A.y. 2023/24 — Leonardo Mazza — leonardo.mazza@universite-paris-saclay.fr

Lecture 3: Quantum algorithms.

The very first quantum algorithm was proposed by Deutsch in 1985 and so it will be the first that we study. This was followed by the Deutsch-Jozsa algorithm in 1992, the first algorithm that was exponentially faster than any deterministic classical algorithm. We will learn however that probabilistic classical algorithms can be much faster than deterministic ones, provided that one can except a small chance of failure. Motivated by this, Simon invented a problem for which even probabilistic classical algorithms take exponentially long time and he found a fast quantum algorithm with bounded failure probability that runs in polynomial time.

These 'toy' problems and quantum algorithms were expressly invented to be difficult for classical computers and easy for quantum computers in order to demonstrate the possibilities of quantum hardware. They are not practically useful algorithms but they paved the way for all subsequent ones which have been invented, including most famously, the Grover search algorithm for unstructured databases and Shor's algorithm for finding the prime factors of large numbers–a task that is required to break RSA public key encryption. We will study a key component of Shor's algorithm, the quantum Fourier transform, since it has wide application.

1) Deutsch algorithm.

We are now ready to study our first algorithm. As mentioned in the introduction to this chapter, in 1985 David Deutsch developed the very first quantum algorithm. It only solves a 'toy' problem, but it opened the door to a new world. Let us now enter that new world.

Let us begin our discussion on the Deutsch algorithm by describing the problem. We have already seen that classically there are only four possible 1-bit functions:

IDENTITY
$0 \to 0$
$1 \to 1$

NOT
$0 \to 1$
$1 \to 0$

ERASE
$0 \to 0$
$1 \to 0$

ERASE–NOT
$0 \to 1$
$1 \to 1$

Alice choses one of these four functions and creates a QUANTUM ORACLE. A quantum oracle should be seen as a black box, an object that we want to understand and to decode. It is given to us, we do not have to create it. It acts as a linear multi-qubit quantum gate.
In the case of the Deutsch algorithm, once the function f is chosen among the four listed above, the oracle works as follows: It acts on two qubits and returns

$$\hat{O}_f |d\rangle \otimes |0\rangle = |d \oplus f(0)\rangle \otimes |0\rangle$$
$$\hat{O}_f |d\rangle \otimes |1\rangle = |d \oplus f(1)\rangle \otimes |1\rangle$$

with $d = 0, 1$.

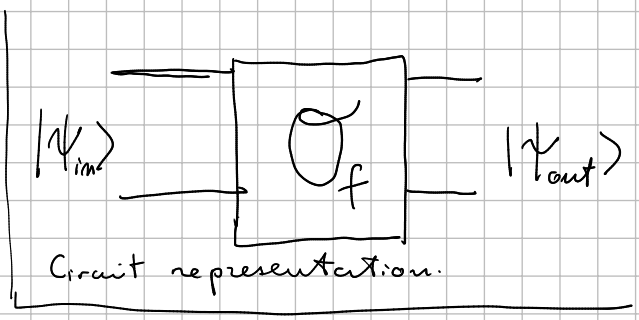
input for f

output

$\oplus$ is the sum mod 2 $\begin{cases} 0+0 = 0 \\ 0+1 = 1+0 = 1 \\ 1+1 = 0 \end{cases}$

For instance, when $d = 0$ we obtain

$$\hat{O}_f |0\rangle|0\rangle = |f(0)\rangle |0\rangle$$
$$\hat{O}_f |0\rangle|1\rangle = |f(1)\rangle |1\rangle$$



Circuit representation.

There are four functions f and we clearly see that they can be separated into two groups:

A) BALANCED functions: identity and not

[balanced because in the output there are as many 1 as 0s]

||| B) CONSTANT functions: erase and erase-not.

The Deutsch algorithm shows that there is an efficient way of using a quantum computer to learn whether the function

of the oracle is balanced or constant.

Let us see how this could work classically: I necessarily need to interrogate the oracle twice. First I ask for $f(0)$ and then for $f(1)$. There is no other way to learn whether the function is balanced or not. 2 interrogations are necessary.

We now show that using a quantum computer it is possible to learn whether f is balanced or constant with only one interrogation.

Key observation:

For a balanced function: $f(0) \oplus f(1) = 1$

For a constant function: $f(0) \oplus f(1) = 0$

We have seen that:

$$\hat{\mathcal{O}}_f |d\rangle \otimes |d'\rangle = |d \oplus f(d)\rangle \otimes |d'\rangle \qquad d, d' = \{0,1\}$$

Let us use the fact that $\hat{\mathcal{O}}_f$ is a quantum gate (unitary and linear operator) to discuss its action on other input states.

