$. In other words, the order is the period of the modular exponentiation function $m_k = x^k \mod N$. As an example let's take $x = 5$ and $N = 21$ giving

$$m_k = \{5, 4, 20, 16, 17, 1, 5, 4, 20, 16, 17, 1, 5, 4, 20, 16, 17, 1, 5, 4\},$$

and we have the order $r = 6$. There is no classical algorithm for finding r which scales polynomially in the number of bits L needed to represent the input, i.e. the integers x and N.

4.3.3 Factoring as order finding

Factoring can be reduced to order finding as follows. Suppose we want to factor $N = pq$. Consider the period r of the function of k defined as $x^k \mod N$ for some x. x is chosen such that r is even and that $x^{r/2} \neq N - 1 \mod N$. Then r allows to find p and q as follows. Define $y = x^{r/2}$. Then $y^2 = x^r = 1 \mod N$ by definition of period r. Therefore we have

$$y^2 - 1 = (y - 1)(y + 1) = 0 \mod N.$$

Therefore$(y - 1)(y + 1)$ is multiple of N, i.e. it contains the two factors pq in his decomposition. However neither $(y - 1)$ nor $(y + 1)$ are multiple N:

$$(y - 1) \neq 0 \mod N, \text{ because otherwise the period would be smaller than r.} \tag{4.11}$$

$$(y + 1) \neq 0 \mod N, \text{ by construction.} \tag{4.12}$$

Therefore $(y - 1)$ and $(y + 1)$ must slip the two factors q and p appearing in the decomposition of N. Say for instance:

$$(y - 1) = lp; \qquad (y + 1) = l'q.$$

We finally obtain therefore

$$p = gcd(N, x^{r/2} - 1); \qquad q = gcd(N, x^{r/2} + 1).$$

In other words, determining the order r of the modular exponentiation function $x^k \mod N$ yields the determination of the two factors p and q such that $N = pq$. For n having L bits, this common factor can be found using Euclid's algorithm in $O(L^3)$ steps. For uniformly chosen x one may calculate a lower bound for the probability of r being even and that $y = x^{r/2}$ is non-trivial,

$$p(r \text{ is even and } x^{r/2} \neq -1 \mod N) \geq 1 - \frac{1}{2^m},$$

where m is the number of different prime-factors in N, i.e. $m \geq 1$.

In the following, we are going to derived an efficient quantum algorithm for order finding.

4.3.4 A quantum algorithm for order finding

Given an integer x of which we want to find the order mod N, consider the L-qubit unitary operation

$$U|y\rangle \equiv \begin{cases} |x \cdot y \mod N\rangle, & 0 \leq y \leq N-1 \\ |y\rangle, & N \leq y \leq 2^L - 1 \end{cases}.$$

The unitarity follows since it basically permutes the basis states and y has a multiplicative inverse modulus N since y and N are co-prime. The states

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left\{ \left[\frac{-i2\pi sk}{r} \right] \right\} |x^k \mod N\rangle,$$

defined for integers $0 \leq s \leq r-1$ are eigenstates of U, since

$$U|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left\{ \left[\frac{-i2\pi sk}{r} \right] \right\} |x^{k+1} \mod N\rangle = \frac{1}{\sqrt{r}} \sum_{k=1}^{r} \exp\left\{ \left[\frac{-i2\pi s(k-1)}{r} \right] \right\} |x^k \mod N\rangle = \exp\left\{ \left[\frac{i2\pi s}{r} \right] \right\} |u_s\rangle,$$

since $x^r = x^0 \mod N$, and $\exp\left\{ \left[\frac{-i2\pi s(r-1)}{r} \right] \right\} = \exp\left\{ \left[\frac{i2\pi s}{r} \right] \right\}$. Using the phase estimation algorithm we can now efficiently determine s/r with high accuracy. One requirement is that we can implement the operators U^{2^k} efficiently, which can be done using a procedure known as modular exponentiation (see Box 5.2. on page 228 in N & C), needing $O(L^3)$ gates. Furthermore we need to produce one or more of the eigenstates $|u_s\rangle$ which is done by noting

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = \frac{1}{r} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} \exp\left\{ \left[\frac{-i2\pi sk}{r} \right] \right\} |x^k \mod N\rangle = |1\rangle.$$

To have enough accuracy in the phase estimation we should use $t = 2L + 1 + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right) \right\rceil$ qubits in the first register and prepare the second register in the $|1\rangle$ state. We'll then get the phase $\phi = s/r$, for a random $0 \leq s < r$, with $2L+1$ bits precision, with a probability of at least $(1-\varepsilon)$. Knowing that the phase $\phi = s/r$ is a rational number, where s and r are integers, not larger than L bits, we can classically determine s and r. The appropriate algorithm is called the continued fraction expansion and needs $O(L^3)$ gates.

4.3.5 Performance

The algorithm fails if $s = 0$, and also if s and r have common factors so that they cannot be extracted from s/r. The probability of failure can be shown to be small and one need only to repeat the procedure a polynomial (in L) number of times to obtain r with high probability. The number of gates needed are $O(L)$ for the initial Hadamards, inverse Fourier transform needs $O(L^2)$ gates, implementing U^{2^k} through modular exponentiation requires $O(L^3)$ gates, and the classical continued fraction algorithm needs $O(L^3)$ (classical) gates. If we need to repeat an $O(L)$ number of times the overall scaling would be $O(L^4)$, but being more clever there are ways to guarantee success in a constant number of attempts, giving the scaling $O(L^3)$ gates.

The algorithm

Find a factor of the composite L-bit integer N.

- 1. If N is even return the factor 2.
- 2. Determine whether $N = a^b$, i.e. is if there is only one prime-factor. This can be done with $O(L^3)$ operations. If so return the factor a.

- 3. Randomly choose $1 < x < (N-1)$, and check whether x and N are co-prime ($O(L^3)$ operations). If not co-prime return the factor $gcd(x, N)$.
- 4. Find the order r of x modulo N, which can be done using $O(L^3)$ quantum gates (quantum subroutine!!)
- 5. If r is even and $x^{r/2} \neq -1 \mod N$ then compute $gcd(x^{r/2}+1, N)$ and $gcd(x^{r/2}-1, N)$ and check if one is a non-trivial factor of N. Return this factor. If r is odd, or $x^{r/2} = -1 \mod N$ the algorithm fails.

The algorithm will succeed with a probability larger than 3/4.

Chapter 5

Quantum machine learning

Quantum computing and machine learning are arguably two of the "hottest" topics in science at the moment. Here in Sweden, this is reflected in the fact that the two largest programs supported by the Knut and Alice Wallenberg foundation are centered around quantum technologies and artificial intelligence. In this chapter, we will discuss efforts to combine the two fields into quantum machine learning. Since this is a course about quantum algorithms, we do not cover applications of classical machine learning to simulating and understanding quantum systems, but focus instead on how machine learning can be enhance by quantum computation. We begin with a brief overview of classical machine learning and then study some examples of quantum machine learning algorithms.

5.1 A brief overview of classical machine learning

What is machine learning? With the success of, and hype around, machine learning in recent years, there are examples of companies and researchers calling many things "machine learning" that would have been called something else a few years ago. According to Wikipedia, "Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead". We like the following definition (source unknown): "Machine learning studies algorithms whose performance improves with data ('learning from experience')".

5.1.1 Types of machine learning

Broadly speaking, there are three paradigms in machine learning for extracting meaning from some data:

- **Unsupervised learning**: Learning structure in $P(\text{data})$ given samples from $P(\text{data})$. Here, the machine learning is used to *generate* knowledge by analyzing unlabelled data. Examples of unsupervised learning are clustering (grouping data points), density estimation (estimating a probability density function giving rise to the data), and much of what is called data mining.

- **Supervised learning**: Learning structure in $P(\text{labels}|\text{data})$ given samples from $P(\text{data}, \text{labels})$. Here, the machine learning is used to *generalize* knowledge gained from studying labelled data to predict correct labels for unlabelled data. Examples of supervised learning are foremost various classificiation tasks, e.g., image recognition.

- **Reinforcement learning**: Learning from (possibly rare) rewards. Here, an agent learns by acting on its environment, observing the results of its actions (the results can include rewards), and updating its

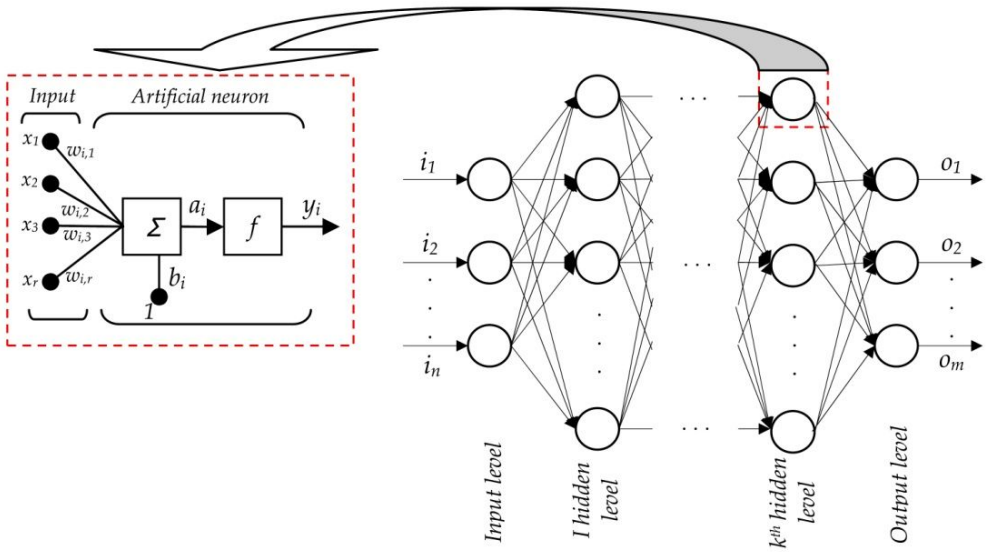


Figure 5.1: Structure of a feed-forward neural network and a single neuron in that network. From Ref. [Tanikic and Despotovic, 2012].

policy for actions based on the outcomes. Examples of reinforcement learning include the superhuman-level game-playing programs for go, chess, StarCraft, etc. by Google's DeepMind.

5.1.2 Neural networks

One common way to implement machine learning is using neural networks. The neurons in such a network can be connected in different layouts. Figure 5.1 shows a feed-forward neural network, where some input (data) is fed into the neurons in the input layer on the left. The output from these neurons becomes the input for neurons in the next layer, and so forth. In the end, some result is defined by the output from the last layer on the right. The layers between the input and output layers are called hidden layers.

The structure of a single neuron is shown on the left in Fig. 5.1. The inputs x_j to neuron i are weighted by weights $w_{i,j}$. To the weighted sum of the inputs, a "bias" b_i can be added. The result a_i is fed into a nonlinear activation function f, which usually is some smoothed version of a step function. The output is then

$$y_i = f\left(\sum_j w_{i,j}x_j + b_i\right). \tag{5.1}$$

A neural network can thus be said to be a complicated function, parameterized by all the weights and biases in the network, which transforms an input into an output. What functions can a neural network represent? The answer, provided by Cybenko in 1989 [Cybenko, 1989], gives a hint of why neural networks are powerful. It turns out that any arbitrary smooth function with vector input and vector output can be approximated to any desired precision by a feedforward neural network with only a single hidden layer. In practice, deep neural networks, i.e., networks with many hidden layers, have turned out to be more efficient at representing various functions. See, e.g., Ref. [Lin et al., 2017].

5.1.3 Training neural networks

To train a neural network to perform a specific task boils down to adjusting the weights w and biases b such that the network approximates a function that solves the task. To do this, we first need to be able to say how close the output of the current network is to the desired output. The difference between the actual and the desired output is measured by some cost function. One example is the mean square error

$$C(w,b) = \frac{1}{2n}\sum_x |y(x) - a|^2, \tag{5.2}$$

where n is the number of examples x (data points), y is the desired output, and a is the actual output. To find good weights and biases for the task is to find weights and biases that minimize $C(w,b)$.

How does one minimize C in a smart way? Clearly, there are too many unknowns to simply find the extremum by setting the gradient of C to zero and solving the resulting equation. Therefore, gradient descent is used, with the update of the parameters being proportional to minus the gradient (the proportionality constant is called the *learning rate*). However, the naïve approach to calculating the gradient of C is time-consuming: to obtain each partial derivative of C the network would need to be run once for each data point x and for each variable in w or b, to see how a small change in the input changes the output. One important reason for the prevalence of neural networks today is that two tricks have been found that can reduce the necessary calculations considerably.

The first trick is to use *stochastic gradient descent*, which means that the partial derivatives are not calculated for all data points x in each step of the gradient descent, but only for a subset, a *mini-batch*, of x. The next step uses another subset, and so on until all subsets have been used (marking the end of an *epoch* of training), whereupon the selection of mini-batches starts over. The use of stochastic gradient descent will only give an approximation to the actual gradient, but if the mini-batches are large enough and the learning rate is low enough, this approximation is usually sufficient.

The second trick is to calculate the partial derivatives not one by one by running the network many times, but by using *back-propagation*. Essentially, back-propagation allows one to calculate all partial derivates by running the network once, noting the result for each neuron, finding the derivative of the cost function with respect to the output from the last layer, and then applying the chain rule repeatedly, going backwards through the network to extract each partial derivative. For more details on how this works, see, e.g., Ref. [Nielsen, 2015].

With the network having so many parameters, often orders of magnitude more than the number of training examples, a valid concern is whether the training will result in overfitting. Over the years, various strategies have been developed to counter this, but that is beyond the scope of this short introduction to classical machine learning.

5.1.4 Reasons for the success of classical machine learning

In explanations of the current success of classical machine learning, three factors are usually brought forward:

- **Data**: There are now a large number of huge data sets that can be used for training of neural networks.
- **Computational power**: We now have much more powerful, and increasingly custom-built, hardware to run machine-learning algorithms on.
- **Algorithms**: Clever algorithms like back-propagation, which enable efficient training, together with a number of other clever tricks discovered in the past decade or so, have led to large improvements.

5.2 Quantum machine learning using qBLAS

To see how quantum computers can aid or enhance machine learning, a first entry point is to note that many machine learning algorithms rely heavily on linear algebra. For classical computation of linear-algebra operations, there are optimized low-level routines called basic linear algebra subprograms (BLAS). For quantum computers, there are several algorithms that deal with linear-algebra problems. Together, these algorithms are sometimes referred to as quantum BLAS (qBLAS).

Some examples of qBLAS are the HHL algorithm for solving systems of linear equations [Harrow et al., 2009], the quantum Fourier transform, and quantum phase estimation for finding eigenvalues and eigenvectors. All of these examples have exponential speed-ups compared to their classical counterparts. However, it is important to "read the fine print" [Aaronson, 2015] for these algorithms. They all rely on the problem being encoded in a quantum random access memory (QRAM). In a QRAM, data is encoded in the probability amplitudes of a large superposition state. For example, a vector $\mathbf{b}$ with n entries can be stored in $\log_2 n$ qubits as $\sum_j b_j |j\rangle$, where b_j are the entries in the vector and $|j\rangle$ are the computational basis states of the qubits.

The problem with the QRAM is that no efficient way is known to encode the data in the QRAM in the first place. The time it takes to encode the problem can therefore negate the exponential speed-up from the qBLAS algorithms. This is sometimes called the *input problem*. There is also an *output problem*: the output of the qBLAS algorithms is not necessarily the direct answer sought, but a state which lets you sample properties of the answer. For example, solving the system of linear equations $A\mathbf{x} = \mathbf{b}$ does not give the solution vector $\mathbf{x}$ as an easily measurable output, but just enables sampling properties of $\mathbf{x}$.

5.3 Quantum support vector machines

5.3.1 Support vector machines

We will now look at an example of a classic machine-learning problem that can be tackled with quantum algorithms: support vector machines (SVMs. A support vector machine is a classifier that divides data points into categories based on some boundary (a hyperplane) in space. To train an SVM is to show it labelled data points such that it can identify the optimal boundary. New unlabelled examples can then be classified by checking on which side of the boundary they fall. This is illustrated in Fig. 5.2. The green line (H_3) does not separate the two categories of points (black and white). The blue and red lines (H_1 and H_2) both separate the points correctly, but the red line (H_2) is optimal, since the distance between H_2 and the nearest points in the training data is maximized. The points nearest the hyperplane thus define the hyperplane. These points are called support vectors.

5.3.2 Classical computation

Mathematically, the problem can be formulated as follows. We are given a set of data points $\mathbf{x}_j$ with labels $y_j \in \{-1, 1\}$. We want to find the equation $\mathbf{w} \cdot \mathbf{x} - b = 0$ defining the optimal separating hyperplane. Here, $\mathbf{w}$ is the normal vector to the hyperplane, normalized such that the closest points in the two classes lie on the hyperplanes $\mathbf{w} \cdot \mathbf{x} - b = \pm 1$. The distance from the optimal separating hyperplane to the closest point is $1/|\mathbf{w}|$. Since we wish to maximize this distance, we should minimize $|\mathbf{w}|$, or, equivalently, $\frac{1}{2}|\mathbf{w}|^2$, under the constraints $y_j(\mathbf{w} \cdot \mathbf{x}_j - b) \geq 1 \; \forall j$.

To perform this minimization under constraints, we introduce Lagrange multipliers, forming the La-

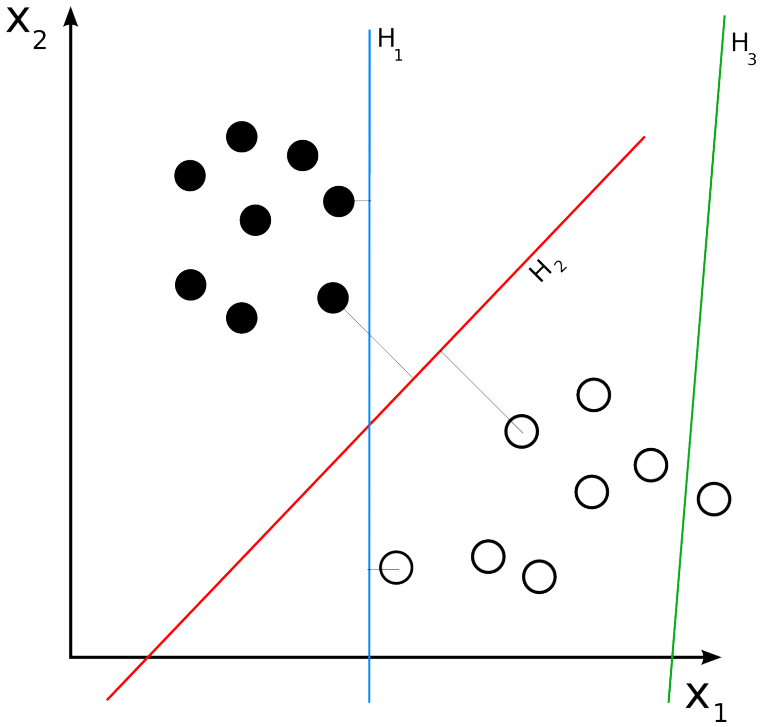


Figure 5.2: Illustration of support vectors. From Wikipedia.

grangian

$$L(\mathbf{w}, b, \lambda) = \frac{1}{2}|\mathbf{w}|^2 - \sum_j \lambda_j [y_j(\mathbf{w} \cdot \mathbf{x}_j - b) - 1]. \tag{5.3}$$

Setting the partial derivatives of L with respect to λ_j equal to zero gives the constraints (the λ_j not corresponding to support vectors will become zero). Setting the partial derivative of L with respect to $\mathbf{w}$ to zero leads to

$$0 = \mathbf{w} - \sum_j \lambda_j y_j \mathbf{x}_j \quad \Rightarrow \quad \mathbf{w} = \sum_j \lambda_j y_j \mathbf{x}_j, \tag{5.4}$$

so we see that $\mathbf{w}$ will be determined by the support vectors. Finally, we also use

$$0 = \frac{\partial L}{\partial b} = \sum_j \lambda_j y_j, \tag{5.5}$$

and substitute these results back into the Lagrangian to obtain

$$\sum_j \lambda_j - \frac{1}{2} \sum \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \tag{5.6}$$

The objective now is to find λ_j that maximize this expression under the constraint given by Eq. (5.5).

In many cases, the separation between two classes of data points cannot be parameterized as a simple hyperplane, as illustrated to the left in Fig. 5.3. The solution commonly used is then to transform the data points to a feature space that admits a hyperplane as a separator. This is encoded by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, which then replaces the dot product in Eq. (5.6).

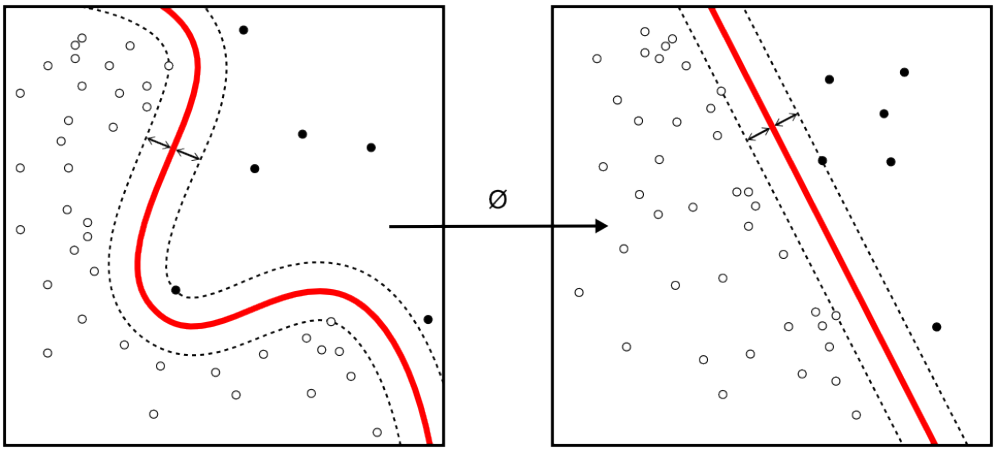


Figure 5.3: Illustration of a kernel for a support vector machine. From Wikipedia.

With this addition, and some further work, the maximization problem in Eq. (5.6) can be shown to lead to the following system of linear equations:

$$\begin{pmatrix} 0 & 1 \\ 1 & K \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{y} \end{pmatrix}, \tag{5.7}$$

where the ones are $1 \times M$ row and column vectors (M is the number of data points) and the entries in the $M \times M$ matrix K are given by $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.

We can now estimate the time it takes to find the support vectors on a classical computer. If the data points $\mathbf{x}_j \in \mathbb{R}^N$, calculating one entry in K takes $\mathcal{O}(N)$ time, so calculating all of K takes $\mathcal{O}(M^2 N)$ time. Solving the system of linear equations takes $\mathcal{O}(M^3)$ time, so in total the classical computer will require $\mathcal{O}(M^2[N+M])$ time.

5.3.3 Quantum computation

As we saw at the end of the previous subsection, the problem of SVMs boils down to two costly computations: calculating the entries in the matrix K and solving the system of linear equations in Eq. (5.7). Quantum algorithms can be applied to both these computations [Rebentrost et al., 2014].

To calculate the dot product $\mathbf{x}_i \cdot \mathbf{x}_j$, we assume that $|\mathbf{x}_i|$ and $|\mathbf{x}_j|$ are known. We then use that

$$\mathbf{x}_i \cdot \mathbf{x}_j = \frac{|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 - |\mathbf{x}_i - \mathbf{x}_j|^2}{2}, \tag{5.8}$$

which reduces our problem to finding the distance $|\mathbf{x}_i - \mathbf{x}_j|^2$. To find this distance, we first construct the two states

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|\mathbf{x}_i\rangle + |1\rangle\,|\mathbf{x}_j\rangle), \tag{5.9}$$

$$|\varphi\rangle = \frac{1}{\sqrt{|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2}}(|\mathbf{x}_i|\,|0\rangle - |\mathbf{x}_j|\,|1\rangle). \tag{5.10}$$

Note that we require QRAM to construct the state $|\mathbf{x}_i\rangle$. Next we do a "swap test" (see Fig. 5.4) on $|\varphi\rangle$ and the ancilla qubit in $|\psi\rangle$. The distance we seek is then given by $\sqrt{|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2}$ times the probability of measuring 1 in the swap test. The computational complexity for this distance calculation is $\mathcal{O}(\log N)$.

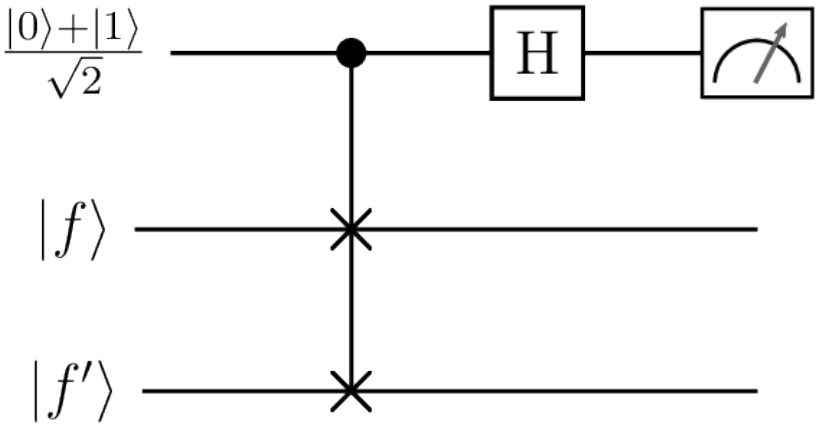


Figure 5.4: The quantum circuit for a swap test. Figure from Peter Wittek. The resulting state of the system before the measurement is $\frac{1}{2}[|0\rangle(|f\rangle|f'\rangle + |f'\rangle|f\rangle) + |1\rangle(|f\rangle|f'\rangle - |f'\rangle|f\rangle)]$. This means that the probability of measuring 1 becomes zero when $f = f'$.

To solve the system of linear equations, we need to invert the matrix

$$F = \begin{pmatrix} 0 & 1 \\ 1 & K \end{pmatrix}. \tag{5.11}$$

Briefly, this is done by approximating $\exp\{-iFt\}$ (which is not trivial, since K is not sparse), and then using quantum phase estimation to extract eigenvalues and eigenvectors. These eigenvalues, together with $\mathbf{y}$ in the eigenbasis, lets us construct the solution state

$$|b, \lambda\rangle \propto b|0\rangle + \sum_j \lambda_j |j\rangle. \tag{5.12}$$

The complexity for this part of the computation is $\mathcal{O}(\log M)$. The total complexity for the quantum SVM is thus $\mathcal{O}(\log NM)$.

5.4 Quantum principal component analysis

The methods applied in the quantum approach to SVMs can also be used in other machine-learning problems. For example, the algorithm for distance calculation can be applied to clustering. Another example is principal component analysis (PCA), where the second part of the quantum SVM algorithm can be re-used. In PCA, an unlabelled set of (high-dimensional) data points $\mathbf{x}_j$ is analyzed to find which are the axes along which the data varies the most (and which thus are most useful for classification). A nice example of PCA for physics researchers is Paperscape (paperscape.org), which takes data from all papers on the arXiv and uses PCA to show the relations between the papers on a two-dimensional map.

A quantum algorithm for PCA [Lloyd et al., 2014] uses the matrix exponentiation and phase estimation from the quantum algorithm for SVMs to find the largest eigenvalues and eigenvectors of the covariance matrix $\sum_j \mathbf{x}_j \mathbf{x}_j^T$. Those eigenvectors are the principal components. We note here that there is recent work on "quantum-inspired" algorithms for PCA [Tang, 2018], where the methods from the quantum algorithm have been adapted to find an improved (in the scaling of some parameters) classical algorithm.

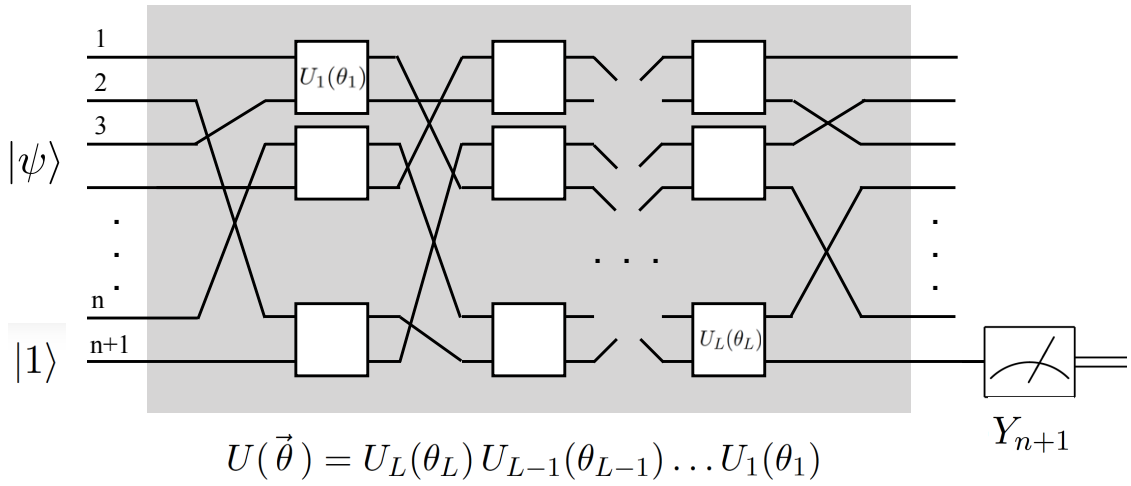


Figure 5.5: A quantum feedforward neural network. Figure from Ref. [Farhi and Neven, 2018].

5.5 Quantum neural networks

In this final section, we discuss quantum versions of neural networks. This is a quite new and rapidly developing field, so it is possible that these notes can become outdated fast. We therefore only try to give a few examples and discuss some general properties of quantum neural networks.

Although combining quantum computing and neural networks sounds interesting, given how they are both promising computing paradigms, it is not straightforward to do so. There are certainly potential upsides to quantum neural networks: they can work with quantum data, or a compact representation of classical data, and there may be quantum algorithms that can speed up training. However, there are several potential downsides or questions. For example, the input problem may be a factor here also. Furthermore, classical neural networks require nonlinear activiation functions, but quantum mechanics is linear. Also, classical neural networks have a large number of neurons and many connections between them, which may be challenging for NISQ devices with few qubits and limited connectivity. Another question is whether back-propagation can be implemented in a quantum neural network, since classical back-propagation requires measuring the output at each point in the network, something that would destroy a quantum superposition.

5.5.1 Quantum feedforward neural networks

In 2018, Farhi (yes, the same Farhi that proposed the QAOA) and Neven proposed an architecture for a quantum feedforward neural network [Farhi and Neven, 2018], as depicted in Fig. 5.5. Here, instead of layers of neurons, there are layers of quantum gates, which act on an n-qubit input state $|\psi\rangle$ and an ancilla qubit prepared in $|1\rangle$. Instead of weights and biases in a classical neural networks, we now have parameters ϑ_j parameterizing these gates; the action of the whole quantum circuit is a unitary operation $U(\vartheta)$. At the end, the ancilla is measured (here, in the Y basis). The outcome of the measurement is used to classify the input state, giving it one of two possible labels.

To train this quantum neural network for its classification task, the authors propose using a loss function

$$C(\vartheta, z) = 1 - l(z)\langle z, 1|U^\dagger(\vartheta) Y_{n+1} U(\vartheta)|z, 1\rangle, \tag{5.13}$$

where z is the input and $l(z)$ is a label function giving the correct label (± 1). This cost function is zero if the network spits out the correct classification, and greater than zero otherwise. The authors use stochastic

gradient descent to find parameters ϑ that minimize this cost function. However, no back-propagation is used (since this does not seem to be applicable to this architecture, as discussed above). Furthermore, to evaluate the gradient, multiple runs of the quantum circuit are required for each partial derivative, since one needs to collect enough statistics to find the expectation value of the output with sufficient precision. The authors point out that a possible advantage of the quantum network is that the form of the unitary U guarantees that the gradient does not "blow up", which can be a problem in some classical machine-learning algorithms.

Since the quantum feedforward neural network lacks a nonlinear element, one can ask whether it has the ability to represent any label function. The authors show that the network indeed has this ability, but some label functions may require an exponential circuit depth.

5.5.2 Quantum convolutional neural networks

Another recent proposal [Cong et al., 2019] for a quantum neural network is a quantum version of a convolutional neural network (CNN), illustrated in Fig. 5.6. Convolutional neural networks are often used for image recognition. As depicted in Fig. 5.6(a), a classical CNN consists of convolutional layers (C) that essentially scan a filter across the image, pooling layers (P) that reduce the size of the feature map produced by the convolution, and a final part with fully connected layers (FC) that do classification based on the features extracted in previous layers. In the quantum version [Fig. 5.6(b)], the filter in convolutional layer number j is replaced by a two-qubit unitary operation U_j, which is applied to all pairs of neighbouring qubits. The pooling layer number k is replaced by measuring half of the qubits and applying a unitary single-qubit operation V_k to the remaining qubits, conditioned on the measurement outcome for the neighbouring qubit. This operation conditioned on measurement has the added benefit of adding a nonlinearity to the setup. Finally, the fully connected layer is replaced by a quantum circuit similar to that proposed by Farhi and Neven, as discussed in the preceding subsection. Similar to that proposal, the training here is also done by gradient descent without any back-propagation.

5.5.3 Quantum Boltzmann machines

The neural networks that have been most frequently studied in some quantum form are Boltzmann machines [Amin et al., 2018], which come in a few different architectures, as shown in Fig. 5.7. A Boltzmann machine has hidden ($\mathbf{h}$) and visible ($\mathbf{v}$) neurons, which take binary values (0 or 1). The aim of training a Boltzmann machine is to make the states of its visible neurons follow a probability distribution $P(\mathbf{v})$ that mimics that of the training data. This probability distribution is given by

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})}, \tag{5.14}$$

where Z is a normalization factor (partition function) and the energy function is, for the case of a restricted Boltzmann machine (Fig. 5.7 center),

$$E(\mathbf{v},\mathbf{h}) = \sum_i a_i v_i + \sum_j b_j h_j + \sum_{i,j} v_i W_{ij} h_j, \tag{5.15}$$

with a_i and b_j biases, and W a weight matrix. The form of the probability distribution (a Boltzmann distribution) explains the name Boltzmann machines.

To train the network, a cost function

$$C(\mathbf{a},\mathbf{b},W) = -\sum_{\mathbf{v}} P_{\text{data}}(\mathbf{v}) \log P(\mathbf{v}) \tag{5.16}$$

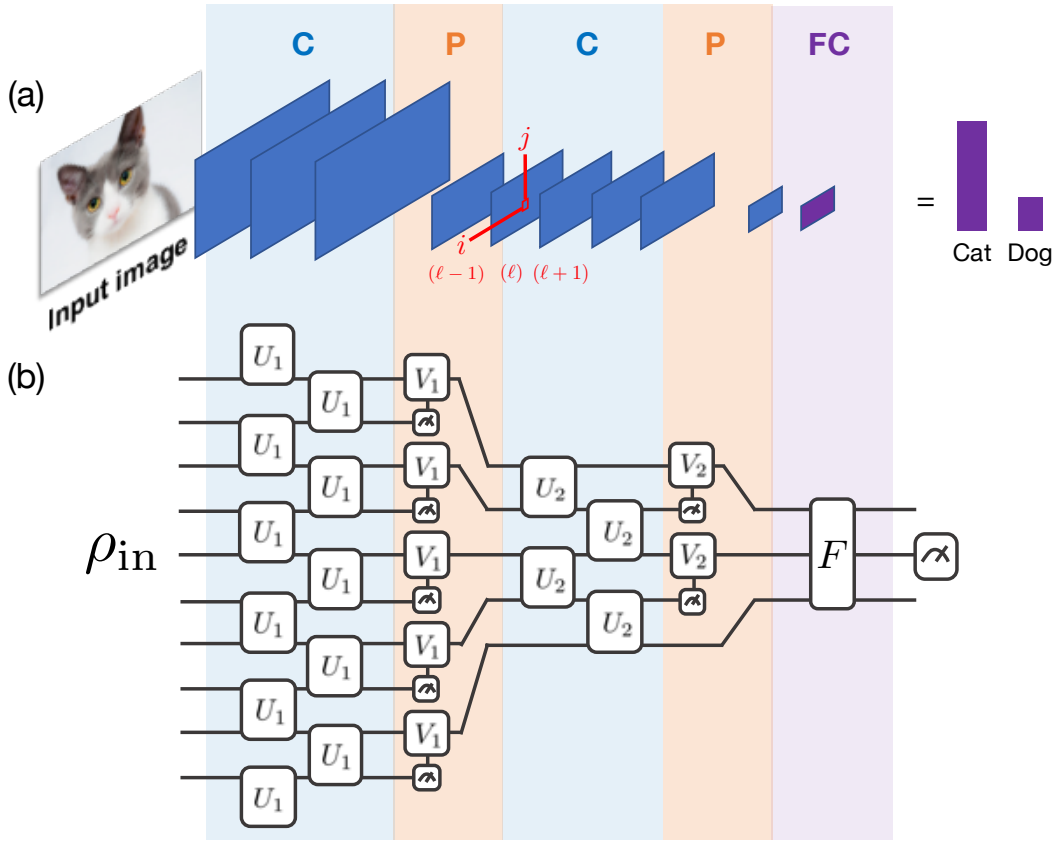


Figure 5.6: Illustrations of (a) a classical convolutional neural network and (b) a quantum convolutional neural network. Figure from Ref. [Cong et al., 2019].

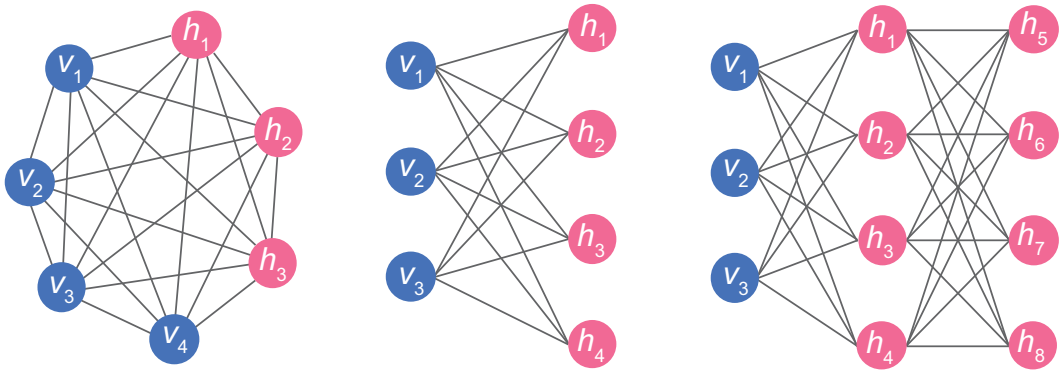


Figure 5.7: Layouts for Boltzmann machines. Left: A fully connected Boltzmann machines, with connections both between the hidden and visible layers, and within the hidden and visible layers. Center: A restricted Boltzmann machine, with connections only going between the hidden and visible layers. Right: A deep Boltzmann machine, with multiple hidden layers. Figure from Ref. [Allcock and Zhang, 2019].

is minimized using gradient descent, often aided by an efficient scheme for sampling from the training data. It is unclear if quantum algorithms can be used for speeding up training of classical Boltzmann machines.

To construct a quantum version of a Boltzmann machine, the neurons are replaced by spins and the energy function is given by a Hamiltonian for the spin system. It is believed that such a quantum Boltzmann machines can represent some probability distributions, particularly ones arising in quantum systems, better than classical Boltzmann machines can (note that classical Boltzmann machines already are used for simulating some quantum systems well). However, the scaling properties and other performance metrics for quantum Boltzmann machines are still unclear. Furthermore, it seems that training of a quantum Boltzmann machine can be more complicated than the training of a classical one [Amin et al., 2018]. A potential upside is that quantum annealers could be used to implement quantum Boltzmann machines.

Chapter 6

Measurement-based quantum computation

The circuit model introduced and discussed in the previous two chapters is not the only way to perform quantum computation. In this and the next chapter, we will look at two other approaches: measurement-based quantum computation (MBQC) and adiabatic quantum computation. We will also see some additional examples later in the course; MBQC will resurface when we discuss quantum computation with continuous variables.

Measurement-based quantum computation was first proposed by Raussendorf and Briegel in 2001 [Raussendorf and Briegel, 2001]. More details can be found in the follow-up paper in Ref. [Raussendorf et al., 2003]. The first experimental demonstration of all basic components of MBQC was performed by the Zeilinger group in 2005 [Walther et al., 2005].

6.1 The basic idea of MBQC

In the circuit model of quantum computation, we prepare an initial state, apply a sequence of quantum gates to this state, and finally do some measurement on the output state to obtain the result of the computation. While this seems a natural order of operations, which agrees with our intuition for how a classical computation is performed, it is possible to mix up the order somewhat.

In MBQC, we first prepare a certain entangled state of qubits, where a subset of these qubits represent the input state. This entangled state is the *resource* for our computation. In the rest of the computation, we never need to apply any multi-qubit operations. All we do is apply single-qubit rotations and single-qubit measurements in some sequence, where later rotations are conditioned on earlier measurement results. The output state will be encoded in a subset of the qubits (different from the subset that encoded the input), and a final measurement can be performed on this output state to read out the result of the computation.

Because of the presence of measurements in earlier steps of the computation, MBQC differs from the circuit model for quantum computation in one important aspect: it is not reversible. In the circuit model, we apply a large unitary transformation (decomposed into a sequence of gates) that takes us from the input state to the output state. Before any measurement is done on the output state, this unitary transformation could be reversed to bring us back to the input state. However, since measurements are done before we reach the output state in MBQC, that computation cannot be reversed. For this reason, MBQC is often referred to as one-way quantum computation.

6.2 The details of MBQC

6.2.1 Definition of the possible operations

Let us first recall from Chapter 1 some of the operations we can perform on single and multiple qubits.

Single-qubit Clifford transformations

Clifford operations C are the unitary operations which map Pauli group operators Σ to Pauli group operators Σ' under conjugation, i.e., $C\Sigma C^\dagger = \Sigma'$ [Horodecki et al., 2006].

$$\left\{H, R_z(\pi/2) = Z_{\pi/2}\right\} \quad \textbf{universal set for single-qubit Clifford operations}$$

Multiqubit Clifford transformations

The addition of any non trivial two-qubit Clifford gate, e.g., $CZ = |0\rangle\langle 0| \otimes I + |1\,|1\rangle\langle 1| \otimes \hat{Z}$, combined with the set of single-qubit operations above allows for any general multi-qubit Clifford operation.

$$\left\{H, U_z(\pi/2) = Z_{\pi/2}, CZ\right\} \quad \textbf{universal set for multi-qubit Clifford operations}$$

These operations are enough to perform some algorithms like error correcting codes, but no algorithm which could not be efficiently simulated on a classical computer [Horodecki et al., 2006].

Single-qubit universal transformations

A general single-qubit transformation can be decomposed as $R_z(\gamma)R_x(\beta)R_z(\alpha)$. To implement any such arbitrary operation with arbitrary accuracy, it is sufficient to be able to perform the Clifford operations together with any non-Clifford operation provide, e.g.,

$$\left\{H, Z_{\pi/2}, Z_{\pi/4}\right\} \quad \textbf{universal set for single-qubit quantum computation}$$

Multi-qubit universal transformations

If we have universal control of single qubits, adding a non-trivial (entangling) two-qubit gate to the gate set is enough to achieve universality for multiple qubits.

$$\left\{H, Z_{\pi/2}, Z_{\pi/4}, CZ\right\} \quad \textbf{universal set for multi-qubit quantum computation}$$

6.2.2 Preparing the initial state

We first consider a setup with two qubits. One qubit contains the initial state that we want to process: $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. The other qubit is prepared in $|+\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$, e.g., by applying a Hadamard gate to an initial state $|0\rangle$. We then apply a CZ gate between the two qubits, obtaining

$$\begin{aligned} CZ(|\psi\rangle \otimes |+\rangle) &= \left[|0\rangle\langle 0| \otimes 1 + |1\rangle\langle 1| \otimes \hat{Z}\right]\left((\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \\ &= \left[\alpha|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \beta|1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] \\ &= \alpha|0\rangle|+\rangle + \beta|1\rangle|-\rangle \end{aligned} \tag{6.1}$$

6.2.3 Measurements and their effect

The next step is to measure the input qubit in some rotated basis. Measuring in a rotated basis with angles (ϑ, φ) is like measuring $R_z(\varphi + \pi/2)R_x(\vartheta)ZR_x(-\vartheta)R_z(-\varphi - \pi/2)$. Here we measure the first qubit with $\vartheta = \pi/2$ and an arbitrary φ, i.e., we measure the observable

$$\begin{aligned}\hat{\sigma}_\varphi &\equiv R_z(\varphi + \pi/2)R_x(\pi/2)ZR_x(-\pi/2)R_z(-\varphi - \pi/2) \\ &= \cos\varphi X + \sin\varphi Y \\ &= e^{-i\varphi}|0\rangle\langle 1| + e^{i\varphi}|1\rangle\langle 0| = |\varphi_+\rangle\langle\varphi_+| - |\varphi_-\rangle\langle\varphi_-|, \end{aligned} \tag{6.2}$$

where we have used Eqs. (1.3)-(1.7) and introduced the rotated measurement basis $|\varphi_\pm\rangle = 1/\sqrt{2}(|0\rangle \pm e^{i\varphi}|1\rangle)$. To obtain the result, note that conversely $|0\rangle = \frac{1}{\sqrt{2}}(|\varphi_+\rangle + |\varphi_-\rangle)$ while $|1\rangle = \frac{1}{\sqrt{2}}e^{-i\varphi}(|\varphi_+\rangle - |\varphi_-\rangle)$, and rewrite the state:

$$\begin{aligned}\alpha|0\rangle|+\rangle + \beta|1\rangle|-\rangle &= \frac{\alpha}{\sqrt{2}}(|\varphi_+\rangle + |\varphi_-\rangle)|+\rangle + \frac{\beta}{\sqrt{2}}e^{-i\varphi}(|\varphi_+\rangle - |\varphi_-\rangle)|-\rangle \\ &= |\varphi_+\rangle\left(\frac{\alpha}{\sqrt{2}}|+\rangle + \frac{\beta}{\sqrt{2}}e^{-i\varphi}|-\rangle\right) + |\varphi_-\rangle\left(\frac{\alpha}{\sqrt{2}}|+\rangle - \frac{\beta}{\sqrt{2}}e^{-i\varphi}|-\rangle\right).\end{aligned}$$

Hence a measurement of the operator in Eq. (6.2) projects the second qubit into:

- $|\psi\rangle_{\text{out}} \propto (\alpha|+\rangle + \beta e^{-i\varphi}|-\rangle)$ if the outcome is 1 $(m = 0)$
- $|\psi\rangle_{\text{out}} \propto (\alpha|+\rangle - \beta e^{-i\varphi}|-\rangle)$ if the outcome -1 $(m = 1)$

The state of the second qubit can then be compactly written as

$$|\psi\rangle_{\text{out}} = X^m HR_z(-2\varphi)|\psi\rangle. \tag{6.3}$$

This is readily verified since

$$\begin{aligned}X^m HR_z(-2\varphi)(\alpha|0\rangle + \beta|1\rangle) &= X^m H(\alpha e^{i\varphi}|0\rangle + \beta e^{-i\varphi}|1\rangle) \\ &= X^m(\alpha e^{i\varphi}|+\rangle + \beta e^{-i\varphi}|-\rangle),\end{aligned} \tag{6.4}$$

giving the result. The extra Pauli operator X^m depends on the outcome of the measurement on qubit 1 and is said to be a "by-product" operator. It can be compensated for by choosing the measurement basis of the following steps in the computation (thus introducing, in general, time-ordering). In the following we rename $-2\varphi \to \varphi$.

