SAPIENZA
UNIVERSITÀ DI ROMA

"Sapienza" University of Rome
Faculty of Information Engineering,
Informatics and Statistics
Department of Computer Science

Quantum Computing

Lecture notes integrated with the book "Quantum Computation and Quantum Information", Michael A. Nielsen, Isaac L. Chuang

Author
Alessio Bandiera

January 3, 2026

Contents

Information and Contacts **1**

1 Introduction on Quantum Computation **2**
1.1 The Qubit . . . 3
1.2 Qubit operations . . . 4
1.2.1 The tensor product . . . 6
1.2.2 Controlled operations . . . 8
1.2.3 Quantum circuits . . . 9
1.3 Peculiarities of quantum mechanics . . . 11
1.3.1 Quantum entanglement . . . 11
1.3.2 Quantum teleportation . . . 13

2 Mathematical foundations **19**
2.1 Hilbert spaces . . . 22
2.1.1 Linear operators . . . 24
2.2 Spectral theory . . . 29
2.3 Projectors . . . 30
2.4 Tensor product . . . 34
2.4.1 General definition . . . 34
2.4.2 Kronecker product . . . 38

3 Postulates of quantum mechanics **40**
3.1 Postulates of quantum mechanics . . . 40
3.1.1 First postulate . . . 40
3.1.2 Second postulate . . . 40
3.1.3 Third postulate . . . 42
3.1.4 Fourth postulate . . . 45
3.2 The density operator . . . 46
3.2.1 Generalization of the third postulate . . . 47
3.2.2 Postulates through density operators . . . 48
3.2.3 Entangled systems . . . 53

4 Quantum algorithms **56**
4.1 Deutsch's algorithm . . . 57
4.2 Deutsch-Josza algorithm . . . 62

5 Quantum Phase Estimation **67**
5.1 A recap on the Discrete Fourier Transform . . . 67
5.2 The Quantum Fourier Transform . . . 71
5.2.1 The quantum circuit . . . 75
5.3 Quantum Phase Estimation . . . 77
5.4 Variational Quantum Algorithms . . . 81
5.4.1 Variable Quantum Eigensolver . . . 82

6 Shor's algorithm **87**
6.1 Order-finding problem . . . 87
6.2 Shor's algorithm . . . 96

7 Grover's algorithm **98**
7.1 The search problem . . . 98
7.1.1 Another perspective . . . 107
7.2 Fixed-Point Quantum Search . . . 108
7.3 Quantum counting . . . 113

8 Quantum circuits **115**
8.1 Rotation operators . . . 115
8.2 Controlled operators . . . 118
8.2.1 Single-qubit conditioning . . . 118
8.2.2 Multi-qubit conditioning . . . 120
8.3 Universality . . . 123
8.3.1 Exact quantum universality . . . 125
8.3.2 Approximate universality . . . 126

9 Quantum Key Distribution **129**
9.1 Background . . . 130
9.1.1 The No-cloning theorem . . . 130
9.1.2 Measurements . . . 131
9.2 The BB84 protocol . . . 133
9.2.1 Eve's attacks . . . 135
9.3 Bell's inequalities . . . 138

10 Quantum Error Correction **143**
10.1 Repetition codes . . . 143
10.1.1 Three bits bit flip code . . . 143
10.1.2 Three qubits bit flip code . . . 144
10.1.3 Three qubits phase flip code . . . 146
10.1.4 The Shor code . . . 147
10.2 Linear codes . . . 149
10.2.1 The Hamming code . . . 154
10.2.2 The Steane code . . . 156

Information and Contacts

Personal notes and summaries collected as part of the *Quantum Computing* course offered by the degree in Computer Science of the University of Rome "La Sapienza".

Further information and notes can be found at the following link:
https://github.com/aflaag-notes. Anyone can feel free to report inaccuracies, improvements or requests through the Issue system provided by GitHub itself or by contacting the author privately:

- Email: **alessio.bandiera02@gmail.com**
- LinkedIn: **Alessio Bandiera**

The notes are constantly being updated, so please check if the changes have already been made in the most recent version.

Suggested prerequisites:

TODO

Licence:

These documents are distributed under the **GNU Free Documentation License**, a form of copyleft intended for use on a manual, textbook or other documents. Material licensed under the current version of the license can be used for any purpose, as long as the use meets certain conditions:

- All previous authors of the work must be **attributed**.
- All changes to the work must be **logged**.
- All derivative works must be **licensed under the same license**.
- The full text of the license, unmodified invariant sections as defined by the author if any, and any other added warranty disclaimers (such as a general disclaimer alerting readers that the document may not be accurate for example) and copyright notices from previous versions must be maintained.
- Technical measures such as DRM may not be used to control or obstruct distribution or editing of the document.

1

Introduction on Quantum Computation

TODO introduction on QM

Quantum computing is a rapidly developing discipline that explores how the laws of quantum mechanics can be used to *process information*. While classical computation is based on *bits* that take values of either 0 or 1, quantum computation relies on quantum bits, or **qubits**. A qubit can exist in a "superposition" of classical states, allowing it to encode richer information than a single bit. Furthermore, qubits can exhibit particular properties that enable forms of information processing with no classical counterpart. Such properties provide the foundation for algorithms that promise to solve certain problems more efficiently than their classical analogues.

The design of quantum algorithms requires a different perspective from that of classical computation. In classical computer science, the majority of widely studied algorithms are *deterministic*, meaning that for a given input they will always produce the *same output*. Some algorithms are *randomized*, making use of probability to achieve efficiency or simplicity, yet even in those cases the computation itself is ultimately classical in nature. In fact, to achieve such *randomness* classical algorithms employ **pseudo-random number generation**, which must ultimately produce finite sequences.

Quantum computation, by contrast, *incorporates probability* at its core. The act of measuring a quantum system does not reveal a single, predetermined result, but rather yields one outcome from a distribution of possible outcomes, with probabilities governed by the system's quantum state. This fundamental probabilistic characteristic distinguishes quantum algorithms from their classical counterparts.

In fact, in the context of quantum computing we are often interested in **probabilistic algorithms**: for such algorithms, a given input i can lead to a finite set of possible outputs $o_1, \dots, o_N$, each occurring with an associated probability $p_1, \dots, p_N$ — where $\sum_{i=1}^{n} p_i = 1$.

1.1 The Qubit

First, we will introduce the **qubit**, the quantum equivalent of the classical bits. First, consider the following vectors: these are called **basis states**

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and they represent the classical bits 0 and 1 respectively — the notation above is called "braket" notation and it will be explored in greater detail in the following chapter.

So what is a qubit? A qubit is the basic unit of information in quantum computing, which represents a **superposition** of states simultaneously. Mathematically speaking, the state of a qubit is a vector

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

where $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$ are called **probability amplitudes**. But why are we talking about probabilities in the first place? The "true" state of a qubit **cannot be observed**, and we say that the qubit is in a *superpotion* of $|0\rangle$ and $|1\rangle$ in the sense that α and β describe the probabilities of getting either states once the qubit is measured. This is because to know the value of a qubit we have to *measure it*, and the measurement operation itself will make the qubit *collapse* into either $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively, i.e.

$$\Pr[\text{measured qubit is } |0\rangle] = |\alpha|^2 \qquad \Pr[\text{measured qubit is } |1\rangle] = |\beta|^2$$

To use a more compact notation, we will denote this property as follows:

$$\alpha\,|0\rangle + \beta\,|1\rangle \begin{cases} |0\rangle & @ \ |\alpha|^2 \\ |1\rangle & @ \ |\beta|^2 \end{cases}$$

where the @ notation (read as "at") denotes the probabilty of the corresponding outcome. Note that if we measure a collapsed qubit we will keep observing the same state indefinitely.

In reality, to be precise qubits actually collapse into any multiple $z\,|0\rangle$ or $z\,|1\rangle$, where $z \in \mathbb{C}$ is a complex number such that $|z| = 1$, but this is not relevant from a physical point of view. In fact, for any θ physicists treat $|\psi\rangle = |0\rangle$ and $|\psi'\rangle = e^{i\theta}\,|0\rangle$ as the *same physical state*, because probabilities depend on squared magnitudes and thus

$$\left|e^{i\theta}\alpha\right|^2 = |\alpha|^2$$

(and the same applies for β too) even though $|\psi\rangle$ and $|\psi'\rangle$ are different vectors mathematically. Therefore, in general we can actually drop the **global phases** from the qubits entirely.

1.2 Qubit operations

What can we do with qubits other then *measure them*? The operations that can be applied on qubits are restricted to **unitary transformations**, which are linear maps that *preserve the norms* — we will discuss the precise definition of this concept in the next chapter. For instance, the identity matrix I is an example of trivial unitary transformation, but also the NOT matrix

$$\text{NOT} := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

which has the effect of *swapping* the input basis state

$$\text{NOT}\,|0\rangle = |1\rangle \qquad \text{NOT}\,|1\rangle = |0\rangle$$

is a unitary transformation. Note that this matrix behaves as the classical NOT gate acting on bits in classical computations, in fact will we refer to *transformations* and *gates* interchangeably.

More in general, the NOT operation belongs to a family of operations represented by the so called **Pauli matrices**.

Definition 1.1: Pauli matrices

The **Pauli matrices** are the following four 2×2 matrices:

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

In particular, we observe that the second matrix σ_x is exactly the matrix of the NOT operator. For this reason, the NOT operator is also called X, and we will see the Z and Y operators (representing the other two matrices, respectively) as well in later sections.

Another very important transformation is represented by the **Hadamard gate**, which is the following matrix

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

This matrix has the effect of "mapping" classical states into superpositions:

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

For instance, in this example given $|0\rangle$ which represents the classical bit 0, we get a qubit as output of the linear transformation. In general, the operation performed by the Hadamard gate can be represented as follows:

$$\forall a \in \{0, 1\} \quad \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^a\,|1\rangle\right)$$

As a side note, as we mentioned at the beginning of the chapter quantum mechanics has randomness intrinsically, and since the operation $H\,|0\rangle$ returns a qubit that has 50% of

probability of being either $|0\rangle$ or $|1\rangle$ once measured, this operation provides a true random number generator.

Lastly, can we *represent* qubits graphically? Well, we may be tempted to anwer negatively to this question, since a qubit is described by two complex numbers $\alpha, \beta \in \mathbb{C}$, which implies that we actually need 4 dimensions to correctly represent our vector. However, through polar coordinates we can actually define a graphical representation which allows us to "picture" qubits, through the so called **Bloch sphere**. First, consider a qubit

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

for some $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$, as usual. Now, recalling that any complex number $z \in \mathbb{C}$ can be actually written as follows

$$z = |z|\, e^{i\theta}$$

for some angle θ, we can rewrite our qubit as follows:

$$\begin{aligned}
|\psi\rangle &= |\alpha|\, e^{i\theta_\alpha}\,|0\rangle + |\beta|\, e^{i\theta_\beta}\,|1\rangle \\
&= e^{i\theta_\alpha}\left(|\alpha|\,|0\rangle + |\beta|\, e^{i\left(\theta_\beta - \theta_\alpha\right)}\,|1\rangle\right) \\
&= |\alpha|\,|0\rangle + |\beta|\, e^{i\left(\theta_\beta - \theta_\alpha\right)}\,|1\rangle && \left(e^{i\theta_\alpha} \text{ is a global phase}\right) \\
&= |\alpha|\,|0\rangle + e^{i\varphi}\,|1\rangle && (\text{let } \varphi := \theta_\beta - \theta_\alpha \in [0, 2\pi))
\end{aligned}$$

and finally, since $|\alpha|^2 + |\beta|^2 = 1$ is precisely the equation of the circumference of radius 1, we usually rewrite the last equation as follows:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

where $\theta \in [0, \pi], \varphi \in [0, 2\pi)$. This formulation of the qubit $|\psi\rangle$ allows us to represent it inside the Bloch sphere: in fact, in this formulation the qubit is normalized, which implies that it will lie on a 3 dimensional unit sphere, and it is described by the two phases θ and φ — in 2D polar coordinates there is only 1 angle, as in 3D polar coordinates there are 2 angles.

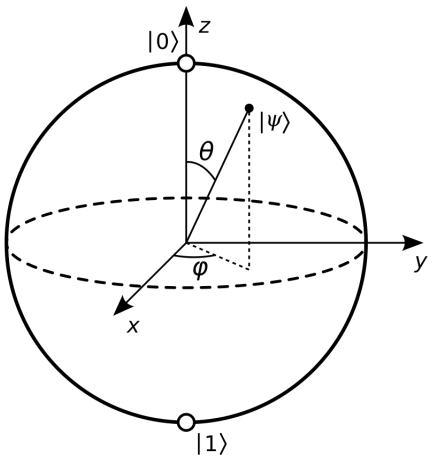


Figure 1.1: The Bloch sphere representing some qubit.

1.2.1 The tensor product

So far we have dealt with only one qubit at a time, but what if we have two qubits? First, let's look at the classical counterpart. If we take two bits $a, b \in \{0, 1\}$, we can represent 4 possible binary numbers, namely 00, 01, 10 and 11, which we can algebraically obtain by computing the usual cartesian product

$$\{0,1\}^2 = \{0,1\} \times \{0,1\} = \{(0,0),(0,1),(1,0),(1,1)\}$$

Note that in the cartesian products it holds that:

- the length of the tuples of the product is linear w.r.t. the number of factors of the cartesian products — in this case, 2
- each element of a tuple is *independent* from the other elements of the tuple

How can we evaluate all the possible states that two qubits can represent, instead? To answer this question, we need to introduce a new operator, which is called **tensor product**. Given two vectors $\begin{pmatrix} a \\ b \end{pmatrix}$ and $\begin{pmatrix} c \\ d \end{pmatrix}$, their tensor product is defined as follows

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} := \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

Hence, consider two qubits

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \qquad |\phi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

To obtain all the possible states of $|\psi\rangle$ and $|\phi\rangle$ we just have to compute the tensor product between them, which is

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle &= \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \\ &= \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_1 \\ \alpha_1\beta_1 \end{pmatrix} \\ &= \alpha_0\beta_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \alpha_0\beta_1 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \alpha_1\beta_0 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \alpha_1\beta_1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

At the beginning of the chapter we defined $|0\rangle$ and $|1\rangle$ to be $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ without providing an explaination; now that we are dealing with more than 2 dimensions we can

show why such names are used. In fact, we will use the following naming convention

$$|00\rangle := \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \quad |01\rangle := \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad |10\rangle := \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \quad |11\rangle := \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

and in general it holds that

$$|\text{bin}(i)\rangle = e_i$$

where $\text{bin}(i)$ represents the binary representation of i, and e_i is the i-th vector of the canonical basis. This implies that we can rewrite the previous tensor product as follows:

$$|\psi\rangle \otimes |\phi\rangle = \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle = \sum_{i,j\in\{0,1\}} \alpha_i\beta_j |ij\rangle$$

where the last sum comes from the fact that

$$\forall i, j \in \mathbb{B} \quad |i\rangle \otimes |j\rangle = |ij\rangle$$

In general, we will follow this convention:

- if we write $|b\rangle$ where b is a binary string $b \in \mathbb{B}^n$ for some length n, we are referring to a vector of the canonical basis — something like $|01\rangle$, $|0\rangle$, $|110\rangle$ etc.
- if we write $|x\rangle$, $|y\rangle$, $|a\rangle$ or any other latin letter inside the label of the "braket" notation, we implicitly mean that x, y and a are treated as binary numbers and these vectors are to be intended still as elements of the canonical basis
- if we write $|\psi\rangle$, $|\phi\rangle$ or any other greek letter inside the label of the "braket" notation, we are referring to a *superposition* of basis states — something like

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$$

The basis of the 2D space formed by $|0\rangle$ and $|1\rangle$ is usually called **computational basis**, since we can directly map $|0\rangle$ to the classical 0, and $|1\rangle$ to the classical 1. Alternatively, it is also called **Z basis**, as opposed to the **X basis** which we will encounter in the next chapters.

Let's do an example of tensor product between two qubits defined in the Z basis: consider

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

When computing their tensor product we get that

$$
\begin{aligned}
|\psi\rangle \otimes |\phi\rangle &= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \\
&= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\
&\begin{cases} |0\rangle \text{ and } |0\rangle & @ \ \frac{1}{4} \\ |0\rangle \text{ and } |1\rangle & @ \ \frac{1}{4} \\ |1\rangle \text{ and } |0\rangle & @ \ \frac{1}{4} \\ |1\rangle \text{ and } |1\rangle & @ \ \frac{1}{4} \end{cases}
\end{aligned}
$$

where the probabilities at the end refer to the two individual qubits. To recap, in general the tensor product $|\psi\rangle \otimes |\phi\rangle$ of two qubits encodes the superposition of 4 basis states, namely $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$.

Moreover, the following property can be proved easily.

Proposition 1.1: Distributive property of $\otimes$

Given three qubits $|\psi\rangle, |\phi\rangle$ and $|\chi\rangle$, it holds that

$$(|\psi\rangle + |\phi\rangle) \otimes |\chi\rangle = |\psi\rangle \otimes |\chi\rangle + |\phi\rangle \otimes |\chi\rangle$$

1.2.2 Controlled operations

Another family of very important gates in quantum computing is the *controlled operators*, but first let's define them classically. The first controlled operation that we are going to discuss is the **Controlled NOT (CNOT)** gate, which is defined as follows:

a	b	CNOT(a, b)
0	0	0
0	1	1
1	0	1
1	1	0

The name comes from the fact that the first input a is called *control bit*, which if set to 1 will flip the *target bit* b. Therefore, in general we have that

$$\text{CNOT}(a, b) = a \oplus b$$

First, we observe that this function is clearly *not invertible*, since for instance if we know that the output is 0 we still need the input a to evaluate if b was 0 or 1. Hence, because we

would like this computation to be invertible — and we will see why later in our discussion — to solve this issue we usually pair the output of CNOT with a itself, so that we can actually invert the computation:

$$\text{CNOT}(a, b) = (a, a \oplus b)$$

We will start from this operator in order to construct its equivalent quantum gate.

So far we only dealt with transformations that expected only 1 qubit argument as input, but the CNOT gate would certainly need 2 inputs to perform any computation, so how do we provide two inputs to it? As we showed before, we know that

$$\forall i, j \in \{0, 1\} \quad |i\rangle \otimes |j\rangle = |ij\rangle$$

which directly implies that the vector $|ij\rangle$ encapsulates two qubits at once *without ambiguity*. Hence, we can actually leverage the tensor product to provide the input to the CNOT matrix, such that the quantum CNOT will behave as follows

$$\text{CNOT}(|a\rangle \otimes |b\rangle) = |a\rangle \otimes |a \oplus b\rangle$$

The matrix that behaves as such is the following

$$\text{CNOT} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

which expects a 4×1 input vector, and outputs a 4×1 output vector as well. Laslty, as for the CNOT operator, we can actually define controlled operators for both Y and Z, which are respectively called CY and CZ operators.

1.2.3 Quantum circuits

Now that we introduced a couple of quantum gates, we can show how computation is actually represented in quantum computing. For instance, consider the following picture:

Figure 1.2: The NOT gate.

In this example, we have 1 single input qubit, namely q, and the box labeled with an X represents the NOT gate. We observe that, by convention, all qubits in quantum circuits are assumed to be set to $|0\rangle$.

In the following example, instead, it is represented how the Hadamard gate looks like in quantum circuits.

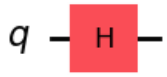


Figure 1.3: The Hadamard gate.

Moreover, if we consider two qubits as inputs q_0 and q_1, we can represent the CNOT operator as follows:

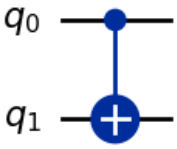


Figure 1.4: The CNOT gate.

We observe that q_1 then becomes the output of the CNOT operation, and q_0 remains unchanged. Lastly, the measurement operation is represented with the following picure:

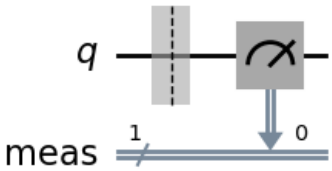


Figure 1.5: The measure operation.

In particular, in this cirtuit we see that:

- the vertical "double line" reprents *a measured state* — which we usually call *classical wire* (even if it's still a qubit) because it became either $|0\rangle$ or $|1\rangle$
- the number 1 next to the label `meas` indicates the number of qubits that have been measured
- the number 0 is the index of the measured qubit

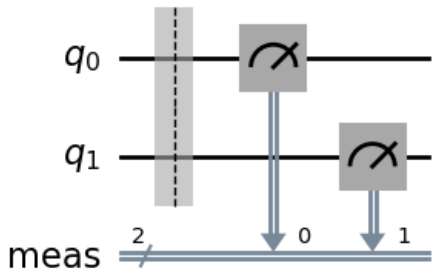


Figure 1.6: An exmaple of measurement of 2 qubits.

Lastly, another very important circuit is the following, which produces the so called **Greenberger-Horne-Zeilinger (GHZ)** state

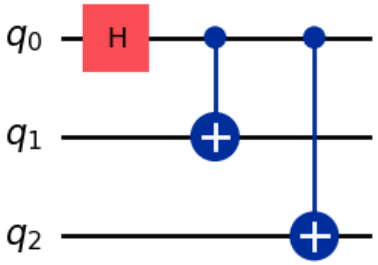


Figure 1.7: The GHZ quantum circuit.

which is represented as follows

$$|\text{GHZ}\rangle := \frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right)$$

All of these examples were drawn using Qiskit, which is a very popular framework that is used to describe quantum computation.

1.3 Peculiarities of quantum mechanics

1.3.1 Quantum entanglement

Consider the following quantum state

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$$

Can this state be rewritten as the tensor product of two distinct quantum states? We observe that for this to be possible we would require some complex values $\alpha_0, \alpha_1, \beta_0, \beta_1$

such that

$$\begin{cases} \alpha_0\beta_0 = \alpha_1\beta_1 = 0 \\ \alpha_0\beta_1 = \alpha_1\beta_0 = \frac{1}{\sqrt{2}} \end{cases}$$

but $\alpha_0\beta_0 = 0$ implies that at least one between α_0 and β_0 has to be 0, meaning that at least one between $\alpha_0\beta_1$ and $\alpha_1\beta_0$ has to be 0 as well. This proves that there is no such pair of quantum states which can describe $|\psi\rangle$ through the tensor product operation. In fact, we see that

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \begin{cases} |01\rangle & @\ \frac{1}{2} \\ |10\rangle & @\ \frac{1}{2} \end{cases}$$

Indeed, the choice of $|\psi\rangle$ was not a coincidence, as it is one of the so called **Bell states**.

Definition 1.2: Bell states

The following are the four **Bell states**:

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

$$|\Phi^-\rangle := \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle := \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\Psi^-\rangle := \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

Whenever we have a state $|\psi\rangle$ that cannot be represented as the tensor product of two simpler quantum states, we say that the state is **entangled** — or that its possible outcomes are entangled. In particular, entangled states describe a very weird phenomenon first proposed as a thought experiment in a groundbreaking paper by **Einstein, Podolsky and Rosen (EPR)** [EPR35], the so called **EPR paradox**.

The thought experiment involves a pair of particles prepared in such *entangled state*. Einstein, Podolsky, and Rosen pointed out that, in this state, if the position of the first particle were measured, the result of measuring the position of the second particle *could be predicted*. If instead the momentum of the first particle were measured, then the result of measuring the momentum of the second particle could be predicted. Einstein famously called this phenomenon "spooky action at a distance", and to the best of our knowledge the theory of quantum mechanics says that if we have two engangled states, and measure one of them — for instance, say that it collapses to $|0\rangle$ — the other state will **instantaneously** collapse to $|1\rangle$ (and viceversa). They are *perfectly anti-correlated*, even if the two states are phisically light-years away from each other. For instance, in our previous example with $|\psi\rangle$ (which we now know is actually $|\Psi^+\rangle$) is that if we have two qubits q_0 and q_1 such that their *total state* is $|\psi\rangle$,l whenever we measure one of them it will collapse in either $|01\rangle$ or $|10\rangle$ with 50%, and the other qubit will collapse with the

opposing outcome.

To be precise, entanglement is not a way to tranfer information — collapsing happens instantaneously, which would violate the fact that nothing can travel faster than light, not even information. Instead, it is a way to share correlations nonlocally. In fact, it is a phenomenon that concerns the *whole quantum system* considered: for instance, given three qubits q_0, q_1, q_2, such that q_1 and q_2 are entangled, we might want to only measure $q_0 \otimes q_1$, which in turn will make q_2 collapse into some quantum state that has to be mathematically computed in order to be predicted — this will be more clear when we will describe **quantum teleportation** in Section 1.3.2.

To finish off this section, we can actually generate entangled states, or **EPR pairs** for short, through quantum gates as such:

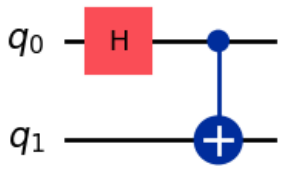


Figure 1.8: The quantum circuit for $|\Phi^+\rangle$.

In particular, we observe that the first Hadamard gate will transform $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and through the CNOT operation we obtain

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

1.3.2 Quantum teleportation

An operation that we take for granted in classical computation is the possibility to *copy* the value of a bit: if Alice has two bits $x, y \in \{0, 1\}$, and she wants to copy the value of x into y, she can do it without any issues. However, in quantum mechanics this is not so straightforward. In 1982 Wootters and Zurek [WZ82] proved the so called **No-cloning theorem**, which states that it is impossible to create an independent and identical copy of an arbitrary quantum state. However, in order to state the theorem properly we need to introduce a lot of mathematical tools which will be discussed in the following chapter. We will prove the theorem formally in Section 9.1.1, but for now we will assume that cloning an arbitrary $|\psi\rangle$ is *not* possible.

Hence, if quantum states cannot be cloned, can we at least *send* them? Suppose that Alice wants to send Bob $|\psi\rangle$, described by some α and β. Clearly, the only thing that Bob has to receive are indeed the probability amplitudes of $|\psi\rangle$, so even if Alice cannot clone her quantum state, nothing prevents her to build a quantum circuit which allows Bob to receive α and β — at the cost of "destroying" her $|\psi\rangle$, i.e. mesuring it. This

process is called **quantum teleportation**, and can be achieved through the following algorithm.

Algorithm 1.1: Quantum Teleportation algorithm

Given three qubits q_0, q_1 and q_2, the algorithm moves the state of q_0 into q_2.

---

function QUANTUMTELEPORTATION(q_0, q_1, q_2)
  $q_1 \leftarrow H(q_1)$
  $q_2 \leftarrow CX(q_1, q_2)$ ▷ entangle q_1 and q_2
  $q_1 \leftarrow CX(q_0, q_1)$
  $q_0, q_1 \leftarrow \text{measure}(q_0, q_1)$
  if $q_1 == |1\rangle$ **then**
    $q_2 \leftarrow X(q_2)$
  end if
  if $q_0 == |1\rangle$ **then**
    $q_2 \leftarrow Z(q_2)$
  end if
  return q_2
end function

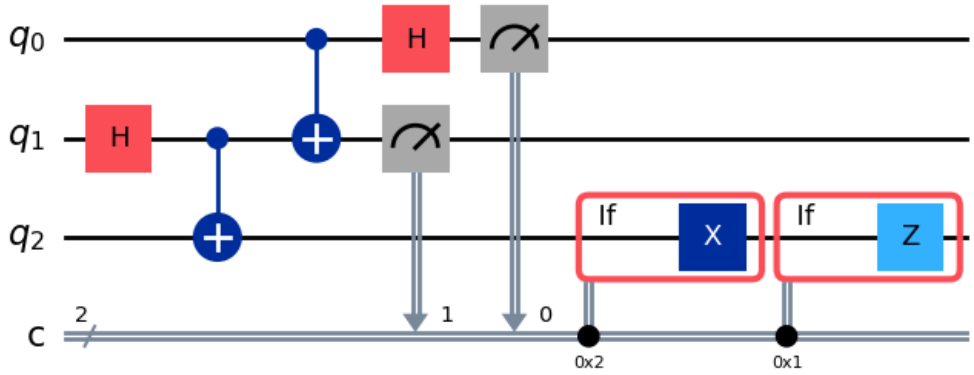


Figure 1.9: The Quantum Teleportation circuit, drawn with Qiskit.

There is quite a lot to unpack in this diagram. First, the quantum state that we want to teleport is q_0, and it will be teleported in q_2 at the end of the quantum computation.

In the first part of the circuit, we see that q_1 and q_2 are entangled (in an initial stage of the process, not performed by Alice nor Bob) in the Bell state $|\Phi^+\rangle$ thanks to the Hadamard and the CNOT gates — as we described in previous sections. In a real-world scenario, we will assume that q_1 and q_2 are given to Alice and Bob respectively (through some **quantum channel** such as optical fibers or free-space links in order to avoid *decoherence*), and quantum mechanics will guarantee that the teleportation will work even if our two

protagonists are thousands of kilometers away from each other.

After creating and entangling q_1 and q_2 (say for instance in a lab as preparation), we have the part of circuit that concerns Alice: in fact, she must apply a CNOT to her entangled qubit q_1, controlled by q_0, and then apply a Hadamard transformation to q_0. At this point, the circuit must apply a measurement to both q_0 and q_1 — and in particular, this operation will *destroy* the original state as previously anticipated.

Finally, it's Bob's turn: to obtain the original quantum state of q_0, the only thing he needs to do is first apply a CNOT to his entangled qubit q_2, controlled by q_1's outcome, followed by an application of a CZ, controlled by q_0's outcome instead — we observe that this part is indicated in the diagram through the `0x2` and `0x1` labels respectively. In fact, in the label `0xX` the number `X` represents the hexadecimal representation of the binary number obtained by joining the classical bits all together; for instance, in this circuit we have that `0x2` represents 10, meaning that only q_1 will be checked in the condition, and `0x1` represents 01, which means that only q_0 will be the control bit.

To show why the circuit actually works, we first need to discuss how computations with qubits and quantum gates is performed. In particular, we do not consider qubits *individually*, but instead we consider the whole **system** of qubits, i.e. $q_0 \otimes q_1 \otimes q_2$ simultaneously, and thus we will perform calculations as such. In fact, even if the drawing represents Alice's measurements of q_0 and q_1 independently, what happens in reality is that Alice is going to measure $q_0 \otimes q_1$ such that q_3 will collapse into its opposite.

Finally, we are ready to prove the correctness of the quantum teleportation circuit. Say that the qubit that Alice wants to send is

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Then, we have that

$$\begin{aligned}
&q_0 \otimes q_1 \otimes q_2 \\
=&\,|\psi\rangle \otimes |\Phi^+\rangle \\
=&(\alpha\,|0\rangle_0 + \beta\,|1\rangle_0) \otimes \frac{1}{\sqrt{2}}(|00\rangle_{12} + |11\rangle_{12}) \\
=&\frac{1}{\sqrt{2}}[\alpha\,|0\rangle_0 \otimes (|00\rangle_{12} + |11\rangle_{12}) + \beta\,|1\rangle_0 \otimes (|00\rangle_{12} + |11\rangle_{12})]
\end{aligned}$$

where the notation $|00\rangle_{12}$ represents for example that we are considering q_1 and q_2's parts of states, respectively. From now on, we will omit the $\otimes$ symbol — as for the "normal" product. The next step is to apply the CNOT on q_0 and q_1, therefore the quantum state of the system becomes the following:

$$\frac{1}{\sqrt{2}}\left[\alpha\,|0\rangle_0\,(|00\rangle_{12} + |11\rangle_{12}) + \beta\,|1\rangle_0\,(|10\rangle_{12} + |01\rangle_{12})\right]$$

Next, we have to apply the Hadamard gate on q_0, which turns the quantum state into the

following

$$
\begin{aligned}
&\frac{1}{\sqrt{2}}\left[\alpha\frac{1}{\sqrt{2}}(|0\rangle_0+|1\rangle_0)(|00\rangle_{12}+|11\rangle_{12})+\beta\frac{1}{\sqrt{2}}(|0\rangle_0-|1\rangle_0)(|10\rangle_{12}+|01\rangle_{12})\right]\\
=&\frac{1}{2}\left[\alpha(|000\rangle_{012}+|011\rangle_{012}+|100\rangle_{012}+|111\rangle_{012})+\beta(|010\rangle_{012}+|001\rangle_{012}-|110\rangle_{012}-|101\rangle_{012})\right]\\
=&\frac{1}{2}\left[|00\rangle_{01}(\alpha|0\rangle_2+\beta|1\rangle_2)+|01\rangle_{01}(\beta|0\rangle_2+\alpha|1\rangle_2)+|10\rangle_{01}(\alpha|0\rangle_2-\beta|1\rangle_2)+|11\rangle_{01}(\alpha|1\rangle_2-\beta|0\rangle_2)\right]\\
=&\frac{1}{2}\left[|00\rangle_{01}|\psi\rangle_2+|01\rangle_{01}X|\psi\rangle_2+|10\rangle_{01}Z|\psi\rangle_2+|11\rangle_{01}XZ|\psi\rangle_2\right]
\end{aligned}
$$

Finally, Alice will perform the measurement on q_0 and q_1, and what will happen is that the *whole* quantum state of the quantum circuit will collapse as follows:

$$
\begin{cases}
|00\rangle_{01}\otimes|\psi\rangle_2 & @\ \frac{1}{4}\\
|01\rangle_{01}\otimes X|\psi\rangle_2 & @\ \frac{1}{4}\\
|10\rangle_{01}\otimes Z|\psi\rangle_2 & @\ \frac{1}{4}\\
|11\rangle_{01}\otimes XZ|\psi\rangle_2 & @\ \frac{1}{4}
\end{cases}
$$

In fact, from this table we can easily explain the last part of the quantum teleportation circuit, i.e. Bob's part, as shown below.

Alice's outcome	Bob's part	Bob's result
0 and 0	I	$\|\psi\rangle$
0 and 1	X	$XX\|\psi\rangle=\|\psi\rangle$
1 and 0	Z	$ZZ\|\psi\rangle=\|\psi\rangle$
1 and 1	XZ	$XZXZ\|\psi\rangle=\|\psi\rangle$

Thus, in the end Bob was able to reconstruct $|\psi\rangle$ correctly, at the cost of having measured q_0, which means that Alice lost her superposition of states forever.

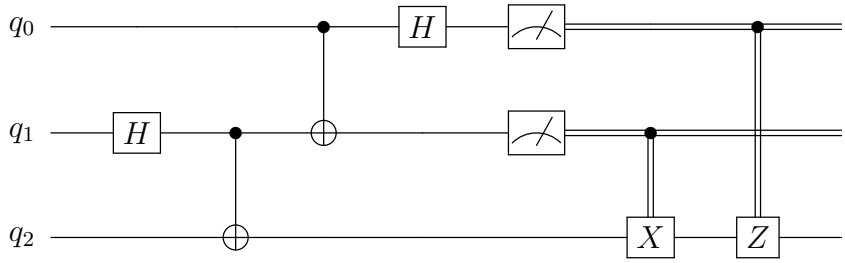


Figure 1.10: The Quantum Teleportation circuit.

As a final note, we observe that the circuit written in this form is rather unusual. Indeed, it helps to visualize the fact that Alice performs the measurement before Bob can have his qubit collapse, but in reality the measurement could have been done at the end of the circuit and it would not have made any difference at all.

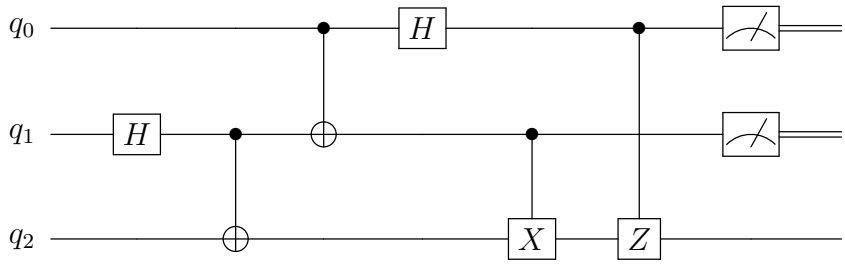


Figure 1.11: The Quantum Teleportation circuit with the measurement moved at the end.

Indeed, all the quantum circuits we will present from now on will have their measurements placed at the end of the whole computation. This idea is summed up below.

Principle 1.1: Principle of deferred measurement

Measurement can always be moved from an intermediate stage of a quantum circuit to the end of the circuit. If the measurement results are used at any stage of the circuit then the classically controlled operations can be replaced by conditional quantum operations.

Algorithmically, this means that the following sequence of operations on two qubits q_0, q_1

1: $b \leftarrow \text{measure}(q_0)$
2: **if** b **then**
3: $\quad q_1 \leftarrow U(q_1)$
4: **end if**

which performs an application of the U gate *classically conditioned* on the result of b, can be equivalently rewritten as

1: $q_0, q_1 \leftarrow \text{C-}U(q_0, q_1)$
2: $b \leftarrow \text{measure}(q_0)$

assuming that C-U is the *controlled* version of the U gate. Let's prove that this is true. Assume that the complete state of the system $|\psi\rangle$ is equal to

$$q_0 \otimes q_1 = \alpha_0 |00\rangle + \alpha_1 |01\rangle + \alpha_2 |10\rangle + \alpha_3 |11\rangle$$

Note that this is the most general case, since q_0 and q_1 might be entangled so it might not be possible to treat them separately. Thus, in the first case we have that after measuring q_0 the state becomes

$$|\psi\rangle \begin{cases} |0\rangle \otimes (\beta_{00} |0\rangle + \beta_{01} |1\rangle) & @ \ |\alpha_0|^2 + |\alpha_1|^2 \\ |1\rangle \otimes (\beta_{10} |0\rangle + \beta_{11} |1\rangle) & @ \ |\alpha_2|^2 + |\alpha_3|^3 \end{cases}$$

for some probability amplitudes β_{ij}. Then, with the application of U conditioned of the outcome of the measurement the state becomes

$$|\psi\rangle \begin{cases} |0\rangle \otimes (\beta_{00}|0\rangle + \beta_{01}|1\rangle) & @ \ |\alpha_0|^2 + |\alpha_1|^2 \\ |1\rangle \otimes (\beta_{10}U|0\rangle + \beta_{11}U|1\rangle) & @ \ |\alpha_2|^2 + |\alpha_3|^3 \end{cases}$$

It's easy to see that this very outcome of probabilities could have been achieved by first applying the controlled U gate to $|\psi\rangle$ at the beginning, getting

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|1\rangle \otimes U|0\rangle + \alpha_3|1\rangle \otimes U|1\rangle$$

and then measuring q_0 afterwards.

2

Mathematical foundations

Now that we have introduced some preliminary concepts in quantum mechanics and quantum computation, we can turn to the **mathematical foundations** that will allow us to develop a deeper understanding of the tools ahead. By the end of this chapter, we will be ready to state the postulates of quantum mechanics in a precise form. To prepare for that, we must first lay out several essential definitions and structures.

We will start our mathematical discussion with the definition of **scalar product** — we will assume the definitions of vector space, basis and linear independence are already known by the reader.

Definition 2.1: Scalar product

Given a scalar product vector space V, a **scalar product** $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{C} : (v, w) \mapsto \langle v, w \rangle$ is a linear function that satisfies the following properties:

- $\forall u, v, w \in V, \alpha, \beta \in \mathbb{C} \quad \langle u, \alpha v + \beta w \rangle = \alpha \langle u | v \rangle + \beta \langle u | w \rangle$
- $\forall u, v \in V \quad \overline{\langle u, v \rangle} = \langle v, u \rangle$ — where $\overline{z}$ is the conjugate of $z \in \mathbb{C}$
- $\forall u \in U \quad \langle u, u \rangle \geq 0$ and $\langle u, u \rangle = 0$ if and only if $u = 0$

Scalar products are also called *inner product*, and are used to define many other tools on top of the vector space considered.

Proposition 2.1

For any scalar product vector space V, any scalar product satisfies the following property:

$$\forall u, v, w \in V, \alpha, \beta \in \mathbb{C} \quad \langle \alpha u + \beta v, w \rangle = \overline{\alpha} \langle u, w \rangle + \overline{\beta} \langle v, w \rangle$$

Proof. TODO □ TODO

In particular, the scalar product that we are going to use for our purposes is defined as follows:

$$\forall u, v \in \mathbb{C}^n \quad \langle u, v \rangle = \sum_{i=1}^{n} \overline{u_i} v_i$$

In fact, we can prove that this is indeed a scalar product as follows:

- TODO TODO
- TODO TODO
- TODO TODO

From now on, when we refer to a "scalar product" we will refer to this particular definition.

We are finally ready to explain the "braket" that we used from the beginning of the previous chapter. This notation was invented by the Nobel Prize in Physics Paul Dirac, and it works as follows: first, observe that our scalar product can be rewritten as follows

$$\langle u, v \rangle = \begin{pmatrix} \overline{u_1} \cdots \overline{u_n} \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

To be precise, this product would yield a 1×1 matrix, which can be interpreted as a scalar. Through Dirac notation, we will write

$$\langle u, v \rangle = \langle u | v \rangle$$

where $\langle u|$ is called **bra**, and $|v\rangle$ is called **ket** (as in "bra-ket"). In other words, we have that $|v\rangle$ is just a regular column vector $v \in V$

$$|v\rangle = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

defined over some scalar product vector space V, while $\langle u|$ is a *linear map* that acts as follows:

$$\langle \cdot | : V \to \overline{V} : \begin{pmatrix} u_1 \cdots u_n \end{pmatrix} \mapsto \begin{pmatrix} \overline{u_1} \cdots \overline{u_n} \end{pmatrix}$$

Theorem 2.1: Cauchy-Schwarz inequality

Given a scalar product vector space V, it holds that

$$\forall u, v \in V \quad |\langle u|v\rangle| \leq \sqrt{\langle u|v\rangle \langle u|v\rangle}$$

where the equality holds if and only if u and v are linearly independent.

