



UNIVERSITATEA DIN
BUCUREȘTI
VIRTUTE ET SAPIENTIA



FACULTATEA DE FIZICĂ

QUANTUM COMPUTING WITH PHOTONS

BACHELOR'S THESIS

Graduate

Dănuț-Valentin Dinu

Scientific advisors

Prof. dr. Radu Ionicioiu

Conf. dr. Iulia Ghiu

Bucharest, 2021

Acknowledgements

The past two years have been hard for all of us. I would like in the beginning of this thesis to thank everyone who made it possible. I am very grateful to my advisors, Prof. dr. Radu Ionicioiu and Conf. dr. Iulia Ghiu, for answering my questions and helping me with the scientific content. I also thank the Department of Theoretical Physics within the National Institute for Physics and Nuclear Engineering "Horia Hulubei" for giving me the opportunity to research the topics of this thesis. I thank my parents and my friends for their unconditional love and support. Special thanks go to David Mazurencu-Marinescu-Pele, George Prodan and Florin Vintilă for interesting discussions and help with the editing of the text.

Contents

1	Motivation	1
2	Quantum Computing	3
2.1	The Qubit	3
2.2	The Gate Model	5
2.2.1	Single-Qubit Gates	5
2.2.2	Multi-qubit Gates	7
2.2.3	Universality	9
2.3	Measurement-Based Quantum Computing	13
2.3.1	The Stabilizer Formalism	13
2.3.2	Graph and cluster states	17
2.3.3	Implementation of Single and Two-Qubit Gates	18
3	Photonic Quantum Information Processing	25
3.1	A quantum description of light	25
3.1.1	The classical electromagnetic field	25
3.1.2	The quantization of the electromagnetic field	27
3.2	Photons as qubits	29
3.2.1	Linear quantum optics devices	29
3.2.2	Dual-rail encoding	33
3.2.3	The KLM protocol	34
3.3	BosonSampling	37
3.3.1	Aaronson & Archipov BosonSampling	37
3.3.2	BosonSampling with Gaussian states	41
4	Conclusions	47
A	The Haar measure	49
B	Simulating BosonSampling	51
C	Simulating GBS	53
	References	54

Motivation

Our understanding of natural phenomena and how we can manipulate them to our advantage has advanced tremendously over the last centuries. Merely 50 years ago, video conferences, trans-continental communication, readily available sources of information and many other modern technological improvements were attributed to some distant future. Now we take them for granted and are far more advanced than our predecessors ever imagined. This is not an exception in history. At the turn of the last century, it was thought impossible that human flight could be achieved. Fast-forward half a century later and commercial flight services have become a dream turned into reality. The telegraph is another similar story, where within a few decades we advanced from horse-carried postal services to continent-spanning communication.

These advancements were not leaps, but gradual, yet fast, changes aided by developments in our understanding of nature. Long distance communication was made possible by the mathematical description of the electromagnetic field through Maxwell's equations. The first nonlinear circuit elements and semiconductor devices were built upon the description of quantum phenomena which was pioneered at the beginning of the 20th century. In each of these cases, the base of technology was a deeper understanding in the fundamental nature of reality itself.

Nowadays, we are living yet another revolution. Since the inception of quantum theory, certain phenomena made us challenge our description of nature. Superposition, quantum interference and entanglement, to name a few, proved that at the microscopic level objects do not play by the rules which we were used to. While for a long time these phenomena have not played any major role in technology, being seen in some cases as limitations (e.g. tunneling effects in transistors), they are now slowly emerging from the lab into the industrial sphere as resources to be exploited.

Quantum computing, quantum simulation, quantum communication and quantum sensing are just a few areas that have developed rapidly in the last decades. The mathematical formalism surrounding them is mature and proofs of concept have been developed, their scalability being the last question that needs to be answered. It is also very interesting how terms and concepts leak from science and technology into our daily lives. Everything is quantum in the world of today. We have popular culture promoting the "quantum" flag in movies and TV shows, corporations that sell "quantum" products and start-ups that bear "quantum" in their name even if they have nothing to do with the field. History shows us that this is not unexpected though. Speaking primarily for the science-fiction literary genre, the same happened for nuclear technology and nanotechnology. The early SF movies of the 50's dealt with the unknown by adding the vague "nuclear" tag, the

movies of the 90's and early 2000's had always the explanation of the nano-devices and now "quantum" rules as the explanatory plot-tool in mainstream media.

Does this mean that quantum theory is so well-versed in explaining the phenomena which we plan to put to practical use? Not in the slightest. We know that quantum theory is not the final picture for a variety of reasons, one common one being its incompatibility with general relativity. Even the foundations of the theory are in constant debate, with dozens of proposed interpretations. But this does not mean that the theory has not reached a level where it is sufficient for practical applications. We did not need an accurate characterization of electric current before putting it to use. Whether or not this is positive development for quantum science is left at each one's opinion.

This thesis is a review of some of the known results and some of the recent developments in the field of photonic quantum computing. The structure is as follows: we firstly present quantum computing and different strategies of implementation; then we present the physical realization of quantum computing using photons.

Quantum Computing

What is quantum computing? To put it simply, it is a paradigm that aims at using quantum phenomena (entanglement, interference, superposition etc.) to boost computing times over the capabilities of existing classical machines. If classical computers use classical systems to perform calculations (e.g. the current flowing through a transistor), quantum computers use quantum systems (e.g. electron spin, charge in a Cooper pair, single photons, etc.). Usually, we restrict ourselves to two-dimensional quantum systems to have the quantum counterpart of the bit, the qubit.

Quantum computers are starting to slowly emerge from the lab into the private sector, with some noisy small-scale devices being even available to the general public [1]. While these devices work as expected, the question that current research needs to answer is: are they scalable? Recently, there have been claims of quantum advantage, the point where a quantum device can massively outperform the best classical machines [2, 3]. While these claims remain highly debatable [4], it is very likely that we will see more of these results in the near future.

In this chapter, we will introduce the foundations of quantum computing. We start with presenting the mathematical formalism behind the qubit. Then we analyze the gate model of quantum computation. Finally, we present measurement based quantum computing.

2.1 The Qubit

The basic unit of quantum computation is the qubit, the quantum counterpart of the classical bit. On the most abstract level, the qubit can be thought of as a general two-level system which lies in a two-dimensional Hilbert space \mathcal{H} . A general state of one qubit is thus given by:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (2.1)$$

where the set $\{|0\rangle, |1\rangle\}$ denotes the computational basis and α and β are complex numbers, constrained by the normalization condition, i.e. $|\alpha|^2 + |\beta|^2 = 1$. As it can be seen, in contrast with the bit, which can only have a discrete number of states, namely 0 and 1, the qubit set of states is continuous.

We are led to a useful representation of qubit states by the normalization condition of eq. (2.1), which allows it to be rewritten as:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$$

Thus any qubit state can be mapped one-to-one onto a point $\{\theta, \phi\}$ on the \mathbb{R}^3 unit sphere, known in this case as the Bloch sphere, named after the physicist Felix Bloch [5]. By convention, the ± 1 points on the z-axis denote the computational basis, as shown in fig. 2.1. The factor of two in the trigonometric functions can be understood by looking at the density matrix of the pure qubit. In general, the density matrix can be written in the Pauli basis as:

$$\rho = r_0 I + \sum_{n=1}^3 r_n \sigma_n \quad (2.2)$$

where σ_n are the Pauli matrices. From $\text{Tr}(\rho) = 1$, r_0 is $\frac{1}{2}$, as the Pauli matrices are traceless. The computational basis is defined as the eigenbasis of σ_3 ; in this basis the density matrix has the representation:

$$\rho = \frac{1}{2} \begin{bmatrix} 1 + r_3 & r_1 - ir_2 \\ r_1 + ir_2 & 1 - r_3 \end{bmatrix} \quad (2.3)$$

where the r_n coefficients are rescaled from eq. (2.2). Consider now a pure qubit state:

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{2} \begin{bmatrix} 2\cos^2\frac{\theta}{2} & 2e^{-i\phi}\cos\frac{\theta}{2}\sin\frac{\theta}{2} \\ 2e^{i\phi}\cos\frac{\theta}{2}\sin\frac{\theta}{2} & 2\sin^2\frac{\theta}{2} \end{bmatrix} \quad (2.4)$$

By identification with eq. (2.3), we can see that $r_1 = \cos\phi\sin\theta$, $r_2 = \sin\phi\sin\theta$ and $r_3 = \cos\theta$. Thus every pure state is represented by a point on the Bloch sphere.

We define the purity of a state as $\mathcal{P} = \text{Tr}\rho^2$. By looking at the purity of the general state in eq. (2.2):

$$\text{Tr}(\rho^2) = \frac{1}{2}(1 + |\vec{r}|^2) \quad (2.5)$$

and since $\text{Tr}(\rho^2) \leq 1$, $|\vec{r}| \leq 1$. So the surface of the Bloch sphere represents pure states, while the interior of the sphere represents mixed states, with the totally mixed state ($1/2I$) at the middle of the sphere.

There are other noteworthy single-qubit states that should be mentioned. The first ones form the Hadamard basis, $\{|+\rangle, |-\rangle\}$:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (2.6)$$

and the next ones are related to the Hadamard states by a relative phase shift of i :

$$|\odot\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), \quad |\oslash\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \quad (2.7)$$

which are denoted here as the right circular state and the left circular state. All of these states are shown in fig. 2.1 on the Bloch sphere.

A group of one or more qubits is called a quantum register, the quantum counterpart of the classical bit register. If the register has N qubits, then the state of the register lies in $\mathcal{H}^{\otimes N}$. For example, let $N = 2$. The computational basis states of the register are then:

$$\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$$

For the rest of this thesis, unless otherwise stated, $|00\rangle = |0\rangle \otimes |0\rangle$, where the qubits are numbered from left to right.

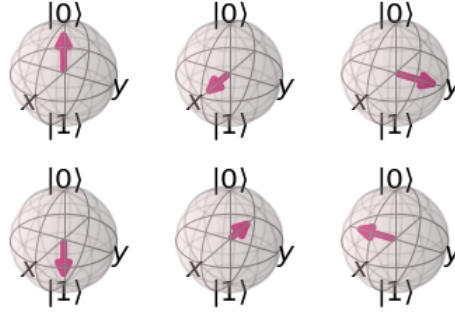


Figure 2.1: The Bloch representation of the computational basis (first column), the Hadamard basis (second column) and left and right circular basis (third column). Image generated using the Qiskit package from IBM [1].

2.2 The Gate Model

2.2.1 Single-Qubit Gates

At the most basic level of computation, one must be able to manipulate the state of at least one qubit. This is done through unitary operators which evolve the qubit state, often referred to as gates, which in general have the following parametrized form:

$$U = e^{i\alpha} \begin{bmatrix} e^{-i\phi_1} \cos \theta & -e^{-i\phi_2} \sin \theta \\ e^{i\phi_2} \sin \theta & e^{i\phi_1} \cos \theta \end{bmatrix} \quad (2.8)$$

Some of the most important of these are the Pauli gates, given here in the computational basis representation:

$$X = \sigma_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \sigma_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (2.9)$$

The action of these operators is straightforward: X acts as a logical NOT on the basis states, with $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$; Z acts as $Z|1\rangle = -|1\rangle$, while it leaves $|0\rangle$ unchanged - this introduction of a relative phase in the computational basis will be called a phase-flip; Y acts as a phase-flip followed by a NOT: $Y|0\rangle = i|1\rangle$, $Y|1\rangle = -i|0\rangle$.

The Z gate is a special case of a more general operator, called the phase-flip operator, denoted by $P(\varphi)$, which introduces a general relative phase φ in the computational basis:

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \quad (2.10)$$

Other special cases of the phase-flip operator include the S gate, with $Z = S^2$, and the T gate, with $S = T^2$:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \quad (2.11)$$

The exponentiation of the Pauli operators gives rise to another set of useful operators on single qubits, the rotation operators around the X , Y and Z axis respectively, with the

angle θ :

$$R_x(\theta) = e^{-iX\frac{\theta}{2}} = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \quad (2.12)$$

$$R_y(\theta) = e^{-iY\frac{\theta}{2}} = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \quad (2.13)$$

$$R_z(\theta) = e^{-iZ\frac{\theta}{2}} = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \quad (2.14)$$

In general, a rotation around an axis given by the unit vector \hat{n} with the angle θ is written as:

$$R_n(\theta) = e^{-i\hat{n} \cdot \vec{\sigma} \frac{\theta}{2}} = \cos \frac{\theta}{2} I - \sin \frac{\theta}{2} (\hat{n} \cdot \vec{\sigma}) \quad (2.15)$$

The action of a rotation on a general qubit state can be found by acting on the density matrix:

$$\rho' = R_n(\theta) \rho R_n(-\theta) = R_n(\theta) \left[\frac{1}{2} (I + \vec{r} \cdot \vec{\sigma}) \right] R_n(-\theta) = \frac{1}{2} (I + \vec{r}' \cdot \vec{\sigma}) \quad (2.16)$$

with $\vec{r}' = \cos \theta \vec{r} + \sin \theta (\hat{n} \times \vec{r}) + (1 - \cos \theta) (\vec{r} \cdot \hat{n}) \hat{n}$. So the rotation operator rotates the Bloch vector around \hat{n} with an angle θ , which justifies the factor of $1/2$ from the definition.

Rotations provide an interesting way of representing single-qubit unitaries U , since any U can be written as [6]:

$$U = e^{iA} \quad (2.17)$$

with A a 2×2 Hermitian matrix. We can write $A = \alpha I + \frac{\theta}{2} \hat{n} \cdot \vec{\sigma}$, where α, θ are real numbers and \hat{n} is a unit vector in 3D space. Plugging this back in the previous equation gives the form:

$$U = e^{i\alpha} R_n(\theta) \quad (2.18)$$

Therefore any single-qubit gate can be written as a rotation of angle θ around an axis \hat{n} . By comparing this result with equation eq. (2.8), we can see that the number of free parameters is the same, as the axes can be chosen arbitrarily. If instead we consider multiple rotations around a fixed axis, we can find many interesting decompositions. An example is [7]:

Theorem 2.2.1 (Z-X Decomposition). *Let U be a unitary operator on a single qubit. Then there exist $\alpha, \beta, \gamma, \delta$ real numbers such that:*

$$U = e^{i\alpha} R_z(\beta) R_x(\gamma) R_z(\delta) \quad (2.19)$$

which can be verified by direct multiplication.

Another oftenly used gate is the Hadamard gate, the gate that diagonalizes the X gate in the computational basis:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.20)$$

The effect of this gate is to put the qubit into an equal superposition: $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$. The following are some direct identities which involve the Hadamard gate and other single qubit-gates:

$$HXH = Z \quad (2.21)$$

$$HZH = X \quad (2.22)$$

$$HYH = -Y \quad (2.23)$$

$$HTH = R_x(\frac{\pi}{4}) \quad (2.24)$$

A quantum circuit is a qubit register together with the gates acting on it. A quantum circuit has a graphical notation which follows some specific conventions¹: time flows from left to right, qubits are represented by a continuous line and start initialized in $|0\rangle$ if not stated otherwise and an arbitrary gate U acting on a qubit is marked as:

$$\text{---} \boxed{U} \text{---}$$

2.2.2 Multi-qubit Gates

One of the basic resources used in quantum computation is quantum entanglement, a phenomenon in which the state of two or more qubits cannot be described individually, even if spatially separated. Quantum entanglement cannot be generated with the single-qubit gates discussed previously. The qubits start in $|0\rangle^{\otimes n}$, a state with no entanglement, and local operations in the form of single-qubit gates cannot increase entanglement [9]. Non-local operations are needed, acting on the entirety of the qubit register.

In this context, we talk now about controlled gates. Controlled gates are gates which operate as $C^n(U) |c_1 c_2 \dots c_n\rangle |t\rangle = |c_1 c_2 \dots c_n\rangle U^{c_1 c_2 \dots c_n} |t\rangle$, where $|t\rangle$ can be multi-qubit register. $|c\rangle$ is the control register and $|t\rangle$ is the target register. For the case when the size of these registers is one, we will adopt the notation $C_{i,j}(U) |i\rangle |j\rangle = |i\rangle U^i |j\rangle$ and we will drop the indices when the target and control can be understood from context.

One of the basic controlled operations is $C(Z)$:

$$\begin{array}{c} \bullet \\ | \\ \text{---} \boxed{Z} \text{---} \end{array} = \begin{array}{c} \bullet \\ | \\ \bullet \end{array}$$

In the computational basis it has the matrix form:

$$C(Z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (2.25)$$

Using eq. (2.22), the following relation is obtained:

$$\begin{array}{c} \bullet \\ | \\ \text{---} \boxed{X} \text{---} \end{array} = \begin{array}{c} \bullet \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \\ | \\ \text{---} \boxed{H} \text{---} \bullet \text{---} \boxed{H} \text{---} \end{array} \quad (2.26)$$

¹This is a case of Penrose graphical tensor notation [8].