6.2.4 Universal single-qubit operations

We can repeat the preceding protocol three times by using four qubits, first using qubit 1 as input and qubit 2 as output, then using qubit 2 as input and qubit 3 as output, and so on. The final output state (qubit 4) then becomes

$$\begin{aligned}|\psi\rangle_{\text{out}} &= X^{m_3}HR_z(\varphi_3)X^{m_2}HR_z(\varphi_2)X^{m_1}HR_z(\varphi_1)|\psi\rangle \\ &= HZ^{m_3}R_z(\varphi_3)HZ^{m_2}R_z(\varphi_2)HZ^{m_1}R_z(\varphi_1)|\psi\rangle \\ &= HZ^{m_3}R_z(\varphi_3)(HZ^{m_2}H)(HR_z(\varphi_2)H)Z^{m_1}R_z(\varphi_1)|\psi\rangle \\ &= HZ^{m_3}R_z(\varphi_3)X^{m_2}R_x(\varphi_2)Z^{m_1}R_z(\varphi_1)|\psi\rangle \\ &= X^{m_3}Z^{m_2}X^{m_1}HR_z((-1)^{m_2}\varphi_3)R_x((-1)^{m_1}\varphi_2)R_z(\varphi_1)|\psi\rangle,\end{aligned} \tag{6.5}$$

where in the first step we used that $X^m H = HZ^m$, in the third that $HZ^m H = X^m$ and later on that $XR_z(\varphi) = R_z(-\varphi)X$ and $ZR_x(\varphi) = R_x(-\varphi)Z$.

Since the most general rotation of a single qubit can be decomposed as $R_z(\gamma)R_x(\beta)R_z(\alpha)$, the above steps lets us implement any single-qubit operation. For the result to be deterministic, we can perform the measurements choosing the basis sequentially, depending on the preceding measurement outcomes, as $\varphi_1 = \alpha$, $\varphi_2 = (-1)^{m_1}\beta$, $\varphi_3 = (-1)^{m_2}\gamma$. The Pauli corrections remaining at the end of the computation are not important and never need to be physically applied; they can be accounted for in the final interpretation of the result (classical post-processing).

If we want to implement a Clifford unitary, by definition $C\Sigma = \Sigma' C$ meaning that interchanging the order of Clifford operators and Pauli matrices will leave the Clifford operator unchanged. This means that there is no need to choose measurements adaptively.

6.2.5 Cluster states as a resource

In the example above, the four qubits were entangled through CZ gates between nearest neighbours. These CZ gates could all have been done before the start of the computation (when qubit 1 was in the input state and qubits 2–4 were in the state $|+\rangle$). This entangled initial state would then be a one-dimensional *cluster state*, which is the resource enabling the rest of the computation being carried by just measurements and a limited set of single-qubit operations.

More generally, a state like this which allows any input state and any unitary transformation on that input (using only single-qubit rotations and measurements) is said to be a universal resource. It has been shown that a square lattice graph (a cluster state) with unit weights is a universal resource for quantum computation. Despite this fact, depending on the specific kind of computation other graphs than a square lattice could be more suitable for implementing the computation [Horodecki et al., 2006].

6.2.6 Two-qubit gates

For universal quantum computation, we need some two-qubit gate in addition to the arbitrary single-qubit rotations that we constructed above. Such two-qubit gates, e.g., the CZ and CNOT gates, can be constructed in a two-dimensional cluster state where two input qubits are entangled with a few other qubits. By a series of single-qubit measurements and rotations, we can end up with two of the other qubits representing the output state corresponding to the two-qubit gate having acted on the input state. The example of the CNOT gate will be demonstrated during the first tutorial of the course.

6.3 Universality and efficiency

From the previous section, it is clear that we can implement arbitrary single-qubit rotations and a two-qubit gate (e.g., CZ or CNOT), which together form a universal gate set for quantum compuation (cf. Chapter 1). Thus, MBQC can implement universal quantum computation.

However, it is clear that MBQC in general requires more qubits than the circuit model does. It is also necessary to compare the number of qubit rotations and measurements needed in MBQC and in the circuit model. If MBQC would turn out to simply be an inefficient reformulation of the circuit model, it would not be useful. Fortunately, it turns out that the overhead of MBQC is polynomial in all these resources, so MBQC is an efficient paradigm for quantum computation.

The efficiency of MBQC opens up new avenues for practical implementations of quantum computation. There may be physical systems where it is hard to implement single two-qubit gates between chosen qubits

at chosen times, but where it is easier to create a large entangled state in one big operation and then only perform single-qubit operations.

As an aside, we note that MBQC usually has been considered for two-dimensional cluster states. However, by working with cluster states in three dimensions, the quantum computation can be made fault-tolerant [Briegel et al., 2009].

Chapter 7

Adiabatic quantum computation

The MBQC that we considered in Chapter 6 was, although distinct from the circuit model for quantum computation, still rather similar to that circuit model. The paradigm of quantum computation that we explore in this chapter, adiabatic quantum computation (AQC), is further removed from the circuit model than MBQC. The idea for AQC was evolved around the turn of the millenium. A recent extensive review of the topic is Ref. [Albash and Lidar, 2018], from which we quote several of the following considerations.

7.1 The basic idea of AQC

Adiabatic quantum computation is based on the *adiabatic theorem* (we give a proof for this theorem in Sec. 7.5). The theorem states that a system which starts in a non-degenerate ground state of a time-dependent Hamiltonian $\hat{\mathcal{H}}(t)$ that is *slowly* changing from some initial form $\hat{\mathcal{H}}_0$ to some final form $\hat{\mathcal{H}}_1$, during time τ, will remain in its instantaneous ground state throughout the evolution, provided that the Hamiltonian varies sufficiently slowly. Assuming a linear time dependence the AQC Hamiltonian can be written as

$$\hat{\mathcal{H}}(t) = \left(1 - \frac{t}{\tau}\right)\hat{\mathcal{H}}_0 + \frac{t}{\tau}\hat{\mathcal{H}}_1, \quad (0 \leq t \leq \tau), \tag{7.1}$$

where the coefficient in front of $\hat{\mathcal{H}}_0$ changes from unity to zero, and the coefficient in front of $\hat{\mathcal{H}}_1$ changes from zero to unity. Moreover, it is crucial that $\hat{\mathcal{H}}_0$ and $\hat{\mathcal{H}}_1$ are two noncommuting Hamiltonians (see Sec. 7.4).

If the initial Hamiltonian $\hat{\mathcal{H}}_0$ is such that its ground state is easy to construct, it is easy to initialize the system in this ground state. The Hamiltonian $\hat{\mathcal{H}}_1$ is chosen such that its ground state encodes the solution of a certain problem. This means that the adiabatic evolution will provide us with this solution.

Adiabatic quantum computation is closely related to quantum annealing (QA), which also encodes the solution to a problem in the ground state of some final Hamiltonian. The relation between AQC and QA is discussed later in the course.

7.2 Adiabatic evolution and quantum speed-up

What is the minimum evolution time τ, such that the time evolution in Eq. (7.1) is adiabatic? According to the adiabatic theorem, it can be shown that for a non-degenerate ground state, the adiabatic evolution is

assured if the evolution time τ satisfies the condition

$$\tau \gg \frac{\max\limits_{0\le s\le 1}\left|\langle\psi_1(s)|\,\frac{d\hat{\mathcal{H}}(s)}{ds}\,|\psi_0(s)\rangle\right|}{\min\limits_{0\le s\le 1}\Delta^2(s)}; \qquad s\equiv\frac{t}{\tau}, \tag{7.2}$$

where $|\psi_0(s)\rangle$ and $|\psi_1(s)\rangle$ are the instantaneous ground and first excited eigenstate of the Hamiltonian in Eq. (7.1), and $\Delta(s) = (E_1(s) - E_0(s))$ is the instantaneous energy gap between the ground state and first excited state energies. If the criterion of Eq. (7.2) is fulfilled, then we can be certain that the system will evolve into the ground state of $\hat{\mathcal{H}}_1$.

While these are useful sufficient conditions, they involve bounding the minimum eigenvalue gap of a complicated many-body Hamiltonian, a notoriously difficult problem. This is one reason that AQC has generated so much interest among physicists: it has a rich connection to well-studied problems in condensed matter physics. For example, because of the dependence of the run time on the gap, the performance of quantum adiabatic algorithms is strongly influenced by the type of quantum phase transition the same system would undergo in the thermodynamic limit.

Nevertheless, a number of examples are known where the gap analysis can be carried out. For example, adiabatic quantum computers can perform a process analogous to Grover search and thus provide a quadratic speedup over the best possible classical algorithm for the Grover search problem. Other examples are known where the gap analysis can be used to demonstrate that AQC provides a speedup over classical computation, including adiabatic versions of some of the key algorithms of the circuit model.

However, much more common is the scenario where either the gap analysis reveals no speedup over classical computation, or where a clear answer to the speedup question is unavailable. In fact, the least is known about adiabatic quantum speedups in the original setting of solving classical combinatorial optimization problems. This remains an area of very active research, partly due to the original (still unmaterialized) hope that AQC would deliver quantum speedups for NP-complete problems [Farhi et al., 2001], and partly due the availability of commercial QA devices such as those manufactured by D-Wave Systems Inc., designed to solve optimization problems using stoquastic Hamiltonians.

7.3 Universality and stoquasticity

Adiabatic quantum computation is a universal model of quantum computation. Problems that are solvable in polynomial time with the circuit model can be solved in polynomial time with an adiabatic quantum computer, and vice versa. The fact that the circuit model can efficiently simulate AQC was proven by Edward Fahri et al. in 2000. The fact that AQC can efficiently simulate the circuit model was proven by Dorit Aharonov in 2007. As such, the two models are computationally equivalent. The proofs for these statements can be found in Ref. [Albash and Lidar, 2018].

When discussing universality and efficiency for AQC, it is important to distinguish whether the final Hamiltonian $\hat{\mathcal{H}}_1$ is *stoquastic* or not. A Hamiltonian H is called stoquastic if it only has real-valued elements, and, in some local basis B, all off-diagonal elements of H are either zero or negative, i.e., $\langle x|H|y\rangle \le 0 \forall x, y \in B$. Stoquastic Hamiltonians seem to exist in the borderlands between the classical and quantum worlds. It is an open question whether stoquastic AQC can be efficiently simulated by a classical computer. It has been shown that stoquastic AQC cannot be universal for quantum computation unless the polynomial hierarchy collapses. The proof by Aharonov of the universality for AQC thus requires non-stoquastic Hamiltonians.

7.4 Reason for non-commuting Hamiltonians

As mentioned above, it is important in AQC that the initial Hamiltonian $\hat{\mathcal{H}}_0$ and final Hamiltonian $\hat{\mathcal{H}}_1$ do not commute, i.e., $\left[\hat{\mathcal{H}}_0, \hat{\mathcal{H}}_1\right] \neq 0$. This can be understood by considering the following trivial example, taken from Pontus Vikstål's Master thesis, Ref. [Vikstål, 2018]. Suppose that the initial and final Hamiltonian in AQC are be given by

$$\hat{\mathcal{H}}_0 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{and} \quad \hat{\mathcal{H}}_1 = \begin{pmatrix} -1 & 0 \\ 0 & -\frac{1}{2} \end{pmatrix}, \tag{7.3}$$

which clearly commute. Since the Hamiltonians are both diagonal in the z-basis, we label the corresponding eigenvectors as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{7.4}$$

It is easy to see that the ground state of $\hat{\mathcal{H}}_0$ is $|1\rangle$ and that the ground state of $\hat{\mathcal{H}}_1$ is $|0\rangle$. We described above how AQC is based on the adiabatic theorem. For the theorem to hold there must always exist an energy gap between the eigenstates (see the proof below in Sec. 7.5). Following the AQC algorithm Eq. (7.1), it can be seen that the energy gap between the eigenstates $|1\rangle$ and $|0\rangle$ closes at some point. In this example, the energies becomes equal at the point $t/\tau = 4/5$, and so the gap between them closes. This is enough to see that $\left[\hat{\mathcal{H}}_0, \hat{\mathcal{H}}_1\right] \neq 0$ is a necessary condition for keeping the gap open.

7.5 Proof of the adiabatic theorem

In this section, we provide a simple and straightforward proof of the famous *adiabatic theorem*, first published by Max Born and Vladimir Fock in 1928. This section is taken from Pontus Vikstål's Master thesis, Ref. [Vikstål, 2018].

Theorem: A particle that begins from the nth eigenstate of a Hamiltonian that is gradually (adiabatically) changing from an initial form $\hat{\mathcal{H}}_i$ into a final form $\hat{\mathcal{H}}_f$, will remain in the nth eigenstate.

Consider an arbitrary time-independent Hamiltonian $\hat{\mathcal{H}}$, for which the Schrödinger equation reads

$$i\hbar \frac{d\Psi(x,t)}{dt} = \hat{\mathcal{H}}\Psi(x,t). \tag{7.5}$$

Through separation of variables, the time-independent Schrödinger equation can be obtained:

$$\hat{\mathcal{H}}\psi_n(x) = E_n\psi_n(x). \tag{7.6}$$

The general solution to the Schrödinger equation is given by a superposition of the separable solutions

$$\Psi(x,t) = \sum_n c_n \Psi_n(x,t) = \sum_n c_n \psi_n(x) e^{-iE_n t/\hbar}, \tag{7.7}$$

or by simply considering a specific eigenfunction

$$\Psi_n(x,t) = \psi_n(x) e^{-iE_n t/\hbar}. \tag{7.8}$$

Hence, the nth eigenstate for a time-independent Hamiltonian remains in the nth eigenstate, simply picking up a phase factor $-E_n t/\hbar$. For a time-dependent Hamiltonian, the eigenenergies and eigenfunctions are themselves time-dependent. The instantaneous eigenstates and eigenenergies are defined as

$$\hat{\mathcal{H}}(t)\psi_n(x,t) = E_n(t)\psi_n(x,t). \tag{7.9}$$

At any instant of time, the eigenfunctions form a complete orthogonal set

$$\langle \psi_m(t)|\psi_n(t)\rangle = \delta_{mn}, \tag{7.10}$$

where the dependence on position is implicit. We have also introduced the bra-ket notation $\psi_n(x) = \langle x|\psi_n\rangle$. The general solution to the Schrödinger equation is now given by

$$\Psi(x,t) = \sum_n c_n(t)\Psi_n(x,t) = \sum_n c_n(t)\psi_n(x,t)e^{i\vartheta_n(t)}, \tag{7.11}$$

where $\vartheta_n(t)$ is known as the *dynamical phase* factor

$$\vartheta_n(t) = -\frac{1}{\hbar}\int_0^t E_n(s)ds. \tag{7.12}$$

Our task is to determine the coefficients $c_n(t)$. By substituting (7.11) into the Schrödinger equation, we obtain

$$i\hbar\sum_n(\dot{c}_n\psi_n + c_n\dot{\psi}_n + ic_n\psi_n\dot{\vartheta}_n)e^{i\vartheta_n} = \sum_n c_nE_n\psi_ne^{i\vartheta_n}. \tag{7.13}$$

The third term on the left cancels the term on the right, since $\dot{\vartheta}_n = E_n$. We are thus left with

$$i\hbar\sum_n(\dot{c}_n\psi_n + c_n\dot{\psi}_n)e^{i\vartheta_n} = 0. \tag{7.14}$$

Multiplying with an arbitrary eigenfunction $\langle\psi_m|$ from the left and using the orthogonality condition Eq. (7.10) yields

$$\dot{c}_m = -\sum_n c_n\langle\psi_m|\dot{\psi}_n\rangle e^{i(\vartheta_n-\vartheta_m)}. \tag{7.15}$$

To calculate the quantity $\langle\psi_m|\dot{\psi}_n\rangle$, we first observe that for $m \neq n$

$$\begin{aligned}\frac{d}{dt}(\langle\psi_m|\hat{\mathcal{H}}|\psi_n\rangle) = 0 &= \langle\dot{\psi}_m|\underbrace{\hat{\mathcal{H}}|\psi_n\rangle}_{E_n|\psi_n\rangle} + \langle\psi_m|\dot{\hat{\mathcal{H}}}|\psi_n\rangle + \underbrace{\langle\psi_m|\hat{\mathcal{H}}}_{E_m\langle\psi_m|}|\dot{\psi}_n\rangle \\ &= E_n\langle\dot{\psi}_m|\psi_n\rangle + \langle\psi_m|\dot{\hat{\mathcal{H}}}|\psi_n\rangle + E_m\langle\psi_m|\dot{\psi}_n\rangle.\end{aligned} \tag{7.16}$$

We then note that

$$\frac{d}{dt}(\underbrace{\langle\psi_m|\psi_n\rangle}_{\delta_{mn}}) = 0 = \langle\dot{\psi}_m|\psi_n\rangle + \langle\psi_m|\dot{\psi}_n\rangle, \tag{7.17}$$

which implies the relation $\langle\dot{\psi}_m|\psi_n\rangle = -\langle\psi_m|\dot{\psi}_n\rangle$, so

$$\langle\psi_m|\dot{\psi}_n\rangle = \frac{\langle\psi_m|\dot{\hat{\mathcal{H}}}|\psi_n\rangle}{E_n - E_m}, \quad (m \neq n). \tag{7.18}$$

This holds as long as no transitions between eigenstates occur. The differential equation in Eq. (7.15) can now be written

$$\dot{c}_m = -c_m\langle\psi_m|\dot{\psi}_m\rangle - \sum_{m\neq n}\frac{\langle\psi_m|\dot{\hat{\mathcal{H}}}|\psi_n\rangle}{E_n - E_m}. \tag{7.19}$$

Now, if the Hamiltonian is slowly changing, such that its time derivative can be considered to be very small and that the energy difference $|E_n - E_m|$ is large compared to $\left|\langle\psi_m|\dot{\hat{\mathcal{H}}}|\psi_n\rangle\right|$, the second term becomes negligible. This approximation is known as the *the adiabatic approximation*, and we conclude that

$$\dot{c}_m \approx -c_m\langle\psi_m|\dot{\psi}_m\rangle. \tag{7.20}$$

By solving this equation one finds

$$c_m(t) = c_m(0) \exp\left(- \int_0^t \langle \psi_m(s) | \dot{\psi}_m(s) \rangle ds \right) = c_m(0) e^{i\gamma_m(t)}, \tag{7.21}$$

where

$$\gamma_m(t) = i \int_0^t \langle \psi_m(s) | \dot{\psi}_m(s) \rangle ds, \tag{7.22}$$

is the *geometrical* (Berry) phase factor. Putting the obtained expression for the coefficients $c_m(t)$ back into (7.11), we obtain that the nth eigenstate is given by

$$|\Psi_n(t)\rangle = e^{i\vartheta_n(t)} e^{i\gamma_n(t)} |\psi_n(t)\rangle \,. \tag{7.23}$$

Hence, a system that starts out in the nth eigenstate, will remain in the nth eigenstate, simply picking up a couple of phase factors.

Chapter 8

Algorithms for solving combinatorial optimization problems

For this part of the notes, we follow Refs. [Vikstål, 2018, Rodríguez-Laguna and Santalla, 2018, Albash and Lidar, 2018].

8.1 Combinatorial optimization problems

A combinatorial optimization problem seeks to find the best answer to a given problem from a vast collection of configurations. A typical example of a combinatorial optimization problem is the travelling salesperson problem, where a salesperson seeks to find the shortest travel distance between different locations, such that all locations are visited once. The naive method to solve this problem would be to make a list of all the different routes between locations that the salesperson could take and from that list find the best answer. This could work when the number of locations is small, but the naive method would fail if the number of locations grows large, since the number of possible routes increases exponentially with the number of locations. Thus the naive method is not efficient in practice, and we should therefore develop more clever optimisaton algorithms.

Typical examples of this kind of relevant combinatorial optimization problems are summarized in the list below.

- The traveling salesperson problem (TSP), as already mentioned. You are given a list of cities and the distances between them, and you have to provide the shortest route to visit them all.
- The knapsack problem. You are given the weights and values of a set of objects and you have to provide the most valuable subset of them to take with you, given a certain bound on the total weight.
- Sorting. Given N numbers, return them in non-ascending order.
- The integer factorization problem. You are given a big number M, and you have to provide two integer factors, p and q, such that $M = pq$.
- The satisfactibility problem (SAT). You are given a boolean expression of many variables $z_i \in \{0, 1\}$, for example, $P(z_1, z_2, z_3, z_4) = z_1 \vee \bar{z}_2 \wedge (z_3 \vee \bar{z}_4)$. Then, you are asked whether there is a valuation of those variables which will make the complete expression true. For example, in that case, making all $z_i = 1$ is a valid solution.

Notice that all these problems have a similar structure: you are given certain input data (the initial numbers, the list of the cities, the list of objects or the integer to factorize) and you are asked to provide a response. The first two problems are written as optimization problems, in which a certain target function should be minimized (or maximized). The sorting problem can be restated as an optimization problem: we can design a penalty function to be minimized, by counting the misplaced consecutive numbers. The factorization problem can also be expressed in that way: find p and q such that $E = (M - pq)^2$ becomes minimal, and zero if possible. SAT can also be regarded as an optimization problem in which the evaluation of the boolean formula should be maximized. Thus, all those problems can be seen as combinatorial optimization problems. This means that, in order to solve them by brute force, a finite number of possibilities must be checked. But this number of possibilities grows very fast with the number of variables or the size of the input. The number of configurations typically involved becomes easily gigantic. It is enough to consider 100 guests at a party with 100 chairs, to obtain $100! \sim 10^{157}$ possible configurations to chose in which assigning each guest to a chair. This is roughly the square of number of particles in the universe, estimated to be about 10^{80}. To calculate the correct configuration in this ocean of possible configurations is often hopeless for classical computers - even for our best and future best super computers. This is why it makes sense to ask the question: could a quantum computer help?

8.1.1 Hardness of combinatorial optimization problems and promises of quantum computers for solving them

Not all optimization problems are equally hard. Let us focus on the minimal possible time required to solve them as a function of the input size, i.e. to which time-complexity class they belong (see Sec.2.2 for a list of relevant complexity classes).

We recall that a problem is said to be NP-hard if an algorithm solving it can be translated into an algorithm for solving any NP-complete problem with a poly-time overhead. NP-hard problems are not necessarily decision problems, but for any optimization problem that is NP-hard, the corresponding decision problem is NP-complete. For example, the decision version of the traveling salesperson problem reads as follows: given a length L, decide whether the graph has a tour of at most L. Another example of a famous NP-hard problem is the Max-Cut problem.

All the decision problems associated to the optimisation problems in the list introduced in the previous subsection are NP: given a candidate solution, it can always be checked in polynomial time. But only one of them is known to be in P: the sorting problem, because it can always be solved in time $O(Nlog(N))$ which is less than $O(N^2)$. For the factorization problem we do not know whether it is in P or not. The other three belong to a special subset: they are NP-complete. This means that they belong to a special group with this property: if a polynomial algorithm to solve one of them is ever found, then we will have a polynomial algorithm to solve all NP problems. The generality and power of this result is known as Cook's theorem. How can the solution to an NP-complete problem be useful to solve all NP problems? By using a strategy called reduction. An instance of the sorting problem, for example, can be converted into an instance of SAT in polynomial time. Thus, the strategy to solve any NP-problem would be: (i) translate your instance to an instance of SAT, (ii) solve that instance of SAT, and (iii) translate back your solution. It is very relevant for physicists to know which combinatorial optimization problems are NP-complete for many reasons, and one of the most important is to avoid losing valuable time with a naive attempt to solve them in polynomial time. The complexity classes relative to the problems listed above are summarized in Fig.8.1.

Actually, we do not expect quantum computers either to solve efficiently problems that are NP-complete. That would mean that NP is contained in BQP, and we do not believe that this is the case (see e.g. https://www.scottaaronson.com/blog/?p=206). Yet, we expect that some problems that are in NP and not in P could be efficiently solvable by a quantum computer, i.e. they belong to BQP (such as Factoring).

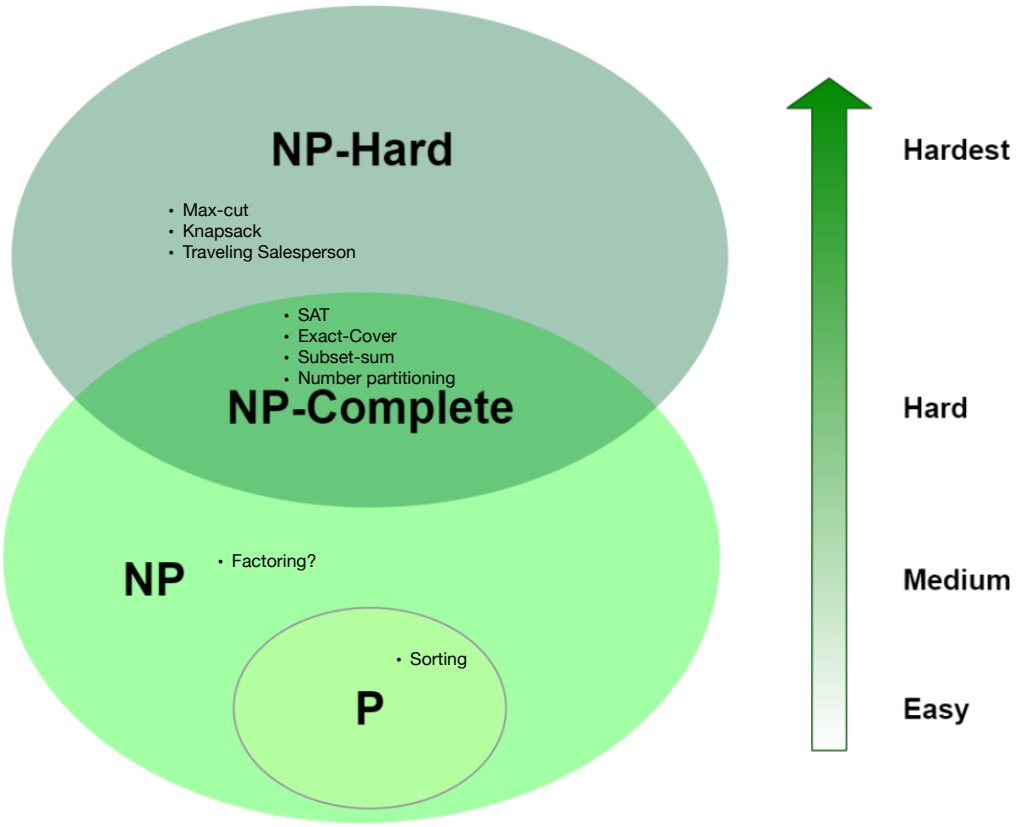


Figure 8.1: Complexity classes relevant for optimisation problems.

For these problems, quantum algorithms could provide an exponential advantage. Furthermore, for the NP-complete problems, maybe we can solve them on a quantum computer with a quadratic advantage with respect to the classical solution. This kind of quantum advantage is of the same kind of the Grover algorithm. The Grover search algorithm can be run on quantum computer, either within the circuit model or using adiabatic quantum computation [Albash and Lidar, 2018], providing the same quadratic advantage, i.e. $\mathcal{O}(\sqrt{N})$ vs $\mathcal{O}(N)$ running time.

Finally, *approximate* solutions of combinatorial optimization problems (even NP-complete ones) might be obtainable efficiently with the QAOA algorithm, that we will see in Sec. 8.4. Not much is known currently, about the time-complexity of finding approximate solutions to those problems, and how this compares to the corresponding approximate classical algorithms.

8.2 Combinatorial optimization and the Ising model

To solve a combinatorial optimization problem using a quantum algorithm, be it Quantum Annealing or the QAOA as we are going to see in the next sections, the quantum algorithm must be able to interpret the specific problem that we wish to solve. This can be done by encoding the optimization problem onto a quantum system. In this section we will see how a combinatorial optimization can be framed as a cost Hamiltonian in the Ising form.

Let $C : \{0,1\}^n \to \mathbb{R}$ be a cost function that encode a combinatorial optimization problem. There are in

total 2^n possible strings and the goal is to find the bit-string $z = z_1 \dots z_n$ that minimizes the cost function $C(z)$. Note that a minimization problem can be transformed into a maximization problem by a minus sign $C(z) \to -C(z)$. One of the most widely used models in physics, that also is used to represent optimization problems, is the Ising model. It is was developed in the 1920s as a way to understand phase transitions in magnetic materials.

The *Ising model* consists of N Ising spins on a lattice that can take the values $s_i = +1$ or $s_i = -1$, which corresponds to the spin-↑ and the spin-↓ direction. Here s_i denotes the Ising spin at site i on the lattice. The Ising spins are coupled together through long-range magnetic interactions, which can be either *ferromagnetic* or *antiferromagnetic*, corresponding to the spins being encouraged to be aligned or anti-aligned respectively. Moreover, an external magnetic field can be applied at each individual spin site which will give a different energy to the spin-↑ and spin-↓ directions. The energy configuration of the Ising model with N Ising spins is given by the Ising Hamiltonian

$$\mathcal{H}(s_1, \dots, s_N) = -\sum_{i,j=1}^{N} J_{ij} s_j s_i - \sum_{i=1}^{N} h_i s_i, \quad (s_i = \pm 1), \tag{8.1}$$

where J_{ij} is the coupling strength between the i^{th} spin and j^{th} spin and h_i is the magnetic field at the i^{th} spin site. Note that for a given set of couplings, magnetic fields and spin-configurations, the Ising Hamiltonian will just "*spit out*" a number that represents the energy of that particular spin-configuration. For a given set of couplings and magnetic fields there always exists at least one spin-configuration that minimizes the energy of the Ising Hamiltonian.

A quantum version of this model is obtained by simply replacing the spin-variables s_i with Pauli-z operators

$$\hat{H}_C \equiv \hat{H}(\hat{\sigma}_1^z, \dots, \hat{\sigma}_n^z) = -\sum_{1 \le i < j \le n} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z - \sum_{i=1}^{n} h_i \hat{\sigma}_i^z, \tag{8.2}$$

were $\hat{\sigma}_i^z$ refers to the Pauli-z matrix acting on the ith qubit. The spectral decomposition of the this Hamiltonian then encodes the different solutions in the computational basis

$$\hat{H}_C = \sum_{z \in \{0,1\}^n} C(z) \, |z\rangle\langle z| \, , \tag{8.3}$$

and such the eigenstate $|z\rangle$ with the lowest eigenvalue corresponds the optimal solution. The Hamiltonian of Eq. 8.2 is formally known as an Ising-Hamiltonian, after its inventor, but it can also be referred to as as *cost Hamiltonian* in the context of optimisation problems, because of its eigenvalues in the computational basis corresponds to the different costs of the cost function.

Also not that in the context of QAOA, we will refer to the Ising or cost Hamiltonian as Eq.(8.2), where however we will take the plus sign in front of both terms of the Hamiltonian.

8.2.1 Mapping combinatorial optimization problems to spin Hamiltonians

Many optimization problems, including all of Karp's 21 NP-complete problems, can be written in the form of Eq. (8.2) and hence solved on a quantum computer, by choosing appropriate values for J_{ij} and h_i [Lucas, 2014]. We now turn to a couple of specific examples, and we will disregard here the minus sign in front of the Ising Hamiltonian.

A special case of the knapsack problem: the subset sum problem

The *subset sum problem* is a famous combinatorial optimization problem that is known to be **NP**-complete, and it is a special case of the decision version of the knapsack problem, where the weights w_i are equal to

the values v_i for each object i, i.e. $w_i = v_i$. It can be formulated as a decision problem, as follows: Given an integer m (the total value) and a set of positive and negative integers $n = \{n_1, n_2, \ldots, n_N\}$ of length N, is there a subset of those integers that sums exactly to m?

Example: Consider the case when $m = 7$ and the set $n = \{-5, -3, 1, 4, 9\}$. In this particular example answer is "*yes*", and the subset is $\{-3, 1, 9\}$.

Example: Consider $m = 13$ and $n = \{-3, 2, 8, 4, 20\}$. This time the answer is "*no*".

This problem can be framed as an energy minimization problem. The energy function for the subset sum problem can be formulated as

$$\mathcal{E}(z_1, \ldots, z_N) = \left(\sum_{i=1}^{N} n_i z_i - m \right)^2, \quad z_i \in \{0, 1\},$$

where N corresponds to the size of the subset. Hence, if a configuration of z_i exists such that $\mathcal{E} = 0$, then the answer is "*yes*". Likewise if $(\mathcal{E}) > 0$ for all possible configurations of z_i, then the answer is "*no*". To map this energy function onto an Ising Hamiltonian we introduce the Ising spins $s_i = \pm 1$ instead of z_i as

$$z_i = \frac{1}{2}(1 + s_i), \tag{8.4}$$

such that $s_i = +1$ (spin-$\uparrow$) corresponds to $z_i = 1$, and $s_i = -1$ (spin-$\downarrow$) corresponds to $z_i = 0$. Then the Hamiltonian is written as

$$\begin{aligned}
\mathcal{H}(s_1, \ldots, s_N) &= \left(\sum_{i=1}^{N} n_i \frac{1}{2}(1 + s_i) - m \right)^2 \\
&= \left(\sum_{i=1}^{N} \frac{1}{2} n_i s_i + \frac{1}{2} \sum_{i=1}^{N} n_i - m \right)^2 \\
&= \frac{1}{4} \sum_{1 \le i,j \le N} n_i n_j s_i s_j + \sum_{i=1}^{N} \left(\frac{1}{2} \sum_{j=1}^{N} n_j - m \right) n_i s_i + \left(\frac{1}{2} \sum_{j=1}^{N} n_j - m \right)^2 .
\end{aligned}$$

We now introduce the coupling J_{ij} and the magnetic field h_i as

$$J_{ij} \equiv n_i n_j, \quad \text{and} \quad h_i = \left(\frac{1}{2} \sum_{j=1}^{N} n_j - m \right) n_i, \tag{8.5}$$

and finally obtain

$$\mathcal{H}(s_1, \ldots, s_N) = \frac{1}{4} \sum_{1 \le i,j \le N} J_{ij} s_i s_j + \sum_{i=1}^{N} h_i s_i + \left(\frac{1}{2} \sum_{j=1}^{N} n_j - m \right)^2 .$$

Notice that the last term is simply a constant. This Hamiltonian is an Ising Hamiltonian cf. Eq. (8.1). Furthermore, by observing that J_{ij} is symmetric and that the sum of the diagonal elements $i = j$ is equal to the trace of J_{ij}, we get

$$\begin{aligned}
\mathcal{H}(s_1, \ldots, s_N) &= \frac{1}{2} \sum_{1 \le i < j \le N} J_{ij} s_i s_j + \sum_{i=1}^{N} h_i s_i + \left(\frac{1}{2} \sum_{j=1}^{N} n_j - m \right)^2 + \frac{1}{4} \mathrm{Tr}\left[J_{ij} \right] \\
&= \sum_{1 \le i < j \le N} J_{ij} s_i s_j + \sum_{i=1}^{N} h_i s_i + \text{const},
\end{aligned} \tag{8.6}$$

where we have absorbed the $1/2$ into J_{ij} in the second step and made an implicit redefinition of the couplings $J_{ij} \equiv n_i n_j/2$. To solve the problem on a quantum computer, one can now quantize the spin variables $s_i \to \hat{\sigma}_i^z$.

Number partitioning problem

The *number partitioning problem* is another well known combinatorial optimization problem that is also **NP**-complete [Albash and Lidar, 2018]. The number partitioning problem can be defined as a decision problem, as follows: Given a set $\mathcal{S}$ of positive integers $\{n_1, n_2, \ldots, n_N\}$ of length N, can this set be partitioned into two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ such that the sum of the sets are equal?

Example: Consider the set $\{1, 2, 3, 4, 6, 10\}$, can this set be partitioned into two sets, such that the sum of both sets are equal? The answer is "*yes*" and the partitions are $\mathcal{S}_1 = \{1, 2, 4, 6\}$ and $\mathcal{S}_2 = \{3, 10\}$.

Example: Consider the set $\{1, 2, 3, 4, 6, 7\}$, can this set be partitioned into two sets, such that the sum of both sets are equal? This time the answer is "*no*", which you can try to convince yourself that it is.

The Ising Hamiltonian for the number partitioning problem can be straightforwardly written down as follows

$$\mathcal{H}(s_1, \ldots, s_N) = \left(\sum_{i=1}^{N} n_i s_i \right)^2, \quad (s_i = \pm 1). \tag{8.7}$$

It is clear that the answer to the number partitioning problem is "*yes*" if $\mathcal{H} = 0$, because then there exist a spin configuration where the sum of the n_i for the $+1$ spins is the equal to the sum of the n_i for the -1 spins [Lucas, 2014]. Likewise the answer is "*no*" if $(\mathcal{H}) > 0$ for all possible spin configurations. Expanding the square of Eq. (8.7) we get

$$\mathcal{H}(s_1, \ldots, s_N) = \sum_{1 \le i,j \le N} J_{ij} s_i s_j = 2 \sum_{1 \le i < j \le N} J_{ij} s_i s_j + \mathrm{Tr}\left[J_{ij}\right], \tag{8.8}$$

where we have introduced the couplings as

$$J_{ij} = n_i n_j.$$

It should be noted that the ground state of Ising Hamiltonian for the number partitioning problem is always at least two-fold degenerate. This has to do with the fact that changing s_i to $-s_i$ does not change $\mathcal{H}$. What is also notable about the number partitioning problem compared to the subset sum problem is that the number partitioning problem does not require any additional magnetic fields on each spin site.

Number partitioning is known as the "easiest hard problem" (Hayes, 2002) due to the existence of efficient approximation algorithms that apply in most (although of course not all) cases, e.g., a polynomial-time approximation algorithm known as the differencing method (Karmarkar and Karp, 1982). This concludes this section of examples.

8.3 Quantum annealing

There is no consensus in the literature, with respect to the therminology for quantum annealing. We adopt the following definition [Hauke et al., 2020]: *Quantum Annealing* (QA) is a heuristic quantum algorithm that is based on the *Adiabatic Quantum Computation* model seen in Chapter 7, and that aims at solving hard *combinatorial optimization problems*, by using an Ising Hamiltonian as the target Hamiltonian. In other words, the main idea of QA is to take advantage of adiabatic evolution expressed by Eq.(7.1) to go from a simple non-degenerate ground state of an initial Hamiltonian to the ground state of an Ising Hamiltonian that encodes the solution of the desired combinatorial optimization problem. It can hence be thought as the restriction of adiabatic quantum computation to optimization problems. As a consequence, a less general Hamiltonian is sufficient for addressing quantum annealing, as compared to adiabatic quantum computation

(see also Scott Pakin Quantum Annealing lecture at the Quantum Science summer school 2017, available at http://qs3.mit.edu/images/pdf/QS3-2017---Pakin-Lecture.pdf).

8.3.1 Solving optimization problems on a quantum annealer

To solve an Ising problem on a qubit based quantum annealer one defines a set of qubits $|q_1 q_2 \dots q_N\rangle$ to store the answer to the problem and interpret $q_i = 0$ to mean $s_i = +1$ (spin-$\uparrow$) and $q_i = 1$ to mean $s_i = -1$ (spin-$\downarrow$). Then one chooses a pair of noncommuting Hamiltonians $\hat{\mathcal{H}}_0$ and $\hat{\mathcal{H}}_1$, such that the ground state of $\hat{\mathcal{H}}_0$ is easy to prepare, and that the ground state of $\hat{\mathcal{H}}_1$ encodes the solution to the Ising problem. To achieve this the initial Hamiltonian is typically assumed to be:

$$\hat{H}_0 = -\sum_i \hat{\sigma}_i^x, \quad \hat{\sigma}^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{8.9}$$

where $(\hat{\sigma}_i^x = \hat{I} \otimes \dots \otimes \hat{\sigma}^x \otimes \dots \hat{I})$ with the Pauli $\hat{\sigma}^x$ matrix on the i:th place. The qubits or (spins) can then be regarded as pointing simultaneously in the spin-$\uparrow$ and spin-$\downarrow$ directions along the z-axis at the beginning. Indeed, it is easy to show using basic quantum mechanics that the ground state of $\hat{\mathcal{H}}_0$ for N-qubits is

$$|\psi(0)\rangle = |+\rangle^{\otimes n} = \left(\frac{1}{\sqrt{2}}\right)^N (|0\rangle + |1\rangle) \otimes \dots \otimes (|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle,$$

which corresponds to a superposition of all possible spin configurations. Constructing $\hat{H}_1$ can be done in a straightforward manner by replacing each s_i in Eq. (8.1) with a corresponding Pauli-z matrix, to yield Eq.(8.2), that we report here for convenience:

$$\hat{H}_1 = -\sum_{i,j=1}^N J_{ij} \hat{\sigma}_j^z \hat{\sigma}_i^z - \sum_{i=1}^N h_i \hat{\sigma}_i^z,$$

where $(\hat{\sigma}_i^z = \hat{I} \otimes \dots \otimes \hat{\sigma}^z \otimes \dots \hat{I})$ with $\hat{\sigma}^z$ on the i:th place. When the time-dependent AQC Hamiltonian smoothly interpolates between $\hat{H}_0$ and $\hat{H}_1$,

$$\hat{H}(t) = (1 - s(t))\hat{H}_0 + s(t)\hat{H}_1, \tag{8.10}$$

with $s(0) = 0$ and $s(\tau) = 1$, τ the total time of the algorithm, the qubits will gradually choose between 0 and 1 corresponding to spin-$\uparrow$ and spin-$\downarrow$, depending on which spin-configuration minimizes the energy of the Ising Hamiltonian [Rodríguez-Laguna and Santalla, 2018]. At the end $t = \tau$ the system will have evolved into $|\psi(\tau)\rangle = |z_1 z_2 \dots z_N\rangle$ and a readout of this state in the computational basis will reveal each bit value from which the corresponding values of the Ising spins can be obtained. If the annealing time was not slow enough, which is in practice occurring in any realistic situation, the state that is read-out will only encode the optimal solution to the problem with a certain success probability p, given by the overlap of the final state with the solution state. It is possible to estimate the success probability at each run by calculating the ratio of the number of runs where the optimal solution was found to the total number of runs.

In order to calculate the time-to-solution with 99% certainty, we can evaluated the success probability after repeating m times the annealing procedure and equate it to 0.99, i.e.

$$P_{succ}^m = 1 - (1 - p)^m = 0.99,$$

from which we can extract

$$m = \frac{\ln(1 - 0.99)}{\ln(1 - p)},$$

where we have used the change of basis of logarithms $\log_b x = \log_a x / \log_a b$, and where ln is the natural logarithm. Therefore we obtain at the end:

$$T_{99} = m\tau = \frac{\ln(1-0.99)}{\ln(1-p)}\tau,$$

A challenge in quantum annealing is that a full connectivity, which manifest in the interaction parameters J_{ij} being $\neq 0$ beyond nearest-neihgbours interactions, is often required in the problem Hamiltonian (for typical hard problems). This full connectivity might be difficult to achieve experimentally. In case the hardware has limited connectivity, embeddings allow one to map the fully connected problem onto a locally connected spin-system, at the price of an overhead in the total number of physical qubits to use. Examples of embedding are the Lechner, Hauke and Zoller (LHZ) scheme or the minor embedding (we will talk about this later).

8.3.2 Heuristic understanding of quantum annealing

The term "annealing" is due to the analogy with the procedure used by metallurgists in order to make perfect crystals: the metal is melted and allowed to reduce its temperature very slowly. When the temperature becomes low enough, the system is expected to be in its global energy minimum with very high probability. Yet, mistakes are bound to take place if the temperature is reduced too fast. Why does this procedure work? Thermal fluctuations allow the system to explore a huge number of configurations. Sometimes, a metastable state is found, but if temperature is still large enough, fluctuations will allow it to escape. The escape probability is always related to the energy barrier separating the metastable state from the deeper energy minima. So, annealing can be considered an analog computation.

Annealing (meaning this metallurgic version) works nicely when the metastable states are separated by low energy barriers. Some target functions have barriers that are tall but very thin. Others will have different peculiarities.

Is there any alternative analog computation suitable for problems with tall and thin barriers? Yes, there is one. Instead of escaping metastable states through thermal fluctuations, we may try to escape them through quantum tunneling, since the probability of such an event is known to decay with the product of the height and width of the energy barriers.

This is what quantum annealing does: in QA, we engineer a quantum system so that its energy is related to the target function that we want to minimize. Thus, its ground state will provide the solution to our problem. If we start out at high temperature and cool the system down, then we are simply performing an annealing analog computation. Alternatively, we can operate always at extremely low temperature, so that quantum effects are always important, but add an extra element to the system which forces strong quantum fluctuations. This extra element (the starting Hamiltonian) is then slowly reduced and, when it vanishes, the GS will give us the solution to our problem. In other terms: we make the system Hamiltonian evolve from a certain starting Hamiltonian, H_0, whose GS presents strong quantum fluctuations in the basis of the final Hamiltonian eigenstates, to our target Hamiltonian, H_1, whose GS is the solution to our problem.

8.3.3 QUBO optimisation

Many optimisation problems can be recast in terms of QUBOs, where the acronym stands for Quadratic Unconstrained Binary Optimisation, i.e. the function to be optimised is a quadratic function over binary variables without further constraints.

The standard format for a QUBO objective function to be minimized over binary variables $z \in \{0,1\}^n$

is the following:

$$q(z) = z^\top Q z = \sum_{j=1}^{n} Q_{jj} z_j + \sum_{\substack{j,k=1 \\ j<k}}^{n} Q_{jk} z_j z_k \tag{8.11}$$

with an upper-triangular quadratic matrix $Q \in \mathbb{R}^{n \times n}$.

Usually these transformations produce overhead, e.g. in terms of increasing the number of variables, the required connections between them or the value of the coefficients.

8.3.4 D-wave quantum annealer

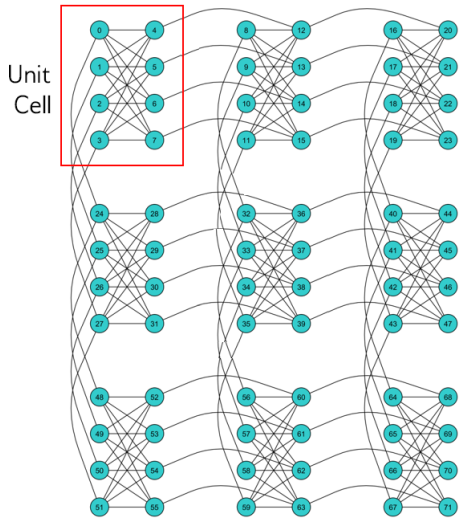


Figure 8.2: A 3x3 Chimera graph, denoted C3. Qubits are arranged in 9 unit cells. From D-Wave website.

D-Wave 200Q is the first commercially available quantum annealer, developed by the Canadian company D-Wave Systems, and it is a heuristic solver using quantum annealing for solving optimisation problems. Their hardware is based on rf-SQUIDs (Superconducting Quantum Interference Devices). In these devices, flux qubits are realized by means of long loops of superconducting wire interrupted by a set of Josephson junctions. A “supercurrent" of Cooper pairs of electrons, condensed to a superconducting condensate, flows through the wires; a large ensemble of these pairs behaves as a single quantum state with net positive or net negative flux, thereby defining the two qubits states. The chip must be kept extremely cold for the macroscopic circuit to behave like a two-level (qubit) system, and nominally runs at 15 mK.

In order to be casted onto D-Wave machines, optimisation problems can be formulated either in terms of Ising Hamiltonians, or in terms of QUBOs.

The hardware layout of the D-Wave 2000Q quantum annealer restricts the connections between binary variables to the so called Chimera graph 8.2. In order to make problems with higher connectivity amenable to the machine, an embedding must be employed, for instance minor embedding. This means coupling various physical qubits together into one logical qubit, representing one binary variable, with a strong ferromagnetic coupling J_F in order to ensure that all physical qubits have the same value after readout.

For how to program a D-Wave machine, see https://docs.dwavesys.com/docs/latest/doc_getting_started.html.

8.3.5 Summary of pros and cons for quantum annealing

As a summary of this quantum annealing section, and before turning to an example of a problem solved on a quantum annealer, we present a list of pros and cons for quantum annealing, taken from the notes of Scott Pakin Quantum Annealing lecture at the Quantum Science summer school 2017, http://qs3.mit.edu/images/pdf/QS3-2017---Pakin-Lecture.pdf.