Moreover, our scalar product induces a **norm**, which is defined as follows.

Definition 2.2: Norm

Given a scalar product vector space V, the **norm** of a vector $v \in V$ is defined as follows

$$||v|| = \sqrt{\langle v|v\rangle}$$

As usual, two vectors $u, v \in V$ are said to be **orthogonal** if $\langle u|v\rangle = 0$. This allows us to define orthonormal bases.

Definition 2.3: Orthonormal basis

Given a scalar product vector space V, a basis $\{e_1, \ldots, e_n\}$ is said to be **orthonormal** if

$$\forall i, j \in [n] \quad \langle e_i|e_j\rangle = \delta_{ij}$$

where $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$ is called **Kronecker delta**.

Let's see the Dirac notation in action. Consider an orthonormal basis $\{e_1, \ldots, e_n\}$ for some scalar product vector space V; by definition, we know that we can write any vector $u \in V$ as follows

$$u = \sum_{i=1}^{n} \alpha_i e_i$$

for some coefficients $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$. Now, we observe that for all $i \in [n]$

$$\begin{aligned} \langle e_i|u\rangle &= \langle e_i| \sum_{j=1}^{n} \alpha_j e_j\rangle \\ &= \langle e_i|\alpha_1 e_1 + \ldots + \alpha_n e_n\rangle \\ &= \alpha_1 \langle e_i|e_1\rangle + \ldots + \alpha_n \langle e_i|e_n\rangle \\ &= \sum_{j=1}^{n} \alpha_j \langle e_i|e_j\rangle \\ &= \sum_{j=1}^{n} \alpha_j \delta_{ij} \\ &= \alpha_i \end{aligned}$$

Indeed, with the scalar product we can compute the projection of u onto the i-th vector of the basis. Hence, we can rewrite the first equation as follows:

$$|u\rangle = \sum_{i=1}^{n} \alpha_i |e_i\rangle = \sum_{i=1}^{n} \langle e_i|u\rangle |e_i\rangle$$

In particular, we observe that

$$|u\rangle = \sum_{i=1}^{n} |e_i\rangle \langle e_i|u\rangle \implies I = \sum_{i=1}^{n} |e_i\rangle \langle e_i|$$

which is a famous identity in quantum mechanics called **resolution of the identity**. In particular, this identity directly implies the following useful property. As a final note, by the properties of scalar products we also have that

$$\langle v|u\rangle = \left\langle v \middle| \sum_{i=1}^{n} \langle e_i|u\rangle \, |e_i\rangle \right\rangle = \sum_{i=1}^{n} \langle v|e_i\rangle \, \langle e_i|u\rangle$$

Proposition 2.2

Given a scalar product vector space V, it holds that

$$\forall u, v, w \in V \quad \langle u|v\rangle \, |w\rangle = (|w\rangle \, \langle u|) \, |v\rangle$$

Proof. TODO □ TODO

2.1 Hilbert spaces

Now that we covered Dirac notation, we can describe what are **Hilbert spaces** — we will see why we care about this particular type of vector spaces later in the chapter. First, consider the following definitions.

Definition 2.4: Weak convergence

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, we say that the sequence **converges weakly** to a vector $v \in V$ if

$$\forall w \in V \quad \lim_{m\to+\infty} \langle v_m|w\rangle = \langle v|w\rangle$$

In other words, this type ocf convergence requires all projections of v_m along any fixed direction w to approach the projection of v. Differently, the next type of convergence is more strict.

Definition 2.5: Strong convergence

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, we say that the sequence **converges strongly** to a vector $v \in V$ if

$$\lim_{m\to+\infty} ||v - v_m|| = 0$$

In fact, this type of convergence requires the actual vectors of the sequence to get close *in norm* to v. We observe the following proposition.

Proposition 2.3

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, if the sequence converges strongly to some vector $v \in V$, it holds that

- the sequence also converges weakly
- the scalar products defined over V are continuous, i.e.

$$\forall u, v \in V \quad \langle u|v\rangle = \lim_{m\to+\infty} \langle u|v_m\rangle$$

Definition 2.6: Cauchy sequence

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, we say that the sequence is a **Cauchy sequence** if it holds that

$$\forall \varepsilon > 0 \quad \exists n_\varepsilon \in \mathbb{N} \quad \forall n, m > n_\varepsilon \quad ||v_n - v_m|| < \varepsilon$$

For example, let's consider the space $\mathbb{R}^2$ equipped with the Euclidean norm

$$||v|| = \sqrt{x^2 + y^2}$$

Then, if we consider the following vector sequence

$$\left\{ \begin{pmatrix} \frac{1}{m} \\ \vdots \\ \frac{1}{m} \end{pmatrix} \right\}_{m\in\mathbb{N}}$$

we see that for any distinct m, n it holds that

$$||v_m - v_n|| = \sqrt{\left(\frac{1}{m} - \frac{1}{n}\right)^2 + \left(\frac{1}{m} - \frac{1}{n}\right)^2} = \sqrt{2}\left|\frac{1}{m} - \frac{1}{n}\right|$$

Therefore, for any $\varepsilon > 0$ it suffices to take any $N > \dfrac{2\sqrt{2}}{\varepsilon}$ such that

$$\forall m, n > N \quad ||v_m - v_n|| < \varepsilon$$

We are finally ready to define Hilbert spaces.

Definition 2.7: Hilbert space

A **Hilbert space** is a *complete* scalar product vector space, i.e. it is a vector space

- equipped with a scalar product
- such that every Cauchy sequence converges strongly to an element in the space.

For example, the space $\mathbb{R}^n$ is a Hilbert space. Indeed, since every finite vector space of size n is isomporphic to $\mathbb{R}^n$, we can immediately derive the following proposition.

Proposition 2.4

Finite-dimensional vector spaces are always complete.

2.1.1 Linear operators

Given a Hilbert space $\mathcal{H}$, we can define **operators** — which are nothing but linear maps.

Definition 2.8: Adjoint operator

Given a Hilbert space $\mathcal{H}$, and an operator A, the **adjoint** operator of A, denoted with $A^\dagger$, is a linear map that satisfies the following property

$$\forall u, v \in \mathcal{H} \quad \langle u|A^\dagger v\rangle = \langle Au|v\rangle$$

We say that an operator A is **self-adjoint**, or *Hermitian*, if and only if $A = A^\dagger$.

For instance, the following matrix $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ is a linear operator whose adjoint is $S^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$. In fact, we have that

$$\langle u|S^\dagger v\rangle = \begin{pmatrix} \overline{u_1} & \overline{u_2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \overline{u_1} & \overline{u_2} \end{pmatrix} \begin{pmatrix} v_1 \\ -iv_2 \end{pmatrix} = \overline{u_1}v_1 - i\overline{u_2}v_2$$

and since

$$Su = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ iu_2 \end{pmatrix} \implies \langle Su| = \begin{pmatrix} \overline{u_1} & \overline{iu_2} \end{pmatrix}$$

but because $\overline{iu_2} = \bar{i} \cdot \overline{u_2} = -i\overline{u_2}$ this implies that

$$\langle Su|v\rangle = \begin{pmatrix} \overline{u_1} & -i\overline{u_2} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \overline{u_1}v_2 - i\overline{u_2}v_2$$

Proposition 2.5

For any adjoint operators A, B defined over some Hilbert space $\mathcal{H}$, it holds that

1. $(AB)^\dagger = B^\dagger A^\dagger$
2. for any scalar z it holds that $(zA)^\dagger = \overline{z}A^\dagger$
3. $(A^\dagger)^\dagger = A$
4. $(A + B)^\dagger = A^\dagger + B^\dagger$

How do we evaluate the adjoint of a given operator?

Proposition 2.6

Given an operator A defined over a scalar product vector space, it holds that

$$a^\dagger_{ij} = \overline{a_{ji}}$$

This property is incredibly useful, because it implies that the adjoint operator of A is its transposed conjugate matrix. Most notably, due to the way we defined our scalar product, it holds that for any column vector $|x\rangle$ we have that

$$\langle x| = |x\rangle^\dagger$$

which gives an intuition of the reason why we defined our scalar product as such.

Proposition 2.7

If an operator A is self-adjoint, it holds that

$$\langle u|Av\rangle = \langle Au|v\rangle = \overline{\langle v|Au\rangle}$$

Proof. TODO □ TODO

Definition 2.9: Positive operator

An operator A defined over a scalar product vector space $\mathcal{H}$ is said to be **positive** if it holds that

$$\forall v \in \mathcal{H} \quad \langle v|Av\rangle \geq 0$$

Proposition 2.8

If an operator A is positive, then it is Hermitian.

Proof. TODO □ pag 8

At the beginning of the previous chapter we said that all quantum gates are *unitary transformations*, but we did now provide a definition of unitary. Now we are ready to introduce it, and start to grasp why we are discussing Hilbert spaces.

Definition 2.10: Unitary operators

Given a Hilbert space $\mathcal{H}$, and an operator U, we say that U is **unitary** if it holds that

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

In other words, U is unitary if and only if its adjoint operator is also its inverse. An interesting characterization of unitary transformations is the following property.

Proposition 2.9: Unitary operators (alt. def.)

An operator U defined over a Hilbert space $\mathcal{H}$ is unitary if and only if

- U is surjective
- $\forall x, y \in \mathcal{H} \quad \langle Ux|Uy\rangle = \langle x|y\rangle$ or equivalently, if it holds that

$$\forall x \in \mathcal{H} \quad ||Ux|| = ||x||$$

Proof. TODO □ TODO

In particular, we observe that the second property of this proposition is very interesting: the *preservation of the scalar product*, i.e. the property for which

$$\forall x, y \in \mathcal{H} \quad \langle Ux|Uy\rangle = \langle x|y\rangle$$

means that the operator U does not change the geometric relationships between vectors — i.e. their lengths and angles remain the same.

Proposition 2.10

If U is a unitary operator, then the rows and the columns of U are orthonormal.

Proof. First, we prove that the rows and columns of U are normalized.

Claim 1: The norm of the rows and the columns of U is 1.

Proof of the Claim. By the previous proposition, we know that

$$\forall x \in \mathcal{H} \quad ||Ux|| = ||x||$$

and in particular, this must hold for the canonical basis as well. This means that

$$||Ue_i|| = ||e_i|| = 1$$

for any vector of the canonical basis e_i. Then, since Ue_i is just the i-th column of U, this proves that each column of U has norm1.

To prove the same result for the rows, observe that by Proposition 2.6 it holds that

$$U^\dagger = \overline{U}^T$$

and □ da finire boh come??

Next, we show orthogonality of rows and columns.

Claim 2: The rows and columns of U are orthogonal.

Proof of the Claim. By the preservation of the scalar product, we know that

$$\forall x, y \in \mathcal{H} \quad \langle Ux|Uy\rangle = \langle x|y\rangle$$

and in particular, this must also hold for the canonical basis. This means that for any pair of vectors of the canonical basis we have that

$$\forall i, j \quad \langle Ue_i|Ue_j\rangle = \langle e_i|e_j\rangle = \delta ij$$

which means that the rows and columnds of U must be orthogonal. □ does it?

The two claims together conclude the proof. □

Proposition 2.11

If A and B are two unitary operators, then AB is a unitary operator.

Proof. Since A and B are unitary it holds that

$$A^\dagger A = AA^\dagger = B^\dagger B = BB^\dagger = I$$

Now, by Proposition 2.5 we have that

$$(AB)^\dagger = B^\dagger A^\dagger$$

from which we conclude that

$$(AB)^\dagger(AB) = B^\dagger A^\dagger AB = B^\dagger IB = B^\dagger B = I$$

□

Definition 2.11: Normal operators

Given a Hilbert space $\mathcal{H}$, and an operator A, we say that A is **normal** if it satisfies the following property

$$A^\dagger A = AA^\dagger$$

Clearly, from their definition we immediately see that both self-adjoint and unitary operators are both normal.

Lastly, say that we know that how an operator U acts on some input $|x\rangle$, and we want to derive U. How can we do this?

Proposition 2.12

For any operator U in a Hilbert space $\mathcal{H}$, if $\mathcal{B}$ is a base of $\mathcal{H}$ it holds that

$$U = \sum_{b\in\mathcal{B}} U\,|b\rangle\,\langle b|$$

Proof. The formula derives directly from the resolution of the identity, and the linearity of operators of Hilbert spaces

$$\begin{aligned} U &= U \cdot I \\ &= U \cdot \sum_{b \in \mathcal{B}} |b\rangle \langle b| \\ &= \sum_{b \in \mathcal{B}} U |b\rangle \langle b| \end{aligned}$$

□

Therefore, if we know how U acts component-wise, we can reconstruct the operator acting on the whole space as such.

To conclude this section, we introduce a definition that will be very useful in the next chapter.

Definition 2.12: Trace

Given a matrix $A \in \mathbb{C}^{n \times n}$, its **trace** is defined as follows:

$$\operatorname{tr}(A) := \sum_{i=1}^{n} a_{ii}$$

In other words, the trace of a matrix is the sum of the elements on its diagonal.

Proposition 2.13

Given two matrices $A, B \in \mathbb{C}^{n \times n}$, it holds that

1. $\operatorname{tr}(AB) = \operatorname{tr}(BA)$, meaning that the trace is *cyclic*
2. $\operatorname{tr}(A + B) = \operatorname{tr}(A) + \operatorname{tr}(B)$, meaning that the trace is *linear*
3. $\forall z \in \mathbb{C} \quad \operatorname{tr}(zA) = z \operatorname{tr}(A)$
4. $\operatorname{tr}(A) = \operatorname{tr}(UAU^\dagger)$ if U is unitary, meaning that the trace is *invariant under unitary similarity transformation*

In particular, we observe that the last property follow trivially from the cyclic property:

$$\operatorname{tr}(UAU^\dagger) = \operatorname{tr}(UU^\dagger A) = \operatorname{tr}(IA) = \operatorname{tr}(A)$$

But perhaps most importantly, the following property holds.

Proposition 2.14

Given a matrix $A \in \mathbb{C}^{n\times n}$, and a vector $|v\rangle$, it holds that

$$\operatorname{tr}(A\,|v\rangle\,\langle v|) = \langle v|Av\rangle$$

Proof. First, we observe that the definition of the trace can be rewritten in terms of the Dirac notation

$$\operatorname{tr}(X) = \sum_{i=1}^{n} x_{ii} = \sum_{i=1}^{n} \langle e_i|X|e_i\rangle$$

From this equality, we conclude that

$$\begin{aligned}
\operatorname{tr}(A\,|v\rangle\,\langle v|) &= \sum_{i=1}^{n} \langle e_i|A|v\rangle\,\langle v|e_i\rangle \\
&= \sum_{i=1}^{n} \langle \psi|e_i\rangle\,\langle e_i|A|v\rangle \\
&= \langle v|\left(\sum_{i=1}^{n} |e_i\rangle\,\langle e_i|\right) A\,|v\rangle \\
&= \langle v|A|v\rangle \\
&= \langle v|Av\rangle
\end{aligned}$$

□

We will discuss the importance of this equality in later sections.

TODO

show that the trace is indep of the chosen basis

2.2 Spectral theory

Since unitary operators are linear maps, we are interested in their eigenvectors and eigenvalues — which are defined as usual. First, let us recall some preliminary definitions.

Definition 2.13: Non-degenerate eigenvalue

Given a matrix A, and an eigenvalue λ of A, we say that λ is **non-degenerate** if the associated eigenspace has dimension 1 (or equivalently, if it has only 1 associated eigenvector).

Definition 2.14: d-fold degenerate eigenvalue

Given a matrix A, and an eigenvalue λ of A, we say that λ is d-fold degenerate if there are d linearly independent eigenvectors $u_1, \ldots, u_d$ associated to λ.

In Dirac notation, if λ is non-degenerate, we refer to the only eigenvector associated to λ as $|\lambda\rangle$, indeed it holds that

$$A|\lambda\rangle = \lambda|\lambda\rangle$$

Proposition 2.15

Given a matrix A defined over a Hilbert space $\mathcal{H}$, and an eigenvalue λ associated to A, it holds that

$$\langle\lambda| A^\dagger = \overline{\lambda}\langle\lambda|$$

Proof. TODO □ TODO

The following theorem provides a characterization of the eigenvalues and eigenvectors of self-adjoint and unitary operators, which have surprisingly nice properties.

Theorem 2.2: Spectral theorem

The following propositions hold:

- The eigenvalues of a self-adjoint operator are real values.
- The eigenvalues of a unitary operator are complex values of modulus 1.
- Eigenvectors of self-adjoint and unitary operators associated to different eigenvalues are orthogonal to each other.

Moreover, for finite-dimensional Hilbert spaces the following holds.

Theorem 2.3: Spectral theorem for fin. Hilbert spaces

Given a finite-dimensional Hilbert space $\mathcal{H}$, and a normal operator A defined over $\mathcal{H}$, the set of all eigenvectors of A can be expanded to form an orthonormal basis for $\mathcal{H}$.

In particular, to basis expansion can be achieve through the **Gram-Shmidt** in the each degenerate eigenspace of A in order to produce an orthonormal set.

2.3 Projectors

Next, we are going to discuss **projectors**, which are another very crucial pieces of quantum computation. We saw how scalar products are able to perform projection over desired direction, in fact we will use the Dirac notation to define precise operators for our purposes. But as always, first some preliminary definitions.

Definition 2.15: Orthogonal space

Given a scalar product vector space U, and two linear subspaces $V, W \subset U$, we say that V is orthogonal to W if

$$\forall v \in V, w \in W \quad \langle v|w\rangle = 0$$

Given a scalar product vector space U, and a linear subspace $V \subset U$, the **orthogonal complement** of V is defined as follows:

$$V^{\perp} := \{u \in U \mid \forall v \in v \quad \langle u|v\rangle = 0\}$$

In particular, we observe that if U is finite-dimentional it holds that $V = U - V^{\perp}$ and that $(V^{\perp})^{\perp} = V$.

Definition 2.16: Topologically closed subspace

Given a Hilbert space $\mathcal{H}$, and a linear subspace $V \subset \mathcal{H}$, we say that V is **topologically closed** if any sequence of vectors defined over V converges in V.

Interestingly enough, given a topologically closed subspace $V \subset \mathcal{H}$ of some Hilbert space, we can write any vector $u \in V$ as the sum of two orthogonal vectors of V and $V^{\perp}$, as follows. Let $\{f_1, \ldots, f_n\}$ be an orthonormal basis of V, and define the following vectors

$$\forall u \in U \quad u_V := \sum_{i=1}^{n} \langle f_i|v\rangle \, f_i$$

Then, if we call

$$u_{V^{\perp}} := u - u_V$$

, we see that TODO which indeed proves that u_V and $u_{V^{\perp}}$ are orthogonal to each other.

decommenta e finisci la formula

With this observation, we can finally define the projector operators.

Definition 2.17: Projector

Given a Hilbert space $\mathcal{H}$, and a closed subspace $V \subset \mathcal{H}$, the **projector** operator that projects any given vector $v \in \mathcal{H}$ onto V is defined as follows:

$$P_V : \mathcal{H} \to V : u \mapsto u_V = \sum_{i=1}^{n} \langle f_i|u\rangle \, f_i$$

where $\{f_1, \ldots, f_n\}$ is an orthonormal basis of V.

Most importantly, given the map in the definition we have that the projector operator P_V is defined as follows

$$P_V := \sum_{i=1}^{n} |f_i\rangle \langle f_i|$$

Clearly, by definition of $u_{V^\perp}$ it holds that

$$P_{V^\perp} : \mathcal{H} \to V^\perp : u \mapsto u_{V^\perp} := u - u_V$$

Moreover, since P_V performs a projection, we have that

$$\forall u \in \mathcal{H} \quad u \in V \iff P_V u = u$$

and that

$$\forall u \in \mathcal{H} \quad u \in V^\perp \iff P_V u = 0$$

Proposition 2.16

Given a Hilbert space $\mathcal{H}$, and a closed subspace $V \subset \mathcal{H}$, the projector P_V is such that

1. $P_V^2 = P_V$, i.e. it is *idempotent*
2. $P_V^\dagger = P_V$, i.e it is Hermitian

Proof. TODO

da fare idempotency

For the second property, we observe that for any vector $|v\rangle$ it holds that

$$(|v\rangle \langle v|)^\dagger = |v\rangle \langle v|$$

that together with Proposition 2.5 it implies that

$$\begin{aligned} P_V^\dagger &= \left(\sum_{i=1}^{n} |f_i\rangle \langle f_i| \right)^\dagger \\ &= \sum_{i=1}^{n} (|f_i\rangle \langle f_i|)^\dagger \\ &= \sum_{i=1}^{n} |f_i\rangle \langle f_i| \\ &= P_V \end{aligned}$$

□

Proposition 2.17

Any projector operator only has 0 and 1 as possible eigenvalues.

Proof. Consider a projector operator P_V; by definition v is an eigenvalue of P_V associated to the eigenvalue λ if it holds that

$$P_V v = \lambda v$$

Hence, we observe that

$$P_V^2 v = P_V(P_V v) = P_V(\lambda v) = \lambda P_V v = \lambda(\lambda v) = \lambda^2 v$$

and by idempotency of P_V it holds that

$$\lambda^2 v = P_V^2 = P_V v = \lambda v$$

which implies that

$$\lambda^2 v - \lambda v = 0 \iff (\lambda^2 - \lambda)v = 0$$

Finally, since v is an eigenvector it holds that $v \neq 0$, therefore it must be that the last equation is true only if

$$\lambda^2 - \lambda = 0 \iff \lambda = 0 \vee \lambda = 1$$

□

Given a Hilbert space $\mathcal{H}$, and two topologically closed subspaces $V, W \subset \mathcal{H}$, we say that P_V and P_W are **orthogonal** if it holds that $V \perp W$. Now, fix a vector $u \in \mathcal{H}$; by definition $P_V u$ is a vector that lies inside V, therefore it holds that

$$P_W(P_V u) = 0$$

since $V \perp W$, and by the same reasoning applied on W first we conclude that

$$P_W P_V = P_V P_W = \mathbf{0}$$

where $\mathbf{0}$ is the **zero operator** — indeed, it is a zero matrix.

As a final note, if A is an linear operator, and λ is an eigenvalue of A, we denote with P_λ the projector that projects vectors onto the eigenspace associated to λ.

Proposition 2.18

If P and Q are two orthogonal projectors, then $P + Q$ is still a projector.

Proof. TODO □ TODO

The following is the last property of projectors that we will present, and it will be very useful when we will introduce the postulates of quantum mechanics in the next section.

Theorem 2.4: Spectral decomposition

Given a Hilbert space $\mathcal{H}$, and a normal operator A defined on $\mathcal{H}$ whose eigenvalues are $\lambda_1, \dots, \lambda_n$, the matrix A can be rewritten as follows:

$$A = \sum_{i=1}^{n} \lambda_i P_{\lambda_i}$$

The theorem above is sometimes written in the following different form

$$A = \sum_{i=1}^{n} \lambda_i \left|\lambda_i\right\rangle \left\langle\lambda_i\right|$$

which *seems* to imply that $P_{\lambda_i} = |\lambda_i\rangle \langle\lambda_i|$, i.e. the projector onto the i-th eigenspace of M is $|\lambda_i\rangle \langle\lambda_i|$. This is a *slight* notation abuse, and an example will make it clear. Consider the following matrix:

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

It is easy to see that the eigenvalues of A are only 2 and 5, however 2 is 2-fold degenerate! Indeed, by solving

$$M |v\rangle = 2 |v\rangle$$

we get an eigenspace of dimension 2, and we can choose $\{|e_1\rangle, |e_2\rangle\}$ as orthonormal eigenbasis; similarly, we find that $\{|e_3\rangle\}$ is an eigenbasis of the second eigenspace of dimension 1. In the end, this means that

$$\begin{aligned} M &= 2 |1\rangle \langle 1| + 2 |2\rangle \langle 2| + 5 |3\rangle \langle 3| \\ &= 2(|1\rangle \langle 1| + |2\rangle \langle 2|) + 5 |3\rangle \langle 3| \\ &= 2P_2 + 5P_5 \end{aligned}$$

which clearly shows that, in general, $P_{\lambda_i} \neq |\lambda_i\rangle \langle\lambda_i|$! We observe that this also matches the definition of projector provided at the beginning of the section

$$P_V := \sum_{i=1}^{n} |f_i\rangle \langle f_i|$$

for some orthonormal basis $\{f_1, \ldots, f_n\}$ of V. In general, the notation

$$A = \sum_{i=1}^{n} \lambda_i |\lambda_i\rangle \langle\lambda_i|$$

implies that if λ is d-fold degenerate, then λ is repeated d times inside the sum.

2.4 Tensor product

2.4.1 General definition

The last operation that we need to introduce into our mathematical background is the **tensor product**. There are multiple ways to define the tensor product.

Definition 2.18: Tensor product

Given two vector spaces V and W defined over the same field, the **tensor product** $V \otimes W$ is a vector space together with a bilinear map

$$\otimes : V \times W \to V \otimes W$$

such that for every vector space X, and a bilinear map $f : V \times W \to X$ there exists a unique linear map $\tilde{f} : V \otimes W \to X$ with

$$f = \tilde{f} \circ \otimes$$

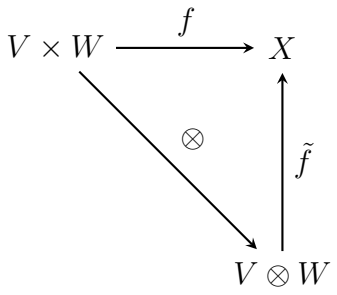


Figure 2.1: The diagram of the definition above.

This definition is quite rigorous and dense, but a concrete example will make it clear. First, we recall that a function f is **linear** if it holds that

- $f(x+y) = f(x) + f(y)$
- $f(\alpha x) = \alpha f(x)$

Then, for a function to be **bilinear** (or *multilinear*, in general) we require f to be linear in both of its arguments:

- $f(x+x', y) = f(x, y) + f(x', y)$
- $f(x, y+y') = f(x, y) + f(x, y')$
- $f(\alpha x, y) = \alpha f(x, y) = f(x, \alpha y)$

From this definition, we immediately point out that the tensor product $\otimes$ satisfies

- $(v + v') \otimes w = v \otimes w + v' \otimes w$
- $v \otimes (w + w') = v \otimes w + v \otimes w'$
- $(\alpha v) \otimes w = v \otimes (\alpha w)$

We may call the first two as *distributive properties*. Let's consider an example we already presented in the first chapter of the notes. At the start of our discussion we "defined" the tensor product between two vectors $\begin{pmatrix} a \\ b \end{pmatrix}$ and $\begin{pmatrix} c \\ d \end{pmatrix}$ as follows

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} := \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

Now that we know what true definition is, we can apply it to understand why this is the case. First, we can assume that $\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix} \in \mathbb{R}^2$ therefore $V = W = \mathbb{R}^2$. Therefore, our tensor product will have the signature

$$\otimes : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2 \otimes \mathbb{R}^2$$

Now, we observe that the output space $V \otimes W$ is a vector space that has

$$\mathcal{B}_{V \otimes W} := \{f_i \otimes g_j \mid f_i \in \mathcal{B}_V, g_j \in \mathcal{B}_W\}$$

as a basis. In particular, this means that

$$\mathcal{B}_{\mathbb{R}^2 \otimes \mathbb{R}^2} = \{e_1 \otimes e_1, e_1 \otimes e_2, e_2 \otimes e_1, e_2 \otimes e_2\}$$

is a basis for $\mathbb{R}^2 \otimes \mathbb{R}^2$. In fact, it is not diffcult to prove that $\mathbb{R}^4 \cong \mathbb{R}^2 \otimes \mathbb{R}^2$ by mapping the canonical basis of $\mathbb{R}^4$ to $\mathcal{B}_{\mathbb{R}^2 \otimes \mathbb{R}^2}$. Let φ be the isorphism between the two. Then, by bilinearity of $\otimes$ we have that

$$\begin{aligned}
\varphi\left(\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix}\right) &= \varphi\left((ae_1 + be_2) \otimes (ce_1 + de_2)\right) \\
&= \varphi\left(ac(e_1 \otimes e_1) + ad(e_1 \otimes e_2) + bc(e_2 \otimes e_1) + bd(e_2 \otimes e_2)\right) \\
&= ac \cdot \varphi(e_1 \otimes e_1) + ad \cdot \varphi(e_1 \otimes e_2) + bc \cdot \varphi(e_2 \otimes e_1) + bd \cdot \varphi(e_2 \otimes e_2) \\
&= ac \cdot e_1 + ad \cdot e_2 + bc \cdot e_3 + bd \cdot e_4 \\
&= \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}
\end{aligned}$$

Thus, in our example we have that $\tilde{f} = \varphi$ itself and $X = \mathbb{R}^4$, indeed $f = \varphi \circ \otimes$ is a function that maps pairs of two-dimensional vectors in $\mathbb{R}^2 \times \mathbb{R}^2$ into 4-dimensional vectors inside $\mathbb{R}^4$. This example also shows that the definition of tensor product does not depend on the choice of the vector space X, and for our purposes it is practical to choose a vector space that is isomporphic to $V \otimes W$.

What is the purpose of the tensor product in the first place? Given a vector space V, let

$$\text{Functions}(V) := \{f : V \to K\}$$

be the set of functions from V to some other field K. It can be proven that this set forms a **vector space**. Then, the **tensor product** is the mathematical object that enables the following

$$\text{Functions}(V \times W) = \text{Functions}(V) \otimes \text{Functions}(W)$$

This property is very useful as it allows to reason about functions on $V \times W$ not as a completely new object, but by studying it in terms of functions on V and W individually. In general, it allows to deal with objects that are "more managable". For instance, in our previous example the pair $\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix}\right) \in \mathbb{R}^2 \times \mathbb{R}^2$ lies on a *grid*, while an object $\begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} \in \mathbb{R}^4$ is just a 4-dimensional vector.

Lastly, from the definition we observe that

$$\dim(V \otimes W) = \dim V \cdot \dim W$$

indeed V and W can have different dimensions. For instance, a pair of vectors $(v, w) \in \mathbb{R}^2 \times \mathbb{R}^3$ lies in a 5-dimensional space, since

$$\dim(V \times W) = \dim V + \dim W$$

however the tensor product generates a bigger space of size

$$\dim(\mathbb{R}^2 \otimes \mathbb{R}^3) = \dim \mathbb{R}^2 \cdot \dim \mathbb{R}^3 = 2 \cdot 3 = 6$$

Indeed, we have that

$$\mathbb{R}^2 \times \mathbb{R}^3 \cong \mathbb{R}^5 \qquad \mathbb{R}^2 \otimes \mathbb{R}^3 \cong \mathbb{R}^6$$

Consider any two vectors $v \in V$ and $w \in W$, and suppose that $n := \dim V$ and $m := \dim W$. Then, we can describe them in terms of $\mathcal{B}_V$ and $\mathcal{B}_W$, respectively, as follows:

$$v = \sum_{i=1}^{n} \alpha_i f_i \qquad w = \sum_{j=1}^{m} \beta_j g_j$$

Then, by the distributive properties in the definition we have that

$$\begin{aligned} v \otimes w &= \left(\sum_{i=1}^{n} \alpha_i f_i \right) \otimes \left(\sum_{j=1}^{m} \beta_j g_j \right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j (f_i \otimes g_j) \end{aligned}$$

which indeed is a vector that lies inside $V \otimes W$ since $(f_i \otimes g_j) \in \mathcal{B}_{V \otimes W}$. Then, if the basis of choice of the output set X is the canonical basis, we observe that

$$\begin{aligned} \varphi(v \otimes w) &= \varphi \left(\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j (f_i \otimes g_j) \right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j \varphi(f_i \otimes g_j) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j e_{(i-1)m+j} \\ &= \begin{pmatrix} \alpha_1 \beta_1 \\ \alpha_1 \beta_2 \\ \vdots \\ \alpha_1 \beta_m \\ \alpha_2 \beta_1 \\ \vdots \\ \alpha_n \beta_m \end{pmatrix} \in \mathbb{R}^{n \cdot m} \end{aligned}$$

where the last column vector contains *all* the pairs (α_i, β_j) — note that $\mathbb{R}^{n \cdot m} \neq \mathbb{R}^{n \times m}$, as the latter denotes the space of real-valued matrices of dimension $n \times m$. Again, we underline that we arbitrarily chose the space $\mathbb{R}^{n \cdot m}$, but we could have chosen *any* space of dimension $n \cdot m$ as output — even $\mathbb{R}^{n \times m}$! A concrete example will clarify: consider two spaces $V = \mathbb{R}^2$ and $W = \mathbb{R}^{2\times 2}$. Then, any element $v \in \mathbb{R}^2$ has the form $v = \begin{pmatrix} a \\ b \end{pmatrix}$ while any matrix of $A \in \mathbb{R}^{2\times 2}$ looks like $A = \begin{pmatrix} c & d \\ e & f \end{pmatrix}$. If we want to compute $v \otimes A$, we can project it onto any space that has dimension

$$\dim(\mathbb{R}^2 \otimes \mathbb{R}^{2\times 2}) = \dim \mathbb{R}^2 \cdot \dim \mathbb{R}^{2\times 2} = 2 \cdot 4 = 8$$

Here are some examples of the possible outputs of $v \otimes A$ depending on the choice of the output space:

$$\begin{pmatrix} ac \\ ad \\ ae \\ af \\ bc \\ bd \\ be \\ bf \end{pmatrix} \in \mathbb{R}^6 \qquad \begin{pmatrix} ac & ad & ae & af \\ bc & bd & be & bf \end{pmatrix} \in \mathbb{R}^{2\times 4} \qquad \begin{pmatrix} ac & bc \\ ad & bd \\ ae & be \\ af & bf \end{pmatrix} \in \mathbb{R}^{4\times 2}$$

2.4.2 Kronecker product

To conclude this chapter, let's look at the tensor product from the point of view of quantum computing. Until now, we presented examples regarding real-valued vectors, however, when talking about Hibert spaces in quantum computing we often need *complex* values, as we already know. Consider a qubit

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

From this very formulation, we see that we are representing $|\psi\rangle$ under the **computational basis** $\{|0\rangle, |1\rangle\}$ we already encountered. However, since $\alpha, \beta \in \mathbb{C}$ this implies that $|\psi\rangle \in \mathbb{C}^2$. Now, consider a unitary operator acting on one qubit, for example the Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Our next goal is to compute $H \otimes H$. Clearly $H \in \mathbb{C}^{2\times 2}$, therefore $H \otimes H \in \mathbb{C}^{2\times 2} \otimes \mathbb{C}^{2\times 2}$ which yields a space isomporphic to $\mathbb{C}^{16}$. However, since H is a matrix it is not convenient to express the result inside $\mathbb{C}^{16}$, so we usually represent it inside $\mathbb{C}^{4\times 4}$, which is still isomporphic to $\mathbb{C}^{16}$. After some calculations, we can show that

$$H \otimes H = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

However, we notice an interesting pattern in this result: this 4×4 matrix is essentially split into 4 sections that are "multiples" of H

$$H \otimes H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \cdot H & 1 \cdot H \\ 1 \cdot H & -1 \cdot H \end{pmatrix}$$

In fact, for any two matrices

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$$

it holds that

$$A \otimes B = \begin{pmatrix} aB & bB \\ cB & dB \end{pmatrix} = \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}$$

We are finally ready to provide the formulation of the tensor product we will adopt for our purposes.

Definition 2.19: Kronecker product

Given two matrice $A \in \mathbb{F}^{m \times n}$ and $B \in \mathbb{F}^{p \times q}$ defined over some field $\mathbb{F}$, it holds that

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix} \in \mathbb{F}^{mp \times nq}$$

From now on, we will assume that the output of any tensor product is in this form. Most importantly, through the Kronecker product the following property holds, which will be used extensively through these notes.

Proposition 2.19

For any field $\mathbb{F}$, given two matrices A and B and two vectors v and w such that

$$A \in \mathbb{F}^{m \times n} \quad B \in \mathbb{F}^{p \times q} \quad v \in \mathbb{F}^{n} \quad w \in \mathbb{F}^{q}$$

it holds that

$$(A \otimes B)(v \otimes w) = Av \otimes Bw$$

As a final note, if U is a matrix (or a vector), we will write

$$U^{\otimes n} = \underbrace{U \otimes \ldots \otimes U}_{n \text{ times}}$$

similar to the usual product

$$U^{n} = \underbrace{U \cdot \ldots \cdot U}_{n \text{ times}}$$

3

Postulates of quantum mechanics

TODO longer intro?

Now that we defined Hilbert spaces and their operators in great detail, we can finally present we needed this mathematical foundations in order to progress: quantum mechanics is developed over Hilbert spaces with *countable* bases, and quantum computing works with finite-dimensional Hilbert spaces. In particular, these are the four fundamental **postulates of quantum mechanics**.

3.1 Postulates of quantum mechanics

3.1.1 First postulate

Postulate 3.1: State postulate

The state of a quantum system is completely described by a vector $|\psi\rangle$ in a Hilbert space $\mathcal{H}$.

As we saw at the beginning of the previous chapter, $|\psi\rangle$ is always considered to be normalized. We observe that different physical systems of different types live in different Hilbert spaces.

3.1.2 Second postulate

Postulate 3.2: Time evolution postulate

A closed system evolves through time according to the **time-dependent Schrödinger equation (TDSE)**:

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}v(t) = Hv(t)$$

We observe that the Schrödinger equation is a first-order linear differential equation, and it is composed by the following elements:

- $v(t)$ which is the state vector at time t (a vector in a Hilbert space)
- H which is the system *Hamiltonian*, a self-adjoint operator that describes the total energy of the system

The solution of the Schrödinger equation is

$$v(t_1) = U(t_2, t_1)v(t_1)$$

where $U(t_2, t_1)$ is called **time-evaluation operator**, and it is defined as follows:

$$U(t_2, t_1) = e^{-\frac{i}{\hbar}H(t_2 - t_1)}$$

(assuming H does not depend on time). We recall that H is a matrix, so we are raising e to the power of a matrix, an operation that is defined by the power series of the exponential as follows:

$$e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}$$

What is interesting about this operator is that U is **unitary**, and in order to show it is suffices to prove that $U^\dagger = U^{-1}$. But how do we compute the adjoint of U? We observe that by the properties of the adjoint operation it holds that

$$\left(e^A\right)^\dagger = \left(\sum_{n=0}^{\infty} \frac{A^n}{n!}\right)^\dagger = \sum_{n=0}^{\infty} \frac{(A^n)^\dagger}{n!} = \sum_{n=0}^{\infty} \frac{(A^\dagger)^n}{n!} = e^{A^\dagger}$$

which means that the adjoint of an exponential is the exponential of the adjoint. This suffices to prove that

$$U^\dagger = \left(e^{-\frac{i}{\hbar}H(t_2 - t_1)}\right)^\dagger = e^{\left(-\frac{i}{\hbar}H(t_2 - t_1)\right)^\dagger} = e^{\frac{i}{\hbar}H(t_2 - t_1)} = \left(e^{-\frac{i}{\hbar}H(t_2 - t_1)}\right)^{-1} = U^{-1}$$

This is a crucial characteristic for quantum mechanics: since U is unitary, we know that it preserves the scalar product by Proposition 2.9, therefore it also preserves **probabilities and norms**. This is why we say that evolution in quantum systems — or *quantum evolution*, for short — is unitary. Indeed, the second postulate is sometimes formulated equivalently as follows.

Postulate 3.3: Time evolution postulate (alt. ver.)

The evolution of a *closed* quantum system is described by a *unitary transformation*. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 as follows

$$|\psi'\rangle = U\,|\psi\rangle$$

where $U(t_2, t_1) = e^{-\frac{i}{\hbar}H(t_2 - t_1)}$.

3.1.3 Third postulate

Postulate 3.4: Measurement postulate

Every measurable (i.e. *observable*) quantity corresponds to a self-adjoint operator on $\mathcal{H}$. In particular, given an observable A, and a state $v \in \mathcal{H}$, it holds that:

- the only possible results of measuring A are one of its eigenvalues
- the probability of measuring eigenvalue λ in state v is given by

$$\Pr[A = \lambda \mid v] = \langle v | P_\lambda v \rangle$$

where P_λ is the linear map that projects v onto the λ-eigenspace.

We can actually explain why we choose that particular scalar product to be the probability. Since by convention any quantum state is normalize, i.e. $||v|| = 1$, it holds that

$$\begin{aligned} 1 &= ||v||^2 \\ &= \langle v | v \rangle \\ &= \left\langle \sum_{i=1}^{m} P_{\lambda_i} v \middle| \sum_{j=1}^{m} P_{\lambda_j} v \right\rangle \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} \langle P_{\lambda_i} v | P_{\lambda_j} v \rangle \end{aligned}$$

Now, since each P_{λ_i} is a projector, we know that when $i \neq j$ it holds that $P_{\lambda_i} P_{\lambda_j} = \mathbf{0}$, therefore by Hermiticity of projectors we have that

- if $i \neq j$ then
$$\langle P_{\lambda_i} v | P_{\lambda_j} v \rangle = \langle v | P_{\lambda_i} P_{\lambda_j} v \rangle = \langle v | \mathbf{0} v \rangle = 0$$
- if $i = j$ then
$$\langle P_{\lambda_i} v | P_{\lambda_j} v \rangle = \langle P_{\lambda_i} v | P_{\lambda_i} v \rangle = ||P_{\lambda_i} v||^2$$

Therefore, by adding only the non-zero terms we get that

$$\sum_{i=1}^{m} ||P_{\lambda_i} v||^2 = 1$$

Hence, we define

$$\Pr[A = \lambda_i \mid v] := ||P_{\lambda_i} v||^2$$

such that

$$\Pr[A \mid v] = \sum_{i=1}^{m} \Pr[A = \lambda_i \mid v] = \sum_{i=1}^{m} ||P_{\lambda_i} v||^2 = ||v||^2 = 1$$

which also means that our probabilities will add up to 1 automatically. Finally, we can rewrite this proability as follows (we will drop the index of the eigenvalue):

$$\begin{aligned}\Pr[A = \lambda \mid v] &= \langle P_\lambda v | P_\lambda v\rangle \\ &= \langle v | P_\lambda^\dagger P_\lambda v\rangle \\ &= \langle v | P_\lambda^2 v\rangle \qquad \text{(by Hermiticity)} \\ &= \langle v | P_\lambda P_\lambda v\rangle \\ &= \langle v | P_\lambda v\rangle\end{aligned}$$

This formulation was refined in 1926 by Max Born [Hal13], when he derived the following property.