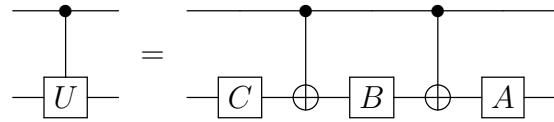
If U_i denotes the unitary U acting on qubit i , then the circuit identity eq. (2.26) can be written as $C_{1,2}(X) = H_2 C_{1,2}(Z) H_2$. The controlled- X , or CNOT, has the following matrix form:

$$C(X) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.27)$$

In general, any two-qubit controlled operation $C(U)$ has the block-diagonal form:

$$C_{1,2}(U) = \begin{bmatrix} I_2 & 0 \\ 0 & U \end{bmatrix} \quad (2.28)$$

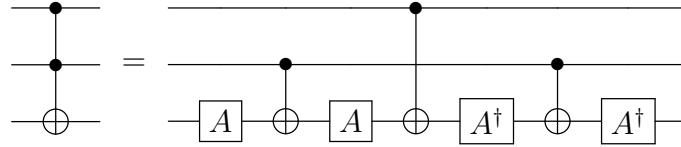
where I_n is the $n \times n$ identity matrix. Any $C(U)$ gate can be decomposed into CNOTs and single-qubit gates [7]:



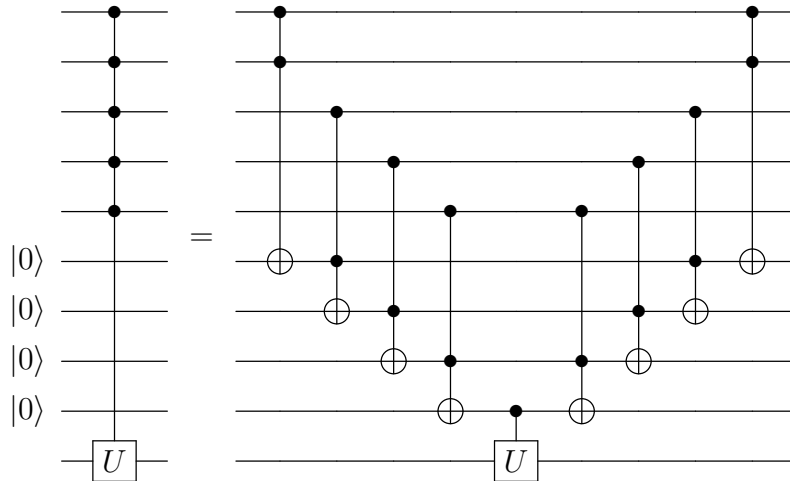
$$\text{Controlled-}U = \text{CNOT}_{1,2} \circ C \circ \text{CNOT}_{2,1} \circ B \circ \text{CNOT}_{1,2} \circ A \quad (2.29)$$

with $U \cong AXBXC$, where \cong means that the equality holds up to a global phase.

In the case of multi-controlled gates, an important example is the Toffoli gate, or CCNOT, which has two controls and a target. This gate executes an X on the target qubit if both controls are $|1\rangle$. Reference [7] gives a decomposition of the Toffoli in single-qubit gates and CNOTs:



up to a global phase, with $A = R_y(\frac{\pi}{4})$. Using Toffoli gates, one can construct a scheme for the general $C^n U$ case[10]:



It is straightforward to see why this works. Notice that if the target qubit is $|0\rangle$ then the Toffoli copies $|c_1 \cdot c_2\rangle$ onto it. We exploit this using ancilla qubits initialized in $|0\rangle$. After the first set of Toffolis, the last ancilla is in the state $|c_1 \cdot c_2 \cdot \dots \cdot c_n\rangle$ and a $C(U)$ applies U conditioned on this state. Finally, Toffolis are used again to uncompute the ancillas. More decompositions in terms of single and two-qubit gates can be found in [7, 10].

2.2.3 Universality

In classical computing, a small set of gates can be used to compute arbitrary functions (e.g. AND, OR, NOT). We say that the respective set is universal to classical computing. In quantum computing, similar results can be proven, where a small set of quantum gates can be used to construct arbitrary operations. In contrast to the classical operations however, quantum operations are continuous, leading us to define two types of universal gate sets:

1. continuous universal sets, where some gates in the set depend on a continuously varying parameter (e.g. rotations). In this case, any arbitrary operations can be exactly decomposed into elements of the universal set.
2. discrete universal sets, where all the gates are fixed. A discrete set is universal to quantum computation if any arbitrary quantum operation can be decomposed into elements of the set up to an arbitrarily small error.

We firstly present a continuous universal set. We have already seen in eq. (2.19) that rotations can be used to construct any single qubit gate. In 1995, DiVincenzo showed that single qubit gates together with CNOTs can be used to construct any two-level unitary [11]. A two-level unitary is an operation which acts non-trivially only on a 2-dimensional subspace of a register.

Let the two-level unitary be U and $|a_1\rangle$ and $|a_2\rangle$ be the basis states on which U acts non-trivially, with a_1 and a_2 binary numbers. Let \tilde{U} be the non-trivial submatrix of U , a two qubit gate, which can be implemented by rotations. The proof uses Gray codes connecting a_1 and a_2 . A Gray code g of a_1 and a_2 is a sequence of binary numbers, starting with a_1 and ending with a_2 , such that any consecutive numbers differ by one bit. As an example, a Gray code of 00101 and 11001 is:

$$\begin{aligned} g_1 &= 00101 \\ g_2 &= 00001 \\ g_3 &= 01001 \\ g_4 &= 11001 \end{aligned}$$

Let m be the length of the Gray code connecting a_1 and a_2 . Notice that m can be at most equal $n + 1$, where n is the number of qubits in the register. To build U , firstly implement the swaps:

$$|g_1\rangle \rightarrow |g_2\rangle \rightarrow \cdots \rightarrow |g_{m-1}\rangle$$

using CNOTs. Then perform a controlled- \tilde{U} on the qubit that differs between $|g_{m-1}\rangle$ and $|g_m\rangle$, with the rest of the qubits as controls. Finally, uncompute the initial swaps:

$$|g_{m-1}\rangle \rightarrow \cdots \rightarrow |g_2\rangle \rightarrow |g_1\rangle$$

The next example is taken from [10]. Say that we want to perform the following two-level

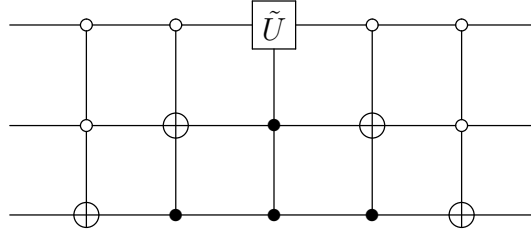
unitary on a register with three qubits:

$$U = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 0 & 0 & 0 & 0 & d \end{bmatrix}.$$

with $\tilde{U} = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$. U acts non-trivially on the states $|000\rangle$ and $|111\rangle$, which are connected by the Grey code:

$$\begin{aligned} g_1 &= 000 \\ g_2 &= 001 \\ g_3 &= 011 \\ g_4 &= 111 \end{aligned}$$

We can now use CNOTs to shuffle $|g_1\rangle \rightarrow |g_2\rangle \rightarrow |g_3\rangle$, then apply the controlled- \tilde{U} operation on the qubit that differs in $|g_3\rangle$ and $|g_4\rangle$, conditioned on the state $|11\rangle$ of the other qubits. At the end we undo the initial shuffling to complete the implementation of U :



Here the open circle indicates that the control is reversed (the control qubit has to be in the state $|0\rangle$ instead of $|1\rangle$). All of these operations can be constructed with the methods presented in the last section. To complete the universality proof, we need to show now that two-level unitaries can be used to construct any arbitrary unitary. This result was proved by Reck *et al.* in 1994 [12]. Let us consider for the beginning a 3×3 arbitrary unitary (example taken from [10]):

$$U = \begin{bmatrix} a & d & f \\ b & e & g \\ c & f & h \end{bmatrix}$$

We will find U_1, U_2, U_3 two-level matrices such that $U_3 U_2 U_1 U = I$ or $U = U_1^\dagger U_2^\dagger U_3^\dagger$. We achieve this by nulling b and c . If $b = 0$ then $U_1 = I$. Else, set U_1 to:

$$U_1 \equiv \begin{bmatrix} \frac{a^*}{\sqrt{|a|^2+|b|^2}} & \frac{b^*}{\sqrt{|a|^2+|b|^2}} & 0 \\ \frac{b}{\sqrt{|a|^2+|b|^2}} & \frac{-a}{\sqrt{|a|^2+|b|^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In either cases, the result is:

$$U_1 U = \begin{bmatrix} a' & d' & g' \\ 0 & e' & h' \\ c' & f' & j' \end{bmatrix}$$

Similarly, if $c = 0$, set U_2 to:

$$U_2 \equiv \begin{bmatrix} a'^* & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Otherwise, set U_2 to:

$$U_2 \equiv \begin{bmatrix} \frac{a'^*}{\sqrt{|a'|^2 + |c'|^2}} & 0 & \frac{c'^*}{\sqrt{|a'|^2 + |c'|^2}} \\ 0 & 1 & 0 \\ \frac{c'}{\sqrt{|a'|^2 + |c'|^2}} & 0 & \frac{-a'}{\sqrt{|a'|^2 + |c'|^2}} \end{bmatrix}$$

In either cases we arrive at:

$$U_2 U_1 U = \begin{bmatrix} 1 & d'' & g'' \\ 0 & e'' & h'' \\ 0 & f'' & j'' \end{bmatrix}$$

where, due to $U_2 U_1 U$ being unitary, $d'' = g'' = 0$. Now set U_3 equal to:

$$U_3 \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & e''^* & f''^* \\ 0 & h''^* & j''^* \end{bmatrix}$$

which completes $U_3 U_2 U_1 U = I$.

Consider now that U is a $d \times d$ matrix. We can use a similar procedure to make the first column of the matrix equal to $(1, 0, 0 \dots 0)$. We then repeat this procedure for the remaining $(d-1) \times (d-1)$, $(d-2) \times (d-2)$, etc. submatrices, until we are left with:

$$U = \prod_{i=1}^{d(d-1)/2} V_i$$

where V_i is a two-level unitary. This completes the universality proof, showing that any arbitrary unitary can be decomposed into CNOTs and single qubit gates.

We present now a discrete universal set. Before this however, some notations should be introduced. Let us say that we want to implement U unitary operator, but instead we implement V . The error when V is implemented instead of U is defined by [10]:

$$E(U, V) = \max_{|\psi\rangle} \|(U - V)|\psi\rangle\| \quad (2.30)$$

We can interpret this in the following way. Let M be a POVM element associated with a measurement outcome. Then $E(U, V)$ encapsulates the difference between the result statistics upon measuring M after applying U or V , respectively. Indeed, using the Cauchy-Schwartz inequality, it can be shown that:

$$|P_U - P_V| \leq 2E(U, V) \quad (2.31)$$

where P_U and P_V are the probabilities of measuring M after applying U or V . Moreover, let us consider the case where we want to apply a sequence of operators $(U_k)_{k=\overline{1,n}}$, but instead we implement $(V_k)_{k=\overline{1,n}}$. Using the triangle inequality, it can be proven that the errors add at most linearly:

$$E(U_n U_{n-1} \cdots U_1, V_n V_{n-1} \cdots V_1) \leq \sum_{i=1}^n E(U_i, V_i) \quad (2.32)$$

The last two results are important in building quantum circuits. If we want to implement a circuit $U_1 U_2 \cdots U_k$ with an error δ , then we should implement each operation with an error of at most $\frac{\delta}{2k}$, due to eq. (2.31) and eq. (2.32).

With these mathematical tools, we analyze now the universality proof for the discrete set composed of the Hadamard, CNOT, phase and T gates (phase gates are used to make the set fault-tolerant, but that will not be considered here). This set was proved to be universal by Boykin *et al.* in 1999 [13].

We first prove that H and T are enough to approximate any single qubit gate up to an arbitrary small error $\epsilon > 0$. Note that $T \cong R_z(\frac{\pi}{4})$ and $HTH \cong R_x(\frac{\pi}{4})$. Their product is thus:

$$R_z(\frac{\pi}{4})R_x(\frac{\pi}{4}) = \cos^2 \frac{\pi}{8} I - i \left[\cos \frac{\pi}{8} (X + Z) + \sin \frac{\pi}{8} Y \right] \sin \frac{\pi}{8} = R_{\hat{n}}(\theta)$$

where $\hat{n} = (\cos \frac{\pi}{8}, \sin \frac{\pi}{8}, \cos \frac{\pi}{8})$ and $\cos \frac{\theta}{2} = \cos^2 \frac{\pi}{8}$. It can be proven [13] that θ is an irrational multiple of 2π . Thus any phase can be accurately approximated by an integer multiple of θ :

$$e^{in\theta} = e^{i\phi}$$

or, equivalently, the sequence $(\theta_k)_{k \geq 0}$, $\theta_k = k\theta \bmod 2\pi$ is dense on $[0, 2\pi)$. Therefore, for any $\epsilon > 0$ and angle α , we can find a positive integer n such that:

$$E(R_{\hat{n}}(\alpha), R_{\hat{n}}^n(\theta)) < \frac{\epsilon}{3} \quad (2.33)$$

It can be proven by direct multiplication that $HR_{\hat{n}}(\theta)H = R_{\hat{m}}(\theta)$, where $\hat{m} = (\cos \frac{\pi}{8}, -\sin \frac{\pi}{8}, \cos \frac{\pi}{8})$. Using the same argument as above, we conclude that for any $\epsilon > 0$ and angle α , we can find a positive integer n such that:

$$E(R_{\hat{m}}(\alpha), R_{\hat{m}}^n(\theta)) < \frac{\epsilon}{3} \quad (2.34)$$

But any single qubit gate U can be decomposed into rotation around the axis \hat{n} and \hat{m} (see theorem 2.2.1):

$$U = R_{\hat{n}}(\beta)R_{\hat{m}}(\gamma)R_{\hat{n}}(\delta)$$

where, from eqs. (2.31), (2.33) and (2.34), we arrive at:

$$E(U, R_{\hat{n}}(\theta)^{n_1} H R_{\hat{n}}(\theta)^{n_2} H R_{\hat{n}}(\theta)^{n_3}) < \epsilon \quad (2.35)$$

for suitable integers n_1 , n_2 and n_3 . Sequences of H and T can thus approximate any unitary operation to an arbitrary error.

The final step in the universality proof is to observe that the CNOT, Hadamard and T gates can be used to construct any arbitrary unitary U . Indeed, as we are able to approximate any single qubit gate up to an error ϵ , we are also able to approximate any two-level unitary up to the same error, which can be used to construct U . If we want to implement U with an error ϵ , then we implement each of the m single qubit gates used in the construction of U with an error $\frac{\epsilon}{m}$ and use eq. (2.32).

There is a remarkable theorem regarding the universality of discrete sets of gates:

Theorem 2.2.2 (Solovay-Kitaev). *Let G be a finite set of quantum gates, such that the group it generates is dense on $SU(2)$. Then for any $\epsilon > 0$ and unitary U , there exist $S \subset G$ and c a constant with $|S| = O(\log^c(1/\epsilon))$, such that $E(U, \prod_{s_i \in S} s_i) < \epsilon$.*

A proof of the theorem can be found in [14]. This result is extremely important in quantum computing as a direct consequence is that a quantum circuit built from m unitary operations can be built from $O(m \log^c(m/\epsilon))$ gates [10].

2.3 Measurement-Based Quantum Computing

2.3.1 The Stabilizer Formalism

In general, the quantum state of n qubits can be written as:

$$|\psi\rangle = \sum_{i_1 i_2 \dots i_n=1}^n c_{i_1 i_2 \dots i_n} |i_1 i_2 \dots i_n\rangle \quad (2.36)$$

implying that an accurate description of this state requires 2^n complex coefficients, exponential resources in the size of the system. There is a subspace though in the Hilbert space of n qubits which requires only polynomial resources and it is described in terms of the stabilizer formalism, introduced by Gottesman [15]. The presentation given here is due to Fujii [16].

Firstly we introduce the stabilizer group. The Pauli group on n qubits is defined as:

$$\mathbb{P}_n = \{\pm 1, \pm i\} \times \{I_2, X, Y, Z\}^{\otimes n} \quad (2.37)$$

An n -qubit stabilizer group is a commutative, Hermitian subgroup of the n -qubit Pauli group:

$$\mathcal{S}_n = \{\mathcal{S}_j | \mathcal{S}_j \in \mathbb{P}_n, \mathcal{S}_j = \mathcal{S}_j^\dagger, [\mathcal{S}_j, \mathcal{S}_k] = 0\} \quad (2.38)$$

The hermiticity condition guarantees that the eigenvalues of the elements of \mathcal{S}_n are ± 1 . An element of the stabilizer group is called a stabilizer operator. A maximal independent set of stabilizer operators is called a stabilizer generator set, denoted as $\mathcal{S}_n^{(g)}$, $\mathcal{S}_n = \langle \mathcal{S}_n^{(g)} \rangle$. For a given stabilizer set, the stabilizer state is defined as the common eigenstate of all the stabilizer operators with eigenvalue $+1$:

$$S_j |\mathcal{S}\rangle = |\mathcal{S}\rangle, \forall S_j \in \mathcal{S}_n \quad (2.39)$$

The trivial action of the stabilizer operators is called stabilization. It is sufficient for the state to be stabilized only by the generators:

$$\mathcal{S}_j |\mathcal{S}\rangle = |\mathcal{S}\rangle, \forall \mathcal{S}_j \in \mathcal{S}_n^{(g)}$$

as this implies eq. (2.39) directly. As an example consider the following Bell state:

$$|\phi^+\rangle = \frac{1}{2}(|00\rangle + |11\rangle)$$

The state is stabilized by:

$$\{II, XX, ZZ, -YY\}$$

where from now on $AB = A \otimes B$ if not stated explicitly. One possibility for the generators of this set is $\langle XX, ZZ \rangle$, as both II and $-YY$ can be written as a product of these operators.