The bad:

- Very difficult to analyze an algorithm's computational complexity
  - Need to know the gap between the ground state and first excited state, which can be costly to compute
  - In contrast, circuit-model algorithms tend to be more straightforward to analyze
- Unknown if quantum annealing can outperform classical computation
  - If gap always shrinks exponentially then no

The good:

- Constants do matter
  - If the gap is such that a correct answer is expected only once every million anneals, and an anneal takes 5 microseconds, that is still only 5 seconds to get a correct answer - may be good enough
  - On current systems, the gap scaling may be less of a problem than the number of available qubits
- We may be able to (classically) patch the output to get to the ground state
  - Hill climbing or other such approaches may help get quickly from a near-groundstate solution into the ground state
  - We may not even need the exact ground state
  - For many optimization problems, "good and fast" may be preferable to "perfect but slow"

8.3.6 Example of the solution of a practical problem on a quantum annealer: Flight-gate assignment

In this section, we present the results outlined in Ref. [Stollenwerk et al., 2019], where the solution of a concrete problem is addressed on a quantum annealer, namely flight gate assignment. The goal of flight gate assignment is reducing the total transit time for passengers in an airport, increasing passenger comfort and punctuality. This problem is related to the quadratic assignment problem, a fundamental problem in combinatorial optimization whose standard formulation is NP-hard. This problem can be brought into a QUBO format by standard transformations. Here, we investigate the solvability of the optimal flight gate assignment problem with a D-Wave 2000Q quantum annealer.

Formal Problem Definition

The typical passenger flow in an airport can usually be divided into three parts: After the airplane has arrived at the gate, one part of the passengers passes the baggage claim and leaves the airport. Other passengers stay in the airport to take connecting flights. These transit passengers can take up a significant fraction of the total passenger amount. The third part are passengers which enter the airport through the security checkpoint and leave with a flight. The parameters of the problem are summarized in table 8.1.

Assignment problems can easily be represented with binary variables indicating whether or not a resource is assigned to a certain facility. The variables form a matrix indexed over the resources and the facilities. The binary decision variables are $x \in \{0,1\}^{F \times G}$ with

$$z_{i\alpha} = \begin{cases} 1, & \text{if flight } i \in F \text{ is assigned to gate } \alpha \in G, \\ 0, & \text{otherwise.} \end{cases} \tag{8.12}$$

Like already stated, the passenger flow divides into three parts and so does the objective function: The partial sums of the arriving, the departing and the transfer passengers sum up to the total transfer time of all passengers. For the arrival part we get a contribution of the corresponding time t_α^{arr} for each of the n_i^{arr} passengers if flight i is assigned to gate α. Together with the departure part, which is obtained analogously, the linear terms of the objective are

$$T^{\text{arr/dep}}(z) = \sum_{i\alpha} n_i^{\text{arr/dep}} t_\alpha^{\text{arr/dep}} \, z_{i\alpha}. \tag{8.13}$$

The contribution of the transfer passengers is the reason for the hardness of the problem: Only if flight i is assigned to gate α and flight j to gate β the corresponding time is added. This results in the quadratic term

$$T^{\text{trans}}(z) = \sum_{i\alpha j\beta} n_{ij} t_{\alpha\beta} \, z_{i\alpha} z_{j\beta}. \tag{8.14}$$

The total objective function is

$$T(z) = T^{\text{arr}}(z) + T^{\text{dep}}(z) + T^{\text{trans}}(z). \tag{8.15}$$

Constraints

Not all binary encodings for z form valid solutions to the problem. There are several further restrictions which need to be added as constraints. In this model a flight corresponds to a single airplane arriving and

Table 8.1: Input data for a flight gate assignment instance

F	Set of flights ($i \in F$)
G	Set of gates ($\alpha \in G$)
n_i^{dep}	No. of passengers departing with flight i
n_i^{arr}	No. of passengers arriving with flight i
n_{ij}	No of transfer passengers between flights i and j
t_α^{arr}	Transfer time from gate α to baggage claim
t_α^{dep}	Transfer time from check-in to gate α
$t_{\alpha\beta}$	Transfer time from gate α to gate β
t_i^{in}	Arrival time of flight i
t_i^{out}	Departure time of flight i
t^{buf}	Buffer time between two flights at the same gate

departing at a single gate. It is obvious, that every flight can only be assigned to a single gate, therefore we have

$$\sum_{\alpha} z_{i\alpha} = 1 \qquad \forall i \in F. \tag{8.16}$$

Furthermore it is clear that no flight can be assigned to a gate which is already occupied by another flight at the same time. The resulting linear inequalities $z_{i\alpha} + z_{j\alpha} \leq 1$ are equivalent to the quadratic constraints

$$z_{i\alpha} \cdot z_{j\alpha} = 0 \qquad \forall (i,j) \in P \; \forall \alpha \in G, \tag{8.17}$$

where P is the subset of forbidden flight pairs with overlapping time slots, that can be aggregated in

$$P = \left\{ (i,j) \in F^2 : t_i^{\text{in}} < t_j^{\text{in}} < t_i^{\text{out}} + t^{\text{buf}} \right\}. \tag{8.18}$$

Mapping to QUBO and Penalty Terms

While the presented objective function already follows the QUBO format Eq.(8.11), the constraints need to be reduced which is shown in this section. The standard way to reduce constraints is to introduce terms penalizing those variable choices that violate the constraints. Just in these cases a certain positive value is added to the objective function to favor valid configurations while minimizing. The quadratic terms

$$C^{\text{one}}(z) = \sum_{i} \left(\sum_{\alpha} z_{i\alpha} - 1 \right)^2 \tag{8.19}$$

and

$$C^{\text{not}}(z) = \sum_{\alpha} \sum_{(i,j) \in P} z_{i\alpha} z_{j\alpha} \tag{8.20}$$

fulfill

$$C^{\text{one/not}} \begin{cases} > 0, & \text{if constraint is violated,} \\ = 0, & \text{if constraint is fulfilled,} \end{cases} \tag{8.21}$$

and therefore are suitable penalty terms which can be combined with the objective function. Since the benefit in the objective function in case of an invalid variable choice should not exceed the penalty, two parameters $\lambda_{\text{one}}, \lambda_{\text{not}} \in \mathbb{R}_+$ need to be introduced:

$$q(z) = T(z) + \lambda^{\text{one}} C^{\text{one}}(z) + \lambda^{\text{not}} C^{\text{not}}(z). \tag{8.22}$$

In theory these parameters could be set to infinity, but in practice this is not possible and they have to be chosen carefully.

The parameters λ^{one} and λ^{not} need to be large enough to ensure that a solution always fullfills the constraints. However due to precision restrictions of the D-Wave machine it is favorable to choose them as small as possible. Two different possibilities to obtain suitable values are studied in Ref. [Stollenwerk et al., 2019]: a worst case analysis for each single constraint, and a bisection algorithm which iteratively checks penalty weights against the solution validity. We leave the technical details of this discussion to [Stollenwerk et al., 2019].

Note that this final cost function could be expressed in terms of an Ising Hamiltonian, by using the mapping in Eq.(8.4) and by quantizing the spin variables. However, QUBO is the native way of encoding cost-functions into D-Wave machines, therefore the authors leave out this step.

Since the constraint Eq. (8.16) introduces $|F|$ complete graphs of size $|G|$, and given the connectivity of chimera graphs, which requires in the use of minor embedding, this results at most in a quadratic increase in the number of physical qubits with the number of logical qubits, which is $|F| \cdot |G|$. With this, the authors were able to embed instances up to 84 binary variables (extracted by real data, but simplified considerably), which requires roughly 1600 qubits. The instances are then solved on D-Wave, and the time-to-solution is then evaluated as a function of the size of the instance.

Time to solution

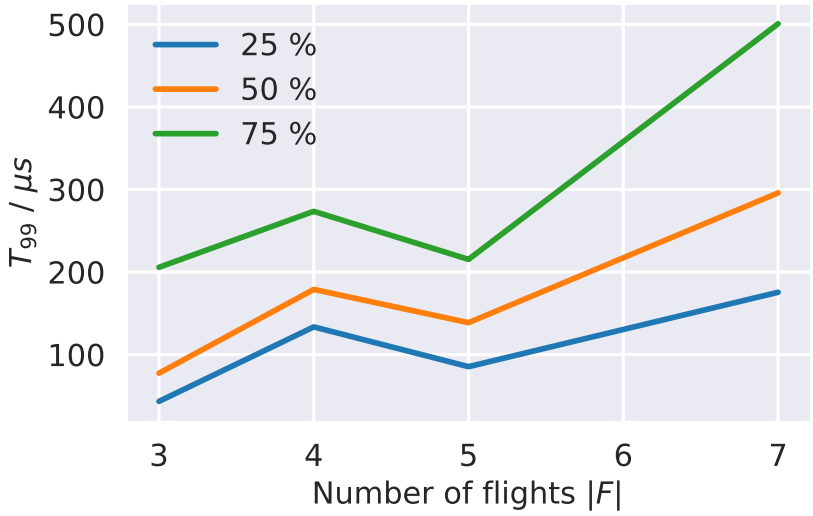


Figure 8.3: Top: Time to solution against the number of flights for the instances from $\mathcal{I}_{\mathrm{BP}}$. The different colors represent the 25th, 50th and 75th percentiles.

The annealing solutions were obtained using 1000 annealing runs, and majority voting as a un-embedding strategy for broken chains of logical qubits. The time to solution is then evaluated using Eq.(8.3.1), where τ is the annealing time, which we fixed to $20\mu s$.

Figure 8.3 shows the time-to-solution in dependence of the number of flights. There is an increase in the time-to-solution with the number of flights, and therefore the problem size.

A conclusive assessment of the scaling behavior is not possible at the moment, because due to the high connectivity of the problem there is a large embedding overhead, and as a result the size of the amenable problems is very small. Future generations of quantum annealers with more qubits and higher connectivity are needed to investigate larger problems.

8.4 Quantum Approximate Optimization Algorithm (QAOA)

8.4.1 Introduction to QAOA: From the quantum adiabatic algorithm to QAOA

This Section is taken from the licentiate thesis of Pontus Vikstål (Chalmers, 2021).

The QAOA [Farhi et al., 2014] is inspired by the adiabatic quantum computing (AQC), and in particular its application to optimisation problems in the context of quantum annealing, that you have seen in Chap.7 and 8.3.1. We report here for convenience the Hamiltonian of Eq.(8.10):

$$\hat{H}(t) = (1 - s(t))\hat{H}_M + s(t)\hat{H}_C. \tag{8.23}$$

Here $s(0) = 0$ and $s(T) = 1$, T is the total time of the algorithm, $\hat{H}_M$ is the initial Hamiltonian, whose maximally excited state is easy to prepare, and $\hat{H}_C$ is the cost Hamiltonian maximally excited state encodes the solution to an optimization problem.

The QAOA is based on the observation that the easiest way to practically implement the quantum annealing Hamiltonian evolution expressed by Eq.(8.23) is to Trotterize it, i.e. to decompose it in small time steps:

$$\hat{U}(T) \equiv \mathcal{T} \exp\left[-\mathrm{i}\int_0^T \hat{H}(t)\,\mathrm{d}t\right] \approx \prod_{k=1}^{p} \exp\Big[-\mathrm{i}\hat{H}(k\Delta t)\Delta t\Big]. \tag{8.24}$$

Here $\hat{U}(T)$ is the evolution operator from 0 to T, $\mathcal{T}$ is the time-ordering operator, and p is a large integer so that $\Delta t = T/p$ is a small time segment. Next, for two non-commuting operators A and B and sufficiently small Δt, one can use the Trotter formula:

$$\mathrm{e}^{\mathrm{i}(A+B)\Delta t} = \mathrm{e}^{\mathrm{i}A\Delta t}\,\mathrm{e}^{\mathrm{i}B\Delta t} + \mathcal{O}(\Delta t), \tag{8.25}$$

and apply it to the discretized time evolution operator Eq. (8.24)

$$\hat{U}(T) \approx \prod_{k=1}^{p} \exp\Big[-\mathrm{i}(1 - s(k\Delta t))\hat{H}_M\Delta t\Big] \exp\Big[-\mathrm{i}s(k\Delta t)\hat{H}_C\Delta t\Big]. \tag{8.26}$$

Thus it is possible to approximate quantum annealing by applying the cost and mixing Hamiltonian in an alternating sequence.

The key idea underlying QAOA is to truncate this product to an arbitrary positive integer and redefine the time dependence in each exponent $(1 - s(k\Delta t))\Delta t \to \beta_k$ and $s(k\Delta t)\Delta t \to \gamma_k$. In this way, and as a crucial difference with quantum annealing, *the fixed time segments become instead variational parameters to be optimized*. Finally letting the product act on the initial state of quantum annealing, the plus state, one obtains the variational state

$$|\vec{\gamma}, \vec{\beta}\rangle \equiv \prod_{k=1}^{p} \mathrm{e}^{-\mathrm{i}\beta_k \hat{H}_M}\,\mathrm{e}^{-\mathrm{i}\gamma_k \hat{H}_C}\,|+\rangle^{\otimes n} = \sum_z d_z^{(\vec{\gamma},\vec{\beta})}|z\rangle, \tag{8.27}$$

where $\vec{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_p)$ and $\vec{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$, and where in the last step we have just made explicit that the variational state is s given superposition of z-eigenstates. If the eigenvalues of the cost Hamiltonian are all integers then γ is 2π-periodic, and can be restricted to lie between $\gamma_k \in [0, 2\pi]$ and the mixer is π-periodic $\beta_k \in [0, \pi]$. However, the task of choosing the time-dependence of the angular variational parameters $(\vec{\gamma}, \vec{\beta})$, remains. The possibility of optimising over variational parameters makes it irrelevant whether the initial state is a ground state or the highest excited state. In the context of QAOA, we might hence re-define the mixing Hamiltonian without the minus sign, i.e. we set $\hat{H}_M$ to be

$$\hat{H}_M = \sum_i \hat{\sigma}_i^x. \tag{8.28}$$

Also, different authors use QAOA for minimisation or maximisation of a given cost-function. What you should pay attention to is the following: once the problem is given, if your cost-function $\hat{H}_C$ encodes the solution in its ground state, then you should proceed to minimisation of the cost function.

Let $E_p(\vec{\gamma}, \vec{\beta})$ be the expectation value of $\hat{H}_C$ in the variational state of Eq.(8.27).

$$E_p(\vec{\gamma}, \vec{\beta}) \equiv \langle \vec{\gamma}, \vec{\beta} | \hat{H}_C | \vec{\gamma}, \vec{\beta} \rangle = \sum_z \text{prob}_z^{(\vec{\gamma}, \vec{\beta})} C(z), \tag{8.29}$$

where, by obtaining the last term in Eq.(8.27), we see that $\text{prob}_z^{(\vec{\gamma}, \vec{\beta})} = |d_z^{(\vec{\gamma}, \vec{\beta})}|^2$ and $C(z) = \langle z | H_C | z \rangle$. By finding good angles $\vec{\gamma}$ and $\vec{\beta}$ that minimize the expectation value above, the probability of finding the qubits in their lowest energy configuration when measuring is increased, because the candidate variational states becomes closer to the ground state of the cost Hamiltonian. Therefore the angles are chosen such that the expectation value is minimized

$$(\vec{\gamma}^*, \vec{\beta}^*) = \arg\min_{\vec{\gamma}, \vec{\beta}} E_p(\vec{\gamma}, \vec{\beta}). \tag{8.30}$$

In general, this requires the quantum computer to query a classical optimizer, to tell the quantum computer how it should update the variational state by slightly changing the angles in order to minimize the expectation value, see Fig. 8.4. This has to be repeated until some convergence criteria is met or if an optimal or a good enough solution is found. In other words, QAOA is converting the search in a space of a combinatorial number of discrete configurations in a search for $2p$ optimal angles along a continuous energy landscape.

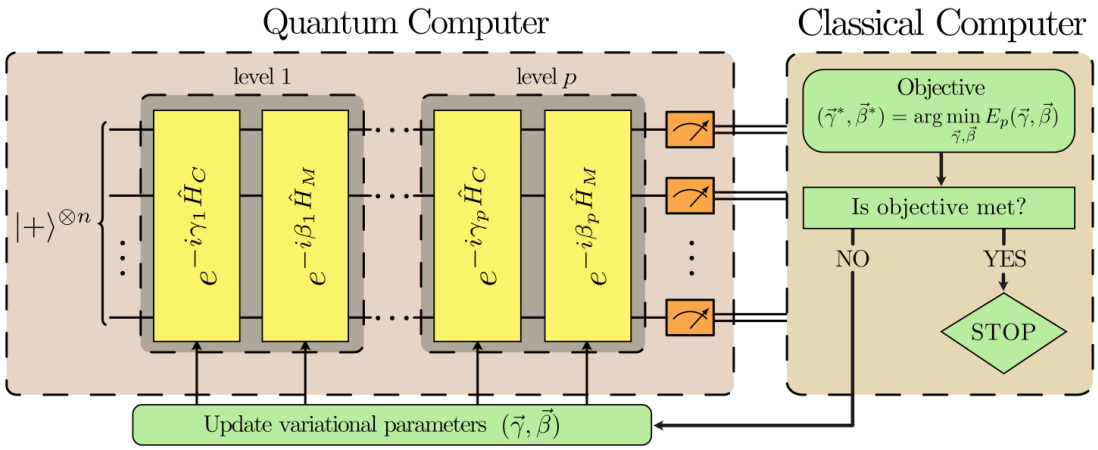


Figure 8.4: Schematic representation of the QAOA. The quantum processor prepares the variational state, depending on variational parameters. The variational parameters $(\vec{\gamma}, \vec{\beta})$ are optimized in a closed loop using a classical optimizer. From Ref. [Vikstål et al., 2020].

We end this section by giving a summary of the QAOA:

1. Pick a p and start with an initial set of $2p$ angles $(\vec{\gamma}, \vec{\beta})$.
2. Construct the state $|\vec{\gamma}, \vec{\beta}\rangle$ using a quantum computer and measure this state in the computational basis. The output is a string z with a probability given by the distribution of states $|z\rangle$.
3. Calculate $C(z) = \langle z | H_C | z \rangle$ using a classical computer. This step is classically efficient.
4. Repeat step 2-3, m times. Record the best observed string z_{best}, and the sample mean $1/m \sum_{i=1}^{m} C(z_i)$, where z_i is the ith measurement outcome. Note that when $m \to \infty$ the sample mean approaches the

expectation value Eq. (8.29) by the law of large numbers, as it is clear in particular from the rightermost hand-side term.

5. If the optimal or a "good enough" solution is found, output $C(z_{\text{best}})$ together with the string z_{best}. Else, query a classical optimizer that updates the angles $(\vec{\gamma}, \vec{\beta})$ based on the minimization of the expectation value and repeat from step 2.

Since QAOA is an algorithm that provides approximate solutions, a relevant metric in order to compare its performance with respect to efficient classical algorithms also producing approximate solutions is the approximation ratio. The approximation ratio is $C(z)$, where z is the output of the quantum algorithm, divided by the maximum of C.

Another relevant quantity in the context of QAOA is its success probability. Analogously as for quantum annealing, it is defined as the square of the overlap between the variational states obtained for the optimal angles, with the actual solution of the problem.

8.4.2 The Quantum Approximate Optimization Algorithm for solving Max-cut

The following Sections are taken from the paper of Fahri and Guttman [Farhi et al., 2014]. In this section, we follow hence the notations of the original reference, and we consider a maximisation of the cost function.

Fixed p Algorithm

We now explain how for fixed p we can do classical preprocessing and determine the angles $\vec{\gamma}$ and $\vec{\beta}$ that maximize $E_p(\vec{\gamma}, \vec{\beta})$. This approach will work more generally but we illustrate it for a specific problem, MaxCut for graphs with bounded degree. The input is a graph with n vertices and an edge set $\{\langle jk\rangle\}$ of size m. The goal is to find a string z that makes

$$\hat{H}_C = \sum_{\langle jk\rangle} \hat{C}_{\langle jk\rangle}, \tag{8.31}$$

where

$$C_{\langle jk\rangle} = \frac{1}{2}\left(-\sigma_j^z \sigma_k^z + 1\right), \tag{8.32}$$

as large as possible. Now

$$E_p(\vec{\gamma}, \vec{\beta}) = \sum_{\langle jk\rangle} \langle s|\, \mathrm{e}^{\mathrm{i}\gamma_1 \hat{H}_C} \cdots \mathrm{e}^{\mathrm{i}\beta_p \hat{H}_M}\; C_{\langle jk\rangle}\, \mathrm{e}^{-\mathrm{i}\beta_p \hat{H}_M} \cdots \mathrm{e}^{-\mathrm{i}\gamma_1 \hat{H}_C}\, |s\rangle\,. \tag{8.33}$$

Consider the operator associated with edge $\langle jk\rangle$

$$\mathrm{e}^{\mathrm{i}\gamma_1 \hat{H}_C} \cdots \mathrm{e}^{\mathrm{i}\beta_p \hat{H}_M}\; C_{\langle jk\rangle}\, \mathrm{e}^{-\mathrm{i}\beta_p \hat{H}_M} \cdots \mathrm{e}^{-\mathrm{i}\gamma_1 \hat{H}_C}\,. \tag{8.34}$$

This operator only involves qubits j and k and those qubits whose distance on the graph from j or k is less than or equal to p. To see this consider $p = 1$ where the previous expression is

$$\mathrm{e}^{\mathrm{i}\gamma_1 \hat{H}_C}\, \mathrm{e}^{\mathrm{i}\beta_1 \hat{H}_M}\; C_{\langle jk\rangle}\, \mathrm{e}^{-\mathrm{i}\beta_1 \hat{H}_M}\, \mathrm{e}^{-\mathrm{i}\gamma_1 \hat{H}_C}\,. \tag{8.35}$$

The factors in the operator $\mathrm{e}^{-\mathrm{i}\beta_1 \hat{H}_M}$ which do not involve qubits j or k commute through $C_{\langle jk\rangle}$ and we get

$$\mathrm{e}^{\mathrm{i}\gamma_1 \hat{H}_C}\; e^{i\beta_1(\sigma_j^x + \sigma_k^x)} C_{\langle jk\rangle} e^{-i\beta_1(\sigma_j^x + \sigma_k^x)}\; \mathrm{e}^{-\mathrm{i}\gamma_1 \hat{H}_C}\,. \tag{8.36}$$

Any factors in the operator $\mathrm{e}^{\mathrm{i}\gamma_1 \hat{H}_C}$ which do not involve qubits j or k will commute through and cancel out. So the operator in equation Eq. (8.36) only involves the edge $\langle jk\rangle$ and edges adjacent to $\langle jk\rangle$, and qubits on

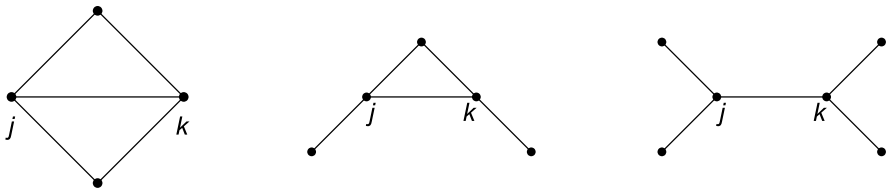
those edges. For any p we see that the operator in Eq. (8.34) only involves edges at most p steps away from $\langle jk \rangle$ and qubits on those edges.

Return to equation Eq. (8.33) and note that the state $|s\rangle$ is the product of σ^x eigenstates

$$|s\rangle = |+\rangle_1 |+\rangle_2 \dots |+\rangle_n \tag{8.37}$$

so each term in equation Eq. (8.33) depends only on the subgraph involving qubits j and k and those at a distance no more than p away. These subgraphs each contain a number of qubits that is independent of n (because the degree is bounded) and this allows us to evaluate E_p in terms of quantum subsystems whose sizes are independent of n.

As an illustration consider MaxCut restricted to input graphs of fixed degree 3. For $p = 1$, there are only these possible subgraphs for the edge $\langle jk \rangle$:



(8.38)

For any subgraph G define the operator C_G which is C restricted to G,

$$C_G = \sum_{\langle \ell \ell' \rangle \varepsilon G} C_{\langle \ell \ell' \rangle}, \tag{8.39}$$

and the associated operator

$$U(C_G, \gamma) = e^{-i\gamma C_G}. \tag{8.40}$$

Also define

$$B_G = \sum_{j \varepsilon G} \sigma_j^x \tag{8.41}$$

and

$$U(B_G, \beta) = e^{-i\beta B_G}. \tag{8.42}$$

Let the state $|s, G\rangle$ be

$$|s, G\rangle = \prod_{\ell \varepsilon G} |+\rangle_\ell .$$

Return to equation Eq. (8.33). Each edge $\langle j, k \rangle$ in the sum is associated with a subgraph $g(j, k)$ and makes a contribution to E_p of

$$\langle s, g(j,k)| U^\dagger \left(C_{g(j,k)}, \gamma_p\right) \cdots U^\dagger (B_{g(j,k)}, \beta_1) C_{\langle jk \rangle} U \left(B_{g(j,k)}, \beta_1\right) \cdots U \left(C_{g(j,k)}, \gamma_p\right) |s, g(j,k)\rangle \tag{8.43}$$

The sum in Eq. (8.33) is over all edges, but if two edges $\langle jk \rangle$ and $\langle j'k' \rangle$ give rise to isomorphic subgraphs, then the corresponding functions of $(\vec{\gamma}, \vec{\beta})$ are the same. Therefore we can view the sum in Eq. (8.33) as a sum over subgraph types. Define

$$f_g\left(\vec{\gamma}, \vec{\beta}\right) = \langle s, g(j,k)| U^\dagger (C_{g(j,k)}, \gamma_1) \cdots U^\dagger (B_{g(j,k)}, \beta_p) C_{\langle jk \rangle} U (B_{g(j,x)} \beta_p) \cdots U(C_{g(j,k)}, \gamma_1) |s, g(j,k)\rangle , \tag{8.44}$$

where $g(j,k)$ is a subgraph of type g. E_p is then

$$E_p\left(\vec{\gamma},\vec{\beta}\right) = \sum_g w_g\, f_g(\vec{\gamma},\vec{\beta}) \tag{8.45}$$

where w_g is the number of occurrences of the subgraph g in the original edge sum. The functions f_g do not depend on the number of decision variables n, neither on the number of constraints or clauses m. The only dependence on those quantities comes through the weights w_g and these are just read off the original graph. Note that the expectation in Eq. (8.44) only involves the qubits in subgraph type g. The maximum number of qubits that can appear in Eq. (8.43) comes when the subgraph is a tree. For a graph with maximum degree v, the numbers of qubits in this tree is

$$q_{\text{tree}} = 2\left[\frac{(v-1)^{p+1}-1}{(v-1)-1}\right], \tag{8.46}$$

(or $2p+2$ if $v=2$), which is n and m independent. For each p there are only finitely many subgraph types.

Using Eq. (8.44), $E_p(\vec{\gamma},\vec{\beta})$ in Eq. (8.45) can be evaluated on a classical computer whose resources are not growing with n. Each f_g involves operators and states in a Hilbert space whose dimension is at most $2^{q_{\text{tree}}}$. Admittedly for large p this may be beyond current classical technology, but the resource requirements do not grow with n.

To run the quantum algorithm we first find the $(\vec{\gamma},\vec{\beta})$ that maximize E_p. The only dependence on n and m is in the weights w_g and these are easily evaluated. Given the best $(\vec{\gamma},\vec{\beta})$ we turn to the quantum computer and produce the state $|\vec{\gamma},\vec{\beta}\rangle$ given in equation Eq. (8.27). We then measure in the computational basis and get a string z and evaluate $C(z) = \langle z|\hat{H}_C|z\rangle$. Repeating gives a sample of values of $C(z)$ between 0 and $+m$ whose mean is $E_p(\vec{\gamma},\vec{\beta})$. An outcome of at least $E_p(\vec{\gamma},\vec{\beta})-1$ will be obtained with probability $1-1/m$ with order $m\log m$ repetitions.

The Ring of Disagrees

We now analyze the performance of the quantum algorithm for MaxCut on 2-regular graphs. Regular of degree 2 (and connected) means that the graph is a ring. The objective operator is again given by equation Eq. (8.31) and its maximum is n or $n-1$ depending on whether n is even or odd. We will analyze the algorithm for all p.

For any p (less than $n/2$), for each edge in the ring, the subgraph of vertices within p of the edge is a segment of $2p+2$ connected vertices with the given edge in the middle. So for each p there is only one type of subgraph, a line segment of $2p+2$ qubits and the weight for this subgraph type is n. We numerically maximize the function given in Eq. (8.44) and we find that for $p = 1,2,3,4,5$ and 6 the maxima are $3/4$, $5/6$, $7/8$, $9/10$, $11/12$, and $13/14$ to 13 decimal places from which we conclude that $E_p = n(2p+1)/(2p+2)$ for all p. So the quantum algorithm will find a cut of size $n(2p+1)/(2p+2)-1$ or bigger. Since the best cut is n, we see that our quantum algorithm can produce an approximation ratio that can be made arbitrarily close to 1 by making p large enough, independent of n. For each p the circuit depth can be made $3p$ by breaking the edge sum in C into two sums over $\langle j, j+1\rangle$ with j even and j odd. So this algorithm has a circuit depth independent of n.

Remarks on the performance of QAOA

It is still an open area of research to establish whether QAOA with a fixed p allows for a better performance in solving approximatively NP-complete problems, with respect to classical algorithms. It is also an open question how QAOA performs with respect to Quantum Annealing. An interesting comparative study for Max-Cut on unweighted graphs and 2-SAT can be found in Ref. [Willsch et al., 2020].

Let us briefly discuss the performance of QAOA on MaxCut on (connected) 3-regular graphs. In Ref. [Farhi et al., 2014], they show that for $p = 1$, the worst case approximation ratio that the quantum algorithm produces is 0.6924. Hence this $p = 1$ result on 3-regular graphs is not as good as known classical algorithms, i.e. 0.9326 [Halperin et al., 2004]. It is possible to analyze the performance of the QAOA for $p = 2$ on 3-regular graphs. However it is more complicated than the $p = 1$ case. A numerical optimisation yields 0.7559. Recently, Wurtz and Love speculated that there is no quantum advantage for QAOA for solving Max-cut on 3-regular graphs for $p < 6$ [Wurtz and Love, 2020].

8.4.3 Other interesting remarks and extensions of QAOA

- Originally (see version 1 of Ref. [Farhi et al., 2014], also available on arXiv) the original authors considered the problem Max E3LIN2, where they showed that QAOA achieves a better approximation ratio than, at that time, the best classical approximation algorithm. The computer science community engaged the challenge and soon came up with a better classical algorithm [Barak et al., 2015].

- It has been shown that QAOA is universal, meaning that for a problem of size n, and a choice of $\hat{H}_C$ and $\hat{H}_M$, QAOA can approximate any unitary U of dimension $2^n \times 2^n$ to arbitrary precision when using a sufficiently large iteration p, which in general depends on n [Farhi et al., 2017, Lloyd, 2018, Morales et al., 2019].

- Farhi et al. showed that level $p = 1$ QAOA is computationally hard to simulate on a classical computer without collapsing the polynomial hierarchy to the third level [Farhi and Harrow, 2019]. However, this does not imply that QAOA for $p = 1$ is able to solve problems more efficiently than classically.

- In the absence of a fully connected set-up, one can split the role of the cost-function used for the classical optimization within QAOA and the Hamiltonian implementing the evolution, and still retain non-trivial approximation ratios, e.g. for Max-Cut [Farhi et al., 2017]. Furthermore, one can explore the advantage stemming from rotating the qubits with different angles when acting with the mixer Hamiltonian, and more generally introducing more free parameters.

- Brandao et al. [Brandao et al., 2018] demonstrated that if the problem instances come from a reasonable distribution, then the expectation value of the cost function concentrates. This suggests that it is possible to train a classical optimizer to find good angles on small instances and reuse those angles on larger instances, as long as they come from the same distribution.

- Furthermore, it was shown that the QAOA is able to realize Grover's search algorithm [Jiang et al., 2017, Niu et al., 2019].

- In 2019, Hadfield et al. put forward the quantum operator alternating ansatz, which generalizes the original QAOA ansatz to allow for more general types of Hamiltonians and initial states [Hadfield et al., 2019].

- In general, QAOA is nowadays an active field of research, and researchers are fervidly trying to characterise, understand and derive bounds on its performance.

8.4.4 More on the relation between QAOA and quantum annealing

In this section, we give a mapping between a Hamiltonian describing a quantum annealing scheme and the QAOA for a given number of steps, taken from Ref. [Willsch et al., 2020]. The annealing Hamiltonian reads

$$H(s) = A(s)(-H_0) + B(s)H_C, \quad s = t/t_a \in [0, 1], \tag{8.47}$$

where (we have to add an additional minus sign to H_0 such that the state $|+\rangle^{\otimes N}$ we start from is the ground state of $H(s)$ and the convention still conforms with the formulation of the QAOA)

$$H_0 = \sum_i \sigma_i^x, \tag{8.48}$$

$$H_C = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z. \tag{8.49}$$

We discretize the time-evolution operator of the annealing process into N time steps of size $\tau = t_a/N$. Approximating each time step to second order in τ yields

$$\begin{aligned} U = & e^{+i\tau A(s_N) H_0/2} e^{-i\tau B(s_N) H_C} \\ & \times e^{+i\tau (A(s_N)+A(s_{N-1})) H_0/2} \dots \\ & \times e^{-i\tau B(s_2) H_C} e^{+i\tau (A(s_2)+A(s_1)) H_0/2} \\ & \times e^{-i\tau B(s_1) H_C} e^{+i\tau A(s_1) H_0/2}, \end{aligned} \tag{8.50}$$

where $s_n = (n - 1/2)/N$, and $n = 1, \dots, N$.

To map Eq. (8.50) to the QAOA evolution

$$V = e^{-i\beta_p H_0} e^{-i\gamma_p H_C} \cdots e^{-i\beta_1 H_0} e^{-i\gamma_1 H_C}, \tag{8.51}$$

we can neglect $e^{+i\tau A(s_1) H_0/2}$ because its action on $|+\rangle^{\otimes N}$ yields only a global phase factor and we can choose

$$\gamma_n = \tau B(s_n), \qquad n = 1, \dots, N \tag{8.52}$$

$$\beta_n = -\tau \left(A(s_{n+1}) + A(s_n) \right)/2, \qquad n = 1, \dots, N-1 \tag{8.53}$$

$$\beta_N = -\tau A(s_N)/2. \tag{8.54}$$

So N time steps for the second-order-accurate annealing scheme correspond to $p = N$ steps for the QAOA.

As an example, we take

$$A(s) = 1 - s, \qquad B(s) = s. \tag{8.55}$$

Using Eqs. (8.52) - (8.54), we obtain

$$\gamma_n = \frac{\tau(n - 1/2)}{N} \tag{8.56}$$

$$\beta_n = -\tau \left(1 - \frac{n}{N} \right) \tag{8.57}$$

$$\beta_N = -\frac{\tau}{4N}. \tag{8.58}$$

Since the time evolution of quantum annealing is necessarily finite, with quantum annealing, too, there is an associated success probability, i.e. the probability of being in the actual desired ground state at the end of the evolution. In general, how the QAOA and quantum annealing performances - measured by the respective success probabilities - compare depends on the problem instance itself. Some detailed comparison are presented in Ref. [Willsch et al., 2020] for weighted Max-Cut and 2-SAT.

Chapter 9

The variational quantum eigensolver

In this chapter, we discuss the variational quantum eigensolver (VQE). The VQE is a heuristic approach to solving various problems with a combination of quantum and classical computation. As we will see later, the QAOA of the preceding chapter can be considered a special case of the VQE.

The content of this chapter is mostly based on the review in Ref. [Moll et al., 2018]. We first outline how the VQE works and then discuss details of some of the steps in the algorithm.

9.1 Outline of the algorithm

The VQE is designed to solve problems that can be cast in the form of finding the ground-state energy E_{GS} of a Hamiltonian H. The ground-state energy is the lowest eigenvalue of the Hamiltonian,

$$H \left|\Psi_{\mathrm{GS}}\right\rangle = E_{\mathrm{GS}} \left|\Psi_{\mathrm{GS}}\right\rangle . \tag{9.1}$$

How hard is this problem in general? If the Hamiltonian is k-local, i.e., if terms in H act on at most k qubits, the problem is know to be QMA-complete for $k \geq 2$. The general problem would thus be hard even for an ideal quantum computer. However, it is believed that physical systems have Hamiltonians that do not correspond to hard instances of this problem, and a heuristic quantum algorithm could still outperform a classical one.

A general Hamiltonian for N qubits can be written

$$H = \sum_{\alpha} h_\alpha P_\alpha = \sum_{\alpha} h_\alpha \bigotimes_{j=1}^{N} \sigma_{\alpha_j}^{(j)}, \tag{9.2}$$

where the h_α are coefficients and the P_α are called Pauli strings. The latter are products of single-qubit Pauli matrices (including the identity matrix).

The steps of the VQE algorithm are the following (see also Fig. 9.1):

0. Map the problem that you wish to solve to finding the ground-state energy of a Hamiltonian on the form in Eq. (9.2).
1. Prepare a trial state $|\Psi(\vartheta)\rangle$ set by a collection of parameters ϑ.
2. Measure expectation values of the Pauli strings in the Hamiltonian, i.e., measure $\mathrm{E}[\Psi(\vartheta)]P_\alpha\Psi(\vartheta)$.
3. Calculate the energy E corresponding to the trial state, $E = \sum_\alpha h_\alpha \mathrm{E}[\Psi(\vartheta)]P_\alpha\Psi(\vartheta)$, by summing up the results of the measurements in the preceding step.

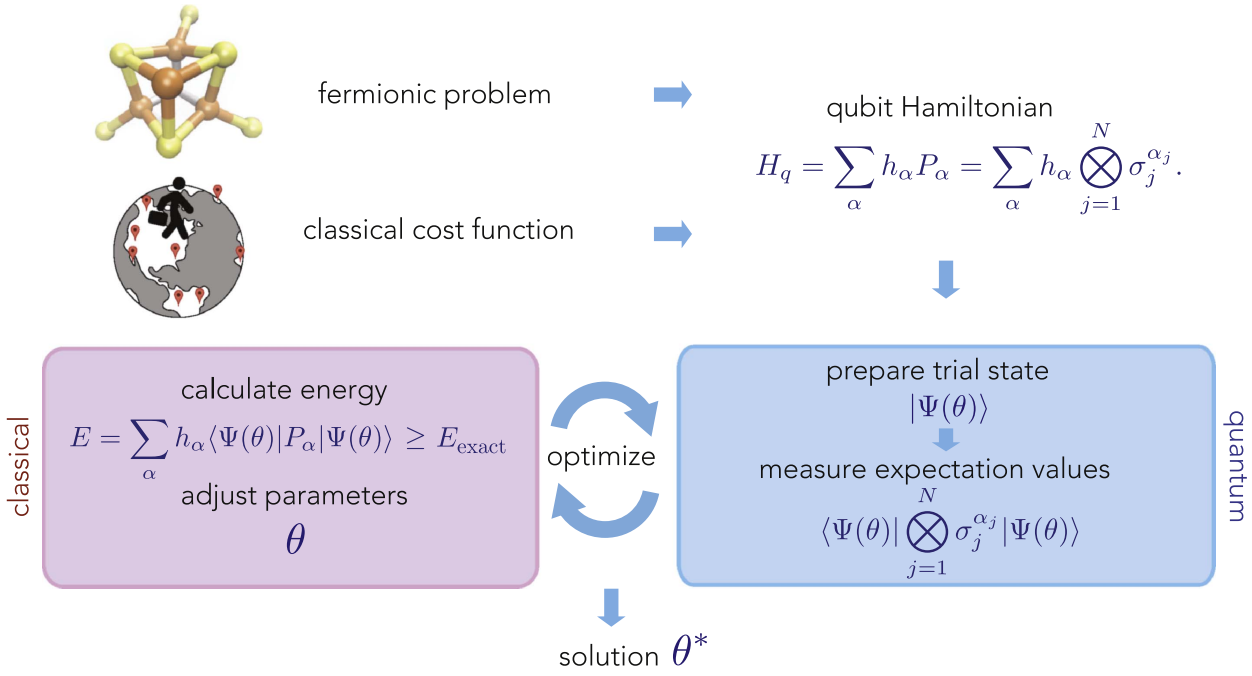


Figure 9.1: The steps of the VQE algorithm. From Ref. [Moll et al., 2018].

4. Update the parameters ϑ based on the result (and results in previous iterations).

Steps 1 and 2 are run on a quantum computer, which can handle the quantum states more efficiently than a classical computer. Steps 3 and 4 are done on a classical computer. The algorithm is iterative, i.e., it starts over from step 1 after step 4, and continues to iterate until some convergence criterion is met, indicating that the ground-state energy has been found. In the following sections, we discuss steps 0, 1, and 4 in more detail.

9.2 More on step 0 – mapping to a Hamiltonian

Broadly speaking, the VQE is currently mostly being considered for two types of problems: optimization problems, where H is a cost function for the problem, and many-body fermionic quantum systems, e.g. molecules (quantum chemistry). The first type of problems was discussed extensively in Chapter 8, where we saw several examples of how optimization problems can be mapped to a Hamiltonian. In this chapter, we therefore focus on quantum-chemistry problems.

Even though a Hamiltonian can be written down for a molecular system, that is not the Hamiltonian that is used in VQE. In classical simulations of molecular systems, there are many different methods, e.g., density functional theory (DFT), where the actual system of interacting electrons is described as non-interacting electrons moving in a modified external potential. An approach more suited to VQE is to describe the system in second quantization. This requires calculating a number of spatial integrals on a classical computer, but that task can be accomplished efficiently. The Hilbert space consists of electron orbitals. The Hamiltonian is

$$H_F = \sum_{i,j} t_{ij} a_i^\dagger a_j + \sum_{i,j,k,l} u_{ijkl} a_i^\dagger a_k^\dagger a_l a_j, \tag{9.3}$$

where a_i ($a_i^\dagger$) annihilates (creates) an electron in the ith orbital. The coefficients t_{ij} and u_{ijkl} describing one- and two-electron interactions are calculated from the spatial integrals mentioned above.

The operators in Eq. (9.3) are fermionic. They thus obey the fermionic anti-commutation relations, e.g., $\left\{a_i, a_j^\dagger\right\} = \delta_{ij}$. These are not the relations that the qubit Pauli operators obey. We thus need to translate the Hamiltonian in Eq. (9.3) to a form that can be implemented on the quantum computer. One well-known mapping from fermionic operators to qubit operators is the Jordan-Wigner transformation:

$$a_i^\dagger \to \mathbf{1}^{\otimes i-1} \otimes \sigma_- \otimes \sigma_z^{\otimes N-i}, \tag{9.4}$$

where N is the number of orbitals and qubits. This mapping is not well suited to the VQE, because it creates highly non-local terms in the qubit Hamiltonian. In actual applications of VQE to quantum chemistry, other mappings are used (Bravyi-Kitaev, parity, ...). There is ongoing research on finding more suitable mappings.

9.3 More on step 1 – the trial state

The trial state $|\Psi(\vartheta)\rangle$ can essentially be parameterized by ϑ in two ways: to form states that have a form that is suggested by the problem Hamiltonian, or to form states that are easy to create with the available quantum-computing hardware.

9.3.1 Problem-specific trial states

In quantum chemistry, a common class of trial states are created using a so-called coupled-cluster approach, often the unitary coupled-cluster (UCC) one. Here, the unitary operator $U(\vartheta)$ creates the trial state:

$$|\Psi(\vartheta)\rangle = U(\vartheta)\,|\Phi\rangle = \exp\left[T(\vartheta) - T^\dagger(\vartheta)\right]|\Phi\rangle\,, \tag{9.5}$$

where $|\Phi\rangle$ is a simple state formed by the Slater determinant for low-energy orbitals. The operator $T(\vartheta)$ is known as a cluster operator. It is given by

$$T(\vartheta) = \sum_k T_k(\vartheta), \tag{9.6}$$

$$T_1(\vartheta) = \sum_{i\in\text{occ},j\in\text{unocc}} \vartheta_i^j a_j^\dagger a_i, \tag{9.7}$$

$$T_2(\vartheta) = \sum_{i,j\in\text{occ},k,l\in\text{unocc}} \vartheta_{ij}^{kl} a_l^\dagger a_k^\dagger a_j a_i, \tag{9.8}$$

where the sums go over occupied and unoccupied orbitals. The coefficients of the higher-order cluster operators decrease rapidly as more orders are included. For this reason, the expansion is usually truncated at the second , "double", order (UCCSD) or the third, "triple", order (UCCSDT).

9.3.2 Hardware-efficient trial states

On an actual quantum computer, particularly a NISQ one, implementing the cluster operators can be hard, especially since the fermionic operators in the cluster operators must be mapped to qubit operators first. Therefore, hardware-efficient trial states are preferred. In the work of Ref. [Kandala et al., 2017], where the H_2, LiH, and BeH_2 molecules were simulated using 2, 4, and 6 qubits, respectively, the trial states were of the form

$$|\Psi(\vartheta)\rangle = \underbrace{U_\text{single}(\vartheta)U_\text{ent}(\vartheta)U_\text{single}(\vartheta)U_\text{ent}(\vartheta)\ldots U_\text{single}(\vartheta)U_\text{ent}(\vartheta)}_{d\text{ repetitions}}\,U_\text{single}(\vartheta)\,|00\ldots0\rangle\,. \tag{9.9}$$

Here, $U_{\text{single}}(\vartheta)$ represent arbitrary-single qubit rotations on each of the N qubits (different rotations in each of the $d+1$ steps) and $U_{\text{ent}}(\vartheta)$ represent two-qubit entangling operations (same in each step) that were easy to implement in the available hardware. For the single-qubit operations alone, there are $N(3d+2)$ independent rotation angles in the parameter vector ϑ (an arbitrary single-qubit rotation can be characterized by 3 Euler angles).

Already for relatively small molecules, d needs to be more than just a few repetitions to reach accuracy that can compete with classical methods. However, a larger d means that the quantum circuit takes longer to run, and thus decoherence will limit the achievable d. Recently, researchers are exploring "error mitigation" to get around this problem. In one type of error mitigation, the experiment is rerun several times with varying levels of added noise. From this, one can extrapolate the answer towards what it would have been for zero noise.

Note that the form of Eq. (9.9) is that of the QAOA in Eq. (8.51). This shows that the QAOA is an example of the broader class of algorithms that is the VQE.

9.4 More on step 4 – updating the parameters

Just like the other steps in the VQE that we have discussed so far, step 4 is also the subject of ongoing research. When searching for the ground-state energy of the problem Hamiltonian, there are several pitfalls that the update step must deal with. For example, the parameter landscape may have local minima. Furthermore, there is evidence that the landscape for larger problems can contain "barren plateaus". Both these problems are hard to deal with if one uses a standard gradient-descent-based search for the optimal parameters. Also, the value of E obtained in step 3 is noisy, since it is based on limited sampling of the expectation values for the Pauli strings making up the Hamiltonian (at some point, it becomes too costly to run the quantum computer enough times to sample all strings enough time eliminate the noise). The search method used needs to be robust against this noise. Another issue is that the number of parameters will be large for a larger problem. One possibility is to use gradient-free algorithms like Nelder-Mead.

There are many considerations that go into choosing the right method for updating the parameters. Yet another is that it can take non-negligible time to change all parameters and set up the instructions (pulse shapes, etc.) needed to implement step 1 on the quantum computer again.

Although VQE is an interesting heuristic hybrid quantum-classical algorithm for NISQ devices, it is clear that there is still much to be understood about the different steps of the algorithm. It is still unclear how well the VQE will scale with the size of the problems it is applied to.

Chapter 10

Sampling models and sub-universal models of quantum computation

These parts of the notes follow Refs. [Lund et al., 2017, Aaronson and Arkhipov, 2013], as well as the notes by Sevag Gharibian at the Quantum Complexity Theory 2019, University of Paderborn, http://groups.uni-paderborn.de/fg-qi/courses/UPB_QCOMPLEXITY/2019/UPB_QCOMPLEXITY_syllabus.html. We also highly recommend to read the short review Ref. [Harrow and Montanaro, 2017].