Theorem 3.1: Born rule

The probability that a qubit $|\psi\rangle$ written in a basis $\{\lambda_i\}_{i=1}^n$ collapses to a particular $|\lambda\rangle \in \{\lambda_i\}_{i=1}^n$ when measured is

$$\Pr[\text{measure}(|\psi\rangle = |\lambda\rangle)] = |\langle\psi|\lambda\rangle|^2$$

Proof. By the Theorem 2.3 it holds that the set of all the eigenvectors $|\lambda\rangle$ of any operator can be expanded to always form an orthonormal basis of the complete Hilbert space. Hence, the idea is to implicitly construct a self-adjoint operator A whose eigenvalues are precisely the possible values in which $|\psi\rangle$ might collapse into. In other words, we want to construct a self-adjoint operator whose Spectral decomposition is exactly defined by $\{\lambda_1, \ldots, \lambda_n\}$ Hence, if $|\psi\rangle$ is defined as

$$|\psi\rangle = \sum_{i=1}^{n} \alpha_n |\lambda_i\rangle$$

we define the operator

$$A_\psi = \sum_{i=1}^{n} \lambda_i |\lambda_i\rangle \langle\lambda_i|$$

Thus, the probability that by $|\psi\rangle$ it collapses to some $|\lambda\rangle \in \{\lambda_1, \ldots, \lambda_n\}$ can be rewritten as follows:

$$\begin{aligned}\Pr[\text{measure}(|\psi\rangle = |\lambda\rangle)] &= \Pr[A_\psi = \lambda |\, |\psi\rangle] \\ &= \langle\psi | P_\lambda \psi\rangle \qquad \text{(by Postulate 3.4)} \\ &= \langle\psi | (|\lambda\rangle \langle\lambda|) \psi\rangle \\ &= \langle\psi |\, |\lambda\rangle \langle\lambda |\, \psi\rangle \\ &= (\langle\psi|\lambda\rangle)(\langle\lambda|\psi\rangle) \\ &= |\langle\psi|\lambda\rangle|^2\end{aligned}$$

□

For instance, if we have a superposition

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

and we want to know what is the probability that $|\psi\rangle$ collapses to $|0\rangle$ after a measurement, we simply have that

$$\begin{aligned}\Pr[\text{measure}(|\psi\rangle = |0\rangle)] &= \Pr[A_\psi = 0|\,|\psi\rangle] \\ &= |\langle\psi|0\rangle|^2\end{aligned}$$

where the A_ψ matrix is precisely:

$$A_\psi = 0 \cdot P_0 + 1 \cdot P_1 = 0 \cdot |0\rangle\langle 0| + 1 \cdot |1\rangle\langle 1|$$

This formulation of the probability of measurements will be used extensively for our purposes, and allows us to avoid the description of the matrix A_ψ completely.

Before presenting to the next postulate, another very important operator that is frequently utilized in quantum mechanics is the **expected value** of a matrix. Given an observable A, we define the expected value of A as the average eigenvector we may obtain after a measurement

$$\mathbb{E}[A|v] = \sum_{i=1}^{m} \lambda_i \Pr[A = \lambda_i \mid v]$$

We usually denote the expected value of the operator A given that we are in state $|\psi\rangle$ as $\langle A\rangle_\psi$ (or $\langle A\rangle$ if the context is clear enough). Moreover, we have the following property.

Proposition 3.1: Expected value of an operator

Given an Hermitian operator A, if the system is in state $|\psi\rangle$ it holds that

$$\langle A\rangle_\psi = \langle\psi|A\psi\rangle$$

Proof. By Postulate 3.4, it follows that

$$\begin{aligned}\langle A\rangle_\psi &= \mathbb{E}[A \mid |\psi\rangle] \\ &= \sum_{i=1}^{m} \lambda_i \Pr[A = \lambda_i \mid |\psi\rangle] \\ &= \sum_{i=1}^{m} \lambda_i \langle\psi|P_{\lambda_i}|\psi\rangle \\ &= \langle\psi| \left(\sum_{i=1}^{m} \lambda_i P_{\lambda_i}\right) |\psi\rangle \\ &= \langle\psi|\,A\,|\psi\rangle \qquad \text{(this is } A\text{'s Spectral decomposition)} \\ &= \langle\psi|A\psi\rangle\end{aligned}$$

□

Notably, Proposition 2.14 directly implies the following observation.

Corollary 3.1

Given an Hermitian operator A, if the system is in state $|\psi\rangle$ it holds that

$$\langle A\rangle_\psi = \mathrm{tr}(A\,|\psi\rangle\,\langle\psi|)$$

3.1.4 Fourth postulate

Postulate 3.5: Composite systems postulate

If system A is defined over $\mathcal{H}_A$, and system B is defined over $\mathcal{H}_B$, the total system lives in

$$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$$

In other words, the last postulate states that the Hilbert space of a composite system is the tensor product of the Hilbert spaces of its subsystems. This postulate immediately tells us something important about **entangled** quantum systems: if a system C defined over a Hilbert space $\mathcal{H}_C$ represents a quantum state, there is no pair of systems A and B defined over $\mathcal{H}_A$ and $\mathcal{H}_B$ such that $\mathcal{H}_C = \mathcal{H}_A \otimes \mathcal{H}_B$.

Speaking of tensor products, when operators can be factored out into smaller operators of smaller systems we obtain linearity of expectations w.r.t. the tensor product.

Proposition 3.2

Given an operator O that can be factored out as $O = A \otimes B$, if the current state $|\psi\rangle$ is not entangled it holds that

$$\langle O\rangle = \langle A\rangle \cdot \langle B\rangle$$

Proof. We need to show that

$$\langle\psi|O\psi\rangle = \langle\psi_A|A\psi_A\rangle \cdot \langle\psi_B|B\psi_B\rangle$$

where $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$, which we know exists because we are assuming that $|\psi\rangle$ is not entangled.

Let the following be the spectral decompositions of A and B, respectively:

$$A = \sum_{i=1}^{m} \lambda_i^A P_{\lambda_i^A}, \qquad B = \sum_{j=1}^{m} \lambda_j^B P_{\lambda_j^B}$$

Then, by computing the tensor product between A and B, we immediately obtain the spectral decomposition of O:

$$O = A \otimes B = \sum_{i,j\in[m]} \lambda_i^A \lambda_j^B (P_{\lambda_i^A} \otimes P_{\lambda_j^B})$$

Hence, the possible values O might be observed into are all the products $\lambda_i^A \lambda_j^B$ for $i, j \in [m]$. Lastly, if we define

$$\forall i, j \in [m] \quad \lambda_k = \lambda_{(i-1)m+j}$$

it suffices to consider the definition of expected value:

$$\begin{aligned}
\langle \psi | O\psi \rangle &= \sum_{k=1}^{m^2} \lambda_k \Pr[O = \lambda_k \mid |\psi\rangle] \\
&= \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i^A \lambda_j^B \Pr[A \otimes B = \lambda_i^A \lambda_j^B \mid |\psi_A\rangle \otimes |\psi_B\rangle] \\
&= \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i^A \lambda_j^B \langle \psi_A \otimes \psi_B | P_{\lambda_i^A} \otimes P_{\lambda_j^B} | \psi_A \otimes \psi_B \rangle \\
&= \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i^A \lambda_j^B \left(\langle \psi_A | P_{\lambda_i^A} | \psi_A \rangle \cdot \langle \psi_B | P_{\lambda_j^B} | \psi_B \rangle \right) \\
&= \sum_{i=1}^{m} \left(\lambda_i^A \langle \psi_A | P_{\lambda_i^A} | \psi_A \rangle \sum_{j=1}^{m} \lambda_j^B \langle \psi_B | P_{\lambda_j^B} | \psi_B \rangle \right) \\
&= \left(\sum_{i=1}^{m} \lambda_i^A \langle \psi_A | P_{\lambda_i^A} | \psi_A \rangle \right) \cdot \left(\sum_{j=1}^{m} \lambda_j^B \langle \psi_B | P_{\lambda_j^B} | \psi_B \rangle \right) \\
&= \langle \psi_A | A\psi_A \rangle \cdot \langle \psi_B | B\psi_B \rangle
\end{aligned}$$

□

This result should not come as a surprise, since for any two distributions X and Y we know that

$$\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$$

only if X and Y are **independent**. Indeed, by assuming that O and $|\psi\rangle$ can be factorized, we are assuming that the two underlying subsystems $\mathcal{H}_A$ and $\mathcal{H}_B$ — in which $|\psi_A\rangle$ and A, and $|\psi_B\rangle$ and B live, respectively — are "independent" in the sense that they are *not entangled*!

3.2 The density operator

We have formulated quantum mechanics using state vectors and the four postulates described in the previous section. However an alternate formulation is possible using a tool known as the **density operator**, or *density matrix*. This formulation will be mathematically equivalent to the state vector approach. However, before proceeding we need to revise some definitions we provided in the previous section.

3.2.1 Generalization of the third postulate

The postulates we have just presented are internally consistent, however, to proceed further we require additional clarification regarding the Measurement postulate. In particular, in the statement of this postulate we required that measurables are self-adjoint (i.e. Hermitian) operators. To be precise, this is a special case of a more general formalization of the Measurement postulate which we need to discuss.

Postulate 3.6: General measurement postulate

Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators* acting on the state space of the system being measured — the index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement, then the probability that result m occurs is given by

$$\Pr[m \mid |\psi\rangle] = \langle\psi| M_m^\dagger M_m |\psi\rangle$$

and the measurement operators satisfy the **completeness equation**

$$\sum_m M_m^\dagger M_m = I$$

We observe that the completeness equation expresses the fact that probabilities of the outcomes sum to one:

$$\begin{aligned}
\sum_m \Pr[m \mid |\psi\rangle] &= \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle \\
&= \langle\psi| \left(\sum_m M_m^\dagger M_m \right) |\psi\rangle \\
&= \langle\psi| I |\psi\rangle \\
&= \langle\psi|\psi\rangle \\
&= 1
\end{aligned}$$

For instance, consider a single qubit

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

and suppose we want to measure it in the computational basis. The two measurement operators we require are precisely

$$M_0 := |0\rangle \langle 0| \qquad M_1 := |1\rangle \langle 1|$$

We observe that these two operators are *projectors*, so they trivially satisfy the completeness equation

$$\begin{aligned}
M_0^\dagger M_0 + M_1^\dagger M_1 &= M_0^2 + M_1^2 \\
&= M_0 + M_1 \qquad \text{(resolution of } I\text{)} \\
&= I
\end{aligned}$$

Then, the probability of obtaining 0 from measuring $|\psi\rangle$ is

$$\Pr[0 \mid |\psi\rangle] = \langle\psi|M_0^\dagger M_0|\psi\rangle = \langle\psi|M_0|\psi\rangle = |\alpha|^2$$

and similarly we ge that $\Pr[1 \mid |\psi\rangle] = |\beta|^2$, as we would expect.

So, what is the connection with the initial version of the third postulate we originally provided? A special case of the General measurement postulate when the measurements are *projective measurements*, i.e. measurements described by Hermitian operators. Indeed, for many applications of quantum computation and quantum information in general we are concerned primarily with projective measurements. By requiring that observables are Hermitian operators the third postulate reduces to the formulation we provided in the previous section, because any observable M would have a spectral decomposition of

$$M = \sum_m mP_m$$

which immediately implies that

$$\Pr[m \mid |\psi\rangle] = \langle\psi|P_m\psi\rangle$$

as defined earlier.

3.2.2 Postulates through density operators

We can now introduce the **density operator** discussed at the beginning of the section. First, we describe the context. Consider a quantum system, and suppose that it is in one of a number of states $|\psi_1\rangle, \ldots, |\psi_N\rangle$. Each $|\psi_i\rangle$ is associated with a probability p_i that the system is in the i-th state. We call

$$\{p_i, |\psi_i\rangle\}_{i=1}^N$$

an **ensable of states**. Then, we the following definition.

Definition 3.1: Density operator

Given a quantum system described by the ensamble $\{p_i, |\psi_i\rangle\}_{i=1}^N$, the **density operator** of the system is defined as follows

$$\rho := \sum_{i=1}^N p_i \, |\psi_i\rangle \, \langle\psi_i|$$

We observe that ρ is a matrix, and this is the reason why it is interchangeably called *density matrix*. Nevertheless, this matrix is important because it turns out that all the postulates of quantum mechanics we presented so far can be reformulated equivalently in terms of the density operator.

Suppose that we have closed quantum system described by some ensamble $\{p_i, |\psi_i\rangle\}_{i=1}^N$ that unitarily evolves following the unitary operator U described earlier

$$U(t_2, t_1) = e^{-\frac{i}{\hbar}H(t_2 - t_1)}$$

This means that the system is in state $|\psi_i\rangle$ with probability p_i, and after the evolution has occurred the system will be in the state $U|\psi_i\rangle$ still with probability p_i. More specifically, we have that

$$|\psi_i\rangle \xrightarrow{U} U|\psi_i\rangle$$

which also directly implies that

$$(|\psi_i\rangle)^\dagger = \langle\psi_i| \xrightarrow{U} (U|\psi_i\rangle)^\dagger = \langle\psi_i| U^\dagger$$

Therefore, after the system has evolved we must *update* its density operator by applying U to each $|\psi_i\rangle$ — we don't know in which state the system is in with certainty by the rules of quantum mechanics, so we take into account all of the possible $|\psi_i\rangle$

this explaination sucks find a better reason tbh

$$\rho = \sum_{i=1}^{N} p_i |\psi_i\rangle\langle\psi_i| \xrightarrow{U} \sum_{i=1}^{N} p_i U|\psi_i\rangle\langle\psi_i| U^\dagger = U\rho U^\dagger$$

Not surprisingly, this shows that by linearity we just need to apply U (and $U^\dagger$) to the matrix directly in order to consider all the evolutions. The function that maps

$$\rho \mapsto U\rho U^\dagger$$

is called **superoperator**.

Measurements can be also easily described through the density operator. Consider the General measurement postulate, and suppose we perform a measurement described by measurement operators M_m. If the initial state was $|\psi_i\rangle$, then we have that

$$\begin{aligned}\Pr[m|\,|\psi_i\rangle] &= \langle\psi_i|M_m^\dagger M_m|\psi_i\rangle \\ &= \operatorname{tr}(M_m^\dagger M_m |\psi_i\rangle\langle\psi_i|) \qquad \text{(by Proposition 2.14)}\end{aligned}$$

Then, thanks to Proposition 2.13, by computing the total probability we obtain that

$$\begin{aligned}\Pr[m] &= \sum_{i=1}^{N} p_i \Pr[m \mid |\psi_i\rangle] \\ &= \sum_{i=1}^{N} p_i \operatorname{tr}(M_m^\dagger M_m |\psi_i\rangle\langle\psi_i|) \\ &= \sum_{i=1}^{N} \operatorname{tr}(p_i M_m^\dagger M_m |\psi_i\rangle\langle\psi_i|) \\ &= \operatorname{tr}\left(\sum_{i=1}^{N} p_i M_m^\dagger M_m |\psi_i\rangle\langle\psi_i|\right) \\ &= \operatorname{tr}\left(M_m^\dagger M_m \sum_{i=1}^{N} p_i |\psi_i\rangle\langle\psi_i|\right) \\ &= \operatorname{tr}(M_m^\dagger M_m \rho)\end{aligned}$$

TODO

skipping rho m, if needed put it here and add what it is needed

This shows that the basic postulates of quantum mechanics related to unitary evolution and measurement can be rephrased in terms of density operators. However, we can do better: we will proivde a characterization of the density operator that does not rely on the idea of state vectors *at all*. But first, as usual, we need some definitions. When a quantum system is an a known exact state $|\psi\rangle$, the system is said to be in a **pure state**, and in this case its density matrix operator is simply

$$\rho = |\psi\rangle \langle\psi|$$

Otherwise, if the state is now known and the system is described by an ensamble of states, we say that ρ is in a **mixed state**. More specifically, we can distinguish between pure and mixed states as follows.

Proposition 3.3

If a system is in a pure state it holds that $\text{tr}(\rho^2) = 1$, otherwise if it is in a mixed state it holds that $\text{tr}(\rho^2) < 1$.

Proof. TODO □ da fare

We can finally present the characterization of density operators we anticipated.

Theorem 3.2: Density operators characterization

Given a system described by the ensamble $\{p_i, |\psi_i\rangle\}_{i=1}^N$, an operator ρ is the density operator of the system if and only if

- ρ is positive
- $\text{tr}(\rho) = 1$

Proof. We first prove the first implication. Consider the density operator

$$\rho = \sum_{i=1}^{N} p_i |\psi_i\rangle \langle\psi_i|$$

Fix an arbitrary $|v\rangle \in \mathcal{H}$; then, we get that:

$$\begin{aligned}
\langle v|\rho v\rangle &= \langle v|\rho|v\rangle \\
&= \langle v| \left(\sum_{i=1}^{N} p_i |\psi_i\rangle \langle\psi_i| \right) |v\rangle \\
&= \sum_{i=1}^{N} p_i \langle v|\psi_i\rangle \langle\psi_i|v\rangle \\
&= \sum_{i=1}^{N} p_i |\langle v|\psi_i\rangle|^2 \\
&\geq 0
\end{aligned}$$

This proves that ρ is positive. For the second statement, by the property of the trace proved in Proposition 2.13 and Proposition 2.14, we obtain that

$$\begin{aligned}
\operatorname{tr}(\rho) &= \operatorname{tr}\left(\sum_{i=1}^{N} p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|\right) \\
&= \sum_{i=1}^{N} p_i \operatorname{tr}(\left|\psi_i\right\rangle \left\langle\psi_i\right|) \\
&= \sum_{i=1}^{N} p_i \left\langle\psi_i\right| I \left|\psi_i\right\rangle \\
&= \sum_{i=1}^{N} p_i \left\langle\psi_i|\psi_i\right\rangle \\
&= \sum_{i=1}^{n} p_i \\
&= 1
\end{aligned}$$

Now we can proceed to prove the converse implication. Suppose that ρ is any operator that satisfies the conditions of the statement. In particular, since ρ is positive by Proposition 2.8 we know that ρ is Hermitian, which implies that it admits a spectral decomposition TODO □

da finire, troppo noioso

Proposition 3.4: Convexity of density matrices

Given density matrices $\rho_1, \ldots, \rho_n$ and probabilities $p_1, \ldots, p_n$ it holds that $\sum_{i=1}^{N} p_i \rho_i$ is a density matrix.

Proof. TODO □

da fare

Given all we just discussed, we can finally rephrase the postulates through the density operator.

Postulate 3.7: State postulate (density operator)

Associated to any isolated physical system is a Hilbert space known as the *state space* of the system, and the system is completely described by its *density operator*, i.e. any positive operator ρ with trace one acting on the state space of the system.

Postulate 3.8: Time evolution postulate (density op.)

The evolution of a *closed* quantum system is described by a *unitary transformation*, i.e. the state ρ of the system at time t_1 is related to the state ρ' of the system at time t_2 as follows

$$\rho' = U\rho U^\dagger$$

where $U(t_2, t_1) = e^{-\frac{i}{\hbar}H(t_2-t_1)}$.

Postulate 3.9: General measurement postulate (dens. op.)

Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators* acting on the state space of the system being measured — the index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is ρ immediately before the measurement, then the probability that result m occurs is given by

$$\Pr[m \mid |\psi\rangle] = \operatorname{tr}(M_m^\dagger M_m \rho)$$

and the measurement operators satisfy the **completeness equation**

$$\sum_m M_m^\dagger M_m = I$$

TODO

ex: do the projection one

Postulate 3.10: Composite system postulate (dens. op.)

The sate space of a composite physical system is the tensor product of the state spaces of the component physical systems. That is, given n systems such that the i-th system is in state ρ_i, the joint state of the total system is

$$\rho = \bigotimes_{i=1}^{n} \rho_i$$

As a final note, observe that knowing the density matrix of a system tells us *nothing* about the ensamble of states of the systems, because different ensambles can generate the same density matrix. For instance, the matrix

$$\rho = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

which can be generated as follows

$$\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$$

This *might* suggest that the ensamble of states of our system is

$$\left\{\left(\frac{1}{2}, |0\rangle\langle 0|\right), \left(\frac{1}{2}, |1\rangle\langle 1|\right)\right\}$$

but it would be a mistake to make this conclusion because ρ can be also written for examòle as follows

$$\rho = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

which describes a completely different ensamble!

3.2.3 Entangled systems

So far we only dealt with one single system, and its associated density operator, but perhaps the deepst application of the latter is as a descriptive tool for *composite* quantum systems. Suppose we have two physical systems A and B, whose composite state is described by a density operator ρ^{AB}. What can we say about ρ^A and ρ^B?

Definition 3.2: Partial trace

Given two quantum systems A and B defined over Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively, we define the **partial trace** of B as follows:

$$\mathrm{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) := |a_1\rangle\langle a_2| \,\mathrm{tr}(|b_1\rangle\langle b_2|)$$

for any $|a_1\rangle, |a_2\rangle \in \mathcal{H}_A$ and $|b_1\rangle, |b_2\rangle \in \mathcal{H}_B$.

The partial trace of A is defined analogously. We observe that, by Proposition 2.14 if follows that

$$\mathrm{tr}(|b_1\rangle\langle b_2|) = \langle b_2|b_1\rangle$$

which means that

$$\mathrm{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) := |a_1\rangle\langle a_2|\langle b_2|b_1\rangle$$

From this operator — which can be proved to be **linear** — we define the **reduced density operator**.

Definition 3.3: Reduced density operator

Given two quantum systems A and B whose composite system is described by ρ^{AB}, we define the **reduced density operator** of A as follows

$$\rho^A := \mathrm{tr}_B(\rho^{AB})$$

The reduced density operator of B is defined analogously.

Firstly, it is not obvious that ρ^A is in any sense a description for the state of system A, so we shall discuss the definition. For instance, suppose that the composite quantum system

of A and B is described by ρ^{AB} that can itself be written as follows:

$$\rho^{AB} = \tau \otimes \sigma$$

where τ is a density operator for A, and σ is a density operator for B. The connection with the reduced density operator is evident by computing ρ^A:

$$\begin{aligned}
\rho^A &= \mathrm{tr}_B(\rho) \\
&= \mathrm{tr}_B(\tau \otimes \sigma) \\
&= \tau \,\mathrm{tr}(\sigma) \\
&= \tau \qquad\qquad \text{(by Theorem 3.2)}
\end{aligned}$$

Similarly, we get that $\rho^B = \sigma$.

A more interesing example, however, is an **entangled system**. For instance, let's consider the bell state

$$|\Phi^+\rangle := \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

We can compute the density operator of this system by following the definition

$$\begin{aligned}
\rho &= |\Phi^+\rangle \langle\Phi^+| \\
&= \left(\frac{|00\rangle + \langle 11|}{\sqrt{2}}\right)\left(\frac{|00\rangle + \langle 11|}{\sqrt{2}}\right) \\
&= \frac{|00\rangle\langle 00| + |11\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00|}{2}
\end{aligned}$$

As we already know, since this system is entangled there is no pair of subsystems A and B that can produce it as a composite tensor product. Nevertherless, let's see what happens when we try to compute ρ^A

$$\begin{aligned}
\rho^A &= \mathrm{tr}_B(\rho) \\
&= \mathrm{tr}_B\left(\frac{|00\rangle\langle 00| + |11\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00|}{2}\right) \\
&= \frac{\mathrm{tr}_B(|00\rangle\langle 00|) + \mathrm{tr}_B(|11\rangle\langle 00|) + \mathrm{tr}_B(|00\rangle\langle 11|) + \mathrm{tr}_B(|11\rangle\langle 00|)}{2} \\
&= \frac{|0\rangle\langle 0|\langle 0|0\rangle + |1\rangle\langle 0|\langle 0|1\rangle + |0\rangle\langle 1|\langle 1|0\rangle + |1\rangle\langle 1|\langle 1|1\rangle}{2} \\
&= \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} \\
&= \frac{I}{2}
\end{aligned}$$

Now, we observe that

$$\mathrm{tr}\left(\left(\frac{I}{2}\right)^2\right) = \frac{1}{2} < 1$$

and by Proposition 3.3 this immediately implies that the system is in a **mixed state**! Observe what just happened: the entangled system is in a *pure state*, because

$$\rho = |\Phi^+\rangle\langle\Phi^+|$$

however, ρ^A (and ρ^B analogously) is mixed! This means that the state of the joint system is known *exactly*, however the state of the first qubit is not maximally known. This strange property is another hallmark of quantum entanglement.

4

Quantum algorithms

Now that we presented all the mathematical tools we need to perform quantum computations, we are ready to explore some of the most famous and most important quantum algorithm that have been developed in recent years. But before introducing any algorithm, let's discuss *why* we are interested in quantum computing at all — keep in mind that is just a brief overview of the general ideas and we will delve into the exact details as soon as we present some quantum algorithm more in depth.

Consider some computable function $f(x)$ and some classical algorithm that is able to compute it for any valid input x. If we have an input x_1, and we want to compute its output we need to run the algorithm in order to compute $f(x_1)$. Analogously, if we have another input x_2 we need to run the algorithm again in order to compute $f(x_2)$. In general, if we want to know the outputs $f(x_1), \dots, f(x_N)$ of N inputs $x_1, \dots, x_N$, with classical computing we must run the algorithm N distinct times, because there is no way to compute more than one output at a time. With quantum computers, however, we will see that not only this is possible, but we can actually compute *all* the possible outputs related to all the possible inputs **simultaneously**. To the best of our knowledge, through the laws of quantum mechanics we *really* can compute all the possible outputs $f(x)$ for each $x \in \mathbb{B}^n$ — for a fixed input length n.

How does this work in practice? Recall that we *seemingly arbitrarily* decided to denote the vectors of the canonical basis with binary strings, such as $|000\rangle$, $|001\rangle$, $|010\rangle$ and so on (we chose $n = 3$ for this example). This is no coincidence: since this vectors form a basis, we can write any quantum state $|\psi\rangle$ as a linear combination of them, and by doing so we obtain something like

$$|\psi\rangle = \sum_{x \in \mathbb{B}^3} \alpha_x |x\rangle$$

Say that we want to compute $f(x)$ for all $x \in \mathbb{B}^3$. In practice, we construct a unitary operator U_f that implements f **reversibly** — a property that we will shortly outline — for example by acting on an auxiliary register

$$U_f(|x\rangle \otimes |0\rangle) = |x\rangle \otimes |f(x)\rangle$$

Now, by linearity of quantum operators when we apply U to $|\psi\rangle \otimes |0\rangle$ we get that

$$U(|\psi\rangle \otimes |0\rangle) = \sum_{x \in \mathbb{B}^3} \alpha_x U |x\rangle |0\rangle = \sum_{x \in \mathbb{B}^3} \alpha_x |x\rangle |f(x)\rangle$$

Notice what happened here! By only applying U to $|\psi\rangle |0\rangle$ we actually applied U on *all the vectors of the basis* **simultaneously**. This explains the choice of the labeling of the vectors of the canonical basis.

So, what's the catch? Well, we also recall that superpositions of states must be measured at some point, and this is the problem: when we measure the result of $U |\psi\rangle |0\rangle$ we will inevitably only get only *one* single outcome. In particular, if we measure both registers, we obtain only one $(x, f(x))$ pair for some $x \in \mathbb{B}^3$ with probability $|\alpha_x|^2$. This makes finding *useful* quantum algorithm extremely difficult, because even if we are performing all the computations at once we can only see 1 possible output, *at random*. Hence, not only quantum algorithms are hard to discover because of this inherent limitation of quantum mechanics, but they must also be more efficient than any classical alternative we currently know, otherwise there is really no point in using this very complicated computing framework (both in terms of hardware and software). The algorithms that we will see in this chapter — and also in the next one — are some of the most important quantum procedures that we know, and sparked a lot of interest in this area of research in recent years.

But before starting our discussion about quantum algorithms, we need to discuss the *reversibility* aspect previously mentioned. When we introduced quantum operators we underlined the fact that each quantum gate has to be a *unitary* operator, and we now have the mathematical foundation to know that if a matrix is unitary, it is clearly also invertible — indeed, its adjoint is its inverse. This directly implies a very important property of quantum computation: except for the measurement operation, every quantum computation operation is **reversible**.

4.1 Deutsch's algorithm

Even though quantum computation provides reversibility "for free", *classical computation* can still achieve invertibility of computation, as we will see in Section 8.2.2. However, the next characteristic that we are going to describe has no classical analogue.

First, let's start with a problem seemingly unrelated to our discussion. Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, we would like to embed f inside a quantum computation. However, when $n \geq 3$ we are guaranteed that $f(x)$ is not reversible — it cannot be injective. This is a problem, since in quantum coputing all gates must be reversible — given that quantum evolution is unitary. Thus, how do we turn f into a reversible computation?

We define a map U_f defined as follows:

$$U_f : \{0,1\}^{n+1} \to \{0,1\}^{n+1} : (x, y) \mapsto (x, y \oplus f(x))$$

First, we observe that

$$(y \oplus f(x)) \oplus f(x) = y \oplus (f(x) \oplus f(x)) = y$$

which trivially proves that U_f is reversible. Moreover, we can actually prove that when applied to qubits the corresponding quantum operator

$$U_f : \mathcal{H} \to \mathcal{H} : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus f(x)\rangle$$

is indeed unitary — we observe that we are omitting the tensor product symbol in the function definition, as usual in the literature.

Proposition 4.1

Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the operator U_f is unitary.

Proof. TODO □ TODO

This proves that the construction of U_f is precisely the gate that allows us to embed f into any quantum computation. Moreover, we observe that

- $|y\rangle = |0\rangle \implies U_f\,|x\rangle\,|0\rangle = |x\rangle\,|f(x)\rangle$
- $|y\rangle = |1\rangle \implies U_f\,|x\rangle\,|1\rangle = |x\rangle\,|\neg f(x)\rangle$

However, until now we only considered already collapsed qubits, but what if we consider a quantum input that is in a superposition? For instance, let

$$|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

and assume that $|y\rangle = |0\rangle$ for simplicity; this implies that

$$\begin{aligned} U_f\,|x\rangle\,|y\rangle &= U_f \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= U_f \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\ &= \frac{1}{\sqrt{2}}(U_f\,|00\rangle + U_f\,|10\rangle) \qquad \text{(by linearity of } U_f\text{)} \\ &= \frac{1}{\sqrt{2}}(|0\rangle\,|f(0)\rangle + |1\rangle\,|f(1)\rangle) \end{aligned}$$

But notice what just happened: both $f(0)$ and $f(1)$ have been computed **simultaneously**, in one gate application. This has no classical equivalent, we would have to evaluate $f(0)$ and $f(1)$ *separately*. This phenomenon is called **quantum parallelism**, and it can be achieved only because:

- qubits are in superpositions
- quantum gates are linear

However, we observe that the result of our calculations is *still a superposition*. In fact, if we measure the output of $U_f\,|x\rangle\,|y\rangle$ we would still get either $|0\rangle\,|f(0)\rangle$ or $|1\rangle\,|f(1)\rangle$, both with 50% probability. This is a problem: the fact that we can compute $f(0)$ and $f(1)$ at the same time seems promising, but can we retrieve their actual values?

Unfortunately, this is not possible. Indeed, quantum parallelism cannot help us with *local* properties — i.e. when we need all individual outputs — it can only help when we need **global** properties. This limit derives from the fact that measurements prevent "seeing" both outcomes, in fact if we were able to compute $f(0)$ and $f(1)$ simultaneously from this superposition we would be violating the laws of quantum mechanics themselves.

Then, how do we extract useful *global* information from the superposition output? In 1985 Deutsch [Deu85] defined a quantum algorithm which is able to compute $f(0) \oplus f(1)$, which clearly tells us if $f(0)$ equals $f(1)$ or not.

Algorithm 4.1: Deutsch algorithm

Given a Boolean function f and 2 qubits, the algorithm returns $|0\rangle$ if f if $f(0) = f(1)$, $|1\rangle$ otherwise.

1: **function** DEUTSCH(f, q_0, q_1)
2: $\quad q_1 \leftarrow X(q_1)$
3: $\quad q_0, q_1 \leftarrow (H \otimes H)(q_0, q_1)$
4: $\quad q_0, q_1 \leftarrow U_f(q_0, q_1)$
5: $\quad q_0 \leftarrow H(q_0)$
6: $\quad$ **return** measure(q_0)
7: **end function**

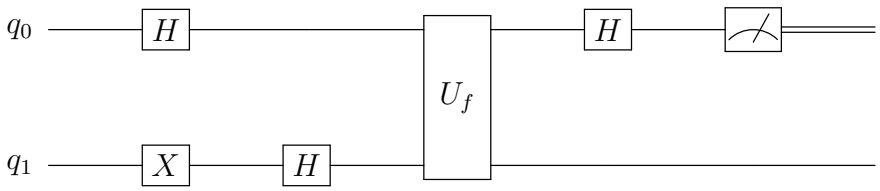


Figure 4.1: The quantum circuit for Deutsch's algorithm. The box labeled with U_f represents a "black-box" for whatever computation U_f represents (which directly depends on the chioce of f).

Proving that this quantum circuit is correct, however, will be a little more involved than what we did for the quantum teleportation. First, we need a lemma that will simplify our calculations.

Lemma 4.1

For any Boolean function f defined on n bits, and $a \in \{0,1\}^n$, it holds that

$$U_f |a\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = (-1)^{f(a)} |a\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Proof. First, by algebraic manipulation we see that

$$\begin{aligned} U_f \left|a\right\rangle \otimes \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) &= U_f \frac{1}{\sqrt{2}}(\left|a0\right\rangle - \left|a1\right\rangle) \\ &= \frac{1}{\sqrt{2}}(U_f \left|a0\right\rangle - \left|a1\right\rangle) \\ &= \frac{1}{\sqrt{2}}(\left|a\ f(a)\right\rangle - \left|a\ \neg f(a)\right\rangle) \end{aligned}$$

and now, we observe that

- if $f(a) = 0$, then

$$\frac{1}{\sqrt{2}}(\left|a0\right\rangle - \left|a1\right\rangle) = (-1)^0 \left|a\right\rangle \otimes \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$$

- if $f(a) = 1$, then

$$\frac{1}{\sqrt{2}}(\left|a1\right\rangle - \left|a0\right\rangle) = (-1)^1 \left|a\right\rangle \otimes \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$$

□

We are now ready to prove the correctness of Deutsch's algorithm. To make things less cluttered, we will use the following standard notation:

$$\left|+\right\rangle := \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle) \qquad \left|-\right\rangle := \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$$

In particular, we observe that

$$H \left|0\right\rangle = \left|+\right\rangle \qquad H \left|1\right\rangle = \left|-\right\rangle$$

Moreover, we will omit the subscript of the corresponding qubit when the context is clear

enough

$$
\begin{aligned}
& q_0 \otimes q_1 \\
=& \left|0\right\rangle \otimes \left|0\right\rangle \\
\xrightarrow{X(q_1)} & \left|0\right\rangle \otimes \left|1\right\rangle \\
\xrightarrow{(H\otimes H)(q_0,q_1)} & \left|+\right\rangle \otimes \left|-\right\rangle \\
=& \frac{1}{\sqrt{2}}(\left|00\right\rangle - \left|01\right\rangle + \left|10\right\rangle - \left|11\right\rangle) \\
=& \frac{1}{\sqrt{2}} \left|0\right\rangle_0 \left|-\right\rangle_1 + \frac{1}{\sqrt{2}} \left|1\right\rangle_0 \left|-\right\rangle_1 \\
\xrightarrow{U_f(q_0,q_1)} & \frac{1}{\sqrt{2}}(-1)^{f(0)} \left|0\right\rangle_0 \left|-\right\rangle_1 + \frac{1}{\sqrt{2}}(-1)^{f(1)} \left|1\right\rangle_0 \left|-\right\rangle_1 && \text{(by the lemma)} \\
=& \frac{1}{\sqrt{2}} \left((-1)^{f(0)} \left|0\right\rangle_0 + (-1)^{f(1)} \left|1\right\rangle_0\right) \otimes \left|-\right\rangle_1 \\
\xrightarrow{H(q_0)} & \frac{1}{\sqrt{2}} \left((-1)^{f(0)} \left|+\right\rangle_0 + (-1)^{f(1)} \left|-\right\rangle_0\right) \otimes \sqrt{2} \left|-\right\rangle_1 \\
=& \frac{1}{2} \left((-1)^{f(0)}(\left|0\right\rangle + \left|1\right\rangle) + (-1)^{f(1)}(\left|0\right\rangle - \left|1\right\rangle)\right) \otimes \sqrt{2} \left|-\right\rangle_1 \\
=& \frac{1}{2} \left(\left((-1)^{f(0)} + (-1)^{f(1)}\right) \left|0\right\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right) \left|1\right\rangle\right) \otimes \sqrt{2} \left|-\right\rangle_1
\end{aligned}
$$

Now, since the final operation of the circuit involves measuring $q_0 = \alpha \left|0\right\rangle + \beta \left|1\right\rangle$, the only two things that we care about are its probability amplitudes, namely

$$\alpha = \frac{1}{2} \left((-1)^{f(0)} + (-1)^{f(1)}\right)$$

$$\beta = \frac{1}{2} \left((-1)^{f(0)} + (-1)^{f(1)}\right)$$

and we see that

- if $f(0) = f(1)$, then

  $$(-1)^{f(0)} = (-1)^{f(1)} \implies \begin{cases} \alpha = \frac{1}{2}\left(2(-1)^{f(0)}\right) = (-1)^{f(0)} \\ \beta = 0 \end{cases}$$

  which implies that

  $$q_0 = (-1)^{f(0)} \left|0\right\rangle + 0 \cdot \left|1\right\rangle = (-1)^{f(0)} \left|0\right\rangle$$

  and we can ignore the $(-1)^{f(0)}$ factor since its a global phase

- if $f(0) \neq f(1)$, then

  $$(-1)^{f(0)} = -(-1)^{f(1)} \implies \begin{cases} \alpha = 0 \\ \beta = \frac{1}{2}\left(2(-1)^{f(0)}\right) = (-1)^{f(0)} \end{cases}$$

  which implies that

  $$q_0 = 0 \cdot \left|0\right\rangle + (-1)^{f(0)} \left|1\right\rangle = (-1)^{f(0)} \left|1\right\rangle$$

  by the same reasoning as the other case

In the end, this proves that if $f(0) = f(1)$, q_0 will collapse to $|0\rangle$, while if $f(0) \neq f(1)$ q_1 will colapse to $|1\rangle$, proving that Deutsch's algorithm works correctly.

4.2 Deutsch-Josza algorithm

Even though Deutsch's algorithm is quite interesting and offers advantages that classical computation cannot achieve, still it seems like it wouldn't be very usefult in practice. In fact, usually we are interested in the *values* of $f(0)$ and $f(1)$, and as we already mentioned quantum mechanics will not allow us to compute both the values at the same time — meaning that even if we use Deutsch's algorithm to now whether $f(0)$ is equal to $f(1)$ or not, we would still need to compute at least one between $f(0)$ and $f(1)$ in order to know both values.

This is because, in reality, the algorithm that we are using is only solving a particular case of a more complex problem. In fact, a couple of years later Deutsch [Deu92] realized that if we use $q_1 = |1\rangle$ and $q_0 = |0\rangle^{\otimes n}$ (i.e. we use n qubits set to $|0\rangle$) this algorithm is actually able to tell **constant** and **balanced** functions apart.

Definition 4.1: Constant function

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be **constant** if

$$\exists b \in \{0,1\} \quad \forall x \in \{0,1\}^n \quad f(x) = b$$

The definition of constant Boolean function has nothing special, and balanced functions are exactly what the name suggests, i.e. half of the inputs output 0 and the other half output 1, which can be succintly expressed as follows.

Definition 4.2: Balanced function

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be **balanced** if it holds that

$$\sum_{x \in \mathbb{B}^n} f(x) = 2^{n-1}$$

We observe that a Boolean function can be neither constant nor balanced, so this decision problem is actually a **promise problem**: given a Boolean function f that is either constant or balanced — note that it cannot be both — decide if the function is constant or balanced. Indeed, we see that Deutsch's algorithm solved the same exact problem for $n = 2$: in fact, if $f(0) = f(1)$ it means that f is constant, otherwise the latter is balanced.

Moreover, this problem actually shows the power of quantum parallelism more evidently: with a classical computation, to solve this decision problem we would need at most

$$2^{n-1} + 1 = O(2^n)$$

queries to f, instead our quantum computation still only requires one evaluation of f to solve the problem.

Algorithm 4.2: Deutsch-Josza algorithm

Given a Boolean function f and $n+1$ qubits, the algorithm returns $|0\rangle^{\otimes n}$ if f is constant, $|1\rangle$ otherwise.

1: **function** DEUTSCHJOSZA(f, q_0, q_1)
2: $\quad q_1 \leftarrow X(q_1)$
3: $\quad q_0, q_1 \leftarrow (H^{\otimes n} \otimes H)(q_0, q_1)$
4: $\quad q_0, q_1 \leftarrow U_f(q_0, q_1)$
5: $\quad q_0 \leftarrow H^{\otimes n}(q_0)$
6: $\quad$ **return** measure(q_0)
7: **end function**

Note that in this algorithm q_0 are actually n qubits, thus q_0 is initially set to $|0\rangle^{\otimes n}$. Before proving the correctness of this general version of the algorithm, let us first take a look at the quantum circuit that defines it.