We define: $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

$$\hat{\mathcal{O}}_f |d\rangle \otimes |+\rangle = \hat{\mathcal{O}}_f |d\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \hat{\mathcal{O}}_f |d\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}} \hat{\mathcal{O}}_f |d\rangle \otimes |1\rangle$$

$$= \frac{|d + f(0)\rangle \otimes |0\rangle + |d + f(1)\rangle \otimes |1\rangle}{\sqrt{2}}$$

Let us now consider another input state:

$$|\psi_{in}\rangle = |-\rangle \otimes |+\rangle = \frac{1}{2}|0\rangle|0\rangle + \frac{1}{2}|0\rangle|1\rangle - \frac{1}{2}|1\rangle|0\rangle - \frac{1}{2}|1\rangle|1\rangle$$

$$\hat{\mathcal{O}}_f |\psi_{in}\rangle = \frac{1}{2}|0 \oplus f(0)\rangle|0\rangle + \frac{1}{2}|0 \oplus f(1)\rangle|1\rangle - \frac{1}{2}|1 \oplus f(0)\rangle|0\rangle - \frac{1}{2}|1 \oplus f(1)\rangle|1\rangle$$

To make further progress we now consider explicitly the two cases.

① f is constant: $f(0) = f(1)$ and $f(0) \oplus f(1) = 0$

In this case: $\hat{O}_f |\psi_{in}\rangle = \frac{1}{2}\{(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) \otimes |0\rangle +$

$+(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) \otimes |1\rangle\}$

$= \underbrace{\frac{|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle}{\sqrt{2}}}_{\pm|-\rangle \text{ when } f(0)=0 \text{ or } f(0)=1} \otimes \underbrace{\frac{|0\rangle + |1\rangle}{\sqrt{2}}}_{|+\rangle}$

$+|-\rangle$ when $f(0) = 0$

$-|-\rangle$ when $f(0) = 1$

$\hat{O}_f |-\rangle \otimes |+\rangle = \pm |-\rangle \otimes |+\rangle$

② f is balanced: $f(0) \oplus f(1) = 1$ and

$0 \oplus f(0) = 1 \oplus f(1)$ and $1 \oplus f(0) = 0 \oplus f(1)$

In this case $\hat{O}_f |\psi_{in}\rangle = \frac{1}{2}\{(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)|0\rangle +$

$-(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)|1\rangle\}$

$= \underbrace{\frac{|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle}{\sqrt{2}}}_{\pm|-\rangle \text{ when } f(0)=0 \text{ or } f(0)=1} \otimes \underbrace{\frac{|0\rangle - |1\rangle}{\sqrt{2}}}_{|-\rangle}$

$\begin{cases} |-\rangle \text{ when } f(0) = 0 \\ -|-\rangle \text{ when } f(0) = 1 \end{cases}$

$\hat{O}_f |\psi_{in}\rangle = \pm |-\rangle \otimes |-\rangle$

In summary:

$$\hat{O}_f |\psi_{in}\rangle = \begin{cases} \pm |-\rangle \otimes |-\rangle & \text{for a balanced function} \\ \pm |-\rangle \otimes |+\rangle & \text{For a constant function.} \end{cases}$$

At this stage it is sufficient to perform a Hadamard on the second qubit:

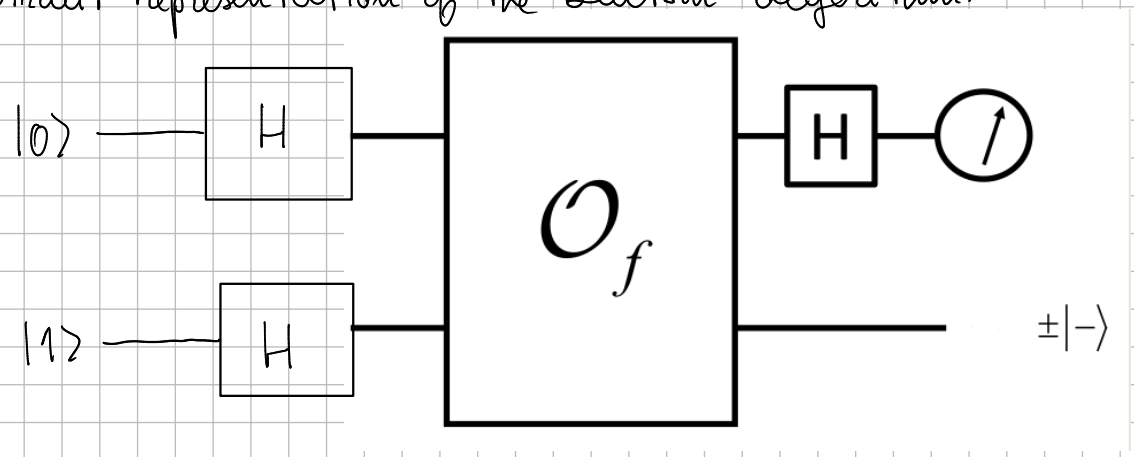
$$(\mathbb{1} \otimes H)\left(|{\pm}\rangle \otimes |-\rangle\right) = \pm |-\rangle|1\rangle$$