Let the number of generators in a generator set of n qubits be k . Each generator splits the Hilbert space into two subspaces, according to the ± 1 eigenvalues, leaving a subspace of dimensionality 2^{n-k} unchanged. If $k = n$ then the stabilizers pinpoint a specific state. On the other hand, if $k < n$, there are 2^{n-k} degrees of freedom in the system which can be used for encoding purposes [17, 18, 15]. As there can be at most n generators for a state of n qubits, the stabilizer formalism provides an efficient way of describing the stabilizer states.

From eq. (2.39) one can easily reconstruct the stabilizer state in eq. (2.36) from the stabilizer generators. As the stabilizer state is the common $+1$ eigenket of the stabilizer generators, a random initial state can be projected onto the stabilizer state by successive projections on the $+1$ eigenspaces of the stabilizer generators:

$$|\mathcal{S}\rangle \cong \prod_{j=1}^n \frac{I + \mathcal{S}_j}{2} |0\rangle^{\otimes n} \quad (2.40)$$

where \cong implies that the states are the same up to global phase factors.

There is a class of unitary operators which act efficiently on the stabilizer states, the Clifford group of operators. Defined as:

$$C = \{U \mid U^\dagger U = I, UP_j U = P'_j, \forall P_j, P'_j \in P_n\} \quad (2.41)$$

with $UP_j U = P'_j$ or $UP_j = P'_j U^\dagger$ known as the propagation relation, they act on the stabilizer states as:

$$U |\mathcal{S}\rangle = U \mathcal{S}_j |\mathcal{S}\rangle = U \mathcal{S}_j U^\dagger U |\mathcal{S}\rangle = \mathcal{S}'_j U |\mathcal{S}\rangle \quad (2.42)$$

where $\mathcal{S}'_j = U \mathcal{S}_j U^\dagger$. Thus $U |\mathcal{S}\rangle$ is still a stabilizer state, stabilized by $\{\langle \mathcal{S}'_j \rangle\}$, which respects conditions eq. (2.38), as U is unitary. In this way we can describe the action of the operator U only with the initial stabilizer group conjugated with respect to U .

Some important Clifford operation were already introduced in sections 2.2.1 and 2.2.2. The Hadamard operator satisfies the conditions for a Clifford operation as it propagates through the Pauli operators as shown in eqs. (2.21) to (2.23):

$$HX = ZH$$

$$HZ = XH$$

$$HY = -YH$$

The S operator has similar propagation relations:

$$XS = SY \quad (2.43)$$

$$ZS = SZ \quad (2.44)$$

The $C(X)$ operation is also a Clifford operation as the following propagation relations can be verified through direct multiplication:

$$C_{1,2}(X)X_1 = X_1X_2C_{1,2}(X) \quad (2.45)$$

$$C_{1,2}(X)X_2 = X_2C_{1,2}(X) \quad (2.46)$$

$$C_{1,2}(X)Z_1 = Z_1C_{1,2}(X) \quad (2.47)$$

$$C_{1,2}(X)Z_2 = Z_1Z_2C_{1,2}(X) \quad (2.48)$$

Note that the propagation relation involving Y is directly implied as $Y = -iXZ$. Similar propagation relations are found for $C(Z)$:

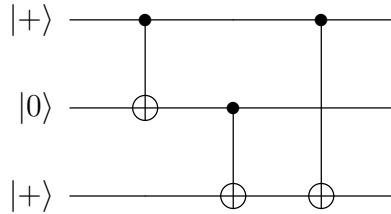
$$C_{1,2}(Z)Z_1 = Z_1C_{1,2}(Z) \quad (2.49)$$

$$C_{1,2}(Z)Z_2 = Z_2C_{1,2}(Z) \quad (2.50)$$

$$C_{1,2}(Z)X_2 = Z_1X_2C_{1,2}(Z) \quad (2.51)$$

$$C_{1,2}(Z)X_1 = X_1Z_2C_{1,2}(Z) \quad (2.52)$$

Clifford circuits are easy to evaluate as one has to keep track of n stabilizer operators instead of 2^n state amplitudes. As an example, consider the circuit:



Direct computation yields the final state:

$$|\psi\rangle = \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle + |111\rangle)$$

But this method is cumbersome as the circuit becomes larger and deeper. If we consider the initial stabilizer group:

$$\langle X_1, Z_2, X_3 \rangle$$

we can keep track of the evolution of these operators under the Clifford circuit using the appropriate propagation relations. The final stabilizers are:

$$\langle X_1X_2, Z_1Z_2, X_3 \rangle$$

which are exactly the operators that stabilize $|\psi\rangle$.

Next let us see how Pauli measurements are handled in the stabilizer formalism. Let M denote the Pauli operator that is to be measured on a stabilizer state. We distinguish between two possibilities:

- M commutes with all the stabilizer generators. Then the state is unchanged, as an M measurement projects onto the same subspace as all other stabilizers.
- M anticommutes with at least one stabilizer generator. Then we can find another generator set such that M anticommutes with only one generator. If the measurement outcome is $m \in \{\pm 1\}$, the state after the measurement is the same as before the measurement, up to the generator which did not commute, which is replaced by mM .

An illustrative example of Pauli measurements on stabilizer states is the entanglement swap protocol given in the stabilizer formalism. The purpose of entanglement swapping is to build long range entanglement from lower range entangled states [19]. We start with two Bell states, stabilized by:

$$\langle X_1X_2, Z_1Z_2, X_3X_4, Z_3Z_4 \rangle$$

Next we perform the following Pauli measurements on qubits 3 and 4 (a Bell basis measurement):

$$\langle X_2X_3, Z_2Z_3 \rangle$$

Note that the measurement X_2X_3 , with outcome m_1 , anticommutes with Z_1Z_2 and Z_3Z_4 , but it commutes with X_1X_2 and X_3X_4 . Thus, after this measurement, the generators are:

$$\langle X_1X_2, m_1X_2X_3, X_3X_4, Z_1Z_2Z_3Z_4 \rangle$$

A similar case is found now for Z_2Z_3 , as it anticommutes with X_1X_2 and X_3X_4 , but it commutes with $X_1X_2X_3X_4$. If the measurement outcome is m_2 , the final stabilizer state is:

$$\langle m_2Z_2Z_3, m_1X_2X_3, X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$$

which is equivalent to:

$$\langle m_2Z_2Z_3, m_1X_2X_3, m_1X_1X_4, m_2Z_1Z_4 \rangle$$

So, depending on the measurement outcomes, the qubit pairs 1 – 4 and 2 – 3 are found in a specific Bell state, in contrast with the initial state (1 – 2 and 3 – 4). The same result could have been found by simply applying the measurement projectors onto the initial state, but the same argument as for the Clifford operations is valid: this becomes hard as the number of qubits grows as we would have to keep track of at most 2^n state amplitudes.

The results regarding the efficiency of the stabilizer formalism and Clifford operations have led D. Gottesman to the following theorem in 1998, which he attributed to E. Knill in a private communication [20]:

Theorem 2.3.1 (Knill-Gottesman). *Any quantum computer performing only:*

- Clifford group gates*
- measurements of Pauli group operators*
- Clifford group operations conditioned on classical bits, which may be the results of earlier measurements*

can be perfectly simulated in polynomial time on a probabilistic classical computer.

Simulation here means that the classical computer can sample from the probability distribution given by the final state of the quantum circuit. We also suppose that the quantum computation starts in a stabilizer state and that the architecture permits feedforward controlled operations. Observe however that, according to section 2.2.3, Clifford gates only are not universal to quantum computation, a consequence of the theorem being that non-Clifford operations are needed for quantum exponential speed-ups, like the T gate. Theorem 2.3.1 also implies that the space of stabilizer states, an exponentially large subspace of $\mathcal{H}^{\otimes n}$, is tractable to classical machines.

2.3.2 Graph and cluster states

An important class of stabilizer states used as a primitive resource in MBQC are the cluster states, introduced by Briegel and Raussendorf in 2001 [21]. Cluster states are a particular case of a more general class of states known as graph states. As shown in [22], any stabilizer state is locally equivalent to a graph state.

A graph state is described by a graph $G(V, E)$, with V the set of vertices and E the set of edges, with each vertex representing a qubit. The graph state is obtained by first initializing every qubit in the $|+\rangle$ state and then applying $C(Z)$ operations between the qubits according to the connectivity of the graph G . As $C(Z)$ is symmetric, the graph does not need to be directed. The resulting state is written as:

$$|G\rangle = \prod_{(i,j) \in E} C_{i,j}(Z) |+\rangle^{\otimes n} \quad (2.53)$$

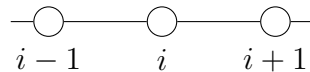
where $n = |V|$ is the number of qubits.

Graph states can be described in the stabilizer formalism, as eq. (2.53) describes a Clifford circuit acting on a stabilizer state. If $N(i)$ denotes the neighbours of the i -th vertex in the graph G , then the stabilizer generators of the graph state are given by:

$$\mathcal{S}_i = X_i \prod_{j \in N(i)} Z_j \quad (2.54)$$

If the graph is a regular lattice then the graph state is known as a cluster state.

Let us analyze what happens to graph states under Pauli measurements. For simplicity, consider a 1D cluster state:



with stabilizers:

$$\langle \dots, Z_{i-2}X_{i-1}Z_i, X_iZ_{i-1}Z_{i+1}, Z_iX_{i+1}Z_{i+2}, \dots \rangle$$

A measurement Z_i with outcome 1 on the i -th qubit anticommutes only with the stabilizer $\mathcal{S}_i = X_iZ_{i-1}Z_{i+1}$. The state after measurement is:

$$\langle \dots, Z_{i-2}X_{i-1}, Z_{i+2}X_{i+1}, \dots \rangle, \langle Z_i \rangle$$

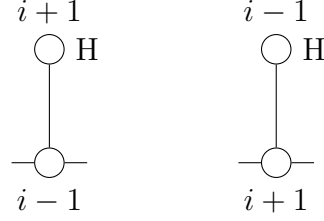
such that the qubit i is isolated and the cluster state is split into two:



A measurement X_i on the other hand commutes with \mathcal{S}_i , but anticommutes with \mathcal{S}_{i-1} and \mathcal{S}_{i+1} . Changing the stabilizer generators yields the final result:

$$\langle \dots, Z_{i-2}X_{i-1}Z_{i+2}X_{i+1}, Z_{i-1}Z_{i+1}, \dots \rangle, \langle X_i \rangle$$

This can be put in the form of eq. (2.54) using a Hadamard on the qubit $i+1$ or $i-1$, thus obtaining the two equivalent results:



If the qubit $i-1$ (or $i+1$) is measured in the X basis as well, that removes the respective qubit from the cluster state as the only stabilizer that anticommutes with X_{i-1} (X_{i+1}) is $Z_{i-1}Z_{i+1}$. This will be called an even contraction:



An odd contraction is possible by using Y measurements instead of X measurements. A detailed analysis of this type of contraction is given in [16].

2.3.3 Implementation of Single and Two-Qubit Gates

Gates in MBQC are implemented on a cluster state using adaptive measurements. To see how this can be achieved, we will firstly describe quantum teleportation, a building block of MBQC.

Let the Bell states be denoted by:

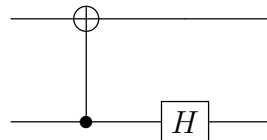
$$|B_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad (2.55)$$

$$|B_{01}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \quad (2.56)$$

$$|B_{10}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \quad (2.57)$$

$$|B_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \quad (2.58)$$

such that $|B_{ij}\rangle = X_1^i Z_1^j |B_{00}\rangle$ and the following circuit maps $|B_{ij}\rangle$ to $|ij\rangle$:



Quantum teleportation is a quantum information transfer protocol, proposed by Bennet *et al.* in 1993 [23]. Consider two parties, Alice and Bob, connected by a classical communication channel and sharing the Bell state $|B_{00}\rangle$. Alice has a qubit in an unknown state $|\psi\rangle$ and wants to send it to Bob. To do so she measures her qubits in the Bell basis and sends the result to Bob via the classical channel. Bob then retrieves the initial state by performing specific local operations on his qubit.

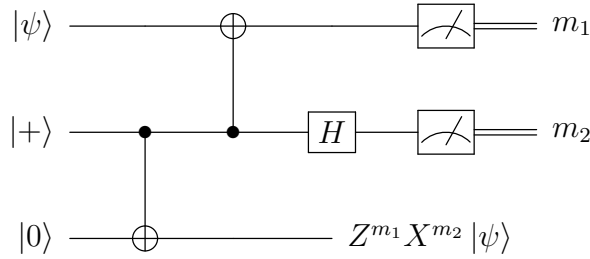
Let the to-be transmitted state be indicated with subscript 1 and the states of the shared qubits with subscript 2 (Alice) and 3 (Bob), respectively. The initial total state is:

$$|\psi_1\rangle \frac{|0_2\rangle |0_3\rangle + |1_2\rangle |2_3\rangle}{\sqrt{2}}$$

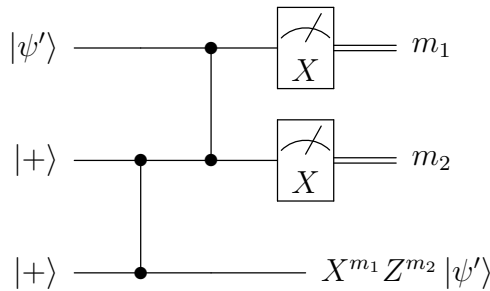
The final state on Bob's side after the measurement, denoted by $|\psi'\rangle$ can be obtained by projecting this state onto the Bell basis in the first and second subsystems:

$$|\psi'_3\rangle \cong \frac{\langle 0_1 | \langle 0_2 | + \langle 1_1 | \langle 1_2 |}{\sqrt{2}} Z_1^j X_1^i |\psi_1\rangle \frac{|0_2\rangle |0_3\rangle + |1_2\rangle |2_3\rangle}{\sqrt{2}} = \frac{1}{2} Z_3^j X_3^i |\psi_3\rangle$$

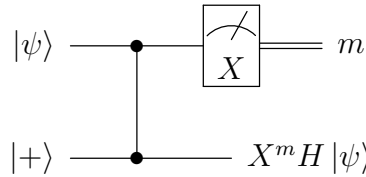
Bob can now perform $X_3^i Z_3^j$ on his qubit to retrieve the state sent by Alice, if he knows the measurement results. This process is represented in the following circuit diagram:



where a gate followed by a classical wire represents a measurement in the basis defined by the gate and the measurement outcomes are 0 or 1. We can now use eq. (2.26) to rewrite the circuit:

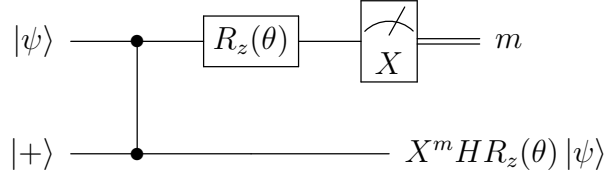


which can be seen to be composed of two identical circuits of the form:



which is sometimes called a one-bit teleportation. The proof that the final state on the second qubit after the measurement is $X^m H |\psi\rangle$ is analogous with the proof for the usual teleportation protocol. If we apply a rotation around the z-axis to the state ψ before the

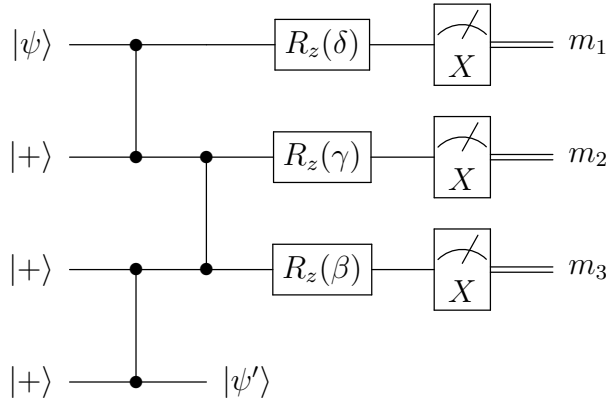
teleportation the final state will be $X^m H R_z(\theta) |\psi\rangle$. But $R_z(\theta)$ commutes with $C(Z)$, so the following circuit has the same action on the register:



Thus by measuring a specific operator on the first qubit, the action of the rotation on the state $|\psi\rangle$ is teleported to the second qubit, up to local operations dependent on the measurement outcome. The operator measured on the first qubit is $R_z(\theta) X R_z(-\theta)$, which is equivalent to $R_z(2\theta) X$ due to the relation:

$$R_z(\theta) X = X R_z(-\theta) \quad (2.59)$$

The operator will not be specified under the measurement for clarity. We can now present the method by which any single-qubit gate can be constructed in MBQC. Consider the circuit:



The state $|\psi'\rangle$ is built from three one-bit teleportations, meaning that:

$$|\psi'\rangle = X^{m_3} H R_z(\beta) X^{m_2} H R_z(\gamma) X^{m_1} H R_z(\delta) |\psi\rangle$$

We can propagate the X-gates through the rotations and Hadamard gates to obtain:

$$|\psi'\rangle = X^{m_3 \oplus m_1} Z^{m_2} H R_z((-1)^{m_2} \beta) H R_z((-1)^{m_1} \gamma) H R_z(\delta) |\psi\rangle$$