10.1 Introduction: motivation for sampling models

One of the difficulties in rigorously proving quantum advantage for computation is linked to the difficulty of bounding the power of classical computers. For instance, the celebrated Shor's polynomial time quantum algorithm for factorisation is important discovery for the utility of quantum computing, but is not satisfying in addressing the issue of quantum advantage due to the unknown nature of the complexity of factoring. The best known classical factoring algorithm, the general number field sieve, is exponential time (growing as $e^{cn^{1/3}\ln^{2/3} n}$ where n is the number of bits of the input number). However, in order to prove quantum advantage, or really any separation between classical and quantum computational models, it must be proven for all possible algorithms and not just those that are known.

The challenge of bounding the power of classical computation can be exemplified by the Shor's trilemma. The Extended Church-Turing Thesis asserts that any "reasonable" model of computation can be efficiently simulated on a standard classical computational model such as a Turing Machine or a Random Access Machine or a cellular automaton. Since Shor's algorithm allows to solve factoring in polynomial time, it is clear that one of the following statements must be false:

1. The Extended Church-Turing Thesis is true.
2. Factoring cannot be solved efficiently on a classical computer.
3. Large-scale universal quantum computers can be built (and hence there is something wrong in quantum mechanics textbooks).

One of the main motivations of studying (selected) sampling problems is that possible to prove that they are indeed problems that can be solved efficiently in polynomial time by a quantum computer, while they cannot be solved in polynomial time by a classical computer (up to widely believed conjectures in computer

science). Experimentally validating a sampling model would hence disprove the Extended Church-Turing thesis and thereby solve Shor's trilemma.

Sampling problems consist in generating output random numbers according to a particular probability distribution (see Fig. 1). All quantum computations on n qubits can be expressed as the preparation of an n-qubit initial state $|0\rangle^{\otimes n}$, a unitary evolution corresponding to a uniformly generated quantum circuit C followed by a measurement in the computational basis on this system. In this picture the computation outputs a length n bitstring $x \in \{0,1\}^n$ with probability

$$P_x = |\langle x|C|0\rangle^{\otimes n}|^2 \tag{10.1}$$

In this way quantum computers produce probabilistic samples from a distribution determined by the circuit C.

One of the appealing features of these models is that they are restricted models of quantum computation, or sub-universal, and hence they do not require a universal quantum computer: they might be implementable on a large scale via near-term "noisy intermediate scale quantum devices" (NISQ). This quest for an experimental demonstration of quantum computational speedup has fallen under the moniker of "quantum advantage", with multiple candidate approaches to date: The Instantaneous Quantum Polynomial-Time (IQP) model, random circuit sampling (RCS), and the deterministic quantum computation with one quantum bit (DQC1) model. Among these candidates, the Google AI-group recently released a demonstration of RCS with 53 qubits [Arute et al., 2019], which is at the edge of current simulation capability of classical computers [Pednault et al., 2019], and the Pann group a proof of Gaussian Boson Sampling with 100 optical modes and 50 input squeezed states [Zhong et al., 2020]. Despite disproving the Extended Church-Turing Thesis is inherently challenging because it is a scaling statement, these two experiment constitute a first proof of quantum advantage, in that classical computers cannot currently simulate the experiments themselves. Some comments on the Google experiments can be found in Sec. 10.3. Here, however, we shall focus on two of the first and historically more important models for quantum advantage: Instantaneous Quantum Polytime and Boson sampling. Note that the historically very first leap towards exhibiting a candidate model for quantum advantage was taken by Terhal and DiVincenzo in their 2008 paper "Adaptive quantum computation, constant depth quantum circuits and Arthur-Merlin games" [Terhal and DiVincenzo, 2002]. Their approach was already to appeal to a complexity-theoretic argument: they gave evidence that there exists a certain class of quantum circuits that cannot be simulated classically by proving that if a classical simulation existed, certain complexity classes strongly believed to be distinct would collapse to the same class.

Beyond the conceptual interest in unveiling quantum advantage with sampling models, we also underline that a convincing demonstration of the impossibility for a classical computer to produce the same output probability distribution as a quantum architecture is also a crucial milestone towards demonstrating the controllability of quantum computers, with the scope of using them to perform useful algorithms.

10.2 Instantaneous Quantum Polytime

IQP circuits are an intermediate model of quantum computation where every circuit has the form $C = H^{\otimes n} D H^{\otimes n}$, where H is a Hadamard gate and D is an efficiently generated quantum circuit that is diagonal in the computational basis. Sampling then simply corresponds to performing measurements in the computational basis on the state $H^{\otimes n} D H^{\otimes n}|0\rangle^{\otimes n}$. In [Bremner et al., 2010] it was argued that classical computers could not efficiently sample from IQP circuits (exact sampling case) where D is chosen uniformly at random from circuits composed of: (1) $\sqrt{C_Z}$ (square-root of controlled-Z), and $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ gates;

or (2) Z, C_Z, and CC_Z (doubly controlled-Z) gates. Between two layers of Hadamard gates is a collection of diagonal gates. Although these diagonal gates may act on the same qubit many times, they all commute so in principle could be applied simultaneously, hence the name "instantaneous". In the case of (1) these circuits correspond to random instances of the Ising model drawn from the complete graph [Bremner et al., 2016].

Ref. [Bremner et al., 2016] extends this result to the approximate sampling case relying on conjectures, similarily to the Boson Sampling model. In contrast to Boson Sampling, though, these conjectures have later been proven by Jens Eisert and collaborators.

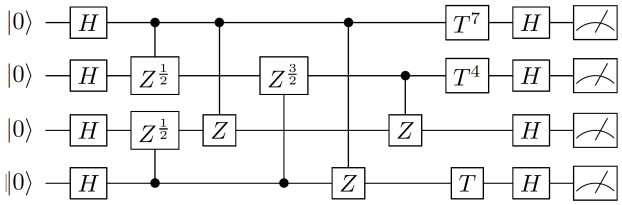


Figure 10.1: Example of an IQP circuit, taken from Ref. [Harrow and Montanaro, 2017].

The worst-case complexity of the problems in both (C1) and (C2) can be seen to be hard to classically sample in two steps.

First, one prove that these families of circuits are examples of sets that become universal under post-selection and as a result their output probabilities are hard to classically sample.

Then, complexity-theoretical arguments can be applied, that allow to conclude on the hardness of the model. We are going to review both aspects briefly here.

It can be shown that the complexity of computing the output probabilities of IQP circuits, $P_x = |\langle x|H^{\otimes n}DH^{\otimes n}|0\rangle^{\otimes n}|^2$, is classically hard in the worst case and this also holds under multiplicative approximation.

10.2.1 Hadamard gadget

This section is taken from the supplementary material of [Douce et al., 2017]. For either of the gate sets (1) or (2), the only missing ingredient for universality is the ability to perform Hadamard gates at any point within the circuit. In Ref. [Bremner et al., 2010] it was shown that such gates can be replaced with a "Hadamard gadget", which requires one post-selected qubit and controlled-phase gate per Hadamard gate. The Hadamard gadget [Bremner et al., 2010] is the very essence of the difficulty to simulate IQP circuits on classical computers. It shows that under post-selection an IQP circuit can implement a Hadamard gate.

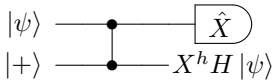


Figure 10.2: Hadamard gadget in a post-selected IQP circuit, where h takes value 0 if $+1$ is measured, while $h = 1$ if the result is -1.

Output state

Suppose one wants to implement a Hadamard gate on an arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Following the circuit structure depicted in Fig.10.2, we add an ancillary qubit initialized in $|+\rangle$ so that we start from

(omitting normalization)

$$|\psi\rangle|+\rangle = \alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle + \beta|11\rangle\,.$$

Then we apply the controlled Z gate and the measurement in the X basis. Conditioned on getting the outcome corresponding to the state $|+\rangle$ when measuring the first qubit we have:

$$\begin{aligned}\alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle + \beta|11\rangle &\xmapsto{\hat{C}_Z} \alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle - \beta|11\rangle \\ &\xmapsto{\langle+|} \alpha(|0\rangle + |1\rangle) + \beta(|0\rangle - |1\rangle) = H|\psi\rangle\,. \end{aligned} \tag{10.2}$$

If instead we get the outcome corresponding to the state $|-\rangle$ when we measure the first qubit, the same kind of calculations give:

$$\begin{aligned}\alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle + \beta|11\rangle &\xmapsto{\hat{C}_Z} \alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle - \beta|11\rangle \\ &\xmapsto{\langle-|} \alpha(|0\rangle + |1\rangle) - \beta(|0\rangle - |1\rangle) = XH|\psi\rangle\,. \end{aligned} \tag{10.3}$$

Defining h the outcome of the measurement, so that $h = 0$ (resp. $h = 1$) corresponds to measuring the state $|+\rangle$ (resp. $|-\rangle$), then the result of the computation is, in the general case

$$X^h H|\psi\rangle\,.$$

So the point of post selecting is to ensure it is indeed H and not $-H$ that has been implemented.

Probability of measuring $|+\rangle$

A subtlety with post-selection that is worth mentioning concerns the probability of the conditioning result. Specifically, if one wants to post-select on a qubit measured in a given state, then the probability associated with this measurement must be non zero. Thus it ensures that the conditional probability describing the post-selection is well-defined. In the case of the Hadamard gadget, we can compute the relevant success probability explicitely. We have after the $\hat{C}_Z$ gate – actually $1/2$ times the following equation for normalization purposes:

$$\alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle - \beta|11\rangle = (\alpha+\beta)|+0\rangle + (\alpha-\beta)|+1\rangle + (\alpha-\beta)|-0\rangle + (\alpha+\beta)|-1\rangle\,. \tag{10.4}$$

It is then obvious to show that the probability to measure $|+\rangle$ is

$$\frac{1}{4}\left(|\alpha+\beta|^2 + |\alpha-\beta|^2\right) = \frac{1}{2}.$$

An interesting feature of this result is that is doesn't depend on the input state $|\psi\rangle$. So even if initialized in $|-\rangle$, the entangling $\hat{C}_Z$ gate sort of smoothes the global state in such a way that the probability of measuring the first qubit in $|+\rangle$ is now $1/2$. Given that the number of post-selected lines l in a DV IQP circuit is of order of the total number of lines in the circuit n, $l \sim O(n)$, the overall success probability distribution $1/2^l$ is exponentially low in the circuit size. However we stress that this post-selection should be regarded as a mathematical tool for the hardness proof, and its actual implementation is not required in practice.

Complexity-theoretical arguments and proof of computational hardness

In Sec.10.2.1 we have shown that IQP with *postselected* measurements is universal for PostBQP (that is, quantum polynomial-time with postselection on possibly exponentially-unlikely measurement outcomes). In other words, to any computation in PostBQP corresponds a post-selected IQP circuit.

Furthermore, Aaronson previously showed that PostBQP = PP. On the other hand, if a classical algorithm existed for the simulation of IQP, then we will show that we could simulate postselected IQP in

PostBPP (that is, *classical* polynomial-time with postselection, also called $\mathsf{BPP}_{\mathsf{path}}$). This would yield to the following chain of inclusions of complexity classes:

$$\mathsf{BPP}_{\mathsf{path}} = \mathsf{PostBPP} = \mathsf{PostBQP} = \mathsf{PP}, \tag{10.5}$$

which is known to imply a collapse of the polynomial hierarchy. The final argument why this happens is that PP is as hard as PH (Toda's theorem, Gödel prize 1998), while PostBPP is contained in the third level of the polynomial hierarchy. Therefore, if exact IQP was efficiently classically simulatable, the full polynomial hierarchy would be contained in the third level, which implies the collapse.

10.3 Random Circuit Sampling

Google has recently published a paper demonstrating quantum advantage using a 53-qubit quantum computer [Arute et al., 2019]. From an information-theoretic perspective, the classical computational hardness of sampling from this circuit family has been demonstrated in Ref. [Bouland et al., 2019].

What do these results mean and what are the implications for WACQT and other efforts to build a quantum computer around the world? This section is the WACQT statement following release of the Google-AI experiment, and has been written by G. Johansson. It has been edited further by G. Ferrini after release of the IBM paper claiming that the output of the Google experiment can be simulated in 2,5 days [Pednault et al., 2019]. For more informations, and for a very clear explanation of the implications of this experiment for quantum advantage, we recommend to read the relevant blogposts on Scott Aaronson blog https://www.scottaaronson.com/blog/. For a very nice discussion on the terminology *quantum supremacy* used by many authors and reasons to avoid it, as well as for a pedagogical introduction to the Google experiment, visit https://phys.cam/2019/10/quantum-supremacy/.

1. It is a major milestone, demonstrating that a gate-based quantum computer can indeed perform a computational task in 3 minutes, which would take 2.5 days to solve on the most powerful supercomputer on earth. Importantly, if Google would now add a few more qubits to their chip, simulating the outcome would become impossible even for that supercomputer, in decent times. E.g., for 60 qubits, you would need 33 such supercomputers for just storing the quantum state of the Google chip.

2. The computation is not claimed to be useful in any way. The task is to sample from a particular probability distribution. This task was chosen carefully, because it is easy to perform on Google's quantum computer, but hard on any classical computer.

3. This does not imply that quantum computers from now on outperform classical computers in general. However as quantum computers evolve, the class of computational tasks where the quantum computer performs better than classical computers will grow. The hope is of course that at some point it will also contain useful computations.

4. The computation was performed without using error correction. The error rate was low enough to give the right answer for this comparably short quantum algorithm. This is a so-called NISQ device, where NISQ stands for Noisy Intermediate-Scale Quantum device.

5. The basic architecture of the 53-qubit device is similar to previously published devices from Google. The breakthrough consists of careful engineering of control hardware and software as well as a thorough analysis of which computational tasks are easy for a quantum computer and hard for classical computers.

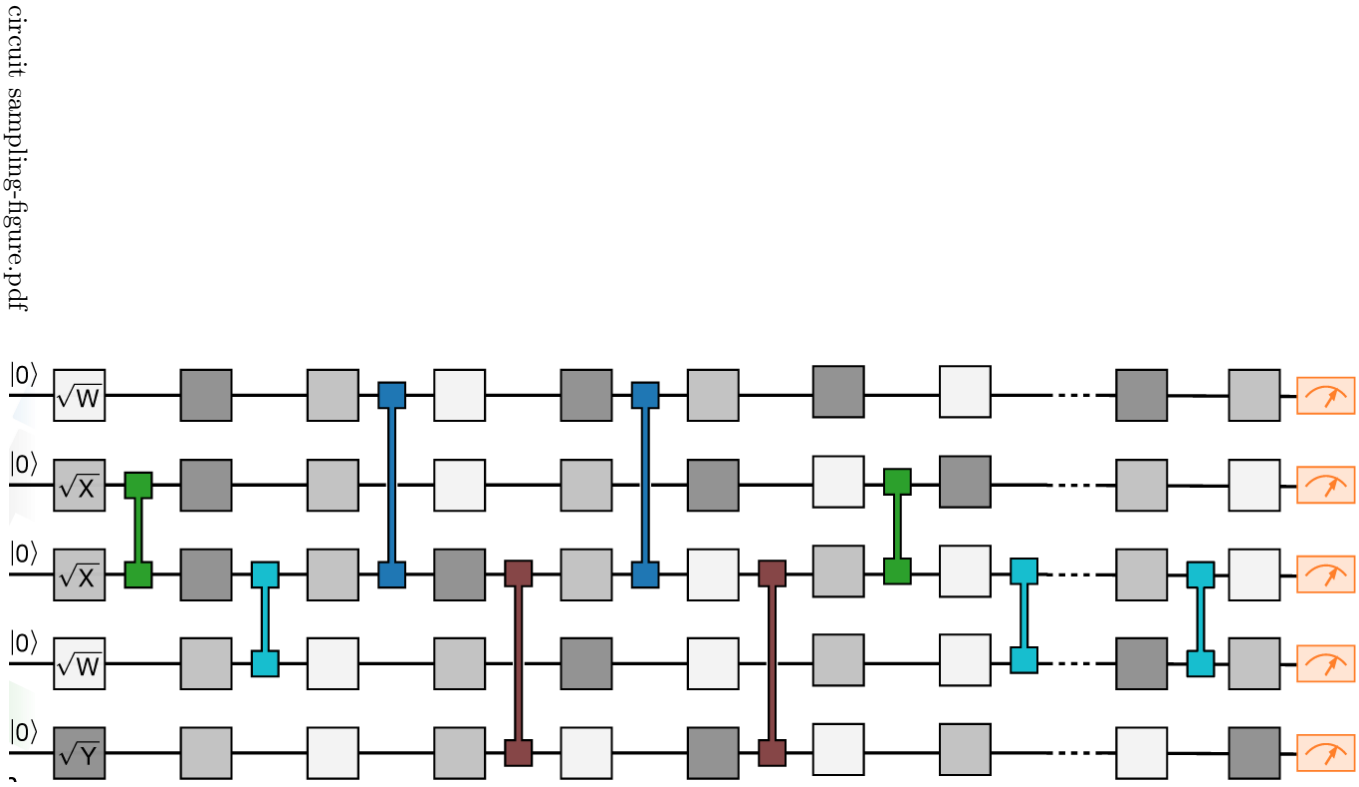


Figure 10.3: Sketch of a Random Circuit Sampling, taken from Ref. [Arute et al., 2019].

6. The algorithm creates a random entangled state by repeating layers of eight sets of gates, where most qubits are taking part in eight entangling gates, two with each nearest neighbour, interlaced with eight single qubit gates. For the longest algorithm, each layer is repeated twenty times, giving on the order of $53 \times (8/2) \times 20 \sim 4000$ two-qubit gates. The algorithm is then repeated one million times, to give appropriate statistics. The full run-time is 200 seconds. To perform the similar sampling on a supercomputer with 1 million cores is estimated to take 2.5 days.

7. In WACQT, we are inspired by this breakthrough. We note that the average lifetime (T1) of the Google qubits is 16 microseconds, while we have demonstrated reproducible lifetimes of around 80 microseconds. However, Google's design gives them very fast two-qubit gates, taking less than 20 ns. At the moment, we are working on increasing the speed of our two-qubit gates. We also note the importance of automated calibration and control software, which we are currently developing also for our setup. WACQT also takes full part in the work to find useful algorithms suitable for superconducting qubit architectures.

8. In contrast to the other circuits in this chapter that are candidate to yield quantum advantage, in [Arute et al., 2019] the gates that are implemented are in principle drawn from a universal gate set. More in detail, regarding single qubit gates, they implement $\sqrt{X}$, $\sqrt{Y}$, $\sqrt{W}$, with $W = (X + Y)/\sqrt{2}$ (see Fig. 10.3). They generate random quantum circuits using the two-qubit unitaries measured for each pair during simultaneous operation, rather than a standard gate for all pairs. The typical two-qubit gate is a full iSWAP with 1/6th of a full CZ. Using individually calibrated gates in no way limits the

universality of the demonstration.

10.4 Boson Sampling

10.4.1 Definition of the Boson Sampling model

Aaronson and Arkipov [Aaronson and Arkhipov, 2013] describe a simple model for producing output probabilities that are hard to classically sample. Their model uses bosons that interact only by linear scattering [1]. The bosons must be prepared in a Fock state and measured in the Fock basis. Consider at this purpose M

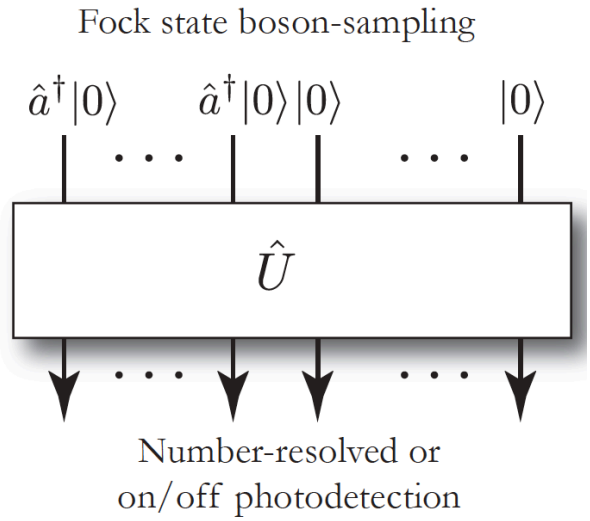


Figure 10.4

input ports of a multi-port splitter, which we feed with N photons. We assume that the input state only has maximum one input photon maximum per mode (see Fig. 10.4). With no loss of generality we order the modes such that the first N input modes contain a photon, while the others are empty. I.e,

$$|\psi_{\text{in}}\rangle = |1_1, ...1_N, ..., 0_M\rangle = \hat{a}_1^\dagger ... \hat{a}_N^\dagger |0_1, ..., 0_M\rangle \equiv |T\rangle. \tag{10.6}$$

Now a linear optics network described by the $M \times M$ matrix U is applied to the input state. Linear bosonic interactions, or linear scattering networks, are defined by dynamics in the Heisenberg picture that generate a linear relationship between the annihilation operators of each mode, i.e.

$$\hat{b}_j = \mathcal{U}^\dagger \hat{a}_j \mathcal{U} = \sum_{k=1}^{M} U_{j,k} \hat{a}_k, \text{ i.e. } \vec{b} = U\vec{a}; \tag{10.7}$$

$$\hat{b}_j^\dagger = \mathcal{U}^\dagger \hat{a}_j^\dagger \mathcal{U} = \sum_{k=1}^{M} U_{j,k}^\dagger \hat{a}_k^\dagger, \text{ i.e. } \vec{b^\dagger} = U^\dagger \vec{a^\dagger}, \tag{10.8}$$

[1] In this sense, therefore, the Boson Sampling model already lives in the bosonic space associated with an infinite dimensional Hilbert space and with continuous-variable operators, such as the quadratures of the field. However, we present this model in the discrete-variable section of the notes, to highlight the contrast with models making use of squeezed states and homodyne detection. The latter are more traditionally associated with the continuous-variable approach, and they will be presented in Chapter 11.

which also implies that $\hat{a}_j^\dagger = \sum_{k=1}^{M} U_{j,k}\hat{b}_k^\dagger$. It is important to make a distinction from the unitary operator $\mathcal{U}$ which acts upon the Fock basis and the unitary matrix defined by U which describes the linear mixing of modes. For optical systems the matrix U is determined by how linear optical elements, such as beam-splitters and phase shifters, are laid out. In fact all unitary networks can be constructed using just beam-splitter and phase shifters.

The set of events which are then output by the algorithm is a tuple of M non-negative integers whose sum is N. This set is denoted $\Phi_{M,N}$. As we will explicitly show in Sec.(10.4.2), the probability distribution of output events is related to the matrix permanent of sub-matrices of U. The matrix permanent is defined in a recursive way like the common matrix determinant, but without the alternation of addition and subtraction. For example

$$\text{Per}\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad + cb; \tag{10.9}$$

$$\text{Per}\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + ahf + bdi + cdh + cge. \tag{10.10}$$

In a more general form

$$\text{Per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)} \tag{10.11}$$

where S_n represents the elements of the symmetric group of permutations of n elements. With this, we can now define the output distribution of the linear network with the input state from Eq.(10.6). For an output event $S = (S_1, S_2, ..., S_M) \in \Phi_{m,n}$, the probability of S

$$P_S = \frac{|\text{Per}(U^S)|^2}{S_1!S_2!...S_N!} \tag{10.12}$$

where the matrix U_S is a $N \times N$ sub-matrix of U where column i is repeated S_i times and only the first N rows are used. One critical observation of this distribution is that all events are proportional to the square of a matrix permanent derived from the original network matrix U. The resulting photon distribution in output is hard to be classically sampled, as we will show later.

10.4.2 Proof that the Boson Sampling probability distribution is proportional to permanents

We now explicitly show that the output probability distribution of the boson sampling circuit is proportional to the permanent of the relevant submatrix. We follow a derivation similar to the one in the supplementary information of Ref. [Spring et al., 2013]. In a more general formulation the input population is $|T\rangle = |T_1, ...T_M\rangle$. The general case of arbitrary input state is treated in Ref. [Scheel, 2004].

In the Heisenberg representation, the input state does not evolve. However, using Eqs.(10.7) and (10.6) we can re-express it as

$$|\psi_{\text{out}}\rangle = |\psi_{\text{in}}\rangle = \left(\prod_{j=1}^{N} \hat{a}_j^\dagger\right)|0_1, ..., 0_M\rangle = \left(\prod_{j=1}^{N}\sum_{k=1}^{M} U_{j,k}\hat{b}_k^\dagger\right)|0_1, ..., 0_M\rangle. \tag{10.13}$$

The sum in Eq.(10.13) is composed of M^N terms, which is the number of ways in which we can put N objects in M modes allowing for repetitions; we can refer to this set of permutation as $\tilde{V}$, and then rename

the terms of the sum in Eq.(10.13), i.e.

$$\left(\prod_{j=1}^{N}\sum_{k=1}^{M}U_{j,k}\hat{b}_k^\dagger\right)=\sum_{j=1}^{M^N}\prod_{k=1}^{N}U_{k,\tilde{V}_k^j}\hat{b}_{\tilde{V}_k^j}^\dagger \tag{10.14}$$

where k is the k-th boson and $\tilde{V}_k^j$ is in which mode that photon is found in the permutation j. In other words, the ensemble $\tilde{V}$ is the set of M^N permutations of N photons in M modes, repetitions allowed. Let's consider as an example the case in which we have in input $N=2$ photons in $M=3$ modes. Then we have to consider $\prod_{j=1}^{2}\hat{a}_j^\dagger$, i.e. the first two rows of the vector

$$\begin{pmatrix}\hat{a}_1^\dagger\\ \hat{a}_2^\dagger\\ \hat{a}_3^\dagger\end{pmatrix}=\begin{pmatrix}U_{11}&U_{12}&U_{13}\\ U_{21}&U_{22}&U_{23}\\ U_{31}&U_{32}&U_{33}\end{pmatrix}\begin{pmatrix}\hat{b}_1^\dagger\\ \hat{b}_2^\dagger\\ \hat{b}_3^\dagger\end{pmatrix}=\begin{pmatrix}U_{11}\hat{b}_1^\dagger+U_{12}\hat{b}_2^\dagger+U_{13}\hat{b}_3^\dagger\\ U_{21}\hat{b}_1^\dagger+U_{22}\hat{b}_2^\dagger+U_{23}\hat{b}_3^\dagger\\ U_{31}\hat{b}_1^\dagger+U_{32}\hat{b}_2^\dagger+U_{33}\hat{b}_3^\dagger\end{pmatrix} \tag{10.15}$$

which gives

$$\begin{aligned}\prod_{j=1}^{2}\hat{a}_j^\dagger &= (U_{11}\hat{b}_1^\dagger+U_{12}\hat{b}_2^\dagger+U_{13}\hat{b}_3^\dagger)(U_{21}\hat{b}_1^\dagger+U_{22}\hat{b}_2^\dagger+U_{23}\hat{b}_3^\dagger)\\ &= U_{11}U_{21}\hat{b}_1^{\dagger 2}+U_{11}U_{22}\hat{b}_1^\dagger\hat{b}_2^\dagger+U_{11}U_{23}\hat{b}_1^\dagger\hat{b}_3^\dagger\\ &+ U_{12}U_{21}\hat{b}_2^\dagger\hat{b}_1^\dagger+U_{12}U_{22}\hat{b}_2^{\dagger 2}+U_{12}U_{23}\hat{b}_2^\dagger\hat{b}_3^\dagger\\ &+ U_{13}U_{21}\hat{b}_3^\dagger\hat{b}_1^\dagger+U_{13}U_{22}\hat{b}_2^\dagger\hat{b}_3^\dagger+U_{13}U_{23}\hat{b}_3^{\dagger 2}.\end{aligned} \tag{10.16}$$

Hence here in the sum of Eq.(10.16) we have

$$\begin{aligned}&j=1: \quad k=1\to\tilde{V}_k^j=1; \quad k=2\to\tilde{V}_k^j=1;\\ &j=2: \quad k=1\to\tilde{V}_k^j=1; \quad k=2\to\tilde{V}_k^j=2;\\ &j=3: \quad k=1\to\tilde{V}_k^j=1; \quad k=2\to\tilde{V}_k^j=3;\\ &j=4: \quad k=1\to\tilde{V}_k^j=2; \quad k=2\to\tilde{V}_k^j=1;\\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ...\end{aligned} \tag{10.17}$$

In practice, our linear network is going to redistribute the photons in the output modes (in the sense of their expected presence). The output state is hence a coherent superposition of various multi-Fock states according to a probability distribution which depends on the matrix U.

Let us indicate with S_i we indicate the number of photons in mode i in the configuration S, S^k indicates the mode in which the photon k is found in the configuration S, and where for each configuration we have $\sum_{i=1}^{M}S_i=N$. The total number of configurations is the number of ways of arranging N bosons in M modes, i.e. $N_{\text{config}}=\begin{pmatrix}N+M-1\\ N\end{pmatrix}$ (repetitions not allowed). If the total number of input photons is small compared to the number of modes such that $N\sim\sqrt{M}$, then the probability that two photons are found in the same output mode is rather small (*birthday paradox*). Nevertheless, we will consider here the general case of an arbitrary output distribution. The probability distribution S is evaluated by projection of the output state (10.13) onto the configuration state $|S\rangle\equiv|S_1,...,S_M\rangle$, i.e.

$$\begin{aligned}P_S &= |\langle S|\psi_{\text{out}}\rangle|^2=\frac{1}{S_1!...S_M!}\left|\langle 0_1,...,0_M|\left(\hat{b}_1^{S_1}...\hat{b}_M^{S_M}\right)\sum_{j=1}^{M^N}\prod_{k=1}^{N}U_{k,\tilde{V}_k^j}\hat{b}_{\tilde{V}_k^j}^\dagger|0_1,...,0_M\rangle\right|^2\\ &= \frac{1}{(S_1!...S_M!)}\left|\langle 0_1,...,0_M|\prod_{k=1}^{N}\hat{b}_{S^k}\left(\sum_{j=1}^{M^N}\prod_{k=1}^{N}U_{k,\tilde{V}_k^j}\hat{b}_{\tilde{V}_k^j}^\dagger\right)|0_1,...,0_M\rangle\right|^2.\end{aligned} \tag{10.18}$$

Before going to the general case of arbitrary output configuration, we want to fix the ideas with an example. Consider the case where we project onto the state $|\{S\}\rangle = |020\rangle = \hat{b}_2^{\dagger 2}/\sqrt{2}|000\rangle$. Then the only contributing terms in Eq.(10.16) is $U_{12}U_{22}\hat{b}_2^{\dagger 2}$, and we obtain:

$$\begin{aligned} P_S &= |\langle S|\psi_{\rm out}\rangle|^2 = \frac{1}{2!}\left|\langle 000|\hat{b}_2^2 U \hat{b}_1^\dagger \hat{b}_2^\dagger|000\rangle\right|^2 \\ &= \frac{1}{2}\left|\langle 000|\hat{b}_2^2 U_{12}U_{22}\hat{b}_2^{\dagger 2}|000\rangle\right|^2 \\ &= \frac{1}{2}4|U_{12}U_{22}|^2 = 2|U_{12}U_{22}|^2, \end{aligned} \tag{10.19}$$

where we have used that $\langle 0_2|\hat{b}_2^2\hat{b}_2^{\dagger S_2}|0_i\rangle = 2$. We can now verify that this expression is equivalent to the permanent of the submatrix of U identified by the rows corresponding to the input photons, and the columns identified by the output configuration of interest, where the columns are repeated as many times as the occupation of the output mode. In practice, we have:

$$U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}, \tag{10.20}$$

so that

$$U^{(S)} = \begin{pmatrix} U_{12} & U_{12} \\ U_{22} & U_{22} \end{pmatrix}, \tag{10.21}$$

and

$$\mathrm{Per}(U^S) = U_{12}U_{22} + U_{12}U_{22} = 2U_{12}U_{22}, \tag{10.22}$$

yielding immediately Eq.(10.19) according to (10.12).

In the general case, let us now indicate with $\tilde{S}_k^j$ the j-th permutations of the N photons in the output state, where k is the photon index. At maximum, there will be $N!$ permutations, corresponding to the ways of arranging N photons in N modes, repetitions not allowed (note that if in the output distribution there are more photon per mode then the number of ways I can put N photons in M modes with S_1 in the first, S_2 in the second... is $N!/(S_1!S_2!...S_M!)$). For example, it is clear that e.g. if we project onto the state $|\{S\}\rangle = |011\rangle$ then we have $S = [2,3]$ and

$$\begin{aligned} j=1: \quad & k=1 \to \tilde{S}_k^j = 2; \quad k=2 \to \tilde{S}_k^j = 3; \\ j=2: \quad & k=1 \to \tilde{S}_k^j = 3; \quad k=2 \to \tilde{S}_k^j = 2. \end{aligned} \tag{10.23}$$

It is clear that the only non-zero terms in the sum of Eq.(10.18) will be those for which $\tilde{V}_k^S = \tilde{S}_k^j$, hence[2]

$$P_S = \frac{1}{S_1!...S_M!}\left|\sum_{j=1}^{N!}\prod_{k=1}^{N} U_{k,\tilde{S}_k^j}\right|^2, \tag{10.24}$$

where we have used that $\langle 0_i|\hat{a}_i^{S_i}\hat{a}_i^{\dagger S_i}|0_i\rangle = S_i!$. For instance, we obtain for the state $|\{S\}\rangle = |011\rangle$ $P_S = |U_{12}U_{23} + U_{13}U_{22}|^2$, while for the state $|020\rangle$ we have seen that $P_S = 2|U_{12}U_{22}|^2$. We can now compare

[2]We have used that

$$\langle 0_1,...,0_M|\prod_{k=1}^{N}\hat{b}_{S^k}\left(\sum_{j=1}^{M^N}\prod_{k=1}^{N}U_{k,\tilde{V}_k^j}\hat{b}_{\tilde{V}_k^j}^\dagger\right)|0_1,...,0_M\rangle = S_1!...S_M!\sum_{j=1}^{M^N}\prod_{k=1}^{N}U_{k,\tilde{S}_k^j} = \sum_{j=1}^{N!}\prod_{k=1}^{N}U_{k,\tilde{S}_k^j}.$$

Eq.(10.24) with the formula for the permanent of an $L \times L$ matrix A, which reads

$$\mathrm{Per}(A) = \sum_{j=1}^{L!} \prod_{k=1}^{L} a_{k,\tilde{\sigma}_k^j}. \tag{10.25}$$

where $\tilde{\sigma}_k^j$ is the k-th element of the j-th permutation of the numbers $1, ..., L$. It is easy to compare Eq.(10.24) to Eq.(10.25); note however that our original matrix was $M \times M$, while we are computing here only the permanent of the sub-matrix involving the first N rows, and columns which correspond to configuration S. Hence we obtain Eq.(10.12).

10.4.3 Sketch of the proof of computational hardness of the Boson Sampling probability distribution

Exact Boson Sampling

The first main result of the original Aaronson paper states the following:

Theorem 1: hardness of exact Boson Sampling The exact Boson Sampling problem is not efficiently solvable by a classical computer, unless $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$ and the polynomial hierarchy collapses to the third level. More generally, let $\mathcal{O}$ be any oracle that "simulates boson computers", in the sense that $\mathcal{O}$ takes as input a random string r (which $\mathcal{O}$ uses as its only source of randomness) and a description of a boson computer A and returns a sample $\mathcal{O}_U(r)$ from the probability distribution $\mathcal{D}_U$ over possible outputs of U. Then $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$.

At least for a computer scientist, it is tempting to interpret Theorem 1 as saying that "the exact BosonSampling problem is #P-hard under $\mathsf{BPP}^{\mathsf{NP}}$-reductions." Notice that this would have a shocking implication: that quantum computers (indeed, quantum computers of a particularly simple kind) could efficiently solve a #P-hard problem! There is a catch, though, that has to do with the fact that BosonSampling is a sampling problem rather than a decision problem. In other words, the "reduction" from #P-complete problems to BosonSampling makes essential use of the hypothesis that we have a *classical* BosonSampling algorithm. Details can be found in the original article.

Two proofs of Theorem 1 can be given. In the first proof, we consider the probability p of some particular basis state when a boson computer is measured. We then prove two facts:

(1) Even *approximating* p to within a multiplicative constant is a #P-hard problem.

(2) *If* we had a polynomial-time classical algorithm for exact BosonSampling, *then* we could approximate p to within a multiplicative constant in the class $\mathsf{BPP}^{\mathsf{NP}}$, by using a standard technique called *universal hashing*.

Combining facts (1) and (2), we find that, if the classical BosonSampling algorithm exists, then $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$, and therefore the polynomial hierarchy collapses.

The second proof is inspired by independent work of Bremner, Jozsa, and Shepherd [Bremner et al., 2010], and namely by the proof of computational hardness for the IQP model that we have seen in Sec.10.2. In this proof, one starts with a result of Knill, Laflamme, and Milburn, which says that linear optics with *adaptive measurements* is universal for BQP, giving name to the respective KLM model. A straightforward modification of their construction shows that linear optics with *postselected* measurements is universal for PostBQP (that is, quantum polynomial-time with postselection on possibly exponentially-unlikely measurement outcomes). Furthermore, Aaronson previously showed that PostBQP = PP. On the other hand, if a classical BosonSampling algorithm existed, then we will show that we could simulate postselected linear

optics in PostBPP (that is, *classical* polynomial-time with postselection, also called $\mathsf{BPP}_{\mathsf{path}}$). We would therefore get exactly the same chain of inclusion as in Eq.(10.5), and the same conclusions on the hardness of the model apply.

Approximate Boson Sampling

While theoretically interesting, Theorem 1 unfortunately does not suffice to rule out the Extended Church Turing thesis, as even an optical setup realistically cannot perform exact Boson Sampling due to experimental noise. Thus, one must consider approximate Boson Sampling, and show that it is also hard to sample. More precisely, what Aaronson has demonstrated is that even sampling from a probability distribution $\mathcal{D'}_U$ that is away from the exact boson sampling one $\mathcal{D}_U$ by a certain error bound ε, i.e. $\sum_i \mathcal{D}^i_U - \mathcal{D'}^i_U < \varepsilon$, it is classically hard.

The proof of computational hardness of approximate boson sampling relies on two extra conjectures, beyond the fact that the polynomial hierarchy does not collapse, namely the Permanent of Gaussians conjecture, and the Permanent anti-concentration conjecture. We will not go into the details of the proof or hardness of approximate boson sampling here.

Experimental realisations and classical simulatability of Boson Sampling

Several proof of principle experiments have been achieved, the former ones with a handful of modes and single photons, see among others [Spring et al., 2013], and latest ones with up to 20 input photons injected in 60 modes [Wang et al., 2019]. Due to the probabilistic nature of single-photon sources, experimentalists have now moved towards Gaussian Boson Sampling to seek for large-scale demonstrations of quantum advantage, which was recently achieved (shortly after the Google experiment on RCS) using 50 input squeezed states, and 100 optical modes [Zhong et al., 2020]. The input squeezed states can be deterministically produced, and the results on the impossibility of simulating the outcome probability distributions still stand, although with a different proof technique [Hamilton et al., 2017]. Also note that, in contrast to random circuit sampling. at least one useful application of Gaussian Boson Sampling has been outlined, namely the calculation of vibronic molecular spectra [Huh et al., 2015]. For an extensive discussion see https://www.scottaaronson.com/blog/?p=5122. We will briefly review more sampling models with squeezed states and continuous-variables in Sec. 11.4.

Finally, note that classical algorithms are also constantly improving, which allow one for the simulation and benchmark of larger and larger Boson Sampling devices [Neville et al., 2017, Li et al., 2020]. Also, imperfections such as noise (photon losses, partial distinguishability of the photons) might render Boson Sampling circuits classically efficiently simulatable [Moylett et al., 2019, Qi et al., 2020].

Chapter 11

Continuous-Variable Approach to Quantum Information

In the Continuous-Variable (CV) approach to quantum information processing, relevant observables are characterised by a continuous spectrum, such as the amplitude $\hat{q} = (\hat{a}+\hat{a}^\dagger)/(\sqrt{2})$ and phase $\hat{p} = (\hat{a}-\hat{a}^\dagger)/(i\sqrt{2})$ quadratures of the electromagnetic field, satisfying $[\hat{q},\hat{p}] = i$. The associated Hilbert space is infinite-dimensional, and the (infinite) energy levels are eigenstates of the number operator $\hat{n} = \hat{a}^\dagger\hat{a}$. This is opposed to the traditional Discrete Variable (DV) approach, where observable have a discrete spectrum, and the Hilbert space is finite-dimensional. Generally speaking, CV systems offer the advantage that the resource states (e.g. squeezed states or large cluster states, that we will introduce in the following) can be deterministically produced. Moreover, new methods for experimental implementations, offering solutions to scalability, have emerged in the context of CV. For instance, A. Furusawa (Tokyo, Japan) and O.Pfister (Charlottesville, USA) have been able to produce CV entangled states of up to one-million optical modes; in the experiments of N. Treps and V. Parigi (Paris, France) several squeezed states are simultaneously available in the same optical cavity. With microwave cavities coupled to superconducting circuits, the Yale group has demonstrated that it is possible to store quantum information for a longer time in a radiation state (namely an encoded “cat state") then if the corresponding state is encoded in a qubit. CV are therefore promising for the implementation of scalable and robust architectures for quantum computing.

11.1 Quantum computing with continuous variables

11.1.1 The underlying physical model: the quantized harmonic oscillator

A review of the formalism underlying CV quantum information, namely the quantization of the harmonic oscillator, is provided in Appendix A. Here we provide the basic tools that we will need and use in the following. Also note that Appendix A uses a different systems of units than the one used in the rest of these notes.

Hamiltonian of the quantized harmonic oscillator

The Hamiltonian of the quantized harmonic oscillator, expressed in terms of the creation and annihilation (or ladder) operators, is given by

$$\hat{\mathcal{H}} = \hbar\omega\Big(\hat{a}^\dagger\hat{a} + \frac{1}{2}\Big).$$

This Hamiltonian has the following energy eigenvalues

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right), \quad n = 0, 1, 2, \ldots$$

and the eigenstates of the Hamiltonian are known as *Fock states*, written in the *Dirac notation* as $|n\rangle$, where n represents the number of quanta or photons in the single-mode field. The Fock states are eigenstates of the *number operator* $\hat{n} = \hat{a}^\dagger\hat{a}$, satisfying

$$\hat{a}^\dagger\hat{a}\,|n\rangle = n\,|n\rangle\,.$$

The vacuum state of the harmonic oscillator is defined by

$$\hat{a}\,|0\rangle = 0. \tag{11.1}$$

Acting with the creation and annihilation operators on the Fock state yields

$$\hat{a}\,|n\rangle = \sqrt{n}\,|n-1\rangle\,, \tag{11.2}$$

$$\hat{a}^\dagger\,|n\rangle = \sqrt{n+1}\,|n+1\rangle\,. \tag{11.3}$$

Hence it is clear that the creation operator $\hat{a}^\dagger$, creates a quanta of energy $\hbar\omega$ and the annihilation operator $\hat{a}$ destroys a quanta of energy $\hbar\omega$ in the single-mode field. Any Fock state can be generated by acting on the vacuum state multiple times with the creation operator

$$\frac{\hat{a}^{\dagger n}}{\sqrt{n!}}\,|0\rangle = |n\rangle\,.$$

Furthermore the number states are orthogonal

$$\langle m|n\rangle = \delta_{mn}$$

and form a complete set

$$\sum_{n=0}^{\infty} |n\rangle\,\langle n| = 1.$$

Coherent states

In quantum optics, the *coherent states* are the states with most resemblance to classical states, in the sense that they give rise to expectation values that look like the classical electric field. These states are the eigenstates of the annihilation operator:

$$\hat{a}\,|\alpha\rangle = \alpha\,|\alpha\rangle\,. \tag{11.4}$$

Because $\hat{a}$ is non-Hermitian α is usually complex. For the creation operator $\hat{a}^\dagger$ we have for obvious reasons

$$\langle\alpha|\,\hat{a}^\dagger = \alpha^*\,\langle\alpha|\,.$$

It is possible to show (see Appendix) that

$$|\alpha\rangle = e^{-|\alpha|^2/2}\sum_{n=0}^{\infty}\frac{\alpha^n}{\sqrt{n!}}\,|n\rangle\,,$$

which is a superposition of infinite many number of Fock states! Furthermore calculating the expectation value of the number operator $\hat{n}$

$$\bar{n} = \langle\alpha|\,\hat{n}\,|\alpha\rangle = |\alpha|^2,$$

we see that $|\alpha|^2$ is related to the mean number of photons in the field. Using this we can compute the probability of finding n photons in the field

$$|\langle n|\alpha\rangle|^2 = e^{-|\alpha|^2}\frac{|\alpha|^{2n}}{n!} = e^{\bar{n}}\frac{\bar{n}^n}{n!},$$

which we recognize as a *Poisson distribution* with a mean of $\bar{n}$. This distribution arises when the probability that an event occurs is independent of earlier events.

Coherent states are known to be *non-orthogonal*. For example, consider the scalar product $\langle\beta|\alpha\rangle$, where $|\alpha\rangle$ and $|\beta\rangle$ are to different coherent states

$$\begin{aligned}\langle\beta|\alpha\rangle &= e^{-|\beta|^2/2}e^{-|\alpha|^2/2}\sum_m\sum_n\frac{(\beta^m)^*\alpha^n}{\sqrt{m!}\sqrt{n!}}\langle m|n\rangle \\ &= e^{-(|\beta|^2+|\alpha|^2)/2}\sum_n\frac{(\beta^*\alpha)^n}{n!} \\ &= e^{-(|\beta|^2+|\alpha|^2-2\beta^*\alpha)/2} \\ &= e^{-|\alpha-\beta|^2/2}e^{(\alpha\beta^*-\beta\alpha^*)/2},\end{aligned}$$

taking the modulus square we get

$$|\langle\beta|\alpha\rangle|^2 = e^{-|\alpha-\beta|^2}. \tag{11.5}$$

From Eq. (11.5) it is evident that two coherent states are non-orthogonal. Only when $|\alpha-\beta|^2$ is large, so that $|\langle\beta|\alpha\rangle|^2 \sim 0$, they become quasi-orthogonal.

Quadrature operators

It is convenient to introduce the two hermitian operators

$$\begin{aligned}\hat{q} &= \frac{1}{\sqrt{2}}(\hat{a}+\hat{a}^\dagger), \\ \hat{p} &= \frac{1}{\sqrt{2}i}(\hat{a}-\hat{a}^\dagger),\end{aligned} \tag{11.6}$$

which satisfy the commutation relation

$$[\hat{q},\hat{p}] = i.$$

These are called the *quadratures* of the field, and they are observables. Measurement if the quadratures is called a homodyne measurement. These observables have a continuous spectrum, from which the name "continuous variables",

$$\hat{q}|s\rangle_q = s|s\rangle_q \tag{11.7}$$

$$\hat{p}|s\rangle_p = s|s\rangle_p. \tag{11.8}$$

As such, the quadratures possess the spectral decomposition

$$\hat{q} = \int ds|s\rangle_{qq}\langle s|s \tag{11.9}$$

$$\hat{p} = \int ds|s\rangle_{pp}\langle s|s. \tag{11.10}$$

Quadrature eigenstates form a basis, i.e. $\mathcal{I} = \int ds|s\rangle_{qq}\langle s| = \int ds|s\rangle_{pp}\langle s|$, which allows one to write arbitrary bosonic states in either the q or p representations, respectively:

$$|\psi\rangle = \int ds\psi_q(s)|s\rangle_p = \int ds\psi_p(s)|s\rangle_p,$$

with $\psi_q(s) = {}_q\langle s\,|\psi\rangle$ and with $\psi_p(s) = {}_p\langle s\,|\psi\rangle$.