$$(\mathbb{1} \otimes H)\left(|{\pm}\rangle \otimes |+\rangle\right) = \pm |-\rangle|0\rangle$$

↑ Perform a measurement of the qubit and you have the answer!

- 1 → balanced
- 0 → constant.

Circuit representation of the Deutsch algorithm:



2) On the notion of quantum oracle.

> *Oracle, a person (such as a priestess of ancient Greece) through whom a deity is believed to speak.*
> *From the Latin oraculum, from orare, 'to speak'*

A quantum oracle is a 'blackbox' whose inner workings *may* be inaccessible to the user. Oracles are often used as inputs to quantum algorithms.

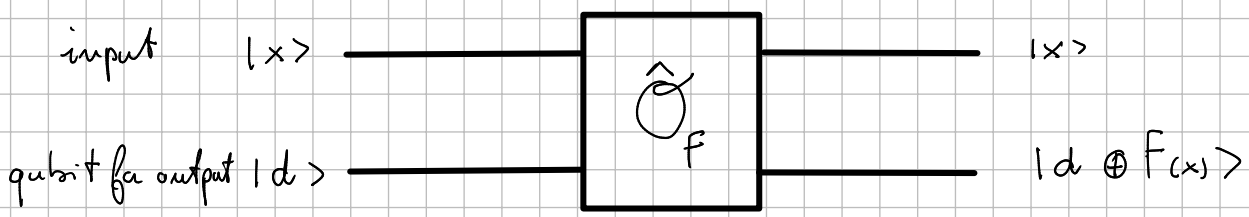
An oracle is a theoretical / conceptual way to bring classical data in a quantum computation. It acts as a quantum gate, that is, as a unitary linear operator.

Deutsch: Take a classical one-bit function $f(x) = y$

$$\hat{O}_f |d\rangle |x\rangle = |d \oplus f(x)\rangle |x\rangle$$

$|x\rangle$: qubit with the input

$|d\rangle$: qubit with the output



The situation can be generalised to a multi-qubit setting. Consider the n-bit function