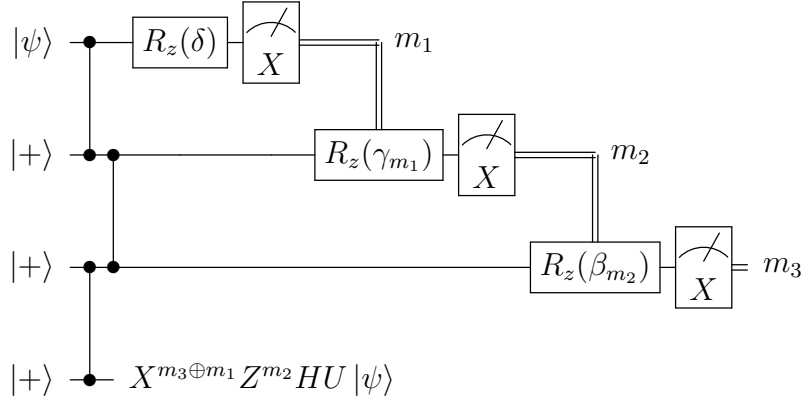
where \oplus denotes addition modulo 2. According to eq. (2.19) however, for any unitary matrix U there exist real numbers β' , γ' and δ' such that:

$$U \cong R_z(\beta') H R_z(\gamma') H R_z(\delta')$$

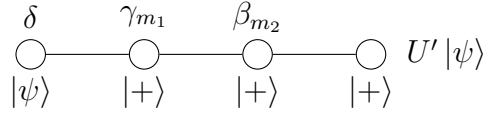
So if we choose $\beta' = (-1)^{m_2} \beta = \beta_{m_2}$, $\gamma' = (-1)^{m_1} \gamma = \gamma_{m_1}$ and $\delta' = \delta$ then the final state is:

$$|\psi'\rangle = X^{m_3 \oplus m_1} Z^{m_2} H U |\psi\rangle$$

This can be achieved by adaptively changing the operator measured at each step:



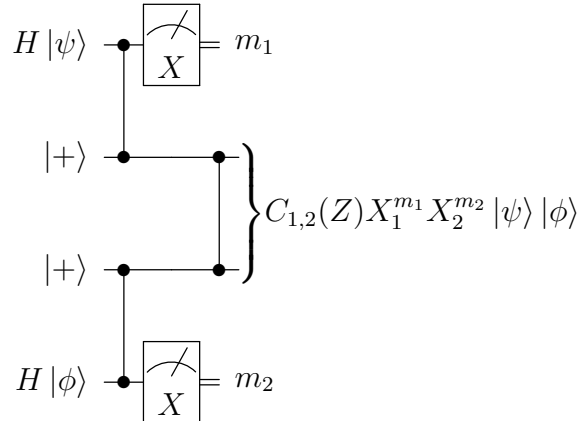
This circuit can be put in a convenient graph form, noticing that before measurements we have a four qubit graph state:



where the measurements have been denoted only by the measurement angles and $U' = X^{m_3 \oplus m_1} Z^{m_2} H U$. The unitary U is thus implemented by adaptive measurements on the graph state, which remove qubits from the graph, using the initial entanglement of the state to propagate quantum information forward in time. We can see now why graph states are a primitive resource for MBQC. Note that the graph state used here is special: the first qubit is not initialized with $|+\rangle$, but with $|\psi\rangle$. This need not be the case, as we can always find a unitary which takes $|+\rangle$ to $|\psi\rangle$ and implement it with the same procedure before the first qubit in the figure above.

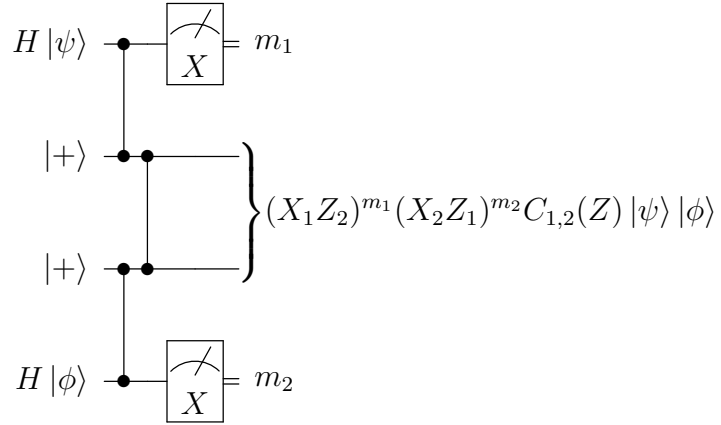
In section 2.2.3 we saw that single qubit gates together with CNOTs are universal for quantum computation. So to prove that MBQC is a universal model, we have to find a method to construct CNOTs, or $C(Z)$ gates, as they are equivalent up to Hadamards. Fortunately, this is an easy task, as $C(Z)$ gates are built into the graph states.

Consider the following circuit:

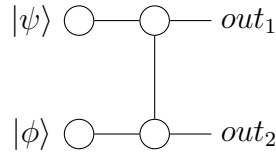


This circuit is build from two one-bit teleportations. At the end we connect a $C(Z)$ gate, leaving the output equal to $C_{1,2}(Z) |\psi\rangle |\phi\rangle$ up to local unitaries. But the $C(Z)$ gate can

be propagated in front of the measurements using eqs. (2.51) and (2.52):

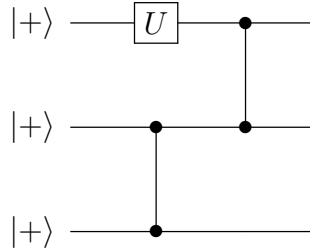


Now we see that we can prepare the gate before the measurement, between two qubits different from our desired target and control. This method is called gate teleportation and was proposed by Gottesmann and Chaung in 1999 [24]. The circuit in this form clearly shows that we can use graph states as a primitive resource for two qubit gates:

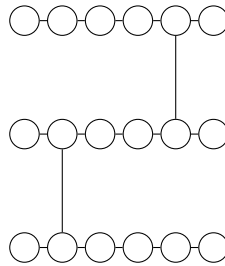


If the inputs do not line up, we can use the contractions described in the previous section. We have thus constructed the gates from a universal set ($C(Z)$ and single qubit gates), proving that MBQC is a universal model of quantum computation.

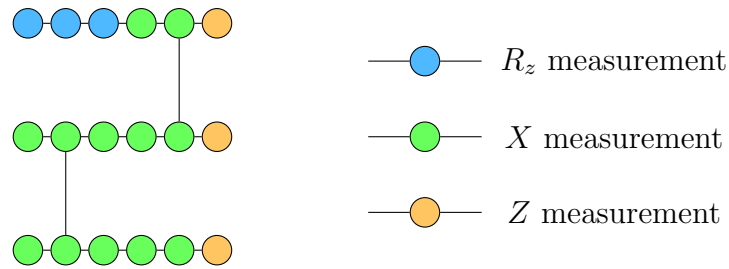
We provide an example. Let us say that we want to implement the circuit:



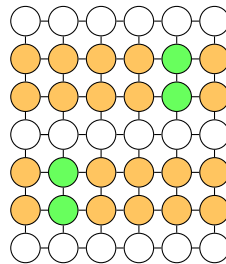
where U is an arbitrary unitary. The graph state used as a primitive resource has the form:



We indicate the measurements using the following color code:



Note that the measurements on each column should be performed simultaneously. Even if this graph state has the minimum number of qubits required, in practice, however, it is difficult to prepare a specific state for each circuit. We can although prepare a $2D$ cluster state and chip qubits away using X and Z measurements:



This shows that MBQC is a top-down perspective of quantum computing. We start with a base building block, a cluster state, and we slowly embed the structure of our circuit using projections. From an experimental point of view, the main challenge in MBQC is the construction of the initial cluster state, task which has different solutions dependent on the platform used.

Photonic Quantum Information Processing

Photons are great carriers of quantum information, as they have many degrees of freedom which can be exploited for processing and communication (path, polarization, orbital angular momentum, time bin, etc.). From an experimental point of view, their long decoherence time and weak interaction with the environment are clear advantages when trying to implement quantum computing. Moreover, integrated photonic microchips provide a scalable and programable platform which does not need expensive cooling systems, as it is the case for superconducting or ion-trap qubits [25].

This chapter is structured as follows. We will firstly present the mathematical formalism that describes the quantum electromagnetic field. Then we will introduce a qubit encoding on photons using the path degree of freedom and analyze how linear optics can be used to modify the qubit state. Finally, we introduce BosonSampling, a simple protocol which could provide a low-resource method of proving quantum advantage.

3.1 A quantum description of light

3.1.1 The classical electromagnetic field

The classical theory of light is based upon the work of James C. Maxwell on electromagnetism. In the presence of free electric charges and currents, the local form of Maxwell's equations is given by¹:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \quad (3.1)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (3.2)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (3.3)$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \quad (3.4)$$

From eqs. (3.2) and (3.4) it is straightforward to show that there exist $\phi(\mathbf{r}, t)$ and $\mathbf{A}(\mathbf{r}, t)$ well-behaved functions such that:

$$\mathbf{B} = \nabla \times \mathbf{A}; \quad \mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla \cdot \phi \quad (3.5)$$

Here ϕ is known as the scalar potential, while \mathbf{A} is the vector potential. The electrodynamic potentials are not unique, being dependent upon the gauge used. In the following

¹The presentation in this section is due to Gerry and Knight [26].

we will be using the Coulomb gauge:

$$\nabla \cdot \mathbf{A} = 0 \quad (3.6)$$

$$\frac{1}{c^2} \frac{\partial \mathbf{A}}{\partial t} - \nabla^2 \mathbf{A} = \mu_0 \mathbf{J} - \nabla \frac{1}{c^2} \frac{\partial \phi}{\partial t} \quad (3.7)$$

$$\nabla^2 \phi = \frac{\rho}{\varepsilon_0} \quad (3.8)$$

In addition to the Coulomb gauge, we also assume that the radiation travels in vacuum ($\mathbf{J} = 0$, $\rho = 0$). Thus:

$$\phi = 0; \quad \frac{1}{c^2} \frac{\partial \mathbf{A}}{\partial t} - \nabla^2 \mathbf{A} = 0 \quad (3.9)$$

The vector potential obeys the wave equation and can be expanded as a series of plane waves:

$$\mathbf{A}(\mathbf{r}, t) = \sum_{ks} \mathbf{e}_{ks} [A_{ks} e^{i(\mathbf{k}\mathbf{r} - \omega_k t)} + A_{ks}^* e^{-i(\mathbf{k}\mathbf{r} - \omega_k t)}] \quad (3.10)$$

where \mathbf{e}_{ks} are polarization directions:

$$\mathbf{e}_{ks} \cdot \mathbf{e}_{ks'} = \delta_{s,s'} \quad (3.11)$$

$$\mathbf{e}_{k1} \times \mathbf{e}_{k2} = \frac{\mathbf{k}}{k} \quad (3.12)$$

for a right-handed coordinate system. Here \mathbf{k} represents the wave vector. The dispersion relation for an electromagnetic wave in vacuum is given by $\omega_k = kc$. The sum over k represents the sum over all possible modes of the radiation field. To find the modes of the field we model the free space as a cubic region of side L with periodic boundary conditions, which along the Ox axis can be written as:

$$e^{ik_x x} = e^{ik_x (x+L)} \quad (3.13)$$

In a similar fashion, the relation can be written for the other two axes. In total:

$$k_x = \frac{2\pi n_x}{L} \quad (3.14)$$

$$k_y = \frac{2\pi n_y}{L} \quad (3.15)$$

$$k_z = \frac{2\pi n_z}{L} \quad (3.16)$$

For a continuum number of modes, the total number of modes in an infinitesimal region in the k -space is:

$$dn = 2dn_x dn_y dn_z = 2 \frac{V}{(2\pi)^3} d\mathbf{k} = \frac{V}{\pi^2} k dk \quad (3.17)$$

where the factor of two comes from the two independent polarization directions. A sum over k can thus be replaced by the integral:

$$\sum_k = \frac{V}{\pi^2} \int k dk \quad (3.18)$$

It is to be noted here that, due to the periodicity of the complex exponential, the following integrals have the value:

$$\int_V e^{\pm i(\mathbf{k}-\mathbf{k}')\mathbf{r}} dV = \delta_{\mathbf{k},\mathbf{k}'}; \quad \int_V e^{\pm i(\mathbf{k}+\mathbf{k}')\mathbf{r}} dV = \delta_{\mathbf{k},-\mathbf{k}'} \quad (3.19)$$

We are now ready to construct the field Hamiltonian. In the absence of electric charges and currents, the Hamiltonian of the electromagnetic field in a region of volume V in vacuum is given by:

$$H = \frac{1}{2} \int_V \left(\varepsilon_0 E^2 + \frac{1}{\mu_0} B^2 \right) dV \quad (3.20)$$

In the following, W_E denotes the energy of the electric field and W_B denotes the energy of the magnetic field. From eqs. (3.5) and (3.10), the expressions of the electric and magnetic field are:

$$\mathbf{E} = i \sum_{ks} \omega_k \mathbf{e}_{ks} [A_{ks} e^{i(\mathbf{k}\mathbf{r}-\omega_k t)} - A_{ks}^* e^{-i(\mathbf{k}\mathbf{r}-\omega_k t)}] \quad (3.21)$$

$$\mathbf{B} = i \sum_{ks} \mathbf{k} \times \mathbf{e}_{ks} [A_{ks} e^{i(\mathbf{k}\mathbf{r}-\omega_k t)} - A_{ks}^* e^{-i(\mathbf{k}\mathbf{r}-\omega_k t)}] \quad (3.22)$$

and using eq. (3.19) we arrive at the expressions for the energies of the electric and magnetic fields:

$$W_E = \varepsilon_0 V \sum_{ks} \omega_k^2 A_{ks} A_{ks}^* - R \quad (3.23)$$

$$W_B = \varepsilon_0 V \sum_{ks} \omega_k^2 A_{ks} A_{ks}^* + R \quad (3.24)$$

where:

$$R = \frac{\varepsilon_0 V}{2} \sum_{kss'} \omega_k^2 \mathbf{e}_{ks} \cdot \mathbf{e}_{-ks'} [A_{ks} A_{-ks'} e^{-2i\omega_k t} + A_{ks}^* A_{-ks'}^* e^{2i\omega_k t}] \quad (3.25)$$

The expression of the total energy is thus:

$$H = 2\varepsilon_0 V \sum_{ks} \omega_k^2 A_{ks} A_{ks}^* \quad (3.26)$$

3.1.2 The quantization of the electromagnetic field

Using the substitution:

$$A_{ks} = \frac{1}{2\sqrt{2\varepsilon_0 V} \omega_k} (\omega_k q_{ks} + i p_{ks}) \quad (3.27)$$

it can be seen that eq. (3.26) is equivalent with the Hamiltonian of a system of independent oscillators of unit mass, with q_{ks} as a generalized position and p_{ks} as a generalized momentum:

$$H = \frac{1}{2} \sum_{ks} p_{ks}^2 + \omega_k^2 q_{ks}^2 \quad (3.28)$$

We introduce now the quantization procedure. In quantum mechanics observable quantities are represented by Hermitian operators that act on a Hilbert space. We replace thus

the position and momentum in eq. (3.28) with the corresponding operators that satisfy the canonical commutation relations:

$$[\hat{q}_{ks}, \hat{q}_{k's'}] = [\hat{p}_{ks}, \hat{p}_{k's'}] = 0 \quad (3.29)$$

$$[\hat{q}_{ks}, \hat{p}_{k's'}] = i\hbar \delta_{kk'} \delta_{ss'} \quad (3.30)$$

Using a procedure similar to the case of the one-dimensional harmonic oscillator, the Hamiltonian of the system can be factorised by introducing the creation and annihilation operators:

$$\hat{a}_{ks}^\dagger = \frac{1}{\sqrt{2\hbar\omega_k}}(\omega_k \hat{q}_{ks} - i\hat{p}_{ks}) \quad (3.31)$$

$$\hat{a}_{ks} = \frac{1}{\sqrt{2\hbar\omega_k}}(\omega_k \hat{q}_{ks} + i\hat{p}_{ks}) \quad (3.32)$$

which satisfy the commutation relations:

$$[\hat{a}_{ks}, \hat{a}_{k's'}^\dagger] = \delta_{kk'} \delta_{ss'} \quad (3.33)$$

$$[\hat{a}_{ks}, \hat{a}_{k's'}] = 0 \quad (3.34)$$

The Hamiltonian becomes:

$$\hat{H} = \sum_{ks} \hbar\omega_k \left(\hat{n}_{ks} + \frac{1}{2} \right) \quad (3.35)$$

where $\hat{n}_{ks} = \hat{a}_{ks}^\dagger \hat{a}_{ks}$ is the number operator for the mode ks . Each of these modes can be relabeled by $ks \rightarrow j$:

$$\hat{H} = \sum_j \hbar\omega_j \left(\hat{n}_j + \frac{1}{2} \right) \quad (3.36)$$

A multimode photon state is just a product of the number states of all modes:

$$|\{n\}\rangle = \prod_j |n_j\rangle \quad (3.37)$$

where $n = (n_1, n_2, n_3, \dots)$ is an array of the numbers of photons in each mode. Given that $|n_j\rangle$ is an eigenstate of \hat{n}_j , then $|\{n\}\rangle$ is an eigenstate of \hat{H} :

$$\hat{H} |\{n\}\rangle = E |\{n\}\rangle \quad (3.38)$$

where E is given by eq. (3.35):

$$E = \sum_j \hbar\omega_j \left(n_j + \frac{1}{2} \right) \quad (3.39)$$

The multimode number states are of course orthogonal:

$$\langle n_1, n_2, \dots | n'_1, n'_2, \dots \rangle = \delta_{n_1 n'_1} \delta_{n_2 n'_2} \dots \quad (3.40)$$