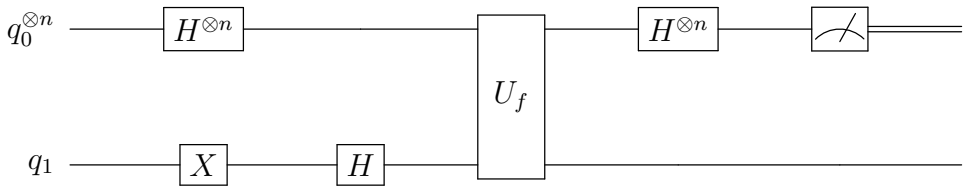


Figure 4.2: The quantum circuit for the Deutsch-Josza algorithm.

Proposition 4.2

For any $x \in \mathbb{B}^n$ it holds that

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{a \in \mathbb{B}^n} (-1)^{x \cdot a} |a\rangle$$

Proof. First, consider the following claim.

Claim: $\forall a \in \mathbb{B} \quad H|a\rangle = \frac{1}{\sqrt{2}} \sum_{b \in \mathbb{B}} (-1)^{a \cdot b} |b\rangle$.

Proof of the Claim. We observe that

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \sum_{b \in \mathbb{B}} (-1)^{0 \cdot b} |b\rangle$$

and analogously

$$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\sum_{b\in\mathbb{B}}(-1)^{1\cdot b}|b\rangle$$

□

In the rest of the proof we will denote with the $\cdot$ symbol the "canonical" scalar product, i.e.

$$\forall x, y \in \mathbb{B}^n \quad x \cdot y := \sum_{i=1}^{n} x_i y_i$$

Fix $x \in \mathbb{B}^n$; by the previous claim, we have that

$$\begin{aligned}
H^{\otimes n}|x\rangle &= \bigotimes_{i=1}^{n} H|x_i\rangle \\
&= \bigotimes_{i=1}^{n}\left(\frac{1}{\sqrt{2}}\sum_{b\in\mathbb{B}}(-1)^{x_i b}|b\rangle\right)|x_i\rangle \quad \text{(by the claim)} \\
&= \frac{1}{\sqrt{2}^n}\bigotimes_{i=1}^{n}(|0\rangle + (-1)^{x_i}|1\rangle) \\
&= \frac{1}{\sqrt{2}^n}\sum_{a\in\mathbb{B}^n}(-1)^{x\cdot a}|a\rangle
\end{aligned}$$

□

Finally, we are ready to prove the correctness of the algorithm.

$$
\begin{aligned}
& q_0 \otimes q_1 \\
=& \left|0\right\rangle^{\otimes n} \otimes \left|0\right\rangle \\
\xrightarrow{X(q_1)} & \left|0\right\rangle^{\otimes n} \otimes \left|1\right\rangle \\
\xrightarrow{H^{\otimes n}(q_0,q_1)} & H^{\otimes n} \left|0\right\rangle^{\otimes n} \otimes H \left|1\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{0^n \cdot a} \left|a\right\rangle \otimes \left|-\right\rangle && \text{(by Proposition 4.2)} \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left|a\right\rangle \otimes \left|-\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left(\left|a\right\rangle \otimes \left|-\right\rangle\right) \\
\xrightarrow{U_f(q_0,q_1)} & \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left((-1)^{f(a)} \left|a\right\rangle \otimes \left|-\right\rangle\right) && \text{(by Lemma 4.1)} \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} \left|a\right\rangle \otimes \left|-\right\rangle \\
\xrightarrow{H^{\otimes n}(q_0)} & H^{\otimes n} \left(\frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} \left|a\right\rangle\right) \otimes \left|-\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left((-1)^{f(a)} H^{\otimes n} \left|a\right\rangle\right) \otimes \left|-\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} \left(\frac{1}{\sqrt{2}^n} \sum_{b \in \mathbb{B}^n} (-1)^{a \cdot b} \left|b\right\rangle\right) \otimes \left|-\right\rangle && \text{(by Proposition 4.2)} \\
=& \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} \sum_{b \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b} \left|b\right\rangle \otimes \left|-\right\rangle \\
=& \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} \sum_{b \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b} \left|b\right\rangle \otimes \left|-\right\rangle \\
=& \sum_{b \in \mathbb{B}^n} \left(\frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b}\right) \left|b\right\rangle_0 \otimes \left|-\right\rangle_1
\end{aligned}
$$

Now note that this state describes the superposition of the system, but the next step of the algorithm will only measure q_0, therefore we can ignore $\left|-\right\rangle_1$ and just focus on the amplitudes of q_0. Then, by calling

$$\forall b \in \mathbb{B}^n \quad \alpha_b := \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b}$$

we can rewrite q_0 as follows

$$q_0 = \sum_{b \in \mathbb{B}^n} \alpha_b \left|b\right\rangle$$

Finally, since we want to determine the probability that q_0 collapses into the state $\left|0\right\rangle^{\otimes n}$

specifically, we can easily evaluate the associated amplitude of the latter, i.e.

$$\alpha_{0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)+a \cdot 0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)}$$

From this, we can easily conclude that:

- if f is constant, then

$$\alpha_{0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} = \frac{1}{2^n} \cdot 2^n \cdot (-1)^b = (-1)^b$$

  where $b \in \mathbb{B}$, meaning that it is guaranteed that q_0 will collapse to 0^n

- if f is balanced, then

$$\alpha_{0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}} (-1)^{f(a)} = \frac{1}{2^n} \cdot 0 = 0$$

  meaning that it is guaranteed that q_0 will *not* collapse to 0^n

5

Quantum Phase Estimation

Shor's algorithm is a quantum algorithm for finding the *prime factors of an integer*, developed in 1994 by the American mathematician Shor [Sho]. It is one of the few known quantum algorithms with compelling potential applications and strong evidence of **superpolynomial** speedup compared to best known classical (non-quantum) algorithm. It is one of the most famous algorithms in the field, and it sparked a lot of interest in recent years due to the implications of its theoretical speedup.

Shor's algorithm utilizes a lot of mathematical tools

In the previous chapter, we explored some of the foundational algorithms of quantum computing, which demonstrated how quantum superpositions can *outperform* classical approaches for specific problems. These algorithms, while conceptually simpler, help understanding the power and subtleties of quantum computation.

place this somewhere else anche change the intro

da finire

Building upon that foundation, we now turn our attention to a more sophisticated and central algorithm: **Quantum Phase Estimation (QPE)**. At its core, QPE allows us to determine the eigenvalues of a unitary operator with remarkable precision, a capability that lies at the heart of many quantum algorithms. Unlike the problems addressed by simpler algorithms, phase estimation tackles inherently more complex tasks and more broadly applicable, as we will see in the subsequent chapters.

The importance of QPE cannot be overstated. In this chapter, we will focus solely on QPE and how it works, while in the next chapters we will see how QPE is actually leveraged in the most widely known quantum algorithm, namely **Grover's algorithm** and **Shor's algorithm**.

5.1 A recap on the Discrete Fourier Transform

The **Discrete Fourier Transform (DFT)** is a very powerful tool used in many applications and probably most widely known for its use in signal processing. Indeed, the Fourier transform is used to transform a signal defined on the *time domain* into its equiv-

alent in the *frequency domain*. We will briefly explore the most important concepts that define the DFT to have a better understanding in order to the progress with its quantum counterpart.

First, what is the "time domain"? When we sample a signal, we usually sample such signal through some finite amount of time, and we discretize such time interval — this is why we consider the Discrete Fourier transform. What we are interested in is to transform such signal into the "frequency domain", which essentially means to understand what frequencies of sinusoids contribute to our signal and in which percentage. To do this, we will use some concepts of Linear Algebra. Say that the signal we sampled is discretized into N parts, thus our sample lives in the $\mathbb{R}^N$ space — it is nothing but a vector of N complex values which describe the signal at each timestep. When we say "time domain", what we mean is basically this exact vector, i.e. expressed in the canonical orthonormal basis

$$e_0 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad e_1 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \dots \quad e_{N-1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

(we will start counting at 0 for convenience sake). In other words, if our signal is defined by some vector

$$\mathbf{x} = \begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_{N-1} \end{pmatrix}$$

it holds that

$$\mathbf{x} = \sum_{n=0}^{N-1} t_n e_n$$

Then, what is the "frequency domain"? We need some preliminary observations to describe it. Our goal is to take into account the frequencies that define our signal, therefore we need to move into a space in which frequencies are "first-class citizens". The idea is to use each possible sinusoid by varying the frequency, add up their contributions, and weight the latter in the precise way that allows us to reconstruct our original signal. Let us fix a component $n \in [0, N-1]$, and first consider only cosinusoids, for instance

$$\cos(2\pi \cdot 0 \cdot n) \quad \cos(2\pi \cdot 1 \cdot n) \quad \cos(2\pi \cdot 2 \cdot n) \quad \dots \quad \cos(2\pi \cdot (N-1) \cdot n)$$

Basically, we are trying to construct a basis built on each possible frequency f by obtaining a basis vector $\cos(2\pi \cdot f \cdot n)$. Moreover, we will consider $\cos(2\pi mn/N)$ for $m \in [0, N-1]$ since it can be shown that the number of possible frequencies that can be represented on a N-sized time window is exactly N, and to scale the frequency accordingly we just need to divide m over N, the size of the sample.

Is this enough to reach our goal? Can we describe any signal in this way? Well, we observe that m is ranging from 0 to $N-1$, but $\cos(-\theta) = \cos(\theta)$, which implies that

$$m > \frac{N}{2} \implies \cos(2\pi mn/N) = \cos(2\pi(N-m)n/N)$$

In other words, basically half of our basis vectors add no information at all. Indeed, the span of the vectors we chose has size

$$\dim\left(\text{span}\left(\{\cos(2\pi\cdot m\cdot n/N)\}_{m=1}^{N-1}\right)\right)=\begin{cases}\frac{N}{2}+1 & N \text{ is even}\\ \frac{N+1}{2} & N \text{ is odd}\end{cases}$$

where the last added 1 comes from the fact that when $m=0$ we generate $\cos(0)=1$ which is really linearly independent from the others cosines. This suggests that cosinusoids alone are not enough to describe our space.

Hence, the most natural thing that we can do is add the contributions of sinusoids as well. A geometric interpretation of the fact that cosinusoids are not enough is that sinusoids are just phase-shifted cosines, but the shift in phase is not captured by changing the frequencies of the cosines alone. We need the contributions of some "altered" cosinusoids — in terms of phases — to get an actual basis and be able to represent any possible vector. Hence, let's consider additional N sinusoids

$$\sin(2\pi\cdot 0\cdot n)\quad \sin(2\pi\cdot 1\cdot n)\quad \sin(2\pi\cdot 2\cdot n)\quad \dots\quad \sin(2\pi\cdot(N-1)\cdot n)$$

Do we get a base of size more than N then? We observe that $\sin(\theta)=-\sin(-\theta)$, therefore we have that

$$m>\frac{N}{2}\implies \sin(2\pi mn/N)=-\sin(2\pi(N-m)n/N)$$

which again it implies that half of these sinusoids add no useful information. However, in this case we also have the fact that

$$\sin(2\pi\cdot 0\cdot n/M)=\sin(0)=0$$

and for N even it also happens that

$$\sin(2\pi\cdot(N/2)\cdot n/N)=\sin(\pi n)=0$$

which means that these two sinusoid cannot be considered because they are the 0 vector of this space. Therefore, we get that

$$\dim\left(\text{span}\left(\{\sin(2\pi\cdot m\cdot n/N)\}_{m=1}^{N-1}\right)\right)=\begin{cases}\frac{N}{2}-1 & N \text{ is even}\\ \frac{N-1}{2} & N \text{ is odd}\end{cases}$$

Finally, this means that putting all these cosines and sinusoids together we form a base for a space that has size

$$\begin{cases}\frac{N}{2}+1+\frac{N}{2}-1=N & N \text{ is even}\\ \frac{N+1}{2}+\frac{N-1}{2}=N & N \text{ is odd}\end{cases}=N$$

Hence, we can fully describe $\mathbb{R}^N$, namely for any vector $\mathbf{x}\in\mathbb{R}^N$ we have that

$$\forall n\in[0,N-1]\quad \mathbf{x}(n)=\sum_{m=0}^{N-1}\alpha_m\cos(2\pi mn/N)+\sum_{m=0}^{N-1}\beta_m\sin(2\pi mn/N)$$

by describing the vector component-wise.

Furthermore, we observe that this basis is actually orthogonal.

prove it, boring

Now, the last thing that we need to do is move into the complex space $\mathbb{C}^N$. Thankfully, we can leverage the very powerful Euler's formula

$$e^{i\theta} = \cos\theta + i\sin\theta$$

to immediately some the contributions of cosinusoids and sinusoids into a single value $e^{2\pi imn/N}$. In other words, for any vector $\mathbf{x} \in \mathbb{C}^N$ it holds that

$$\forall n \in [0, N-1] \quad \mathbf{x}(n) = \sum_{m=0}^{N-1} \gamma_m e^{2\pi imn/N}$$

The basis of this space is thus

$$e^{2\pi i \cdot 0 \cdot x/N} \quad e^{2\pi i \cdot 1 \cdot x/N} \quad \dots \quad e^{2\pi i \cdot (N-1) \cdot n/N}$$

We observe that this basis is orthogonal as well, but it's not orthonormal in fact Hence, because the norm of each vector is

prove it, boring too

$$\begin{aligned}
\sqrt{\langle e^{2\pi imn/N} | e^{2\pi imn/N} \rangle} &= \sqrt{\sum_{n=0}^{N-1} e^{2\pi imn/N} \cdot \overline{e^{2\pi imn/N}}} \\
&= \sqrt{\sum_{n=0}^{N-1} |e^{2\pi imn/N}|^2} \\
&= \sqrt{\sum_{n=0}^{N-1} 1} \\
&= \sqrt{N}
\end{aligned}$$

we usually consider the same base but scaled by a factor of $\frac{1}{\sqrt{N}}$, i.e.

$$\forall n \in [0, N-1] \quad \mathbf{x}(n) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \delta_m e^{2\pi imn/N}$$

We define δ_m to be the m-th component of the **discrete Fourier transform** of $\mathbf{x}$, and the operation that reconstructs $\mathbf{x}(n)$ is called **inverse discrete Fourier transform**. In other words, the DFT of a vector computes the projection of said vector into the space $\mathbb{C}^N$ through the orthonormal basis

$$\left\{ \tfrac{1}{\sqrt{N}} e^{2\pi imn/N} \right\}_{m=0}^{N-1}$$

Indeed, we can compute the DFT by simply applying the matrix that performs the **change of basis**, but we will see the details of this idea in the next section.

5.2 The Quantum Fourier Transform

Now that we have a general understanding of what the DFT is, let's see how the matrix operation is actually defined.

Definition 5.1: n-th root of unity

Given a commutative ring R, and a natural number $n \in N$, an element $\omega \in R$ is called **n-th root of unity** if $\omega^n = 1$.

For instance, $-i$ is a 4-th root of unity, in fact $(-i)^4 = 1$. We observe that, since multiplication in the complex plane is a rotation, the $n-$-th roots of 1 evenly divide the unit circle as shown below:

TODO drawing

Now, among all possible n-th roots of unity we are going to provide a more specific definition. Let the **order of an n-th root of unity** ω be the smallest power d such that $\omega^d = 1$. For instance, $(-1)^4 = 1$ indeed -1 is a 4-th root of unity, however its order is 2 since $(-1)^2 = 1$ and $2 < 4$.

Definition 5.2: Principal n-th root of unity

Given an n-th root of unity ω, we say that ω is **principal** if and only if $\omega \neq 1$ and the order of ω is n.

In other words, the principal n-th root of unity is the first root (after 1) that we encounter on the unit circle. Indeed, thanks to Euler's formula we usually define the principal n-th principal root of unity as

$$\omega := e^{2\pi i/n}$$

since $\frac{2\pi}{n}$ is the n-th slice of the unit circle. Furthermore, the second condition of the definition is usually not provided in terms of order of the root, and it's written as follows

$$\forall p \in [n-1] \quad \sum_{j=0}^{n-1} \omega^{jp} = 0$$

Aside from the geometric interpretation of this sum, this is a finite complex geometric series with common ratio ω^p, which implies that

- if $\omega^p = 1$, then every term is $(\omega^p)^j = 1^j = 1$ for any j, meaning that the sum is equal to
$$\sum_{j=0}^{n-1} \omega^{jp} = \sum_{j=0}^{n-1} 1 = n$$
- if $\omega^p \neq 1$, then we know that
$$\sum_{j=0}^{n-1} \omega^{jp} = \frac{1 - (\omega^p)^n}{1 - \omega^p}$$

and this ratio is equal to 0 if and only if

$$1 - (\omega^p)^n = 0 \iff \omega^{pn} = 1$$

Therefore, we have that this sum is equal to 0 if and only if $\omega^p \neq 1$ and $\omega^{pn} = 1$. However, we observe that if $\omega^p = 1$, since $p \in [n-1]$ this would imply that the order of ω is less than n, meaning that ω was not a principal root.

Now that we presented roots of unity we can finally present how the DFT matrix is usually defined.

Definition 5.3: Discrete Fourier Transform

Given a commutative ring R of dimension N, the **Discrete Fourier Transform (DFT)** matrix in R is defined as follows:

$$\forall j, k \in [0, N-1] \quad \mathrm{DFT}_{jk} = \omega^{-jk}$$

where ω is the principal N-th root of unity.

Therefore, in general the DFT matrix looks like this:

$$\mathrm{DFT} := \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(N-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \cdots & \omega^{-(N-1)^2} \end{pmatrix}$$

Moreover, we usually define the DFT matrix by setting $\omega = e^{\frac{2\pi i}{n}}$, thus getting

$$\forall j, k \in [0, N-1] \quad \mathrm{DFT}_{jk} = e^{-2\pi i jk/N}$$

Indeed, having presented the intuition behing the DFT beforehand, it's now fairly obvious the reason why the DFT matrix looks like this: its just the matrix that performs the change of basis, and since the basis of the DFT space can be defined in terms of n-th roots of unity, this is the matrix we get.

Furthermore, not surprisingly the DFT matrix is invertible, in fact we have that

$$\forall j, k \in [0, N-1] \quad \mathrm{IDFT}_{jk} = \mathrm{DFT}_{jk}^{-1} = \frac{1}{N}\omega^{jk} = \frac{1}{N}e^{+2\pi i jk/N}$$

and this matrix takes the name of **Inverse Discrete Fourier Transform (IDFT)**.

Then, can't we just use the DFT matrix in quantum computations whenever we need it and be done? The problem is that the DFT matrix is clearly not unitary: if we look closely, we see that the columns of the DFT matrix are only orthogonal and not orthonormal. Thus, to solve this problem, we need to normalize the column vectors.

Definition 5.4: Quantum Fourier Transform

Given a Hilbert space $\mathcal{H}$ of size N, the **Quantum Fourier Transform (QFT)** matrix in $\mathcal{H}$ is defined as follows:

$$\forall j, k \in [0, N-1] \quad \mathrm{QFT}_{jk} = \frac{1}{\sqrt{N}} \omega^{jk}$$

where ω is the principal N-th root of unity.

We observe that the QFT matrix has positive exponents, which is just a convention employed in quantum computing. Therefore, we get the following matrix:

$$\mathrm{QFT} := \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{pmatrix}$$

In particular, for any $|x\rangle$ basis state, the quantum Fourier transform can also be expressed as follows:

$$\mathrm{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{+2\pi i x k/N} \,|k\rangle$$

which directly implies that we can rewrite the QFT matrix as follows

$$\mathrm{QFT} = \sum_{j=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} \,|k\rangle \right) \langle j|$$

thanks to Proposition 2.12.

Proposition 5.1

The QFT operator is a unitary, and in particular

$$\mathrm{QFT}^\dagger = \sum_{j=0}^{N-1} |j\rangle \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-2\pi i j k/N} \,\langle k| \right)$$

Proof. We will leave the proof of correctness of $\mathrm{QFT}^\dagger$ as an exercise, and we will focus on proving that the QFT operator is indeed unitary. To prove it, is satisfies to show that

$$\mathrm{QFT}^\dagger \,\mathrm{QFT} = \mathrm{QFT}\,\mathrm{QFT}^\dagger = I$$

We are going to prove the leftmost product first.

$$\begin{aligned}
\mathrm{QFT}^\dagger\ \mathrm{QFT} &= \sum_{j=0}^{N-1} |j\rangle \left(\frac{1}{\sqrt{N}} \sum_{r=0}^{N-1} e^{-2\pi ijr/N} \langle r|\right) \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} e^{2\pi isk/N} |s\rangle\right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{-2\pi ijr/N} \langle r|\right) \left(\sum_{s=0}^{N-1} e^{2\pi isk/N} |s\rangle\right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r,s=0}^{N-1} e^{2\pi i(-jr+ks)/N} \langle r|s\rangle\right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi ir(k-j)/N}\right) \langle k| && (\langle r|s\rangle = \delta_{rs}) \\
&= \frac{1}{N} \sum_{\substack{j,k=0: \\ j=k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} 1\right) \langle k| + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi ir(k-j)/N}\right) \langle k| \\
&= \frac{1}{N} \sum_{j=0}^{N-1} |j\rangle \langle j| + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi ir(k-j)/N}\right) \langle k| \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi ir(k-j)/N}\right) \langle k| && (\text{resolution of } I) \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} \left(e^{2\pi i(k-j)/N}\right)^r\right) \langle k| && (\text{resolution of } I)
\end{aligned}$$

Now, we observe that the inner sum is a geometric series of ratio $e^{2\pi i(k-j)/N}$, and we can use its known result to simplify the calculations. However, we observe that this is true only if the ratio of the series is not equal to 1, but due to the way we split the sums we already took care of all the possible terms that could be equal to 1 so we do not need to make additional assumptions. Thus, we have that

$$\begin{aligned}
\mathrm{QFT}^\dagger\ \mathrm{QFT} &= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} \left(e^{2\pi i(k-j)/N}\right)^r\right) \langle k| \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\frac{1 - \left(e^{2\pi i(k-j)/N}\right)^N}{1 - e^{2\pi i(k-j)/N}}\right) \langle k| \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\frac{1 - e^{2\pi i(k-j)}}{1 - e^{2\pi i(k-j)/N}}\right) \langle k|
\end{aligned}$$

Lastly, since $k - j$ is an integer for any $j, k \in [0, N-1]$ — and in particular whenever $j \neq k$ — it holds that

$$e^{2\pi i(k-j)} = \cos(2\pi i(k-j)) + i \sin(2\pi i(k-j)) = 1$$

Therefore, the numerator is always equal to $1 - 1 = 0$, so the sum vanishes. Finally, we can conclude that

$$\text{QFT}^\dagger \text{ QFT} = I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j \neq k}}^{N-1} |j\rangle \left(\frac{1-1}{1 - e^{2\pi i(k-j)/N}} \right) \langle k| = I$$

TODO □

prove the second product

5.2.1 The quantum circuit

The previous section showed how the QFT matrix operator is defined, why its unitary and its similarities with the DFT. It's finally time to address the most important question: how do we turn the QFT matrix into a quantum circuit? Assume that $N = 2^n$, and consider the definition we provided

$$\text{QFT} |x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{+2\pi i x k/N} |k\rangle$$

The problem arises because we are using fractional exponents, however $x \in \mathbb{B}^n$. To solve this issue, we need to take a closer look at the exponent. Since $x \in \mathbb{B}^n$, there exist some bits $x_1, \ldots, x_n \in \mathbb{B}$ such that x can be written as

$$x = x_1 \cdot 2^{n-1} + \ldots + x_n \cdot 2^0$$

Since $k \in [0, N-1]$, we can also define such $k_1, \ldots, k_n \in \mathbb{B}$, and we can see what happens when we evaluate xk/N:

$$\begin{aligned} kx/N &= \frac{(k_1 \cdot 2^{n-1} + \ldots + k_n \cdot 2^0) \cdot (x_1 2^{n-1} + \ldots + x_n 2^0)}{2^n} \\ &= \left(k_1 \cdot 2^{n-1} + \ldots + k_n \cdot 2^0\right) \cdot \left(x_1 \cdot \frac{2^{n-1}}{2^n} + \ldots + x_n \cdot \frac{2^0}{2^n} \right) \\ &= \left(k_1 \cdot 2^{n-1} + \ldots + k_n \cdot 2^0\right) \cdot 0.x_1 \ldots x_n \\ &= k_1 \cdot 2^{n-1} \cdot 0.x_1 \ldots x_n + \ldots + k_n \cdot 2^0 \cdot 0.x_1 \ldots x_n \end{aligned}$$

This is true because we recall that

$$0.x_1 \ldots x_n = x_1 \cdot 2^{-1} + \ldots + x_n \cdot 2^{-n}$$

and since we are computing x/N and $N = 2^n$ what we get is just a "displacement of the decimal point", but in binary. Now, we observe that

$$\forall \ell \in [n] \quad 2^{n-\ell} \cdot 0.x_1 \ldots x_n = x_1 \ldots x_{n-\ell}.x_{n-\ell+1} \ldots x_n$$

since again, we are just moving the decimal point $n - \ell$ times to the right. Therefore, we get that

$$\begin{aligned} kx/N &= k_1 \cdot x_1 \ldots x_{n-1}.x_n + \ldots + k_n 0.x_1 \ldots x_n \\ &= k_n \cdot 0.x_1 \ldots x_n + \sum_{h=1}^{n-1} k_h \cdot x_1 \ldots x_{n-h}.x_{n-h+1} \ldots x_n \end{aligned}$$

we observe that we need to put $0.x_1 \dots x_n$ just because we cannot include it in the same sum. Now, let's see what happens when we put this term at the exponent:

$$
\begin{aligned}
& e^{2\pi ikx/N} \\
&= \exp(2\pi ikx/N) \\
&= \exp\left(2\pi i\left(k_n 0.x_1 \dots x_n + \sum_{h=1}^{n-1} k_h \cdot x_1 \dots x_{n-h}.x_{n-h+1} \dots x_n\right)\right) \\
&= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot x_1 \dots x_{n-h}.x_{n-h+1} \dots x_n\right) \\
&= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot (x_1 \dots x_{n-h}) + 2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right) \\
&= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot (x_1 \dots x_{n-h})\right) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right) \\
&= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot 1 \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right)
\end{aligned}
$$

The last semplitifcation derives immediately from the fact that $e^{2\pi i\theta} = 1$ when θ is an integer, as we mentioned in the previous section, so that is what we are left with — since $x_1 \dots x_{n-h}$ is definitely an integer. We can proceed as follows:

$$
\begin{aligned}
& \mathrm{QFT}\,|x\rangle \\
=& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right)\right) |k\rangle \\
=& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{m=2}^{n} k_{n-m+1} \cdot 0.x_m \dots x_n\right)\right) |k\rangle \\
=& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(2\pi i \sum_{m=1}^{n} k_{n-m+1} \cdot 0.x_m \dots x_n\right) |k\rangle \\
=& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \prod_{m=1}^{n} \exp\left(2\pi i k_{n-m+1} \cdot 0.x_m \dots x_n\right) |k\rangle
\end{aligned}
$$

If we look closely, this sum of products as an interesting property:

- $m = 1 \implies k_{n-m+1} = k_{n-1+1} = k_n$
- $m = n \implies k_{n-m+1} = k_{n-n+1} = k_1$

implying that it can be rewritten as follows

$$
\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(e^{2\pi i k_n \cdot 0.x_1 \dots x_n} \cdot \dots \cdot e^{2\pi i k_1 \cdot 0.x_n}\right)
$$

In other words the bits of k are *indexing* which exponentials to use in the k-th coefficient of the sum. This means that we can rewrite the last step as follows:

$$\mathrm{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \bigotimes_{m=1}^{n} \left(|0\rangle + e^{2\pi i 0.x_m \ldots x_n}\,|1\rangle\right)$$

Finally, the QFT written in this form can be actually implemented in a quantum circuit.

da finire sonostufo

5.3 Quantum Phase Estimation

Let's see a very useful application of the QFT introduced by Kitaev [Kit95] in 1995. By Theorem 2.2 we know that the eigenvalues of a unitary operator are complex values of modulus 1. This means that any eigenvalue of a unitary operator can be ewritten as $e^{2\pi i\varphi}$ for some real value $\varphi \in [0, 1]$ — φ being the *phase* of the complex value, i.e. the angle w.r.t. the unitary circumference.

Given a unitary operator U, and an eigenvalue $\lambda = e^{2\pi i\varphi}$, can we determine φ? Let u be an eigenvector associated to the unknown eigenvalue λ, i.e.

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

Now, let U^{2^k} be the quantum gate that applies the U operator 2^k times repeatedly, and define the following operator

$$\text{C-}U^{2^k} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U^{2^k}$$

This operator is the **controlled** version of U^{2^k}, and it can easily be showed by plugging $|0\rangle$ and $|1\rangle$ as first argument — alongside with some other quantum state $|\psi\rangle$

$$\begin{aligned}
\text{C-}U^{2^k}(|0\rangle \otimes |\psi\rangle) &= (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U^{2^k})(|0\rangle \otimes |\psi\rangle) \\
&= (|0\rangle\langle 0| \otimes I)\,|0\rangle \otimes |\psi\rangle + (|1\rangle\langle 1| \otimes U^{2^k})\,|0\rangle \otimes |\psi\rangle \\
&= |0\rangle \otimes |\psi\rangle
\end{aligned}$$

Hence, if the first input is $|0\rangle$ the state $|\psi\rangle$ is unchanged, otherwise if the former is $|1\rangle$ we get that we actually apply U^{2^k} to $|\psi\rangle$:

$$\begin{aligned}
\text{C-}U^{2^k}(|1\rangle \otimes |\psi\rangle) &= (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U^{2^k})(|1\rangle \otimes |\psi\rangle) \\
&= (|0\rangle\langle 0| \otimes I)\,|1\rangle \otimes |\psi\rangle + (|1\rangle\langle 1| \otimes U^{2^k})\,|0\rangle \otimes |\psi\rangle \\
&= |1\rangle \otimes U^{2^k}\,|\psi\rangle
\end{aligned}$$

Indeed, in general if we have an operator V, its controlled version can be easily built in this manner:

$$\text{C-}V = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes V$$

However, this controlled gate can also be "abused": what happens if the control bit is a superposition of states? In particular, what happens when the control is set to

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

meaning that $|0\rangle$ and $|1\rangle$ are equally probable?

$$\begin{aligned}\text{C-}U^{2^k}(|+\rangle \otimes |\psi\rangle) &= \frac{1}{\sqrt{2}}\text{C-}U^{2^k}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\psi\rangle) \\ &= \frac{1}{\sqrt{2}}(\text{C-}U^{2^k}(|0\rangle \otimes |\psi\rangle) + \text{C-}U^{2^k}(|1\rangle \otimes |\psi\rangle)) \\ &= \frac{1}{\sqrt{2}}((|0\rangle \otimes |\psi\rangle) + (|1\rangle \otimes U^{2^k}|\psi\rangle))\end{aligned}$$

As we would expect, what happens is that the two outcomes we previously described are now equally likely, but this gets interesting if we carefully choose the target qubit. Instead of any possible $|\psi\rangle$, let's pick $|u\rangle$, the eigenvector presented at the beginning of the discussion. First, we observe that

$$U|u\rangle = e^{2\pi i\varphi}|u\rangle \implies U^{2^k}|u\rangle = \left(e^{2\pi i\varphi}\right)^{2^k}|u\rangle = e^{2\pi i 2^k\varphi}|u\rangle$$

Indeed, intuitively we are just performing the rotation that U performs on $|u\rangle$ exactly 2^k times. This means that

$$\begin{aligned}\text{C-}U^{2^k}(|+\rangle \otimes |u\rangle) &= \frac{1}{\sqrt{2}}((|0\rangle \otimes |u\rangle) + (|1\rangle \otimes U^{2^k}|u\rangle)) \\ &= \frac{1}{\sqrt{2}}((|0\rangle \otimes |u\rangle) + (|1\rangle \otimes e^{2\pi i 2^k\varphi}|u\rangle)) \\ &= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 2^k\varphi}|1\rangle) \otimes |u\rangle\end{aligned}$$

Notice what happened here: $|0\rangle$ and $|1\rangle$ are parts of the first input, and $|u\rangle$ is the second input. In other words, by putting $|u\rangle$ as target input of the controlled gate, the target is **unchanged**, and the effect is seen on the control. We observe that this could not have been done with any other state $|\psi\rangle$ because the trick works precisely because U becomes a scalar $e^{2\pi i\varphi}$ when applied to $|u\rangle$, so it can be grouped as shown. The following quantum circuit computes the **Quantum Phase Estimation (QPE)** algorithm, and it uses precisely this trick.

Algorithm 5.1: Quantum Phase Estimation algorithm

Given a unitary operator U, and an eigenvector $|u\rangle$ of U, the algorithm returns the phase φ to which $|u\rangle$ is associated to.

1: **function** QuantumPhaseEstimation(U, $|u\rangle$)
2: $q_t \leftarrow |u\rangle$ ▷ q_t is the $(t+1)$-th register
3: **for** $k \in [0, t-1]$ **do**
4: $q_k \leftarrow H(q_k)$
5: $q_k, q_t \leftarrow U^{2^k}(q_k, q_t)$
6: **end for**
7: $q_0, \ldots, q_{t-1} \leftarrow \text{QFT}^\dagger(q_0, \ldots, q_{t-1})$
8: **return** measure$(q_0, \ldots, q_{t-1})$
9: **end function**

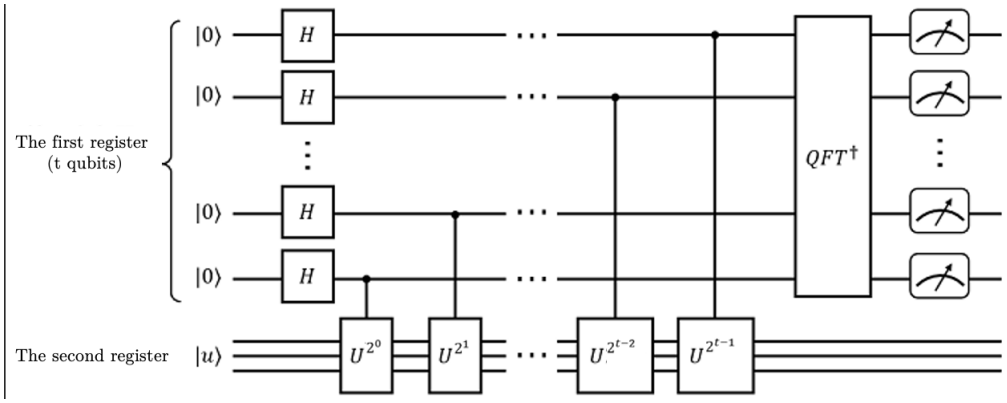


Figure 5.1: The Quantum Phase Estimation circuit.

The circuit is composed of the following elements:

- the first are $t > 0$ qubits $|0^t\rangle$ to which we apply $H^{\otimes t}$ at the beginning of the algorithm, which means now each state is set to $|+\rangle$
- the $(t+1)$-th register is a qubit set to $|u\rangle$ — we are not going to cover the practical details about how to set a qubit to this precise quantum state
- these superpositions are used as control to the lower part of the circuit, which conditionally computes U^{2^k} for each $k \in [0, t-1]$
- at the end we compute $\text{QFT}^\dagger$ to the first t registers

The interesting part of this circuit is that the last register remains unchanged throughout the whole computation, set precisely to $|u\rangle$, for the reason we described earlier. In fact, what happens is that what we are actually changing are the first t qubits which will be chaned according to the k-th controlled gate we are using. Therefore, if we compute what happens through the QPE circuit we get the following — we will refer to the first t

registers as $q_0, \ldots, q_{t-1}$:

$$
\begin{aligned}
&|q_0 \cdots q_{t-1}\rangle \otimes |u\rangle \\
=&|0\rangle^{\otimes t} \otimes |u\rangle \\
\xrightarrow{H(q_0)}&H(|0\rangle_0) \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
=&|+\rangle_0 \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
=&|+\rangle_0 \otimes |u\rangle \otimes |0\rangle^{\otimes t-1} \\
\xrightarrow{\text{C-}U^{2^0}(q_0,|u\rangle)}&\text{C-}U^{2^0}(|+\rangle_0 \otimes |u\rangle) \otimes |0\rangle^{\otimes t-1} \\
&\frac{1}{\sqrt{2}}(|0\rangle_0 + e^{2\pi i 2^0 \varphi}|1\rangle_0) \otimes |u\rangle \otimes |0\rangle^{\otimes t-1} \\
=&\frac{1}{\sqrt{2}}(|0\rangle_0 + e^{2\pi i 2^0 \varphi}|1\rangle_0) \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
=&\frac{1}{\sqrt{2}}(|0\rangle_0 \otimes |0\rangle^{\otimes t-1} + e^{2\pi i 2^0 \varphi}|1\rangle_0 \otimes |0\rangle^{\otimes t-1}) \otimes |u\rangle \\
\xrightarrow{H(q_1)}&\frac{1}{\sqrt{2}}(|0\rangle_0 \otimes H(|0\rangle_{q_1}) \otimes |0\rangle^{\otimes t-2} + e^{2\pi i 2^0 \varphi}|1\rangle_0 \otimes H(q_1) \otimes |0\rangle^{\otimes t-2}) \otimes |u\rangle \\
=&\frac{1}{2}\left(|00\rangle_{01} + |01\rangle_{01} + e^{2\pi i 2^0 \varphi}|10\rangle_{01} + e^{2\pi i 2^0 \varphi}|11\rangle_{01}\right) \otimes |0\rangle^{\otimes t-2} \otimes |u\rangle \\
\xrightarrow{\text{C-}U^{2^1}(q_1,|u\rangle)}&\frac{1}{2}\left(|00\rangle_{01} + e^{2\pi i 2^1 \varphi}|01\rangle_{01} + e^{2\pi i 2^0 \varphi}|10\rangle_{01} + e^{2\pi i (2^0+2^1) \varphi}|11\rangle_{01}\right) \otimes |0\rangle^{\otimes t-2} \otimes |u\rangle \\
\xrightarrow{\cdots}&\ldots \\
=&\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}\prod_{m=1}^{t} e^{2\pi i k_{t-m+1} \cdot 2^{m-1} \varphi}|k\rangle \otimes |u\rangle \\
=&\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}\left(|0\rangle + e^{2\pi i 2^{m-1} \varphi}|1\rangle\right) \otimes |u\rangle
\end{aligned}
$$

Furthermore, we obseve that $\varphi \in [0, 1)$, because as we already metioned there is no need to consider the integral part of the phases. Thus, suppose that φ can be written through t bits $\varphi_1, \ldots, \varphi_t \in \mathbb{B}$ such that

$$\varphi = 0.\varphi_1 \ldots \varphi_t$$

This implies that

$$
\begin{aligned}
2^{m-1} \cdot \varphi &= 2^{m-1} \cdot 0.\varphi_1 \ldots \varphi_t \\
&= \varphi_1 \ldots \varphi_{m-1}.\varphi_m \ldots \varphi_t
\end{aligned}
$$

Thus, since this is the exponent of $e^{2\pi i}$, we already know that

$$\exp(2\pi i \varphi_1 \ldots \varphi_{m-1}.\varphi_m \ldots \varphi_t) = \exp(2\pi i 0.\varphi_m \ldots \varphi_t)$$

therefore the ternsor product can be rewritten as

$$\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}(|0\rangle + e^{2\pi i 0.\varphi_m \ldots \varphi_t}|1\rangle)$$

Very elegantly, this is exactly $\mathrm{QFT}_t |\varphi\rangle$ we computed in the previous section (only applied in a t-dimensional space). Finally, if we place the inverse QFT — namely $\mathrm{QFT}_t^\dagger$ — at the end of the circuit we retrieve $|x\rangle$ itself, which is also the entire state of the system at this point (without considering $|u\rangle$). This means that if we measure each register at the end of the circuit we retrieve the bits of $\varphi_i \in \mathbb{B}^t$ — i.e. the bits that describe φ — with probability 1. In the end, we recovered the phase φ of the eigenvalue associated to $|u\rangle$ with probability 1.

scrivere QPE come alg

As a final note, what if φ cannot be expressed in exactly t bits? It can be shown that φ can be estimated with n bits of precision through a QPE circuit composed of $n + \left\lceil \log(2 + \frac{1}{2\varepsilon}) \right\rceil$ qubits with a success probability at least $1 - \varepsilon$, for some $\varepsilon > 0$.

5.4 Variational Quantum Algorithms

While the QPE algorithm shows the theoretical power of quantum computing, it must be pointed out that it assumes access to **large** and **error-free** quantum machines, hardware that *does not yet exist*. In fact, today's quantum devices are known as **Noisy Intermediate-Scale Quantum (NISQ)** machines, which have a limited number of qubits, noisy gates, short coherence times, and can only run shallow circuits before errors overwhelm the computation. In this section we will present a fairly recent approach that allows to work with such noisy machines, called **Variational Quantum Algorithms (VQAs)**. They are a type of quantum algorithms that are designed to work on NISQ devices we have today. By combining short, parameterized quantum circuits with classical optimization, VQAs allow the quantum computer to handle tasks it performs best — such as exploring complex quantum states and measuring key properties — assisted by a classical computer that iteratively adjusts the circuit parameters. This hybrid method tolerates noise and imperfections, making it possible to obtain useful results even with limited, error-prone hardware.

VQAs are a leading method for achieving quantum advantage on today's NISQ devices. They work by combining shallow, parameterized quantum circuits — well-suited to noisy hardware — with classical optimization techniques that adjust those parameters. This hybrid quantum-classical strategy resembles **machine-learning** approaches like neural networks and helps manage NISQ constraints while avoiding the deep circuits required by fully fault-tolerant quantum algorithms.