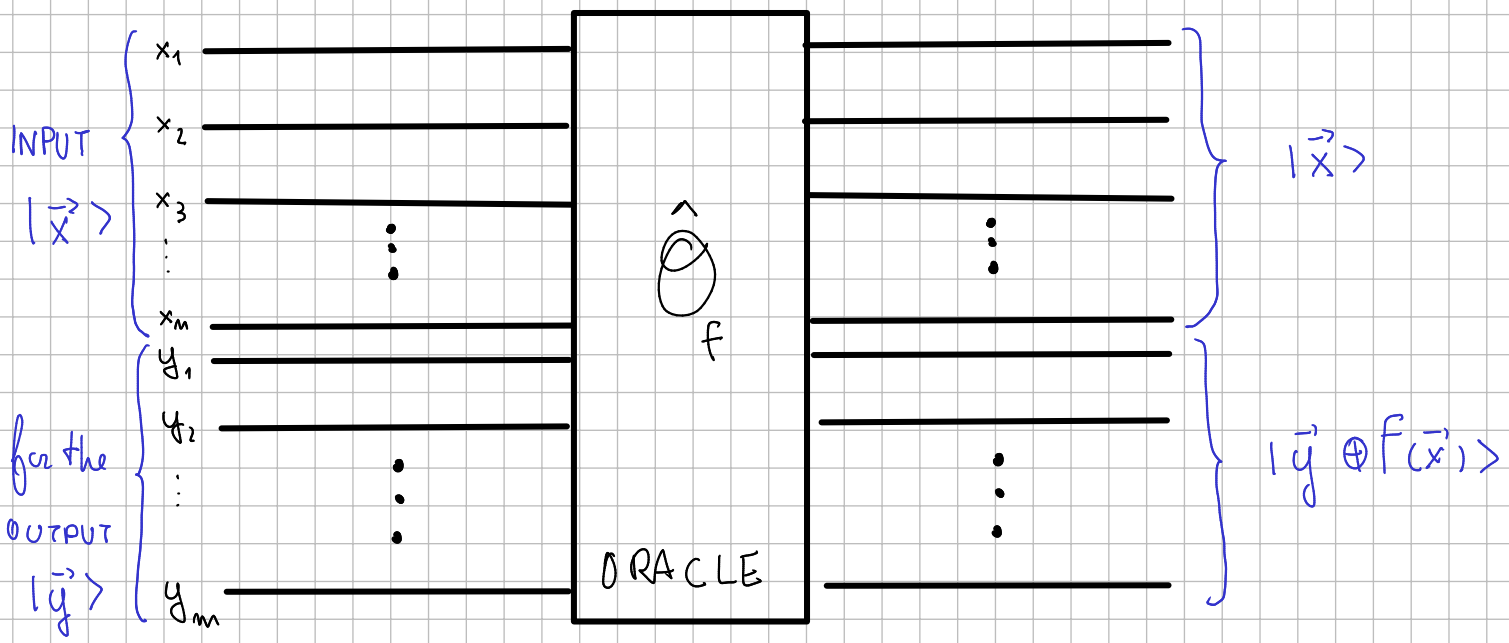
$$f(\vec{x}) = \vec{y}$$

$\vec{x}$: vector with n bits

$\vec{y}$: vector with m bits

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \qquad \vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}$$

We can construct an oracle as follows:



Why such a complicated construction? The problem is that a quantum gate is reversible, whereas the F functions can be irreversible. Let us go back to single qubit functions. We know that identity and not are reversible. They can also be implemented as follows

$$\hat{O}'_F |x\rangle = |F(x)\rangle$$

For the identity: $\hat{O}'_{identity} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ OK! These are unitaries!

For the not: $\hat{O}'_{not} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

However, the erase and erase-not are not unitaries.

For the erase: $\hat{O}'_{erase} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ NOT OK!

For the erase-not: $\hat{O}'_{erase\text{-}not} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$ These are not unitaries!

Let us inspect the form of the oracle $\hat{O}_{erase}$ in the version of the Deutsch algorithm.

$$\hat{O}_{erase}\,|y\rangle|x\rangle = |y \oplus \underbrace{f(x)}\rangle\,|x\rangle = |y\rangle|x\rangle$$

$f(x) = 0 \;\; \forall x$

It is the two-qubit identity! Great result!

Before we had the problem that $|f(0)\rangle = |f(1)\rangle$. The initial states $|0\rangle$ and $|1\rangle$ were orthogonal but now the two states $|f(0)\rangle$ and $|f(1)\rangle$ are parallel.

Things changed now: $|y\rangle|0\rangle$ and $|y\rangle|1\rangle$ are orthogonal and $|y \oplus f(0)\rangle|0\rangle$ and $|y \oplus f(1)\rangle|1\rangle$ remain orthogonal!

Thus, in general terms the oracle is a unitary linear operator that describes a reversible operation on the quantum system, even if it carries information about a f function that is irreversible.

3) History of quantum algorithms.

1985 – Deutsch algorithm.

- toy problem
- classically: 2 interrogation of oracle; quantum: 1 interrogation.
- first example of a quantum advantage, there is one task that a quantum computer can do better than a classical computer.

1992 – Deutsch – Josza algorithm.

- toy problem: discover whether a n-bit function is constant or balanced
- exponential separation (in the number of qubits) between the calls to the oracle of a classical computer and of a quantum computer. Quantum: one interrogation of the oracle. Classical: $2^{n-1}+1$ interrogations.
- first example of a qualitative quantum advantage. Different scaling!
- It is possible to do better with a classical stochastic algorithm.

1997 – Bernstein – Vazirani algorithm.

- Identify a n-bit function such that $f(\vec{x}) = \vec{w} \cdot \vec{x}$. What is $\vec{w}$? It is a variation of the Deutsch-Josza algorithm for a more interesting problem.
- Classical: n-interrogations are necessary.
- Quantum: 1 interrogation.

1996 – Grover's algorithm

- Performs a search in an unstructured database. It is considered an interesting and relevant problem.

- Best classical deterministic algorithm: N interrogations, where N is the number of entries of the database. In average, $N/2$ interrogations. Same linear scaling with N.
- Quantum algorithm: it is probabilistic and it find the result with a high accuracy with $\sqrt{N}$ interrogations of the oracle.