The action of the j -th annihilation operator is to reduce the number of photons in mode j by one:

$$\hat{a}_j |\{n\}\rangle = \sqrt{n_j} |\dots, n_j - 1, \dots\rangle \quad (3.41)$$

The action of the j -th creation operator is to increase the number of photons in mode j by one:

$$\hat{a}_j^\dagger |\{n\}\rangle = \sqrt{n_j} |\dots, n_j + 1, \dots\rangle \quad (3.42)$$

The vacuum state will be denoted by:

$$|\odot\rangle = |0, 0, 0, \dots\rangle \quad (3.43)$$

with:

$$\hat{a}_j |\odot\rangle = 0 \quad (3.44)$$

By repeatedly applying the creation operator, any state can be obtained from the ground state:

$$|\{n\}\rangle = \prod_j \frac{(\hat{a}_j^\dagger)^{n_j}}{\sqrt{n_j!}} |\odot\rangle \quad (3.45)$$

Using eqs. (3.21), (3.22) and (3.27), we can write the operators for the electric and magnetic fields:

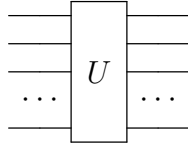
$$\hat{\mathbf{E}} = i \sum_{ks} \sqrt{\frac{\hbar\omega_k}{2\varepsilon_0 V}} \mathbf{e}_{ks} [\hat{a}_{ks} e^{i(\mathbf{k}\mathbf{r} - \omega_k t)} - \hat{a}_{ks}^\dagger e^{-i(\mathbf{k}\mathbf{r} - \omega_k t)}] \quad (3.46)$$

$$\hat{\mathbf{B}} = \frac{i}{c} \sum_{ks} \sqrt{\frac{\hbar\omega_k}{2\varepsilon_0 V}} \frac{\mathbf{k}}{k} \times \mathbf{e}_{ks} [A_{ks} e^{i(\mathbf{k}\mathbf{r} - \omega_k t)} - A_{ks}^* e^{-i(\mathbf{k}\mathbf{r} - \omega_k t)}] \quad (3.47)$$

3.2 Photons as qubits

3.2.1 Linear quantum optics devices

Consider the following quantum optical device, with N input modes and N output modes²:



The device is said to be linear if it transforms the creation operators linearly:

$$\hat{b}_i^\dagger = \sum_j u_{ij} \hat{a}_j^\dagger \quad (3.48)$$

where \hat{b}_i^\dagger , \hat{a}_j^\dagger are the output and input creation operators and u_{ij} are the elements of a unitary matrix. This general device is called an interferometer. An alternate description of an interferometer is given by the Hamiltonian which describes the evolution in eq. (3.48):

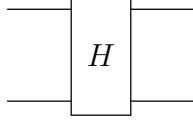
$$\hat{\mathbf{b}}^\dagger = e^{i\hat{H}} \hat{\mathbf{a}}^\dagger e^{-i\hat{H}} \quad (3.49)$$

where $\hat{\mathbf{a}}^\dagger$ is a column matrix of the input creation operators and $\hat{\mathbf{b}}^\dagger$ is a column matrix of the output creation operators. Time is included in the definition of \hat{H} . The matrix

²In the diagrams in this chapter, continuous lines are optical modes, not qubits.

elements u_{ij} can be found from eq. (3.49) using the Baker-Hausdorff lemma.

An ideal interferometer preserves the number of photons. To see this, consider the particular case of a 2-mode optical device:



This device is described in general by the Hamiltonian [27]:

$$\hat{H} = \underbrace{\hbar\omega_1 \left(\hat{a}_1^\dagger \hat{a}_1 + \frac{1}{2} \right) + \hbar\omega_2 \left(\hat{a}_2^\dagger \hat{a}_2 + \frac{1}{2} \right)}_{\hat{H}_0} + \underbrace{g(\hat{a}_1^\dagger \hat{a}_2 + \hat{a}_2^\dagger \hat{a}_1)}_{\hat{H}_i} \quad (3.50)$$

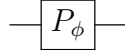
where \hat{H}_0 is the free energy and \hat{H}_i is the interaction energy between the two spatial modes. Using the usual commutation relations, we arrive at:

$$[\hat{H}_0, \hat{H}_i] = \hbar g(\omega_1 - \omega_2)(\hat{a}_1^\dagger \hat{a}_2 - \hat{a}_2^\dagger \hat{a}_1)$$

We can see that if $\omega_1 = \omega_2$, $[\hat{H}_0, \hat{H}_i] = 0$. This means that the interaction term H_i does not affect the eigenstates of H_0 , i.e. $|n_1, n_2\rangle$. This result is easily generalized to the n -port interferometer. Moreover, this also shows that the interferometer is a passive device, it does not require an energy intake to function.

Let's examine now two devices important in quantum optics and photonic quantum information processing: the phase shifter and the beamsplitter. The phase shifter is a passive device which introduces a phase shift in the electromagnetic field:

$$\hat{b}^\dagger = e^{i\phi} \hat{a}^\dagger \quad (3.51)$$

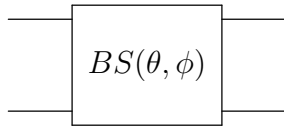


The Hamiltonian that describes the phase shifter is:

$$\hat{H}_{PS} = \phi \hat{a}^\dagger \hat{a} \quad (3.52)$$

The beamsplitter is a passive device which in the classic regime splits the energy of the incoming beam. In the quantum regime, the beamsplitter performs the following transformation on two modes [28]:

$$U_{BS}(\theta, \phi) = \begin{bmatrix} \cos \theta & -e^{-i\phi} \sin \theta \\ e^{i\phi} \sin \theta & \cos \theta \end{bmatrix} \quad (3.53)$$



where $\cos^2 \theta = R$ is the reflectivity of the beamsplitter and ϕ is the reflection phase shift. The Hamiltonian which describes the beamsplitter is:

$$\hat{H}_{BS} = \theta e^{i\phi} \hat{a}_1^\dagger \hat{a}_2 + \theta e^{-i\phi} \hat{a}_1 \hat{a}_2^\dagger \quad (3.54)$$

The case $\theta = \pi/4$, $\phi = 0$ corresponds to a 50:50 beamsplitter:

$$U_{BS} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = XH \quad (3.55)$$

which corresponds to a Hadamard operation up to a swap of the two modes. From now on, if θ and ϕ are not specified we will assume that the beamsplitter is 50:50.

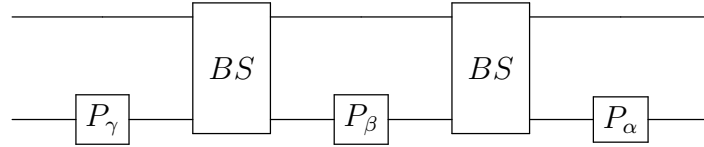
A general 2-mode device can be decomposed in phase shifters and 50:50 beamsplitters. The matrix U describing the device is a 2×2 unitary, so it can be decomposed along the X-Z axes:

$$U \cong R_z(\alpha) H R_z(\beta) H R_z(\gamma)$$

with α, β, γ real numbers. But observe that $R_z(\phi)$ is nothing but a phase shift on one of the modes:

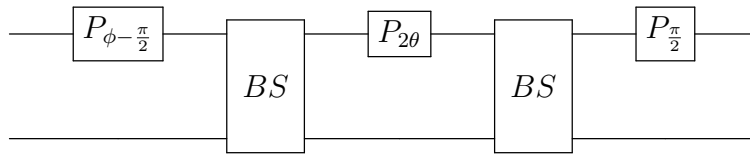
$$R_z(\phi) = \begin{bmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{bmatrix} \cong \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \quad (3.56)$$

and a Hadamard is just a 50:50 beamsplitter. So a general 2-mode device can be deconstructed as:



which is a Mach-Zehnder interferometer (MZI) with added phase shifts. Moreover, as it was first shown by Reck *et al.* in 1994 [12], any device described by a general $N \times N$ unitary U as in eq. (3.48) can be decomposed in a mesh of MZIs. We present here the scheme proposed by Clements *et al.* in 2016 [29], as it has lower optical depth and higher fidelity.

Both schemes use as a building block the following MZI:



If this MZI acts on the modes m and n ($m = n - 1$), then it is described by the unitary:

$$T_{m,n}(\theta, \phi) = \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & & & & & \vdots \\ \vdots & & \ddots & & & & \vdots \\ \vdots & & & e^{i\phi} \cos \theta & -\sin \theta & & \vdots \\ \vdots & & & e^{i\phi} \sin \theta & \cos \theta & & \vdots \\ \vdots & & & & & \ddots & \vdots \\ \vdots & & & & & & 1 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 1 \end{bmatrix} \quad (3.57)$$

where the non-trivial terms are on the lines m, n and columns m, n . This transformation is identical to a variable reflectivity beamsplitter with a phase shift on the first input port. The specific dependence of $T_{m,n}$ on θ and ϕ will not be shown in the following for clarity.

We use two important properties of $T_{m,n}$ in the decomposition scheme. One is that for any unitary U there are θ, ϕ such that $T_{m,n}U$ nulls an element on the line m or n . The other property is that $UT_{m,n}^{-1}$ nulls an element on the column m or n for appropriate θ and ϕ . At the end of these operations, the final matrix will still be called U .

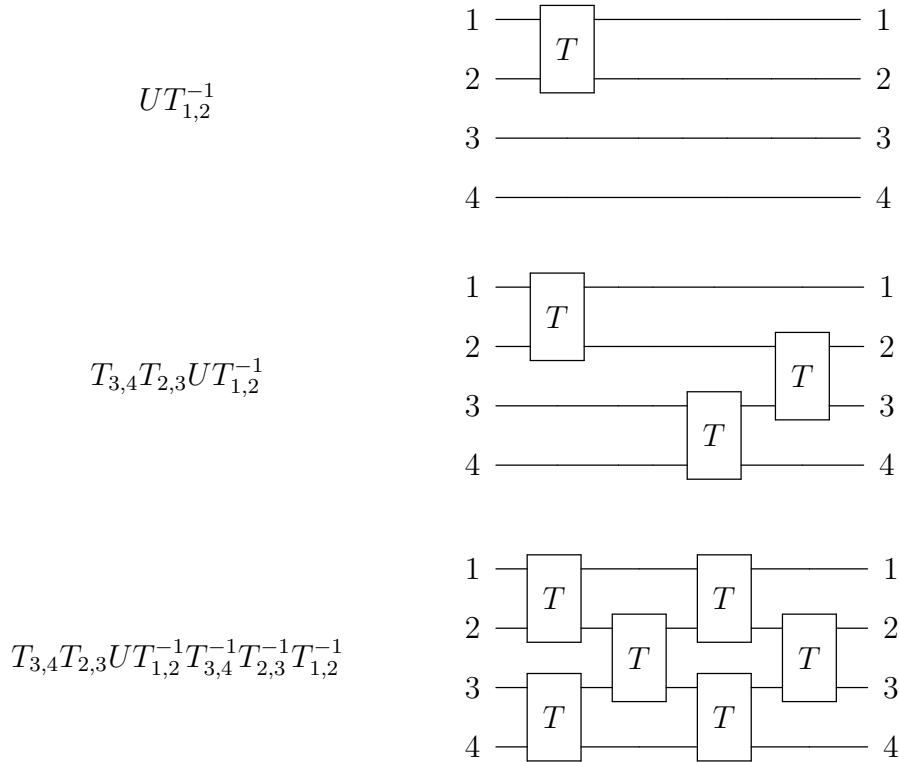
The scheme consists in nulling the lower half of U by alternating between T and T^{-1} operations, such that subsequent nulling operations do not interfere with previous ones. This is achieved with the nulling pattern:

$$\begin{bmatrix} \text{red } 6 \\ \text{blue } 3 & \text{red } 5 \\ \text{red } 1 & \text{blue } 2 & \text{red } 4 \end{bmatrix}$$

where the integers represent the order of nulling operations, red is T^{-1} and blue is T . At the end, U will be a diagonal phase matrix, due to it being unitary, which can be easily implemented using phase shifters. We will call this matrix D_N :

$$D_N = \text{diag}(e^{i\phi_1}, e^{i\phi_2}, \dots, e^{i\phi_N}) \quad (3.58)$$

As an example, consider a 4×4 unitary. The steps required to decompose this matrix are:



The final form can be rewritten as:

$$U = T_{2,3}^{-1} T_{3,4}^{-1} D_4 T_{1,2} T_{2,3} T_{3,4} T_{1,2}$$

It can be shown through direct multiplication that for any D_N and $T_{m,n}$ there is a D'_N such that $T_{m,n}^{-1} D_N = D'_N T_{m,n}$. Using this property, we can propagate the phase shift matrix at the end of the relation:

$$U = D'_4 T_{2,3} T_{3,4} T_{1,2} T_{2,3} T_{3,4} T_{1,2}$$

In this way any $N \times N$ unitary can be decomposed in $N(N+1)/2$ MZIs and N phase shifters. This shows not only that the decomposition is efficient in terms of physical resources, but also in terms of computational resources, as the time needed to factorize U is $O(N^2)$.

3.2.2 Dual-rail encoding

A qubit can be constructed using a single photon in two spatial modes, as the system has two degrees of freedom:

$$|0\rangle_L = \hat{a}_1^\dagger |\odot\rangle = |1, 0\rangle; |1\rangle_L = \hat{a}_2^\dagger |\odot\rangle = |0, 1\rangle \quad (3.59)$$

Here L denotes the logical states. This encoding is usually called dual-rail. From the previous section, it is easy to see that single qubit gates can be implemented on dual-rail qubits using beamsplitters and phase shifters.

Firstly, a 50:50 beamsplitter performs a Hadamard on the logical qubit:

$$|0\rangle_L = |1, 0\rangle \xrightarrow{BS} \frac{1}{\sqrt{2}}(|1, 0\rangle + |0, 1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle_L + |1\rangle_L) \quad (3.60)$$

$$|1\rangle_L = |0, 1\rangle \xrightarrow{BS} \frac{1}{\sqrt{2}}(|1, 0\rangle - |0, 1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle_L - |1\rangle_L) \quad (3.61)$$

Secondly, a phase shift on the second spatial mode performs a rotation around the z -axis, up to a phase factor (see eq. (3.56))

$$|0\rangle_L = |1, 0\rangle \xrightarrow{P_\phi} |1, 0\rangle = |0\rangle_L \quad (3.62)$$

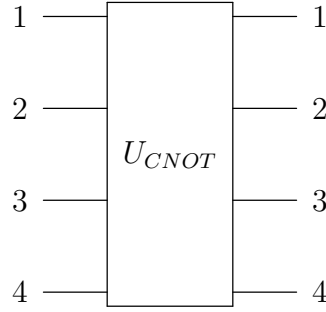
$$|1\rangle_L = |0, 1\rangle \xrightarrow{P_\phi} e^{i\phi} |0, 1\rangle = e^{i\phi} |1\rangle_L \quad (3.63)$$

And, as we have previously seen, any single qubit unitary can be written as a product of these operations:

$$U \cong R_z(\alpha) H R_z(\beta) H R_z(\gamma)$$

All we need now to achieve universality is a way to implement a CNOT. Here we encounter a problem with the linear optics architecture. Consider an interferometer which would implement a CNOT in the dual-rail encoding. This interferometer would have four inputs

and outputs:



and would transform the creation operators according to eq. (3.48):

$$\hat{b}_i^\dagger = \sum_{j=1}^4 (U_{CNOT})_{ij} \hat{a}_j^\dagger$$

But a CNOT performs the following transformation on the basis states:

$$\begin{aligned} |00\rangle_L &= |1, 0, 1, 0\rangle \xrightarrow{CNOT} |00\rangle_L = |1, 0, 1, 0\rangle \\ |01\rangle_L &= |1, 0, 0, 1\rangle \xrightarrow{CNOT} |01\rangle_L = |1, 0, 0, 1\rangle \\ |10\rangle_L &= |0, 1, 1, 0\rangle \xrightarrow{CNOT} |11\rangle_L = |0, 1, 0, 1\rangle \\ |11\rangle_L &= |0, 1, 0, 1\rangle \xrightarrow{CNOT} |10\rangle_L = |0, 1, 1, 0\rangle \end{aligned}$$

or, equivalently, on the creation operators:

$$\begin{aligned} \hat{b}_1^\dagger &= \hat{a}_1^\dagger; \quad \hat{b}_2^\dagger = \hat{a}_2^\dagger \\ \hat{b}_3^\dagger &= \hat{a}_3^\dagger; \quad \hat{b}_4^\dagger = \hat{a}_3^\dagger \\ \hat{b}_4^\dagger &= \hat{a}_4^\dagger; \quad \hat{b}_3^\dagger = \hat{a}_4^\dagger \end{aligned}$$

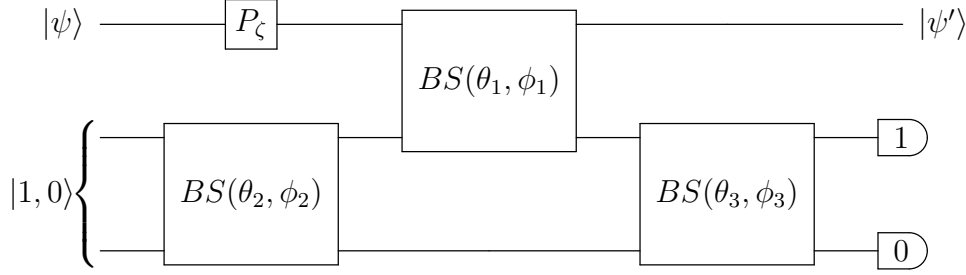
Clearly, there is no unitary U_{CNOT} such that all the above equalities are true at the same time. So a CNOT cannot be constructed in the dual-rail encoding using linear optics. This should be expected as interferometers do not provide the nonlinearities needed for photons to interact with each other. We can, however, induce effective nonlinearities using photon number measurements. This is the idea behind the protocol discussed in the following section.

