



HAL
open science

Quantum algorithms

Kenneth Maussang

► **To cite this version:**

Kenneth Maussang. Quantum algorithms. Master. Introduction to Quantum Computing, France. 2023, pp.107. hal-04423789

HAL Id: hal-04423789

<https://cel.hal.science/hal-04423789>

Submitted on 29 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Quantum algorithms

Introduction to Quantum Computing

Kenneth MAUSSANG

Université de Montpellier

2022 – 2023

Quantum algorithms

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation
- 8 Quantum error correction

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation

I. Introduction

We might divide quantum algorithms in two categories

Polynomial speed up	Exponential speed up
Grover's search Quantum walks Graph algorithms Minimum finding	Integer Matrix Inversion Phase Estimation Quantum Fourier Transform

Quantum algorithms

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation

II. Bernstein-Vazirani algorithm

Bernstein-Vazirani algorithm is a restricted version of Deutsch-Josa algorithm.

Instead of distinguishing between two different classes of functions, it tries to learn a string encoded in a function. One is given an oracle implementing a function f

$$f : \{0, 1\}^n \longrightarrow \{0, 1\}.$$

It is given that $f(x)$ is a dot product between x and a secret string $s \in \{0, 1\}^n$ modulo 2

$$f(x) = x \cdot s = x_1 \cdot s_1 + x_2 \cdot s_2 + \cdots + x_n \cdot s_n.$$

Bernstein-Vazirani algorithm aims at finding s .

II. Bernstein-Vazirani algorithm

Classically, it requires to evaluate n times the function $f(x)$, with $x = 2^k$, $k \in \{0, 1, \dots, n-1\}$.

$$f(1000 \cdots 00) = s_1,$$

$$f(0100 \cdots 00) = s_2,$$

\vdots

$$f(0000 \cdots 01) = s_n,$$

Thanks to Bernstein-Vazirani algorithm, only one query is needed with a quantum computer.

II. Bernstein-Vazirani algorithm

Algorithm

- 1 Initialize a quantum register of n qubit in the state $|0\rangle^{\otimes n}$

$$|\psi_0\rangle = |0\rangle^{\otimes n}.$$

- 2 Apply a Hadamard gate to each qubit of the quantum register, providing the state

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n} |x\rangle.$$

- 3 Apply the Oracle to the previous superposed state to obtain the following state

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n} (-1)^{f(x)} |x\rangle.$$

II. Bernstein-Vazirani algorithm

Algorithm

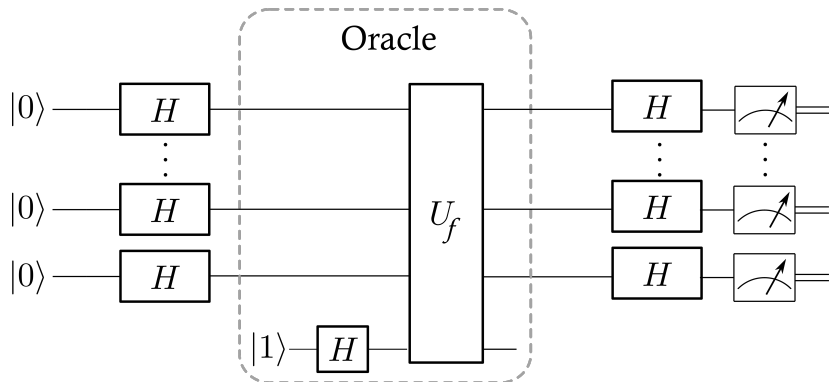
- 1 Apply Hadamard gate on each qubit of the quantum register.
If $s_i = 1$, it converts the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ to $|1\rangle$. If $s_i = 0$, it converts the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ to $|0\rangle$.

$$|\psi_3\rangle = |s_1, s_2, \dots, s_n\rangle.$$

- 2 To obtain s , the classical measurement on the $\{|0\rangle, |1\rangle\}$ basis provides the result.

II. Bernstein-Vazirani algorithm

Circuit diagram of Bernstein-Vazirani algorithm



- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm**
 - Grover's problem - unstructured search
 - Grover's algorithm
 - Analysis of Grover's algorithm
 - Geometrical interpretation of Grover's algorithm
 - Number of iterations
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm**
 - Grover's problem - unstructured search
 - Grover's algorithm
 - Analysis of Grover's algorithm
 - Geometrical interpretation of Grover's algorithm
 - Number of iterations
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation

III.1. Grover's problem - unstructured search

A simple example of a problem that fits into the query complexity model is **unstructured search on a set of N elements, in which only one element is marked**. In this problem, we are given a function

$$f : \{x_i, i \in \llbracket 0, N - 1 \rrbracket\} \longrightarrow \{0, 1\},$$

with the promise that it exists only one $p \in \llbracket 0, N - 1 \rrbracket$ such that

$$f(x_p) = 1, \text{ and for } q \neq p, f(x_q) = 0.$$

Then, x_p is the "marked" element.