VQAs have been explored for a wide range of quantum computing applications and may offer the best chance for near-term quantum advantage. Due to the inherent versatility of VQAs, there is a wide variety of different algorithmics structures with different levels of complexity. Nevertheless, most of VQAs share the same basic elements. A VQA begins with a clearly defined task and, if needed, relevant training data. The first step is to design a **cost function** C that captures what it means to solve the problem.

Next, it must be choosen an **ansatz**, a German term that can be translated to *approach* or *attempt*. In the context of physics and mathematics, an ansatz is kind of an initial estimate to the solution of the problem considered. For instance, given a set of experimental data that looks to be clustered about a line, a *linear ansatz* could be made to find the parameters of the line by a *least squares* curve fit. In other words, if experimental data

looks linear, we might choose a linear model

$$y = mx + b$$

because we believe that the solution lives somewhere in the "space of all straight lines". Thus, in the quantum context specifically, the ansatz is the **family of quantum states** we allow the VQA to explore

$$|\psi(\theta)\rangle = U(\theta)\,|\psi_0\rangle$$

for some initial state $|\psi_0\rangle$, and a set of parameters θ of the model — θ is not an angle here! This means that choosing the structure of the circuit *is* the ansatz itself. In fact, this is the reason why choosing an appropriate ansatz is one of the most important steps of the whole process, and there are a wide range of ansatze that are currently being studied in order to model different type of problems. Cerezo, Arrasmith, Babbush, et al. [CAB+20] compiled a detailed list of the best known ansatze depending on the application of the VQA.

The algorithm then trains these parameters through a hybrid quantum-classical optimization loop, seeking the values

$$\theta^* = \arg\min_{\theta} C(\theta)$$

that minimize the cost function chosen. The quantum computer then evaluates the cost, and a classical optimizer updates the parameters according to a preffered optimization strategy — classical **Stochastic Gradient Descent (SGD)** is usually employed but newer alternatives are being developed in recent years that are "gradient-free".

VQAs are highly flexible because they support task-oriented programming, making them suitable for nearly every major application envisioned for quantum computers. In fact, they are powerful enough to enable **universal quantum computation**. A quantum model is said to be **universal** if it can approximate any unitary operation on any number of qubits to arbitrary accuracy, only using some kind of restricted architecture. In some sense, it is the quantum analog of what being a *universal gate* means with classical gates.

5.4.1 Variable Quantum Eigensolver

The best-known application of VQAs is by far the so called **Variable Quantum Eigensolver (VQE)** since it was the very first proposed VQA, developed to provide a near-term application for VQAs.

First, let's describe the problem we are interested in solving. In chemistry and many other fields, we are often interested in finding the fundamental physical properties of the system we are considering. Interestingly, such properties can be directly evaluated from the eigenvalues of the **Hamiltonian**, which is an operator that we briefly introduced in Postulate 3.2. The Hamiltonian of a system represents its total energy — including kinetic energy, potential energy and the interactions between particles. This fact derives from the **time-independent Schrödinger equation (TISE)**:

$$H\,|\psi\rangle = E\,|\psi\rangle$$

which is used when looking at stationary states — states whose probability distributions don't change over time. In this formula, we have that

- H is the Hamiltonian
- E is an eigenvalue of H, which is the energy of $|\psi\rangle$
- $|\psi\rangle$ is an eigenvector associated to E, which is the stationary state we are considering

Each eigenvalue E of H tells us an allowed energy level of the system, and we are interested in the *lowest one*, which is called **ground-state energy**, often denoted as E_G. While higher eigenvalues describe **excited states** that govern how mulecules absorb and emit light, E_G describes a molecule's stability, preferred structure and how it participates in chemical reactions. This is the eigenvalue we are interested in, and it will be the main focus of our discussion. Determining the ground-state energy is essential for understanding spectroscopy, photochemistry, and materials used in solar cells or sensors. Moreover, beyond chemistry many problems in physics, materials science, and engineering reduce to eigenvalue calculations as well, such as the behavior of electrons in solids and the vibrations of mechanical structures.

What is the problem in finding the eigenvalues of H then? We observe that when we have a system of n particles we require a Hilbert space of dimension 2^n, which means that the size of the Hamiltonian is $2^n \times 2^n$. This exponential growth makes the search for eigenvalues **classically intractable**. Traditional approximate methods help but cannot achieve efficient, exact solutions for large systems. Luckily, quantum computing offers a promising path forward.

Actually, we already know a quantum algorithm that is able to yield the eigenvalue of a unitary operator: it's the QPE. Indeed, this algorithm *does* provide exponential speedups w.r.t. classical approaches, however it requires long coherent evolution and *extremely deep circuits*, making it impractical for near-term devices. Today, in the NISQ era, QPE is still not practical. However, in 2014 Peruzzo, McClean, Shadbolt, et al. [PMS+14] published a landmark paper, which proposed a VQA that avoids the long coherent runtimes required by QPE.

From a theoretical point of view, the idea is fairly straightforward: we want to find the state $|\psi^*\rangle$ that minimizes the eigenvalue, in order to find E_G. But first, let's consider a more general scenario: let A be any observable (i.e. a self-adjoint operator), and consider its expected value $\langle A\rangle$. In Proposition 3.1 we already proved that if the current state is $|\psi\rangle$ it holds that

$$\langle A\rangle = \langle\psi|A\psi\rangle$$

We are interested in finding the state $|\psi^*\rangle$ that will yield the lowest possible eigenvalue

of A after measuring the latter. We observe that

$$\begin{aligned}\langle\psi|A\psi\rangle &= \langle A\rangle \\ &= \mathbb{E}[A \mid |\psi\rangle] \\ &= \sum_{i=1}^{m} \lambda_i \Pr[A = \lambda_i \mid |\psi\rangle] \\ &\geq \min_{i\in[m]} \lambda_i \\ &= \lambda_{\min}\end{aligned}$$

with the equality holding if $|\psi\rangle = |\psi^*\rangle$. This means that by searching for

$$|\psi^*\rangle = \arg\min_{|\psi\rangle\in\mathcal{H}} \langle\psi|A\psi\rangle$$

we can immediately find $\lambda_{\min}$. It is easy to see that this idea is clearly applicable for finding E_G since H is self-adjoint, thus

$$|\psi^*\rangle = \arg\min_{|\psi\rangle\in\mathcal{H}} \langle\psi|H\psi\rangle \implies H|\psi^*\rangle = E_G|\psi^*\rangle$$

where E_G is the lowest eigenvalue of H, by definition. As a consequence, the idea of the VQE is to set the cost function as

$$C(\theta) = \langle\psi(\theta)|H\psi(\theta)\rangle$$

where in this case $|\psi(\theta)\rangle$ is the "trial" state that has to be defined as

$$|\psi(\theta)\rangle = U(\theta)|\psi_0\rangle$$

where $U(\theta)$ represents the ansatz of choice. Therefore, we are interested in finding the best parameters such that

$$\theta^* = \arg\min_{\theta} \langle\psi(\theta)|H\psi(\theta)\rangle$$

We observe that there is no universal ansatz for the VQE, as there are multiple variants that have been developed depending on the specific Hamiltonian of interest — different Hamiltonians may have different structures, entanglement patterns and symmetries. In conclusion, from a theoretical standpoint of view the VQE aims at minimizing $C(\theta)$ by computing the cost with a NISQ machine, and by optimizing θ through a classical computer.

In particular, we observe that the evaluation of the cost depends on the value of $\langle H\rangle$, and the best way we have to evaluate this quantity is by repeatedly measuring H. However, due to the size of H and the noise of the machines this step is often implemented with a better alternative. In fact, there is a very useful property of Hamiltonians that we can leverage: any Hamiltonian can be written as linear combination of tensor products of Pauli matrices

$$H = \sum_j h_j \bigotimes_{k=1}^{n} \sigma_{\alpha,jk}$$

where $\alpha \in \{x, y, z\}$ — we observe that the tensor product of n matrices 2×2 yields a matrix of size $2^n \times 2^n$, as expected. The original approach of the VQE only considers Hamiltonians that can be written as a sum of a polynomial number of terms. This property has a very practical implication: by linearity of expectations it holds that

$$\langle H \rangle = \sum_{j=1} h'_j \left\langle \bigotimes_{k=1}^{n} \sigma_{\alpha,jk} \right\rangle$$

which means that in order to compute the cost $C(\theta)$ we can instead evaluate the following

$$C(\theta) = \langle H \rangle = \langle \psi(\theta) | H \psi(\theta) \rangle = \sum_{j=1} h'_j \left\langle \bigotimes_{k=1}^{n} \sigma_{\alpha,jk} \right\rangle$$

This is much more feasible on current NISQ hardware, since the tensor producto of Pauli matrices can be treated as n different qubits

$$\left\langle \bigotimes_{k=1}^{n} \sigma_{\alpha,jk} \right\rangle = \langle \sigma_{\alpha,j1} \otimes \ldots \otimes \sigma_{\alpha,jn} \rangle$$

therefore measuring this expected value requires shallower circuits than computing the whole $\langle H \rangle$ which has size exponential size. We observe that this is an oversemplification and we are glossing over a lot of details, both from a theoretical and physical point of view, but such specifics are definitely outside the scope of this discussion as they would require an entire chapter on their own.

The last thing that we are going to mention about the VQE is its versatility. We showed how with this technique we are able to determine the ground-state energy of a Hamiltonian H, which is useful to know in various scientific research areas. However, in reality we observe that the method we outlined *does not depend* on the meaning of H. In fact, H really can be any self-adjoint operator that can be written as sum of "easier" operators to measure. Consider any cost function $S(x)$ that we seek to minimize; what we need is a matrix H_S whose eigenvalues are exactly the possible values that $S(x)$ can assume. Hence, by using the same trick that we used in the Born rule it suffices to consider a matrix whose Spectral decomposition is exacly

$$H_S = \sum_{y \in Y} S(y) \, |y\rangle \, \langle y|$$

Thus, by construction this immediately implies that

$$H_S \, |y\rangle = S(y) \, |y\rangle$$

Therefore, by executing the VQE on H_S we are actually finding the lowest value of $S(y)$:

$$\begin{aligned} \langle \psi | H_S \psi \rangle &= \langle H_S \rangle \\ &= \mathbb{E} \, H_S \mid |\psi\rangle \\ &= \sum_{y \in Y} S(y) \Pr[H_S = S(y) \mid |\psi\rangle] \\ &\geq \min_{y \in Y} S(y) \end{aligned}$$

In the end, we can simply run the VQE algorithm with

$$C(\theta) = \langle H_S \rangle = \langle \psi(\theta) | H_S \psi(\theta) \rangle$$

to get an approximation of the minimum value for our original cost function. This strategy is used to approximate various classically intractable computational problems outside chemistry and physics, such as

- **max-cut** problems, in which H_S encodes the cut costs
- **combinatorial optimization** problems, in which H_S encodes the constraints

The following procedure is a high-level overview of the implementation of this idea.

Algorithm 5.2: Optimization through VQE

Given a cost function S, and a fixed parameter N, the algorithm returns an approximation of $\min_{y \in Y} S(y)$.

---

function VQEOPTIMIZATION(S, N)
  $\theta \leftarrow \theta_0$
  while true do
    Generate circuit Q_θ from ansatz based on θ
    $c_\theta \leftarrow$ `[]`
    for $i \in [N]$ **do**
      $|\psi(\theta)\rangle \leftarrow Q_\theta |0^n\rangle$
      $c_i \leftarrow H_C$ measured on $|\psi(\theta)\rangle$
      c_θ`.append(`c_i`)`
    end for
    $\hat{c}_\theta \leftarrow \text{avg}\,(c_\theta)$
    if classical machine determines that $\hat{c}_\theta$ is ok **then**
      return θ
    end if
    $\theta \leftarrow \theta'$ ▷ θ' computed classically
  end while
end function

6

Shor's algorithm

TODO

TODO

introduction importance alg + connection w qpe

talk about RSA and wtf is shor's alg

6.1 Order-finding problem

The last ingredient that we need for Shor's algorithm is the **order-finding** algorithm. We recall the following property of number theory.

Proposition 6.1: Euclidean division

Given some $p \in \mathbb{N}_{>0}$, for any $n \in \mathbb{N}$ there exists unique integers $q, r \in \mathbb{Z}$ such that $0 \leq r < p$ and

$$n = p \cdot q + r$$

This is the usual division with remainder, which defines modular arithmetic, for instance

$$31 = 4 \cdot 7 + 3$$

Indeed, with equivalence classes we would write that

$$31 \equiv 3 \bmod 7$$

Definition 6.1: Order of a number

Given a number $N \in \mathbb{N}$, for any $x \in \mathbb{Z}$ the **order** of x modulo N is the least integer r such that

$$x^r \equiv 1 \bmod N$$

For instance, the order of 4 modulo 7 is 3, because

$$4^3 = 64 = 9 \cdot 7 + 1 \implies 4^3 \equiv 1 \bmod 7$$

We observe that throughout our discussion N is *not necessarily* a power of 2 as we used to assume in previous sections, however the symbol N is conventionally used in this context.

Given a number N, and a number $x < N$ that is coprime with N, can we compute its order modulo N? This is the so called **order-finding problem**, and currently there is no classical algorithm able to solve it in polynomial time. Let's see what quantum computation can achieve.

Consider the following unitary operator U_x that computes as follows:

$$\forall y \in \{0,1\}^k \quad U_x |y\rangle = \begin{cases} |xy \bmod N\rangle & y < N \\ |y\rangle & y \geq N \end{cases}$$

To derive the complete operator U_x we just follow Proposition 2.12:

$$\begin{aligned} U_x &= \sum_{y \in \{0,1\}^k} U_x |y\rangle \langle y| \\ &= \sum_{y<N} U_x |y\rangle \langle y| + \sum_{y \geq N} U_x |y\rangle \langle y| \\ &= \sum_{y<N} U_x |y\rangle \langle y| + \sum_{y \geq N} |y\rangle \langle y| \end{aligned}$$

We will not replace U_x inside the first sum for now in order to prove that U_x is unitary. But first, we need to present a result in number theory.

Theorem 6.1

If x and N are coprime, it holds that

$$\forall y, z \in \mathbb{Z} \quad xy \equiv xz \bmod N \iff y \equiv z \bmod N$$

Proposition 6.2

The operator U_x is unitary, and in particular

$$U_x^\dagger = \sum_{y<N} |y\rangle (U_x |y\rangle)^\dagger + \sum_{y \geq N} |y\rangle \langle y|$$

Proof. It is easy to prove the correctness of $U_x^\dagger$, so we are going to prove that U_x is unitary

directly.

$$\begin{aligned}
U_x^\dagger U_x &= \left(\sum_{y<N} |y\rangle \left(U_x |y\rangle\right)^\dagger + \sum_{y\geq N} |y\rangle\langle y|\right)\left(\sum_{z<N} U_x |z\rangle\langle z| + \sum_{z\geq N} |z\rangle\langle z|\right) \\
&= \left(\sum_{y<N} |y\rangle\langle U_x y| + \sum_{y\geq N} |y\rangle\langle y|\right)\left(\sum_{z<N} |U_x z\rangle\langle z| + \sum_{z\geq N} |z\rangle\langle z|\right) \\
&= \sum_{y,z<N} |y\rangle\langle U_x y|U_x z\rangle\langle z| + \sum_{\substack{y<N\\z\geq N}} |y\rangle\langle U_x y|z\rangle\langle z| + \sum_{\substack{y\geq N\\z<N}} |y\rangle\langle y|U_x z\rangle\langle z| + \sum_{y,z\geq N} |y\rangle\langle y|z\rangle\langle z|
\end{aligned}$$

Now note that if $z \geq N$ and $y < N$, by definition of U_x we have that $U_x |y\rangle = |xy \bmod N\rangle$, hence it will be a basis state among $|0\rangle, \ldots, |N-1\rangle$. Therefore, if $z > N$ we are guaranteed that $|z\rangle$ and $|U_x y\rangle$ are orthogonal, i.e. $\langle U_x y|z\rangle = 0$ — we recall that N is *not* the size of the space in this context, is just a composite number, indeed the size of the space we are considering is 2^k. Therefore, we have that

$$\sum_{\substack{y<N\\z\geq N}} |y\rangle\langle U_x y|z\rangle\langle z| = \sum_{\substack{y\geq N\\z<N}} |y\rangle\langle y|U_x z\rangle\langle z| = 0$$

so we conclude that

$$\begin{aligned}
U_x^\dagger U_x &= \sum_{y,z<N} |y\rangle\langle U_x y|U_x z\rangle\langle z| + \sum_{\substack{y<N\\z\geq N}} |y\rangle\langle U_x y|z\rangle\langle z| + \sum_{\substack{y\geq N\\z<N}} |y\rangle\langle y|U_x z\rangle\langle z| + \sum_{y,z\geq N} |y\rangle\langle y|z\rangle\langle z| \\
&= \sum_{y,z<N} |y\rangle\langle U_x y|U_x z\rangle\langle z| + \sum_{y,z\geq N} |y\rangle\langle y|z\rangle\langle z| \\
&= \sum_{\substack{y,z<N:\\ y\equiv z \bmod N}} |y\rangle\langle U_x y|U_x z\rangle\langle z| + \sum_{\substack{y,z<N:\\ y\not\equiv z \bmod N}} |y\rangle\langle U_x y|U_x z\rangle\langle z| + \sum_{y,z\geq N} |y\rangle\langle y|z\rangle\langle z|
\end{aligned}$$

To progress, we observe that when $y, z < N$ we get that

$$\langle U_x y|U_x z\rangle = \langle xy \bmod N|xz \bmod N\rangle = \begin{cases} 1 & xy \equiv xz \bmod N \\ 0 & \text{otherwise} \end{cases}$$

and by Theorem 6.1 we know that

$$xy \equiv xz \bmod N \iff y \equiv z \bmod N$$

thus getting

$$\begin{aligned}
U_x^\dagger U_x &= \sum_{\substack{y,z<N:\\ y\equiv z \bmod N}} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{\substack{y,z<N:\\ y\not\equiv z \bmod N}} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \langle y|z\rangle \langle z| \\
&= \sum_{\substack{y,z<N:\\ y\equiv z \bmod N}} |y\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \langle y|z\rangle \langle z| \\
&= \sum_{\substack{y,z<N:\\ y=z}} |y\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \delta_{yz} \langle z| \\
&= \sum_{y<N} |y\rangle \langle y| + \sum_{y\geq N} |y\rangle \langle y| \\
&= \sum_{y} |y\rangle \langle y| \\
&= I
\end{aligned}$$

Now, we need to prove the opposite product □ da fare

We did not provide any reason to why we defined the operator U_x as such, but before giving a geometrical intuition consider the following proposition.

Theorem 6.2

Given $N \in \mathbb{N}$, and $x \in [0, N-1]$, if r is the order of x modulo N, it holds that

$$\forall s \in [0, r-1] \quad |u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x^k \bmod N\rangle$$

is an eigenvector of U_x.

Proof. Fix $s \in [0, r-1]$; to prove that $|u_s\rangle$ is an eigenvector of U_x we need to show that there exists some phase φ_s such that

$$U_x |u_s\rangle = e^{2\pi i \varphi_s} |u_s\rangle$$

because of Theorem 2.2. Hence, we get that

$$
\begin{aligned}
U_x |u_s\rangle &= U_x \left(\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x^k \bmod N\rangle \right) \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} U_x |x^k \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x \left(x^k \bmod N\right) \bmod N\rangle && \left(x^k \bmod N < N\right) \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x^{k+1} \bmod N \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x^{k+1} \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \cdot \frac{e^{2\pi i s/r}}{e^{2\pi i s/r}} \cdot \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x^{k+1} \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \cdot \sum_{k=0}^{r-1} e^{-2\pi i s (k+1)/r} |x^{k+1} \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \cdot e^{2\pi i s/r} \left(\sum_{k=0}^{r-2} e^{-2\pi i s (k+1)/r} |x^{k+1} \bmod N\rangle + e^{-2\pi i s r/r} |x^r \bmod N\rangle \right) \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \left(\sum_{k=0}^{r-2} e^{-2\pi i s (k+1)/r} |x^{k+1} \bmod N\rangle + e^{-2\pi i s} |1\rangle \right) && \left(x^r \equiv 1 \bmod N\right) \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \left(\sum_{m=1}^{r-1} e^{-2\pi i s m/r} |x^m \bmod N\rangle + |1\rangle \right) \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \left(\sum_{m=0}^{r-1} e^{-2\pi i s m/r} |x^m \bmod N\rangle \right) && \left(|1\rangle = |x^0 \bmod N\rangle\right) \\
&= e^{2\pi i s/r} |u_s\rangle
\end{aligned}
$$

Therefore, we conclude that $\varphi_s = s/r$. □

TODO

intuizione geometrica

What happenns if the input of U_x is $|x^0\rangle$? By definition of our problem $x < N$, therefore

$$U_x |x^0\rangle = |x \cdot x^0 \bmod N\rangle = |x^1 \bmod N\rangle$$

Indeed in general it's easy to see that

$$U_x |x^k\rangle = |x^{k+1} \bmod N\rangle$$

In other words U_x is cycling through x's powers, which also implies that when $k = r - 1$ we get that

$$U_x |x^{r-1}\rangle = |x^r \bmod N\rangle = |1 \bmod N\rangle$$

since r is the order of x modulo N. Moreover, as we already know these are basis states so they are both normalized and orthogonal to each other, thus the powers x form an orthonormal base of the following space

$$\mathcal{H}_r = \text{span}\left(|x^k \bmod N\rangle \mid k \in [0, r-1]\right)$$

which is a restriction of the whole Hilbert space that has exactly r dimensions. Now look again at how the eigenvectors of U_x are defined:

$$\forall s \in [0, r-1] \quad |u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x^k \bmod N\rangle$$

This looks very similar to how we defined $\text{QFT}\,|x\rangle$:

$$\text{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k/N} |k\rangle$$

Indeed, it holds that $|u_s\rangle$ is basically $\text{QFT}\,|s\rangle$ but taken inside $\mathcal{H}_r$ — the only difference being the sign of the exponent, ndeed it technically holds that

$$|u_s\rangle = \text{QFT}_r\,|-s \bmod r\rangle$$

but as we said the sign of the exponent is just a convention either sign is found in literature, we only need to be consistent with the calculations.

non ho capito che c'entra però

The most interesting part is that, as proved in the last theorem, for any fixed $s \in [0, N-1]$ its associated phase φ_s is exactly s/r, thus if we knew how to prepare the last register of the QPE as $|u_s\rangle$ we could recover its phase, and maybe get closer to know r itself. However, we have two probems with this idea:

- there is really no easy or practical way to prepare the second register to some arbitrary state
- $|u_s\rangle$ actually depends on r itself, which actually is a very big problem on its own

Thankfully, the following property will solve both of these issues at the same time.

Proposition 6.3

Given $N \in \mathbb{N}$, and $x \in [0, N-1]$, if r is the order of x modulo N it holds that

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$$

Proof. Through some algebraic manipulation we see that

$$\begin{aligned}
\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle &= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}e^{-2\pi isk/r}\,|x^k \bmod N\rangle\right)\\
&= \frac{1}{r}\sum_{s,k=0}^{r-1}e^{-2\pi isk/r}\,|x^k \bmod N\rangle\\
&= \frac{1}{r}\sum_{s=0}^{r-1}e^{-2\pi is\cdot 0/r}\,|x^0 \bmod N\rangle + \frac{1}{r}\sum_{s=0,k=1}^{r-1}e^{-2\pi isk/r}\,|x^k \bmod N\rangle\\
&= \frac{1}{r}\sum_{s=0}^{r-1}|1\rangle + \frac{1}{r}\sum_{s=0,k=1}^{r-1}e^{-2\pi isk/r}\,|x^k \bmod N\rangle\\
&= \frac{1}{r}r\,|1\rangle + \frac{1}{r}\sum_{s=0,k=1}^{r-1}e^{-2\pi isk/r}\,|x^k \bmod N\rangle\\
&= |1\rangle + \frac{1}{r}\sum_{k=1}^{r-1}|x^k \bmod N\rangle\sum_{s=0}^{r-1}e^{-2\pi isk/r}\\
&= |1\rangle + \frac{1}{r}\sum_{k=1}^{r-1}|x^k \bmod N\rangle\sum_{s=0}^{r-1}\left(e^{-2\pi ik/r}\right)^s
\end{aligned}$$

As we did for the proof of Proposition 5.1, because of how we split the sums we know that $k \neq 0$ hence $e^{-2\pi ik/r} \neq 1$ because k/r is not an integer (since $k \in [1, r-1]$), therefore we get that

$$\begin{aligned}
\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle &= |1\rangle + \frac{1}{r}\sum_{k=1}^{r-1}|x^k \bmod N\rangle\,\frac{1-(e^{-2\pi ik/r})^r}{1-e^{-2\pi ik/r}}\\
&= |1\rangle + \frac{1}{r}\sum_{k=1}^{r-1}|x^k \bmod N\rangle\,\frac{1-e^{-2\pi ik}}{1-e^{-2\pi ik/r}}\\
&= |1\rangle + \frac{1}{r}\sum_{k=1}^{r-1}|x^k \bmod N\rangle\,\frac{1-1}{1-e^{-2\pi ik/r}}\\
&= |1\rangle
\end{aligned}$$

□

Indeed, this property is incredibly useful because we can provide $|1\rangle$ to the lower register of the QPE circuit instead of a single eigenvector, in order to compute the same algorithm but to all the eigenvectors **simultaneously**. We observe that $|1\rangle$ is essentially a superposition of all the eigenvectors of U_x, which means that when fed to C-$U_x^{2^k}$ as target qubit we don't

really get $|1\rangle$ back, indeed

$$\begin{aligned}
\text{C-}U_x^{2^k}(|+\rangle_0 \otimes |1\rangle_1) &= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes U_x^{2^k}|1\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes U_x^{2^k}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}U_x^{2^k}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{2\pi i 2^k s/r}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle_1 + |1\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{2\pi i 2^k s/r}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2r}}\sum_{s=0}^{r-1}\left(|0\rangle_0 + e^{2\pi i 2^k s/r}|1\rangle_0\right)\otimes|u_s\rangle_1
\end{aligned}$$

Nevetheless, notice what happened here: we engangled the control qubit with the target qubit, such that now the target *picked up the phase* $e^{2\pi i 2^k s/r}$ — exactly as the standard QPE — and we are still preserving the superposition of eigenvectors $|u_s\rangle$. This is exactly what we need, however it also means that the result at the end of the circuit is not very straightforward: in standard QPE the resulting state is just the tensor product of the single control qubits because they are independent from each other, but here we are entangling all the control and the target qubits together at each application of C-$U_x^{2^k}$. Hence, we need to compute what happens step by step: fallo

$$\begin{aligned}
&TODO \\
=&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}\left(|0\rangle + e^{2\pi i 2^{m-1}s/r}|1\rangle\right)\otimes|u_s\rangle \\
=&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}\left(|0\rangle + e^{2\pi i 0.\varphi_{s_m}\cdots\varphi_{s_t}}|1\rangle\right)\otimes|u_s\rangle \\
=&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\text{QFT}_t|\varphi_s\rangle_{q_0\cdots q_{t-1}}\otimes|u_s\rangle \\
\xrightarrow{\text{QFT}_t^\dagger(|q_0\cdots q_{t-1}\rangle)}&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\varphi_s\rangle\otimes|u_s\rangle
\end{aligned}$$

assuming that $\varphi_s = s/r$ can be written through t bits. This entangled quantum state is exactly what we want, because now by measuring the first t registers we get that for each $s \in [0, r-1]$ they will collapse into $|\varphi_s\rangle$ with probability $\frac{1}{r}$, and the last register will collapse into $|u_s\rangle$ which is exactly the eigenvector associated to $e^{2\pi i\varphi_s}$. This proves that we truly computed the phases of all the possible eigenvectors simultaneously.

We are almost done. We now know that we can retrieve $\varphi_s = s/r$ for each $s \in [0, r-1]$ without even knowing a single eigenvector of U_x, but there is a problem. Say that we

want to find the order of $x = 3$ modulo $N = 11$, which is $r = 5$, and suppose that the true phase that the algorithm *should* output is

$$\varphi_s = \frac{s}{r} \implies \varphi_2 = \frac{2}{5} = 0.4$$

The QPE algorithm, however, cannot output this exact value, because 0.4 cannot be written precisely in binary — in particular, the reason is that 5 does not divide 2^t for any t. This means that our algorithm will return an *approximation* of φ_2. Say that we are working with 6 registers of precision and fix $t = 6$; then QPE will output the closest approximation of 0.4 with 6 bits:

$$\tilde{\varphi}_2 = 0.40625 = \frac{13}{32}$$

Now, it would be a mistake to assume that $r = 32$! How do we proceed?

Definition 6.2: Continued fraction

Given a rational number $x \in \mathbb{Q}$, the continued fraction of x is composed by a set of numbers $a_0, \ldots, a_n \in \mathbb{N}$ — where $a_n \neq 0$ — such that

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}}$$

The continued fraction is usually denoted with the following notation

$$x = [a_0; a_1, a_2, \ldots, a_n]$$

Moreover, each "partial" continued fraction

$$\begin{aligned}
&[a_0] \\
&[a_0; a_1] \\
&[a_0; a_1, a_2] \\
&\ldots \\
&[a_0; a_1, a_1, \ldots, a_n]
\end{aligned}$$

is called **convergent** of the original number.

For instance, the number $\frac{338}{121}$ can be written as follows:

$$\frac{338}{121} = [2; 1, 3, 1, 5, 4]$$

and its convergents are

$$[2]$$
$$[2;1]$$
$$[2;1,3]$$
$$\dots$$
$$[2;1,3,\dots,4]$$

We will use continued fractions to recover r from $\tilde{\varphi}_s$. Suppose that QPE outputs $\tilde{\varphi}_2 = \frac{13}{32}$ as in the previous example. Without going into the details, there is a classical algorithm which is able to recover its continued fraction in $O(L^3)$ time — where $L = \lceil \log N \rceil$. Thus, we get that

$$\tilde{\varphi}_2 = 0 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{6}}} = [0;2,2,6]$$

Now, to recover $\frac{2}{5}$ we just need to compute the convergents:

$$[0] = 0$$
$$[0;2] = \frac{1}{2}$$
$$[0;2,2] = \frac{2}{5}$$
$$[0;2,2,6] = \frac{13}{32}$$

Finally, since we know that r is the order of x modulo N, we just need to pick the convergent with the smallest denominator r such that $x^r \equiv 1 \bmod N$. In our example we had $x = 3$ and $N = 11$, thus

$$3^2 \equiv 9 \not\equiv 1 \bmod 11$$
$$3^5 \equiv 243 \equiv 1 \bmod 11$$

This is how we can recover $r = 5$ — to be precise, there are theoretical results which guarantee that $\frac{2}{5}$ shows among the convergents of $\frac{13}{32}$ but this is ouside the scope of our discussion.

6.2 Shor's algorithm

TODO intro

Theorem 6.3: Fundamental Theorem of Arithmetic

Given any $N \in \mathbb{N}$, there exists unique $p_1, \dots, p_m \in \mathbb{P}$ primes and $\alpha_1, \dots, \alpha_m$ exponents — for some $m \in \mathbb{N}$ — such that

$$N = p_1^{\alpha_1} \cdot \ldots \cdot p_m^{\alpha_m}$$

The set of primes $p_1, \ldots, p_m \in \mathbb{P}$ is called **prime factorization** of N. The **Integer Factorization Problem (IFP)** asks to retrieve the prime factorization of any given $N \in \mathbb{N}$.

Theorem 6.4

Given any L-bit long composite natural number $N \in \mathbb{N}$, let $x \in [2, N]$ different from $N-1$ be a solution to the equation

$$x^2 \equiv 1 \bmod N$$

Then, at least one of $\gcd(x-1, N)$ and $\gcd(x+1, N)$ is a non-trivial factor of N that can be computed using $O(L^3)$ operations.

Theorem 6.5

Given any $N \in \mathbb{N}$ odd composite positive integer with m prime factors, let $x \in_R [1, N-1]$ such that $\gcd(1, N-1) = 1$. Then, if r is the order of x modulo N, it holds that

$$\Pr[r \text{ is even and } x^{r/2} \not\equiv 1 \bmod N] \geq 1 - \frac{1}{2^m}$$

TODO

TODO TODO

TODO

write the alg and explain stuff?

size of QPE

write QPE alg

size of Ux gates

7 Grover's algorithm

The algorithm that we are going to present in this chapter is, together with Shor's algorithm, one of the most famous quantum algorithms we currently know. In 1996, Grover [Gro97] published a landmark paper called "Quantum Mechanics Helps in Searching for a Needle in a Haystack", which contains the algorithm that we will present in this chapter. In the same year, Grover famously noted:

> It might be possible to combine the search scheme of this paper with Shor [Sho] and other quantum mechanical algorithms to design faster algorithms

This statement led many to speculate that Grover may have drawn inspiration from Shor's algorithm, although the exact influence remains unclear. Nevertheless, we will see at the end of the chapter that it is indeed possible to apply the QPE technique to Grover's algorithm.

7.1 The search problem

The name of Grover's work already suggests the problem his work tried to solve: the search problem. The setting is the following: we are given an array of N elements — we can assume that N is always a power of 2 for some n, i.e. $N = 2^n$ — that contains M "solution" elements. However, we do not know their positions, and the problem asks to find the index of any solution element.

More formally, given a Boolean function

$$f : \{0, \dots, N-1\} \to \mathbb{B} : x \mapsto \begin{cases} 1 & A[x] \in S \\ 0 & \text{otherwise} \end{cases}$$

where A is our array, and S is the set of solution elements, the problem asks to return an x such that $f(x) = 1$, i.e. such that $A[x]$ is a solution.

With a classical computation, it is easy to see that we need $O\left(\frac{N}{M}\right)$ accesses to A to solve our problem, however we will see that the algorithm Grover developed is able to return

a "solution index" in $O\left(\sqrt{\frac{M}{N}}\right)$ with *high probability*, i.e. Grover's algorithm provides a **quadratic** speedup compared to any classical algorithm — however, it is probabilistic.

Before explaining the details of the algorithm, we need to define some new operators that will be used in Grover's algorithm, and introduce some general notation:

- given an arbitrary qubit $|\psi\rangle$, we will write its superposition of states as follows

  $$|\psi\rangle = \sum_{b \in \mathbb{B}^n} \alpha_b |b\rangle$$

  where $\sum_{b \in \mathbb{B}^n} |\alpha_b| = 1$
- we define an operator O_f (where f is the indicator function of the array defined before) that computes as follows:

  $$\forall x \in \mathbb{B}^n \quad O_f |x\rangle := (-1)^{f(x)} |x\rangle$$

- given an arbitrary qubit $|\psi\rangle$, we define a new operator W as follows:

  $$W := 2 |s\rangle \langle s| - I$$

  where $|s\rangle$ is the **uniform superposition**, a superposition of states in which each amplitude is equally likely

  $$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{B}^n} |x\rangle$$

- finally, we will define an operator G that will simply compose the last two operators we described

  $$G = W \cdot O_f$$

Algorithm 7.1: Grover's algorithm

Given a Boolean function f that describes the solution elements of an array having M solution elements, and n qubits, the algorithm returns a solution index with high probability TODO SPIEGA CHE VUOL DIRE.

1: **function** GROVER(f, q_0)
2: $q_0 \leftarrow H^{\otimes n}(q_0)$
3: **for** $i \in \left[1, O\left(\sqrt{\frac{N}{M}}\right)\right]$ **do** ▷ where $N = 2^n$
4: $q_0 \leftarrow G(q_0)$
5: **end for**
6: **return** measure(q_0)
7: **end function**

First of all, we see that this algorithm takes only n qubits as input, however the actual implementation of the algorithm is slightly different, as shown in the following quantum circuit below.

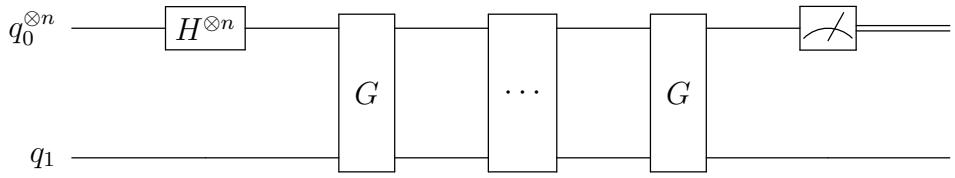


Figure 7.1: The quantum circuit for Grover's algorithm.

Indeed, we can see that the real quantum circuit takes $n+1$ inputs, and the additional register is called "ancilla" becuase its only purpose is to actually implement the O_f operator, which is designed exactly as if it was the U_f black-box we discussed in previous algorithms. This can be done thanks to Lemma 4.1, which guarantees that if we give $|-\rangle$ as the second input to U_f we get

$$O_f |x\rangle \otimes |-\rangle = (-1)^{f(x)} \otimes |-\rangle$$

which implies that the ancilla register will sill be $|-\rangle$, therefore we can just ignore the second register completely throughout the whole computation, and we are sure that O_f computes correctly.

Furthermore, before proceeding, let us prove that G is actually a unitary operator, i.e. that this is a valid quantum computation.

Claim: The operator G is unitary.

Proof of the Claim. By Proposition 2.11 we know that proving that W and O_f are unitary is sufficient to prove the claim. □

todo da finire

Now that we know this operator is unitary, we can delve into the details of the algorithm. To see what happens at each iteration, we will describe the complete state of the system in a rather unusual way. Let $|a\rangle$ and $|b\rangle$ be the following superpositions:

$$|a\rangle := \frac{1}{\sqrt{N-M}} \sum_{x \in \overline{S}} |x\rangle \qquad |b\rangle := \frac{1}{\sqrt{M}} \sum_{x \in S} |x\rangle$$

In other words, $|a\rangle$ is the uniform superposition of non-solution indices, and $|b\rangle$ is the superposition of solution ones. To explain the $\frac{1}{\sqrt{M}}$ factor in front of $|b\rangle$, we recall that in quantum mechanics all state vectors must be normalized, and since the squared norm of

$|\tilde{b}\rangle$ (which will denote $|b\rangle$ without the multiplicative factor in front) is equal to

$$\begin{aligned}
\langle\tilde{b}|\tilde{b}\rangle &= \left(\sum_{x\in S}|x\rangle\right)^{\dagger}\left(\sum_{x\in S}|x\rangle\right) \\
&= \left(\sum_{x\in S}\langle x|\right)\left(\sum_{y\in S}|y\rangle\right) \\
&= \sum_{x\in S}\sum_{y\in S}\langle x|y\rangle \\
&= \sum_{x\in S}\sum_{y\in S}\delta_{xy} \qquad \text{(basis states are orthonormal)} \\
&= \sum_{x\in S}1 \\
&= |S| \\
&= M
\end{aligned}$$

the norm of $|\tilde{b}\rangle$ is $\sqrt{M}$, hence to normalize it it suffices to add $\frac{1}{\sqrt{M}}$ in front of $|\tilde{b}\rangle$— the reasoning for $|a\rangle$ is analogous. Moreover, it's easy to see that $|a\rangle$ and $|b\rangle$ are orthogonal, so they form an orthonormal basis for a 2D space.

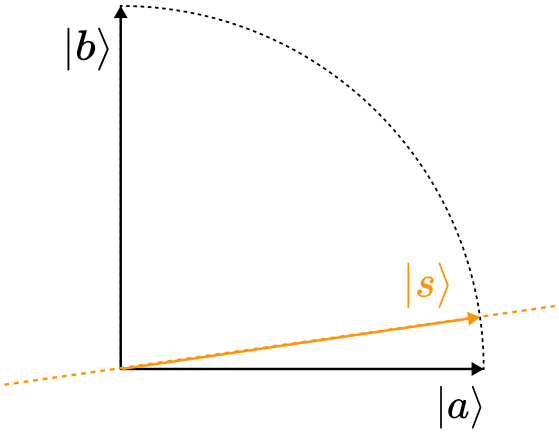
Now, because of how we defined $|a\rangle$ and $|b\rangle$, we can rewrite the uniform superposition

$$|s\rangle = \frac{1}{\sqrt{N}}\sum_{x\in\mathbb{B}}|x\rangle$$

as shown below

$$|s\rangle = \sqrt{\frac{N-M}{N}}\,|a\rangle + \sqrt{\frac{M}{N}}\,|b\rangle$$

This is quite interesting, because it means that we can describe $|s\rangle$ in terms of $|a\rangle$ and $|b\rangle$. Let us do exactly this, and plot the resulting graph on a 2D space that has $|a\rangle$ and $|b\rangle$ as axis.



We observe that:

- both $|s\rangle$, $|a\rangle$ and $|b\rangle$ are normalized, and all the vectors that we are going to consider are quantum states, so they will be normalized too, therefore we can restrict our focus on a 2D circumference of radius 1 — this plane is usually called **Grover plane**
- since we expect that $M \ll N$, we have that $\sqrt{\frac{M}{N}} \ll \sqrt{\frac{N-M}{N}}$, which basically means that the vector $|s\rangle$ is almost parallel to $|a\rangle$ (the bigger is the numer of solution indices, the bigger the angle between $|s\rangle$ and $|a\rangle$)

Consider any state $|\psi\rangle = \sum_{x \in \mathbb{B}^n} \alpha_x |x\rangle$; we observe that

$$\begin{aligned} O_f |\psi\rangle &= O_f \sum_{x \in \mathbb{B}^n} \alpha_x |x\rangle \\ &= \sum_{x \in \mathbb{B}^n} \alpha_x O_f |x\rangle \\ &= \sum_{x \in \mathbb{B}^n} \alpha_x (-1)^{f(x)} |x\rangle \end{aligned}$$

which basically means that each time we apply the O_f operator we are flipping the sign of the amplitudes of the compoents of $|\psi\rangle$ that represent solution indices. In other words, the O_f operator flips its input w.r.t. $|a\rangle$.