3.2.3 The KLM protocol

In 2001, Knill, Laflamme and Millburn proved that linear optics and photon measurements are universal to quantum computation by constructing a non-deterministic $C(Z)$ gate on dual-rail qubits [30].

The building block of the KLM protocol is a conditional phase shift induced by photon

number measurements. Consider the following interferometer:



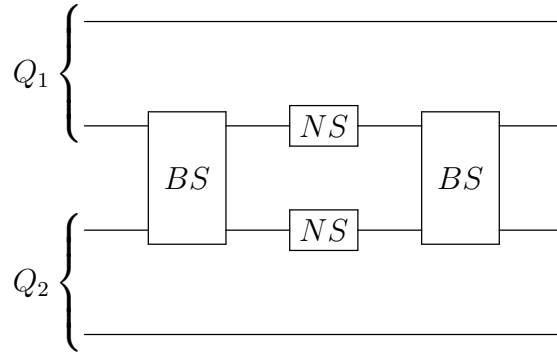
Notice that this is the form of a general unitary on three modes as per the decomposition scheme discussed earlier. We input only one photon in mode 2 and we measure the photon number in modes 2 and 3. If the output is the one shown in the figure, then the circuit performs a phase shift on state $|\psi\rangle$ in mode 1, which can be $|0\rangle$, $|1\rangle$ or $|2\rangle$. The value of this phase shift and the state it acts upon is dependent on the circuit parameters. The transformation which interests us is:

$$\alpha_1 |0\rangle + \alpha_2 |1\rangle + \alpha_3 |2\rangle \rightarrow \alpha_1 |0\rangle + \alpha_2 |1\rangle - \alpha_3 |2\rangle \quad (3.64)$$

It can be shown that for this transformation, the circuit parameters are: $\zeta = 180^\circ$, $\theta_1 = 65.5302^\circ$, $\phi_1 = 0^\circ$, $\theta_2 = 22.5^\circ$, $\phi_2 = 0^\circ$, $\theta_3 = -22.5^\circ$, $\phi_3 = 0^\circ$. The operation succeeds with probability 0.25, that is in one out of four cases we register the desired measurement output. Note that the setup requires photon number discriminating detectors, as we want to detect exactly one photon in mode 2 to preserve the number of photons in mode 1. This in turn increases the experimental cost of the setup, as current photon number discriminating detectors are bulky and require low temperatures. We will denote a successful operation by:

$$\text{---} \boxed{NS} \text{---}$$

We can easily build a $C(Z)$ operation using NS . The circuit which implements it is:

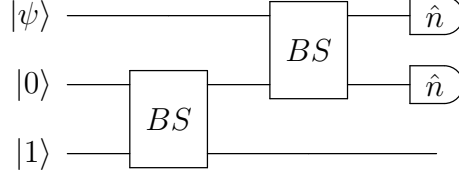


Here Q_1 and Q_2 mark the dual-rail qubits. For Q_2 we swap the usual convention in eq. (3.59), which can be implemented by physically swapping the modes.

Suppose we input two qubit basis states. When the logic state is $|11\rangle_L$, there is one photon in modes 2 and 3 each. The 50:50 beamsplitter transforms this state as $|1, 1\rangle \rightarrow |2, 0\rangle + |0, 2\rangle$, up to normalization factors. Afterwards the NS operations add an overall phase of π which is carried through the second beamsplitter. The output is thus $-|0, 1, 1, 0\rangle$ or $-|11\rangle_L$. The other basis states are unaffected because NS acts trivially

on $|1\rangle$ and $|0\rangle$. This is exactly the behaviour of a $C(Z)$ gate.

Both NS operations need to succeed, so the probability of applying $C(Z)$ is $1/16$. If we want to implement a circuit with n $C(Z)$ gates, we would need to run it 16^n times on average, an exponential scaling in the size of the circuit. To avoid this, KLM uses a modified version of the gate teleportation protocol presented in section 2.3.3 which takes into account the absence of a deterministic $C(Z)$ gate. Consider the following setup:

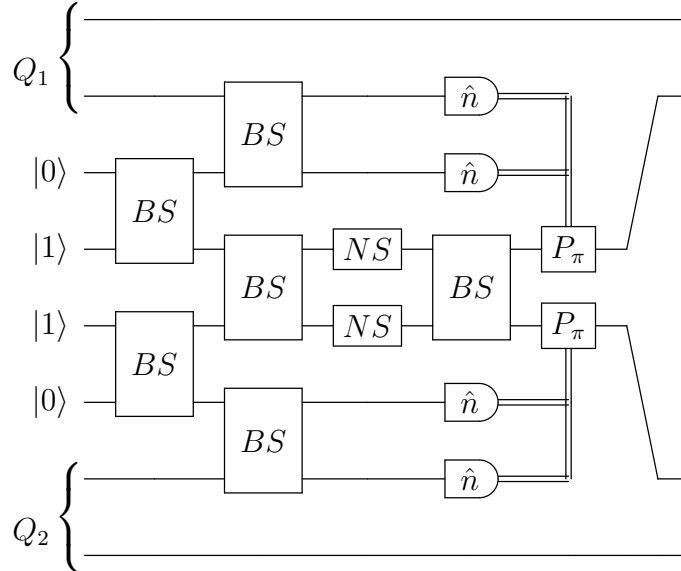


We are interested in states with at most one photon, so we analyze the cases $|\psi\rangle = |0\rangle$ and $|\psi\rangle = |1\rangle$:

$$|0, 0, 1\rangle \rightarrow \frac{1}{\sqrt{2}} |0, 0, 1\rangle - \frac{1}{2} |0, 1, 0\rangle + \frac{1}{2} |1, 0, 0\rangle \quad (3.65)$$

$$|1, 0, 1\rangle \rightarrow \frac{1}{2} (|1, 0, 1\rangle + |0, 1, 1\rangle + |0, 2, 0\rangle + |2, 0, 0\rangle) \quad (3.66)$$

So half of the time the first mode will be teleported to the third, with an added phase shift of π , depending on the measurement outcomes. We can now construct a gate teleportation protocol. Using the above mentioned setup, we can teleport the second mode of each qubit to a new mode and apply the required conditional phase shifts. We can then perform the $C(Z)$ gate between the new modes and observe that it commutes with the conditional phase shifts. The $C(Z)$ gate can thus be performed before the teleportations, so we only execute the teleportations when it succeeds (the gate is thus prepared offline). The circuit which implements this protocol is:



For $C(Z)$ to be applied between Q_1 and Q_2 , both teleportations must be successful. As each teleportation has a success rate of $1/2$, the success rate of the protocol is $1/4$, an improvement over the initial $1/16$.

Moreover, Knill, Laflamme and Milburn designed a teleportation protocol which uses n ancilla photons. The success rate of this protocol is $n/(n+1)$, boosting the success rate of the $C(Z)$ gate to $n^2/(n+1)^2$. This approaches unity as the numbers of photons used in the teleportation grows.

3.3 BosonSampling

As the previous section has shown, interferometers, single photon sources and measurements are not enough for a universal model of quantum computation. As the KLM protocol showed though, universality can be achieved with the addition of post-selection, feed-forward and photon number measurements. Research into LOQC has thus mainly focused on improvements on the KLM model, with significant advances made in recent years [31].

Considering this, it came as a surprise when Aaronson and Archipov showed that simulating photons propagating through a random interferometer is a hard task for classical computers. More specifically, the task of sampling from the output of an interferometer described by a uniformly random unitary U scales out of the reach of classical machines, but is perfectly tractable for the quantum system itself (or for other systems which emulate it [32]). This task is aptly named BosonSampling [33].

BosonSampling is of course a restricted model of quantum computing, as it can only perform a specific task. Nonetheless, as we can see from recent experimental results, it is a straightforward method of achieving quantum advantage [2]. Besides this, problems from many fields, ranging from quantum chemistry [34, 35] to graph theory [36, 37], map to variants of the BosonSampling problem.

3.3.1 Aaronson & Archipov BosonSampling

Consider an interferometer described by an $N \times N$ unitary matrix U . This interferometer transforms the creation operators according to eq. (3.48):

$$\hat{b}_i^\dagger = \sum_j u_{ij} \hat{a}_j^\dagger$$

with u_{ij} the entries of U . We ask the question: in the case of an arbitrary input state $|n_1, n_2, \dots, n_N\rangle$, what is the probability of measuring the output state $|m_1, m_2, \dots, m_N\rangle$? Before answering, we need to present a matrix function important to the discussion, the permanent.

The permanent is a matrix function $\text{per} : \mathcal{M}_N(\mathbb{C}) \rightarrow \mathbb{C}$ defined by:

$$\text{per}(U) = \sum_{\sigma \in S_N} \prod_{i=1}^N u_{i, \sigma(i)} \quad (3.67)$$

The sum here extends over the elements σ of the symmetric group S_N . Notice that this

definition closely resembles that of the determinant:

$$\det(U) = \sum_{\sigma \in S_N} \left(\text{sgn}(\sigma) \prod_{i=1}^N u_{i, \sigma_i} \right)$$

but it lacks the sign of the permutation. The permanent is not thus a multiplicative map ($\text{per}(AB) \neq \text{per}(A)\text{per}(B)$ in general), a property which in the case of the determinant can be exploited to ease computation (in the Gaussian elimination algorithm, for example). The problem of computing permanents is thought to be in $\#P$ -hard, extended to $\#P$ -complete for matrices with $(0, 1)$ entries [38].

The relation between interferometers and matrix permanents was shown by Scheel [39], so we are going to use the notation proposed by him. Let $U[k_1, \dots, k_m | l_1, \dots, l_m]$ be the $m \times m$ matrix whose elements are those from the original matrix U , with row indices k_1, \dots, k_m and column indices l_1, \dots, l_m . For example:

$$U[k_1, k_2, k_3 | l_1, l_2, l_3] = \begin{bmatrix} u_{k_1 l_1} & u_{k_1 l_2} & u_{k_1 l_3} \\ u_{k_2 l_1} & u_{k_2 l_2} & u_{k_2 l_3} \\ u_{k_3 l_1} & u_{k_3 l_2} & u_{k_3 l_3} \end{bmatrix}$$

The object $U[(1^{m_1}, 2^{m_2}, \dots) | (1^{n_1}, 2^{n_2}, \dots)]$ denotes a matrix whose elements are taken from U and whose row index i occurs exactly m_i times and column index j exactly n_j times. As an example:

$$U[(1^1, 2^1, 3^1) | (1^0, 2^2, 3^1)] = \begin{bmatrix} u_{12} & u_{12} & u_{13} \\ u_{22} & u_{22} & u_{23} \\ u_{32} & u_{32} & u_{33} \end{bmatrix}$$

We now return to the problem of computing the probability that the output of an interferometer is $|m_1, m_2, \dots, m_N\rangle$. Let \hat{U} denote the evolution operator from eq. (3.49) associated to U . Then:

$$\hat{U} |n_1, n_2, \dots, n_N\rangle = \prod_{i=1}^N \frac{1}{\sqrt{n_i!}} \left(\sum_{k_i=1}^N u_{k_i, i} \hat{a}_{k_i}^\dagger \right)^{n_i} |0\rangle^{\otimes N}$$

such that the required probability is $|\langle m_1, m_2, \dots, m_N | \hat{U} |n_1, n_2, \dots, n_N\rangle|^2$. Using combinatorial arguments, Scheel showed that:

$$\langle m_1, m_2, \dots, m_N | \hat{U} |n_1, n_2, \dots, n_N\rangle = \left(\prod_{i,j=1}^N n_i! m_j! \right)^{-1/2} \text{per}(U[\Omega' | \Omega]) \quad (3.68)$$

where $\Omega = (1^{n_1}, 2^{n_2}, \dots, N^{n_N})$ and $\Omega' = (1^{m_1}, 2^{m_2}, \dots, N^{m_N})$. This relation clearly shows the relation between interferometers and matrix permanents.

Let us consider a simple example: a 50:50 beamsplitter. The matrix U in this case is equal to the Hadamard matrix:

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

For simplicity we consider only single photon inputs. If the input is $|1, 0\rangle$, then we have two possibilities for the output: $|1, 0\rangle$ and $|0, 1\rangle$ as the interferometer preserves the photon number. For each of those the probability is $1/2$ as $U[1^1, 2^0 | 1^1, 2^0] = U[1^1, 2^0 | 1^0, 2^1] = 1/\sqrt{2}$.

If the input is $|1, 1\rangle$ we have three possible outputs: $|2, 0\rangle$, $|0, 2\rangle$, $|1, 1\rangle$. For the first two cases, the probability is $1/2$ as $\text{per}(U[1^1, 2^1 | 1^2, 2^0]) = -\text{per}(U[1^1, 2^1 | 1^0, 2^2]) = 1$. But the case $|1, 1\rangle$ can never occur as $\text{per}(U) = 0$. This is an explanation for the Hong-Ou-Mandel effect [40].

Aaronson and Archipov showed that sampling from the distribution in eq. (3.68) under specific conditions is computationally hard and constructed a protocol named BosonSampling [33]. In BosonSampling, U is randomly chosen from a uniform distribution over the space of unitary matrices according to the Haar measure (see appendix A). This is to ensure that the interferometer has no symmetries which could be exploited in classical algorithms. Conditions must also be imposed to the input state. If we look at eq. (3.68) and at the definition of the permanent, we observe that the distribution becomes easier to compute when many photons start in the same mode. Thus the photons should be spread as much as possible throughout the modes.

Moreover, even approximate sampling is found to be intractable to classical computers under these conditions. By approximate sampling we mean sampling from a distribution which is ε close to the initial distribution in terms of some distance. In this case, the distance is the total variation distance:

$$d(P, Q) = \sup_{A \in E} |P(A) - Q(A)| \quad (3.69)$$

where P and Q are probability distributions and E is the set of events.

The original article in reference [33] proves the hardness of BosonSampling in terms of computational complexity theory results which are above the purpose of this thesis. Instead, we present some arguments from a physical point of view for the difficulty of BosonSampling [41]:

1. The Hilbert space for the system considered has a faster-than-exponential growth. Indeed, consider the task of finding the number of ways in which n indistinguishable photons can be spread among m spatial modes. As we can put any number of photons in a spatial mode, the answer is m multichoose n :

$$\binom{m}{n} = \binom{m+n-1}{n} = \frac{(m+n-1)!}{n!(n-1)!} \quad (3.70)$$

which is seen to rapidly increase with the number of modes and with the number of photons.

2. The probability distribution of BosonSampling is related to the permanent. As stated earlier, the task of computing the permanent is in $\#P$, a complexity class even higher than NP. While there are algorithms that can approximate permanents in polynomial time in special situations, the best known general algorithm executes in $O(2^{n-1}n)$ time [42], where n is the size of the square matrix.

3. Identical photons tend to bunch together. By bunching we mean the presence of more than one photon in an output mode. This phenomenon is called the "boson birthday paradox", as it resembles the well-known birthday paradox. The paradox can be summed up in the following theorem [33, 43]:

Theorem 3.3.1 (Boson birthday paradox). *Let n photons be distributed at the input of an m mode interferometer sampled from the Haar measure, such that no more than one photon is in each mode. Then we need at least $n \sim \sqrt{m \ln 2}$ photons for bunching events to have majority probability.*

Theorem 3.3.1 has implications in experimental BosonSampling. If we use $|1\rangle^{\otimes n} |0\rangle^{\otimes m-n}$ as input state with $m = O(n^2)$, then the output state will exclusively consist of single photon states with a high probability. An experimental setup in this case could use only on/off ("bucket") detectors instead of number-discriminating detectors, which are more expensive.

A very important question that needs to be considered is what happens to the hardness of BosonSampling in the presence of errors, as any system suffers from imperfections. Let P denote the probability that we sample from the desired probability distribution. The proof of Aaronson and Archipov considered the regime where $P > 1/\text{poly}(n)$. To the knowledge of the author, this limit has not yet been loosened.

A simple error model was proposed by Rhode [44]. Written in density matrix form, the desired input state for a BosonSampling protocol is:

$$\rho_{\text{in}} = (|1\rangle\langle 1|)^{\otimes n} \otimes (|0\rangle\langle 0|)^{\otimes m-n} \quad (3.71)$$

But consider now that for a mode i we can produce exactly one photon only with a probability of p , while with a probability of $1 - p$ we produce a random state $\rho_{\text{err}}^{(i)}$. In this case the input state would be:

$$\rho_{\text{in}}^{\text{err}} = \left(\bigotimes_{i=1}^n [p|1\rangle\langle 1| + (1-p)\rho_{\text{err}}^{(i)}] \right) \otimes [|0\rangle\langle 0|]^{\otimes m-n} \quad (3.72)$$

such that we would sample from the desired distribution with a probability of p^n . This shows that the protocol does not tolerate large error rates as the exponential rapidly outpaces the polynomial in $P > 1/\text{poly}(n)$.