Our task is to output x_p , f being given by an Oracle.

III.1. Grover's problem - unstructured search

It is intuitively clear that the unstructured search problem requires about N queries to be solved classically.

Let \mathcal{A} be a classical algorithm which solves the unstructured search problem on a set of N elements with a failure probability $\lesssim 1/2$. Then, \mathcal{A} makes $\mathcal{O}(N)$ queries in the worst case.

Grover (1997): *there is a quantum algorithm which solves the unstructured search problem using $\mathcal{O}(\sqrt{N})$ queries.*

For simplicity, we assume that $N = 2^n$, $n \in \mathbb{N}$ (this is not an essential restriction). Thus, we associate any element of $\{x_i\}$ with an n -bits string.

1 Introduction

2 Bernstein-Vazirani algorithm

3 Grover's algorithm

- Grover's problem - unstructured search

- **Grover's algorithm**

- Analysis of Grover's algorithm
- Geometrical interpretation of Grover's algorithm
- Number of iterations

4 Grover's algorithm in the case of multiple marked elements

5 Phase estimation

III.2. Grover's algorithm

We are given access to

$$f : \{0, 1\}^n \longrightarrow \{0, 1\},$$

with the property that $f(x_p) = 1$ for a unique element x_p . We use a quantum circuit on n qubits with an initial state

$$|\psi_0\rangle = |0\rangle^{\otimes n}.$$

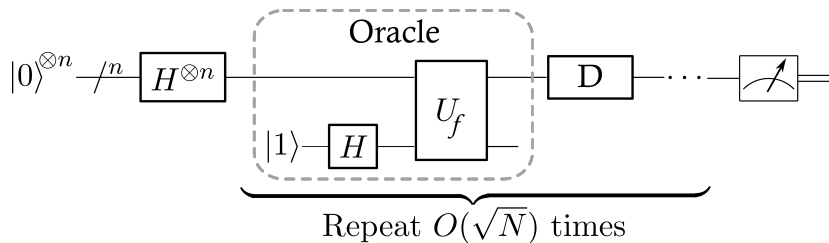
Let \hat{H} denote the Hadamard gate, and let \hat{U}_0 denote the n -qubit operation which inverts the phase of only $|0\rangle^{\otimes n}$

$$\begin{cases} \hat{U}_0 |0\rangle^{\otimes n} = -|0\rangle^{\otimes n} \\ \hat{U}_0 |x\rangle = |x\rangle \text{ for } |x\rangle \neq |0\rangle^{\otimes n} \end{cases} \quad (1)$$

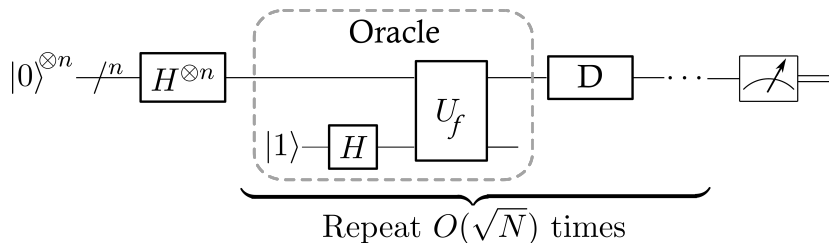
III.2. Grover's algorithm

Grover's algorithm:

- 1 Apply $\hat{H}^{\otimes n}$;
- 2 Repeat the following operation T times, for some T to be determined later
 - 1 Apply \hat{U}_f ;
 - 2 Apply $\hat{D} = -\hat{H}^{\otimes n} \hat{U}_0 \hat{H}^{\otimes n}$.
- 3 Measure all the qubits and output the results.



III.2. Grover's algorithm



The overall operation performed, applied on the initial state $|0\rangle^{\otimes n}$, is unitary

$$\hat{D}^T \hat{H}^{\otimes n} = \left(-\hat{H}^{\otimes n} \hat{U}_0 \hat{H}^{\otimes n} \right)^T \hat{H}^{\otimes n}.$$

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm**
 - Grover's problem - unstructured search
 - Grover's algorithm
 - Analysis of Grover's algorithm**
 - Geometrical interpretation of Grover's algorithm
 - Number of iterations
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation

III.3. Analysis of Grover's algorithm

To describe Grover's algorithm, we introduce unitary operators

$$\hat{I}_{|\psi\rangle} = \mathbb{I} - 2|\psi\rangle\langle\psi| \quad \text{and} \quad \hat{R}_{|\psi\rangle} = -\hat{I}_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - \mathbb{I},$$

where \mathbb{I} is the identity operator, and $|\psi\rangle$ is an arbitrary state.