1994 – Shor's algorithm: quantum Fourier transform and the factorisation of prime numbers.

4) Grover's search algorithm.

The problem. We have a database with $N = 2^n$ entries.

$\vec{x}$ is a n-bit vector. ← one value $\vec{x}$ for each entry.

We want to search into this database and find $\vec{x}_0$, the goal of our search.

EXAMPLE: a list of names and phone numbers and we are looking for a specific one.

The search process is represented by a function:

$$F(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} = \vec{x}_0 \\ 0 & \text{otherwise.} \end{cases}$$

The function F is now encoded in an oracle, which is a reversible quantum gate:

$$\hat{O}_F \, |d\rangle \otimes |\vec{x}\rangle = |d \oplus F(\vec{x})\rangle \otimes |\vec{x}\rangle$$

An aside: the person that performs the search can create the oracle $\hat{O}_F$ without knowing the position of $\vec{x}_0$.

$F(\vec{x})$ is a function that goes to the position $\vec{x}$ and checks what is in the database at that position.

EXAMPLE of a database with 8 entries

$\vec{x}$	ENTRY
000	Charles
001	Guillaume
010	Emma
011	Alice
100	Harry
101	Bob
110	Dean
111	Fanny

Find $\vec{x}$ corresponding to "Harry".

THE ALGORITHM.

Prepare the input state (1 qubit ↦ |−⟩, n qubit state ↦ $|\vec{x}\rangle$)

$$|\psi_{in}\rangle = |-\rangle \otimes |\vec{x}\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |\vec{x}\rangle$$

so that:

$$\hat{\mathcal{O}}_F |\psi_{in}\rangle = \frac{|0 \oplus F(\vec{x})\rangle - |1 \oplus F(\vec{x})\rangle}{\sqrt{2}} \otimes |\vec{x}\rangle .$$

If $F(x) = 0$ then $\hat{\mathcal{O}}_F |-\rangle \otimes |\vec{x}\rangle = |-\rangle \otimes |\vec{x}\rangle$

If $F(x) = 1$ then $\hat{\mathcal{O}}_F |-\rangle \otimes |\vec{x}\rangle = -|-\rangle \otimes |\vec{x}\rangle$

With compact notation: $\hat{\mathcal{O}}_F |-\rangle \otimes |\vec{x}\rangle = (-1)^{F(\vec{x})} |-\rangle \otimes |\vec{x}\rangle$.

Since $(-1)^{F(\vec{x})}$ is a scalar, we can put it together to $|-\rangle$ but also to $|\vec{x}\rangle$. We choose the latter:

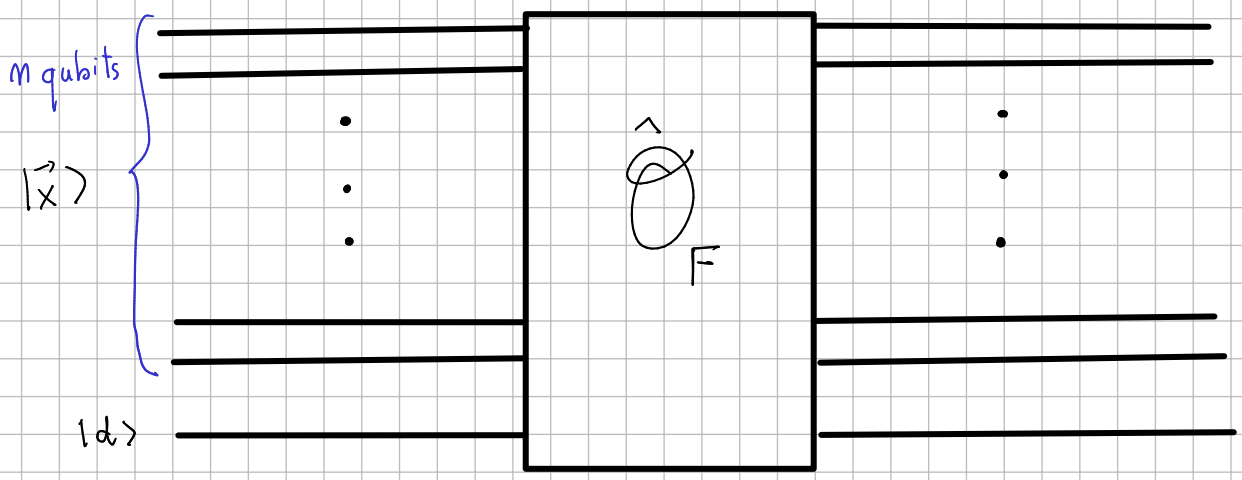
$$\hat{\mathcal{O}}_F |-\rangle \otimes |\vec{x}\rangle = |-\rangle \otimes \left((-1)^{F(\vec{x})} |\vec{x}\rangle \right).$$

The $|-\rangle$ state is considered to be a quantum catalyst, that lets the phase $(-1)^{F(\vec{x})}$ appear in front of the state $|\vec{x}\rangle$, while remaining unchanged.