We summarise now the steps in the BosonSampling protocol [45]:

1. Prepare an n photon m mode input state. There are two conditions that this state must obey. First the photons should be spread as much as possible among the modes, ideally at most one per mode; this is to ensure that the permanent in eq. (3.68) remains hard to compute. Secondly, $m = O(n^2)$ to ensure that bunching events are rare.
2. Evolve the state through an interferometer U sampled uniformly from the Haar measure. The uniformity of U ensures that there are no symmetries which could be exploited by classical algorithms and is a second requirement for the rarity of bunching events.

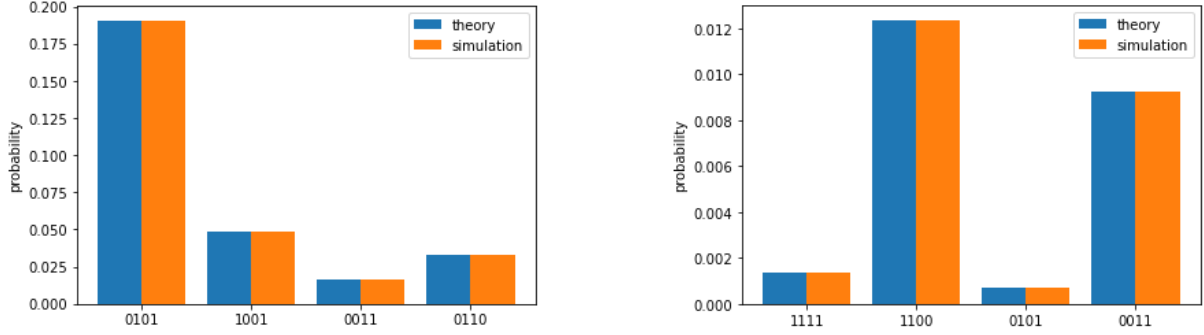


Figure 3.1: Simulation results of BosonSampling and GBS using the Strawberry Fields package by Xanadu [46]. The results have been compared to theory. On the left, BosonSampling simulation with $|1, 1, 0, 0\rangle$ input. On the right, simulation of GBS on four spatial modes. For details, see appendices B and C.

3. Measure the output state using photodetectors. As the output should consist mainly of single photon states, detectors that only distinguish between vacuum and non-vacuum should suffice.

3.3.2 BosonSampling with Gaussian states

In this section we will examine a variant of the BosonSampling protocol which uses a different type of input state. Single photon states are hard to achieve in a realistic setup in the form required by the BosonSampling protocol. While current single photon sources can reach emission probabilities of even 99% [47], the exponential nature of the Hilbert space presented in eq. (3.70) means that even small errors would have a huge impact on the sampling time.

In 2016, Hamilton *et al.* proposed a setup where squeezed vacuum states are used instead of Fock states and proved that the problem remains hard for classical machines. This protocol was named Gaussian BosonSampling (GBS) [48]. Before describing GBS though, we discuss single mode squeezing and squeezed vacuum states³.

Squeezing operations correspond to hamiltonians of the form $H \propto (a^\dagger)^2 + h.c.$. We can see that squeezing is a nonlinear transformation as the hamiltonian involves a higher order power of the creation operator. Squeezing thus does not conserve the photon number and is not a passive transformation (i.e. it requires an intake of energy). The single-mode squeezing operator is usually written as (we drop the operator hat for simplicity):

$$S(\xi) = \exp \left\{ \frac{1}{2} \xi (a^\dagger)^2 - \frac{1}{2} \xi^* a^2 \right\} \quad (3.73)$$

corresponding to an evolution given by:

$$S^\dagger(\xi) a S(\xi) = \mu a + \nu a^\dagger, \quad S^\dagger(\xi) a^\dagger S(\xi) = \mu a^\dagger + \nu^* a \quad (3.74)$$

where $\mu \in \mathbb{R}, \nu \in \mathbb{C}, \mu = \cosh r, \nu = e^{i\psi} \sinh r, \xi = r e^{i\psi}$. We can obtain squeezed vacuum states by applying the single-mode squeezing operator to the vacuum $|\xi\rangle = S(\xi) |0\rangle$. Their

³The information on squeezing is taken from [49].

expansion in the Fock basis contains only even components:

$$|\xi\rangle = \frac{1}{\sqrt{\mu}} \sum_{k=0}^{\infty} \left(\frac{\nu}{2\mu} \right)^k \frac{\sqrt{(2k)!}}{k!} |2k\rangle \quad (3.75)$$

Even if the name suggests otherwise, squeezed vacuum states have a non-zero mean photon number, as $\langle \xi | a^\dagger a | \xi \rangle = |\nu|^2$.

We can generalize eqs. (3.31) and (3.32) to define the quadrature operators:

$$x_\phi = \frac{1}{2} (ae^{-i\phi} + a^\dagger e^{i\phi}) \quad (3.76)$$

where we drop the ks label as we consider identical photons and scaled the operators to the constants. For $\phi = 0$ and $\phi = \pi/2$ we obtain position- and momentum-like operators.

For squeezed vacuum states $\langle \xi | x_\phi | \xi \rangle = 0$, $\forall \phi$, but the variance of the quadrature is given by:

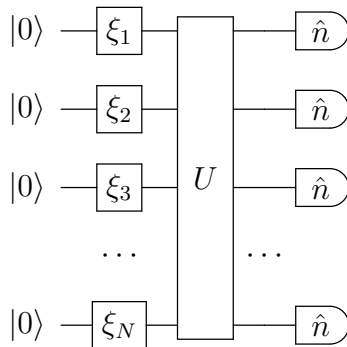
$$\Delta x_\phi^2 = \langle \xi | x_\phi^2 | \xi \rangle = \frac{1}{4} [e^{2r} \cos^2(\phi - \psi/2) + e^{-2r} \sin^2(\phi - \psi/2)] \quad (3.77)$$

such that squeezed vacuum states have minimum uncertainty for the pair of variables $x_{\psi/2}$ and $x_{\psi/2+\pi/2}$ (for $\psi = 0$ these are position and momentum), for which $\Delta x_{\psi/2}^2 = 1/4e^{2r}$ and $\Delta x_{\psi/2+\pi/2}^2 = 1/4e^{-2r}$. While for vacuum $\Delta x_\phi^2 = 1/4$, such that the uncertainty is equally spread around the origin, for squeezed vacuum states x and p uncertainties are different, while maintaining the same area in the $x - p$ plane.

What is important in the context of experimental GBS is that squeezing can be done deterministically. We will denote the squeezing operation $S(\xi)$ by:

$$\text{---} \boxed{\xi} \text{---}$$

A basic GBS protocol has the following setup:



where U is sampled from the Haar measure. Notice that we only input vacuum states. The squeezing operations can be absorbed in the interferometer matrix U , leaving no information about the input state. This means that the output probabilities will be dictated only by U and the output pattern, in comparison with BosonSampling where the input matters as well. A consequence of this is that the sampling space is reduced, enhancing data acquisition and analysis at the cost of computational complexity (the problem still

remains hard to simulate classically).

While BosonSampling revolved around the permanent, GBS results are based on a more general matrix function called the Hafnian. The Hafnian $\text{haf} : \mathcal{M}_N(\mathbb{C}) \rightarrow \mathbb{C}$ is defined by:

$$\text{Haf}(A) = \frac{1}{n!2^n} \sum_{\sigma \in S_{2N}} \prod_{i=1}^N A_{\sigma(2i-1)\sigma(2i)} \quad (3.78)$$

where S_{2N} is the symmetric group of order $2N$.

Both the permanent and the Hafnian have useful graph theory interpretations. Firstly, we give some definitions:

1. A perfect matching in a graph $G(E, V)$ is a subset $M \subset E$ such that every vertex in V touches exactly one edge in M .
2. A graph $G(E, V)$ is said to be bipartite if V is the union of two disjoint sets U and W and every edge in E connects a vertex from U to one in W .
3. The adjacency matrix of a simple graph $G(E, V)$ with $V = v_1, v_2, \dots, v_n$ is a square $n \times n$ symmetric matrix A with elements a_{ij} such that a_{ij} is one if there is an edge connecting v_i and v_j and zero otherwise.

If $G(E, V)$ is a bipartite graph with adjacency matrix A , then $\text{per}(A)$ is the number of perfect matching in G . The Hafnian generalizes this property to any simple graph, i.e. if G is a simple graph and A is its adjacency matrix, then $\text{Haf}(A)$ is the number of perfect matching in G . This generalization is shown directly by the relation:

$$\text{per}(A) = \text{Haf} \left(\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \right) \quad (3.79)$$

Equation (3.79) also tells us that the Hafnian is as hard to compute as the permanent.

As in the case of BosonSampling, we consider the problem of finding the probability of measuring a specific output from the GBS device. We can write eq. (3.74) using matrix notation:

$$\begin{bmatrix} b \\ b^\dagger \end{bmatrix} = \begin{bmatrix} \cosh r & \sinh r \\ \sinh r & \cosh r \end{bmatrix} \begin{bmatrix} a \\ a^\dagger \end{bmatrix} \quad (3.80)$$

where we choose $\psi = 0$. For two modes, we choose to arrange the modes in the following way:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_1^\dagger \\ b_2^\dagger \end{bmatrix} = \begin{bmatrix} \cosh r_1 & 0 & \sinh r_1 & 0 \\ 0 & \cosh r_2 & 0 & \sinh r_2 \\ \sinh r_1 & 0 & \cosh r_1 & 0 \\ 0 & \sinh r_2 & 0 & \cosh r_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_1^\dagger \\ a_2^\dagger \end{bmatrix} \quad (3.81)$$

such that for N modes the squeezing matrix S is written as:

$$S = \begin{bmatrix} \oplus_{j=1}^N \cosh r_j & \oplus_{j=1}^N \sinh r_j \\ \oplus_{j=1}^N \sinh r_j & \oplus_{j=1}^N \cosh r_j \end{bmatrix} \quad (3.82)$$

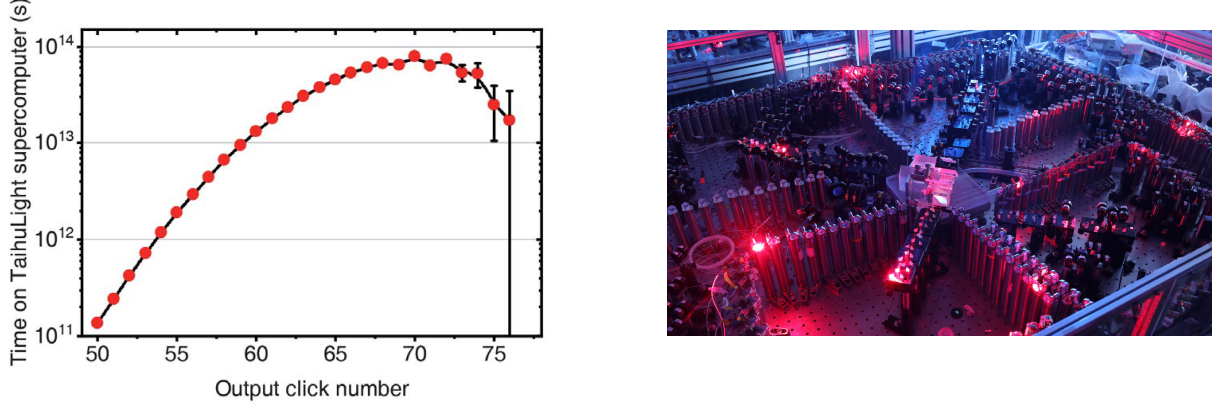


Figure 3.2: GBS experiment by Zhong *et al.* [2]. The figure on the right is a photograph of the photonic network used, with the interferometer in the middle of the table. While the interferometer has 50 spatial modes, the experiment uses both horizontally and vertically polarized light. This yields an effective number of spatial modes equal to 100, as orthogonal states of polarisation do not interfere. The plot on the left shows the sampling time on the TaihuLight supercomputer as a function of the total number of photons at the output. The sampling task that GBS performed would take this supercomputer 8×10^{16} s or 2.5 billion years to accomplish.

with $\oplus_{j=1}^N u_i = \text{diag}(u_1, u_2, \dots, u_N)$. Let $s = (s_1, s_2, \dots, s_n)$ be an output configuration with $s_i \in \{0, 1\}$. By analysing the evolution of the squeezed states through the interferometer in the phase-space, Hamilton *et al.* showed that the probability of measuring s is given by:

$$\Pr(s) = |\sigma|^{-1/2} |\text{Haf}(B[s|s])|^2 \quad (3.83)$$

Here $|A|$ denotes the absolute value of the determinant of A and:

$$\sigma = \frac{1}{2} \begin{bmatrix} U & 0 \\ 0 & U^* \end{bmatrix} S S^\dagger \begin{bmatrix} U^\dagger & 0 \\ 0 & U^t \end{bmatrix} + \frac{I}{2}$$

$$B = U \left(\oplus_{j=1}^M \tanh r_j \right) U^t$$

The presence of the Hafnian in eq. (3.83) allows GBS to inherit some the complexity proofs of BosonSampling due to eq. (3.79). Moreover, the Hafnian allows the mapping of many problems of interest to the GBS protocol as it is a more general function than the permanent.

We discuss now briefly recent applications and experimental implementations of GBS. One of the straightforward applications of GBS is a proof of quantum advantage as it solves a problem intractable for classical machines. An experiment by Zhong *et al.* (see fig. 3.2) in 2020 using 50 optical modes reported an output space of $\sim 10^{30}$ and a sampling rate $\sim 10^{14}$ faster than state-of-the-art simulation strategies and supercomputers [2].

Due to the graph theory interpretation of the Hafnian, many graph-based problems can be mapped onto GBS. Let $G = (E, V)$ be a simple graph and $S(E_S, V_S)$ a subgraph of G . Then the density of S is defined as:

$$d(S) = \frac{|E_S|}{|V_S|} \quad (3.84)$$

A simple counting argument shows that $d(S) \leq 1/2(|V_S| - 1)$, with the maximum for a fully connected subgraph (also called a clique). We note here two problems related to dense subgraphs: the k -densest subgraph problem and the maximum clique problem. The task in the k -densest subgraph problem is to find the densest subgraph S with $|V_S| = k$, while in the maximum clique problem we want to find a clique such that $|V_S|$ is the maximum possible. These problems model a variety of concepts, from social and financial networks to molecular docking in biological systems.

Subgraph problems are usually hard to solve for classical machines, the two mentioned above being in NP. By mapping onto GBS however, we can circumvent this issue. This is generally done by noticing that any symmetric matrix A , in our case the adjacency matrix of the graph, can be factorized as⁴:

$$A = U_A \left(\oplus_{j=1}^M \lambda_j \right) U_A^t \quad (3.85)$$

with U_A a unitary matrix and $0 \leq \lambda_i < 1$. Thus if we program the GBS device with an interferometer U_A and squeezing parameters $\tanh r_i = \lambda_i$, we will sample from the distribution:

$$\Pr(s) \propto |\text{Haf}(A[s|s])|^2 \quad (3.86)$$

which gives us access to subgraphs $A[s|s]$. Subgraph algorithms have been experimentally implemented on small scale GBS platforms [50].

⁴This is known as the Autonne-Takagi factorization.

Conclusions

In this thesis, we have presented the basics of quantum computing and possible implementations using photonic qubits. Photons have many properties that can be exploited in quantum computing. Their long decoherence time and weak coupling with the environment makes them ideal candidates for robust qubits. Added to this is the fact that optical elements like beamsplitters and phase shifters can be integrated on microchips which operate at room temperature, leading to better scalability than their superconducting or ion-trap counterparts. On the other hand, the weak interaction between photons poses a difficult experimental challenge in the implementation of controlled-operations. In section 3.2.3 we presented the KLM protocol which addressed this issue by inducing effective non-linear interactions using photon-number measurements. The author would like to expand on this topic in the future, in particular on connections to MBQC, the quantum computing model presented in section 2.3 [31].

We have also seen that the photonic platform permits for a cheap and reliable protocol that could prove quantum advantage, namely BosonSampling. Recent quantum advantage claims [2] prove that GBS is capable of tackling problems intractable to the best of the current supercomputers. These kind of results and debates upon their validity will surely become more frequent in the near future. While the problem of quantum advantage is of great interest to both the fields of quantum computing and computational complexity, we should not overlook the near-term applications of GBS. In particular, the author would like to expand on graph embeddings in GBS devices as the one presented in section 3.3.2. These graph-based protocols could have near-term applications in the simulation of a wide range of systems, from physical to biological.

The Haar measure

In this appendix we will present the Haar measure from a physical perspective, as a proper mathematical description in terms of measure theory is beyond the scope of this thesis. In section 3.3 we saw that a main part of the BosonSampling protocol is the uniform sampling of an $N \times N$ unitary U . To see what this actually means, we focus on elements of $U(2)$, as they can be visualised by plotting $U|0\rangle$ on the Bloch sphere.