$\hat{I}_{|\psi\rangle}$ can be seen as an inversion around $|\psi\rangle$ operation, while $\hat{R}_{|\psi\rangle}$ can be seen as a reflection around $|\psi\rangle$ operation. An arbitrary state $|\phi\rangle$ can be expressed as

$$|\phi\rangle = \alpha|\psi\rangle + \beta|\psi^\perp\rangle,$$

with $(\alpha, \beta) \in \mathbb{C}$ and $|\psi^\perp\rangle$ belongs to the subspace perpendicular to $|\psi\rangle$

$$\langle\psi|\psi^\perp\rangle = 0.$$

III.3. Analysis of Grover's algorithm

Then

$$\hat{I}_{|\psi\rangle} |\phi\rangle = -\alpha |\psi\rangle + \beta |\psi^\perp\rangle.$$

$\hat{I}_{|\psi\rangle}$ has flipped the phase of the component corresponding to $|\psi\rangle$.

$\hat{R}_{|\psi\rangle}$ has the opposite effect

$$\hat{R}_{|\psi\rangle} |\phi\rangle = \alpha |\psi\rangle - \beta |\psi^\perp\rangle.$$

\hat{U}_f is an Oracle such that

$$\hat{U}_f |x\rangle = (-1)^{f(x)} |x\rangle,$$

where one forgets the ancilla qubit required for unitary evolution.

In the unstructured search problem with a marked element x_p

$$\hat{U}_f = \hat{I}_{|x_p\rangle}.$$

III.3. Analysis of Grover's algorithm

Furthermore,

$$\hat{H}^{\otimes n} \hat{U}_0 \hat{H}^{\otimes n} = \hat{H}^{\otimes n} (\mathbb{I} - 2|0\rangle^{\otimes n} \langle 0|^{\otimes n}) \hat{H}^{\otimes n} = \mathbb{I} - 2\hat{H}^{\otimes n} |0\rangle^{\otimes n} \langle 0|^{\otimes n} \hat{H}^{\otimes n}.$$

Introducing the $|+\rangle$ state defined as follow

$$|+\rangle = \hat{H}^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle,$$

one obtains immediately $\hat{D} = -\hat{I}_{|+\rangle}$. After T iterations, the final state $|\psi_f\rangle$ measured is the following

$$\begin{aligned} |\psi_f\rangle &= (\hat{D} \hat{U}_f)^T \hat{H}^{\otimes n} |0\rangle^{\otimes n} = (\hat{D} \hat{U}_f)^T |+\rangle, \\ &= (-\hat{I}_{|+\rangle} \hat{I}_{|x_p\rangle})^T |+\rangle = (-\hat{R}_{|+\rangle} \hat{R}_{|x_p\rangle})^T |+\rangle, \end{aligned}$$

$$|\psi_f\rangle = (-\hat{R}_{|+\rangle} \hat{R}_{|x_p\rangle})^T |+\rangle.$$

III.3. Analysis of Grover's algorithm

Properties:

- 1 For any states $|\psi\rangle$, $|\phi\rangle$, and any state $|\xi\rangle$ in the plan defined by $|\psi\rangle$ and $|\phi\rangle$, the states $\hat{R}_{|\psi\rangle}|\xi\rangle$ and $\hat{R}_{|\phi\rangle}|\xi\rangle$ remain in this plan.
- 2 For two orthogonal states, $\hat{R}_{|\psi^\perp\rangle} = -\hat{R}_{|\psi\rangle}$.

Demonstration:

$$\begin{aligned} -\hat{R}_{|\psi\rangle}(\alpha|\psi\rangle + \beta|\psi^\perp\rangle) &= -\alpha|\psi\rangle + \beta|\psi^\perp\rangle \\ &= \hat{R}_{|\psi^\perp\rangle}(\alpha|\psi\rangle + \beta|\psi^\perp\rangle) \end{aligned}$$

- 3 If $|\xi\rangle$ is in the plan defined by two orthogonal states $|\phi\rangle$ and $|\phi^\perp\rangle$

$$\hat{R}_{|\phi\rangle}|\xi\rangle = \langle\phi|\xi\rangle|\phi\rangle - \langle\phi^\perp|\xi\rangle|\phi^\perp\rangle.$$

Demonstration is straightforward.

III.3. Analysis of Grover's algorithm

Consequently

$$|\psi_f\rangle = \left(\hat{R}_{|+\perp\rangle} \hat{R}_{|x_p\rangle} \right)^T |+\rangle.$$

Grover's algorithm is based on successive rotations around $|x_p\rangle$ and $|+\perp\rangle$, starting from the initial states superposition $|+\rangle$.