The variance of an arbitrary operator is defined by

$$\langle(\Delta\hat{A})^2\rangle = \langle\hat{A}^2\rangle - \langle\hat{A}\rangle^2 \tag{11.11}$$

and can interpreted as the uncertainty of an observable. The expectation value of the quadratures

$$\langle\hat{q}\rangle = \frac{1}{\sqrt{2}}\langle n|\,(\hat{a}+\hat{a}^\dagger)\,|n\rangle = 0,$$
$$\langle\hat{p}\rangle = \frac{1}{\sqrt{2}i}\langle n|\,(\hat{a}-\hat{a}^\dagger)\,|n\rangle = 0$$

are evaluated to zero, which means that the expectation value of the electric field is also zero (see Eq. (A.17) in the Appendix). On the other hand the expectation value of the square is non-zero

$$\langle\hat{q}^2\rangle = \frac{1}{2}\langle n|\,(\hat{a}^2+\hat{a}\hat{a}^\dagger+\hat{a}^\dagger\hat{a}+\hat{a}^{\dagger 2})\,|n\rangle = \frac{1}{2}(1+2n),$$
$$\langle\hat{p}^2\rangle = \frac{1}{2}\langle n|\,(\hat{a}^2+\hat{a}\hat{a}^\dagger+\hat{a}^\dagger\hat{a}+\hat{a}^{\dagger 2})\,|n\rangle = \frac{1}{2}(1+2n).$$

Thus it follows from Eq. (11.11) that the uncertainty in both quadratures are equal, and when $n = 0$ (corresponding to the vacuum state), the uncertainty is minimum

$$\langle(\Delta\hat{q})^2\rangle_{\rm vac} = \frac{1}{2} = \langle(\Delta\hat{p})^2\rangle_{\rm vac}, \tag{11.12}$$

also implying the saturation of the Heisenberg uncertainty principle

$$\langle(\Delta\hat{q})^2\rangle_{\rm vac}\langle(\Delta\hat{p})^2\rangle_{\rm vac} = \frac{1}{4}. \tag{11.13}$$

This is known as the vacuum fluctuations. Since coherent states are nothing else than vacuum displaced in phase space, it can be easily verified that the quantum uncertainty for both quadratures on coherent states is the same as for vacuum, for all amplitude α:

$$\langle(\Delta\hat{q})^2\rangle_\alpha = \frac{1}{2} = \langle(\Delta\hat{p})^2\rangle_\alpha,$$

also resulting in minimum-uncertainly states:

$$\langle(\Delta\hat{q})^2\rangle_\alpha\langle(\Delta\hat{p})^2\rangle_\alpha = \frac{1}{4}. \tag{11.14}$$

Squeezed states

In contrast to coherent states, squeezed states are characterised by asymmetric fluctuations in q and p, i.e. $\langle(\Delta\hat{q})^2\rangle_\xi < \langle(\Delta\hat{p})^2\rangle_\xi$ for a q-squeezed state, and $\langle(\Delta\hat{q})^2\rangle_\xi > \langle(\Delta\hat{p})^2\rangle_\xi$ for a p-squeezed state, yet satisfying in both cases the Heisenberg principle with equality sign Eq.(11.14). Squeezed states are obtained applying the squeezing operator to the vacuum, i.e.

$$|\xi\rangle = S(\xi)|0\rangle = e^{-\frac{i\xi(\hat{q}\hat{p}+\hat{p}\hat{q})}{2}}|0\rangle.$$

In the limit of infinite squeezing, which corresponds to infinite energy, one obtains the infinitely squeezed states, which are the eigenstates of position and momentum with zero eigenvalue:

$$|0\rangle_q \text{ infinitely } q\text{-squeezed state, such that } \hat{q}|0\rangle_q = 0|0\rangle_q \tag{11.15}$$
$$|0\rangle_p \text{ infinitely } p\text{-squeezed state, such that } \hat{p}|0\rangle_p = 0|0\rangle_p. \tag{11.16}$$

Generalization of these states exist, where the state that is squeezed in Eq.(11.1.1) is an arbitrary coherent state.

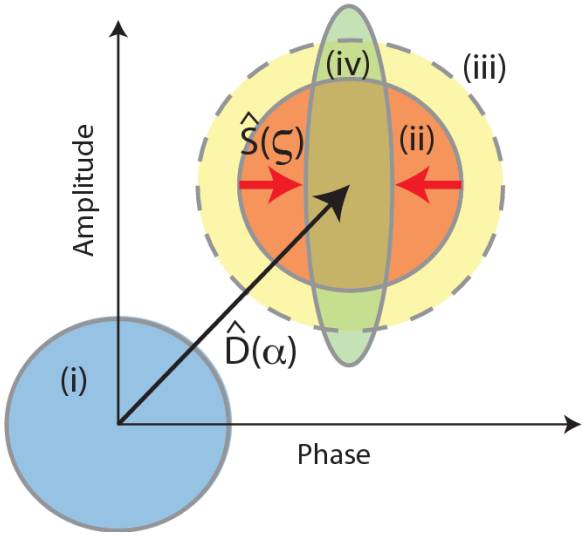


Figure 11.1: Coherent, Squeezed and Thermal States. Wigner function ball-on-stick representations of (i) vacuum state (blue), (ii) coherent state (red), (iii) thermal state (yellow, dashed line), and (iv) squeezed state (green). This picture is taken from Ref.["Storage and manipulation of optical information using gradient echo memory in warm vapours and cold ensembles", B. Sparkes, 2013].

Phase space representation

In quantum mechanics a system can be fully described by its density operator $\hat{\varrho}$, however the density operator can be a rather abstract object and it can be hard to read off its properties. Therefore we employ the *phase space* representation. One can think of phase space as a mathematical abstract space, where the state of a harmonic oscillator is represented in terms of its quadratures. For a general quantum system, since position $\hat{q}$ and momentum $\hat{p}$ are non commutative operators in quantum physics, the state cannot be represented as a point in phase space as it would in classical physics, because both position and momentum are not allowed to have precise values. The same holds true for the quadratures of the electromagnetic field, $\hat{q}$ and $\hat{p}$.

Among the possible representations of states in phase space there is the Wigner function. It is defined by

$$W(q,p) \equiv \frac{1}{2\pi}\int_{-\infty}^{\infty} \langle q+\tfrac{1}{2}x|\,\hat{\varrho}\,|q-\tfrac{1}{2}x\rangle\, e^{ixp}dx$$

where x, q and p are now interpreted as quadratures, and $\hat{\varrho}$ is the density operator for a quantum system. $W(q,p)$ is known as a *quasi-probability distribution* since it can take on negative values. Even though the Wigner function is not a "real" probability distribution it can still be associated to one, for example calculating the probability distribution (also referred to as the marginal distribution) over p

$$\begin{aligned}\Pr(q) = \int_{-\infty}^{\infty} W(q,p)dp &= \frac{1}{2\pi}\int_{-\infty}^{\infty} dp \int_{-\infty}^{\infty} dx\psi^*(q-\tfrac{1}{2}x)\psi(q+\tfrac{1}{2}x)e^{ixp} \\ &= \int_{-\infty}^{\infty} dx\psi^*(q-\tfrac{1}{2}x)\psi(q+\tfrac{1}{2}x)\delta(x) \\ &= \psi^*(q)\psi(q) = |\psi(q)|^2\end{aligned}$$

returns the probability density of q, where for simplicity of calculation we have considered the case of a pure

state $\hat{\varrho} = |\psi\rangle\langle\psi|$. Similarly, one can show that

$$\Pr(p) = \int_{-\infty}^{\infty} W(q,p)dq = |\psi(p)|^2$$

is the probability density of p, where $\psi(p)$ is the wave function in p-representation.

For instance, both the vacuum and the coherent state are shaped like a Gaussian which does not display any negativity, while the Wigner function for the single-photon Fock state shows clear indication of negativity. A cut of the Wigner function (as seen from the top) for a coherent state, a squeezed state, the vacuum and a thermal state is represented in Fig.11.1.

In Eq.(11.1.1), the Wigner function is associated to a quantum state (possibly mixed) $\hat{\varrho}$. However, it is also possible to associate a Wigner function to a process, or also to a measurement, where in the latter case ϱ is replaced by the projector associated with a given outcome, e.g. $|p\rangle\langle p|$ for the outcome p of the homodyne measurement $\hat{p} = \int ds|s\rangle_{pp}\langle s|s$. If the Wigner function is positive for all outcomes, then the we say that the measurement is Wigner-positive.

Mari-Eisert theorem

Any given CV quantum circuit is defined by (i) a specific input state, (ii) a unitary evolution and (iii) measurements. The Mari-Eisert theorem [Mari and Eisert, 2012] states that if all these elements are described by positive Wigner functions, then there exists a classical algorithm able to efficiently simulate this circuit. This theorem is the analog of the Gottesman-Knill theorem seen for qubits in Chapter 1. For a nice demonstration, see the licentiate thesis of Ingrid Strandberg (Chalmers, 2019), available at https://research.chalmers.se/publication/513592/file/513592_Fulltext.pdf.

Hence, including a negative Wigner function element is mandatory in order to design a CV sub-universal quantum circuit that cannot be efficiently simulated by a classical device. By virtue of the Hudson theorem, this necessarily corresponds to the use of non-Gaussian resources. Indeed, the Hudson theorem states that the only pure states to possess a non-negative Wigner function are Gaussian states.

Previous criteria for the simulatability of CV circuits were given in terms of the Gaussianity of a circuit: if all elements in a CV circuits are Gaussian, then the circuit is classically efficiently simulatable [Bartlett et al., 2002]. This criterion is strictly included in the Mari-Eisert theorem, i.e. it recognises as classically efficiently simulatable a smaller class of CV circuits: there are indeed states and processes that are non-Gaussian, and yet for which the Wigner function is positive. Consider for instance the mixed state of vacuum $|0\rangle$ and single-photon state $|1\rangle$, $\varrho = p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$ for any $p > 1/2$.

An even more general criterion with respect to the Mari-Eisert theorem for the simulatability of continuous-variable circuits was given in Ref. [Rahimi-Keshari et al., 2016], based on other quasi-probability distributions than the Wigner function.

11.1.2 First definitions, elementary operations and universal gate-sets

In the following, we are going to introduce the basic CV quantum operations, and to learn about several quantum computation models and protocols in CV. We start by introducing a notion of computational universality in CV, as well as the elementary CV operations.

Definition of CV universality (1)

The first definition of computational universality in CV that we will encounter in this course (and the first one historically) is the following: a CV QC system is universal if it can simulate the action of any Hamiltonian

$e^{iH(\hat{p}_i,\hat{q}_j)}$ consisting of a polynomial of the quadratures $\hat{q}_i$ and $\hat{p}_j$ in each mode i,j, to an arbitrary fixed accuracy [Lloyd and Braunstein, 1999, Gu et al., 2009].

It is also convenient to introduce the notion of Gaussian universality. This is given by LUBOs operations, for Linear unitary Bogoliubov transformations. They consist of any evolution operator involving a at most quadratic polynomial in $\hat{q}$ and $\hat{p}$. Introducing $\hat{\mathbf{x}} = (\hat{q}_1, \hat{q}_2 ... \hat{q}_N, \hat{p}_1, \hat{p}_2 ... \hat{p}_N)^T \equiv (\vec{q}, \vec{p})^T$, then these operations can be represented as

$$\hat{U}^\dagger \hat{\mathbf{x}} \hat{U} = S\hat{\mathbf{x}} + \hat{\mathbf{c}}, \tag{11.17}$$

with S a symplectic matrix, $\hat{\mathbf{c}}$ a displacement (see e.g. Ref. [Menicucci et al., 2011]). In the following, we are going to list the relevant CV elementary Gaussian operations, which allows reaching Gaussian universality:

Single mode Gaussian transformations

a) Rotations: $R(\vartheta) = e^{\frac{i\vartheta(\hat{q}^2+\hat{p}^2)}{2}}$[1]. Particular example: Fourier transform $R(\pi/2) = F$, with[2]

$$\begin{aligned} F|s\rangle_q &= \frac{1}{\sqrt{2\pi}} \int_\infty^\infty dr e^{irs} |r\rangle_q = |s\rangle_p \\ F^\dagger |s\rangle_p &= \frac{1}{\sqrt{2\pi}} \int_\infty^\infty dr e^{-irs} |r\rangle_p = |s\rangle_q \end{aligned} \tag{11.20}$$

with ${}_q\langle r|s\rangle_p = \frac{e^{irs}}{\sqrt{2\pi}}$ and $|s\rangle_p, |s\rangle_q$ the eigenstates of the quadrature operators introduced in Eq.(11.7).

b) Quadrature displacements:

$$\begin{aligned} X(s) &= e^{-is\hat{p}}, \quad \text{such that} \quad X(s)|r\rangle_q = |r+s\rangle_q \\ Z(s) &= e^{is\hat{q}}, \quad \text{such that} \quad Z(s)|r\rangle_p = |r+s\rangle_p. \end{aligned} \tag{11.21}$$

c) Squeezing: $S(s) = e^{-\frac{is(\hat{q}\hat{p}+\hat{p}\hat{q})}{2}}$

d) Shear: $D_{2,q} = e^{\frac{is\hat{q}^2}{2}}$

Any single mode Gaussian operation can be decomposed in a) rotations; b) quadrature displacement; c) or d), i.e. squeezing or shear. I.e,

$\{D_{1,q}(s) = Z(s), D_{2,q}(s), F = R(\pi/2)\}$ **universal set for single-mode Gaussian operations**

[1] Note that the action of a rotation of the quadratures of a single mode looks like

$$R(\vartheta) \begin{pmatrix} \hat{q} \\ \hat{p} \end{pmatrix} = \begin{pmatrix} \cos\vartheta & -\sin\vartheta \\ \sin\vartheta & \cos\vartheta \end{pmatrix} \begin{pmatrix} \hat{q} \\ \hat{p} \end{pmatrix}, \tag{11.18}$$

not to be confused with an equivalent matrix acting on the two-mode amplitude quadratures vector $\begin{pmatrix} \hat{q}_1 \\ \hat{q}_2 \end{pmatrix}$, realising the rotation of one quadrature $\hat{q}_1$ with respect to another one $\hat{q}_2$.

[2] The corresponding rotation of the second mode quadrature is expressed by

$$\begin{pmatrix} q' \\ p' \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} -p \\ q \end{pmatrix} \tag{11.19}$$

which corresponds to the transformations $F^\dagger \hat{q} F = -\hat{p}$ and $F^\dagger \hat{p} F = \hat{q}$, i.e. $F^\dagger \hat{a} F = i\hat{a}$ and $F^\dagger \hat{a}^\dagger F = -i\hat{a}^\dagger$.

Multimode Gaussian transformations

The addition of any non trivial two-mode Gaussian gate, such as the C_Z interaction $e^{ig\hat{q}\times\hat{q}}$, combined with the set of single-mode operations above allows for any general multi-mode Gaussian operation (see Ref. [Menicucci et al., 2011] for comments on the generalization to imperfect Gaussian operations and Gaussian measurements).

$$\{D_{1,q}(s) = Z(s), D_{2,q}(s), F = R(\pi/2), C_Z\} \quad \textbf{universal set for multi-mode Gaussian operations}$$

They are enough to perform some algorithms such as error correcting code regarding errors on single channels (though, non-Gaussian measurements are required to correct errors such as loss on all channels simultaneously). Yet, all the algorithms involving only Gaussian unitaries acting on Gaussian states with Gaussian measurements could be efficiently simulated on a classical computer, as we have seen in the previous section.

Single-mode universal transformations

For a single mode, all Gaussian operations together with any non-Gaussian operation provide universality. For example, the set $D_{k,q} = e^{\frac{is\hat{q}^k}{k}}$ for $k = 1, 2, 3$ for all $s \in R$ together with F provides universal single mode quantum computation (i.e., can be used to implement any single-mode unitary operation to arbitrary fixed accuracy):

$$\{D_{1,q}(s) = Z(s), D_{2,q}(s), D_{3,q}(s), F = R(\pi/2)\} \quad \textbf{universal set for single-mode q.c.}$$

Multi-mode universal transformations

Adding to this any non-trivial two-mode interaction provide universal quantum computation [Lloyd and Braunstein, 1999]. I.e.

$$\{D_{1,q}(s) = Z(s), D_{2,q}(s), D_{3,q}(s), F = R(\pi/2), C_Z\} \quad \textbf{universal set for multi-mode q.c.}$$

$$\left\{e^{i\hat{q}s}, e^{i\hat{q}^2 s}, e^{i\frac{\pi}{4}(\hat{q}^2+\hat{p}^2)}, e^{i\hat{q}_1\otimes\hat{q}_2}, e^{i\hat{q}^3 s}\right\} \quad \textbf{universal set for multi-mode q.c.}$$

11.2 Measurement-based quantum computation: the general paradigm in CV

It is in principle possible to perform sequences of gates from the elementary gate set that we have identified above [Hillmann et al., 2020], and to engineer thereby quantum computations in CV within the circuit model. However, CV quantum computation finds its most natural formulation within the measurement-based model. The reason for this is the availability of massively large cluster states, that can be deterministically generated (see experimental results by Furusawa, Pfister, Treps as well as latest releases from the Canadian start-up Xanadu). In this framework, we can reformulate the notion of universal quantum computation introduced in Sec. 11.1.2 as follows: for any CV unitary $U = e^{iH(\hat{q}_i,\hat{p}_j)}$ (corresponding to an arbitrary polynomial of $\hat{q}_i$ and $\hat{p}_j$) and any given input $|\varphi\rangle$ there exists an appropriate graph state such that by entangling the graph state locally with $|\varphi\rangle$ and applying an appropriate sequence of single-mode measurements, $U|\varphi\rangle$ is computed. We are now going to retrace all the steps seen in Sec. 6.2 when we have introduce the measurement-based quantum computation model for qubits, but here in the framework of CV. We start with the definition of CV cluster states.

11.2.1 Cluster states in Continuous Variables

Ideal cluster states are defined as follows: given a graph with N vertices and a certain number of edges relying these vertices according to a specific structure modeled by an adjacency matrix V, a CV cluster state is obtained starting from a collection of N infinitely p-squeezed states, and applying C_Z interactions according to the graph structure, i.e. the controlled-Z gate $e^{ig\hat{q}\times\hat{q}}$, yielding

$$|\psi_V\rangle = \hat{C}_Z[V]|0\rangle_p^{\otimes N} = \prod_{j,k}^{N} e^{\frac{i}{2}V_{jk}\hat{q}_j\hat{q}_k}|0\rangle_p^{\otimes N} = e^{\frac{i}{2}\hat{q}^T V\hat{q}}|0\rangle_p^{\otimes N} \tag{11.22}$$

Here V is a real and symmetric matrix, with finite elements.

Eq.(11.22) implies that a cluster state satisfies a nullifier relation, as detailed here below. Let us first introduce the following definition: If an operator K satisfies for a state $|\varphi\rangle$

$$K|\varphi\rangle = |\varphi\rangle \tag{11.23}$$

we call it a "stabilizer" for the state $|\varphi\rangle$. Eq.(11.23) implies that

$$UKU^\dagger(U|\varphi\rangle) = U|\varphi\rangle, \tag{11.24}$$

i.e that $UKU^\dagger$ stabilizes $U|\varphi\rangle$. Note furthermore that

$$e^{-is\hat{p}}|0\rangle_p \equiv X(s)|0\rangle_p = |0\rangle_p \forall s. \tag{11.25}$$

From Eqs.(11.25) and (11.24) it follows that the cluster state (11.22) is stabilized by the set

$$\begin{aligned} K_i &= \hat{C}_Z[V]X_i(s)\hat{C}_Z[V]^\dagger = e^{\frac{i}{2}\hat{q}^T V\hat{q}}X_i(s)e^{-\frac{i}{2}\hat{q}^T V\hat{q}} \\ &= \prod_{j,k}\prod_{l,m} e^{\frac{i}{2}V_{j,k}\hat{q}_j\hat{q}_k}e^{-is\hat{p}_i}e^{-\frac{i}{2}V_{l,m}\hat{q}_l\hat{q}_m} \end{aligned} \tag{11.26}$$

for each i. Using that $e^{i\hat{q}_1\hat{q}_2}\hat{p}_1 e^{-i\hat{q}_1\hat{q}_2} = \hat{p}_1 - \hat{q}_2$, we finally obtain that

$$K_i = e^{-is\hat{p}_i}\prod_k V_{i,k}e^{is\hat{q}_k} = X_i(s)\prod_k V_{i,k}Z_k(s). \tag{11.27}$$

Eq.(11.27) is formally equivalent to its analog in the discrete variable case (see Eq.(20.11) in Ref.[D. Druss and G. Leuchs, "Lectures on Quantum Information", Wiley-VCH (2007)]). The K_i form a group. The generators of the corresponding algebra are found by derivation since $K_i = e^{-isH_i}$. Note that because of Eq.(11.23) it follows that $H_i|\psi_V\rangle = 0$, i.e. $\forall i$. Hence the H operators are called "nullifiers" for the state $|\psi_V\rangle$. They can be easily calculated as

$$\begin{aligned} H_i &= i\left(\frac{dK_i}{ds}\right)_{(s=0)} = i\frac{d}{ds}\left[e^{-is\hat{p}_i}\prod_k V_{i,k}e^{is\hat{q}_k}\right]_{(s=0)} \\ &= i\left[-i\hat{p}_i e^{-is\hat{p}_i}\prod_k V_{i,k}e^{is\hat{q}_k} + e^{-is\hat{p}_i}\frac{d}{ds}\prod_k V_{i,k}e^{is\hat{q}_k}\right]_{(s=0)} \\ &= \hat{p}_i + i\left[e^{-is\hat{p}_i}\sum_k V_{i,k}i\hat{q}_k\prod_l e^{is\hat{q}_l}\right]_{(s=0)} = \hat{p}_i - \sum_k V_{i,k}\hat{q}_k \end{aligned} \tag{11.28}$$

from which follows that

$$\left(\hat{p}_i - \sum_k V_{i,k}\hat{q}_k\right)|\psi_V\rangle = 0. \tag{11.29}$$

From Eq.(11.29) follows immediately that

$$\langle\psi_V|\Delta^2\left(\hat{p}_i-\sum_k V_{i,k}\hat{q}_k\right)|\psi_V\rangle=0, \tag{11.30}$$

which for states with zero average also reads $\langle\psi_V|\left(\hat{p}_i-\sum_k V_{i,k}\hat{q}_k\right)^2|\psi_V\rangle=0$.

11.2.2 The CV MBQC paradigm

Consider the scheme in Fig. 11.2.

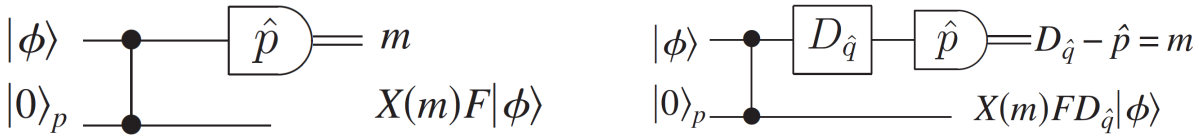


Figure 11.2

- **1) Preparation:** One mode contains the initial state that we want to process $|\varphi\rangle=\int dsf(s)|s\rangle_q$; the other mode is initialized to $|0\rangle_p$. The input state is hence $|\varphi\rangle\otimes|0\rangle_p=\int dsf(s)|s\rangle_q|0\rangle_p$. Apply a C_Z gate between the two, obtaining

$$\begin{aligned} C_Z(|\varphi\rangle\otimes|0\rangle_p) &= \int dsf(s)e^{i\hat{q}\otimes\hat{q}}|s\rangle_q|0\rangle_p=\int dsf(s)e^{is\hat{q}_2}|s\rangle_q|0\rangle_p=\int dsf(s)|s\rangle_q|s\rangle_p \\ &= \frac{1}{\sqrt{2\pi}}\int dsf(s)\int dre^{-irs}|r\rangle_p|s\rangle_p, \end{aligned} \tag{11.31}$$

  where we have used that $e^{is\hat{q}}|0\rangle_p=|s\rangle_p$.

- **2-pre) Measure:** Measure the input mode in the $\hat{p}$ basis with outcome m: this projects the second qumode into the state

$$|\psi\rangle_{\text{out}}\propto\int dsf(s)e^{-ims}|s\rangle_p=e^{-im\hat{p}}\int dsf(s)|s\rangle_p=X(m)F|\varphi\rangle. \tag{11.32}$$

  The last equality is obtained since $F|\varphi\rangle=\int dsf(s)F|s\rangle_q=\int dsf(s)|s\rangle_p$. The effect of this circuit is to apply the identity (modulo a displacement and a rotation).

- **2)** If we send as an input state the rotated state $D_{\hat{q}}|\varphi\rangle=\int dsf(s)D_{\hat{q}}|s\rangle_q$ where $D_{\hat{q}}=e^{if(\hat{q})}$ is an operator diagonal in the computational basis, then measuring $\hat{p}$ in the first qumode projects the second mode into

$$|\psi\rangle_{\text{out}}\propto X(m)FD_{\hat{q}}|\varphi\rangle. \tag{11.33}$$

  Since the C_Z gate commutes with $D_{\hat{q}}$, the same result is obtained with $|\varphi\rangle$ as an input state, and a rotation on the first mode after the C_Z as in the left panel of Fig.11.2. This is in turn equivalent to the situation in which no rotation is applied, but the first mode is measured in a rotated basis $D_{\hat{q}}^\dagger\hat{p}D_{\hat{q}}\equiv\hat{p}_{f(\hat{q})}$. The extra displacement $X(m)$ depends on the outcome of the measurement on mode 1, and can be compensated by choosing the measurement basis of the following steps (thus introducing in general time-ordering).

- **3) Universality of single mode operations:** Repeat twice the previous protocol, for two different operators $D^1_{\hat{q}}$ and $D^2_{\hat{q}}$. The output state is:

$$\begin{aligned} |\psi\rangle_{\text{out}} &= X(m_2)F(D^2_{\hat{q}}X(m_1))FD^1_{\hat{q}}|\varphi\rangle \\ &= X(m_2)FX(m_1)D^2_{\hat{q}+m_1}FD^1_{\hat{q}}|\varphi\rangle \\ &= X(m_2)FX(m_1)FD^2_{-\hat{p}+m_1}D^1_{\hat{q}}|\varphi\rangle \end{aligned} \tag{11.34}$$

  where we have used the inequalities $X(-m)\hat{q}X(m) = \hat{q} + m$, $Z(-m)\hat{p}Z(m) = \hat{p} + m$, $F^\dagger(-\hat{q})F = \hat{p}$, $F^\dagger\hat{p}F = \hat{q}$. If instead of measuring the second mode on $\hat{p}_{f(\hat{q})}$ I would have measured it in the outcome-dependent basis $\hat{p}_{f(-\hat{q}-m_1)}$ I would have obtained as a result my deterministic desired output

$$|\psi\rangle_{\text{out}} = X(m_2)FX(m_1)FD^2_{\hat{p}}D^1_{\hat{q}}|\varphi\rangle \tag{11.35}$$

  (universal for single-mode operations if I repeat other times: I can obtain the desired transformation concatenating various $D_{\hat{q}}$ and $D_{\hat{p}}$), a part from by-product operations which do not need to be corrected.

- **3)-4) Triviality of measurement adaptivity for Gaussian unitaries** Let us focus on the building blocks of the universal set given above. For the Gaussian operations:
  - F is obtained at each step of the computation.
  - $D_{\hat{q}} = e^{is\hat{q}}$ is obtained by measuring $\hat{p}_{s\hat{q}} = e^{-is\hat{q}}\hat{p}e^{is\hat{q}} = \hat{p} + s$ (measure $\hat{p}$ and add s to the result). Note that $\hat{p}_{s(\hat{q}+m)} = \hat{p}_{s\hat{q}} = \hat{p} + s$ (no adaptation is required).
  - $D_{\hat{q}} = e^{is\frac{\hat{q}^2}{2}}$ is obtained by measuring in the basis $\hat{p}_{s\hat{q}^2/2} = e^{-is\frac{\hat{q}^2}{2}}\hat{p}e^{is\frac{\hat{q}^2}{2}} = \hat{p} + s\hat{q} = g(\hat{q}\sin\vartheta + \hat{p}\cos\vartheta)$ with $g = \sqrt{1+s^2}$ and $\vartheta = \arctan s$. This is readily verified because the latter definition implies $\cos\vartheta = 1/\sqrt{1+s^2}$ and $\sin\vartheta = s/\sqrt{1+s^2}$. This corresponds to a rotated homodyne quadrature. Note that if I would have to adapt the basis I should measure according to $\hat{p}_{s(\hat{q}+m)^2/2} = \hat{p} + s\hat{q} + ms$. This can be achieved by measuring in the same basis as without adaptation (i.e. $\hat{p} + s\hat{q}$) and adding ms to the result

  The adaptation required for these measurements is trivial and can be done after (as a post-processing). Hence Gaussian operations can be implemented in any order or simultaneously ("parallelism").

  A cubic phase gate would require instead

  - $D_{\hat{q}} = e^{is\frac{\hat{q}^3}{3}}$ is obtained by measuring in the basis $\hat{p}_{s\hat{q}^3/3} = e^{-is\frac{\hat{q}^3}{3}}\hat{p}e^{is\frac{\hat{q}^3}{3}} = \hat{p} + s\hat{q}^2$. If I have to measure according to $\hat{p}_{s(\hat{q}+m)^3/3} = \hat{p} + s\hat{q}^2 + 2ms\hat{q} + m^2s$, the term $2ms\hat{q}$ requires a non-trivial adaptation of the measurement basis.

- **5) Cluster states as a resource:** Given the fact that the C_Z gates commute with the measurements, in practice the state used as initial resource in the quantum computation protocol presented is a generalized cluster state in which some of the modes (the input modes), also linked to the other nodes of the cluster, are initialized to code the modes of the input state. However, one can think of taking an initial cluster state (e.g, a square cluster state) and "writing" in some of its nodes the modes of the physical input state (e.g. by C_Z gates and measure of the input modes, or by teleportation [Ukai et al., 2010]). A state which allows this for each U and each input state is said to be a universal resource. It has been demonstrated by Briegel that a square lattice graph (a cluster state) with unit weights is a universal resource for quantum computation. Depending on the specific kind of computation, other graphs than a square lattice could be more suitable for implementing the computation [Horodecki et al., 2006].

- **6) Two mode interactions:** A sequence of single mode operations can be implemented via following measurements on a linear cluster. To achieve full universality we have to add to the previous toolbox a two-mode interaction, e.g. the C_Z gate. Such two-qubit gates, e.g., the CZ and CNOT gates, can be constructed in a two-dimensional cluster state where two input qubits are entangled with a few other qubits, in analogy to the case of qubit MBQC discussed in Sec.6.2. By a series of single-qubit measurements and rotations, we can end up with two of the other qubits representing the output state corresponding to the two-qubit gate having acted on the input state.

In conclusions, note that the procedure above is an idealization: in real life, squeezed states will always have finite energy, i.e. squeezing degree. As a result, the state output of the computation - even in the presence of ideal entangling gates and measurements - will always be affected by Gaussian noise, caused by the finite squeezing. How to avoid accumulation of this (and other types of) noise is the object of the following section.

11.3 GKP encoding and Error Correction

In classical informatics, when it comes to make sure that the errors that can occur during a computation can be corrected, it is convenient to resort to digitalized information, i.e. bits. For this reason combined with versatility, analog computers have been outperformed by digital computers in the 50s-60s, when the latter became sufficiently performant. Also note that from a computer science perspective, the definition of computational models based on real numbers is problematic and less studied[3].

Analogously, with quantum information, if the goal is to achieve fault-tolerant quantum computation, we must resort to qubit-like quantum information even when using continuous-variable hardware. An example of qubit-like quantum information encoding in CV is based on the use of cat states, where the qubit-like information is encoded in codewords $|0\rangle_L = |\alpha\rangle + |-\alpha\rangle$ and $|1\rangle_L = |i\alpha\rangle + |-i\alpha\rangle$, where we have omitted normalization constants. This encoding has allowed demonstrating a break-even point, in the sense that quantum information encoded in such cat states has been living longer than the one encoded in the corresponding qubit (within a transmon architecture) [Ofek et al., 2016].

In Ref. [Gottesman et al., 2001], another way of encoding qubits in quantized harmonic oscillators was introduced by Gottesman, Kitaev and Preskill, yielding the GKP encoding. This encoding has been shown to allow for the correction of arbitrary type of noise, while cat codes are specialized to correct for single-photon losses. Essentially, and without attempting to be rigorous, this is because GKP codes allows one to correct for single-mode displacements, and any noise-map can be decomposed in single-mode displacements [Gottesman et al., 2001]. GKP state have been generated and encoded experimentally both in superconducting microwave cavities [Campagne-Ibarcq et al., 2020] and with trapped ions [Flühmann et al., 2018].

In what follows, we are going to introduce the GKP encoding and the corresponding error-correcting scheme in detail.

11.3.1 GKP encoding

GKP code-words

We start by recalling the basis of GKP encoding. The starting point relies on the definition of qubits as continuous wave-functions made of an infinite number of Dirac peaks [Gottesman et al., 2001]:

$$\begin{aligned} |0_L\rangle &= \sum_n |2n\sqrt{\pi}\rangle_q = \sum_n |n\sqrt{\pi}\rangle_p \,, \\ |1_L\rangle &= \sum_n |(2n+1)\sqrt{\pi}\rangle_q = \sum_n (-1)^n |n\sqrt{\pi}\rangle_p \,. \end{aligned} \tag{11.36}$$

Realistic logical qubit states are normalizable finitely squeezed states, rather than non-normalizable infinitely squeezed states. The Dirac peaks are hence replaced by a normalized Gaussian of width Δ, while the infinite sum itself will become a Gaussian envelope function of width δ^{-1} (see Figure 11.3). Overall, the realistic states wavefunctions read:

$$\begin{aligned} \langle q|\tilde{0}_L\rangle &= \int dudv G(u)F(v)e^{-iu\hat{p}}e^{-iv\hat{q}}\langle q|0\rangle_L = N_0 \sum_n \exp\left\{\left(-\frac{(2n)^2\pi\delta^2}{2}\right)\right\}\exp\left\{\left(-\frac{(q-2n\sqrt{\pi})^2}{2\Delta^2}\right)\right\}, \\ \langle q|\tilde{1}_L\rangle &= \int dudv G(u)F(v)e^{-iu\hat{p}}e^{-iv\hat{q}}\langle q|1\rangle_L = N_1 \sum_n \exp\left\{\left(-\frac{(2n+1)^2\pi\delta^2}{2}\right)\right\}\exp\left\{\left(-\frac{(q-(2n+1)\sqrt{\pi})^2}{2\Delta^2}\right)\right\}, \end{aligned} \tag{11.37}$$

[3] If real computation were physically realizable, one could use it to solve NP-complete problems, and even #P-complete problems, in polynomial time. Unlimited precision real numbers in the physical universe are prohibited by the holographic principle and the Bekenstein bound.

where we have introduced the noise distributions

$$G(u) = \frac{1}{\Delta\sqrt{2\pi}} e^{-\frac{u^2}{2\Delta^2}}; \qquad F(v) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{v^2}{2\delta^2}}, \tag{11.38}$$

and N_0 and N_1 are normalization constants.

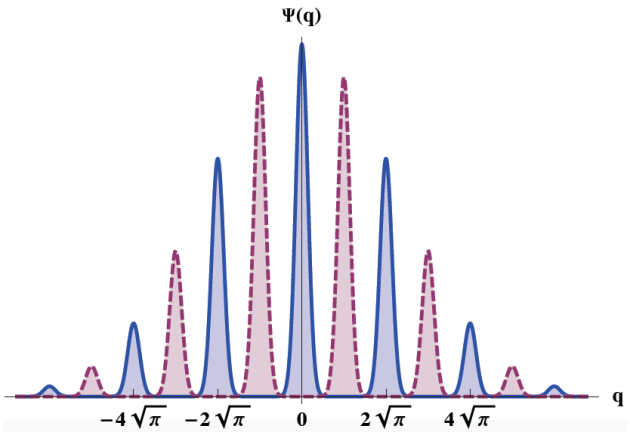


Figure 11.3: Wavefunction in position representation of GKP $|\tilde{0}_L\rangle$ state in continuous blue ($|\tilde{0}_L\rangle$ in dashed red) with $\delta = \Delta = 0.25$ from Equation Eq. (11.37)

.

Definition of CV universality (2)

A second definition of CV universality is based upon encodings, such as the GKP encoding: consider qubits encoded in CV hardware. In this case, universality is achieved when one can implement at least one of the universal gate sets for qubits quantum computation, that we introduced in Chapter 1, encoded in GKP.

On the (non-normalizable) states introduced in Eq.(11.36), Clifford operations correspond to the gates:

$$\bar{Z} = e^{i\sqrt{\pi}\hat{q}}, \bar{C}_Z = \ e^{i\hat{q}_k\hat{q}_l}, \ \bar{H} = F. \tag{11.39}$$

These are all implemented by Gaussian CV operations, as introduced in Sec.11.1.2. To promote this set of operations to universality we need a non-Clifford gate. This requires instead a non-Gaussian operation:

$$\bar{T} = e^{\frac{i\pi}{4}\left[\left(\frac{2\hat{q}}{\sqrt{\pi}}\right)^3 + \left(\frac{\hat{q}}{\sqrt{\pi}}\right)^2 - \left(\frac{2\hat{q}}{\sqrt{\pi}}\right)\right]}. \tag{11.40}$$

GKP encoding and fault-tolerance

In this Section, prepared with the contribution of Tom Douce, we prove that Continuous Variables MBQC with finite squeezing and an additional supply of GKP states yield fault tolerant quantum computation [Gottesman et al., 2001,Menicucci, 2014]. In Ref. [Gu et al., 2009] they showed how to implement standard quantum gates in CV MBQC, which would be sufficient for universal QC with GKP states [Gottesman et al., 2001], i.e. relying on a DV encoding embedded in a CV hardware. What remains to prove is that these gates can be performed fault-tolerantly, admitting use of GKP ancillary resource states. This was achieved in [Menicucci, 2014], where it is shown that the noise in the $\hat{p}$ quadrature of a GKP encoded quantum state can be replaced by the noise of the ancillary $|\tilde{0}_L\rangle$ state following the procedure shown in Fig. 11.4. Repeating this gadget after a Fourier transform allows for correction of the other quadrature, thereby enabling fault tolerance.

In order to explain this EC procedure, we follow a toy-model approach that has been developed by Glancy and Knill [Glancy and Knill, 2006]. This approach is based on a decomposition of the noise in several realizations of displacements, resulting in blurred ideal GKP states. Within this approach, we are going to show explicitly how GKP EC allows one to correct for displacements, by analyzing the output of the circuit in Fig. 11.5 with merely displaced perfect GKP states at the input, Sections 11.3.2 and 11.3.3. Since displacements form an operator basis, it follows that GKP states can correct any type of noise. Note that this works in principle for arbitrary noise, even when this is non Gaussian.

This approach is exact in the infinite squeezing regime. For physical states, i.e. with finite squeezing there are subtleties. Some of these are discussed in a recent paper by Barbara Terhal's group, that discusses the difference between error correction with coherent-enveloped GKP states versus blurred ideal GKP states. These subtle differences are not completely captured by the Glancy-Knill picture, but for pedagogical reasons we will omit further details of this discussion.

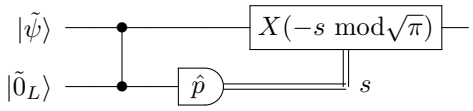


Figure 11.4: Procedure to correct for errors in the $\hat{p}$ quadrature. $|\tilde{0}_L\rangle$ is a noisy GKP state and $|\tilde{\psi}\rangle$ is a noisy GKP-encoded CV state. $X(m)$ is a displacement operator $e^{-im\hat{p}}$.

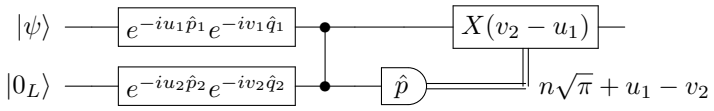


Figure 11.5: Modeling the noise in the protocol. $|0_L\rangle$ is a perfect – unphysical – GKP state and $|\psi\rangle$ is a perfect GKP-encoded CV state.

11.3.2 Single noise realization: intermediate measurement and threshold condition

Firstly, we compute the measurement result obtained at the output of the circuit in Fig. 11.5. The measurement is homodyne detection, corresponding to the observable $\hat{p} = \int p|s\rangle_{pp}\langle s|$, featuring the projectors

$$\hat{P}_s = |s\rangle_{pp}\langle s| \tag{11.41}$$

associated with the distinct measurement outcome s, where $|s\rangle_p$ are eigenstate of the $\hat{p}$ operator. In the following, we are going to omit the subscript when this does not cause confusion. In the following and unless specified all integrals will run over the whole real axis.

More specifically, we're interested in the observable $\mathcal{I}\otimes\hat{p}$ where $\mathcal{I} = \int \mathrm{d}q_1 \, |q_1\rangle \langle q_1|$ is the identity operator, acting on mode 1 which is unmeasured. From the initial unphysical and perfect input states, a set of displacements in position and momentum are applied before the $\hat{C}_Z$ gate and the measurement. Thus we want to work out the following quantity, in a sort of Heisenberg representation fashion:

$$e^{iv_2\hat{q}_2}e^{iu_2\hat{p}_2}e^{iv_1\hat{q}_1}e^{iu_1\hat{p}_1}\hat{C}_Z^\dagger(\mathcal{I}\otimes\hat{p})\hat{C}_Z e^{-iu_1\hat{p}_1}e^{-iv_1\hat{q}_1}e^{-iu_2\hat{p}_2}e^{-iv_2\hat{q}_2} \tag{11.42}$$

Step by step we have:

$$\begin{aligned}\hat{C}_Z^\dagger(\mathcal{I}\otimes\hat{p})\hat{C}_Z &= \hat{C}_Z^\dagger \int \mathrm{d}q_1\mathrm{d}p_2 p_2 \left|q_1, p_2\right\rangle \left\langle q_1, p_2\right| \hat{C}_Z \\ &= \int \mathrm{d}q_1\mathrm{d}p_2 p_2 \left|q_1, p_2 - q_1\right\rangle \left\langle q_1, p_2 - q_1\right| \end{aligned} \tag{11.43}$$

Then:

$$\begin{aligned} & e^{iv_2\hat{q}_2}e^{iu_2\hat{p}_2}e^{iv_1\hat{q}_1}e^{iu_1\hat{p}_1}\hat{C}_Z^\dagger(\mathcal{I}\otimes\hat{p})\hat{C}_Z e^{-iu_1\hat{p}_1}e^{-iv_1\hat{q}_1}e^{-iu_2\hat{p}_2}e^{-iv_2\hat{q}_2} \\ &= e^{iv_2\hat{q}_2}e^{iu_2\hat{p}_2}e^{iv_1\hat{q}_1}e^{iu_1\hat{p}_1}\int \mathrm{d}q_1\mathrm{d}p_2 p_2 \left|q_1, p_2 - q_1\right\rangle \left\langle q_1, p_2 - q_1\right| e^{-iu_1\hat{p}_1}e^{-iv_1\hat{q}_1}e^{-iu_2\hat{p}_2}e^{-iv_2\hat{q}_2} \\ &= \int \mathrm{d}q_1\mathrm{d}p_2 p_2 \left|q_1 - u_1, p_2 - q_1 + v_2\right\rangle \left\langle q_1 - u_1, p_2 - q_1 + v_2\right| \\ &= \int \mathrm{d}q_1\mathrm{d}p_2 p_2 \left|q_1, p_2 - q_1 - u_1 + v_2\right\rangle \left\langle q_1, p_2 - q_1 - u_1 + v_2\right| \\ &= \int \mathrm{d}q_1\mathrm{d}p_2 (q_1 + p_2 + u_1 - v_2) \left|q_1, p_2\right\rangle \left\langle q_1, p_2\right| = \hat{q}_1 + \hat{p}_2 + u_1 - v_2 \end{aligned} \tag{11.44}$$

where in the last steps we have performed the changes of variables:

$$\begin{aligned} q_1 - u_1 = q_1' &\to q_1 = q_1' + u_1 \\ p_2 - q_1 + v_2 = p_2' &\to p_2 = p_2' + q_1' + u_1 - v_2 \end{aligned}$$

and we have renamed $q_1' \to q_1$ and $p_2' \to p_2$. So what kind of measurement results are we to expect? To figure it out we should express Eq. (11.44) in a different manner, namely:

$$\begin{aligned} & e^{iv_2\hat{q}_2}e^{iu_2\hat{p}_2}e^{iv_1\hat{q}_1}e^{iu_1\hat{p}_1}\hat{C}_Z^\dagger(\mathcal{I}\otimes\hat{p})\hat{C}_Z e^{-iu_1\hat{p}_1}e^{-iv_1\hat{q}_1}e^{-iu_2\hat{p}_2}e^{-iv_2\hat{q}_2} \\ &= \int \mathrm{d}q_1\mathrm{d}p_2 (p_2 + q_1 + u_1 - v_2) \left|q_1, p_2\right\rangle \left\langle q_1, p_2\right| . \end{aligned} \tag{11.45}$$

We recall that the quantum state that will be measured is $|\psi_L, 0_L\rangle$ so a product of perfect GKP states. On these states, a quadrature projection can only give rise to integer multiples of $\sqrt{\pi}$. So the sum $q_1 + p_2$ will be a multiple of $\sqrt{\pi}$, say $n\sqrt{\pi}$. Specifically we get

$$s = n\sqrt{\pi} + u_1 - v_2. \tag{11.46}$$

The outcome of the measurement is a value corresponding to the noise of the data qubit, blurred by the noise coming from the ancilla.

The error threshold is defined in relation with the following displacement $X(-s \bmod \sqrt{\pi})$. The modulo function has range $[-\sqrt{\pi}/2, \sqrt{\pi}/2]$, and the procedure succeeds if $u_1 - v_2$ is small. Otherwise a logical error occurs and the displacement acts as Pauli-X gate in terms of the GKP encoding. Mathematically the constraint reads:

$$|u_1 - v_2| \leq \sqrt{\pi}/2. \tag{11.47}$$

The remaining logical errors can be taken care of by concatenating the GKP code with qubit codes and quantum error correction. A limit on the tolerated logical error probability depending on the chosen qubit code translates then into a constraint on the squeezing parameters [Menicucci, 2014].