Moreover, we observe that any state $|\psi\rangle$ can be decomposed into

$$|\psi\rangle = \alpha |s\rangle + \beta |\psi_\perp\rangle$$

where $\alpha |s\rangle$ is the projection of $|\psi\rangle$ along $|s\rangle$'s space — thus $\alpha = \langle s|\psi\rangle$ — and $|\psi_\perp\rangle$ is the

projection of $|\psi\rangle$ along the space that is orthogonal to $|s\rangle$'s. Therefore, we have that

$$\begin{aligned} W|\psi\rangle &= W(\alpha|s\rangle + \beta|\psi_\perp\rangle) \\ &= 2|s\rangle\langle s|(\alpha|s\rangle + \beta|\psi_\perp\rangle) - (\alpha|s\rangle + \beta|\psi_\perp\rangle) \\ &= 2\alpha|s\rangle\langle s|s\rangle + 2\beta|s\rangle\langle s|\psi_\perp\rangle - (\alpha|s\rangle + \beta|\psi_\perp\rangle) \\ &= 2\alpha|s\rangle \cdot 1 + 0 - (\alpha|s\rangle + \beta|\psi_\perp\rangle) \\ &= \alpha|s\rangle - \beta|\psi_\perp\rangle \end{aligned}$$

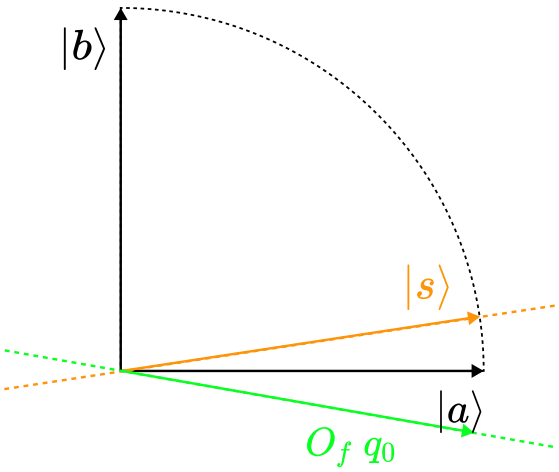
which means that what W actually performs is leaving the component along $|s\rangle$'s space unchanged, and it flips the sign of the component of the orthogonal space. In other words, what W computes is the reflection of $|\psi\rangle$ w.r.t. $|s\rangle$.

We can finally describe Grover's algorithm in detail. First, we see that

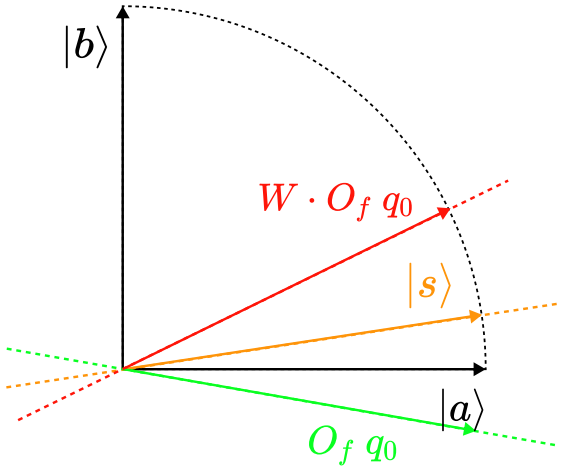
$$\begin{aligned} & q_0 \\ &= |0\rangle^{\otimes n} \\ \xrightarrow{H^{\otimes n}(q_0)} & \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} (-1)^{0^n \cdot x} |x\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{B}^n} |x\rangle \\ &= |s\rangle \end{aligned}$$

Indeed, the only purpose of the first Hadamard operator is to "move" the initial state slightly away from $|a\rangle$ — and also making each component initially equally likelly. Now, let's see what happens at each application of the $G = W \cdot O_f$ operator:

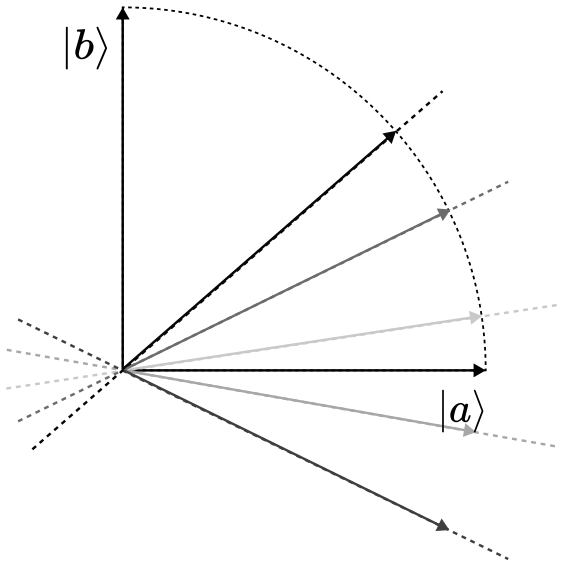
- as previously shown, the operator O_f reflects its input w.r.t. the $|a\rangle$ axis — below we show what happens when we first apply $O_f\ q_0 = O_f|s\rangle$:



- additionally, as previously explained the operator W reflects its input w.r.t. the axis described by the $|s\rangle$ vector, thus when we compute $W \cdot O_f \; q_0$ we will end up with the following vector:



- this suggests that each time we apply the operator G we are making q_0 closer and closer to $|b\rangle$, as depicted below:



This is the core idea of Grover's algorithm: if we rewrite q_0 in terms of $|a\rangle$ and $|b\rangle$

$$q_0 = \beta_0 \, |a\rangle + \beta_1 \, |b\rangle$$

we get that through Grover's algorithm we transformed q_0 such that it is now very close to $|b\rangle$, meaning that $\beta_0 \ll \beta_1$. This direcly implies that when we will measure q_0 at the end

of Grover's procedure the likelihood that it will collapse into some $|x\rangle$ that is a component of $|b\rangle$ — i.e. a solution index — is very high. In other words, what happens with Grover's algorithm is that we gradually increase the amplitudes of the solution indices, in order to maximize the probability that our qubit will collapse in one them when it will be measured at the end of the procedure.

Now that we know how Grover's algorithm works, the only thing left to discuss is the $O\left(\sqrt{\frac{N}{M}}\right)$ factor. Why is it guaranteed that after this amount of applications of the G operator we are done with the algorithm? Well, we actually have the opposite problem: in reality, we have to *stop early enough*. Consider again how Grover's algorithm operates in the Grover plane; clearly, if we apply G too many times, what happens is that q_0 will end up past $|b\rangle$ itself: drawing

Let the angle between $|a\rangle$ and $|s\rangle$ be θ; since O_f flips q_0 w.r.t. $|a\rangle$, and W flips O_f q_0 w.r.t. $|s\rangle$, G will cumulatively rotate q_0 by 2θ: More formally, we can actually show that drawing the matrix G is a rotation matrix of 2θ.

Proposition 7.1

The Grover operator G can be rewritten in the basis $\{|a\rangle, |b\rangle\}$ as follows:

$$G = \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix}$$

where $|s\rangle = \cos\theta\,|a\rangle + \sin\theta\,|b\rangle$.

Proof. First, we need two small identities in trigonometry.

Claim 1: For any angle θ it holds that $2\cos^2\theta - 1 = \cos 2\theta$.

Proof of the Claim. Through algebraic manipulation we have that

$$\begin{aligned} 2\cos^2\theta - 1 &= 2\cos^2\theta - (\sin^2\theta + \cos^2\theta) \\ &= 2\cos^2\theta - \sin^2\theta - \cos^2\theta \\ &= \cos^2\theta - \sin^2\theta \\ &= \cos 2\theta \qquad \text{(double-angle formula)} \end{aligned}$$

□

Claim 2: For any angle θ it holds that $1 - 2\sin^2\theta = \cos 2\theta$.

Proof of the Claim. From the previous claim we have that

$$1 - 2\sin^2\theta = 1 - 2(1 - \cos^2\theta) = 1 - 2 + 2\cos^2\theta = 1 + \cos 2\theta - 1 = \cos 2\theta$$

□

Let $|v\rangle$ be any vector described inside the $\{|a\rangle, |b\rangle\}$:

$$|v\rangle = \alpha\,|a\rangle + \beta\,|b\rangle$$

We already saw how the O_f operator flips its input w.r.t. the $|a\rangle$ axis, indeed its action on $|v\rangle$ would be the following:

$$O_f |v\rangle = \alpha |a\rangle - \beta |b\rangle$$

Thus, starting from the definition of G, by the two previous claims we have that

$$\begin{aligned}
G|v\rangle &= W \cdot O_f |v\rangle \\
&= W \cdot (\alpha |a\rangle - \beta |b\rangle) \\
&= (2|s\rangle\langle s| - I)(\alpha |a\rangle - \beta |b\rangle) \\
&= 2|s\rangle\langle s|(\alpha |a\rangle - \beta |b\rangle) - (\alpha |a\rangle - \beta |b\rangle) \\
&= 2|s\rangle(\cos\theta |a\rangle + \sin\theta |b\rangle)^\dagger(\alpha |a\rangle - \beta |b\rangle) - (\alpha |a\rangle - \beta |b\rangle) \\
&= 2|s\rangle(\cos\theta \langle a| + \sin\theta \langle b|)(\alpha |a\rangle - \beta |b\rangle) - (\alpha |a\rangle - \beta |b\rangle) \\
&= 2|s\rangle(\alpha\cos\theta - \beta\sin\theta) - (\alpha |a\rangle - \beta |b\rangle) \\
&= 2(\cos\theta |a\rangle + \sin\theta |b\rangle)(\alpha\cos\theta - \beta\sin\theta) - (\alpha |a\rangle - \beta |b\rangle) \\
&= 2\alpha\cos^2\theta |a\rangle - 2\beta\cos\theta\sin\theta |a\rangle + 2\alpha\sin\theta\cos\theta |b\rangle - 2\beta\sin^2\theta |b\rangle - \alpha |a\rangle + \beta |b\rangle \\
&= \left[\alpha\left(2\cos^2\theta - 1\right) - 2\beta\cos\theta\sin\theta\right]|a\rangle + \left[\beta\left(1 - 2\sin^2\theta\right) + 2\alpha\sin\theta\cos\theta\right]|b\rangle \\
&= (\alpha\cos 2\theta - \beta\sin 2\theta)|a\rangle + (\alpha\sin 2\theta + \beta\cos 2\theta)|b\rangle \\
&= \begin{pmatrix} \alpha\cos 2\theta - \beta\sin 2\theta \\ \alpha\sin\theta + \beta\cos\theta \end{pmatrix} \\
&= \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\
&= \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin\theta & \cos\theta \end{pmatrix} |v\rangle
\end{aligned}$$

which concludes that G must be the rotation matrix of 2θ. □

Indeed, with each application of G we are rotating q_0 by 2θ, which means that at the k-th application it holds that

$$G^k\ q_0 = \cos(2k+1)\theta |a\rangle + \sin(2k+1)\theta |b\rangle$$

for any $k \in \mathbb{N}$, where the additional 1 comes from the fact that $q_0 = |s\rangle$ through the Hadamard transformation at the start of the process. Thus, to evaluate the optimal number of iterations we need to find the optimal k, i.e. the one that maximizes the probability of measuring a solution index, which is equal to the squared amplitude of $|b\rangle$, namely

$$\Pr[\text{measure}(q_0) = |b\rangle] = \sin^2(2k+1)\theta$$

Hence, we have that $\sin^2(2k+1)\theta = 1$ when $(2k+1)\theta = \frac{\pi}{2}$, and solving for k we get that

$$k = \frac{\pi}{4\theta} - \frac{1}{2}$$

Lastly, since θ is the angle between $|a\rangle$ and $|s\rangle$, we can rewrite $|s\rangle$ as

$$|s\rangle = \cos\theta |a\rangle + \sin\theta |b\rangle$$

which directly implies that

$$\sin\theta = \sqrt{\frac{M}{N}} \iff \theta = \arcsin\sqrt{\frac{M}{N}}$$

and therefore

$$\begin{aligned} k &= \frac{\pi}{4\theta} - \frac{1}{2} \\ &= \frac{\pi}{4\arcsin\sqrt{\frac{M}{N}}} - \frac{1}{2} \\ &\leq \frac{\pi}{4\arcsin\sqrt{\frac{M}{N}}} \\ &\leq \frac{\pi}{4\sqrt{\frac{M}{N}}} \\ &= \frac{\pi}{4}\sqrt{\frac{N}{M}} \\ &= O\left(\sqrt{\frac{N}{M}}\right) \end{aligned}$$

which finally explains the quadratic speedup of Grover's algorithm w.r.t. the classical version of the problem.

As a final note, we observe that Grover's algorithm assumes that $\theta \leq \frac{\pi}{4}$, otherwise we overshoot $|b\rangle$ with a single iteration of the G operator — since $\theta > \frac{\pi}{4}$ would mean that $|s\rangle$ is placed on the "upper half" of the Grover plane. However, to ensure this constraint on θ we only need that $M \leq \frac{N}{2}$, i.e. at most half of the elements in our array are solution elements. Indeed, we see that

$$M \leq \frac{N}{2} \implies \sin\theta = \sqrt{\frac{M}{N}} \leq \sqrt{\frac{1}{2}} \implies \theta \leq \arcsin\sqrt{\frac{1}{2}} = \frac{\pi}{4}$$

What can we do if the number of solutions is more than half the size of the array? We simply invert the problem and find the non-solutions!

7.1.1 Another perspective

Interestingly enough, we can look at what happens to the amplitudes of q_0 from another perspective. Usually, the W operator is called **diffusion operator**, and what it does is computing the so called *inversion about the mean* of its input. Consider any state

$|\psi\rangle = \sum_{x\in\mathbb{B}^n} \alpha_x |x\rangle$, and observe that

$$\begin{aligned}
\langle s|\psi\rangle &= \left(\frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}|y\rangle\right)^\dagger \left(\sum_{x\in\mathbb{B}^n}\alpha_x|x\rangle\right) && \text{(since } \langle s| = |s\rangle^\dagger\text{)}\\
&= \left(\frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}\langle y|\right)\left(\sum_{x\in\mathbb{B}^n}\alpha_x|x\rangle\right)\\
&= \frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}\sum_{x\in\mathbb{B}^n}\alpha_x\langle y|x\rangle\\
&= \frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}\sum_{x\in\mathbb{B}^n}\alpha_x\delta_{xy} && \text{(basis states are orthonormal)}\\
&= \frac{1}{\sqrt{N}}\sum_{x\in\mathbb{B}^n}\alpha_x\sum_{y\in\mathbb{B}^n}\delta_{xy}\\
&= \frac{1}{\sqrt{N}}\sum_{x\in\mathbb{B}^n}\alpha_x\\
&= \sqrt{N}\overline{\alpha}_\psi
\end{aligned}$$

where $\overline{\alpha}_\psi$ is the average amplitude of $|\psi\rangle$. This implies that

$$\begin{aligned}
W|\psi\rangle &= (2|s\rangle\langle s| - I)|\psi\rangle\\
&= 2|s\rangle\langle s|\psi\rangle - \sum_{x\in\mathbb{B}^n}|x\rangle\\
&= 2|s\rangle\left(\sqrt{N}\overline{\alpha}_\psi\right) - \sum_{x\in\alpha_x|x\rangle} && \text{(for the previous observation)}\\
&= 2\left(\frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}|y\rangle\right)\sqrt{N}\overline{\alpha}_\psi - \sum_{x\in\mathbb{B}^n}\alpha_x|x\rangle\\
&= \sum_{x\in\mathbb{B}^n}(2\overline{\alpha}_\psi - \alpha_x)|x\rangle
\end{aligned}$$

In other words, when we apply W to a superposition of states, what happens is that each amplitude of the basis states is trasformed as follows:

$$W : \alpha_x \to 2\overline{\alpha}_\psi - \alpha_x$$

To understand why this is important, let's look at what happens in Grover's algorithm after the first Hadamard application. TODO da finire

7.2 Fixed-Point Quantum Search

As we saw in the previous sections, with Grover's algorithm we need to be careful on how many times we apply the G operator, however we observe that the right value for k — i.e. the numeber of iterations — strictly depends on both N and M. Assuming that N, the

length of the array, is known, it is not guaranteed that we know M too (the number of solutions in the array). Therefore, if we want to apply Grover's algorithm we also need to be able to to a rough estimate on M, otherwise we might end up stopping either too early or too late. This makes Grover's algorithm fragile in some real-world scenarios.

Grover was actually aware of this problem, and in 2005 he designed a new algorithm which is able to solve this issue [Gro05]. The idea of the algorithm is to *monotonically* increase the probability of finding a solution, without oscillating back down $|a\rangle$. This algorithm is called **Fixed-Point Quantum Search**, becuase the solutions actually become a "stable fixed point" of the transformation — i.e. once you reach a good solution, further iterations leave it basically unchanged. Indeed, with Grover's search each iteration is a constant-angle rotation, while in fixed-point search we will see that the phase angles in each rotation changes such that the rotation angle decreases over time.

First, we need to define two new operators:

$$R_s := I - (1 - e^{i\theta}) |s\rangle \langle s| \qquad R_t := I - (1 - e^{i\theta}) |t\rangle \langle t|$$

where $|s\rangle$ is the starting state and $|t\rangle$ is the target state (in Grover's algorithm this was $|b\rangle$) — this is the original notation that Grover used in his paper, and actually explains why we used $|s\rangle$ in the previous version of the algorithm, it's just the "start". These two operators are called **phase shift operators**; as usual, before proceeding we need to show that both operators are actually unitary.

Claim: R_t and R_s are unitary operators.

Proof of the Claim. TODO □ todo

What are these two operators in the first place? When we presented the W operator, we also noticed how it actually performes a reflection of its input w.r.t. the space of $|s\rangle$. Well, through a very similar argument it can be shown that

$$W^{\perp} := I - 2 |s\rangle \langle s|$$

performs a reflection of its input w.r.t. the space *orthogonal* to $|s\rangle$ — indeed, we end up with

$$W^{\perp} |\psi\rangle = \beta |\psi_{\perp}\rangle - \alpha |s\rangle$$

If we now look at the R_s operator, we can see that when we actually compute the reflection it performs we end up with

$$R_s |\psi\rangle = \beta |\psi_{\perp}\rangle + e^{i\theta}\alpha |s\rangle$$

This suggests that what R_s actually computes is a "soft reflection" w.r.t. the space perpendicular to $|s\rangle$. Through an analogous argument, we can see that

$$R_t |\psi\rangle = \beta |\psi_{\perp}\rangle + e^{i\theta}\alpha |t\rangle$$

meaning that R_t computes a "soft reflection" w.r.t. the space perpendicular to $|t\rangle$ — however, we observe that the latter is literally the space of $|a\rangle$, indeed O in Grover's algorithm could have been defined as

$$O = I - 2 |b\rangle \langle b|$$

and indeed it is sometimes defined such.

Now, we are going to define an addition operator called U as such: let U be any unitary operator such that, for some small $\varepsilon > 0$, it holds that

$$|\langle t|Us\rangle|^2 = 1 - \varepsilon$$

We observe that

- by the laws of quantum mechanics we have that

  $$\Pr[\text{measure}\,(U\,|s\rangle) = |t\rangle] = |\langle t|Us\rangle|^2$$

  therefore we require U to be an operator such that it "drives $|s\rangle$ close to $|t\rangle$ with high probability" — i.e. $1 - \varepsilon$
- we know that U exists since it's just a rotation in a 2D space
- also, a geometric interpretation of the scalar product is that we are measuring the cosine of the angle between $|t\rangle$ and $|Us\rangle$, and we want this value to be very high (such that the angle woule be very small) — we obeserve that we are in Hilbert spaces so the notion of "angle" is not the same of the one we are used to with Euclidean spaces, but this is just to have an idea of what is happening with U

Moreover, define the following operator

$$G := UR_sU^\dagger R_tU$$

Let's see what happens when we evaluate $G\,|s\rangle$. By denoting $U_{ts} = \langle t|Us\rangle$, we can prove the following equality.

Lemma 7.1

It holds that

$$G\,|s\rangle = U\,|s\rangle\left[e^{i\theta} + |U_{ts}|^2\left(e^{i\theta} - 1\right)^2\right] + |t\rangle\,U_{ts}\left(e^{i\theta} - 1\right)$$

Proof. First, we prove the following equality.

Claim: It holds that $\langle s|\,U^\dagger\,|t\rangle\,U_{ts} = |U_{ts}|^2$.

Proof of the Claim.

$$\begin{aligned}
\langle s|\,U^\dagger\,|t\rangle\,U_{ts} &= \langle s|U^\dagger t\rangle\,U_{ts} \\
&= \langle Us|t\rangle\,U_{ts} && \text{(by def. of adjoint)} \\
&= \overline{\langle t|Us\rangle}U_{ts} && \text{(by prop. of scalar products)} \\
&= \overline{U_{ts}}U_{ts} \\
&= |U_{ts}|^2 && (z\cdot\overline{z} = |z|^2)
\end{aligned}$$

□

Let $P_s = |s\rangle\langle s|$ and $P_t = |t\rangle\langle t|$; thus, we have that

$$\begin{aligned}
G &= UR_sU^\dagger R_tU \\
&= U(I-(1-e^{i\theta})P_s)U^\dagger(I-(1-e^{i\theta})P_t)U \\
&= (U-(1-e^{i\theta})UP_s)U^\dagger(U-(1-e^{i\theta})UP_t) \\
&= (UU^\dagger-(1-e^{i\theta})UP_sU^\dagger)(U-(1-e^{i\theta})UP_t) \\
&= (I-(1-e^{i\theta})UP_sU^\dagger)(U-(1-e^{i\theta})UP_t) && (UU^\dagger = I) \\
&= U-(1-e^{i\theta})P_tU-(1-e^{i\theta})UP_s+(1-e^{i\theta})^2UP_sU^\dagger P_tU
\end{aligned}$$

Therefore, we find that $G|s\rangle$ can be evaluated as follows:

$$\begin{aligned}
G|s\rangle &= \left(U-(1-e^{i\theta})P_tU-(1-e^{i\theta})UP_s+(1-e^{i\theta})UP_sU^\dagger P_tU\right)|s\rangle \\
&= U|s\rangle-(1-e^{i\theta})P_tU|s\rangle-(1-e^{i\theta})UP_s|s\rangle+(1-e^{i\theta})^2UP_sU^\dagger P_tU|s\rangle \\
&= U|s\rangle-(1-e^{i\theta})|t\rangle U_{ts}-(1-e^{i\theta})U|s\rangle+(1-e^{i\theta})^2U|s\rangle\langle s|U^\dagger|t\rangle U_{ts} \\
&= U|s\rangle-(1-e^{i\theta})|t\rangle U_{ts}-(1-e^{i\theta})U|s\rangle+(1-e^{i\theta})^2U|s\rangle|U_{ts}|^2 && \text{(by the claim)} \\
&= \dots && \text{(algebraic manipulation)} \\
&= U|s\rangle\left[e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right]+|t\rangle U_{ts}\left(e^{i\theta}-1\right)
\end{aligned}$$

□

Most importantly, this equality can be used in the following proposition, which shows that we can actually find an angle θ for which the distance from $|t\rangle$ decreases significantly.

Proposition 7.2

There exists an angle θ such that

$$\Pr[\text{measure}(G\ |s\rangle) = |t\rangle] = 1-\varepsilon^3$$

Proof. Thanks to the previous lemma, we obtain that

$$\begin{aligned}
\Pr[\text{measure}\,(G|s\rangle) = |t\rangle] &= |\langle t|Gs\rangle|^2 \\
&= \left|\langle t|\left[U|s\rangle\left(e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right)+|t\rangle U_{ts}\left(e^{i\theta}-1\right)\right]\right|^2 \\
&= \left|\langle t|Us\rangle\left(e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right)+\langle t|t\rangle U_{ts}\left(e^{i\theta}-1\right)\right|^2 \\
&= \left|U_{ts}\left(e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right)+U_{ts}\left(e^{i\theta}-1\right)\right|^2 \\
&= \left|U_{ts}\left[2e^{i\theta}-1+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right]\right|^2 \\
&= |U_{ts}|^2\cdot\left|2e^{i\theta}-1+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right|^2 \\
&= (1-\varepsilon)\cdot\left|2e^{i\theta}-1+(1-\varepsilon)\left(e^{i\theta}-1\right)^2\right|^2 && \left(|U_{ts}|^2 = 1-\varepsilon\right)
\end{aligned}$$

Now, let's see what happens if we set $\theta = \frac{\pi}{3}$: we obtain that

$$\begin{aligned}
\Pr[\text{measure}\,(G\,|s\rangle) = |t\rangle] &= (1-\varepsilon)\cdot\left|2e^{i\frac{\pi}{3}} - 1 + (1-\varepsilon)\left(e^{i\frac{\pi}{3}} - 1\right)^2\right|^2 \\
&= (1-\varepsilon)\cdot\left|2e^{i\frac{\pi}{3}} - 1 - (1-\varepsilon)i\tfrac{\pi}{3}\right|^2 && \left(\left(e^{i\frac{\pi}{3}} - 1\right)^2 = -e^{i\frac{\pi}{3}}\right) \\
&= (1-\varepsilon)\cdot\left|(1+\varepsilon)e^{i\frac{\pi}{3}} - 1\right|^2 \\
&= (1-\varepsilon)\cdot\left|(1+\varepsilon)\left(\frac{1}{2} + i\frac{\sqrt{4}}{2}\right) - 1\right|^2 && \text{(by Euler's formula)} \\
&= (1-\varepsilon)\cdot\left|\frac{1}{2} + i\frac{\sqrt{3}}{2} + \frac{\varepsilon}{2} + i\frac{\sqrt{3}}{2}\varepsilon - 1\right|^2 \\
&= (1-\varepsilon)\cdot\left|\frac{\varepsilon}{2} - \frac{1}{2} + i\frac{\sqrt{3}}{2}(1+\varepsilon)\right|^2 \\
&= (1-\varepsilon)\cdot\left[\left(\frac{\varepsilon}{2} - \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}(1+\varepsilon)\right)^2\right] && \left(|z|^2 = \Re^2(z) + \Im^2(z)\right) \\
&= (1-\varepsilon)\cdot(1+\varepsilon+\varepsilon^2) \\
&= 1+\varepsilon+\varepsilon^2-\varepsilon-\varepsilon^2-\varepsilon^3 \\
&= 1-\varepsilon^3
\end{aligned}$$

This shows that by applying G the probability of measuring $|t\rangle$ has increased from $1-\varepsilon$ to $1-\varepsilon^3$. □

Indeed, it can be shown that by defining the following recursive sequence of operators

$$\begin{cases} U_0 := U & m = 0 \\ U_m = U_{m-1} R_s U_{m-1}^\dagger R_t U_{m-1} & m \geq 1 \end{cases}$$

we get that

$$\Pr[\text{measure}(|U_m\rangle\,|s\rangle) = |t\rangle] = \langle\langle t|U_m s\rangle\rangle^2 = 1 - \varepsilon^{2q_m+1}$$

where q_m is the number of queries of $f(x)$.

TODO

drawing of the vector that grows monotonically

Unfortunately, there already exists a classical probabilistic algorithm that the failure probability drops as ε^{q+1} after q queries to f. Thus, the quantum advantage with this method is lost.

So, what do we do now? In 2014 Yoder, Low, and Chuang [YLC14] proposed a fixed-point quantum serach algorithm that monotonically converges to the target state while still retaining the quadratic advantage of the original Grover's algorithm over classical algorithms. The algorithm involves phase-shift operators that are parametrized with angles different from $\theta = \frac{\pi}{3}$, and again involes building a sequence of operators using

said phase shifts. However, the details of this result are way beyond the scope of our discussion, so we won't describe the details of their findings.

7.3 Quantum counting

At the end of the previous section we ended our discussion by noticing how the G operator was actually performing a *rotation* of an angle 2θ, but there is still something left to be explained.

Let's try to understand how we can actually employ some of the ideas used in Shor's algorithm for Grover's quantum search. Recall that in Proposition 7.1 we proved that the Grover operator can be rewritten as follows:

$$G = \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix}$$

We observe the following property.

Proposition 7.3

The eigenvalues of G are $e^{\pm 2\theta i}$.

Proof. The characteristic polynomial of G is given by

$$\begin{aligned} p(\lambda) &= \det(G - \lambda I) \\ &= \det\left(\begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right) \\ &= \det\begin{pmatrix} \cos 2\theta - \lambda & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta - \lambda \end{pmatrix} \\ &= \lambda^2 - 2\lambda\cos 2\theta + \cos^2 2\theta + \sin^2 2\theta \\ &= \lambda^2 - 2\lambda\cos 2\theta + 1 \end{aligned}$$

and solving for $p(\lambda) = 0$ yields the following two eigenvalues:

$$\begin{aligned} \lambda &= \frac{2\cos 2\theta \pm \sqrt{4\cos^2 2\theta - 4}}{2} \\ &= \cos 2\theta \pm \sqrt{\cos^2 2\theta - 1} \\ &= \cos 2\theta \pm \sqrt{-\sin^2 2\theta} \\ &= \cos 2\theta \pm i \sin 2\theta \\ &= e^{\pm 2\theta i} \end{aligned}$$

□

This means that we can use the QPE algorithm with the matrix G in order to obtain an estimate of θ! In particular, with this information we can actually solve two problems at once:

- we can determine an estimate how many iterations of Grover's algorithm we need to run, since
$$k = \frac{\pi}{4\theta} - \frac{1}{2}$$
- we can also evaluate an estimate on M, the number of solutions, since
$$\theta = \arcsin\sqrt{\frac{M}{N}} \iff M = N\sin^2\theta$$

The problem of estimating M is usually referred to as **quantum counting** in the literature, since the output is the number of solutions inside the given array.

TODO drawing

It can be proven that the quantum counting circuit estimates $\pm 2\theta$ — to be precise, it actually estimates 2θ or $2\pi - 2\theta$ since QPE does not return negative angles — to a degree of accuracy up to a desired 2^-m with probability at least $1-\varepsilon$. Moreover, it can be shown that
$$\left|\tilde{M} - M\right| < \left(2\sqrt{MN} + \frac{N}{2^{m+1}}\right) 2^{-m}$$
where $\tilde{M}$ is the estimated value of M through the QPE algorithm. For instance, choosing $m = \left\lceil \frac{n}{2} \right\rceil + 1$ and $\varepsilon = \frac{1}{6}$, we get that $t = \left\lceil \frac{n}{2} \right\rceil + 3$ and an estimation of M with an error of about
$$\left|\tilde{M} - M\right| < \sqrt{\frac{M}{2}} + \frac{1}{4} = O(\sqrt{M})$$
with only $O(2^t) = O(\sqrt{N})$ iterations of the Grover operator, i.e. array accesses. Classically, to know the value of M with the same error in the estimate we would necessarily need $O(N)$ accesses.

Lastly, another problem that quantum counting solves is knowing if M is 0 or not. Indeed, Grover's algorithm relies on the assumpion that $M \neq 0$, i.e. there are solution elements, otherwise the procedure actually does nothing.

8

Quantum circuits

So far, we have treated quantum gates primarily as "abstract" unitary operators acting on quantum states. While this perspective is sufficient to define quantum algorithms and reason about their correctness, it leaves an important question unanswered: how are these gates *actually* constructed? In other words, given a desired unitary transformation, how can it be realized using a finite set of elementary operations that a quantum computer can physically implement? The central result of this chapter shows that any controlled single-qubit unitary can be constructed using only a restricted set of *single-qubit gates*, and a small number of CNOT gate.

8.1 Rotation operators

Together with the Pauli matrices and the Hadamard gate, we shall introduce 2 additional quantum gates that will play a large part in this chapter, namely the **S gate** and the **T gate**:

$$S := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad T := \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{pmatrix}$$

The S gate is usually called **phase gate**, while the T gate is sometimes called the $\pi/8$ **gate** for historical reasons (even if $\pi/4$ is the fraction that appears in its definition).

Proposition 8.1

It holds that $S = T^2$.

In this chapter, we will see *any* arbitrary quantum computation can be reduced to a combination of

- the Hadamard H, S and T gates
- the CNOT gate

The importance of this result stems from the fact that these gates can be physically realized, and can therefore be treated as the *building blocks* of any other unitary transformation. We will discuss we this set of gates is both *necessary* and *sufficient* to achieve **quantum approximate universality** to arbitrary accuracy — unfortunately, it is not known if we can achieve exact universality avoiding an exponential number of gates in the general case.

In the first section of this chapter we will discuss the necessity of Pauli matrices. But first, we need to present a new type of matrices, the **rotation operators** which apply rotations on the x, y and z axes.

Definition 8.1: Rotation operators

The **rotation operators** $R_i(\theta)$ that apply a rotation of an angle θ on the i-th axis (for $i \in \{x, y, z\}$) are defined as follows:

$$R_x(\theta) := \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \equiv e^{-i\theta X/2}$$

$$R_y(\theta) := \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \equiv e^{-i\theta Y/2}$$

$$R_z(\theta) := \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \equiv e^{-i\theta Z/2}$$

Lemma 8.1

For any real nubmer $x \in \mathbb{R}$ and matrix A such that $A^2 = I$, it holds that

$$e^{iAx} = I\cos(x) + iA\sin(x)$$

Proof. From the Taylor expansions of e^x, $\sin x$ and $\cos x$ we get the following

$$
\begin{aligned}
e^{iAx} &= \sum_{n=0}^{+\infty} \frac{(iAx)^n}{n!} \\
&= \sum_{n \text{ even}}^{+\infty} \frac{(iAx)^n}{n!} + \sum_{n \text{ odd}}^{+\infty} \frac{(iAx)^n}{n!} \\
&= \sum_{k=0}^{+\infty} \frac{(iAx)^{2k}}{(2k)!} + \sum_{k=0}^{+\infty} \frac{(iAx)^{2k+1}}{(2k+1)!} \\
&= \sum_{k=0}^{+\infty} \frac{i^{2k} A^{2k}}{(2k)!} x^{2k} + \sum_{k=0}^{+\infty} \frac{i^{2k+1} A^{2k+1}}{(2k+1)!} x^{2k+1} \\
&= \sum_{k=0}^{+\infty} \frac{(i^2)^k (A^2)^k}{(2k)!} x^{2k} + \sum_{k=0}^{+\infty} \frac{i \cdot (i^2)^k \cdot A \cdot (A^2)^k}{(2k+1)!} x^{2k+1} \\
&= I \sum_{k=0}^{+\infty} \frac{(-1)^k}{(2k)!} x^{2k} + iA \sum_{k=0}^{+\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} \qquad (A^2 = I = I^k) \\
&= I \cos(x) + iA \sin(x)
\end{aligned}
$$

□

Corollary 8.1

The rotation operators can be rewritten through the Pauli gates as follows:

$$
\begin{aligned}
R_x(\theta) &= \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} X \\
R_y(\theta) &= \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Y \\
R_z(\theta) &= \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z
\end{aligned}
$$

Most importantly, the corollary above already implies that any type of rotation can be expressed in terms of only **linear combinations of Pauli matrices**. However, there is a catch: the rotation matrices written in these form are *not* constructible in practice. In the first part of this chapter we will not address this issue, and we will assume that we can construct rotation matrices of arbitrary angles. Then, in the second part of of our discussion we will discuss physical realizability of the ideas we will present.

Nevertheless, the above result together with the next theorem allow us to express *any* arbitrary single-qubit operation in terms of rotation matrices only.

Theorem 8.1: Z-Y single-qubit decomposition

If U is a unitary operation on a single-qubit, there exist $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ such that

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

Proof sketch. Since U is unitary, by Proposition 2.10 we know that its rows and columns must be orthonormal. This property implies the existence of real numbers $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ such that HOW???

$$U = \begin{pmatrix} e^{i(\alpha-\beta/2-\delta/2)} \cos\frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)} \sin\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)} \sin\frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)} \cos\frac{\gamma}{2} \end{pmatrix}$$

Lastly, this formulation can be rewritten as the statement describes. □

This theorem already shows how much we can achieve through rotation matrices alone. However, as we already encountered multiple times throughout our discussion, we know that this is not general enough. Indeed, what we are still missing is **controlled operations**. For instance, consider the controlled NOT, the CNOT operator; even if the CNOT is a *unitary transformation*, it acts on 2 qubits, which means that we cannot apply the theorem above. We observe that this problem already hints at reason why we included the CNOT in the discussion at the beginning of this chapter — akeen eye might have also noticed that without the CNOT we don't know how to create EPR pairs! We will see how to solve these problems in the next section.

8.2 Controlled operators

8.2.1 Single-qubit conditioning

In Section 5.3 we showed how any single-qubit operator U can be turned into a controlled operator for a single-qubit as follows:

$$\text{C-}U = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$$

However, we cannot turn this controlled operator directly into a quantum gate. This formulation tells us how C-U *looks like* in a matrix form, but what does the actual gate look like?

Before we progress, we must first prove a corollary of the *Z-Y* single-qubit decomposition.

Corollary 8.2

If U is a unitary operation on a single-qubit, there exist unitary operators A, B and C on a single-qubit such that $ABC = I$ and

$$U = e^{i\alpha} AXBXC$$

where α is some overall phase factor.

Proof. Consider the real values α, β, γ and δ obtained from the previous theorem, and define the following operators

- $A := R_z(\beta) R_y\left(\frac{\gamma}{2}\right)$
- $B := R_y\left(-\frac{\gamma}{2}\right) R_z\left(-\frac{\delta+\beta}{2}\right)$

- $C := R_z\left(\frac{\delta-\beta}{2}\right)$

Proving that $ABC = I$ is left as an exercise.

Claim: $XBX = R_y(\frac{\gamma}{2})R_z(\frac{\delta+\beta}{2})$.

Proof of the Claim. By the properties of Pauli matrices, we know that

- $X^2 = I$
- $XYX = -Y$
- $XZX = -Z$

from which it follows that

$$\begin{aligned}
XBX &= X\Big(R_y(-\tfrac{\gamma}{2})R_z(-\tfrac{\delta-\beta}{2})\Big)X \\
&= XR_y(-\tfrac{\gamma}{2})X\,XR_z(-\tfrac{\delta-\beta}{2})X && (X^2 = I) \\
&= \big(\cos(\tfrac{\gamma}{2})I + i\sin(\tfrac{\gamma}{2})XYX\big)\big(\cos(\tfrac{\delta-\beta}{2})I + i\sin(\tfrac{\delta-\beta}{2})XZX\big) && \text{(by Lemma 8.1)} \\
&= \big(\cos(\tfrac{\gamma}{2})I + i\sin(\tfrac{\gamma}{2})(-Y)\big)\big(\cos(\tfrac{\delta-\beta}{2})I + i\sin(\tfrac{\delta-\beta}{2})(-Z)\big) \\
&= \big(\cos(\tfrac{\gamma}{2})I - i\sin(\tfrac{\gamma}{2})Y\big)\big(\cos(\tfrac{\delta-\beta}{2})I - i\sin(\tfrac{\delta-\beta}{2})Z\big) \\
&= R_y(\tfrac{\gamma}{2})R_z(\tfrac{\delta+\beta}{2}) && \text{(by Lemma 8.1)}
\end{aligned}$$

□

From this claim, it immediately follows that

$$\begin{aligned}
e^{i\alpha}AXBXC &= e^{i\alpha}\cdot R_z(\beta)R_y\left(\tfrac{\gamma}{2}\right)R_z\left(\tfrac{\delta+\beta}{2}\right)R_z\left(\tfrac{\delta-\beta}{2}\right) \\
&= e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta) \\
&= U
\end{aligned}$$

which concludes the proof. □

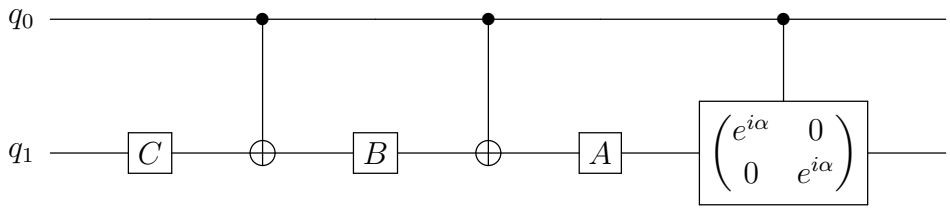
Now, given any unitary transformation U, our next goal is to understand how to implement C-U using only single-qubit operations and the CNOT gate, leveraging the result above. Consider two qubits q_0 and q_1, and construct the operators A, B and C as described by the corollary. Suppose that q_0 is the control qubit; then, the controlled-U operator should then work as follows:

- if q_0 is not set, q_1 must be left unchanged
- if q_0 is set, we must apply U to q_1

In the first case, we can leverage the fact that A, B and C satisfy

$$ABC = I$$

by the corollary, indeed the following circuit is already enough to construct C-U:



Since $U = e^{i\alpha}AXBXC$ by the corollary, we get that:

- if q_0 is not set, we apply $ABC = I$ to q_1
- if q_0 is set, we apply $e^{i\alpha}IAXBXC = U$ to q_1

However, the controlled application of $e^{\alpha}I$ is *not* a gate that we can use, since we want to only use CNOT's as controlled gates. Nevertheless, we can rewrite the controlled operator as follows

$$\begin{aligned} |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{i\alpha}I &= (|0\rangle\langle 0| + |1\rangle\langle 1| \otimes e^{i\alpha}) \otimes I \\ &= \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}\right) \otimes I \\ &= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \otimes I \end{aligned}$$

which means that the circuit can be rewritten as follows:

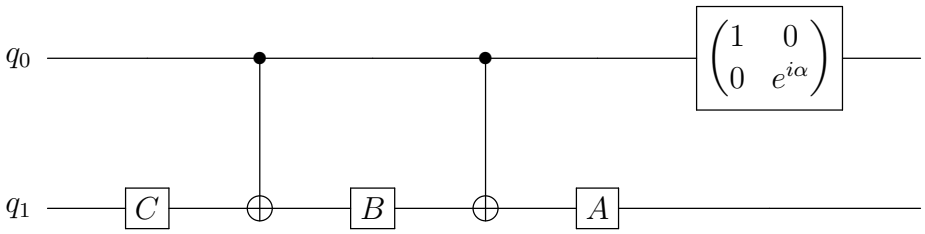


Figure 8.1: The circuit for the controlled-U operator.