For this reason, we can give a simple expression to $\hat{\mathcal{O}}_F$:

$$\hat{\mathcal{O}}_F = \hat{\mathbb{1}}_2 \otimes \left(\hat{\mathbb{1}}_{2^n} - 2\, |\vec{x}_0\rangle\langle\vec{x}_0| \right)$$

Circuit representation. n-bits for $2^n = N$ entries of the database.



Consider now this more complex input state: $\frac{1}{\sqrt{2^n}}\sum_{\vec{x}}|\vec{x}\rangle$

$$|\Psi_{in}\rangle = |-\rangle \otimes |+\rangle^{\otimes n} \quad \text{where:} \quad |+\rangle^{\otimes n} = \underbrace{|+\rangle \otimes |+\rangle \otimes \dots \otimes |+\rangle}_{n \text{ times for } n \text{ qubits.}}$$

What happens when the oracle acts on this novel input state?

$$\hat{O}_F |-\rangle \otimes |+\rangle^{\otimes n} = |-\rangle \otimes \left(\hat{\mathbb{1}}_{2^n} - 2|\vec{x}_0\rangle\langle\vec{x}_0|\right)|+\rangle^{\otimes n}$$

$$= |-\rangle \otimes |+\rangle^{\otimes n} - 2|-\rangle \otimes \left(\underbrace{\langle\vec{x}_0|+\rangle^{\otimes n}}_{\left(\frac{1}{\sqrt{2}}\right)^n = \frac{1}{2^{n/2}}}\right)|\vec{x}_0\rangle$$

$$= |-\rangle \otimes |+\rangle^{\otimes n} - \frac{2}{2^{n/2}}|-\rangle \otimes |\vec{x}_0\rangle$$

We now introduce the notation: $|\Phi\rangle \doteq |+\rangle^{\otimes n}$

Consider now the operator $\hat{R} = \underbrace{+2|\Phi\rangle\langle\Phi| - \hat{\mathbb{1}}_{2^n}}_{\text{Grover's diffusion operator.}}$

and the composite operation $\hat{G} \equiv \hat{R} \circ \left(\hat{\mathbb{1}}_{2^n} - 2|\vec{x}_0\rangle\langle\vec{x}_0|\right)$

We want to characterize the action of $\hat{G}$.

Before, note that it is a quantum gate that can be implemented experimentally. The term on the left is the Oracle. The second is a second oracle built by us that identifies the state $|+\rangle$. It can be implemented using Hadamard gates.

Idea: the initial state $|\Phi\rangle$ is almost orthogonal to the target state $|\vec{x}_0\rangle$. Indeed:

$$\langle \vec{x}_0 | \Phi \rangle = \frac{1}{\sqrt{2^n}} \ll 1$$

$|\vec{x}_0\rangle$ and $|\Phi\rangle$ thus do not form a basis but are linearly independent. We can easily create an orthonormal basis for this space:

$$\mathcal{B} = \left\{ |\vec{x}_0\rangle, \; |\chi\rangle \doteq \frac{1}{\sqrt{1-\frac{1}{2^n}}} \left(|\Phi\rangle - \frac{1}{\sqrt{2^n}} |\vec{x}_0\rangle \right) \right\}$$

The initial state is:

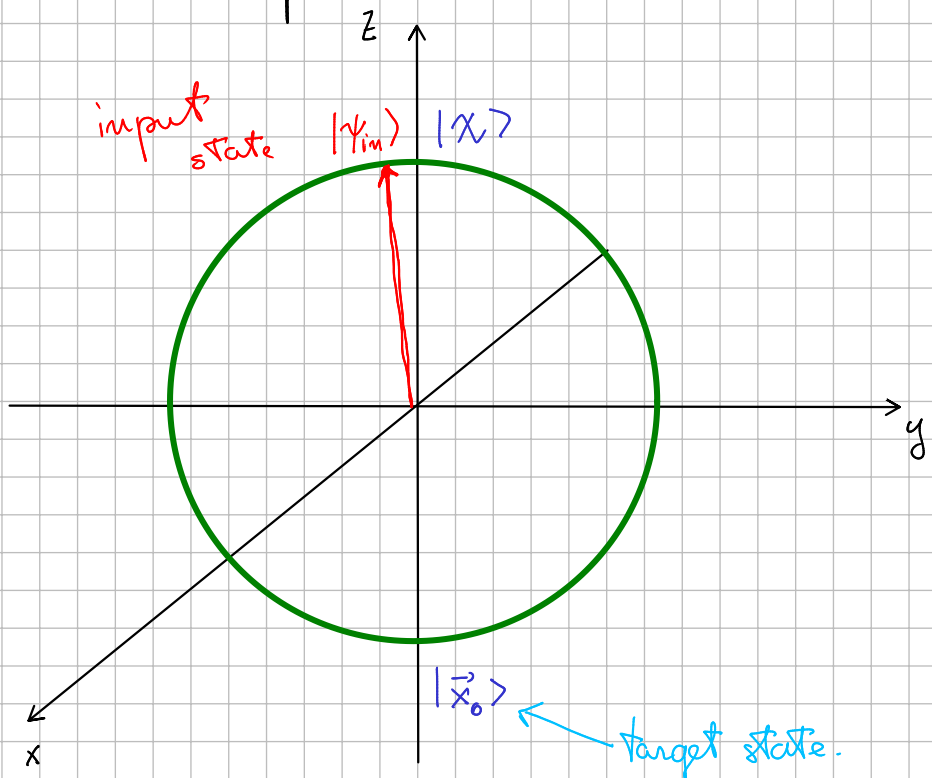
$$|\Phi\rangle = \sqrt{1-\frac{1}{2^n}}\, |\chi\rangle + \frac{1}{\sqrt{2^n}} |\vec{x}_0\rangle \sim |\chi\rangle.$$

The goal is to create a process that changes the state from $|\Phi\rangle \sim |\chi\rangle$ to $|\vec{x}_0\rangle$.

Let me represent a generic state: $|\psi\rangle = \cos\theta\, |\chi\rangle + e^{i\varphi} \sin\theta\, |\vec{x}_0\rangle$ on a Bloch sphere:

$\theta \in [0, \frac{\pi}{2}]$

$\varphi \in [0, 2\pi[$



How can we go from $|\chi\rangle$ to $|\vec{x}_0\rangle$?

Let us locate $|\psi_{in}\rangle = |\Phi\rangle$ on this Bloch sphere

$$\cos\theta_{in} = \sqrt{1 - \frac{1}{2^n}} \sim 1 \qquad \sin\theta_{in} = \sqrt{\frac{1}{2^n}} \sim 0 \qquad \varphi = 0$$

New notation $g \doteq \sqrt{\frac{1}{2^n}}$ $\qquad W = 1 - g^2$

Hence: $\cos\theta_{in} = \sqrt{W}$ $\qquad \sin\theta_{in} = g$.

$$\boxed{|\Phi\rangle = \sqrt{W}\,|\chi\rangle + g\,|\vec{x}_0\rangle}$$

We now determine the action of G.

Explicit calculation in the space spanned by $\{|\chi\rangle, |\vec{x}_0\rangle\}$

$$\left(\mathbb{1}_{2^n} - 2\,|\vec{x}_0\rangle\langle\vec{x}_0|\right)|\chi\rangle = |\chi\rangle - 2\underbrace{\langle\vec{x}_0|\chi\rangle}_{=0}\,|\vec{x}_0\rangle = |\chi\rangle$$

$$\left(\mathbb{1}_{2^n} - 2\,|\vec{x}_0\rangle\langle\vec{x}_0|\right)|\vec{x}_0\rangle = |\vec{x}_0\rangle - 2\,|\vec{x}_0\rangle = -|\vec{x}_0\rangle$$

} The oracle leaves the space invariant

The subspace of our interest is invariant under the action of the oracle.

$$\left(-\mathbb{1}_{2^n} + 2\,|\Phi\rangle\langle\Phi|\right)|\chi\rangle = -|\chi\rangle + 2\underbrace{\langle\Phi|\chi\rangle}_{\sqrt{W}}\,|\Phi\rangle =$$

$$= -|\chi\rangle + 2W\,|\chi\rangle + 2\sqrt{W}\,g\,|\vec{x}_0\rangle$$

$$= (2W-1)\,|\chi\rangle + 2\sqrt{W}\,g\,|\vec{x}_0\rangle$$

$$\left(-\mathbb{1}_{2^n} + 2\,|\Phi\rangle\langle\Phi|\right)|\vec{x}_0\rangle = -|\vec{x}_0\rangle + 2\underbrace{\langle\Phi|\vec{x}_0\rangle}_{g}\left(\sqrt{W}\,|\chi\rangle + g\,|\vec{x}_0\rangle\right)$$

$$= \underbrace{(2g^2-1)}_{2-2W-1 \,=\, 1-2W} |\vec{x}_0\rangle + 2g\sqrt{W}\,|\chi\rangle$$