11.3.3 Single noise realization: Output state of the GKP error-correcting gadget

We now compute the output state of the circuit Fig. 11.5. We show that the noise in the $\hat{p}$ quadrature of the logical qubit is replaced by the one given by the ancilla $|\tilde{0}_L\rangle$. Given Eq.(11.41) and standard quantum measurement theory [Nielsen and Chuang, 2011] after that the outcome s is obtained, the state is projected onto

$$|\psi_s\rangle \propto \hat{P}_s|\psi\rangle_{12} \tag{11.48}$$

where $|\psi\rangle_{12}$ is the state of input and ancilla after displacement and $\hat{C}_Z$ gate, i.e.

$$|\psi\rangle_{12} = \hat{C}_Z e^{-iu_1\hat{p}_1} e^{-iv_1\hat{q}_1} e^{-iu_2\hat{p}_2} e^{-iv_2\hat{q}_2} |\psi_L, 0_L\rangle\,, \tag{11.49}$$

and where again the identity $\mathcal{I} = \int \mathrm{d}q_1 \,|q_1\rangle\langle q_1|$ is implicitly acted in mode 1, and the projector $\hat{P}_s$ defined in Eq.(11.41) acts on mode 2. We therefore now compute explicitly,

$$\hat{P}_s\,|\psi_{12}\rangle = \int \mathrm{d}q_1\,|q_1\rangle\langle q_1| \otimes |s\rangle\langle s|\hat{C}_Z e^{-iu_1\hat{p}_1} e^{-iv_1\hat{q}_1} e^{-iu_2\hat{p}_2} e^{-iv_2\hat{q}_2}\,|\psi_L, 0_L\rangle \equiv |\Phi\rangle\,|s\rangle \tag{11.50}$$

with $|\Phi\rangle$ given by

$$\begin{aligned}
|\Phi\rangle &= \int \mathrm{d}q_1\,|q_1\rangle\langle q_1, s|\,\hat{C}_Z e^{-iu_1\hat{p}_1} e^{-iv_1\hat{q}_1} e^{-iu_2\hat{p}_2} e^{-iv_2\hat{q}_2}\,|\psi_L, 0_L\rangle \\
&= \int \mathrm{d}q_1\,|q_1\rangle\langle q_1, s-q_1|\,e^{-iu_1\hat{p}_1} e^{-iv_1\hat{q}_1} e^{-iu_2\hat{p}_2} e^{-iv_2\hat{q}_2}\,|\psi_L, 0_L\rangle \\
&= e^{i(v_1u_1-u_2s)} \int \mathrm{d}q_1 e^{-i(v_1-u_2)q_1}\,|q_1\rangle\langle q_1-u_1, s-q_1+v_2|\psi_L, 0_L\rangle.
\end{aligned} \tag{11.51}$$

Now we make explicit use of the form of the variable $s = u_1 + n\sqrt{\pi} - v_2$, and we have

$$\begin{aligned}
|\Phi\rangle &= e^{i(v_1u_1-u_2(u_1+n\sqrt{\pi}-v_2))} \int \mathrm{d}q_1 e^{-i(v_1-u_2)q_1}\,|q_1\rangle\langle q_1-u_1, n\sqrt{\pi}-q_1+u_1|\psi_L, 0_L\rangle \\
&= e^{-iu_2(n\sqrt{\pi}-v_2)} \int \mathrm{d}q_1 e^{-i(v_1-u_2)q_1}\,|q_1+u_1\rangle\langle q_1, n\sqrt{\pi}-q_1|\psi_L, 0_L\rangle.
\end{aligned} \tag{11.52}$$

Now let's go through the inner product:

$$\begin{aligned}
\langle n\sqrt{\pi}-q_1|0_L\rangle &= \sum_{l\in\mathcal{Z}} \langle n\sqrt{\pi}-q_1|l\sqrt{\pi}\rangle \\
&= \sum_{l\in\mathcal{Z}} \delta(q_1-(n-l)\sqrt{\pi}) \\
&= \sum_{l\in Z} \delta(q_1-l\sqrt{\pi}),
\end{aligned} \tag{11.53}$$

so we have:

$$\begin{aligned}
|\Phi\rangle &= e^{-iu_2(n\sqrt{\pi}-v_2)} \sum_{l\in\mathcal{Z}} \int \mathrm{d}q_1 \delta(q_1-l\sqrt{\pi}) e^{-i(v_1-u_2)q_1}\,|q_1+u_1\rangle\langle q_1|\psi_L\rangle \\
&= e^{-iu_2(n\sqrt{\pi}-v_2)} \sum_{l\in\mathcal{Z}} e^{-i(v_1-u_2)l\sqrt{\pi}}\,|l\sqrt{\pi}+u_1\rangle\langle l\sqrt{\pi}|\psi_L\rangle.
\end{aligned} \tag{11.54}$$

Since (the position wavefunction of) $|\psi_L\rangle$ is made of Dirac pikes on integer multiples of $\sqrt{\pi}$, the following result is straightforward: using that $\sum_{l\in\mathcal{Z}} |l\sqrt{\pi}\rangle\langle l\sqrt{\pi}|\psi_L\rangle = |\psi_L\rangle$ we obtain the final state before the displacement:

$$|\Phi\rangle = e^{-iu_2(n\sqrt{\pi}-v_2)} e^{-iu_1\hat{p}_1} e^{-i(v_1-u_2)\hat{q}_1}\,|\psi_L\rangle \tag{11.55}$$

and from Eq.(11.50)

$$\hat{P}_s \left|\psi_{12}\right\rangle = \left|\Phi\right\rangle \left|u_1 + n\sqrt{\pi} - v_2\right\rangle_p. \tag{11.56}$$

The conditional state on mode 2 is obtained as:

$$\hat{\varrho}_{k,\text{cond }1} = \text{Tr}_2[\hat{P}_s \left|\psi_{12}\right\rangle \left\langle\psi_{12}\right| \hat{P}_s] \tag{11.57}$$

$$= \int dp \delta(p - (u_1 + n\sqrt{\pi} - v_2)) \left|\Phi\right\rangle \left\langle\Phi\right| = \left|\Phi\right\rangle \left\langle\Phi\right| \tag{11.58}$$

corresponding to the pure state $\left|\Phi\right\rangle$.

Now we just have to deal with the remaining correction, i.e. the displacement by the actual measurement result obtained modulo $\sqrt{\pi}$, yielding $s[\text{mod}\sqrt{\pi}] = (u_1 - v_2)[\text{mod}\sqrt{\pi}]$

$$e^{is[\text{mod}\sqrt{\pi}]\hat{p}_1} \left|s\right\rangle \propto e^{is[\text{mod}\sqrt{\pi}]\hat{p}_1} \left|\Phi\right\rangle = e^{-iu_2(n\sqrt{\pi}-v_2)} e^{-iv_2\hat{p}_1} e^{-i(v_1-u_2)\hat{q}_1} \left|\psi_L\right\rangle \equiv \left|\psi_L\right\rangle_{\text{out}}, \tag{11.59}$$

which can also be expressed as

$$\left|\psi_L\right\rangle \longmapsto e^{-iu_2(p_0-u_1+n\sqrt{\pi})} e^{-iv_2\hat{p}_1} e^{-i(v_1-u_2)\hat{q}_1} \left|\psi_L\right\rangle \tag{11.60}$$

with $p_0 = u_1 - v_2$. So we have the equation of the output state on a single realization of the noise. Like in [Glancy and Knill, 2006], we can see that the remaining $\hat{p}_1$ displacement is given by v_2 and is independent of the original noise u_1. For $\hat{q}_1$ though the noise from the ancilla u_2 has been added to the original value v_1. It appears more clearly if we write Eq. (11.60) as in from [Glancy and Knill, 2006]. Then the output state would read – for an intermediate measurement result s:

$$e^{-iu_2(s-u_1)} e^{-iv_2\hat{p}_1} e^{-i(v_1-u_2)\hat{q}_1} \left|\psi_L\right\rangle \tag{11.61}$$

and we can see that the same discussion applies.

This error-correcting gadget can then be followed by a Fourier transform and by a second round of the same error-correcting gadget, with the use of another fresh GKP ancilla. This has the effect to correct for the noise in the second quadrature.

11.4 Sampling models and sub-universal models in Continuous Variables

In the previous Section, we have dealt with MBQC, which is a model for (universal) quantum computation. In this Section, we turn to sub-universal computational models in CV. In Sec. 11.1.1 we have seen that Wigner negativity is necessary in order to obtain quantum advantage - at least of the exponential type.

However, if one aims at minimal extensions of Gaussian models, the Boson Sampling model that we have seen in Section 10.4 appears as potentially "over-kill": both the input state and the measurement are described by negative Wigner functions, as they correspond respectively to single-photon states and photon counting measurement. Can we define other sub-universal model of quantum computation where only one of these elements is non-Gaussian, and show that they yield hard-to-sample output probability distributions?

Three different families of non trivial sub-universal quantum circuits can be defined, depending on whether the element yielding the Wigner function negativity is provided by the input state, the unitary evolution, or the measurement. This concept is exemplified in Fig.11.6. Although Wigner negativity allows stepping outside the range of applicability of the theorem in Ref. [Mari and Eisert, 2012], it is by itself not sufficient to imply classical hardness [García-Álvarez et al., 2020]. The classical hardness of these circuits, therefore, has yet to be proven, for each circuit type. For of the latter kind, corresponding to Gaussian Boson Sampling (GBS), this was done in Refs. [Lund et al., 2014, Hamilton et al., 2017]. These circuits are composed of input squeezed states, passive linear optics evolution, and photon counters. Circuits of the second kind are for instance related to the CV implementation of Instantaneous Quantum Computing – another sub-universal model, where input states and measurements are Gaussian, while the evolution contains non-Gaussian gates [Douce et al., 2017, Douce et al., 2019]. First definitions of the former class of CV circuits, i.e. that display non-Gaussian input state and Gaussian operations and measurements, have been very recently considered [Chakhmakhchyan and Cerf, 2017, Chabaud et al., 2017].

In this Chapter we are going to introduce the model CV-IQP (the continuous-variable version of the IQP model introduced in Sec.10.2) and to sketch the proof of its computational hardness.

11.4.1 Continuous-Variable Instantaneous Quantum Polytime

Definition of the model

We define the model Instantaneous Quantum Computing in CV (Fig.11.7). This model is composed of the elementary gates

$$\left\{ e^{id\hat{q}},\, e^{is\hat{q}^2}, e^{ic\hat{q}^3},\, e^{ib\hat{q}_1\hat{q}_2} \right\}. \tag{11.62}$$

All the gates in this model are diagonal in the position representation. As such, the Fourier transform is absent. Therefore, this model is not universal. For pedagogical purposes in these notes, we can assume in the definition that we are able to implement all the gates corresponding to all the possible choices of the real parameters in Eq.(11.62). However, this has shown to be un-necessary [Douce et al., 2019]. We require momentum squeezed states $|\sigma\rangle$ with $\sigma < 1$ to be present at the input. This model is a simpler version than the one introduced in Ref. [Douce et al., 2017]. Note the reminiscence with the IQP model defined in Sec.10.2. Here, too, we have a "crossed" structure of the type: $p - q - p$ in input state, evolution and measurement, as it was $x - z - x$ for the discrete-variable model.

For the proof of hardness, we proceed in the same way as seen in Chapter 10, when dwelling with the IQP model: we show that adding post-selection, the model becomes universal. In analogy to the Hadamard gadget of the discrete-variable model, we need therefore to develop a Fourier gadget.

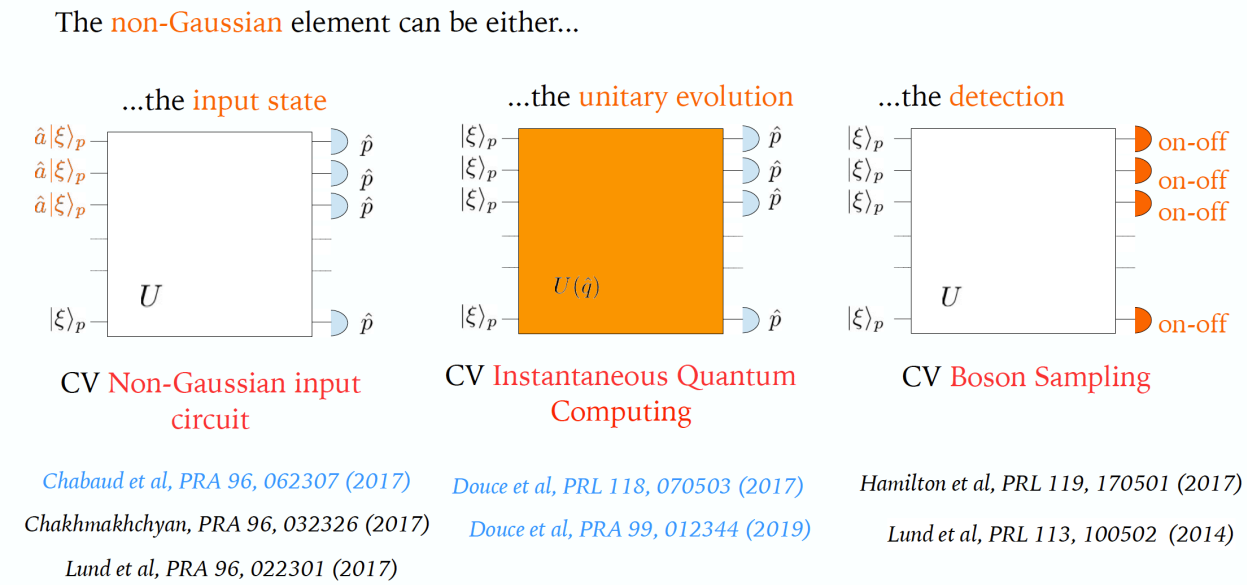


Figure 11.6: Families of quantum circuits in continuous-variable displaying minimal non-Gaussian character. Circuits with photon-subtracted squeezed states, linear evolution and heterodyne detection (instead than homodyne as represented in the leftermost sketch) have been shown to be hard to classically sample in Ref. [Chabaud et al., 2017]).

In the idealised case of infinite squeezing input states, this can be obtained fairly easily with the gadget of Fig.11.8. Adding post-selection allows to recover the Fourier transform, which completes the set of gates in the model.

However, a realistic model departs from this idealized situation in at least two unavoidable aspects. First, in order to obtain a probability of success different from zero for the homodyne measurement, thereby giving meaning to post-selection, we must introduce a binning of the real axis. This can still be equivalently modeled by using as a projective measurement. Instead that the ideal homodyne detector $\hat{p} = \int p \, |p\rangle \langle p|$, we are going to consider the finitely-resolved homodyne detector $\hat{p}^\eta$ operator that we define as [Paris et al., 2003]

$$\hat{p}^\eta = \sum_{k=-\infty}^{\infty} p_k \int_{-\infty}^{\infty} \mathrm{d}p \chi_k^\eta(p) |p\rangle\langle p| \equiv \sum_{k=-\infty}^{\infty} p_k \hat{P}_k \tag{11.63}$$

with $\chi_k^\eta(p) = 1$ for $p \in [p_k - \eta, p_k + \eta]$ and 0 outside, $p_k = 2\eta k$ and 2η the resolution, associated with the width of the detector pixels [4]. It is easy to check that this is still a projective measurement, since $\sum_{k=-\infty}^{\infty} \hat{P}_k = \mathcal{I}$, and $\hat{P}_k \hat{P}_{k'} = \hat{P}_k \delta_{k,k'}$ [5]. Note that this modelization is distinct from modeling imperfect detection efficiency [Leonhardt, 1997, Leonhardt and Paul, 1993, Paris et al., 2003]. Using this operator for the measurements introduces errors in the Fourier-transform applied with the Fourier-gadget. In particular, the output state is a mixes state. Second, realistic input states do have finite squeezing. Below, we are going to address the effect of these sources of noise, as well as how to deal with them.

[4] Note that this model turns out to be equivalent to an ideal scheme with perfectly resolving homodyne detectors and a discretization (binning) of the measurement outcomes.

[5] This result uses that $\int_{-\infty}^{\infty} dp' \chi_{k'}^\eta(p') \langle p'|\delta(p-p') = \chi_{k'}^\eta(p)\langle p|$ despite $\chi_{k'}^\eta(p')$ is not a smooth function, which can be verified with Riemann sum formalism.

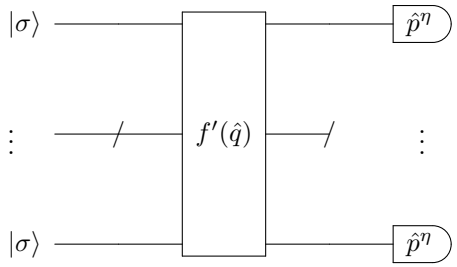


Figure 11.7: IQP circuit in CVs. $|\sigma\rangle$ are finitely squeezed states with variance σ in the $\hat{p}$ representation. The gate $f'(\hat{q})$ is a uniform combination of elementary gates from the set in Eqs.(11.62). The finitely-resolved homodyne measurement $\hat{p}^\eta$ has resolution 2η.

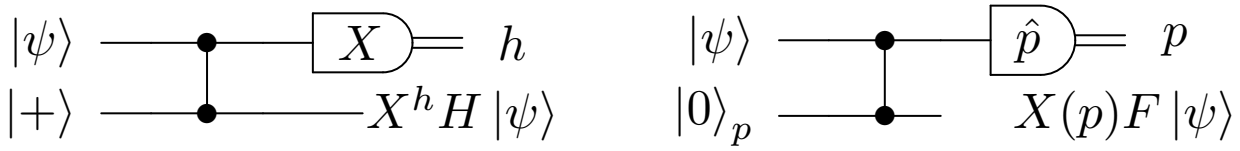


Figure 11.8: Left: Hadamard gadget in a post-selected IQP circuit, where h takes value 0 if $+1$ is measured, while $h = 1$ if the result is -1. Right: Ideal Fourier gadget in CVs, exact translation of the Hadamard gadget. $|0\rangle_p$ represents an infinitely $\hat{p}$-squeezed state with $\sigma = 0$, thus satisfying $\hat{p}|0\rangle_p = 0$.

Fourier Gadget for Continuous Variables

We consider in this subsection the actual Fourier gadget, provided by the circuit in Fig. 11.9, where we have removed the idealizations introduced for simplifying the discussion above. Namely, the ancillary squeezed state is finitely squeezed, and the homodyne detection performed on the first mode possesses a finite resolution. This subsection is taken from the Supplementary Material of Ref. [Douce et al., 2017].

Output state We compute the output state of the realistic Fourier transform gate implementation. The circuit is reproduced in Fig. 11.9. By convention the first (resp. second) ket in the tensorial product will refer to the upper (resp. lower) arm.

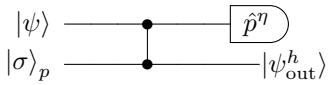


Figure 11.9

We recall that we start from:

$$|\psi\rangle \otimes |\sigma\rangle_p = \int \mathrm{d}q\, \psi(q)\, |q\rangle_q \otimes \frac{1}{\pi^{1/4}\sqrt{\sigma}} \int \mathrm{d}t\, e^{-\frac{t^2}{2\sigma^2}}\, |t\rangle_p\,. \tag{11.64}$$

Step by step we have first the $\hat{C}_Z$ gate:

$$\begin{aligned}\hat{C}_Z\, |\psi\rangle \otimes |\sigma\rangle_p &= \frac{1}{\pi^{1/4}\sqrt{\sigma}} \int \mathrm{d}q\mathrm{d}t\, e^{-\frac{t^2}{2\sigma^2}} \psi(q)\, |q\rangle_q\, |q+t\rangle_p \\ &= \frac{1}{\pi^{1/4}\sqrt{\sigma}} \int \mathrm{d}q\mathrm{d}t\, e^{-\frac{(t-q)^2}{2\sigma^2}} \psi(q)\, |q\rangle_q\, |t\rangle_p \equiv |\psi_{1,2}\rangle\,. \end{aligned} \tag{11.65}$$

We measure on the upper arm the finitely resolved $\hat{p}^{\eta}$ operator defined in Eq.(1) of the main text. When obtaining an outcome p_k, the measurement yields the conditional state on the lower arm

$$\begin{aligned}\hat{\varrho}_{k,\text{cond}} &= \text{Tr}_1\left[\hat{P}_k \otimes \mathcal{I}_2 \left|\psi_{1,2}\right\rangle \left\langle\psi_{1,2}\right| \hat{P}_k \otimes \mathcal{I}_2\right] \\ &= \int_{p_k-\eta}^{p_k+\eta} \mathrm{d}s \; {}_{p1}\langle s \left|\psi_{1,2}\right\rangle \left\langle\psi_{1,2}\right| s\rangle_{p1} \\ &= \frac{\eta}{\pi^{3/2}\sigma} \int \mathrm{d}q\mathrm{d}t\mathrm{d}q'\mathrm{d}t' e^{-\frac{(t-q)^2}{2\sigma^2}} e^{-\frac{(t'-q')^2}{2\sigma^2}} \psi(q)\psi^*(q')\text{sinc}(\eta(q-q'))e^{ip_k(q-q')} \left|t\right\rangle_p \left\langle t\right|'_p \end{aligned} \tag{11.66}$$

where we have used

$$\int_{-\eta}^{\eta} \mathrm{d}s \, e^{is(q-q')} = 2\eta\text{sinc}(\eta(q-q')). \tag{11.67}$$

We remark that the same expression as in Eq.(11.66) is obtained if the homodyne detectors are perfectly resolved, and a discretization is performed after measurement by binning the measurement outcomes.

This state then has to be normalized by the probability of getting the outcome corresponding to the projection operator above. What really matters to us is $\hat{\varrho}_{k=0,\text{cond}}$ corresponding to the outcome $p_k = 0$, because it is indeed the particular post-selected state that corresponds to the implementation of the Fourier transform. For this specific outcome we have:

$$\hat{\varrho}_{k=0,\text{cond}} = \frac{\eta}{\pi^{3/2}\sigma} \int \mathrm{d}q\mathrm{d}t\mathrm{d}q'\mathrm{d}t' e^{-\frac{(t-q)^2}{2\sigma^2}} e^{-\frac{(t'-q')^2}{2\sigma^2}} \psi(q)\psi^*(q')\text{sinc}(\eta(q-q')) \left|t\right\rangle_p \left\langle t\right|'_p . \tag{11.68}$$

Notice that in the limit of perfect resolution $\eta \to 0$ (upon normalization) we re-obtain the state that would be obtained in an MBQC implementation of the Fourier transform with a finitely squeezed ancillary state. As can be seen in Eq. (11.68), finite squeezing means convoluting the state with a Gaussian in the momentum representation, or equivalently multiplication with a Gaussian in the position representation.

Probability of measuring $p_k = 0$, Prob$[k = 0]$ We evaluate here the probability of measuring an outcome $p_k = 0$ within a window function of width 2η, yielding the conditional state in Eq.(11.68). More precisely, we consider the expectation value of the following operator:

$$\hat{P}_0 = \int_{-\eta}^{\eta} \mathrm{d}s \left|s\right\rangle_p \left\langle s\right| \tag{11.69}$$

taken in the state after the $\hat{C}_Z$ gate, that is (see Eq. (11.65))

$$\left|\psi_{1,2}\right\rangle = \frac{1}{\pi^{1/4}\sqrt{\sigma}} \int \mathrm{d}q\mathrm{d}t \, e^{-\frac{(t-q)^2}{2\sigma^2}} \psi(q) \left|q\right\rangle_q \left|t\right\rangle_p .$$

The calculation reads:

$$\begin{aligned}\text{Prob}[k=0] &= \left\langle\psi_{1,2}\right| \hat{P}_0 \otimes \mathcal{I}_2 \left|\psi_{1,2}\right\rangle \\ &= \frac{1}{\sigma\sqrt{\pi}} \int \mathrm{d}q\mathrm{d}q'\mathrm{d}t\mathrm{d}t'\mathrm{d}s \, e^{-\frac{(t-q)^2}{2\sigma^2}} e^{-\frac{(t'-q')^2}{2\sigma^2}} \psi^*(q')\psi(q)\delta(t-t')_q\langle q'|s\rangle_{pp}\langle s|q\rangle_q \\ &= \frac{1}{2\sigma\pi^{3/2}} \int \mathrm{d}q\mathrm{d}q'\mathrm{d}t\mathrm{d}s \, e^{-\frac{(t-q)^2}{2\sigma^2}} e^{-\frac{(t-q')^2}{2\sigma^2}} \psi^*(q')\psi(q)e^{is(q-q')} \\ &= \frac{1}{2\pi} \int \mathrm{d}q\mathrm{d}q'\mathrm{d}s \, e^{-\frac{(q-q')^2}{4\sigma^2}} \psi^*(q')\psi(q)e^{is(q-q')} \\ &= \frac{2\eta\sigma}{\sqrt{\pi}} \int \mathrm{d}q\mathrm{d}q' \frac{1}{2\sigma\sqrt{\pi}} e^{-\frac{(q-q')^2}{4\sigma^2}} \psi^*(q')\psi(q)\text{sinc}(\eta(q-q')). \end{aligned} \tag{11.70}$$

where from the second to the third line we used that

$$\int_{-\infty}^{+\infty} \mathrm{d}t\, e^{-\frac{(t-q)^2}{2\sigma^2}} e^{-\frac{(t-q')^2}{2\sigma^2}} = \sqrt{\pi}\sigma e^{-\frac{(q-q')^2}{4\sigma^2}}. \tag{11.71}$$

while in the last step we have used Eq.(11.67). The probability can be Taylor expanded in terms of powers of η:

$$\mathrm{Prob}[k=0] = \frac{2\eta\sigma}{\sqrt{\pi}} \left(\int \mathrm{d}q\mathrm{d}q' \frac{1}{2\sigma\sqrt{\pi}} e^{-\frac{(q-q')^2}{4\sigma^2}} \psi^*(q')\psi(q) + O(\eta^2) \right). \tag{11.72}$$

The first term in the parenthesis is precisely the norm $\langle \psi_{1,2} | \psi_{1,2} \rangle$ hence is equal to 1. Consequently the probability reads:

$$\mathrm{Prob}[k=0] = \frac{2\eta\sigma}{\sqrt{\pi}} + O(\eta^3). \tag{11.73}$$

The dominating order is thus proportional to the resolution 2η.

Large squeezing limit We note that Gaussian distributions obey the following relation: $\frac{1}{2\sqrt{\pi}\sigma} e^{-\frac{(q-q')^2}{4\sigma^2}} \underset{\sigma\to 0}{\longrightarrow} \delta(q-q')$. Based on this property, the integral in Eq. 11.70 actually yield:

$$\begin{aligned} &\int \mathrm{d}q\mathrm{d}q' \frac{1}{2\sigma\sqrt{\pi}} e^{-\frac{(q-q')^2}{4\sigma^2}} \psi^*(q')\psi(q)\mathrm{sinc}(\eta(q-q')) \underset{\sigma\to 0}{\sim} 1 \\ &\int \mathrm{d}q\mathrm{d}q' \frac{1}{2\sigma\sqrt{\pi}} e^{-\frac{(q-q')^2}{4\sigma^2}} \psi(q)\psi^*(q') \underset{\sigma\to 0}{\sim} 1. \end{aligned} \tag{11.74}$$

Thus the probability of obtaining the outcome $p_k = 0$ becomes dominated by the pure state contribution, and is determined by the expression:

$$\mathrm{Prob}[k=0] \underset{\sigma\to 0}{\sim} \mathrm{Prob}^{(1)}[k=0] \underset{\sigma\to 0}{\sim} \frac{2\eta\sigma}{\sqrt{\pi}}. \tag{11.75}$$

We notice that this probability is given as a function of the squeezed state variance σ. Eq. (11.75) ensures that the post-selection probability is non-zero, a necessary requirement to define it properly. This probability also needs to satisfy

$$\mathrm{Prob}[k=0] \gtrsim \frac{1}{2^n}. \tag{11.76}$$

This exponentially low probability is still compatible with the definition of post-selected class as explained in 2.2.3.

Dealing with errors

These sources of noise can in principle spoil the result of the computation and reduce the power of the post-selected version of the circuit family, thereby preventing from achieving arbitrary Post-BQP computations and spoiling the proof of hardness structure. How to solve this problem?

The proof of hardness of CV-IQP with input finite-squeezing states makes use of GKP states. In Ref. [Douce et al., 2017] it was assumed that input GKP states were present at the input. The universal set of CV operations given in Eq.(11.62) clearly includes a universal set of DV operations in GKP encoding:

$$\left\{ \bar{Z} = e^{i\hat{q}\sqrt{\pi}}, \bar{C_Z} = e^{i\hat{q}_1\hat{q}_2}, \bar{T} = e^{i\frac{\pi}{4}\left[2\left(\frac{\hat{q}}{\sqrt{\pi}}\right)^3 + \left(\frac{\hat{q}}{\sqrt{\pi}}\right)^2 - 2\frac{\hat{q}}{\sqrt{\pi}}\right]} \right\}, \tag{11.77}$$

plus the Hadamard gate which in GKP encoding corresponds to the Fourier-transform, that hence is obtained with post-selection:

$$\left\{ \bar{H} = F \right\}. \tag{11.78}$$

Therefore, any post-selected BQP computation can be simulated with a post-selected instance of IQP circuits, i.e. Post-CVIQP = PostBQP. In other words, for every computation in Post-BQP we can find a CV-IQP circuit that, with post selection corresponding to a non zero probability (and consistent with the definition of the Post-BQP class), simulates that computation, despite the finite input squeezing and finite resolution.

However, it has later been shown in Ref. [Douce et al., 2019] that the circuit family is hard-to-sample even without input GKP states. The trick is to show that generation of GKP states can be subsumed in the gates of the circuit itself. The technicalities of that work are out of the scope of this lecture.

11.5 Quantum annealing

So far, in the Continuous-Variable quantum computing Chapter but in general in this course notes, we had in mind a "device-independent formalism", that could be applied to any physical system modeled by the quantized harmonic oscillator (for this Chapter), or by a two-level quantum system (for the qubit part). In this section instead, we are going to focus on a specific implementation of a Continuous-Variable Quantum annealer, that is implemented with microwave cavities (our quantized harmonic oscillator), coupled to superconducting quantum circuits allowing to modify the quantum state of the field in the cavity. The model that we present here is drawn from Ref. [Puri et al., 2017]. The full Section is taken from Ref. [Vikstål, 2018].

The heart of our continuous variable quantum annealer device is the two-photon pumped *Kerr-Nonlinear Resonator* (KNR). It consists of a superconducting resonator made nonlinear with a Kerr-type nonlinearity, that causes the energy levels of the resonator to be non-equidistant. In this Section we begin with a brief overview of circuit QED and superconductivity. After that, we will introduce the two-photon pumped KNR in more detail. This device will allow encoding a combinatorial optimization problem in a network of two-photon pumped KNR's where a single Ising spin is mapped to two quasi-orthogonal coherent states with opposite phases. The latter constitute a two-fold degenerate eigenspace for the two-photon pumped KNR in the rotating frame. Furthermore, we will demonstrate how the AQC Hamiltonian Eq. (7.1) is realized by adiabatically controlling the frequency and amplitude of the pumps, and how N linearly coupled KNRs that are initialized to vacuum $|0\rangle$ at $t = 0$, will evolve to the ground state of an Ising Hamiltonian at $t = \tau$.

11.5.1 Circuit QED

Superconductivity is a quantum phenomenon that appears when certain metals are cooled down below a critical temperature. There, the metals become superconducting and a current can flow without electrical resistance [Tinkham, 2004]. *Circuit quantum electrodynamics*, or simply circuit QED, is based on superconducting electrical circuits. Similar to cavity QED, which studies atoms coupled to a single mode of the electromagnetic field inside a cavity, circuit QED studies on-chip implementation of QED using microwave photons. In circuit QED a superconducting coplanar waveguide between two capacitors plays the role of the cavity and the atom is replaced with an artificial one. Fig. 11.10 to the left illustrates a simple cavity that uses two mirrors facing others each to confine photons. The illustration to the right depicts the corresponding circuit QED implementation.

While photons do not interact with each other in vacuum an effective photon-photon interaction can be engineered by the use of non-linear materials. One such effective interaction is called the *Kerr effect*, which happens when a material has a refractive index that is proportional to the intensity of the electromagnetic field. A resonator can be made nonlinear with the use of a *Josephson junction* that introduces a Kerr-type nonlinearity. A Josephson junction consists of two superconducting metals separated by a weak link, that is made of a thin insulating barrier. For our purpose the Josephson junction can simply be viewed as a dissipationless nonlinear inductor.

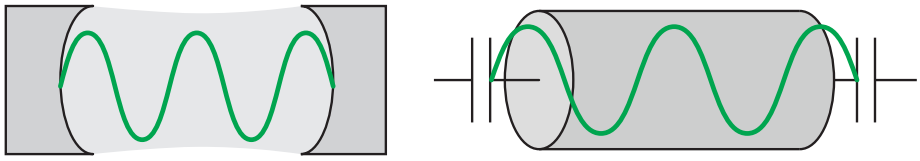


Figure 11.10: Left: Sketch of a cavity that confines photons between two mirrors. Right: Sketch of a superconducting resonator where microwave photons are confined between two capacitors that act as semi-transparent mirrors. Inspiration taken from Ida-Maria Svensson PhD Thesis (Chalmers, 2018).

By connecting two Josephson junctions in a superconducting loop we get what is called a *Superconducting QUantum Interference Device* (SQUID). SQUIDs are very sensitive to magnetic fields and are therefore mainly used as magnetometers for various purposes. However the SQUID can also be used as a two-photon pump if a time-varying magnetic flux is threaded through the superconducting loop at twice the resonator frequency. The SQUID will be an important component in the physical implementation of the continuous variable quantum annealer.

11.5.2 Two-photon pumped Kerr-nonlinear resonator

We now consider the proposed physical implementation of a continuous variable quantum annealer that was developed by Puri *et al* [Puri et al., 2017]. As already mentioned, the main component of the quantum annealer consists of a two-photon pumped KNR. A two-photon pumped KNR can be engineered by embedding a SQUID in the middle of a half-wavelength ($\lambda/2$) resonator, see Fig. 11.11. By modulating the magnetic flux Φ_{ext} through the SQUID at twice the resonator frequency the SQUID can be used as a parametric two-photon pump. The Hamiltonian of a two-photon pumped KNR can be derived by expanding the Josephson potential of the SQUID up to fourth order and doing the rotating wave approximation, see Appendix B.1. It results in[6]

$$\hat{\mathcal{H}}(t) = \omega_r \hat{a}^\dagger \hat{a} - K\hat{a}^{2\dagger}\hat{a}^2 + G\left(\hat{a}^{\dagger 2} e^{-2i\omega_r t} + \hat{a}^2 e^{2i\omega_r t}\right)$$

where ω_r denotes the mode frequency of the resonator, K denotes the amplitude of the Kerr-nonlinearity, G denotes the amplitude of the two-photon pump, and the two-photon pump frequency is set to twice the resonator frequency. The time-dependence of the Hamiltonian can be removed by transforming to a frame rotating at the resonator frequency. Indeed, by doing the following unitary transformation $\hat{\mathcal{U}}(t) = e^{i\omega_r t \hat{a}^\dagger \hat{a}}$,

[6] From here forth we will use natural units where $\hbar = c = 1$.

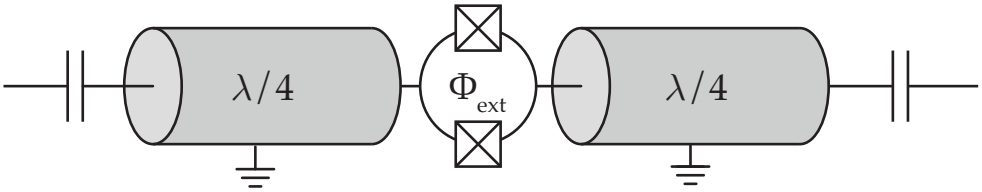


Figure 11.11: Schematic illustration of a two-photon pumped KNR: it consists of a SQUID sandwiched between two quarter wavelength ($\lambda/4$) resonators. The SQUID consists of two Josephson junctions connected in a superconducting loop. The Josephson junctions are drawn as square boxes with a cross in the figure. The loop is furthermore penetrated by a magnetic flux Φ_{ext} and the ends of the resonators are capacitively coupled to a transmission line.

we get

$$\hat{\mathcal{H}} = -K\hat{a}^{2\dagger}\hat{a}^2 + G\left(\hat{a}^{\dagger 2} + \hat{a}^2\right) = -K\left(\hat{a}^{\dagger 2} - \frac{G}{K}\right)\left(\hat{a}^2 - \frac{G}{K}\right) + \frac{G^2}{K}. \tag{11.79}$$

From the last expression it is evident that the two coherent states $|\pm\alpha\rangle$ with $\alpha = \sqrt{G/K}$ are are two degenerate eigenstates of this Hamiltonian with eigenenergy

$$\hat{\mathcal{H}}\,|\pm\alpha\rangle = \frac{G^2}{K}\,|\pm\alpha\rangle\,.$$

Following Puri *et al.* [Puri et al., 2017] we now take advantage of this well-defined two state subspace and choose to encode the logical spins $|\bar{0}\rangle$ and $|\bar{1}\rangle$ onto these coherent states, i.e. we do the mapping $\{|\bar{0}\rangle, |\bar{1}\rangle\} \to \{|-\sqrt{G/K}\rangle, |\sqrt{G/K}\rangle\}$, where the bar is used to distinguish the logical spin states from the vacuum and single-photon Fock state. For sufficiently large $|\alpha|$ the states can be considered orthogonal, indeed following Eq. (11.5) we have that

$$|\langle\bar{1}|\bar{0}\rangle|^2 = e^{-4|\alpha|^2}.$$

For instance if $|\alpha| = \sqrt{3}$, then $|\langle\bar{1}|\bar{0}\rangle|^2 \approx 10^{-6}$. Remarkably, these "code-words" are quite stable even in the presence of noise induced by single-photon losses from the resonator (which we will consider as the main loss mechanism). Lousely speaking, this is due to the fact that the coherent states are eigenstates of the loss operator $\hat{a}$.

In order to corroborate this fact, we start by showing that single-photon loss does not lead to spin-flip error, which is when a qubit flips e.g. $|0\rangle \to |1\rangle$. Consider the quantity $|\langle\bar{1}|\hat{a}|\bar{0}\rangle|^2$ that describes the overlap between the two logical spin states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ after a single-photon has been lost from the resonator. Using Eq. (11.4) and Eq. (11.5) we have that

$$|\langle\bar{1}|\hat{a}|\bar{0}\rangle|^2 = |\alpha|^2|\langle\bar{1}|\bar{0}\rangle|^2 = |\alpha|^2 e^{-4|\alpha|^2} \approx 10^{-5},$$

for $|\alpha| = \sqrt{3}$. Therefore we propose to use a two-photon pump amplitude set to $G = 3K$ so that $|\alpha| = \left|\sqrt{3K/K}\right| = \sqrt{3}$, as such we will use $G = 3K$ throughout this section.

More formally (beyond the scope of the lecture but given here as a complement), we compute the semi-classical steady-state solution of the KNR. In the presence of single-photon loss the collapse operator in the Lindblad-master equation takes the form $\hat{C} = \sqrt{\gamma}\hat{a}$ and the time-evolution of the density matrix is then given by

$$\frac{d\hat{\varrho}(t)}{dt} = -i\left[\hat{\mathcal{H}}, \hat{\varrho}\right] + \gamma\left(2\hat{a}\hat{\varrho}\hat{a}^\dagger - \hat{\varrho}\hat{a}^\dagger\hat{a} - \hat{a}^\dagger\hat{a}\hat{\varrho}\right).$$

This furthermore assumes that the system is in thermal equilibrium with a zero-temperature environment, so that the number of thermal photons in the system is negligible. This is a good approximation since the typical microwave cavity is at mK. The described process is schematically sketched in Fig. 11.12. While it is possible to obtain the steady-state solution analytically [Bartolo et al., 2016, Meaney et al., 2014], it can be found quite easily (see Appendix B.3) from the solution of the semi-classical equations of motion that the expected position of the states $|\pm\alpha\rangle$ in presence of single-photon loss are

$$\alpha = \frac{1}{2}\left(\frac{16G^2 - \gamma^2}{K^2}\right)^{1/4} \exp\left(-\frac{i}{2}\arctan\left(\frac{\gamma}{\sqrt{16G^2 - \gamma^2}}\right)\right).$$

Thus, if the two-photon pump amplitude is much larger than the single-photon loss rate, namely if $4G \gg \gamma$ or equivalently $12K \gg \gamma$, since $G = 3K$, then $\alpha \simeq \sqrt{G/K}$ and the mapping is confined to the spin subspace spanned by $|\bar{0}\rangle$ and $|\bar{1}\rangle$. Furthermore $K/30 > \gamma$ is already achievable today in superconducting

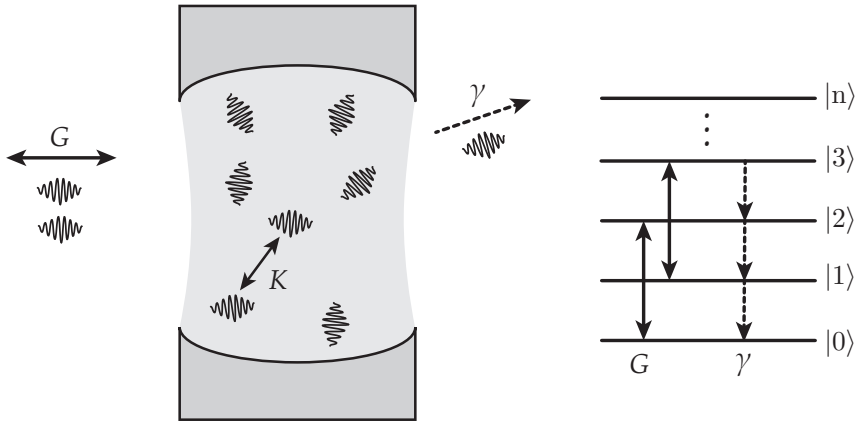


Figure 11.12: Left: Schematic illustration of a cavity with a Kerr-nonlinear element that introduces an effective photon-photon interaction of strength K. The cavity is subject to two-photon pumping with amplitude G and single-photon loss at a rate γ. Right: The aforementioned effects sketched on the Fock states $|n\rangle$.

resonators [Kirchmair et al., 2013]. When this condition is satisfied the steady-state density matrix of the system, $d\hat{\varrho}/dt = 0$, takes the form

$$\hat{\varrho}_{\text{ss}} \approx \frac{1}{2}\left(|\alpha\rangle\langle\alpha| + |-\alpha\rangle\langle-\alpha|\right),$$

which is an equal weighted statistical mixture of the two coherent states $|\pm\alpha\rangle$ or equivalently of the two logical spin states $\{|\bar{0}\rangle, |\bar{1}\rangle\}$. This can furthermore be confirmed by numerical integration of the master equation Eq. (11.5.2). The numerical results show that the mapping remains robust in presence of single-photon loss as long as $12K \gg \gamma$. Figure 11.13 shows the density plot of the steady-state Wigner function for $\gamma = 0.12K$ and $\gamma = 6K$, respectively. When $\gamma = 0.12K$ the fidelity is $\mathcal{F} = 99.99\%$ with respect to the ideal steady state Eq. (11.5.2). When $\gamma = 6K$, instead, outside the regime of low photon loss, the fidelity drops to $\mathcal{F} = 85.13\%$.

11.5.3 Two- & one-photon pumped Kerr-nonlinear resonator

We now turn our attention to the realization of the Ising Hamiltonian Eq. (8.1). We will demonstrate that a sufficiently weak single-photon pump can act like an effective magnetic field [Puri et al., 2017]. The single-photon pump can be physically implemented by capacitively coupling one end of the two-photon pumped KNR to a transmission line through which the single-photons are added to the resonator, see Figure 11.14. By considering the addition of a weak single-photon pump with a frequency that is in resonance with the resonator we get in a frame rotating at the resonator frequency that the the Hamiltonian is

$$\hat{\mathcal{H}}_1 = -K\hat{a}^{\dagger 2}\hat{a}^2 + G\left(\hat{a}^{\dagger 2} + \hat{a}^2\right) + F\left(\hat{a}^\dagger + \hat{a}\right), \tag{11.80}$$

where F denotes the amplitude of the single-photon pump. With this additional pump it can be once again found (see Appendix B.4) from the solution to the semi-classical equations of motion (neglecting losses) that the expected position of the coherent states $|\pm\alpha\rangle$ of the resonator are

$$\pm\alpha \simeq \pm\sqrt{\frac{G}{K} + \frac{F}{4G}},$$

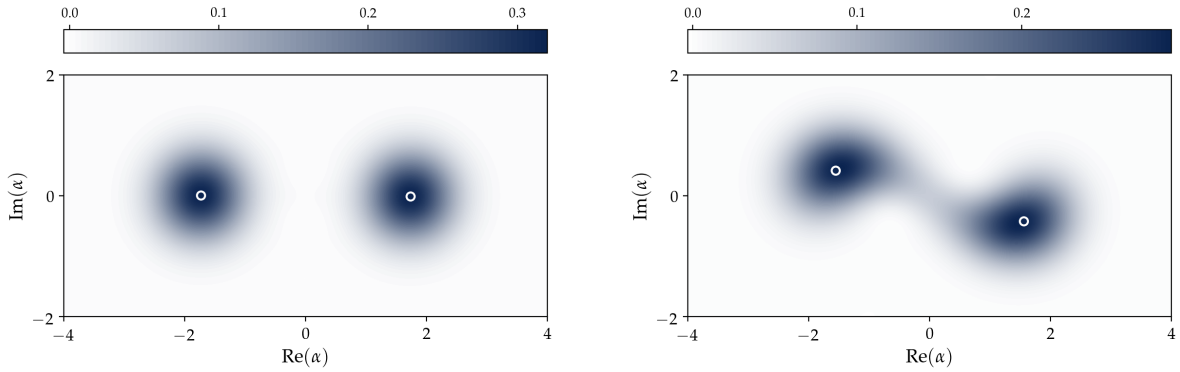


Figure 11.13: Steady state of a two-photon driven KNR in presence of single-photon loss. Figure (a) is the steady state phase space density plot of the Wigner function with a photon loss rate $\gamma = 0.12K$ and figure (b) with $\gamma = 6K$. The two-photon drive amplitude was set to $G = 3K$ in both figures. The white circles indicate the expected position of the coherent states. On the axes, $\mathrm{Im}(\alpha)$ and $\mathrm{Re}(\alpha)$ are the momentum and position quadrature respectively.

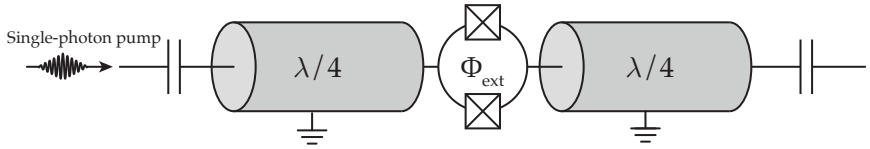


Figure 11.14: The two- and one-photon pumped KNR. One end of the two-photon pumped KNR is capacitively coupled to a transmission line through which microwave pulses of amplitude amplitude F and frequency ω_r drive the resonator.

i.e. they are slightly displaced by a factor $F/4G$, but if $4G \gg F$ or equivalently $12K \gg F$, then $\pm\alpha \simeq \pm\sqrt{G/K}$, and the states are once again confined to the spin subspace spanned by $|\bar{0}\rangle$ and $|\bar{1}\rangle$. Most importantly, this single-photon pump lifts the degeneracy between the states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ by an amount

$$\Delta E = \langle\alpha|\hat{\mathcal{H}}_1|\alpha\rangle - \langle-\alpha|\hat{\mathcal{H}}_1|-\alpha\rangle = 4F\alpha.$$

In the spin subspace spanned by our two logical spins $|\bar{0}\rangle$ and $|\bar{1}\rangle$ the Hamiltonian of Eq. (11.80) can therefore be expressed as

$$\bar{\hat{I}}\hat{\mathcal{H}}_1\bar{\hat{I}} = (|\bar{0}\rangle\langle\bar{0}| + |\bar{1}\rangle\langle\bar{1}|)\left(-K\hat{a}^{\dagger 2}\hat{a}^2 + G\left(\hat{a}^{\dagger 2} + \hat{a}^2\right) + F\left(\hat{a}^\dagger + \hat{a}\right)\right)(|\bar{0}\rangle\langle\bar{0}| + |\bar{1}\rangle\langle\bar{1}|)$$

using the definition of a coherent state $\hat{a}\,|\bar{0}/\bar{1}\rangle = \mp\alpha\,|\bar{0}/\bar{1}\rangle$ and the quasi-orthogonality condition $\langle\bar{1}|\bar{0}\rangle \approx 0$, we get

$$\begin{aligned}\bar{\hat{I}}\hat{\mathcal{H}}_1\bar{\hat{I}} &= (-K\alpha^4 + 2G\alpha^2)\bar{\hat{I}} + 2F\alpha\bar{\hat{\sigma}}^z \\ &= 2F\alpha\bar{\hat{\sigma}}^z + \text{const}\end{aligned}$$

where $\bar{\hat{\sigma}}^z = |\bar{1}\rangle\langle\bar{1}| - |\bar{0}\rangle\langle\bar{0}|$ is the Pauli z-matrix. This is the Ising Hamiltonian for a single Ising spin in a magnetic field of strength $|2F\alpha|$. Therefore the application of a weak single-photon drive induces an effective magnetic field.