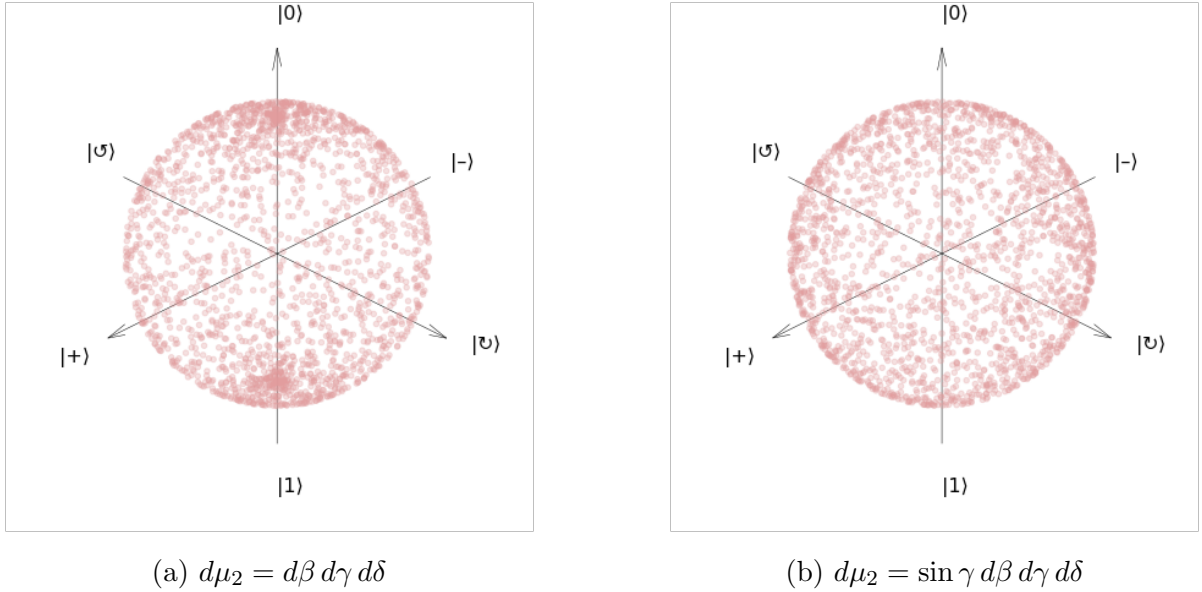


Figure A.1: An ensemble of 2021 unitary matrices visualised on the Bloch sphere, generated according to (a) a uniform distribution of the parameters (b) the Haar measure. Figures generated using PennyLane by Xanadu [51] and code snippets from [52].

As eq. (2.19) states, any single-qubit unitary can be parameterized by three real numbers:

$$U \cong R_z(\beta)R_x(\gamma)R_z(\delta)$$

If we want to sample U , we can thus sample β , γ and δ and construct U using the relation above. We now have to find the distribution of these parameters that produces a uniform distribution over $U(2)$. We may think at first that a uniform distribution of β , γ and δ would suffice:

$$d\mu_2 = d\beta d\gamma d\delta \tag{A.1}$$

but this produces a non-uniform distribution of the unitaries as it can be seen from fig. A.1a. As the matrices are concentrated around the poles, we should distribute them

with a higher probability around the equator. If we use $Pr(\gamma) = \sin \gamma$, such that:

$$d\mu_2 = \sin \gamma d\beta d\gamma d\delta \quad (\text{A.2})$$

then the unitaries become uniformly distributed around the Bloch sphere, as depicted in fig. A.1b. Equation (A.2) is known as the Haar measure for $U(2)$.

For $U(N)$, we can write the matrix U as a product of $U(2)$ operations using decomposition schemes like the ones presented in section 3.2.1 [29, 12]. In this way the Haar measure can be expressed in a recursive form [52]:

$$d\mu_N = d\mu'_{N-1} \times \sin \gamma_{N-1} \sin^{2(N-2)} \left(\frac{\gamma_{N-1}}{2} \right) d\gamma_{N-1} d\beta_{N-1} \times d\mu_{N-1} \quad (\text{A.3})$$

where γ_i is a parameter for the i -th $U(2)$ element in the decomposition of U .

While eq. (A.3) could provide us with the probability distribution for the parameters, the computational cost to generate each of them would be too high. In practice we use an algorithm detailed in [53], based on the QR decomposition. The steps of the algorithm are:

1. Generate a matrix U with entries $a + bi$, where a and b are sampled from a normal distribution with 0 mean and variance equal to 1.
2. Apply the QR decomposition, $U = QR$, Q -orthogonal matrix, R -upper triangular matrix.
3. compute $Q' = QA$, where $A = \text{diag}(R_{ii}/|R_{ii}|)$. This matrix is Haar-random.

The explanation for the algorithm is as follows. We firstly generate a matrix with random entries. As a unitary is constrained by $U^\dagger U = I$, we use the QR decomposition to extract Q which will be unitary as the initial matrix is complex valued. As shown in [53], a Haar-random matrix has uniformly-random eigenvalues, which is not the case for Q as the QR decomposition is not unique ($U = QR = QV^\dagger VR = Q'R'$, with V a unitary). We thus fix the eigenvalues using the matrix A .

This algorithm is used in the `unitary_group` function from the python package `scipy` [54].

Simulating BosonSampling

In this appendix¹ we present the simulation of BosonSampling from fig. 3.1. The simulation was performed using the Strawberry Fields python package from Xanadu [46].

Firstly, we import necessary libraries:

```
import strawberryfields as sf
from strawberryfields.ops import *
import thewalrus as w
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st
```

Besides Strawberry Fields, we also import Numpy, Scipy, Thewalrus and Matplotlib [56, 57, 58] for data generation, manipulation and visualisation. Using the function `unitary_group` from Scipy we can generate a Haar-random unitary (see appendix A):

```
modes = 4
u = st.unitary_group.rvs(modes)
```

In this example we restrict ourselves to four spatial modes. Next, we define a function which will compute the probability of a specific output from a specific input in the interferometer R , according to eq. (3.68):

```
def perm_prob(R, inp, out):
    #R - unitary matrix, inp - input string n, out - output string m
    omegap, omega = [], []
    for i in range(len(inp)):
        if inp[i] != '0':
            omega.append(int(i**float(inp[i])))
        if out[i] != '0':
            omegap.append(int(i**float(out[i])))
    return np.absolute(w.perm(R[:, omega][omegap]))**2
```

$|1, 1, 0, 0\rangle$ is our input. We choose four random output strings and compute the theoretical probabilities:

```
inp = ['1100']*4
out = ['0101', '1001', '0011', '0110']
expected = np.array([perm_prob(u, *z) for z in zip(inp, out)])
```

¹This appendix follows the guide in [55].

We can now construct the Strawberry Fields circuit (called a program). We define a program with four modes, initialize the modes with the desired Fock states, and attach to it an interferometer defined by u :

```
b_sampling = sf.Program(4)

with b_sampling.context as q:
    Fock(1) | q[0]
    Fock(1) | q[1]
    Vac    | q[2]
    Vac    | q[3]

    Interferometer(u) | q
```

Note that we do not add measurements. This is because we want the program to output the full state, not measurement outcomes. We run now the program on a Fock backend. As the Fock space is infinite, we need to define a cutoff dimension. In this case it is 7:

```
eng = sf.Engine(backend="fock", backend_options={'cutoff_dim':7})
result = eng.run(b_sampling)
probs = result.state.all_fock_probs()
wwg = [probs[0,1,0,1], probs[1,0,0,1], probs[0,0,1,1], probs[0,1,1,0]]
```

Here, wwg is an array containing the desired probabilities from the simulation. The following code snippet produces the bar plot in fig. 3.1:

```
x = np.array([0,1,2,3])
plt.bar(x-0.2, expected, 0.4, label="theory")
plt.bar(x+0.2, wwg, 0.4, label="simulation")
plt.ylabel("probability")
plt.xticks(x, ['0101', '1001', '0011', '0110'])
plt.legend()
```

The values seem to be equal, but they actually differ due to the dimension cutoff. We can see this if we compute the difference between the expected and simulation results:

```
expected-wwg
array([-1.38777878e-16, -4.16333634e-17,  6.93889390e-18,  0.00000000e+00])
```

Simulating GBS

In this appendix¹ we present the simulation of GBS from fig. 3.1. The simulation was performed using the Strawberry Fields python package from Xanadu [46].

We use the same imports and generate the unitary in the same way as in appendix C. In addition, we also generate random squeezing parameters $1 \leq r_i < 2$:

```
sq = np.random.random(size = modes)+1
```

We now build the program. We use four spatial modes, apply the squeezing operations and append the interferometer:

```
prog = sf.Program(modes)
with prog.context as q:
    for i in range(modes):
        Sgate(sq[i]) | q[i]
    Interferometer(U) | q
```

For GBS we run the simulation on a Gaussian backend, as we want to obtain the final Gaussian state:

```
eng = sf.Engine(backend='gaussian')
result = eng.run(prog)
```

To obtain the theoretical probabilities, we need to compute the matrices σ and B . This is done in the following snippet:

```
S = np.block([
    [np.diag(np.cosh(sq)), np.diag(np.sinh(sq))],
    [np.diag(np.sinh(sq)), np.diag(np.cosh(sq))]
])

Z = np.zeros((modes,modes))
sigma = 1/2*np.block([[U,Z],[Z,U.conjugate()]])@S@S.T.conjugate()@np.block([[U.T,
sigma = sigma+1/2*np.identity(2*modes)

B = U@np.diag(np.tanh(sq))@U.T
```

where @ denotes matrix multiplication. We define the following function to extract the probabilities:

¹This appendix follows the guide in [59].

```

def prob(out):
    #out - output configuration s as tuple/array
    #     containing the indexes of modes with
    #     different than zero photon number
    haf = w.hafnian(B[:,out][out])
    p = np.abs(haf)**2/np.sqrt(np.abs(sp.linalg.det(sigmav)))
    return p

```

We can now extract the theoretically predicted probabilities and the results from the simulation. We use randomly selected outputs:

```

expected = np.array([prob([0,1,2,3]),prob([0,1]),
                    prob([1,3]),prob([2,3])])
wwg = np.array([result.state.fock_prob([1,1,1,1]),
                result.state.fock_prob([1,1,0,0]),
                result.state.fock_prob([0,1,0,1]),
                result.state.fock_prob([0,0,1,1])])

```

To print fig. 3.1, we use the same code as in appendix B. We show here the difference between the expected values and the simulation:

```

expected-wwg
array([-8.8905e-18, -8.6736e-18, -8.6736e-18, -6.5919e-17])

```

As in the case of BosonSampling, the difference is due to internal dimension cutoffs.

References

- [1] Héctor Abraham *et al.* Qiskit: An open-source framework for quantum computing. 10.5281/zenodo.2562110, 2019.
- [2] Zhong *et al.* Quantum computational advantage using photons. 370(6523):1460–1463, Dec 2020.
- [3] Arute *et al.* Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, Oct 2019.
- [4] Feng Pan and Pan Zhang. Simulating the sycamore quantum supremacy circuits. <https://arxiv.org/abs/2103.03074v1>, 2021.
- [5] F. Bloch. Nuclear induction. *Phys. Rev.*, 70(7-8):460–474, Oct 1946.
- [6] George B. Arfken and Hans-Jurgen Weber. *Mathematical methods for physicists*. Elsevier, 6th edition.
- [7] Barenco *et al.* Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, Nov 1995.
- [8] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1971.
- [9] Horodecki *et al.* Quantum entanglement. *Rev. Mod. Phys.*, 81(2):865–942, Jun 2009.
- [10] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 10th edition.
- [11] David P. DiVincenzo. Two-bit gates are universal for quantum computation. *Phys. Rev. A*, 51(2):1015–1022, Feb 1995.
- [12] Reck *et al.* Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73:58–61, Jul 1994.
- [13] P. Oscar Boykin *et al.* On universal and fault-tolerant quantum computing. <https://arxiv.org/abs/quant-ph/9906054>, 1999.
- [14] A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, dec 1997.
- [15] D. Gottesman. Stabilizer codes and quantum error correction. <https://arxiv.org/abs/quant-ph/9705052>, May 1997.

- [16] Keisuke Fujii. *Quantum computation with topological codes: from qubit to topological fault-tolerance*. Springer, 1st edition.
- [17] Fowler *et al.* Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86(3):032324, Sep 2012.
- [18] Brown *et al.* Poking holes and cutting corners to achieve clifford gates with the surface code. *Phys. Rev. X*, 7(2):021029, May 2017.
- [19] Basso Basset *et al.* Entanglement swapping with photons generated on demand by a quantum dot. *Phys. Rev. Lett.*, 123(16):160501, Oct 2019.
- [20] D. Gottesman. The heisenberg representation of quantum computers. Jul 1998.
- [21] Hans J. Briegel and Robert Raussendorf. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.*, 86(5):910–913, Jan 2001.
- [22] Van den Nest *et al.* Graphical description of the action of local clifford transformations on graph states. *Phys. Rev. A*, 69(2):022316, Feb 2004.
- [23] Bennett *et al.* Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70(13):1895–1899, Mar 1993.
- [24] Daniel Gottesman and Isaac L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390–393, Nov 1999.
- [25] Wang *et al.* Integrated photonic quantum technologies. *Nature Photonics*, 14(5):273–284, May 2020.
- [26] Christopher C. Gerry and Peter L. Knight. *Introductory quantum optics*. Cambridge University Press, 1st edition.
- [27] Terry Rudolph. Presentation at "frontiers of quantum science". <https://www.youtube.com/watch?v=FISagFmx0aU>, 2019.
- [28] Pieter Kok. Five lectures on optical quantum computing. *Theoretical Foundations of Quantum Information Processing and Communication*, (1616-6361):187–219, Sep 2009.
- [29] William R. Clements *et al.* Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, Dec 2016.
- [30] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, Jan 2001.
- [31] Sara Bartolucci *et al.* Fusion-based quantum computation. <https://arxiv.org/abs/2101.09310v1>, 2021.
- [32] Borja Peropadre, Alan Aspuru-Guzik, and Juan Jose Garcia-Ripoll. Spin models and boson sampling. <https://arxiv.org/abs/1509.02703>, 2016.

- [33] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, page 333–342, 2011.
- [34] Jahangiri *et al.* Quantum algorithm for simulating molecular vibrational excitations. *Phys. Chem. Chem. Phys.*, 22(44):25528–25537, Oct 2020.
- [35] Huh *et al.* Boson sampling for molecular vibronic spectra. *Nature Photonics*, 9(9):615–620, Aug 2015.
- [36] Juan Miguel Arrazola and Thomas R. Bromley. Using gaussian boson sampling to find dense subgraphs. *Phys. Rev. Lett.*, 121(3):030503, July 2018.
- [37] Banchi *et al.* Molecular docking with gaussian boson sampling. *Science Advances*, 6(23):eaax1950, Jun 2020.
- [38] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
- [39] S. Scheel. Permanents in linear optical networks, 2004. <https://arxiv.org/abs/quant-ph/0406127v1>.
- [40] C. K. Hong, Z. Y. Ou, and L. Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Phys. Rev. Lett.*, 59(18):2044–2046, Nov 1987.
- [41] Gard *et al.* An introduction to boson-sampling. *From Atomic to Mesoscale*, page 167–192, Jun 2015.
- [42] Jacek Wesolowski and Grzegorz Rempala. *Symmetric Functionals on Random Matrices and Random Matchings Problems*. Springer, 1st edition.
- [43] Alex Arkhipov and Greg Kuperberg. The bosonic birthday paradox. <https://arxiv.org/abs/1106.0849>, 2011.
- [44] Peter P. Rohde, Keith R. Motes, Paul A. Knott, and William J. Munro. Will boson-sampling ever disprove the extended church-turing thesis? <https://arxiv.org/abs/1401.2199>, 2014.
- [45] Daniel J. Brod *et al.* Photonic implementation of boson sampling: a review. *Advanced Photonics*, 1(3):1–14, May 2019.
- [46] Killoran *et al.* Strawberry fields: A software platform for photonic quantum computing. *Quantum*, 3(2521-327X):129, Mar 2019.
- [47] Yong Lu *et al.* Quantum efficiency, purity and stability of a tunable, narrowband microwave single-photon source. <https://arxiv.org/abs/2105.11234>, 2021.
- [48] Hamilton *et al.* Gaussian boson sampling. *Phys. Rev. Lett.*, 119(17):170501, Oct 2017.
- [49] Matteo G. A. Paris Alessandro Ferraro, Stefano Olivares. Gaussian states in continuous variable quantum information. <https://arxiv.org/abs/quant-ph/0503237>, 2005.

- [50] Arrazola *et al.* Quantum circuits with many photons on a programmable nanophotonic chip. *Nature*, 591(7848):54–60, Mar 2021.
- [51] Ville Bergholm *et al.* PennyLane: Automatic differentiation of hybrid quantum-classical computations. <https://arxiv.org/abs/1811.04968>, 2020.
- [52] PennyLane dev team. Understanding the haar measure. https://pennyLane.ai/qml/demos/tutorial_haar_measure.html, 2021.
- [53] Francesco Mezzadri. How to generate random matrices from the classical compact groups. *NOTICES of the AMS*, 54:592–604, 2007.
- [54] Scipy. <https://docs.scipy.org/doc/scipy/reference/index.html>.
- [55] Boson sampling and the permanent. https://strawberryfields.ai/photonics/demos/run_boson_sampling.html.
- [56] Numpy. <https://numpy.org/doc/stable/>.
- [57] Matplotlib. <https://matplotlib.org/stable/contents.html>.
- [58] Thewalrus. <https://the-walrus.readthedocs.io/en/latest/code/thewalrus.html>.
- [59] Gaussian boson sampling and the hafnian. https://strawberryfields.ai/photonics/demos/run_gaussian_boson_sampling.html.