In summary:

$$\left(-\mathbb{1}_{2^n} + 2|\Sigma\rangle\langle\Sigma|\right)\left(\mathbb{1}_{2^n} - 2|\vec{x}_0\rangle\langle\vec{x}_0|\right)|\chi\rangle = (2W-1)|\chi\rangle + 2\sqrt{W}g\,|\vec{x}_0\rangle$$

$$\underbrace{\left(-\mathbb{1}_{2^n} + 2|\Sigma\rangle\langle\Sigma|\right)\left(\mathbb{1}_{2^n} - 2|\vec{x}_0\rangle\langle\vec{x}_0|\right)}|\vec{x}_0\rangle = -2\sqrt{W}g\,|\chi\rangle + (2W-1)\,|\vec{x}_0\rangle$$

These 2 operation leave the space spanned by $\{|\chi\rangle, |\vec{x}_0\rangle\}$ invariant.

What are the numbers that appear here?

$$\sin 2\theta_{in} = 2\cos\theta_{in}\sin\theta_{in} = 2g\sqrt{W}$$

$$\cos 2\theta_{in} = 1 - 2\sin^2\theta_{in} = 1 - 2g^2 = 2W-1$$

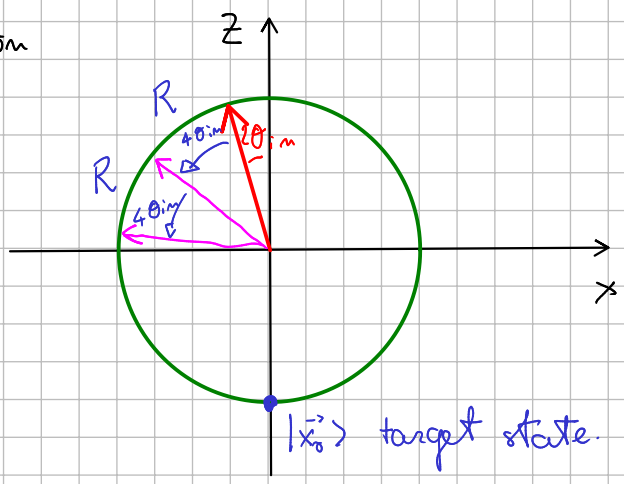
Hence, the action of R is a rotation in the space spanned by $\{|\chi\rangle, |\vec{x}_0\rangle\}$

$$R = \begin{pmatrix} \cos 2\theta_{in} & \sin 2\theta_{in} \\ -\sin 2\theta_{in} & \cos 2\theta_{in} \end{pmatrix}$$

The idea by Grover is to repeatedly apply R to turn $|\Sigma\rangle$ into a state that is very close to $|\vec{x}_0\rangle$.

No complex phase appears, hence we stay in the $y=0$ plane. It is a rotation around the y axis.

Geometric visualisation



Hence, after M applications of R, $\theta_{in} \to \theta_M = \theta_{in} + 2M\theta_{in}$

What is the value of M that makes θ_M close to $\frac{\pi}{2}$?

M is an integer close to $\frac{\pi}{4\theta_{in}} - \frac{1}{2}$

$$\theta_{in} = \arctan \tan \theta_{in} = \arctan \frac{\sin\theta_{in}}{\cos\theta_{in}} = \arctan \frac{1}{2^{n/2}}$$

$$M \sim \frac{1}{2}\left(\frac{\pi}{2\arctan 2^{-n/2}} - 1\right) \approx \frac{1}{2}\left(\frac{\pi}{2}\, 2^{n/2} - 1\right) \approx \frac{\pi}{4}\sqrt{N} \quad !$$

This is the number of times we apply R.
It is thus the number of times that we apply the oracle!

$\sim$ It scales as $\sqrt{N}$, where N is the number of entries.

The classical algorithm would scale as $\sim N$!

Polynomial advantage of the quantum algorithm.

Important remark: the quantum algorithm is probabilistic: there is a non-zero probability that it gives the wrong answer!