Now, recall from Sec.7.4 that we require an initial Hamiltonian that does not commute with the final Hamiltonian and furthermore should have a simple non-degenerate ground state. We consider the case when the resonators are initialized to vacuum, since it is relative simple to prepare the resonators in such a state. We choose an initial Hamiltonian in the rotating frame of the following form

$$\hat{\mathcal{H}}_0 = -\delta \hat{a}^\dagger \hat{a} - K \hat{a}^{\dagger 2} \hat{a}^2 \tag{11.81}$$

where δ is a finite positive detuning to separate the vacuum state $|0\rangle$ from the single-photon Fock state $|1\rangle$ by an energy gap δ. At a first glance Eq. (11.81) might fill us with fear and trepidation since the Hamiltonian isn't bounded from below. Fear not! One can show that this is a consequence of the approximations used in the derivation, namely that we truncated the expansion of the Josephson cosine potential to fourth order, see Appendix B.1, and that the microscopic Hamiltonian from which it was derived from Eq. (B.5) does not exhibit this property. We also have to remember that photon loss also naturally leads to lower photon number states. Therefore as pointed out by Nigg *et al.* (2016) [Nigg et al., 2017] we have to think of the zero photon state in the rotating frame as the state with highest energy instead of the one with lowest energy. We also adopt the terminology by Nigg *et al.* and refer to the ground state, the state with highest energy, as the *roof state*.

The time-dependent Hamiltonian for the QA algorithm is obtained by adiabatically varying the two- and single-photon pump amplitudes and frequencies (see Appendix B.2)

$$\begin{aligned} \hat{\mathcal{H}}(t) &= \left(1 - \frac{t}{\tau}\right) \hat{\mathcal{H}}_0 + \frac{t}{\tau} \hat{\mathcal{H}}_1 \\ &= \left(1 - \frac{t}{\tau}\right) \left(-\delta \hat{a}^\dagger \hat{a} - K \hat{a}^{\dagger 2} \hat{a}^2\right) + \frac{t}{\tau} \left(-K \hat{a}^{\dagger 2} \hat{a}^2 + G\left(\hat{a}^{\dagger 2} + \hat{a}^2\right) + F\left(\hat{a}^\dagger + \hat{a}\right)\right), \end{aligned} \tag{11.82}$$

yielding Eq. (7.1) for the single Ising spin in a magnetic field problem. Notice that the Kerr term in the Hamiltonian above is actually time independent so that one only needs to vary the frequency and amplitude of the pumps.

11.5.4 Coupled two-photon pumped Kerr-nonlinear resonators

After having demonstrated that a weak single-photon pump induces an effective magnetic field we will now explain how the coupling between spins is realized.

The coupling between two resonators can in the simplest form be expressed as

$$g(\hat{a}_1 \hat{a}_2^\dagger + \hat{a}_1^\dagger \hat{a}_2),$$

where g describes single-photon exchange rate between two capacitively coupled resonators. We will begin by considering the case of two capacitively coupled two-photon pumped KNRs as shown in Fig 11.15 and assume that the parameters K and G are identical for both resonators. We furthermore neglect the single-photon

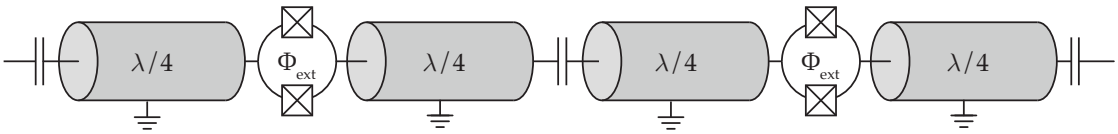


Figure 11.15: Two capacitively coupled KNRs.

pump for the moment. The Hamiltonian for two coupled KNRs in the rotating frame is given by

$$\hat{\mathcal{H}}_1 = \sum_{i=1}^{2} \left(-K \hat{a}_i^{\dagger 2} \hat{a}_i^2\right) + G\left(\hat{a}_i^{\dagger 2} + \hat{a}_i^2\right) + g\left(\hat{a}_1^\dagger \hat{a}_2 + \hat{a}_1 \hat{a}_2^\dagger\right).$$

A steady state analysis shows that for sufficiently small coupling strengths $g \ll 6K$ the two-photon pumped KNR is approximately kept in the subspace spanned by $|\bar{0},\bar{0}\rangle$, $|\bar{0},\bar{1}\rangle$, $|\bar{1},\bar{0}\rangle$ and $|\bar{1},\bar{1}\rangle$ (see Appendix B.5). Similar to the previous section it can be shown that in the spin subspace spanned by the logical spins $\{|\bar{0},\bar{0}\rangle, |\bar{0},\bar{1}\rangle, |\bar{1},\bar{0}\rangle, |\bar{1},\bar{1}\rangle\}$ this Hamiltonian takes the form

$$\hat{\mathcal{H}}_1 = 2g\alpha^2 \bar{\hat{\sigma}}_1^z \bar{\hat{\sigma}}_2^z + \text{const.}$$

This is the Hamiltonian for two magnetically coupled spins with $g > 0$ ($g < 0$) corresponding to the ferromagnetic (antiferromagnetic) coupling between the spins in the rotating frame. The initial Hamiltonian for two coupled KNRs is chosen to be:

$$\hat{\mathcal{H}}_0 = \sum_{i=1}^{2} \left(-\delta \hat{a}_i^\dagger \hat{a}_i - K \hat{a}_i^{\dagger 2} \hat{a}_i^2 \right) + g \left(\hat{a}_1^\dagger \hat{a}_2 + \hat{a}_1 \hat{a}_2^\dagger \right).$$

Furthermore, in order to ensure that the vacuum state $|0,0\rangle$ is the roof state of the initial Hamiltonian $\hat{\mathcal{H}}_0$ the detuning has to be greater than the single-photon exchange rate $\delta > g$, which can be seen by looking at the eigenstate following the roof state

$$\hat{\mathcal{H}}_0 \left(\frac{|1,0\rangle + |0,1\rangle}{\sqrt{2}} \right) = (g - \delta) \left(\frac{|1,0\rangle + |0,1\rangle}{\sqrt{2}} \right).$$

The two-spin Ising problem is then realized with the time-dependent Hamiltonian

$$\hat{\mathcal{H}}(t) = \left(1 - \frac{t}{\tau} \right) \hat{\mathcal{H}}_0 + \frac{t}{\tau} \hat{\mathcal{H}}_1.$$

What is interesting about the two-spin Ising problem is that the ground state exhibits frustration. This implies that the ground state for the two-spin problem is degenerate. At $t = \tau$ the two degenerate roof states for ferromagnetic and antiferromagnetic coupling are $\{|\bar{0},\bar{0}\rangle, |\bar{1},\bar{1}\rangle\}$ and $\{|\bar{0},\bar{1}\rangle, |\bar{1},\bar{0}\rangle\}$, respectively. It might seem like this can pose a problem since the gap closes between the two "highest" energy states, and therefore violates the adiabatic theorem. However, the theorem is not violated for the following reason. Recall from your introductory course in quantum mechanics that an observable A is conserved if its Hermitian operator $\hat{A}$ commutes with the Hamiltonian, i.e. $\left[\hat{A}, \hat{\mathcal{H}}\right] = 0$. It can be shown that the parity operator which is defined by $\hat{\mathcal{P}}_2 = \exp\left(i\pi \sum_{i=1}^{2} \hat{a}_i^\dagger \hat{a}_i \right)$ commutes with both the initial and final Hamiltonian, so that $\left[\hat{\mathcal{P}}_2, \hat{\mathcal{H}}(t)\right] = 0$. This means that parity is conserved and that transitions only occur between even-even and odd-odd parity eigenstates. So since the zero-photon Fock state $|0\rangle$ has even parity and the single-photon Fock states $|1\rangle$ has odd parity, degeneracy is not a problem.

We now possess the tools to write down the final Hamiltonian $\hat{\mathcal{H}}_1$ for N linearly coupled KNRs, which reads

$$\hat{\mathcal{H}}_1 = \sum_{i=1}^{N} \left(-K \hat{a}_i^{\dagger 2} \hat{a}_i^2 + G \left(\hat{a}_i^{\dagger 2} + \hat{a}_i^2 \right) + F_i \left(\hat{a}_i^\dagger + \hat{a}_i \right) \right) + \sum_{1 \le i < j \le N} g_{ij} \left(\hat{a}_i^\dagger \hat{a}_j + \hat{a}_j^\dagger \hat{a}_i \right), \tag{11.83}$$

where g_{ij} is the single-photon coupling strength between resonator i and j. In the computational basis the final Hamiltonian becomes

$$\hat{\mathcal{H}}_1 = 2\alpha^2 \sum_{1 \le i < j \le N} g_{ij} \bar{\hat{\sigma}}_i^z \bar{\hat{\sigma}}_j^z + 2\alpha \sum_{i=1}^{N} F_i \bar{\hat{\sigma}}_i^z + \text{const}, \tag{11.84}$$

which corresponds to the Ising Hamiltonian for N spins. The corresponding initial Hamiltonian can just as readily be written down

$$\hat{\mathcal{H}}_0 = \sum_{i=1}^{N} \left(-\delta \hat{a}_i^\dagger \hat{a}_i - K \hat{a}_i^{\dagger 2} \hat{a}_i^2 \right) + \sum_{1 \leq i < j \leq N} g_{ij} \left(\hat{a}_i^\dagger \hat{a}_j + \hat{a}_i \hat{a}_j^\dagger \right).$$

We now compare the couplings and the single-photon pumping strengths of the KNR Ising Hamiltonian Eq. (11.84) with the couplings and magnetic field strengths of the Ising Hamiltonian Eq. (8.2). This yields to the following correspondence for the couplings:

$$-2\alpha^2 g_{ij} \leftrightarrow J_{ij}. \tag{11.85}$$

The minus sign is introduced because $g_{ij} > 0$ ($g_{ij} < 0$) corresponds to ferromagnetic (antiferromagnetic) coupling while $J_{ij} > 0$ ($J_{ij} < 0$) corresponds to antiferromagnetic (ferromagnetic) coupling. Similar for the single-photon pumps we make the following correspondence:

$$2\alpha F_i \leftrightarrow h_i. \tag{11.86}$$

Lastly, the final component for this continuous variable quantum annealing architecture is the readout of the state of the computational states. This can be implemented through balanced homodyne detection that is enabled via capacitively coupled transmission lines [Puri et al., 2017], corresponding to the measurement of a suitably chosen field quadrature, namely $\hat{q} = \int dq |q\rangle\langle q|$.

11.5.5 Simulation of relevant combinatorial optimization problems

Subset sum problem

To demonstrate the capabilities of the two-photon pumped KNR architecture described in chapter 11.5, we consider a small instance of the subset sum problem that was introduced in section 8.2.1. We can consider the subset sum problem defined by the set $n = \{-2, 1, 2\}$ and $m = 3$. This problem requires three coupled KNRs and the Ising spin configuration that satisfy this problem is $s_1 = -1$ and $s_2 = s_3 = +1$. By dividing Eq. (11.85) with $-2\alpha^2$ and replacing J_{ij} with $n_i n_j/2$ we get that the coupling between the KNR's for the subset sum problem should be chosen as

$$g_{ij} = -A \frac{n_i n_j}{4\alpha^2}, \quad g_{ii} = 0.$$

Here A is a positive *scale factor* which will leave the Ising problem invariant and is convenient to introduce in order to satisfy the physical constraints on the coupling strengths. Next we divide Eq. (11.86) by 2α and replace h_i with Eq. (8.5), to find that the single-photon pump amplitudes should be chosen as

$$F_i = A \frac{1}{2\alpha} \left(\frac{1}{2} \sum_{j=1}^{N} n_j - m \right) n_i,$$

where A is the same scale factor. To make sure that the conditions $6K \gg g_{ij}$ and $12K \gg F_i$ are satisfied, we chose A to be

$$A = \frac{K}{2 \max|h_i|} = \frac{K}{10}.$$

It should be noted that the scale factor may vary depending on the given problem and the set of numbers in the problem. After having defined all the couplings and single-photon pump strengths we have diagonalized the time-dependent QA Hamiltonian

$$\hat{\mathcal{H}}(t) = \left(1 - \frac{t}{\tau} \right) \hat{\mathcal{H}}_0 + \frac{t}{\tau} \hat{\mathcal{H}}_1, \tag{11.87}$$

where the initial and final Hamiltonian are

$$\hat{\mathcal{H}}_1 = \sum_{i=1}^{3}\left(-K\hat{a}_i^{\dagger 2}\hat{a}_i^2 + G\left(\hat{a}_i^{\dagger 2} + \hat{a}_i^2\right) + F_i\left(\hat{a}_i^\dagger + \hat{a}_i\right)\right) + \sum_{1\le i<j\le 3} g_{ij}\left(\hat{a}_i^\dagger\hat{a}_j + \hat{a}_j^\dagger\hat{a}_i\right),$$

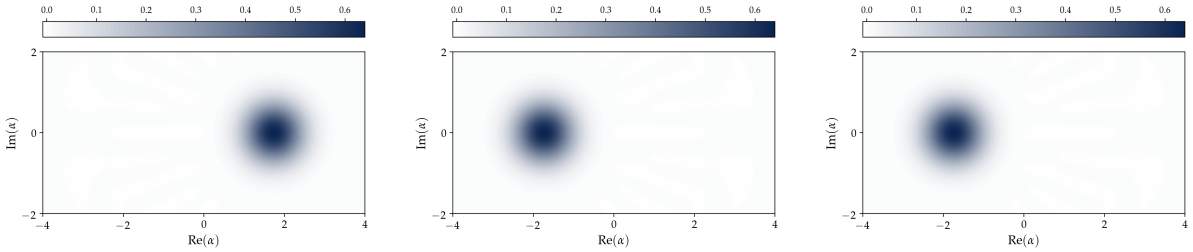
$$\hat{\mathcal{H}}_0 = \sum_{i=1}^{3}\left(-\delta\hat{a}_i^\dagger\hat{a}_i - K\hat{a}_i^{\dagger 2}\hat{a}_i^2\right) + \sum_{1\le i<j\le 3} g_{ij}\left(\hat{a}_i^\dagger\hat{a}_j + \hat{a}_i\hat{a}_j^\dagger\right).$$

In the spin subspace spanned by the computational basis the final Hamiltonian takes the form (dropping the constant)

$$\hat{\mathcal{H}}_1 = A\left(-\frac{1}{2}\sum_{1\le i<j\le 3} n_i n_j \bar{\hat{\sigma}}_i^z\bar{\hat{\sigma}}_j^z + \sum_{i=1}^{3}\left(\frac{1}{2}\sum_{j=1}^{3} n_j - m\right) n_i\bar{\hat{\sigma}}_i^z\right),$$

realizing the Ising Hamiltonian for the subset sum problem.

Pontus Vikstål has performed numerical simulations of this problem [Vikstål, 2018]. From diagonalizing the instantaneous QA Hamiltonian Eq. (11.87) he found that the minimum energy gap was $\Delta_{\min} = 0.046K$ and that the adiabatic condition was $\tau \gg 79/K$ which requires an evolution time that is approximately $\tau = 7900/K$. Next he initialized each individual resonator to the zero-photon Fock state $|0\rangle$, and numerically solved the Schrödinger equation with $\hat{\mathcal{H}}(t)$ given by Eq. (11.87). The success probability was then computed as the probability of occupation of the correct roof state at $t = \tau$, that is, $\langle\bar{1},\bar{0},\bar{0}|\hat{\varrho}(\tau)|\bar{1},\bar{0},\bar{0}\rangle$, which corresponds to the solution $s_1 = -1$ and $s_2 = s_3 = +1$. The calculated success probability was found to be 99.99% which shows that the evolution is to a very good approximation restricted to the computational subspace and that the KNR annealer correctly computes the solution of the subset sum problem. Fig. 11.16 shows the Wigner function at the end of the evolution for each individual resonator.



(a) First KNR (b) Second KNR (c) Third KNR

Figure 11.16: The Wigner function for each individual KNR at $t = \tau$.

Number partitioning problem

To map the number partitioning problem onto a network of two-photon pumped KNRs we begin by comparing the KNR Ising Hamiltonian Eq. (11.84) with the number partitioning problem Ising Hamiltonian Eq. (8.8). The comparison shows that the single-photon exchange rate coupling between i:th and j:th resonator should be chosen as

$$g_{ij} = -A\frac{n_i n_j}{\alpha^2}, \quad g_{ii} = 0.$$

Since this problem does not require any additional single-photon pumps, we can safely include α^2 that is in the denominator inside the scale factor A, so that $g_{ij} = -An_in_j$. We considered the number partitioning problem defined by the set $n = \{3, 5, 8\}$. This problem also requires three resonators coupled together. The

fair partitioning to this problem is $\mathcal{S}_1 = \{3, 5\}$ and $\mathcal{S}_2 = \{8\}$ for which the Ising spin configurations that satisfy this problem are $s_1 = s_2 = +1$ and $s_3 = -1$ and the configuration with all spins flipped. We chose $A = 0.0015K$, to satisfy $6K \gg g_{ij}$. The initial and final Hamiltonian is given by

$$\hat{\mathcal{H}}_1 = \sum_{i=1}^{3} \left(-K\hat{a}_i^{\dagger 2}\hat{a}_i^2 + G\left(\hat{a}_i^{\dagger 2} + \hat{a}_i^2\right)\right) + \sum_{1\le i<j\le 3} g_{ij}\left(\hat{a}_i^\dagger \hat{a}_j + \hat{a}_j^\dagger \hat{a}_i\right),$$

$$\hat{\mathcal{H}}_0 = \sum_{i=1}^{3} \left(-\delta\hat{a}_i^\dagger \hat{a}_i - K\hat{a}_i^{\dagger 2}\hat{a}_i^2\right) + \sum_{1\le i<j\le 3} g_{ij}\left(\hat{a}_i^\dagger \hat{a}_j + \hat{a}_i \hat{a}_j^\dagger\right).$$

In the spin subspace spanned by the coherent eigenstates of the two-photon pumped KNR in the rotating frame the final Hamiltonian takes the form

$$\hat{\mathcal{H}}_1 = -A \sum_{1\le i<j\le 3} n_i n_j \bar{\hat{\sigma}}_i^z \bar{\hat{\sigma}}_j^z,$$

realizing the Ising Hamiltonian for the number partitioning problem. From diagonalizing the time-dependent QA Hamiltonian we found that the minimum energy gap was $\Delta_{\min} = 0.3K$ and that the adiabatic condition was $\tau \gg 3.92/K$, which requires an evolution time that is approximately $\tau = 392/K$. Each resonator was then initialized to the vacuum $|0\rangle$ and we numerically solved the Schrödinger equation. From this the success probability of reaching the desired state $|\bar{0}, \bar{0}, \bar{1}\rangle$ or $|\bar{1}, \bar{1}, \bar{0}\rangle$ was found to be 99.97%. Furthermore a calculation of the fidelity showed that the system reaches the entangled state $\mathcal{N}\left(|\bar{0}, \bar{0}, \bar{1}\rangle + |\bar{1}, \bar{1}, \bar{0}\rangle\right)$, where $\mathcal{N} = 1/\sqrt{2\left(1 + \exp(-6|\alpha|^2)\right)}$ is a normalization constant, with 100% fidelity.

Discussion

The numerical results that we have obtained show that in the ideal case with no losses the success probability was 99.98% for the subset sum problem and 99.97% for the number partitioning problem. The $0.02\% - 0.03\%$ error mostly comes from deviations from the computational subspace. A higher success probability could have been achieved by choosing a smaller scale factor since then the conditions $6K \gg g_{ij}$ and $12K \gg F_i$ would have been better satisfied. However, choosing a smaller scale factor decreases the minimum gap energy which leads to an increase in the evolution time of the algorithm. For example choosing $A = 0.00025K$ for the number partitioning problem we get a success probability that is 100%, but the evolution time increases approximately by a factor of 6.

11.5.6 Remarks on scalability & the model

The results that we have presented confirm that the KNR architecture is a promising way of implementing quantum annealing. An important question that we didn't address though, is the scalability of this model. The two-photon pumped KNR architecture could be scaled up using pairwise couplings between resonators only [Puri et al., 2017]. However, a combinatorial optimization requires in general long-range interactions between spins. Indeed, the coupling terms that appear in the problems are $\propto n_i n_j$, and does so that the we require an all-to-all connectivity. It is usually difficult to implement experimentally such long-range interactions. One possible solution would be to embed the Ising spins in a so-called Chimera graph, as it is done in the D-wave system [Choi, 2008, Choi, 2011]. Another possibility is to embed the Ising spins in a LHZ scheme which maps the Ising problem on a graph with local interactions only [Lechner et al., 2015]. In the LHZ scheme N logical spins (the spins that define the optimization problem) are mapped to $M = N(N-1)/2$ physical spins (the spins available the lab). This was furthermore investigated by Puri *et*

al. (2016) [Puri et al., 2017] where they showed how the LHZ scheme for two-photon pumped KNRs could be physically realized. A third connectivity solution was presented by Nigg *et al.* (2016) [Nigg et al., 2017] where they showed that by leveraging the phenomenon of flux quantization for a network of Kerr-parametric oscillators they could achieve an all-to-all connected architecture.

Appendix A

Quantization of the electromagnetic field in a cavity

This Appendix is taken from Ref. [Vikstål, 2018].

A.1 Quantizing the electromagnetic field

In this section we will consider the quantization of a single-mode electromagnetic field following Ref. [Gerry et al., 2005, Meystre and Sargent, 2007]. Note that the dimensions in this Appendix are different than those used in the main text, resulting in a different definition of the field quadratures. To quantize the electromagnetic field we will begin by considering a closed cavity of volume V with mirrors of perfect reflection located at $z = 0$ and $z = L$. We imagine that we have a monochromatic, single-mode electromagnetic field that is assumed to be polarized along the x-direction.

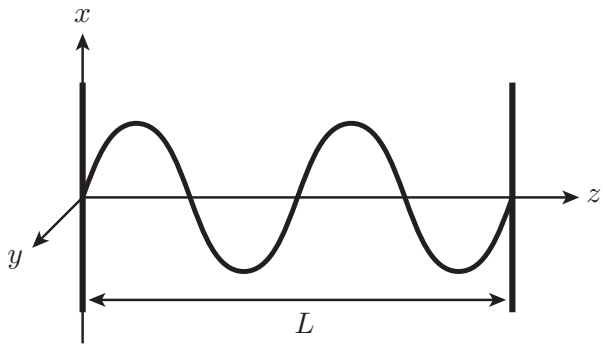


Figure A.1: Cavity with two perfectly reflecting mirrors located at $z = 0$ and $z = L$. The electric field is assumed to be polarized along the x-direction

In the absence of sources and charges the Maxwell equations read

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}, \tag{A.1}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \epsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}, \tag{A.2}$$

$$\nabla \cdot \boldsymbol{E} = 0, \tag{A.3}$$

$$\nabla \cdot \boldsymbol{B} = 0. \tag{A.4}$$

Using Maxwell's equations in the absence of sources and charges and the given boundary conditions, the electric field has the form

$$\boldsymbol{E}(z,t) = \Big(\frac{2\omega^2}{V\epsilon_0}\Big)^{1/2} q(t)\sin(kz)\boldsymbol{e}_x, \tag{A.5}$$

where V is the volume of the cavity, $q(t)$ is a function of time and $\boldsymbol{e}_x$ is the unit-vector along the x-direction. From the boundary conditions the allowed frequencies are found to be

$$\omega_n = \frac{c\pi n}{L}, \quad n = 1, 2, 3, \ldots$$

with $k_n = \omega_n/c$ as the corresponding wave number. In Eq. (A.5) we have assumed a specific frequency ω, i.e. a specific standing wave which is called a mode of the field. We find the corresponding magnetic field by substituting Eq. (A.5) into (A.2),

$$\boldsymbol{B}(z,t) = \frac{\mu_0\epsilon_0}{k}\Big(\frac{2\omega^2}{V\epsilon_0}\Big)^{1/2} \dot{q}(t)\cos(kz)\boldsymbol{e}_y. \tag{A.6}$$

We now identify $q(t)$ as a canonical coordinate, and $p(t) \equiv \dot{q}(t)$ as the momenta canonically conjugate to $q(t)$. The energy stored in the field of the single-mode is

$$\mathcal{H} = \frac{1}{2}\int_V \Big(\frac{1}{\epsilon_0}|\boldsymbol{E}|^2 + \mu_0|\boldsymbol{B}|^2\Big)dV = \frac{1}{2}(p^2 + \omega^2 q^2).$$

We see that this is nothing but the energy of a harmonic oscillator with unit mass[1]. To quantize the electromagnetic field we promote q and p to operators

$$q \to \hat{q} \quad \text{and} \quad p \to \hat{p},$$

and impose that they obey the canonical commutation relation

$$[\hat{q}, \hat{p}] = i\hbar.$$

Thus the electric and magnetic field are also promoted to operators

$$\hat{E}_x(z,t) = \Big(\frac{2\omega^2}{V\epsilon_0}\Big)^{1/2} \hat{q}\sin(kz), \tag{A.7}$$

$$\hat{B}_y(z,t) = \frac{\mu_0\epsilon_0}{k}\Big(\frac{2\omega^2}{V\epsilon_0}\Big)^{1/2} \hat{p}\cos(kz). \tag{A.8}$$

The subscript x and y denotes the constituent components of the fields. The Hamiltonian now reads

$$\hat{\mathcal{H}} = \frac{1}{2}(\hat{p}^2 + \omega^2\hat{q}^2). \tag{A.9}$$

[1]The amplitude of the electric field in Eq. (A.5) was cleverly chosen to yield the energy of a harmonic oscillator of unit mass.

Next we define the very useful non-Hermitian *ladder operators*

$$\hat{a} = (2\hbar\omega)^{-1/2}(\omega\hat{q} + i\hat{p}), \tag{A.10}$$

$$\hat{a}^\dagger = (2\hbar\omega)^{-1/2}(\omega\hat{q} - i\hat{p}). \tag{A.11}$$

which obey the boson commutation relation $[\hat{a}, \hat{a}^\dagger] = 1$. Inverting Eq. (A.10) and Eq. (A.11) and substituting it into Eq. (A.9) we obtain a Hamiltonian entirely written in terms of the ladder operators as in Eq.(11.1.1).

In the main text, we haven't discussed the time-dependence of the operators. What we've done so far is assumed to hold at some time t, for example $t = 0$. In the *Schrödinger picture* the states are time-dependent and the operators are time-independent. On the contrary, in the *Heisenberg picture* the operators are time-dependent and the states are time-independent. In the Heisenberg picture the time evolution of the annihilation operator is given by

$$\begin{aligned} \frac{d\hat{a}}{dt} &= \frac{i}{\hbar}[\hat{\mathcal{H}}, \hat{a}] \\ &= \frac{i}{\hbar}[\hbar\omega\Big(\hat{a}^\dagger\hat{a} + \frac{1}{2}), \hat{a}] \\ &= i\omega(\hat{a}^\dagger\hat{a}\hat{a} - \hat{a}\hat{a}^\dagger\hat{a}) \\ &= i\omega[\hat{a}, \hat{a}^\dagger]\hat{a} = -i\omega\hat{a}. \end{aligned} \tag{A.12}$$

Solving this equation yields

$$\hat{a}(t) = \hat{a}e^{-i\omega t}, \tag{A.13}$$

where $\hat{a}(0) \equiv \hat{a}$. Taking the Hermitian adjoint of Eq. (A.13) we also find that

$$\hat{a}^\dagger(t) = \hat{a}^\dagger e^{i\omega t}.$$

After substituting Eq. (A.10) into Eq. (A.7) and Eq. (A.11) into Eq. (A.8), the electric and magnetic field with the inclusion of the time-dependence become, respectively

$$\hat{E}_x(z,t) = \Big(\frac{\hbar\omega}{V\epsilon_0}\Big)^{1/2}(\hat{a}e^{-i\omega t} + \hat{a}^\dagger e^{i\omega t})\sin(kz), \tag{A.14}$$

$$\hat{B}_y(z,t) = \frac{\mu_0\epsilon_0}{ik}\Big(\frac{\hbar\omega^3}{V\epsilon_0}\Big)^{1/2}(\hat{a}e^{-i\omega t} - \hat{a}^\dagger e^{i\omega t})\cos(kz). \tag{A.15}$$

It is convenient to introduce the two hermitian operators

$$\begin{aligned} \hat{X} &= \sqrt{\frac{\omega}{2\hbar}}\hat{q} = \frac{1}{2}(\hat{a} + \hat{a}^\dagger), \\ \hat{X}_{\pi/2} &= \frac{1}{\sqrt{2\hbar\omega}}\hat{p} = \frac{1}{2i}(\hat{a} - \hat{a}^\dagger), \end{aligned} \tag{A.16}$$

which satisfy the commutation relation

$$[\hat{X}, \hat{X}_{\pi/2}] = \frac{i}{2}.$$

From here it is easy to show that the electric field operator Eq. (A.14) can be written in terms of the dimensionless quantities $\hat{X}$ and $\hat{X}_{\pi/2}$ as

$$\hat{E}_x(z,t) = 2\Big(\frac{\hbar\omega}{\epsilon_0 V}\Big)^{1/2}(\hat{X}\cos(\omega t) + \hat{X}_{\pi/2}\sin(\omega t))\sin(kz). \tag{A.17}$$

This expression shows that $\hat{X}$ and $\hat{X}_{\pi/2}$ are associated with the electric field amplitude, where the second term is offset by $\pi/2$ compared to the $\cos(\omega t)$ term.

Now, let's understand why coherent states are the most classical states. To construct a state with close resemblance to the classical electromagnetic field, one can observe that by replacing $\hat{a}$ and $\hat{a}^\dagger$ with a complex variable in Eq. (A.14) and Eq. (A.15) it would produce a "classical field", i.e. a field that oscillates. This is achieved in view of the definition of the coherent state as an eigenstate to the annihilation operator, as in Eq.(11.4). The expectation value of the electric field given by Eq. (A.14) becomes then

$$\begin{aligned}\langle \hat{E}_x(z,t)\rangle_\alpha = \langle\alpha|\,\hat{E}_x(z,t)\,|\alpha\rangle &= \left(\frac{\hbar\omega}{V\epsilon_0}\right)^{1/2}\langle\alpha|\,(\hat{a}e^{-i\omega t}+\hat{a}^\dagger e^{i\omega t})\,|\alpha\rangle\sin(kz)\\ &= \left(\frac{\hbar\omega}{V\epsilon_0}\right)^{1/2}(\alpha e^{-i\omega t}+\alpha^* e^{i\omega t})\sin(kz).\end{aligned}$$

Writing α in polar coordinates $\alpha = |\alpha|e^{i\varphi}$ we get

$$\langle \hat{E}_x(z,t)\rangle_\alpha = \left(\frac{\hbar\omega}{V\epsilon_0}\right)^{1/2} 2|\alpha|\cos(\omega t-\varphi)\sin(kz),$$

and we see that the field oscillates very much like the classical electric field. Likewise, for the quadrature operator Eq. (A.16), we have

$$\langle \hat{X}\rangle_\alpha = \frac{1}{2}\langle\alpha|\,(\hat{a}+\hat{a}^\dagger)\,|\alpha\rangle = \frac{1}{2}(\alpha+\alpha^*) = \mathrm{Re}\,\alpha = |\alpha|\cos(\varphi)$$

and similarly $\langle \hat{X}_{\pi/2}\rangle_\alpha = \mathrm{Im}\,\alpha = |\alpha|\sin(\varphi)$, so the mean of the quadratures are related to the real and imaginary part of α.

Let us now derive Eq.(11.1.1) of the main text. Since the Fock states form a complete set, we will use them to express α

$$|\alpha\rangle = \sum_{n=0}^{\infty} c_n\,|n\rangle\,, \tag{A.18}$$

where $c_n = \langle n|\alpha\rangle$ denotes a complex number which is to be determined. Inserting Eq. (A.18) into Eq. (11.4) and using Eq. (11.1) and Eq. (11.2) we obtain

$$\sum_{n=1}^{\infty} c_n\sqrt{n}\,|n-1\rangle = \sum_{n=0}^{\infty} c_n\alpha\,|n\rangle\,.$$

Since the Fock-states form an orthogonal basis, we can multiply with an arbitrary state $\langle m|$ from left and use the orthogonality condition $\langle m|n\rangle = \delta_{mn}$ to obtain

$$c_{m+1}\sqrt{m+1} = \alpha c_m.$$

By the substitution $m \to n-1$

$$c_n = \frac{\alpha}{\sqrt{n}}c_{n-1} = \frac{\alpha^2}{\sqrt{n(n-1)}}c_{n-2} = \ldots = \frac{\alpha^n}{\sqrt{n!}}c_0,$$

we obtain a recursion formula. Hence Eq. (A.18) can be expressed as

$$|\alpha\rangle = c_0\sum_{n=0}^{\infty}\frac{\alpha^n}{\sqrt{n!}}\,|n\rangle\,.$$

We use the normalization condition to determine the coefficient $|c_0|^2$,

$$|\langle\alpha|\alpha\rangle|^2 = 1 = |c_0|^2\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\frac{\alpha^{*m}\alpha^n}{\sqrt{m!}\sqrt{n!}}\langle m|n\rangle = |c_0|^2\sum_{n=0}^{\infty}\frac{|\alpha|^2}{n!} = |c_0|^2 e^{|\alpha|^2}.$$

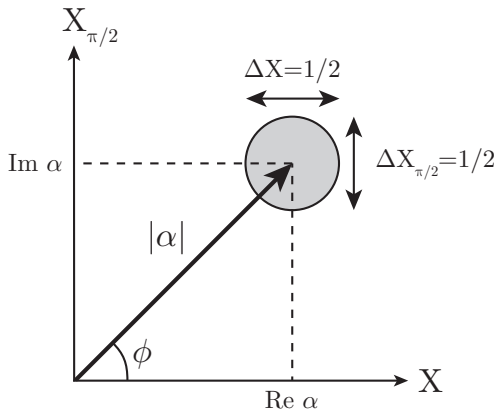


Figure A.2: Phase-space diagram of a coherent state $|\alpha\rangle$. The displacement from the origin is equal to $|\alpha|$ and the angle φ is the phase, measured from the X-axis. The quantum uncertainty is displayed as a grey circle with a diameter of 1/2.

Therefore $|c_0| = e^{-|\alpha|^2/2}$ and our final expression for the coherent state $|\alpha\rangle$ expressed in terms of Fock states is Eq.(11.1.1) in the main text.

A neat way to illustrate a coherent state is in *phase space*. In Fig. A.2 the phase space diagram of a coherent state $|\alpha\rangle$ is illustrated. The coherent state can be viewed as a displacement of the vacuum state with a distance $|\alpha|$ from the origin and an angle φ measured from the $\hat{X}$-axis. The grey circle area represents the uncertainty of the coherent state and has a constant diameter of 1/2.

Appendix B

Superconducting quantum circuits

B.1 Circuit Lagrangian

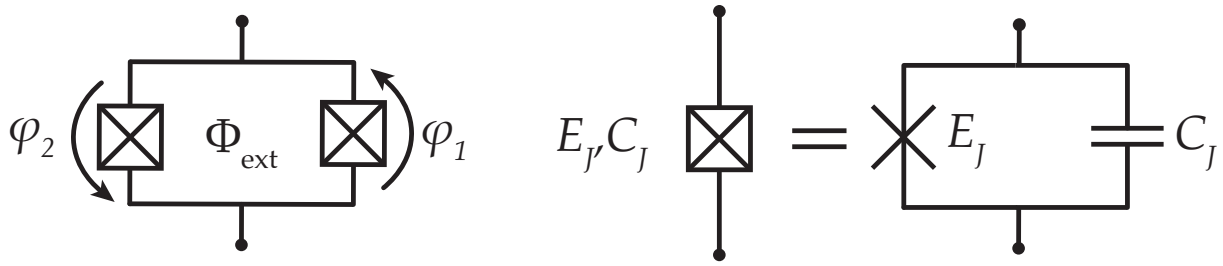


Figure B.1: Left: Schematic illustration of a SQUID. Right: Circuit symbol for a Josephson junction with Josephson energy E_J and capacitance C_J.

In this section of the appendix we aim to derive the two-photon pumped Kerr Hamiltonian starting from the circuit Lagrangian of a SQUID and follow the derivation presented by Nigg *et al.* [Nigg et al., 2017]. The SQUID consists of two Josephson junctions connected in a superconducting loop with a magnetic flux $\Phi_{\text{ext}}(t)$ penetrating the loop. Fig B.1 is meant to illustrate the SQUID and the corresponding circuit symbol of the Josephson junction. The circuit Lagrangian for a SQUID is given by

$$\mathcal{L} = \frac{1}{2}C_J\left(\dot{\phi}_1^2 + \dot{\phi}_2^2\right) + E_J\left(\cos(2\pi\phi_1/\Phi_0) + \cos(2\pi\phi_2/\Phi_0)\right) \tag{B.1}$$

where ϕ_i is the phase difference across the junction and $\Phi_0 = h/(2e)$ is the *magnetic flux quantum*. For convenience we chose units such that the flux quantum is $\Phi_0 = h/(2e) = 2\pi$. The parameters C_J and E_J in Eq. (B.1) denote the capacitance and Josephson energy of each Josephson junction. Due to the fluxoid quantization condition, ϕ_1 and ϕ_2 are not independent variables. If a flux Φ_{ext} is penetrating the loop, then the flux quantization requires that

$$\phi_2 - \phi_1 = \Phi_{\text{ext}}, \tag{B.2}$$

where we have set the geometrical phase term to zero and neglected the induced flux term which is reasonable for small loops. We introduce the new variable

$$\vartheta = \frac{1}{2}(\phi_1 + \phi_2).$$

which together with Eq. (B.2) implies

$$\phi_1 = \vartheta - \frac{1}{2}\Phi_{\text{ext}}, \tag{B.3}$$

$$\phi_2 = \vartheta + \frac{1}{2}\Phi_{\text{ext}}. \tag{B.4}$$

Inserting Eq. (B.3) and Eq. (B.4) into Eq. (B.1) and simplifying we get

$$\mathcal{L} = \frac{1}{2}C\dot{\vartheta}^2 + 2E_J \cos\left(\frac{\Phi_{\text{ext}}}{2}\right)\cos(\vartheta)$$

with $C = 2C_J$. We have also neglected the $\frac{1}{4}C_J\dot{\Phi}_{\text{ext}}^2$ term which is independent of coordinates and does not affect the dynamics of the system. The Hamiltonian function is obtained as usual through the Legendre transformation

$$\mathcal{H} = \sum_i Q_i\dot{\vartheta}_i - \mathcal{L},$$

with $Q_i = \dfrac{\partial\mathcal{L}}{\partial\dot{\vartheta}_i} = C\dot{\vartheta}_i$ being the generalized momenta. From the Legendre transformation we obtain the Hamiltonian

$$\mathcal{H} = E_C\frac{Q^2}{2e^2} - 2E_J\cos\left(\frac{\Phi_{\text{ext}}}{2}\right)\cos(\vartheta). \tag{B.5}$$

where $E_C \equiv e^2/C$ is the charging energy. Next, we consider a monochromatic magnetic flux modulation threaded through the loop that is of the form

$$\frac{\Phi_{\text{ext}}(t)}{2} = \Phi_{\text{dc}} + \delta\Phi(t)$$

where dc is the static magnetic flux and $\delta\Phi(t)$ is a time-dependent perturbation that we choose of the form

$$\delta\Phi(t) = \delta\Phi_{\text{ac}}\cos(\Omega t),$$

where $\delta\Phi_{\text{ac}}$ is the ac flux. The external flux cosine term in Eq. (B.5) can now be written as

$$\cos\left(\frac{\Phi_{\text{ext}}(t)}{2}\right) = \cos(\Phi_{\text{dc}})\cos(\delta\Phi_{\text{ac}}\cos(\Omega t)) - \sin(\Phi_{\text{dc}})\sin(\delta\Phi_{\text{ac}}\cos(\Omega t)).$$

Using the following mathematical tricks

$$\cos(z\cos(\vartheta)) = J_0(z) + 2\sum_{n=1}^{\infty}(-1)^n J_{2n}(z)\cos(2n\vartheta)$$

$$\sin(z\cos(\vartheta)) = 2\sum_{n=1}^{\infty}(-1)^n J_{2n-1}(z)\cos\left[(2n-1)\vartheta\right],$$

where $J_n(z)$ is the Bessel function of the first kind, we get

$$\cos\left(\frac{\Phi_{\text{ext}}(t)}{2}\right) = \cos(\Phi_{\text{dc}})\left(J_0(\delta\Phi_{\text{ac}}) + 2\sum_{n=1}^{\infty}(-1)^n J_{2n}(\delta\Phi_{\text{ac}})\cos(2n\Omega t)\right)$$
$$- \sin(\Phi_{\text{dc}})\left(2\sum_{n=1}^{\infty}(-1)^n J_{2n-1}(\delta\Phi_{\text{ac}})\cos\left[(2n-1)\Omega t\right]\right).$$

If the ac-flux is sufficiently small, $|\delta\Phi_{\rm ac}| \ll 1$, then we can keep only the two leading terms of the sum, that's J_0 and J_1 and use the approximations $J_0(z) \simeq 1$ and $J_1(z) \simeq z/2$ yielding

$$\cos\left(\frac{\Phi_{\rm ext}(t)}{2}\right) \simeq \cos(\Phi_{\rm dc}) + \sin(\Phi_{\rm dc})\delta\Phi_{\rm ac}\cos(\Omega t)\,.$$

For convenience we now separate the Hamiltonian into a time independent and time dependent part as $\mathcal{H}_0 = \mathcal{H}_1 + \mathcal{H}_2(t)$, where

$$\begin{aligned}\mathcal{H}_1 &= E_C\frac{Q^2}{2e^2} - 2E_J\cos(\Phi_{\rm dc})\cos(\vartheta)\\ \mathcal{H}_2(t) &= -2E_J\sin(\Phi_{\rm dc})\delta\Phi_{\rm ac}\cos(\Omega t)\cos(\vartheta).\end{aligned}$$

We now expand the $\cos(\vartheta)$ to fourth order, keeping only the leading nonlinearity. Furthermore since we focus on the weak ac modulation $|\delta\Phi_{\rm ac}| \ll 1$, we can neglect the nonlinear part of the time dependent term, then we have

$$\begin{aligned}\mathcal{H}_1 &\simeq E_C\frac{Q^2}{2e^2} + E_J\cos(\Phi_{\rm dc})\vartheta^2 - \frac{E_J}{12}\cos(\Phi_{\rm dc})\vartheta^4,\\ \mathcal{H}_2(t) &\simeq E_J\sin(\Phi_{\rm dc})\delta\Phi_{\rm ac}\cos(\Omega t)\,\vartheta^2,\end{aligned}$$

where we have dropped the coordinate independent terms that does not contribute to the dynamics. As the next step we promote ϑ and Q to operators with the commutation relation

$$\left[\hat{\vartheta}, \hat{Q}\right] = i.$$

By introducing the creation and annihilation operator $\hat{a}^\dagger$ and $\hat{a}$ that obey the bosonic commutation relation $\left[\hat{a}, \hat{a}^\dagger\right] = 1$, we have that

$$\hat{\vartheta} = \sqrt{\frac{1}{2Z}}\left(\hat{a} + \hat{a}^\dagger\right) \quad \text{and} \quad \hat{Q} = -i\sqrt{\frac{Z}{2}}\left(\hat{a} - \hat{a}^\dagger\right),$$

where

$$Z = \sqrt{\frac{2e^2E_J\cos(\Phi_{\rm dc})}{E_C}}.$$

Thus the Hamiltonian in terms of the creation and annihilation operator $\hat{a}^\dagger$ and $\hat{a}$ can be written as

$$\begin{aligned}\hat{\mathcal{H}}_1 &= \omega_r\hat{a}^\dagger\hat{a} - \frac{E_C}{96e^2}\left(\hat{a} + \hat{a}^\dagger\right)^4,\\ \hat{\mathcal{H}}_2(t) &= \frac{E_J}{2Z}\sin(\Phi_{\rm dc})\delta\Phi_{\rm ac}\cos(\Omega t)\left(\hat{a} + \hat{a}^\dagger\right)^2,\end{aligned}$$

where we have defined the resonator frequency

$$\omega_r = \sqrt{\frac{2E_CE_J\cos(\Phi_{\rm dc})}{e^2}},$$

and have furthermore dropped the $\omega_r/2$ which simply corresponds to a constant energy shift. Next we move to the interaction picture. To switch into the interaction picture we divide the Schrödinger picture Hamiltonian into two parts:

$$\hat{\mathcal{H}}_{\rm S} = \hat{\mathcal{H}}_{\rm S,0} + \hat{\mathcal{H}}_{\rm S,1}$$

where

$$\hat{\mathcal{H}}_{\rm S,0} = \omega_r\hat{a}^\dagger\hat{a}.$$

and

$$\hat{\mathcal{H}}_{\mathrm{S},1} = -\frac{E_C}{96e^2}\left(\hat{a}+\hat{a}^\dagger\right)^4 + \frac{E_J}{2Z}\sin(\Phi_{\mathrm{dc}})\delta\Phi_{\mathrm{ac}}\cos\left(\Omega t\right)\left(\hat{a}+\hat{a}^\dagger\right)^2.$$

The Hamiltonian in the interaction picture is defined as $\hat{\mathcal{H}}_{\mathrm{int}} = e^{i\hat{\mathcal{H}}_{\mathrm{S},0}t}\hat{\mathcal{H}}_{\mathrm{S}}e^{-i\hat{\mathcal{H}}_{\mathrm{S},0}t}$, and we thus get

$$\hat{\mathcal{H}}_{\mathrm{int}} = \omega_r\hat{a}^\dagger\hat{a} - \frac{e^{i\omega_r\hat{a}^\dagger\hat{a}t}\left(\hat{a}+\hat{a}^\dagger\right)^4 E_C e^{-i\omega_r\hat{a}^\dagger\hat{a}t}}{96e^2} + \frac{E_J}{2Z}\sin(\Phi_{\mathrm{dc}})\delta\Phi_{\mathrm{ac}}\cos\left(\Omega t\right)e^{i\omega_r\hat{a}^\dagger\hat{a}t}\left(\hat{a}+\hat{a}^\dagger\right)^2 e^{-i\omega_r\hat{a}^\dagger\hat{a}t}. \quad \text{(B.6)}$$

Next normal ordering the terms yields

$$:\left(\hat{a}+\hat{a}^\dagger\right)^4: = \hat{a}^{\dagger 4} + 6\hat{a}^{\dagger 2} + 4\hat{a}^{\dagger 3}\hat{a} + 6\hat{a}^{\dagger 2}\hat{a}^2 + 4\hat{a}^\dagger\hat{a}^3 + 12\hat{a}^\dagger\hat{a} + \hat{a}^4 + 6\hat{a}^2 + 3,$$
$$:\left(\hat{a}+\hat{a}^\dagger\right)^2: = \hat{a}^{\dagger 2} + 2\hat{a}^\dagger\hat{a} + \hat{a}^2 + 1.$$

Under the unitary transformation $\hat{\mathcal{U}}(t) = e^{-i\omega_r t\hat{a}^\dagger\hat{a}}$ each $\hat{a}$ and $\hat{a}^\dagger$ transform according to

$$e^{i\omega_r t\hat{a}^\dagger\hat{a}}\hat{a}e^{-i\omega_r t\hat{a}^\dagger\hat{a}} = \hat{a}e^{i\omega_r t},$$
$$e^{i\omega_r t\hat{a}^\dagger\hat{a}}\hat{a}^\dagger e^{-i\omega_r t\hat{a}^\dagger\hat{a}} = \hat{a}^\dagger e^{-i\omega_r t}.$$

Now we use the *rotating wave approximation* (RWA) which corresponds to neglecting all rotating terms proportional to $\exp(i\omega_r t)$. In the second term of Eq. (B.6) the only non-rotating terms are $6\hat{a}^{\dagger 2}\hat{a}^2$ and $12\hat{a}^\dagger\hat{a}$. The latter leads to a renormalization of the oscillator frequency by an amount $\Delta\omega_r = -E_C/(8e^2)$ which we absorb into ω_r and make an implicit redefinition of the resonator frequency. For the third and last term in Eq. (B.6) we specifically consider a two-photon pump frequency that is close to twice the resonator frequency, i.e. $\Omega \simeq 2\omega_r$. By writing $\cos\left(\Omega t\right)$ in Euler form we find that the only non-rotating terms are $\hat{a}^{\dagger 2}$ and $\hat{a}^2$. Putting it all together we have

$$\hat{\mathcal{H}}_{\mathrm{int}} = \omega_r\hat{a}^\dagger\hat{a} - \frac{E_C}{16e^2}\hat{a}^{\dagger 2}\hat{a}^2 + \frac{E_J}{4Z}\sin(\Phi_{\mathrm{dc}})\delta\Phi_{\mathrm{ac}}\left(\hat{a}^{\dagger 2}+\hat{a}^2\right).$$