Finally, the matrix obtained as a transformation for q_0 can be rewritten as

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} = e^{i\alpha/2}R_z(\alpha)$$

meaning it is just a z-axis rotation up to a global phase.

8.2.2 Multi-qubit conditioning

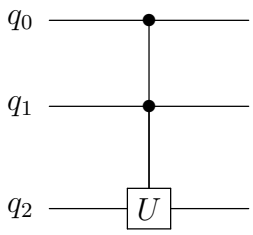
Now that we know how to condition the application of a single-qubit operator on one qubit, what about conditioning on *multiple* qubits? Given a unitary operator U acting on k qubits, we define the controlled operator $C^n(U)$ as follows

$$C^n(U)\,|x\rangle\,|\psi\rangle = |x\rangle\,U^{x_1 \ldots x_n}\,|\psi\rangle$$

where

- $|x\rangle$ is an n-qubit state
- x_1 through x_n are the *control* qubits of the operator
- $x_1, \ldots, x_n$ in the exponent of U means the *product* of the bits $x_1 \cdot \ldots \cdot x_n$, indeed we want to apply U on $|\psi\rangle$ only if *all* the bits of $|x\rangle$ are set

For now, let's restrict our focus on an easy example: suppose that U is a single-qubit operator, and we want to construct $C^2(U)$, depicted below



Even if we will not prove the following result, we can use this powerful property to construct a circuit as shown that computes $C^2(U)$.

Proposition 8.2

Every untary operator has a unitary square root.

This property implies that there exists an operator V such that

$$V^2 = U$$

and since it is unitary, it can be used as a gate in a quantum circuit.

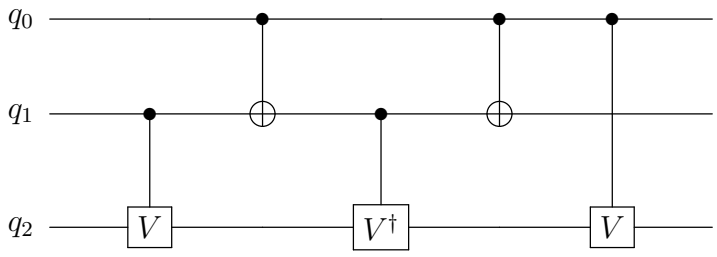


Figure 8.2: The quantum circuit that computes $C^2(U)$.

We see that this circuit computes $C^2(U)$ through the following table.

q_0^i	q_1^i	q_0^o	q_1^o	q_2^o
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	q_2
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$XX\|0\rangle = \|0\rangle$	$VV^\dagger q_2 = Iq_2 = q_2$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$V^\dagger V q_2 = Iq_2 = q_2$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$XX\|1\rangle = \|1\rangle$	$VVq_2 = V^2 q_2 = Uq_2$

This seeminly uninteresting quantum circuit is actually *very important*. What happens when $U = X$? It can be proven that

$$V = \frac{(1-i)(I+iX)}{2}$$

is such that $V^2 = X = U$. This quantum gate depicted below

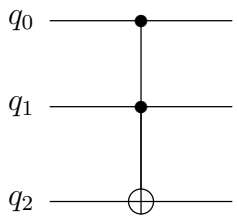


Figure 8.3: The quantum Toffoli gate.

is called **quantum Toffoli gate**, and as the name suggests there is also a classical Toffoli gate that computes exactly as we would expect:

$$T : \{0,1\}^3 \to \{0,1\}^3 : (a,b,c) \mapsto (a,b,c \oplus (a \wedge b))$$

We see that this binary function flips c only if both a and b are set to 1, since

$$\neg c = c \oplus 1$$

First, we shall show how the quantum Toffoli can be constructed using only the gates we allowed ourselves at the beginning of the chapter:

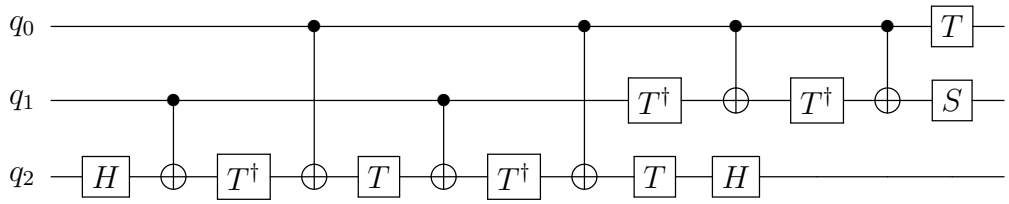


Figure 8.4: The quantum Toffoli gate written with only H, S, T and CNOT gates.

Note that the label T in this diagram represents the T gate

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

not the Toffoli gate! We will leave the correctness of this quantum circuit as an exercise for the reader.

Surprisingly, we can utilize the quantum Toffoli gate to construct $C^n(U)$. The idea is to simply chain multiple quantum Toffoli gates in order to enforce that all the control qubits

are set. To achieve this, we need an overhead of $n-1$ work qubits set to $|0\rangle$, as shown below.

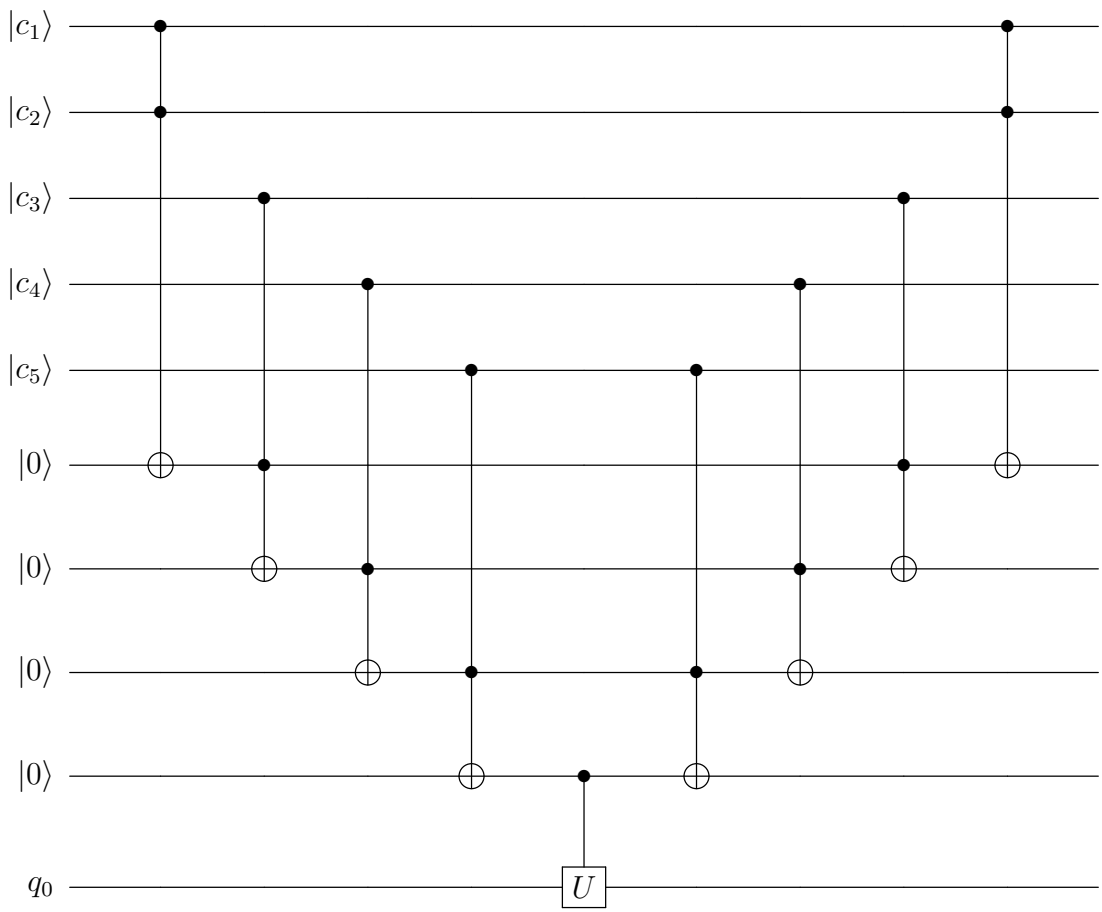


Figure 8.5: A quantum circuit implementing $C^n(U)$, with $n = 5$.

This circuit can be fully constructed with our restrictions, since it only uses quantum Toffoli gates and a single-control single-qubit U gate.

As a final note, it can be proven that this idea can be extended such that any unitary operator U acting on an arbitrary number of qubits can be

is this even true?

8.3 Universality

Now that we have introduced the Toffoli gate in the *quantum context*, it's a good opportunity to look at what the Toffoli gate can do in classical computing, and why it's actually special even in the **classical context**.

We already mentioned how quantum computation is **reversible**, since each operator is unitary. However, we also metioned that classical computaion admits reversible computation as well. It turns out that the Toffoli gate allows such goal. Indeed, we observe that

by associativity of the XOR it holds that

$$(c \oplus (a \wedge b)) \oplus (a \wedge b) = c \oplus (a \wedge b) \oplus (a \wedge b) = c$$

which directly implies that

$$T^2 = I \implies T = T^{-1}$$

This means that the Toffoli gate is completely invertible, and it is its own inverse.

But the Toffoli gate is not only special because it enables reversible classical computation. Indeed, above all its most important property is that it is **universal**.

Proposition 8.3: Universality of the Toffoli gate

The Toffoli gate is universal.

Proof. Since it is well known that the NAND gate is univesal, we just need to simulate the NOT and the AND operators with the Toffoli gate. Then, if we denote with $T_3(a, b, c)$ the third output of the Toffoli gate, we get that

- the NOT gate can be simulated as follows:

$$\forall x \in \mathbb{B} \quad T_3(1, 1, x) = x \oplus (1 \wedge 1) = x \oplus 1 = \neg x$$

- the AND gate can be simulated as follows:

$$\forall x, y \in \mathbb{B} \quad T_3(x, y, 0) = 0 \oplus (x \wedge y) = x \wedge y$$

□

This means that the Toffoli gate alone is enough to perform any classical and fully reversible computation.

But we do we need 3 inputs? It is easy to see that a 1-bit gate is definitely not enough to achieve classical reversibility. Our intuition would suggest that adding an *ancilla* wire that keeps track of the original input should be able to solve this issue. However, the following holds.

Theorem 8.2: Non-universality of classical 2-bit rev. logic

There exists a Boolean function that cannot be constructed from one and two bit reversible logic gates, and ancilla bits.

Proof. TODO □ todo?

This result concludes that the Toffoli gate cannot be improved any further. Therefore, from a classical viewpoint the construction presented in Section 8.2.2 is remarkable: in quantum computing we can construct the Toffoli gate from one and two qubit gates, and since reversibility is an intrinsic property of quantum operators, the following holds.

8.3.1 Exact quantum universality

Thus, in the end, is the set of quantum gates composed of

$$\{X, Y, Z, H, S, T, \text{CNOT}\}$$

truly universal? Well, we recall that we initially pretended Pauli matrices allowed us to construct any rotation matrix, however unfortunately this is not true. The problem is that there it is not possible to construct exact rotation matrices of arbitrary angles. Interestingly enough, if we were able to actually construct rotation matrices, most of the quantum gates in the set we chose would even be redundant:

- $X = HR_z(\pi)H$
- $Y = SHR_z(\pi)HS^\dagger$
- $Z = R_z(\pi)$
- $S = R_z(\pi/2)$
- $T = R_z(\pi/4)$
- $R_x(\theta) = HR_z(\theta)$
- $R_y(\theta) = SHR_z(\theta)HS^\dagger$

We see that with rotation matrices of arbitrary angles we only need the H gate and the CNOT gate to achieve universality!

This means that we cannot actually achieve **exact quantum universality** with the strategies we proposed. Unfortunately, while they are useful and can be utilized in practice for a wide range of unitrary transformations that have "nice" angle values, the ideas we laid down don't guarantee realizable exact quantum universality.

Each time we have discussed quantum universality, we have been careful to state that it was *exact*. Let's see how we can achieve it, and which issues arise. First, consider the following definition.

Definition 8.2: Two-level unitary matrix

A unitary matrix $U \in \mathbb{C}^{d\times d}$ is said to be **two-level** if it acts non-trivially on at most components.

An example of a 3×3 two-level unitary matrix is the following:

$$U := \begin{pmatrix} e^{i\alpha} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Indeed, we observe that

$$Ux = \begin{pmatrix} e^{i\alpha} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 e^{i\alpha} \\ x_2 \\ x_3 \end{pmatrix}$$

so only the first component of x was non-trivially transformed.

The following result shows what we can achieve with two-level unitary matrices.

Theorem 8.3: Two-level unitary decomposition

Any unitary transformation $U \in \mathbb{C}^{d\times d}$ can be written as the product of

$$U = V_1 \dots V_k$$

where each matrix V_i is a two-level unitary matrix, and $k \leq \frac{d(d-1)}{2}$.

We will not present the construction that allows such decomposition, but this theorem has an important consequence. Indeed, if we have a unitary operator U acting on an n qubit system, we have that $d = 2^n$, therefore the bound on k becomes

$$k \leq 2^{n-1}(2^n - 1) = O(4^n)$$

This bound is definitely *not* practical, and it already shows how difficult it is to construct arbitrary unitrary transformation exactly! However, we can go a step further.

Theorem 8.4

Any two-level unitary operator acting on the state space of n qubits can be constructed through $O(n^2)$ single-qubit and CNOT gates.

This result together with the previous theorem, provide the following corollary which states that single-qubit unitaries, together with CNOT gates, achieve **exact quantum universality**. However, there is a catch.

Corollary 8.3: Single-qubit and CNOT gates universality

Any unitary transformation acting on the state space of n qubits can be constructed through $O(n^2 4^n)$ single-qubit and CNOT gates.

This construction does not provide very efficient quantum circuits! However, it can be actually proven that this result is *close to optimal* if we demand exact quantum universality.

write thm section 4.5.4?

In the end, to find *reasonably fast* quantum algorithms we will clearly need a totally different approach.

8.3.2 Approximate universality

Now that we know exact quantum universality is not reasonably achievable, instead of continuing to chase it, we should focus on reaching an *approximate* form of universality. In particular, when we talk about **approximate quantum universality** we require that

it is possible to simulate an arbitrary unitary transformation *to arbitrary accuracy*. In this section, we will find a *discrete* set of gates which can be used to achieve it. First, we shall describe what it means to approximate a unitary transformation.

Definition 8.3: Approximation error

Given a *target* unitary transformation U, and an *actually implementable* unitary operator V, we define the **error** when V is implemented instead of U as

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

This measure of error has the interpretation that if $E(U,V)$ is small,, then anymeasurement performed on the state $V|\psi\rangle$ will give approximately the same measurement statistics as a measurement of $U|\psi\rangle$, for any initial state $|\psi\rangle$. Amazingly, the following results shows that with multiple unitaries the errors add at most linearly.

Proposition 8.4: Chaining inequality

If a sequence of unitaries $V_1, \ldots, V_m$ intends to approximate a sequence of unitaries $U_1, \ldots, U_m$, then

$$E(U_m U_{m-1} \ldots U_1, V_m V_{m-1} \ldots V_1) \leq \sum_{j=1}^{m} E(U_j, V_j)$$

Consider the following set of quantum gates, called the **standard set**

$$\{H, S, T, \text{CNOT}\}$$

Our final goal for this chapter will be to show that this set can achieve approximate quantum universality.

Lemma 8.2

Given any angle α, it holds that

$$\forall \varepsilon > 0 \quad \exists k \quad E(R_{\hat{n}}(\alpha), R_{\hat{n}}(\theta)^k) < \frac{\varepsilon}{3}$$

where $\overrightarrow{n} = (\cos(\pi/8), \sin(\pi/8), \cos(\pi/8))$ and θ is such that $\cos(\theta/2) = \cos^2(\pi/8)$.

Proof sketch. Since it holds that

- the T gate is a rotation by $\pi/4$ radians on the $\hat{z}$ axis on the Bloch sphere (up to an unimportant global phase)
- the transformation HTH is a rotation by $\pi/4$ radians around the $\hat{x}$ axis on the Bloch sphere

through Lemma 8.1 it can be showed that $THTH$ computes precisely $R_{\hat{n}}(\theta)$. Proving the repeated application of $R_{\hat{n}}(\theta)$ can be used to approximate to arbitrary accuracy any rotation $R_{\hat{n}}(\alpha)$ is outside the scope of our discussion. □

In particular, this lemma implies that H and T gates can be used to approximate any single-qubit unitary operation to arbitrary accuracy.

Theorem 8.5

Any single-qubit unitary operator U, there exists a unitary tranformation V composed of only H and T gates such that

$$\forall \varepsilon > 0 \quad E(U, V) < \varepsilon$$

Since H and T gates allow us to approximate, any single qubit unitary operator, we can leverage known constructions in order to approximate any m gate quantum circuit.

Corollary 8.4

Any m gate quantum circuit can be approximated to an accuracy ε with $\Theta(m^2/\varepsilon)$ gates of the standard set.

Even if this is a **quadratic** increase of the original size, it is already good enough for many applications. Rather remarkably, however, the following result is much more efficient.

Theorem 8.6: Solovay-Kitaev theorem

Any single-qubit unitary transformation U can be approximated to an accuracy ε using $O(\log^c(1/\varepsilon))$ gates of the standard set.

From this well-known result, it follows a **polylogarithmic** increase over the size of the original circuit, instead of the previous *quadratic*.

Corollary 8.5

Any m gate quantum circuit can be approximated to an accuracy ε using $O(m \log^c(m/\varepsilon))$ gates of the standard set.

9 Quantum Key Distribution

Why is Shor's algorithm so important? Most widely used **public-key cryptography** algorithms rely on the difficulty of one of the three following mathematical problems:

- the Integer Factorization problem
- the Discrete Logarithm problem
- the Elliptic-Curve Discrete Logarithm problem

In particular, all of the problems are currently *vary hard* to solve efficiently, and basically any modern cryptosystem is based on the fact that we don't know any fast-enough algorithm that is able to solve them. However, all of these could be easily solved on a sufficiently powerful quantum computer that runs Shor's algorithm, as we already seen.

explain details

Nevertheless, at the time of writing quantum computers lack the processing power to break widely used cyrptographic algorithm, so we are still safe from a world without crypto. However, because of the length of time required for migration to **quantum-safe cryptograpy**, are already desining new algorithms to prepare for the so called **Q-Day**, the day when current algorithms will be vulnerable to quantum computing attacks. For example, NIST already initiated a proces to quotes "solicit, evaluate and standardize one or more quantum-resistand public-key cryptography algorithms".

In this chapter we will cover

vedi de che parla prima de parla a vuoto

is a **Quantum Key Distribution (QKD)** protocol. In particular, we are interested in finding secure protocols that able to allow two parties to share a *private key* quantum-safely.

9.1 Background

9.1.1 The No-cloning theorem

Before explaining any protocol however, we need to talk about a theorem that we did not properly discuss in Section 1.3.2, namely the **No-cloning theorem**. In particular, we stated that there is no quantum transformation that copies any quantum state. Now that we have the mathematical tools to prove the theorem, we can actually show the real formulation of the theorem. But first, let's try to understand what it means to "clone" a qubit. Say that we have a qubit set to some unknown superposition of states $|\chi\rangle$ that we would like to clone; in other words, we need an ancilla qubit $|\gamma\rangle$ and some operator U (which must be unitary) such that

$$U(|\chi\rangle \otimes |\gamma\rangle) = |\chi\rangle \otimes |\chi\rangle$$

Essentially, the ancilla qubit is used to store the clone of $|\chi\rangle$. We observe that the initial state of the ancilla can be considered to be some fixed superposition, since the qubit $|\chi\rangle$ is *unknown* and we want U to work on any possible state. However, the No-cloning theorem states that such a transformation U cannot exist if the choice of the possible state for $|\chi\rangle$ isn't restricted enough.

Theorem 9.1: No-cloning theorem

Let $\mathcal{H}$ be a Hilbert space, and let $S \subseteq \mathcal{H}$ be a set of vectors. If S contains two distinct non-orthogonal vectors, there is no unitary operator

$$U : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$$

such that there exists a fixed vector $|\gamma\rangle \in \mathcal{H}$ such that

$$\forall\, |\chi\rangle \in S \quad U(|\chi\rangle \otimes |\gamma\rangle) = |\chi\rangle \otimes |\chi\rangle$$

Proof. By way of contradiction, suppose that there exists a unitary transformation U for which there exists a fixed $|\gamma\rangle \in \mathcal{H}$ that can be universally used as ancilla qubit to copy any $|\chi\rangle \in S$. Moreover, let $|\psi\rangle, |\phi\rangle \in S$ two distinct vectors such that $\langle\psi|\phi\rangle \neq 0$ meaning that they are non-orthogonal. Because U is unitary, by Proposition 2.9 we know that it must preserve the scalar product, meaning that

$$\forall x, y \in \mathcal{H} \quad \langle Ux|Uy\rangle = \langle x|y\rangle$$

In particular, this must hold for

$$x = |\psi\rangle \otimes |\gamma\rangle$$
$$y = |\phi\rangle \otimes |\gamma\rangle$$

however we see that

$$\begin{aligned}
&\langle U(|\psi\rangle \otimes |\gamma\rangle)|U(|\phi\rangle \otimes |\gamma\rangle)\rangle = \langle \psi \otimes \gamma | \phi \otimes \gamma \rangle \\
\iff &\langle \psi \otimes \psi | \phi \otimes \phi \rangle = \langle \psi \otimes \gamma | \phi \otimes \gamma \rangle \\
\iff &\langle \psi | \phi \rangle \langle \psi | \phi \rangle = \langle \psi | \phi \rangle \langle \gamma | \gamma \rangle \\
\iff &\langle \psi | \phi \rangle = 1 \qquad\qquad (\langle \phi | \psi \rangle \neq 0) \\
\iff &|\psi\rangle = |\phi\rangle
\end{aligned}$$

This means that $|\psi\rangle$ and $|\phi\rangle$ are actually the same vector, contradicting the fact that we chose them distinct in S ↯. □

The most important details of this formulation are the following:

- we require S to have at least two distinct non-orthogonal vectors
- the theorem is defined on a *set* of vectors S which may not necessarily be the whole Hilbert space considered

The second observation is crucial because depending on the choice of S "cloning" is *possible*. Indeed, suppose that we restrict our interest in the set of the only two states $|0\rangle$ and $|1\rangle$, i.e. $S = \{|0\rangle, |1\rangle\}$ In this scenario, we *can* actually define an operator such that

$$\exists\, |\gamma\rangle \quad \forall\, |\chi\rangle \in \{|0\rangle, |1\rangle\} \quad U(|\chi\rangle \otimes |\gamma\rangle) = |\chi\rangle \otimes |\chi\rangle$$

and it's just a CNOT with $y = |0\rangle$:

$$\text{CNOT}(|0\rangle \otimes |0\rangle) = |0\rangle \otimes |0\rangle$$

$$\text{CNOT}(|1\rangle \otimes |1\rangle) = |1\rangle \otimes |1\rangle$$

This operation is called **duplication**, and depending on the set S it can be performed without any issue. As a final note, we observe that the No-cloning theorem is usually stated as follows.

Corollary 9.1

There is no quantum transformation that copies an unknown quantum state.

This is just the particular case in which $S = \mathcal{H}$, and $\mathcal{H}$ obviously contains non-orthogonal pair of distinct states.

The hypothesis of the No-cloning theorem regarding the choice of S is used strategically by the BB84 protocol, as we will see in the next section.

9.1.2 Measurements

Before describing the next step of the protocol, we need to do a step back and talk about *measurement* in greater detail. Each time we discussed "measuring" we never actually mentioned that when we are *physically* performing a measurement what we are really

doing is applying a "polarizing lens" to the final state. For instance, this operation can be implemented through Faraday rotators, which leverage the **Faraday effect**:

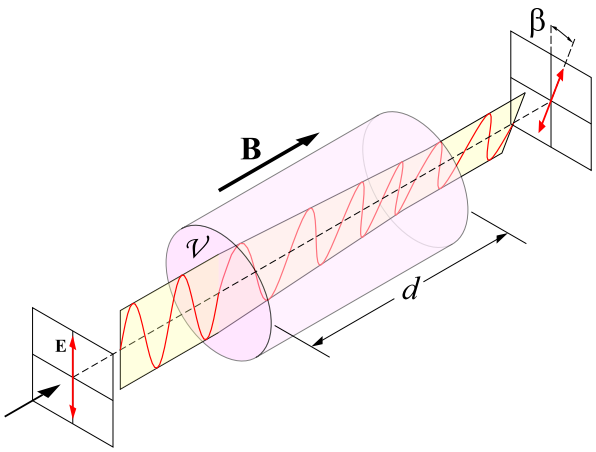


Figure 9.1: A Faraday polarization rotator.

This is important because it means that when we talk about measurement we need to define the "angle" at which we perform it. Mathematically speaking, this is equivalent to defining the **basis of choice** for the measurement — we always consider orthonormal bases only, so all the possible bases of a space are identical up to a rotation. Indeed, the importance of the choice of the basis can be illustrated through the BB84 protocol itself.

- If we have a qubit in the state $|\phi\rangle = |+\rangle$, we know that the probability of getting either $|0\rangle$ or $|1\rangle$ when measured in the Z basis is precisely 50% for both outcomes because
$$\Pr[\text{measure}(|\phi\rangle = |0\rangle)] = |\langle 0|+\rangle|^2 = \frac{1}{2}$$

- Now, consider the orthonormal basis formed by the vector $|+\rangle$ and $|-\rangle$ — which is usually called **X basis**; we observe that this space is just a rotation of 45° clockwise of the Z basis. This space is interesting: when we measure a qubit $|\phi\rangle$ set to the state $|0\rangle$ in the Z basis we get $|0\rangle$ with 100% certainty, but in this space we have that
$$\begin{aligned}\Pr[\text{measure}(|\phi\rangle = |+\rangle)] &= |\langle 0|+\rangle|^2 \\ &= \left|\frac{1}{\sqrt{2}}\langle 0|\left(|0\rangle + |1\rangle\right)\right|^2 \\ &= \left|\frac{1}{\sqrt{2}}(1+0)\right|^2 \\ &= \frac{1}{2}\end{aligned}$$
Indeed, as we would expect $|0\rangle$ and $|1\rangle$ in the X basis behave exactly as $|+\rangle$ and $|-\rangle$ do in the Z one.

We will see how the BB84 protocol takes advantage of this fact later in our discussion. The last detail that we want to mention about measurements in non-diagonal bases is that we can actually define unitary operators that allow us to perform measurements in the Z basis no matter the basis of choice, by first applying some particular unitary operator that creates a "map" between the two bases. For instance, say that we have some vector

$$|v\rangle = \alpha |+\rangle + \beta |-\rangle$$

defined in the X basis, and we want to measure this vector in the Z basis. This can be done by first applying the following operator:

$$U = |0\rangle \langle +| + |1\rangle \langle -|$$

In fact, it holds that

$$\begin{aligned} U|v\rangle &= (|0\rangle \langle +| + |1\rangle \langle -|)(\alpha |+\rangle + \beta |-\rangle) \\ &= |0\rangle \langle +| (\alpha |+\rangle + \beta |-\rangle) + |1\rangle \langle -| (\alpha |+\rangle + \beta |-\rangle) \\ &= \alpha |0\rangle + \beta |1\rangle \end{aligned}$$

We can actually generalize this idea: if we have two bases $\{f_i\}_{i=1}^n$ and $\{g_i\}_{i=1}^n$ an operator that maps each f_i to g_i for all $i \in [n]$ is exactly

$$U_{fg} = \sum_{i=1}^{n} |g_i\rangle \langle f_i|$$

Interestingly enough, we already know an operator that acts as a map between the X and the Z bases:

$$\begin{aligned} U &= |0\rangle \langle +| + |1\rangle \langle -| \\ &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \left[\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \right]^\dagger + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \left[\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \right]^\dagger \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= H \end{aligned}$$

This should not come as a suprise, since we already knew that

$$H|0\rangle = |+\rangle \qquad H|1\rangle = |-\rangle$$

9.2 The BB84 protocol

The key exchange procedure we will present is called **BB84 protocol**, and it was published in 1984 by Bennett and Brassard [BB14]. The protocol works as follows: suppose

that our two usual protagonists Alice and Bob need to share a key quantum-safely. At the beginning, Alice will choose two random binary strings $(4+\delta)n$-long — for some $\delta > 0$ sufficiently large — which are usually referred to as a and b. Then, depending on the bits of these two strings she will construct a superposition of states based on the following table:

a_i	b_i	Chosen state
0	0	$\|0\rangle$
1	0	$\|1\rangle$
0	1	$\|+\rangle$
1	1	$\|-\rangle$

For example, if her two random strings are

$$a = 01101$$

$$b = 00110$$

then she will construct the following superposition

$$|\psi\rangle = |0\rangle \otimes |1\rangle \otimes |-\rangle \otimes |+\rangle \otimes |1\rangle$$

Now, we will assume that Alice and Bob share two communication channels:

- one quantum channel, which can be eavesdropped by a third party Eve that wants to know their shared key
- one public classical *authenticated* channel — in particular Alice and Bob must be sure about the identity of the other

Now that Alice has constructed her superposition $|\psi\rangle$, she will send it to Bob which will receive $\mathcal{E}(|\psi\rangle)$, where $\mathcal{E}$ describes the quantum operation due to the combined effect of the channel's noise and Eve's actions — we will describe what Eve can do later in our discussion.

Bob will then proceed to generate a random string of $(4+\delta)n$ bits as well, which we will call b', and based on this very string he will measure the received quantum superposition — in particular, if $b'_i = 0$ he will use the Z base, and if $b'_i = 1$ he will employ the X one. In our example, assuming that Bob generated the random string

$$b' = 10111$$

he will measure something like

$$\text{measure}(|\psi\rangle) = |-\rangle \otimes |1\rangle \otimes |-\rangle \otimes |+\rangle \otimes |+\rangle$$

We observe that the first and the last states randomly collapsed to $|-\rangle$ and $|+\rangle$ respectively because they were measured in the wrong basis — indeed, the original string was $b = 00110$ and $b \oplus b' = 10001$. Bob then tries to reconstruct the original string a based on its b', thus getting the string

$$a' = 11100$$

When this process is complete, Bob publicly announces that he measured the superposition he received in the authenticated channel to Alice.

After Alice has heard that Bob has measured the state, she can proceed to send b itself *as it is* over the authenticated public channel to Bob — we will see why she is sure that she can perform this operation safely. Through b Bob can then discard all the bits of a' that were generated through wrong bits of b' — we observe that their remaining bits satisfy $a_i = a'_i$ for all kept indices i. This step is usually called **basis reconciliation**, and the expected number of bits kept by Bob after this step is

$$\mathbb{E}[\text{kept bits}] = \sum_{i=1}^{(4+\delta)n} 1 \cdot \Pr[b_i = b'_i] = \sum_{i=1}^{(4+\delta)n} \frac{1}{2} = \frac{1}{2} \cdot (4+\delta)n = \left(2 + \frac{\delta}{2}\right) n$$

At this point, Alice and Bob will agree on $2n$ bits to keep from the reconciled string (which can be done again through the public medium) — δ can be chosen sufficiently large so that there at least $2n$ bits to choose with exponentially high probability.

At this point, the protocol is basically finished, and the last step involves **error correction** which has to take into account both the possible noise of the quantum medium, and the potential actions of Eve. Thus, as final step Alice and Bob agree on a split of their $2n$ bits into 2 sets of n bits (choosen UAR), such that half of them will be used as the **shared key**, and the other half are used as *check bits*. In particular, Alice and Bob publicly share the check bits, such that

- if more than t bits disagree, they abort and re-try the protocol from the start
- if less that t bits disagree, the error rate is estimated and used to apply *error correction* algorithms to the key bits — we will not cover the details of the error correction procedures since they are not part of the key distribution protocol itself

9.2.1 Eve's attacks

Now, it's time to discuss the Eve's role in the protocol. Suppose that Eve has access to the quantum channel Alice and Bob are using, and wants to understand the key they are trying to share. At the beginning of the protocol Alice sends her $|\psi\rangle$ which encodes the states she generated through a and b, so what happens if Eve can see this qubit? Due to the No-cloning theorem, she cannot clone $|\psi\rangle$ because in our case we have that

$$S = \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$$

and half of the possible pairs are actually non-orthogonal. However, she could theoretically do is intercept $|\psi\rangle$, measure it herself, and send the qubit back to Bob acting as a *man-in-the-middle* — and Bob would have no way to know this actually happened. Nevertheless, nor Bob neither Eve know the sequence of bases Alice chose originally, so the best thing Eve can do is try randomly (exactly as Bob does in the protocol anyway). Hence, fix a state $|\psi\rangle_i$ of the states that constitute $|\psi\rangle$; the following can happen:

- Eve chooses the correct basis for $|\psi\rangle_i$: this means that she does not introduce any disturbance to the state Bob will receive, and Bob still has 50% chance of recovering Alice's original qubit

- Eve chooses the wrong basis for $|\psi\rangle_i$: this means that she introduces some disturbance to the state Bob will receive, in the sense that now Bob has only 25% chance of recovering Alice's original qubit

For instance, suppose that Alice sends $|+\rangle$; if Eve guesses the correct basis it holds that

$$\begin{aligned}\Pr[\text{Bob measures } |+\rangle \mid \text{Eve chooses } X] &= \Pr[\text{Bob measures } |+\rangle \mid \text{Bob chooses } X, \text{Eve sends } |+\rangle] \\ &\quad \cdot \Pr[\text{Bob chooses } X] \\ &\quad \cdot \Pr[\text{Eve sends } |+\rangle \mid \text{Eve chooses } X] \\ &= 1 \cdot \frac{1}{2} \cdot 1 \\ &= \frac{1}{2}\end{aligned}$$

(obviously, all the probabilities are conditioned under the fact that Alice sent $|+\rangle$). Differently, if Eve guesses the wrong basis we have that

$$\begin{aligned}\Pr[\text{Bob measures } |+\rangle \mid \text{Eve chooses } Z] &= \sum_{b \in \mathbb{B}} (\Pr[\text{Bob measures } |+\rangle \mid \text{Bob chooses } X, \text{ Eve sends } |b\rangle] \\ &\quad \cdot \Pr[\text{Bob chooses } X] \\ &\quad \cdot \Pr[\text{Eve sends } |b\rangle \mid \text{Eve chooses } Z]) \\ &= \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \\ &= \frac{1}{4}\end{aligned}$$

Note that 25% is *not* the chances that Bob has to recover $|+\rangle$ *in general*, because that is given by

$$\begin{aligned}\Pr[\text{Bob measures } |+\rangle] &= \sum_{B \in \{X,Z\}} (\Pr[\text{Bob measures } |+\rangle \mid \text{Eve chooses } B] \cdot \Pr[\text{Eve chooses } B]) \\ &= \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2} \\ &= \frac{3}{8}\end{aligned}$$

which is still less than $\frac{1}{2}$.

We observe that, on average, with big enough bit strings the chance that Eve correctly chooses each basis can be made exponentially low. Moreover, Eve's actions are exactly the reason why in the last step of the algorithm we perform a thresholded check on the error rate: if the error rate is too high (some noise has to be expected) on average it probably means that someone eavesdropped on the channel.

Is this everything Eve can do? The attack we presented has a very high chance of introducing too much disturbance and being detected by the error rate check step, so for Eve to have some chance of not being detected she needs to **avoid introducing disturbance completely**. The idea is based on the fact that she actually does not need to apply the

full No-cloning theorem: suppose that there exists some unitary operator U that on input $|\psi\rangle \otimes |x\rangle$ — for some state $|x\rangle$ — it behaves as follows:

$$U(|\psi\rangle \otimes |x\rangle) = |\psi\rangle \otimes |y\rangle$$

In other words, U leaves $|\psi\rangle$ unchanged and turns $|x\rangle$ into $|y\rangle$. For now, consider $|\psi\rangle$ as 1 single qubit. If such operator exists, Eve could use it in order to try to measure $|y\rangle$ in a later moment and gain some information about $|\psi\rangle$, without ever measuring the latter directly.

This idea seems compelling, however for Eve's attack to be effective U must compute as follows:

$$U(|\psi\rangle \otimes |x\rangle) = |\psi\rangle \otimes |y\rangle$$
$$U(|\phi\rangle \otimes |x\rangle) = |\phi\rangle \otimes |y'\rangle$$

where $|y\rangle$ and $|y'\rangle$ must be **different**. In this way

- U leaves $|\psi\rangle$ unchanged, and the latter can be sent to Bob without anyone noticing
- U behaves differently depending on the input superposition, i.e. $|y\rangle \neq |y'\rangle$, otherwise she cannot distinguish between $|\psi\rangle$ and $|\phi\rangle$ upon measurement — to be clear, this has nothing to do with *entanglement*, in fact for Eve it is sufficient to look at the second qubit and infer what the first must have been based on the fact that $|y\rangle$ and $|y'\rangle$ are distinguishable

Indeed, if such U exists she can recover what the original qubit Alice sent only looking at $|y\rangle$ and using the possible values of the ancilla qubit as "lookup table". Let's see if she can employ this strategy.

Suppose that Alice sends some qubit which can be either $|\psi\rangle$ or $|\phi\rangle$, where

$$|\psi\rangle, |\phi\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$$

Since U must be unitary, by Proposition 2.9 we know that U preserves the scalar product, meaning that

$$\begin{aligned}
& \langle U(|\psi\rangle \otimes |x\rangle) | U(|\phi\rangle \otimes |x\rangle)\rangle = \langle \psi \otimes x | \phi \otimes x\rangle \\
\iff & \langle \psi \otimes y | \phi \otimes y'\rangle = \langle \psi \otimes x | \phi \otimes x\rangle \\
\iff & \langle \psi | \phi\rangle \langle y | y'\rangle = \langle \psi | \phi\rangle \langle x | x\rangle \\
\iff & \langle \psi | \phi\rangle \langle y | y'\rangle = \langle \psi | \phi\rangle
\end{aligned}$$

We observe that between all the possible choices of pairs of $|\psi\rangle$ and $|\phi\rangle$, half of them are such that $\langle \psi | \phi\rangle \neq 0$, i.e. $|\psi\rangle$ and $|\phi\rangle$ are *non-orthogonal*. This means that we can simplify the last equality even more, obtaining that

$$\begin{aligned}
& \langle \psi | \phi\rangle \langle y | y'\rangle = \langle \psi | \phi\rangle \\
\iff & \langle y | y'\rangle = 1 \qquad\qquad (\langle \psi | \phi\rangle \neq 0) \\
\iff & |y\rangle = |y'\rangle
\end{aligned}$$

Once again, through an argument fairly similar of the one we used for the No-cloning theorem, this proves that such an operator U cannot exist, which means that there is no way for Eve to gain any information at all without introducing disturbance.

To be precise, there are more subtle attacks that do employ *quantum entagnlement*, such as **optimal intercept–resend tradeoffs**, where Eve applies a unitary operator that *partially entangles* the signal with a probe to trade a small amount of information for a small disturbance. It can be proven that the BB84 protocol is also resistant against these type of attacks, however this is beyond the scope of our discussion.

Lastly, we observe that BB84 requires Alice to wait before hearing back from Bob that he *actually measured* $|\psi\rangle$. This is to prevent that Alice sends b too early, so that Eve could use the latter to decode $|\psi\rangle$ into the original string a without any issue. Therefore, we require Alice to wait for Bob's measurement so that there is no way for Eve to recover a after looking at b becasuse the original superposition is definitively destroyed — unless she can go back in time!

9.3 Bell's inequalities

Imagine we perform the following experiment: there are two parties involved, namely Alice and Bob, and a third party Charlie which prepares two particles. It does not matter how he prepares the particles, the only thing that matters is that he is capable of repeating the experimental procedure which he uses. Once he has completed the preparation procedure, he sends one particle to Alice, and one to Bob.

After Alice receives her particle, she performs a measurement on it. In particular, let's imagine that she has available two different measurement apparatuses, which measure different physical properties that we will label as P_Q and P_R, respectively. For simplicity, we can suppose that the measurements can each have one of two outcomes, namely $+1$ or -1, and the outcome of measuring property P_Q and P_R are described by two random variables $Q, R \in \{+1, -1\}$ respectively. It is also important to assume that Alice does not know in advance which measurement she will choose to perform, instead when she receives her particle she flips a coin and chooses which property to measure accordingly. Similarly, suppose that Bob is capable of measuring one of two properties, P_S or P_T, and the outcomes of such measurements are also described by random variables $S, T \in \{+1, -1\}$. Again, the choice of the measurement apparatus is random, and we also assume that Alice and Bob do their measurements *at the same time*. This implies that there is no way for Alice's measurement to disturb the result of Bob's measurement (and vice versa) since physical influences cannot propagate faster than light. Interestingly, we can obtain the following result.