Switching back to the Schrödinger picture we get

$$\hat{\mathcal{H}}(t) = \omega_r\hat{a}^\dagger\hat{a} - K\hat{a}^{\dagger 2}\hat{a}^2 + G\left(\hat{a}^{\dagger 2}e^{-2i\omega_r t} + \hat{a}^2 e^{2i\omega_r t}\right) \quad \text{(B.7)}$$

which is the two-photon pumped Kerr-nonlinear resonator Hamiltonian, where we have defined the Kerr-nonlinear amplitude as

$$K \equiv \frac{E_C}{16e^2},$$

and the two-photon pump amplitude as

$$G \equiv \frac{E_J}{4Z}\sin(\Phi_{\mathrm{dc}})\delta\Phi_{\mathrm{ac}}.$$

B.2 Transformation to the rotating frame

In this part of the appendix we will demonstrate how to derive the time-dependent AQC Hamiltonian for the two- and one-photon pumped KNR. Let's consider the two- and one-photon parametrically driven Kerr-nonlinear resonator Hamiltonian that is given by

$$\hat{\mathcal{H}}(t) = \omega_r\hat{a}^\dagger\hat{a} - K\hat{a}^{\dagger 2}\hat{a}^2 + G(t)\left(e^{-i\omega_p(t)t}\hat{a}^{\dagger 2} + e^{i\omega_p(t)t}\hat{a}^2\right) + F(t)\left(e^{-i\omega_p(t)t/2}\hat{a}^\dagger + e^{i\omega_p(t)t/2}\hat{a}\right),$$

where ω_r is the frequency of the resonator, $\omega_p(t)$ is the two-photon pump frequency and the one-photon pump frequency is chosen half times the two-photon pump frequency. We transform to a frame rotating at the single-photon pump frequency by performing the following unitary transformation $\hat{\mathcal{U}}(t) = e^{i\omega_p(t)t\hat{a}^\dagger\hat{a}/2}$. Under a unitary transformation the state vector $|\psi(t)\rangle$ transforms according to

$$|\tilde{\psi}(t)\rangle = \hat{\mathcal{U}}(t)\,|\psi(t)\rangle \rightarrow |\psi(t)\rangle = \hat{\mathcal{U}}^\dagger(t)\,|\tilde{\psi}(t)\rangle\,. \tag{B.8}$$

Substituting the transformed state vector into the Schrödinger equation we obtain

$$\begin{aligned} i\frac{\partial\,|\tilde{\psi}(t)\rangle}{\partial t} &= i\frac{\partial\hat{\mathcal{U}}(t)}{\partial t}\,|\psi(t)\rangle + \hat{\mathcal{U}}(t)i\frac{\partial\,|\psi\rangle\,(t)}{\partial t} \\ &= i\frac{\partial\hat{\mathcal{U}}(t)}{\partial t}\,|\psi(t)\rangle + \hat{\mathcal{U}}(t)\hat{\mathcal{H}}(t)\,|\psi(t)\rangle\,, \end{aligned}$$

by using Eq. (B.8) we replace $|\psi(t)\rangle$ with $\hat{\mathcal{U}}^\dagger(t)\,|\tilde{\psi}(t)\rangle$ and get

$$i\frac{\partial\,|\tilde{\psi}(t)\rangle}{\partial t} = \left(i\frac{\partial\hat{\mathcal{U}}(t)}{\partial t}\hat{\mathcal{U}}^\dagger(t) + \hat{\mathcal{U}}(t)\hat{\mathcal{H}}(t)\hat{\mathcal{U}}^\dagger(t)\right)|\tilde{\psi}(t)\rangle = \tilde{\hat{\mathcal{H}}}(t)\,|\tilde{\psi}(t)\rangle\,,$$

where

$$\tilde{\hat{\mathcal{H}}}(t) \equiv i\frac{\partial\hat{\mathcal{U}}(t)}{\partial t}\hat{\mathcal{U}}^\dagger(t) + \hat{\mathcal{U}}(t)\hat{\mathcal{H}}(t)\hat{\mathcal{U}}^\dagger(t) \tag{B.9}$$

is the transformed Hamiltonian. We proceed by evaluating the first term in this expression

$$i\frac{\partial\hat{\mathcal{U}}(t)}{\partial t}\hat{\mathcal{U}}^\dagger(t) = -\frac{1}{2}\left(\omega_p(t) + \dot{\omega}_p(t)t\right)\hat{a}^\dagger\hat{a}\underbrace{\hat{\mathcal{U}}(t)\hat{\mathcal{U}}^\dagger(t)}_{1} = -\frac{1}{2}\left(\omega_p(t) + \dot{\omega}_p(t)t\right)\hat{a}^\dagger\hat{a},$$

where the dot $\dot{\omega}_p(t)$ denotes the time-derivative. We then calculate the second term of Eq. (B.9) by invoking the *Baker-Hausdorff* lemma,

$$e^A a e^{-A} = a + [A,a] + \frac{1}{2!}[A,[A,a]] + \ldots$$

where A is a Hermitian operator. Applying this formula we get that each operator $\hat{a}$ and $\hat{a}^\dagger$ transform according to

$$\begin{aligned} \hat{\mathcal{U}}^\dagger(t)\hat{a}\hat{\mathcal{U}}(t) &= \hat{a}\left(1 - i\omega_p(t)t/2 + \frac{(-i\omega_p(t)t/2)^2}{2!} + \ldots\right) = \hat{a}e^{-i\omega_p(t)t/2}, \\ \hat{\mathcal{U}}^\dagger(t)\hat{a}^\dagger\hat{\mathcal{U}}(t) &= \hat{a}^\dagger\left(1 + i\omega_p(t)t/2 + \frac{(i\omega_p(t)t/2)^2}{2!} + \ldots\right) = \hat{a}^\dagger e^{i\omega_p(t)t/2}, \end{aligned}$$

and thus the Hamiltonian in the rotating frame is given by (we can now drop the tilde)

$$\hat{\mathcal{H}}(t) = \left(\omega_r - \frac{1}{2}(\omega_p(t) + \dot{\omega}_p(t)t)\right)\hat{a}^\dagger\hat{a} - K\hat{a}^{\dagger 2}\hat{a}^2 + G(t)\left(\hat{a}^{\dagger 2} + \hat{a}^2\right) + F(t)\left(\hat{a}^\dagger + \hat{a}\right).$$

By cleverly choosing the pump frequency

$$\omega_p(t) = 2\omega_r + 2\delta\left(1 - \frac{t}{2\tau}\right) \Rightarrow \dot{\omega}_p(t) = -\delta\frac{1}{\tau}$$

and the drive strengths

$$G(t) = \frac{t}{\tau}G \quad \text{and} \quad F(t) = \frac{t}{\tau}F,$$

the Hamiltonian in the rotating frame now reads

$$\begin{aligned}\hat{\mathcal{H}}(t) &= -\left(1-\frac{t}{\tau}\right)\delta\hat{a}^\dagger\hat{a} - K\hat{a}^{\dagger 2}\hat{a}^2 + \frac{t}{\tau}G\left(\hat{a}^{\dagger 2}+\hat{a}^2\right) + \frac{t}{\tau}F\left(\hat{a}^\dagger+\hat{a}\right) \\ &= \left(1-\frac{t}{\tau}\right)\left(-\delta\hat{a}^\dagger\hat{a} - K\hat{a}^{\dagger 2}\hat{a}^2\right) + \frac{t}{\tau}\left(-K\hat{a}^{\dagger 2}\hat{a}^2 + G\left(\hat{a}^{\dagger 2}+\hat{a}^2\right) + F\left(\hat{a}^\dagger+\hat{a}\right)\right) \\ &= \left(1-\frac{t}{\tau}\right)\hat{\mathcal{H}}_0 + \frac{t}{\tau}\hat{\mathcal{H}}_1,\end{aligned} \tag{B.10}$$

where we have defined

$$\hat{\mathcal{H}}_0 \equiv \left(-\delta\hat{a}^\dagger\hat{a} - K\hat{a}^{\dagger 2}\hat{a}^2\right) \quad \text{and} \quad \hat{\mathcal{H}}_1 \equiv -K\hat{a}^{\dagger 2}\hat{a}^2 + G\left(\hat{a}^{\dagger 2}+\hat{a}^2\right) + F\left(\hat{a}^\dagger+\hat{a}\right).$$

B.3 Steady state & stability

In this section of the appendix we will demonstrate one way to obtain the steady state fixed points[1] for the two-photon pumped KNR in presence of single-photon loss.

The Lindblad master equation, which is an operator valued equation, can be turned into a differential equation (Fokker-Planck like equation) by first writing the density operator $\hat{\varrho}$ describing the state of the system, in terms of the coherent state projectors $|\alpha\rangle\langle\alpha|$. The coherent state representation of the density matrix is defined by [Glauber, 1963]

$$\hat{\varrho}(t) = \int d^2\alpha P(\alpha,\alpha^*,t)\,|\alpha\rangle\langle\alpha|\,, \tag{B.11}$$

where $P(\alpha,\alpha^*,t)$ is some weight function that is normalized such that $\mathrm{Tr}\left[\hat{\varrho}(t)\right]=1$. The integral of Eq. (B.11) is formally known as the Glauber Sudarshan P-representation of the state $\hat{\varrho}(t)$. To derive an equation for $P(\alpha,\alpha^*,t)$ we substitute the P-representation into the Lindblad master equation Eq. (11.5.2) and use the properties

$$\begin{aligned}\hat{a}\,|\alpha\rangle\langle\alpha| &= \alpha\,|\alpha\rangle\langle\alpha|\,, \\ |\alpha\rangle\langle\alpha|\,\hat{a}^\dagger &= \alpha^*\,|\alpha\rangle\langle\alpha|\,, \\ |\alpha\rangle\langle\alpha|\,\hat{a} &= \left(\frac{\partial}{\partial\alpha}+\alpha^*\right)|\alpha\rangle\langle\alpha|\,, \\ \hat{a}^\dagger\,|\alpha\rangle\langle\alpha| &= \left(\frac{\partial}{\partial\alpha^*}+\alpha\right)|\alpha\rangle\langle\alpha|\,,\end{aligned}$$

which can be derived from the definition of the coherent state. Next we perform integration by parts and assume that the boundary conditions at infinity are zero. This introduces a minus sign in front of each differential operator. The above properties show that we have the following correspondences

$$\begin{aligned}\hat{a}\hat{\varrho}(t) &\leftrightarrow \alpha P(\alpha,\alpha^*,t), \\ \hat{a}^\dagger\hat{\varrho}(t) &\leftrightarrow \left(\alpha^* - \frac{\partial}{\partial\alpha}\right)P(\alpha,\alpha^*,t), \\ \hat{\varrho}(t)\hat{a} &\leftrightarrow \left(\alpha - \frac{\partial}{\partial\alpha^*}\right)P(\alpha,\alpha^*,t), \\ \hat{\varrho}(t)\hat{a}^\dagger &\leftrightarrow \alpha^* P(\alpha,\alpha^*,t).\end{aligned} \tag{B.12}$$

[1] A fixed point is when a solution does not change in time.

Substituting Eq. (B.11) into the master equation Eq. (11.5.2) and using the identities given by Eq. (B.12) we get

$$\frac{\partial P(\alpha,\alpha^*,t)}{\partial t} = \Big(-\frac{\partial}{\partial\alpha}(i2K\alpha^2\alpha^* - i2G\alpha^* - \frac{\gamma}{2}\alpha) - \frac{\partial}{\partial\alpha^*}(-i2K\alpha^{*2}\alpha + i2G\alpha - \frac{\gamma}{2}\alpha^*), \\ + \frac{\partial^2}{\partial\alpha^2}(iK\alpha^2 - iG) + \frac{\partial^2}{\partial\alpha^{*2}}(-iK\alpha^{*2} + iG)\Big)P(\alpha,\alpha^*,t). \quad \text{(B.13)}$$

This equation is on the form of a *Fokker-Planck equation* and can moreover be written as a *stochastic differential equation* [Walls and Milburn, 2007]

$$\begin{aligned}\frac{d\alpha}{dt} &= i2\left(K\alpha^2\alpha^* - G\alpha^* + i\frac{\gamma}{4}\alpha\right) + \sqrt{i2(K\alpha^2 - G)}\eta(t),\\ \frac{d\alpha^*}{dt} &= i2\left(-K\alpha^{*2}\alpha + G\alpha + i\frac{\gamma}{4}\alpha^*\right) + \sqrt{i2(K\alpha^2 - G)}\eta^*(t).\end{aligned} \quad \text{(B.14)}$$

It can be shown that this stochastic differential equation is indeed equivalent to the Fokker-Planck equation. Here $\eta(t)$, $\eta^*(t)$ are stochastic forces (Langevin terms) with zero mean. The *semi-classical* or *mean value* equation is obtained by taking the average, the Langevin terms thus disappear since $\langle\eta(t)\rangle = \langle\eta(t)^*\rangle = 0$ [Walls and Milburn, 2007]. The semi-classical equation of motion thus are

$$\begin{aligned}\frac{d\alpha}{dt} &= i2\left(K\alpha^2\alpha^* - G\alpha^* + i\frac{\gamma}{4}\alpha\right),\\ \frac{d\alpha^*}{dt} &= i2\left(-K\alpha^{*2}\alpha + G\alpha + i\frac{\gamma}{4}\alpha^*\right).\end{aligned} \quad \text{(B.15)}$$

The steady-state to the semi-classical equation of motion can now readily be obtained by setting $d\alpha/dt = 0$, and solving this equation. The equation $K\alpha^2\alpha^* - G\alpha^* + i\frac{\gamma}{4}\alpha = 0$ has three solutions which are given by $\alpha_1^s = 0$ and $\alpha_{2,3}^s = \pm re^{i\vartheta}$, where

$$r = \frac{1}{2}\left(\frac{16G^2 - \gamma^2}{K^2}\right)^{1/4} \quad \text{and} \quad \vartheta = -\frac{1}{2}\arctan\left(\frac{\gamma}{\sqrt{16G^2 - \gamma^2}}\right).$$

which reduces to $\alpha_{2,3}^s \simeq \sqrt{G/K}$ in the case of $4G \gg \gamma$.

To investigate the stability of these solutions we linearise Eq. (B.15) around the steady states

$$\alpha_i(t) = \alpha_i^s + \delta\alpha_i(t), \quad (i = 1,2,3),$$

where α_i^s is the i:*th* steady-state solution to Eq. (B.15) and $\delta\alpha_i$ is a small perturbation. The linearized equations written in matrix form are

$$\frac{d}{dt}\begin{pmatrix}\delta\alpha_i(t)\\ \delta\alpha_i^*(t)\end{pmatrix} \simeq i2\begin{pmatrix}2K|\alpha_i^s|^2 + i\gamma/4 & K(\alpha_i^s)^2 - G\\ -K(\alpha_i^{s*})^2 + G & -2K|\alpha_i^s|^2 + i\gamma/4\end{pmatrix}\begin{pmatrix}\delta\alpha_i(t)\\ \delta\alpha_i^*(t)\end{pmatrix},$$

and the eigenvalues to the matrix equation are

$$\lambda_\pm = -\frac{\gamma}{2} \pm 2\,\mathrm{Im}\left(\sqrt{3K^2|\alpha_i^s|^4 + G\left(K(\alpha_i^{s*})^2 - G + (\alpha_i^s)^2K\right)}\right).$$

Stability of the fixed points require all eigenvalues to have a positive real part less than or equal to zero [Walls and Milburn, 2007]. We begin by examining the first steady state solution which is the origin $\alpha_1^s = 0$, it has eigenvalues $\lambda_\pm = -\gamma/2 \pm 2G$, so if $4G > \gamma$, then the origin is an unstable solution. The eigenvalues to the second and third steady state solution $\alpha_{2,3}^s = \pm re^{i\vartheta}$ is most easily analyzed in the case when $4G \gg \gamma$, where both steady state solutions has the degenerate eigenvalue $-\gamma/2$, and are thus stable solutions.

B.4 Effect of single-photon pump

In this section we will study the effect that is obtained when a single-photon pump is added to the two-photon pumped KNR. For the moment we will neglect the effect of single-photon loss, since it has already been treated in appendix B.3. The Hamiltonian for the two- and one-photon KNR written in a frame rotating at the resonator frequency is given by

$$\hat{\mathcal{H}} = -K\hat{a}^{\dagger 2}\hat{a}^2 + G\left(\hat{a}^{\dagger 2} + \hat{a}^2\right) + F\left(\hat{a}^\dagger + \hat{a}\right).$$

In the Heisenberg picture the time evolution of the annihilation operator is given by

$$\frac{d\hat{a}}{dt} = i\Big[\hat{\mathcal{H}}, \hat{a}\Big] = i\left(2K\hat{a}^\dagger\hat{a}^2 - 2G\hat{a}^\dagger - F\right).$$

To obtain the semi-classical equations of motion we take the average $\langle\hat{a}\rangle = \alpha$, which is obtained by simply replacing $\hat{a}$ with the complex variable α,

$$\frac{d\alpha}{dt} = i\left(2K\alpha^*\alpha^2 - 2G\alpha^* - F\right).$$

The steady state to this equation has three solutions which are of the form $\alpha_1 = (-2\varepsilon, 0)$ and $\alpha_{2,3} = (\pm\alpha + \varepsilon, 0)$ where $\alpha = \sqrt{G/K}$ and $\varepsilon = F/4G$. Hence if $4G \gg F$ so that $\varepsilon \to 0$, then $\alpha_{2,3} \simeq \pm\sqrt{G/K}$. To investigate the stability of the steady state solutions we do a small perturbation around the fixed point. The linearized equation on motion are

$$\begin{pmatrix} \dfrac{d(\delta\alpha)}{dt} \\ \dfrac{d(\delta\alpha^*)}{dt} \end{pmatrix} \simeq i2 \begin{pmatrix} 2K|\alpha_i^s|^2 & K(\alpha_i^s)^2 - G \\ -K(\alpha_i^{s*})^2 + G & -2K|\alpha_i^s|^2 \end{pmatrix} \begin{pmatrix} \delta\alpha \\ \delta\alpha^* \end{pmatrix} \tag{B.16}$$

which has the eigenvalues

$$\lambda_\pm = \pm 2\,\mathrm{Im}\left(\sqrt{4K^2|\alpha_i^s|^4 - |G - K(\alpha_i^s)^2|^2}\right). \tag{B.17}$$

We begin by examining the eigenvalues for the first steady state solution $\alpha_1^s = -2\varepsilon$. It has eigenvalues of the form $\lambda_\pm \simeq \pm 2\,\mathrm{Im}\left(\sqrt{KF^2/(2G) - G^2}\right)$. If the expression inside the square root is negative then the α_1^s will have an eigenvalue with a positive real part. This happens if $F < \sqrt{2}K\alpha^3$. The eigenvalue of the two other solutions α_2^s and α_3^s both has $\lambda_\pm = 0$ as eigenvalue and are therefore stable.

B.5 Coupling between two Kerr-nonlinear resonators

In this section of the appendix we will obtain the fixed points for two linearly coupled KNR's. The Hamiltonian for two linearly coupled two-photon pumped KNRs in the rotating frame is given by

$$\hat{\mathcal{H}} = \sum_{i=1}^{2}\left(-K\hat{a}_i^{\dagger 2}\hat{a}_i^2\right) + G\left(\hat{a}_i^{\dagger 2} + \hat{a}_i^2\right) + g\left(\hat{a}_1^\dagger\hat{a}_2 + \hat{a}_1\hat{a}_2^\dagger\right).$$

The Heisenberg equations of motion tell us how the the operators evolve in time

$$\frac{d\hat{a}_1}{dt} = i\Big[\hat{\mathcal{H}}, \hat{a}_1\Big] = i\left(2K\hat{a}_1^\dagger\hat{a}_1^2 - 2G\hat{a}_1^\dagger - g\hat{a}_2\right),$$
$$\frac{d\hat{a}_2}{dt} = i\Big[\hat{\mathcal{H}}, \hat{a}_2\Big] = i\left(2K\hat{a}_2^\dagger\hat{a}_2^2 - 2G\hat{a}_2^\dagger - g\hat{a}_1\right).$$

Once again the mean field or semi classical equations are obtained by replacing $\hat{a}_1 \to \alpha$ and $\hat{a}_2 \to \beta$, where α and β are two complex variables. We thus get that the semi-classical equations of motion for two coupled KNRs are

$$\frac{d\alpha}{dt} = i\left(2K\alpha^*\alpha^2 - 2G\alpha^* - g\beta\right),$$
$$\frac{d\beta}{dt} = i\left(2K\beta^*\beta^2 - 2G\beta^* - g\alpha\right).$$

The solutions to the steady state equations are

$$\{\alpha_1^s, \beta_1^s\} = \{0, 0\},$$
$$\{\alpha_{2,3}^s, \beta_{2,3}^s\} = \{\pm\sqrt{\frac{2G+g}{2K}}, \pm\sqrt{\frac{2G+g}{2K}}\}$$
$$\{\alpha_{4,5}^s, \beta_{4,5}^s\} = \{\mp\sqrt{\frac{2G+g}{2K}}, \pm\sqrt{\frac{2G+g}{2K}}\}.$$

which reduces to $\alpha_{2,3}^s \simeq \sqrt{G/K}$ when $2G \gg g$, and the states are kept in the subspace spanned by $|\bar{0}, \bar{0}\rangle$, $|\bar{0}, \bar{1}\rangle$, $|\bar{1}, \bar{0}\rangle$ and $|\bar{1}, \bar{1}\rangle$. Following the same kind of stability analysing procedure that we did in section B.3 we find likewise that the origin is an unstable solution while $\alpha_{2,3}^s$ are stable solutions.

B.6 Error estimation

In this part of the appendix we aim at providing an order of magnitude of the error that is inherent in the truncation of the Hilbert space dimension corresponding to $n = 16$ photons, as we have used in the numerical simulations of Chapter 5. In order to do so, we evaluate the error that this truncation entails on the normalization condition for a coherent state. From the normalization condition of a coherent we have that

$$\langle\alpha|\alpha\rangle = 1 = e^{-|\alpha|^2}\sum_{n=0}^{\infty}\frac{|\alpha|^{2n}}{n!},$$

which can be written as

$$1 - e^{-|\alpha|^2}\sum_{n=0}^{\infty}\frac{|\alpha|^{2n}}{n!} = 0.$$

If we truncate the Fock space at the number x then the numerical error ε is given by

$$\varepsilon = 1 - e^{-|\alpha|^2}\sum_{n=0}^{x}\frac{|\alpha|^{2n}}{n!},$$

and the error is bounded between $0 \leq \varepsilon \leq 1 - e^{-|\alpha|^2}$. To have an error that's bounded between zero and one we have to divide by $1 - e^{-|\alpha|^2}$, such that

$$\varepsilon = \left(1 - e^{-|\alpha|^2}\sum_{n=0}^{x}\frac{|\alpha|^{2n}}{n!}\right) / \left(1 - e^{-2|\alpha|^2}\right).$$

Now if $x = 16$ and $|\alpha| = \sqrt{3}$ we find that the numerical error is approximately $\varepsilon \approx 10^{-8}$, which is a considerably small numerical error.

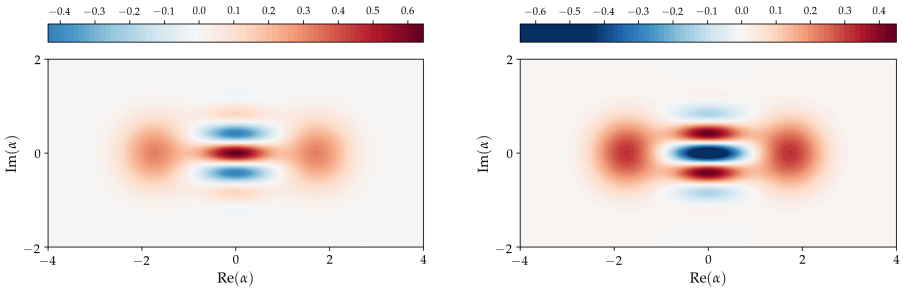
B.7 Generation of cat states using a two-photon pumped KNR

The two-photon pumped KNR can be used as a mean to generate Schrödinger cat states. Cat states are a superposition of two-coherent states with opposite phase, defined by

$$|C_\alpha^\pm\rangle = \frac{|\alpha\rangle \pm |-\alpha\rangle}{\sqrt{2(1 \pm e^{-2|\alpha|^2})}},$$

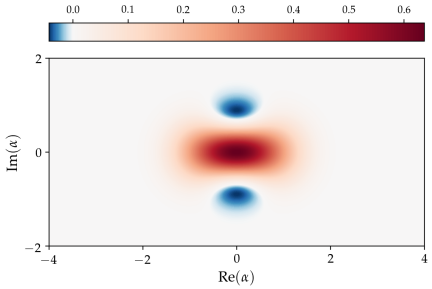
where the term in the denominator is a normalization factor. $|C_\alpha^+\rangle$ is called an even-cat state because when written in the Fock basis it only contain Fock states with even numbers and $|C_\alpha^-\rangle$ is called an odd-cat state because when written in the Fock-basis it only contains Fock states with odd numbers. Cat states are of interest because they have the potential to be used as logical states for universal quantum computation [Puri et al., 2017], and hold potential for error-correction against single-photon losses (see experimental work from Yale's group). To generate a cat state it can first be noted that since the two-coherent states $|\pm\alpha\rangle = |\pm\sqrt{G/K}\rangle$ are two-degenerate eigenstates of the two-photon pumped KNR Eq. (11.79) so is any superposition of theses states. It can be easily checked that the even and odd cat state with $\alpha = \sqrt{G/K}$ are also two degenerate eigenstates of the two-photon pumped KNR Eq. (11.79) with eigenenergy G^2/K. Second, to generate a cat state the detuning and single-photon pump is set to zero in the time-dependent Hamiltonian Eq. (11.82). The vacuum and single-photon Fock state are then degenerate eigenstates of the initial Hamiltonian. Under adiabatic evolution the zero-photon Fock state will evolve into the even cat-state since parity is conserved, and the single-photon Fock state will evolve into the odd-cat state for the same reason.

Numerical simulations show that, for $G = 3K$ and $\tau = 158$ as given by the adiabatic condition, if the resonator is initialized to $|0\rangle$ ($|1\rangle$) it will evolve into the even cat state (odd cat state) with 100% fidelity, within the numerical error. Figure B.2 (a) and (b) show the Wigner function of the even and odd cat. The Wigner negativity that appears between the two Gaussian peaks are interference fringes and are sometimes referred to as the cat's whiskers. These interference fringes are very characteristic for a cat state. When the amplitude of $|\alpha|$ is small the states are often referred to as "kitten" states. For example an even cat ("kitten") state with $|\alpha| = \sqrt{1/2}$ is shown in Fig. B.2 (c).



(a) Even cat state

(b) Odd cat state



(c) Kitten state

Figure B.2: Cat states.

Bibliography

[Aaronson, a] Aaronson, S. The complexity zoo.

[Aaronson, b] Aaronson, S. Postbqp postscripts: A confession of mathematical errors.

[Aaronson, 2005] Aaronson, S. (2005). Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 461:3473.

[Aaronson, 2015] Aaronson, S. (2015). Read the fine print. *Nat. Phys.*, 11:291.

[Aaronson, 2018] Aaronson, S. (2018). Lecture Notes for Intro to Quantum Information Science.

[Aaronson and Arkhipov, 2013] Aaronson, S. and Arkhipov, A. (2013). The computational complexity of linear optics. *Theory of Computing*, 9:143.

[Albash and Lidar, 2018] Albash, T. and Lidar, D. A. (2018). Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002.

[Allcock and Zhang, 2019] Allcock, J. and Zhang, S. (2019). Quantum machine learning. *Natl. Sci. Rev.*, 6:26.

[Amin et al., 2018] Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., and Melko, R. (2018). Quantum Boltzmann Machine. *Phys. Rev. X*, 8:021050.

[Arute et al., 2019] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M. P., Hartmann, M. J., Ho, A., Hoffmann, M., Huang, T., Humble, T. S., Isakov, S. V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P. V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M. Y., Ostby, E., Petukhov, A., Platt, J. C., Quintana, C., Rieffel, E. G., Roushan, P., Rubin, N. C., Sank, D., Satzinger, K. J., Smelyanskiy, V., Sung, K. J., Trevithick, M. D., Vainsencher, A., Villalonga, B., White, T., Yao, Z. J., Yeh, P., Zalcman, A., Neven, H., and Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510.

[Barak et al., 2015] Barak, B., Moitra, A., O'Donnell, R., Raghavendra, P., Regev, O., Steurer, D., Trevisan, L., Vijayaraghavan, A., Witmer, D., and Wright, J. (2015). Beating the random assignment on constraint satisfaction problems of bounded degree.

[Bartlett et al., 2002] Bartlett, S. D., Sanders, B. C., Braunstein, S. L., and Nemoto, K. (2002). Efficient classical simulation of continuous variable quantum information processes. *Phys. Rev. Lett.*, 88:9.

[Bartolo et al., 2016] Bartolo, N., Minganti, F., Casteels, W., and Ciuti, C. (2016). Exact steady state of a kerr resonator with one-and two-photon driving and dissipation: Controllable wigner-function multimodality and dissipative phase transitions. *Physical Review A*, 94(3):033841.

[Bouland et al., 2019] Bouland, A., Fefferman, B., Nirkhe, C., and Vazirani, U. (2019). On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163.

[Brandao et al., 2018] Brandao, F. G. S. L., Broughton, M., Farhi, E., Gutmann, S., and Neven, H. (2018). For Fixed Control Parameters the Quantum Approximate Optimization Algorithm's Objective Function Value Concentrates for Typical Instances.

[Bremner et al., 2010] Bremner, M. J., Josza, R., and Shepherd, D. (2010). Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. R. Soc. A*, 459:459.

[Bremner et al., 2016] Bremner, M. J., Montanaro, A., and Shepherd, D. J. (2016). Average-case complexity versus approximate simulation of commuting quantum computations. *Physical review letters*, 117(8):080501.

[Briegel et al., 2009] Briegel, H. J., Browne, D. E., Dür, W., Raussendorf, R., and Van den Nest, M. (2009). Measurement-based quantum computation. *Nat. Phys.*, 5:19.

[Campagne-Ibarcq et al., 2020] Campagne-Ibarcq, P., Eickbusch, A., Touzard, S., Zalys-Geller, E., Frattini, N., Sivak, V., Reinhold, P., Puri, S., Shankar, S., Schoelkopf, R., et al. (2020). Quantum error correction of a qubit encoded in grid states of an oscillator. *Nature*, 584(7821):368–372.

[Chabaud et al., 2017] Chabaud, U., Douce, T., Markham, D., van Loock, P., Kashefi, E., and Ferrini, G. (2017). Continuous-variable sampling from photon-added or photon-subtracted squeezed states. *Physical Review A*, 96:062307.

[Chakhmakhchyan and Cerf, 2017] Chakhmakhchyan, L. and Cerf, N. J. (2017). Boson sampling with gaussian measurements. *Physical Review A*, 96(3):032326.

[Choi, 2008] Choi, V. (2008). Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. *Quantum Information Processing*, 7(5):193–209.

[Choi, 2011] Choi, V. (2011). Minor-embedding in adiabatic quantum computation: Ii. minor-universal graph design. *Quantum Information Processing*, 10(3):343–353.

[Cong et al., 2019] Cong, I., Choi, S., and Lukin, M. D. (2019). Quantum convolutional neural networks. *Nat. Phys.*, 15:1273.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control. Signals, Syst.*, 2:303.

[Dieks, 1982] Dieks, D. (1982). Communication by EPR devices. *Physics Letters A*, 92:271.

[Douce et al., 2017] Douce, T., Markham, D., Kashefi, E., Diamanti, E., Coudreau, T., Milman, P., van Loock, P., and Ferrini, G. (2017). Continuous-variable instantaneous quantum computing is hard to sample. *Phys. Rev. Lett.*, 118:070503.

[Douce et al., 2019] Douce, T., Markham, D., Kashefi, E., van Loock, P., and Ferrini, G. (2019). Probabilistic fault-tolerant universal quantum computation and sampling problems in continuous variables. *Physical Review A*, 99:012344.

[Farhi et al., 2014] Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.

[Farhi et al., 2001] Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., and Preda, D. (2001). A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, 292:472.

[Farhi et al., 2017] Farhi, E., Goldstone, J., Gutmann, S., and Neven, H. (2017). Quantum algorithms for fixed qubit architectures. *arXiv preprint arXiv:1703.06199*.

[Farhi and Harrow, 2019] Farhi, E. and Harrow, A. W. (2019). Quantum supremacy through the quantum approximate optimization algorithm.

[Farhi and Neven, 2018] Farhi, E. and Neven, H. (2018). Classification with Quantum Neural Networks on Near Term Processors.

[Flühmann et al., 2018] Flühmann, C., Negnevitsky, V., Marinelli, M., and Home, J. P. (2018). Sequential modular position and momentum measurements of a trapped ion mechanical oscillator. *Phys. Rev. X*, 8:021001.

[Fowler et al., 2012] Fowler, A. G., Mariantoni, M., Martinis, J. M., and Cleland, A. N. (2012). Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86:032324.

[García-Álvarez et al., 2020] García-Álvarez, L., Calcluth, C., Ferraro, A., and Ferrini, G. (2020). Efficient simulatability of continuous-variable circuits with large wigner negativity. *Phys. Rev. Research*, 2:043322.

[Gerry et al., 2005] Gerry, C., Knight, P., and Knight, P. L. (2005). *Introductory quantum optics*. Cambridge university press.

[Glancy and Knill, 2006] Glancy, S. and Knill, E. (2006). Error analysis for encoding a qubit in an oscillator. *Phys. Rev. A*, 73:012325.

[Glauber, 1963] Glauber, R. J. (1963). Coherent and incoherent states of the radiation field. *Physical Review*, 131(6):2766.

[Gottesman et al., 2001] Gottesman, D., Kitaev, A., and Preskill, J. (2001). Encoding a qubit in an oscillator. *Phys. Rev. A*, 64:012310.

[Gu et al., 2009] Gu, M., Weedbrook, C., Menicucci, N. C., Ralph, T. C., and van Loock, P. (2009). Quantum computing with continuous-variable clusters. *Phys. Rev. A*, 79:062318.

[Hadfield et al., 2019] Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E., Venturelli, D., and Biswas, R. (2019). From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms*, 12(2):34.

[Halperin et al., 2004] Halperin, E., Livnat, D., and Zwick, U. (2004). Max cut in cubic graphs. *Journal of Algorithms*, 53(2):169–185.

[Hamilton et al., 2017] Hamilton, C. S., Kruse, R., Sansoni, L., Barkhofen, S., Silberhorn, C., and Jex, I. (2017). Gaussian boson sampling. *Physical review letters*, 119(17):170501.

[Harrow et al., 2009] Harrow, A. W., Hassidim, A., and Lloyd, S. (2009). Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.*, 103:150502.

[Harrow and Montanaro, 2017] Harrow, A. W. and Montanaro, A. (2017). Quantum computational supremacy. *Nature*, 549(7671):203.

[Hauke et al., 2020] Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H., and Oliver, W. D. (2020). Perspectives of quantum annealing: Methods and implementations. *Reports on Progress in Physics*, 83(5):054401.

[Hillmann et al., 2020] Hillmann, T., Quijandría, F., Johansson, G., Ferraro, A., Gasparinetti, S., and Ferrini, G. (2020). Universal gate set for continuous-variable quantum computation with microwave circuits. *Phys. Rev. Lett.*, 125:160501.

[Horodecki et al., 2006] Horodecki, P., Bruß, D., and Leuchs, G. (2006). Lectures on quantum information.

[Huh et al., 2015] Huh, J., Guerreschi, G. G., Peropadre, B., McClean, J. R., and Aspuru-Guzik, A. (2015). Boson sampling for molecular vibronic spectra. *Nature Photonics*, 9(9):615–620.

[Jiang et al., 2017] Jiang, Z., Rieffel, E. G., and Wang, Z. (2017). Near-optimal quantum circuit for Grover's unstructured search using a transverse field. *Physical Review A*, 95(6):062317.

[Kandala et al., 2017] Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J. M., and Gambetta, J. M. (2017). Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549:242.

[Kirchmair et al., 2013] Kirchmair, G., Vlastakis, B., Leghtas, Z., Nigg, S. E., Paik, H., Ginossar, E., Mirrahimi, M., Frunzio, L., Girvin, S. M., and Schoelkopf, R. J. (2013). Observation of quantum state collapse and revival due to the single-photon kerr effect. *Nature*, 495(7440):205.

[Kockum, 2014] Kockum, A. F. (2014). *Quantum optics with artificial atoms*. PhD thesis, Chalmers University of Technology.

[Kockum and Nori, 2019] Kockum, A. F. and Nori, F. (2019). Quantum Bits with Josephson Junctions. In Tafuri, F., editor, *Fundamentals and Frontiers of the Josephson Effect*, pages 703–741. Springer.

[Kuperberg, 2015] Kuperberg, G. (2015). How hard is it to approximate the jones polynomial? *Theory of Computing*, 11:183.

[Lechner et al., 2015] Lechner, W., Hauke, P., and Zoller, P. (2015). A quantum annealing architecture with all-to-all connectivity from local interactions. *Science advances*, 1(9):e1500838.

[Leonhardt, 1997] Leonhardt, U. (1997). *Measuring the Quantum State of Light*. Cambridge University Press, New York, NY, USA, 1st edition.

[Leonhardt and Paul, 1993] Leonhardt, U. and Paul, H. (1993). Realistic optical homodyne measurements and quasiprobability distributions. *Phys. Rev. A*, 48:4598.

[Li et al., 2020] Li, Y., Chen, M., Chen, Y., Lu, H., Gan, L., Lu, C., Pan, J., Fu, H., and Yang, G. (2020). Benchmarking 50-photon gaussian boson sampling on the sunway taihulight. *arXiv preprint arXiv:2009.01177*.

[Lin et al., 2017] Lin, H. W., Tegmark, M., and Rolnick, D. (2017). Why Does Deep and Cheap Learning Work So Well? *J. Stat. Phys.*, 168:1223.

[Lloyd, 2018] Lloyd, S. (2018). Quantum approximate optimization is computationally universal.

[Lloyd and Braunstein, 1999] Lloyd, S. and Braunstein, S. L. (1999). Quantum computation over continuous variables. *Phys. Rev. Lett.*, 82:1784.

[Lloyd et al., 2014] Lloyd, S., Mohseni, M., and Rebentrost, P. (2014). Quantum principal component analysis. *Nat. Phys.*, 10:631.

[Lucas, 2014] Lucas, A. (2014). Ising formulations of many np problems. *Frontiers in Physics*, 2:5.

[Lund et al., 2017] Lund, A., Bremner, M. J., and Ralph, T. (2017). Quantum sampling problems, boson-sampling and quantum supremacy. *npj Quantum Information*, 3(1):15.

[Lund et al., 2014] Lund, A. P., Rahimi-Keshari, S., Rudolph, T., OÕBrien, J. L., and Ralph, T. C. (2014). Boson sampling from a gaussian state. *Phys. Rev. Lett.*, 113:100502.

[Mari and Eisert, 2012] Mari, A. and Eisert, J. (2012). Positive wigner functions render classical simulation of quantum computation efficient. *Phys. Rev. Lett.*, 109:230503.

[Meaney et al., 2014] Meaney, C. H., Nha, H., Duty, T., and Milburn, G. J. (2014). Quantum and classical nonlinear dynamics in a microwave cavity. *EPJ Quantum Technology*, 1(1):7.

[Menicucci, 2014] Menicucci, N. C. (2014). Fault-tolerant measurement-based quantum computing with continuous-variable cluster states. *Phys. Rev. Lett.*, 112:120504.

[Menicucci et al., 2011] Menicucci, N. C., Flammia, S. T., and van Loock, P. (2011). Graphical calculus for gaussian pure states. *Physical Review A*, 83(4):042335.

[Meystre and Sargent, 2007] Meystre, P. and Sargent, M. (2007). *Elements of quantum optics*. Springer Science & Business Media.

[Moll et al., 2018] Moll, N., Barkoutsos, P., Bishop, L. S., Chow, J. M., Cross, A., Egger, D. J., Filipp, S., Fuhrer, A., Gambetta, J. M., Ganzhorn, M., Kandala, A., Mezzacapo, A., Müller, P., Riess, W., Salis, G., Smolin, J., Tavernelli, I., and Temme, K. (2018). Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci. Technol.*, 3:030503.

[Morales et al., 2019] Morales, M. E. S., Biamonte, J., and Zimboras, Z. (2019). On the universality of the quantum approximate optimization algorithm.

[Moylett et al., 2019] Moylett, A. E., García-Patrón, R., Renema, J. J., and Turner, P. S. (2019). Classically simulating near-term partially-distinguishable and lossy boson sampling. *Quantum Science and Technology*, 5(1):015001.

[Neville et al., 2017] Neville, A., Sparrow, C., Clifford, R., Johnston, E., Birchall, P. M., Montanaro, A., and Laing, A. (2017). Classical boson sampling algorithms with superior performance to near-term experiments. *Nature Physics*, 13(12):1153–1157.

[Nielsen, 2015] Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.

[Nielsen and Chuang, 2000] Nielsen, M. A. and Chuang, I. L. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press.

[Nielsen and Chuang, 2011] Nielsen, M. A. and Chuang, I. L. (2011). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition.

[Nigg et al., 2017] Nigg, S. E., Lörch, N., and Tiwari, R. P. (2017). Robust quantum optimizer with full connectivity. *Science advances*, 3(4):e1602273.

[Niu et al., 2019] Niu, M. Y., Lu, S., and Chuang, I. L. (2019). Optimizing QAOA: Success Probability and Runtime Dependence on Circuit Depth.

[Ofek et al., 2016] Ofek, N., Petrenko, A., Heeres, R., Reinhold, P., Leghtas, Z., Vlastakis, B., Liu, Y., Frunzio, L., Girvin, S., Jiang, L., et al. (2016). Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature*, 536(7617):441–445.

[Paris et al., 2003] Paris, M. G. A., Cola, M., and Bonifacio, R. (2003). Quantum-state engineering assisted by entanglement. *Phys. Rev. A*, 67:042104.

[Pednault et al., 2019] Pednault, E., Gunnels, J. A., Nannicini, G., Horesh, L., and Wisnieff, R. (2019). Leveraging secondary storage to simulate deep 54-qubit sycamore circuits.

[Puri et al., 2017] Puri, S., Andersen, C. K., Grimsmo, A. L., and Blais, A. (2017). Quantum annealing with all-to-all connected nonlinear oscillators. *Nature communications*, 8:15785.

[Qi et al., 2020] Qi, H., Brod, D. J., Quesada, N., and García-Patrón, R. (2020). Regimes of classical simulability for noisy gaussian boson sampling. *Phys. Rev. Lett.*, 124:100502.

[Rahimi-Keshari et al., 2016] Rahimi-Keshari, S., Ralph, T. C., and Caves, C. M. (2016). Sufficient conditions for efficient classical simulation of quantum opticss. *Phys. Rev. X*, 6:021039.

[Raussendorf and Briegel, 2001] Raussendorf, R. and Briegel, H. J. (2001). A One-Way Quantum Computer. *Phys. Rev. Lett.*, 86:5188.

[Raussendorf et al., 2003] Raussendorf, R., Browne, D. E., and Briegel, H. J. (2003). Measurement-based quantum computation on cluster states. *Phys. Rev. A*, 68:022312.

[Rebentrost et al., 2014] Rebentrost, P., Mohseni, M., and Lloyd, S. (2014). Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.*, 113:130503.

[Rodríguez-Laguna and Santalla, 2018] Rodríguez-Laguna, J. and Santalla, S. N. (2018). Building an adiabatic quantum computer simulation in the classroom. *American Journal of Physics*, 86(5):360–367.

[Scheel, 2004] Scheel, S. (2004). Permanents in linear optical networks. *arXiv preprint quant-ph/0406127*.

[Shor, 1995] Shor, P. W. (1995). Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52:R2493.

[Spring et al., 2013] Spring, J. B., Metcalf, B. J., Humphreys, P. C., Kolthammer, W. S., Jin, X.-M., Barbieri, M., Datta, A., Thomas-Peter, N., Langford, N. K., Kundys, D., Gates, J. C., Smith, B. J., Smith, P. G. R., and Walmsley, I. A. (2013). Boson sampling on a photonic chip. *Science*, 339:798.

[Stollenwerk et al., 2019] Stollenwerk, T., Lobe, E., and Jung, M. (2019). Flight gate assignment with a quantum annealer. In *International Workshop on Quantum Technology and Optimization Problems*, pages 99–110. Springer.

[Tang, 2018] Tang, E. (2018). Quantum-inspired classical algorithms for principal component analysis and supervised clustering.

[Tanikic and Despotovic, 2012] Tanikic, D. and Despotovic, V. (2012). Artificial Intelligence Techniques for Modelling of Temperature in the Metal Cutting Process. In *Metall. - Adv. Mater. Process.* InTech.

[Terhal and DiVincenzo, 2002] Terhal, B. M. and DiVincenzo, D. P. (2002). Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games. *arXiv preprint quant-ph/0205133*.

[Tinkham, 2004] Tinkham, M. (2004). *Introduction to superconductivity*. Courier Corporation.

[Ukai et al., 2010] Ukai, R., Yoshikawa, J.-i., Iwata, N., van Loock, P., and Furusawa, A. (2010). Universal linear bogoliubov transformations through one-way quantum computation. *Physical review A*, 81(3):032315.

[Vikstål, 2018] Vikstål, P. (2018). Continuous-variable quantum annealing with superconducting circuits. *Master Thesis, Chalmers*.

[Vikstål et al., 2020] Vikstål, P., Grönkvist, M., Svensson, M., Andersson, M., Johansson, G., and Ferrini, G. (2020). Applying the quantum approximate optimization algorithm to the tail-assignment problem. *Physical Review Applied*, 14(3):034009.

[Walls and Milburn, 2007] Walls, D. F. and Milburn, G. J. (2007). *Quantum optics*. Springer Science & Business Media.

[Walther et al., 2005] Walther, P., Resch, K. J., Rudolph, T., Schenck, E., Weinfurter, H., Vedral, V., Aspelmeyer, M., and Zeilinger, A. (2005). Experimental one-way quantum computing. *Nature*, 434:169.

[Wang et al., 2019] Wang, H., Qin, J., Ding, X., Chen, M.-C., Chen, S., You, X., He, Y.-M., Jiang, X., You, L., Wang, Z., et al. (2019). Boson sampling with 20 input photons and a 60-mode interferometer in a 10 power 14-dimensional hilbert space. *Physical review letters*, 123(25):250503.

[Watrous, 2009] Watrous, J. (2009). *Encyclopedia of Complexity and Systems Science*, chapter Quantum Computational Complexity, pages 7174–7201. Springer New York, New York, NY.

[Wendin, 2017] Wendin, G. (2017). Quantum information processing with superconducting circuits: a review. *Reports Prog. Phys.*, 80:106001.

[Willsch et al., 2020] Willsch, M., Willsch, D., Jin, F., De Raedt, H., and Michielsen, K. (2020). Benchmarking the quantum approximate optimization algorithm. *Quantum Information Processing*, 19:197.

[Wootters and Zurek, 1982] Wootters, W. K. and Zurek, W. H. (1982). A single quantum cannot be cloned. *Nature*, 299:802.

[Wurtz and Love, 2020] Wurtz, J. and Love, P. J. (2020). Bounds on maxcut qaoa performance for p$>$ 1. *arXiv preprint arXiv:2010.11209*.

[Zhong et al., 2020] Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Qin, J., Wu, D., Ding, X., Hu, Y., Hu, P., Yang, X.-Y., Zhang, W.-J., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., Wang, Z., Li, L., Liu, N.-L., Lu, C.-Y., and Pan, J.-W. (2020). Quantum computational advantage using photons. *Science*.