Proposition 9.1: Bell inequality

It holds that

$$\mathbb{E}[QS] + \mathbb{E}[RS] + \mathbb{E}[RT] - \mathbb{E}[QT] \leq 2$$

Proof. Let q and r be two outcomes of the random variables Q and R, respectively. Since $q, r, \in \{+1, -1\}$, either $q = r$ or $q = -r$, therefore either $q - r = 0$ or $q + r = 0$. Define

$$p(q, r, s, t) := \Pr[Q = q, R = r, S = s, T = t]$$

Then, we have that

$$\begin{aligned}
\mathbb{E}[QS] + \mathbb{E}[RS] + \mathbb{E}[RT] - \mathbb{E}[QT] &= \mathbb{E}[QS + RS + RT - QT] \\
&= \sum_{q,r,s,t \in \{+1,-1\}} p(q,r,s,t) \cdot (qs + rs + rt - qt) \\
&= \sum_{q,r,s,t \in \{+1,-1\}} p(q,r,s,t) \cdot [(q+r)s + (r-q)t] \\
&\leq \sum_{q,r,s,t \in \{+1,-1\}} p(q,r,s,t) \cdot 2 \\
&= 2 \cdot \sum_{q,r,s,t \in \{+1,-1\}} p(q,r,s,t) \\
&= 2
\end{aligned}$$

□

This result is also known as the **CHSH inequality**, named after the initials of its four discoverers. Nevertheless, it is generally referred to as the Bell inequality since it is part of a larger set of inequalities known as **Bell inequalities**, named after the work of Bell [Bel64] published in 1964.

By repeating the experiment multiple times, Alice and Bob can determine each term on the LHS of the Bell inequality — for example, after a set of experiments they get a sample of values for Q and S, and by multiplying the results they get multiple estimates of QS which can subsequently be averaged out to get $\mathbb{E}[QS]$. We must underline that in the experiment we described there is *no quantum mechanics involved*: the only thing we did was evaluating expected values and doing calculations with random variables by following a seemingly logical reasoning.

Now, we are going to introduce quantum mechanics in the picture and see what happens with our experiment. Suppose that Charlie prepares a quantum system of two entangled qubits in the Bell state

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

and gives the two qubits to Alice and Bob. To obtain the same exact numbers of the experiment we outlined, we need to consider some properties that Alice and Bob can measure which ultimately yield ± 1 as possible outcomes. Fortunately, we do know such observables and they are the Pauli matrices. In fact, by Postulate 3.4 we know that an observable is a self-adjoint operator, and we already know that all Pauli matrices are self-adjoint. Let's first consider the Z Pauli matrix:

- it's easy to prove that its eigenvalues are exactly $+1$ and -1
- it can also be easily calculated that $|0\rangle$ is an eigenvector of Z associated to $+1$, and $|1\rangle$ is an eigenvector of Z associated to -1 (by solving $Z - \lambda I = 0$ for $\lambda \in \{-1, +1\}$)

This matrix seems to have everything we need for our purposes, so let's set the random variable Q equal to Z. Now, we need to find another observable (i.e. another self-adjoint

operator) to assign to Alice, which also has $+1$ and -1 as eigenvalues and that *does not commute* with Q, meaning that

$$QR \neq RQ$$

This is a crucial detail because we want to express the fact that the order of measurements matters, otherwise the outcomes could be modeled by a single classical random variable. This is easy enough by again turning our attention to the Pauli matrices, in particular we just need to set $R = X$, in fact

- the eigenvalues of X are $+1$ and -1
- $|+\rangle$ is an eigenvector of X associated to $+1$, and $|-\rangle$ is an eigenvector of X associated to -1 (obtained by solving $X - \lambda I = 0$ for $\lambda \in \{-1, +1\}$)
- it can be easily shown that

$$ZX = -XZ$$

The only thing left to define are S and T. For now, we will set them as follows

$$S = \frac{Z - X}{\sqrt{2}} \qquad T = \frac{Z - X}{\sqrt{2}}$$

without justifying our choice — we will provide a reason later in our discussion. Most importantly, it can be proven that the eigenvalues of two matrices are still ± 1.

Our goal now will be to evaluate all the terms of the Bell inequality, and check what quantum mechanics expects as outcome. We are not going into the details of all the calculations — its a quite tedious task and definitely outside our interests. We will only outline the general idea: for example, to evaluate $\mathbb{E}[Q \otimes S \mid |\Psi^-\rangle]$ we just need to follow what we described in Proposition 3.1

da copiare

Finally, after calculating all 4 terms we end up with the following results:

$$\langle QS \rangle = \frac{1}{\sqrt{2}} \quad \langle RS \rangle = \frac{1}{\sqrt{2}} \quad \langle RT \rangle = \frac{1}{\sqrt{2}} \quad \langle QT \rangle = -\frac{1}{\sqrt{2}}$$

which ultimately yield that

$$\langle QS \rangle + \langle RS \rangle + \langle RT \rangle - \langle QT \rangle = 2\sqrt{2}$$

What? The Bell inequality told us that the LHS could not ever exceed 2, however quantum mechanics predicts that this sum is equal to $2\sqrt{2}$! What is going on?

Short answer: we are not sure. Over the years, multiple experiments have been done to check the prediction of quantum mechanics versus the Bell inequality, and the results are resoundingly in favor of the quantum mechanical outcome. Most notably, in 2022 the Nobel Prize in Physics was awarded to Aspect, Clauser and Zeilinger for having established experimentally the violation of Bell inequalities. In simpler terms, the Bell inequality it *not* obeyed by Nature.

What does this mean? It means that at least one of the assumptions that went into the derivation of the Bell inequality must be incorrect. However, we notice that in the

experiment we described we relied upon very few assumptions. Among all the various hypotheses on which assumptions are wrong that have been made over the years, two arguments mainly stand out.

1. Assumption of **realism**: this assumes that the physical properties P_Q, P_R, P_S and P_T have definite values Q, R, S and T which exist independent of observation. In fact, our intuition would suggest that every measurement outcome has a definite value *before* it is measured: when we say that
$$Q, R, S, T \in \{+1, -1\}$$
we are also presupposing that the properties to be measured exist before we even perform the measurements. Classically, through measurement we are only looking at definite values that were "already there".

2. Assumption of **weak locality**: this assumes that measurement performed by Alice does not influence the result of Bob's measurement. In other words, this assumes that Alice's choice of measurement cannot change Bob's values, and vice versa. We want to underline that with this assumption we are not talking about the *outcomes*, only about the *choice* of which measurement apparatus one party selects.

Together, these two assumptions are known as the assumptions of **local realism**. Clearly they seem to be plausible assumptions that match our everyday experience, yet the Bell inequalities show that at least one of the two is incorrect. Most of the physicists believe that the assumption to be dropped is realism, indeed current quantum mechanics believes that there are no values of Q, R, S and T inside the particles and we can only talk about possible measurement outcomes along with their probabilities, but such outcomes do not exist before our measurement. Indeed, this is exactly what we have been assuming from the start: we consider *superposition* of states that collapse only after we have performed our measurement.

To be precise, there is a third possible assumption to consider that we might deem false, which is called **measurement independence**, through which we assume that there exists the possibility of free choice between the measurements that Alice and Bob can perform. However, if this presumption if false, it implies that *free will does not exist* and there are some kind of "hidden variables" that know what settings Alice and Bob will choose in advance. Nevertheless, to this day almost no physicist believes this assumption to be false, so we will only focus on the first two.

As a final note, the choice of S and T was lead by the following bound proved by Tsirelson [Tsi80] in 1980.

Theorem 9.2: Tsirelson bound

Given 4 dichotomic observables A_0, A_1, B_0, B_1 with outcomes $+1$ and -1 such that
$$\forall i, j \in \{0, 1\} \quad A_i B_j = B_j A_i$$
it holds that
$$\langle A_0 B_0 \rangle + \langle A_0 B_1 \rangle + \langle A_1 B_0 \rangle - \langle A_1 B_1 \rangle \leq 2\sqrt{2}$$

The choices for S and T are just the matrices that maximize this bound, given the choices for Q and R.

To this day, it is not known if the world follows either one of them with certainty, but this discrepancy between the classical and the quantum model must imply that the world is not *locally realistic*. Modern quantum mechanics believes in weak locality but not in realism, and interestingly enough such assumption can actually be leveraged in order to design a more quantum-secure key distribution protocol!

Consider again the experiment performed by Alice and Bob, and let $\mathcal{A}$ be the quantity

$$\mathcal{A} := \langle QS \rangle + \langle RS \rangle + \langle RT \rangle - \langle QT \rangle$$

Can an eavesdropper Eve disturb this experiment without any of the two parties involved notice? Recall that the choice of measurements performed by Alice and Bob are executed *randomly*, which means that Eve has no way to know in advance which setting Alice and Bob will pick — not even Alice and Bob do! Therefore, if Eve wants to cheat successfully she must find a way to arrange the context such that she knows all the possible outcomes in advance regardless of the choices of the measurement apparatus performed by the two parties. In other words, Eve must be able to set things up in such a way that she can predict all possible outcomes ahead of time. However, because we are assuming that Nature is weakly local, Eve cannot influence nor Alice's neither Bob's outcomes after they chose which property to measure. This must imply that the only chance she has to cheat is to have some way of "assigning" an outcome to every possible setting *in advance*. This is exactly **realism**, and by Bell inequality any system that is locally realistic must yield a value of $\mathcal{A} \leq 2$.

This is the core idea of the **E91 protocol**, published by Ekert [Eke91] in 1991: suppose that Alice and Bob take a random portion of their qubits and use it in order to compute the quantity $\mathcal{A}$. Then, if we set Q, R, S and T as we outlined previously we are guaranteed that, without any external disturbance in the setting, the value of $\mathcal{A}$ will be

$$\mathcal{A} \approx 2\sqrt{2}$$

This means that if Alice and Bob measure a value $\mathcal{A} \leq 2$, then their correlations could have come from a locally realistic (and therefore *predetermined*) model. And if their correlations could come from such a model, then Eve could have predicted all outcomes in advance, so their exchange would not be quantum-safe anymore and should be aborted. Hence, by chekcing the value of $\mathcal{A}$ Alice and Bob can choose to proceed with the exchange of the key or abort their attempt and try again.

10 Quantum Error Correction

TODO introduction

10.1 Repetition codes

10.1.1 Three bits bit flip code

Before discussing error-correction in quantum contexts, let's present what how the classical world handles noise in communications. Suppose we have two parties, Alice and Bob, and say that Alice wants to send Bob some bit $b \in \mathbb{B}$. However, the channel they has at their disposal is **noisy**, meaning that with some probability p the bit she sends will be flipped and Bob will receive the wrong information — we will assume that each flip is independent of each other. How can Alice raise the odds of Bob getting the bit she originally sent? What she needs is some type of *redundancy*, and the most straightforward and way to achieve it is clearly **repetition**.

Therefore, suppose Alice now sends 2 bits to Bob instead of one, such that both bits are equal to the original b she intended to sent

$$b = b_0 = b_1$$

Her idea is to make Bob able to recover the original b by looking at both b_0 and b_1. However, there is an issue with this idea: since both b_0 and b_1 could flip during the transmission, its easy to see that if $b_0 \neq b_1$ Bob has no way to determine what b was. This suggests that what Alice needs to provide is *more* information!

Suppose that Alice now sends b_0, b_1 and b_2 such that

$$b = b_0 = b_1 = b_2$$

and for now, let's assume that b_0 flipped during the transmission. Then, Bob will see that

$$b_0 \neq b_1 = b_2$$

which ultimately helps him determine that the value Alice wanted to send is contained in the bits b_1 and b_2, and b_0 flipped because of the noise. For a more practical example, if Alice wants to send 0 to Bob she transmits 000 through the noisy medium, but whenever Bob receive 100 he will infer that what Alice actually sent was 000, therefore the bit he was meant to receive was 0.

This idea is called *majority voting*, as Bob decides how to recover the bit by looking at the value of the bits that appears more often, however its easy too see a very big flaw of this approach: with 2 or more bit flips the majority voting fails! If Bob receives 110, by majority he will infer that we was meant to receive 1 from Alice, however he cannot be sure that what really happened is that both b_0 and b_1 flipped through the transmission and what Alice originally sent was 0. In general, we have that

$$\begin{aligned}\Pr[\text{recovering the wrong bit}] &= \Pr[\text{at least 2 two bits flipped}] \\ &= \Pr[\text{2 bits flipped}] + \Pr[\text{3 bits flipped}] \\ &= 3p^2(1-p) + p^3 \\ &= 3p^2 - 3p^3 + p^3 \\ &= 3p^2 - 2p^3\end{aligned}$$

This means that as long as

$$3p^2 - 2p^3 < p \iff p < \frac{1}{2}$$

this error-correction code improves reliability. The codes of these type are called **repetition codes**, as they rely on repeating the same information multiple times for increase the probability of transmitting the correct information.

10.1.2 Three qubits bit flip code

Can we replicate the same idea in a quantum setting? More specifically, suppose that Alice wants to send a qubit

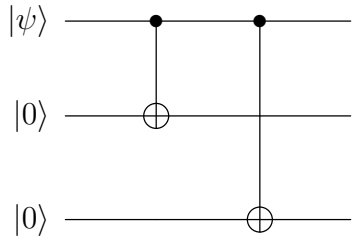
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

to Bob through a quantum channel, however said channel is noisy and might introduce errors with probability p at each transmission they perform. Because of the peculiarities of the quantum worlds, we know there are some important differences between classical and quantum information that require new ideas in order to apply the same redundancy technique presented:

- first and foremost, the No-cloning theorem forbids the cloning of a quantum state multiple times, so it is not as straightforward as in the classical context to produce the needed redundancy
- in the quantum context values are **continuous**, and errors are continuous as well, meaning that determining which error occurred would appear to require infinite precision — more details about this will be discussed later
- in classical error-correction Bob can just check the values of the received bits and that's it, however upon receiving qubits Bob cannot simply **measure** them to "look

at their value", otherwise he would destroy all the information making the recovery impossible

Fortunately, we can circumvent all of these problems with some clever ideas. First, consider the following quantum circuit:



Let's analyze what this circuit does:

$$\begin{aligned}
&|\psi\rangle \otimes |0\rangle \otimes |0\rangle \\
=&(\alpha |0\rangle + \beta |1\rangle) \otimes |0\rangle \otimes |0\rangle \\
=&(\alpha |00\rangle + \beta |10\rangle) \otimes |0\rangle \\
\xrightarrow{\text{CNOT}_{(q_0,q_1)}}&\text{CNOT}(\alpha |00\rangle + \beta |10\rangle) \otimes |0\rangle \\
=&(\alpha \text{CNOT} |00\rangle + \beta \text{CNOT} |10\rangle) \otimes |0\rangle \\
=&(\alpha |00\rangle + \beta |11\rangle) \otimes |0\rangle \\
\xrightarrow{\text{CNOT}_{(q_0,q_2)}}&\alpha |000\rangle + \beta |111\rangle
\end{aligned}$$

This is exactly what we needed: we started with a state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ and we ended up with

$$|\hat{\psi}\rangle = \alpha |000\rangle + \beta |111\rangle$$

which yields the redundancy we need in order to utilize the repetition strategy.

Now, suppose that Alice sends each of the three qubits through the noisy quantum channel, and each qubit will experience the effect of the noise, independently, with probability p. But what is this effect in practice? So far, we did not mention the word *flip*, because we actually need to define what a *flip* even is in this context. Well, if a "classical bit flip" is an application of a NOT operator on some bit b_i, it makes sense to define a "quantum bit flip" as the application of the X operator of some q_i sent, analogously.

However, this immediately shows that there is a **continuous spectrum** of possible errors that can occur to the qubits — namely every possible linear transformation — and apart from the X gate itself none of them have a classical analogue! For now, let's just focus on the X operator, and suppose that *at most one* bit flip occurred on the qubits Alice sent to Bob. How can Bob recover the original message? Consider the following table:

Projector	Error occured
$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$	No error occurred
$P_1 = \|100\rangle\langle 000\| + \|011\rangle\langle 011\|$	First bit flipped
$P_2 = \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$	Second bit flipped
$P_3 = \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$	Third bit flipped

This table contains 4 projectors that Bob can apply to discover which qubit, if any, flipped during the transmission. We underline that Bob *must* use these projectors in order to understand which bit flip happend, because he cannot measure what he received. For instance, say that the first bit flipped during the transmission, i.e. Bob receives

$$|\hat{\psi}\rangle = \alpha|100\rangle + \beta|011\rangle$$

Then, when he applies P_1 to $|\hat{\psi}\rangle$ he discovers that

$$\begin{aligned}\langle\hat{\psi}|P_1\hat{\psi}\rangle &= \langle\hat{\psi}|(|100\rangle\langle 000| + |011\rangle\langle 011|)(\alpha|000\rangle + \beta|111\rangle)\rangle \\ &= \langle\hat{\psi}|\alpha|100\rangle + \beta|011\rangle\rangle \\ &= \langle\hat{\psi}|\hat{\psi}\rangle \\ &= 1\end{aligned}$$

This means that Bob is sure that the second bit flipped, hence the original message can be recovered flawlessly by flipping the second qubit received. Again, this error-correction procedure works perfectly, provided that bit flips occur on at most one qubit per message, so reliability still requires that $p < 1/2$.

TODO

add slide 14 and 15

10.1.3 Three qubits phase flip code

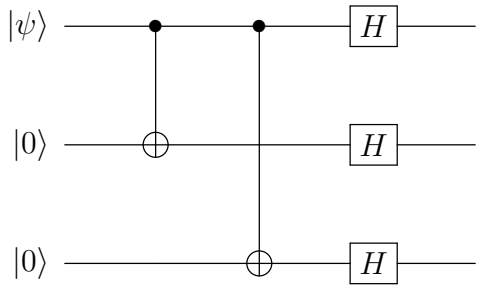
As previously mentioned, the case of the X gate is "easy" to solve, in the sense that it requires no significant innovation w.r.t. any classical context. However, what if instead of performing an application of the X operator with probability p, our noisy channel applies the Z transformation instead? When Alice sends her qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ it is transformed with probability p into

$$Z|\psi\rangle = \alpha|0\rangle - \beta|1\rangle$$

As previously mentioned, this scenario has no classical analogue, but it is still easy to handle. In fact, we already know a key fact: the Z operator acts like a standard bit flip in the Z basis, i.e.

$$Z|+\rangle = |-\rangle \qquad Z|-\rangle = |+\rangle$$

This suggests that we can still employ the same redundancy strategy of the previous section, provided that we perform a change of basis. Luckily, we already know a matrix that performs the change from the X to the Z bases, as discussed in Section 9.1.2, namely the Hadamard operator! Therefore, all Alice has to do is send the following:



meaning that at the end of the circuit we will get exactly:

$$\begin{aligned}
&|\psi\rangle \otimes |0\rangle \otimes |0\rangle \\
\xrightarrow{\cdots} &\dots \\
=&\alpha\,|000\rangle + \beta\,|111\rangle \\
\xrightarrow{H^{\otimes 3}(q_0,q_1,q_2)} &H^{\otimes 3}(\alpha\,|000\rangle + \beta\,|111\rangle) \\
=&\alpha H^{\otimes 3}\,|000\rangle + \beta H^{\otimes 3}\,|111\rangle \\
=&\alpha\,|+++\rangle + \beta\,|---\rangle
\end{aligned}$$

Then, Bob can discover if a qubit phase flip occured through the following table:

Projector	Error occured
$P_0 = \|+++\rangle\langle+++\| + \|---\rangle\langle---\|$	No error occurred
$P_1 = \|-++\rangle\langle-++\| + \|+--\rangle\langle+--\|$	First phase flipped
$P_2 = \|+-+\rangle\langle+-+\| + \|-+-\rangle\langle-+-\|$	Second phase flipped
$P_3 = \|++-\rangle\langle++-\| + \|--+\rangle\langle--+\|$	Third phase flipped

Once Bob learns which qubit phase flipped (if any) he can apply a Z operator to the corresponding qubit. We say that the channel that applies the X matrix and the one that applies the Z martix are **unitarily equivalent**, since there is a unitary operator — the Hadamard gate, in this case — such that the action of one channel is the same as the other, provided that the first channel is preceded by U and followed by $U^\dagger$ in every operation performed — we will leave the proof of this fact as an exercise.

TODO

i have no idea what slide 19 is saying

10.1.4 The Shor code

The last quantum repetition code we will present is called **Shor code**, after its inventor, and it is a simple code that protecs againts the effects of at most one *arbitrary Pauli error* on any qubit. Not surprisingly, the code is a combination of the three qubit bit and phase flip codes we saw earlier. In particular, we observe that in the bit flip code the quantum circuit acted on $|0\rangle$ and $|1\rangle$ as follows:

$$|0\rangle \xrightarrow{\text{bit flip code}} |000\rangle$$

$$|1\rangle \xrightarrow{\text{bit flip code}} |111\rangle$$

Similarily, the circuit for the phase flip acted as follows:

$$|0\rangle \xrightarrow{\text{phase flip code}} |+++\rangle$$

$$|1\rangle \xrightarrow{\text{phase flip code}} |---\rangle$$

Then, the Shor code applies the following transformation:

$$|0\rangle \xrightarrow{\text{phase flip code}} |+++\rangle \xrightarrow{\text{bit flip code}} \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)^{\otimes 3}$$

$$|1\rangle \xrightarrow{\text{phase flip code}} |---\rangle \xrightarrow{\text{bit flip code}} \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle)^{\otimes 3}$$

Indeed, this code is able to detect any Pauli error on any qubit, meaning that it can detect if either

- a bit flip
- a phase flip
- both a bit and a phase flip

occurred on at most one transmitted qubit — and it also works by switching the order of the concatenated codes. This is because

- bit flips correspond to σ_x
- phase flip correspond to σ_z
- $\sigma_y = i\sigma_x\sigma_z$, so we can correct Y transformations as well

Therefore, we just need to apply the procedures for detecting bit and phase flips one after the other in order to retrieve the original message.

We note that the fact that the Shor code actually enables the correction of combined bit and phase flip errors on a single qubit suggests that we could use the Shor code for more complicated type of errors. However, we might expect some additional work to be done in order to be protected against **arbitrary errors**, however quite suprisingly we can actually prove that this code detects any possible error with *no additional work required*! At the beginning of our discussion we said that there is a *continuum* of errors that may occur on a single qubit, which may seem to imply that we need much more sophisticated quantum error-correction procedures, however such continuum can actually be handled by correcting only a *discrete* subset of it. All the other possible errors are corrected automatically. The discretization of the errors is central to why quantum error-correction works, and should be regarded in contrast to classical error-correction for analog systems, where no such discretization of errors is possible.

Theorem 10.1

Any quantum error-correcting code that corrects at most k Pauli errors on at most k qubits will also correct an arbitrary quantum operation on thos qubits.

Idea of the proof. TODO □

pag 434 of the book

10.2 Linear codes

Let's go back to the classical scenario for a moment. So far we only discussed codes that involve repetition of the same information multiple times in order to recover the original message. However, we saw how the repetition codes we presented fail whenever the number of error is two or more. In particular, this occurs because the majority rule cannot be applied reliably anymore. Nevertheless, one might suggest that by simply sending *even more* repetition of the same information we could increase the chances of recovering the original message, which is indeed true. However, we see that

- when the information to transfer is more than 1 bit the situation becomes much more chaotic
- a "good" error-correcting code should also be able to provide a good balance between the transmitted bits and the redundancy bits

Let's give some terminology: each message sent through an error-correcting code is called **codeword**, and they are assumed to have always the same length — for instance, in the repetition code we analyzed previously the length of each codeword was 3. Then, we have the following definition.

Definition 10.1: Code rate

Given an error-correcting code that has codewords of length n, such that each codeword contains k *information symbols* and $n-k$ *reduncancy symbols*, we define the **code rate** of the code as

$$R = \frac{k}{n}$$

It's easy to see that a "good" code is required to have R as close to 1 as possible, in order to reduce the number of *redundancy symbols*. However, any repetition-based error-correcting code has a rate significantly smaller than 1, in fact as the number of repetitions required to correct more errors increases, the rate approaches 0! In this section we present a more rubust type of error-correcting codes, called **linear codes**.

Definition 10.2: Linear code

A **linear code** encoding k bits of information into an n bit code space — written as $[n, k]$ — is an error-correcting code specified by a matrix $G \in \mathbb{Z}_2^{n\times k}$ called **generator matrix**, such that each k bit message x is encoded as follows

$$\text{Enc} : \mathbb{Z}_2^{k\times 1} \to \mathbb{Z}_2^{n\times 1} : x \mapsto Gx$$

where the columns of G are linearly independent.

For instance, consider the following generator matrix:

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

This matrix has dimensions 3×1, so it expects inputs of size 1×1 and outputs codewords of size 3×1. In fact, a keen eye might have noticed that this matrix performs the three repetition code we saw earlier:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} b \end{bmatrix} = \begin{bmatrix} 1 \cdot b \\ 1 \cdot b \\ 1 \cdot b \end{bmatrix} = \begin{bmatrix} b \\ b \\ b \end{bmatrix}$$

As said earlier, this is an example of a $[3, 1]$ code.

We observe that in the definition we used $\mathbb{Z}_2$ instead of $\mathbb{B}$ in order to highlight the fact that all the operations are performed modulo 2. Moreover, the set of possible codewords for a code corresponds to the vector space spanned by the columns of its generator matrix G, so for all the messages to be uniquely encoded we require that the columns of G are linearly independent. Lastly, we will identify a linear code with the codewords that it can generate, so a code C of type $[n, k]$ can be written as

$$C := \{Gx \mid x \in \mathbb{Z}_2^k\}$$

where G is the generator matrix of C.

Proposition 10.1

If C is a linear code, then $C^\perp$ is a linear code.

A great advantage of linear codes over more general error-correcting codes is their compact specification: a general code encoding k information bits into n bits requires 2^k codewords, each of length n, for a total of $n2^k$ bits needed to provide a description of the entire code. With a linear code instead, we only need to specify the nk bits of the generator matrix, and that's it. This is an exponential saving in the amount of memory required!

Now that we know what linear codes are, we can proceed to describe how to perform error-correction with them. In fact, in the repetition codes we just had to use a majority rule and that was enough to retrieve the origianl message, however in this case the error-correction procedure is not as intuitive. But first, we need to provide a different definition of linear codes.

Theorem 10.2: Linear code (alt. def.)

Given an $[n, k]$ generator code C it holds that

$$C = \ker H$$

where $H \in \mathbb{Z}_2^{(n-k)\times n}$ is the *parity check matrix* of C.

Proof. Let G be the generator matrix of C; pick $n-k$ linearly independent vectors $y_1, \ldots, y_{n-k}$ that are orthogonal to the columns of G — where the scalar product must be equal to 0 modulo 2. Then, construct the parity check matrix H of C as follows:

$$H := \begin{bmatrix} y_1^T \\ \vdots \\ y_{n-k}^T \end{bmatrix}$$

Claim 1: Given that $C = \{Gx \mid x \in \mathbb{Z}_2^k\}$, it holds that $C = \ker H$.

Proof of the Claim. First, we show that $C \subseteq \ker H$. Fix any $c \in C$, and consider the input $x \in \mathbb{Z}_2^k$ such that $c = Gx$. We observe that

$$Hc = H(Gx) = (HG)x$$

and by construction each row y_i^T is orthogonal to all columns of G, which implies that

$$\forall i, j \quad y_i^T g_j = 0 \implies Hc = (HG)x = 0 \cdot x = 0 \iff c \in \ker H$$

Now note that H has $n-k$ linearly independent rows by construction, which implies that $\mathrm{rk}(H) = n - k$. Therefore, by the rank-nullity theorem

$$\dim(\ker H) = n - \mathrm{rk}(H) = n - (n-k) = k$$

Then, since $C \subseteq \ker H$, and C has dimension k, this must imply that $C = \ker H$. □

Now we need to do the opposite in order to prove that the two definitions are actually equivalent. Let H be the parity check matrix of H; pick k linearly independent vectors $y_1, \ldots, y_k$ spanning the kernel of H. Then, construct the generator matrix G of C as follows:

$$G := \begin{bmatrix} y_1 & \ldots & y_k \end{bmatrix}$$

Claim 2: Given that $C = \ker H$, it holds that $C = \{Gx \mid x \in \mathbb{Z}_2^k\}$.

Proof of the Claim. Fix any $c \in \{Gx \mid x \in \mathbb{Z}_2^k\}$ generated by some $x \in \mathbb{Z}_2^k$. Then, it holds that $c = Gx$ meaning that

$$c = x_1 y_1 + \ldots + x_k y_k$$

Now, we compute Hc by linearity, obtaining

$$Hc = H(x_1 y_1 + \ldots + x_k y_k) = H(x_1 y_1) + \ldots + H(x_k y_k)$$

Observe that $x_1, \ldots, x_k$ are scalars, and each $y_i \in \ker H \implies Hy_i = 0$, therefore

$$Hc = x_1 H y_1 + \ldots x_k H y_k = 0$$

This proves that $Hc \in \ker H = C$, therefore $\{Gx \mid x \in \mathbb{Z}_2^k\} \subseteq G$.

For the second inclusion, observe that

$$\text{span}\{y_1, \ldots, y_k\} = \ker H = C$$

by construction of G. Therefore, for any fixed $c \in C = \ker H$ it must hold that c can be written as a linear combination of $y_1, \ldots, y_k$, i.e.

$$\exists x_1, \ldots, x_k \in \mathbb{Z}_2 \quad c = x_1 y_1 + \ldots + x_k y_k$$

which concludes that $c = Gx$ where x is the vector composed of the coefficients of the linear combination. □

This shows that the two definitions are equivalent. □

As an example of parity check matrix, consider the $[3, 1]$ repetition code described by the generator matrix we saw earlier

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

To construct H, we pick $3 - 1 = 2$ linearly independent vectors orthogonal to the columns of G, say

$$y_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \qquad y_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

and define the parity check matrix as

$$H := \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Ok, how do we use this matrix now? Suppose that we have a linear code C generated by a matrix G, and we encode a message x as $y = Gx$, but an error e due to noise corrupts y giving the corrupted codeword

$$y' = y + e$$

Because the parity check matrix H is such that $C = \ker H$, it holds that

$$Hy = 0 \implies Hy' = H(y + e) = Hy + He = He$$

We call $Hy' = He$ the **error syndrome** of y', and it is the key component that enables error-correction in linear codes.

To see how to perform error-correction, suppose that at most 1 error occurred. Then, the error syndrome Hy' is equal to 0 in the no error case, and it is equal to He_j when an error occurs on the j-th bit (where e_j is the j-th vector of the canonical basis). Then, under the assumption that errors occur on at most one bit, it is possible to perform error-correction by computing the error syndrome Hy' and comparing it to the different possible values of He_j, to dermine which (if any) bit need to be corrected. This looks like an interesting strategy, but we would like to be even more general.

Definition 10.3: Hamming distance

Given two words x and y each of n bits, we define the **Hamming distance** of x and y as follows:

$$d(x, y) := |\{i \mid x_i \neq y_i\}|$$

In simpler terms, the Hamming distance between two bit strings is the number of places at which they differ, for instance

$$d\left(\begin{bmatrix}1 & 1 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 1 & 0 & 1\end{bmatrix}\right) = 2$$

Definition 10.4: Hamming weight

Given a word x of n bits, we define the **Hamming weight** of x as follows:

$$w(x) := d(x, 0)$$

The weight of a word is the number of places at which x is non-zero. It is easy to show that the following proposition holds.

Proposition 10.2

Given two words x and y each of n bits, it holds that

$$d(x, y) = w(x + y)$$

where "+" is the addition modulo 2.

These two definitions are crucial because global properties of a linear code can be understood using the Hamming distance.

Definition 10.5: Distance of a code

The **distance** of a linear code C of type $[n, k]$ is defined as the minimum distance between its codewords

$$d(C) := \min_{\substack{x,y \in C \\ x \neq y}} d(x, y)$$

We usually refer to C as a linear code of type $[n, k, d]$, where $d = d(C)$.

Note that for any codewords $x, y \in C$ it holds that $x + y \in C$ by linearity of the code C, therefore by the previous proposition we obtain that

$$d(C) = \min_{\substack{x \in C \\ x \neq 0}} w(x)$$

which means that the minimum distance is actually also equal to the **minimum weight**.

The importance of the distance of a code is highlighted in the following result.

Theorem 10.3

A linear code C of type $[n, k, d]$ can correct $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors, and can detect $d - 1$ errors.

Proof. TODO □

da scrivere slide 6

TODO

how does ts work

10.2.1 The Hamming code

Let $r \geq 2$ be an integer, and consider the matrix H whose columns are all the $2^r - 1$ bit strings of length r different from the 0 bit string. This is a parity check matrix of dimensions $(2^r - 1) \times (2^r - r - 1)$ and it defines a linear code known as the **Hamming code**, named after Turing Prize winner Richard Hamming. A very famous special example of the Hamming code is obtained with $r = 3$, which defines the following parity check matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

which describes a linear code of type $[7, 4, 3]$. By Theorem 10.3, we know that this code is able to correct

$$t = \left\lfloor \frac{3-1}{2} \right\rfloor = 1$$

Let's give an example of error-correction through the $r = 3$ case of the Hamming code in action: suppose that the received codeword is

$$y' = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Now, we have to compute the syndrome of y', namely

$$Hy' = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Then, since this is not the 0 vector some error must have occurred, and because we know that $Hy' = He_j$ where e_j has 1 only in the j-th position we have that

$$He_j = \begin{bmatrix}1\\0\\1\end{bmatrix} \implies e_j = \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\end{bmatrix} \implies j = 5$$

which means that the error occured in the 5-th position of y'. This means that the original y was

$$y = \begin{bmatrix}1\\0\\1\\1\\0\\1\\0\end{bmatrix}$$

Now, to retrieve the original message x we need to construct a matrix that we did not discuss yet. First, we need the generator matrix G, and we can costruct it by following the construction in Theorem 10.3. For this example we will employ the most common generator matrix usually defined for the Hamming code $[7, 4, 3]$, which is the following

$$G := \begin{bmatrix}0&0&1&1\\0&1&0&1\\0&1&1&1\\1&0&0&1\\1&0&1&1\\1&1&0&1\\1&1&1&1\end{bmatrix}$$

This matrix is not invertible because it is not a square matrix, however we can simply choose k linearly indepentend rows from G and construct the following matrix

$$\hat{G} := \begin{bmatrix}0&0&1&1\\0&1&0&1\\0&1&1&1\\1&0&0&1\end{bmatrix}$$

which is now guaranteed to be invertible. Most importantly, we observe that we chose the 4 rows of the matrix G, and we will use this information later. Now, it can be proven

that the inverse (modulo 2) of $\hat{G}$ is the following matrix:

$$\hat{G}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Proving that $\hat{G}^{-1}\hat{G} = I \bmod 2$ is left as exercise. Finally, to obtain the message x that generated y we must first truncate y to the first 4 bits — because of how we constructed $\hat{G}$ — thus obtaining

$$\hat{y} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

which ultimately allows us to retrieve the original message

$$\hat{G}^{-1}\hat{y} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = x$$

10.2.2 The Steane code

So far we've only introduce linear codes in order to perform classical error-correction. Now we will see how linear codes are used to also correct errors in quantum communications. First, we need to provide the following definition.

Definition 10.6: Calderbank-Shor-Steane code

Given two linear codes C_1 and C_2 of types $[n, k_1]$ and $[n, k_2]$ such that

- $C_2 \subset C_1$
- both C_1 and $C_2^{\perp}$ correct t errors

we define the **Calderbank-Shor-Steane (CSS) code** of C_1 over C_2 of type $[n, k_1 - k_2]$ as

$$\mathrm{CSS}(C_1, C_2) := \mathrm{span}(\{|x + C_2\rangle \mid x \in C_1/C_2\})$$

where we define that

$$|x + C_2\rangle := \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle$$

We will not go into the details of this very dense definition, however what we really matter is a special case of if, usually referred to as **Steane code** — named after its inventor Andrew Steane. It is derived by using the $[7, 4, 3]$ Hamming code we saw earlier, and it is capable of correcting errors on t qubits. We can construct it as follows: let H be this exact Hamming space, and set

- $C_1 = H$
- $C_2 = H^\perp$

Then, since $H^\perp \subset H$, and $\left|H^\perp\right| = 2^{\dim\left(H^\perp\right)} = 2^3 = 8$ we have that

$$x = |0000000\rangle \implies |0_L\rangle := |x + H^\perp\rangle = \frac{1}{\sqrt{8}} \sum_{y \in H^\perp} |y\rangle$$

$$x = |1111111\rangle \implies |1_L\rangle := |x + H^\perp\rangle = \frac{1}{\sqrt{8}} \sum_{y \in H^\perp} |1111111 + y\rangle$$

where $|0_L\rangle$ and $|1_L\rangle$ indicate the "logical" 0 and 1, respectively. Then, if we need to tranfer a qubit

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

we just need to encode it as

$$|\psi_L\rangle = \alpha\,|0_L\rangle + \beta\,|1_L\rangle = \frac{\alpha}{\sqrt{8}} \sum_{y \in H^\perp} |y\rangle + \frac{\beta}{\sqrt{8}} \sum_{y \in H^\perp} |1111111 + y\rangle$$

exactly as we saw in the previous section with the repetition codes.

Additionally, from these equations we see that in the Steane code we are mapping the logical 0 to the superposition of all the codewords of $H^\perp$, while the logical 1 is mapped to the superposition of codewords that are in $H - H^\perp$ — this is because $1111111 \in H - H^\perp$, but we will not prove this fact. Moreover, it can be shown that for any received codeword $b = \begin{bmatrix} b_1 & \dots & b_7 \end{bmatrix}^T$ the syndrome Hb can be succintly computed as follows:

$$Hb = \begin{bmatrix} b_4 \oplus b_5 \oplus b_6 \oplus b_7 \\ b_2 \oplus b_3 \oplus b_6 \oplus b_7 \\ b_1 \oplus b_3 \oplus b_5 \oplus b_7 \end{bmatrix}$$

In particular, this formulation is useful because it can be translated into the following quantum circuit: TODO After computing the syndrome, we can proceed as before with the error-correction and message-retrieval procedure.

slide 13

The last detail that we need to point out is the importance of the Steane code. The Shor code we saw earlier was able to retrieve any qubit sent with an arbitrary single-qubit error, so would we want to adopt the Steane code? TODO

missing last 2 slides, i'm so bored

Bibliography

[BB14] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical Computer Science* 560 (Dec. 2014), 7–11. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.05.025. URL: http://dx.doi.org/10.1016/j.tcs.2014.05.025.

[Bel64] J. S. Bell. "On the Einstein Podolsky Rosen paradox". In: 1.3 (Nov. 1964), 195–200. ISSN: 0554-128X. DOI: 10.1103/physicsphysiquefizika.1.195. URL: http://dx.doi.org/10.1103/PhysicsPhysiqueFizika.1.195.

[CAB+20] M. Cerezo, Andrew Arrasmith, Ryan Babbush, et al. "Variational Quantum Algorithms". In: (2020). DOI: 10.48550/ARXIV.2012.09265. URL: https://arxiv.org/abs/2012.09265.

[Deu85] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[Deu92] Josza Deutsch. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (Dec. 1992), 553–558. ISSN: 2053-9177. DOI: 10.1098/rspa.1992.0167. URL: http://dx.doi.org/10.1098/rspa.1992.0167.

[Eke91] Artur K Ekert. "Quantum cryptography based on Bell's theorem". In: *Physical review letters* 67.6 (1991), p. 661.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Physical Review* 47.10 (May 1935), 777–780. ISSN: 0031-899X. DOI: 10.1103/physrev.47.777. URL: http://dx.doi.org/10.1103/PhysRev.47.777.

[Gro05] Lov K. Grover. "Fixed-Point Quantum Search". In: *Physical Review Letters* 95.15 (Oct. 2005). ISSN: 1079-7114. DOI: 10.1103/physrevlett.95.150501. URL: http://dx.doi.org/10.1103/PhysRevLett.95.150501.

[Gro97] Lov K. Grover. "Quantum Mechanics Helps in Searching for a Needle in a Haystack". In: *Physical Review Letters* 79.2 (July 1997), 325–328. ISSN: 1079-7114. DOI: 10.1103/physrevlett.79.325. URL: http://dx.doi.org/10.1103/physrevlett.79.325.

[Hal13] Brian C. Hall. *Quantum Theory for Mathematicians*. Springer New York, 2013. ISBN: 9781461471165. DOI: 10.1007/978-1-4614-7116-5. URL: http://dx.doi.org/10.1007/978-1-4614-7116-5.

[Kit95] A. Yu. Kitaev. *Quantum measurements and the Abelian Stabilizer Problem*. 1995. DOI: 10.48550/ARXIV.QUANT-PH/9511026. URL: https://arxiv.org/abs/quant-ph/9511026.

[PMS+14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, et al. "A variational eigenvalue solver on a photonic quantum processor". In: *Nature Communications* 5.1 (July 2014). ISSN: 2041-1723. DOI: 10.1038/ncomms5213. URL: http://dx.doi.org/10.1038/ncomms5213.

[Sho] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. SFCS-94. IEEE Comput. Soc. Press, 124–134. DOI: 10.1109/sfcs.1994.365700. URL: http://dx.doi.org/10.1109/SFCS.1994.365700.

[Tsi80] B. S. Tsirelson. "Quantum generalizations of Bell's inequality". In: *Letters in Mathematical Physics* 4.2 (Mar. 1980), 93–100. ISSN: 1573-0530. DOI: 10.1007/bf00417500. URL: http://dx.doi.org/10.1007/bf00417500.

[WZ82] W. K. Wootters and W. H. Zurek. "A single quantum cannot be cloned". In: *Nature* 299.5886 (Oct. 1982), 802–803. ISSN: 1476-4687. DOI: 10.1038/299802a0. URL: http://dx.doi.org/10.1038/299802a0.

[YLC14] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. "Fixed-Point Quantum Search with an Optimal Number of Queries". In: *Physical Review Letters* 113.21 (Nov. 2014). ISSN: 1079-7114. DOI: 10.1103/physrevlett.113.210501. URL: http://dx.doi.org/10.1103/PhysRevLett.113.210501.