1 Introduction

2 Bernstein-Vazirani algorithm

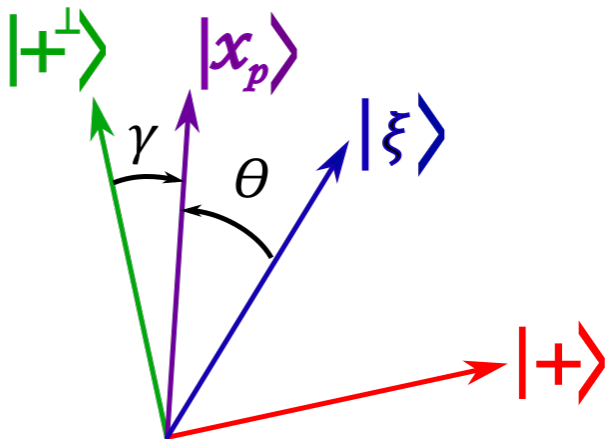
3 Grover's algorithm

- Grover's problem - unstructured search
- Grover's algorithm
- Analysis of Grover's algorithm
- **Geometrical interpretation of Grover's algorithm**
- Number of iterations

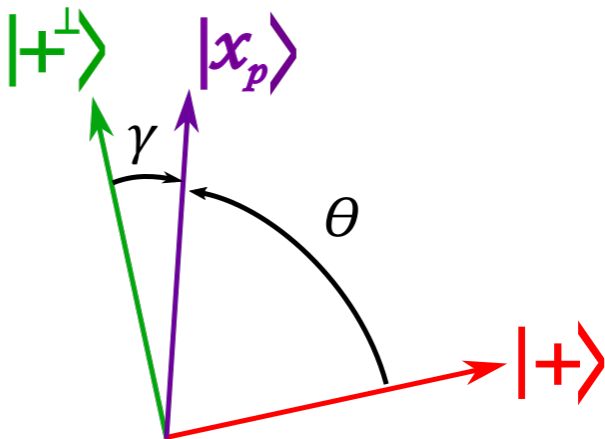
4 Grover's algorithm in the case of multiple marked elements

5 Phase estimation

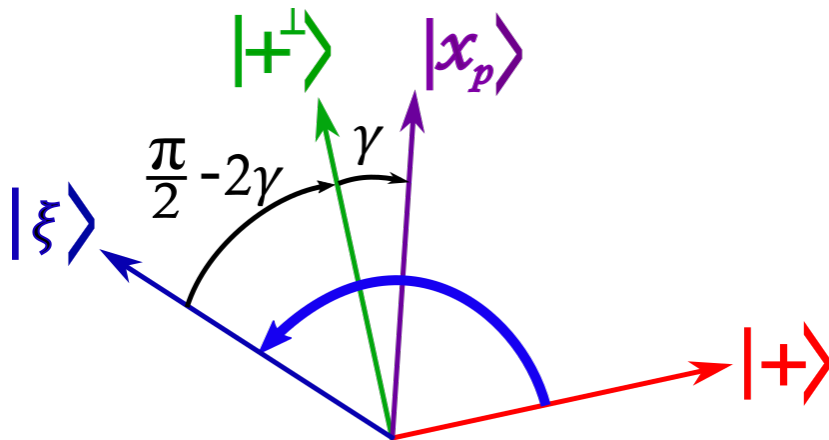
III.4. Geometrical interpretation of Grover's algorithm



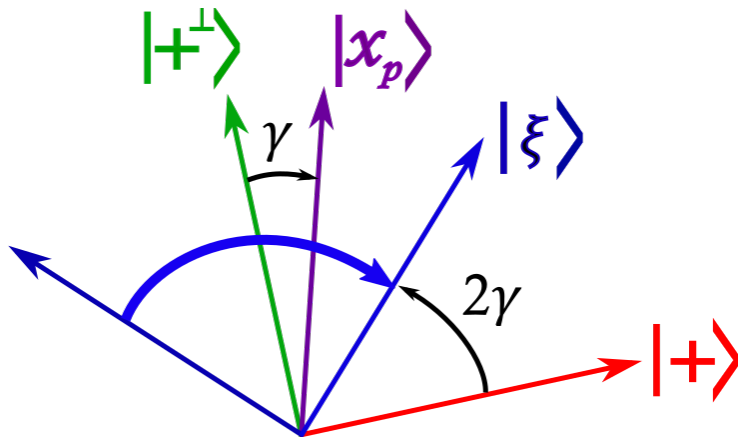
III.4. Geometrical interpretation of Grover's algorithm



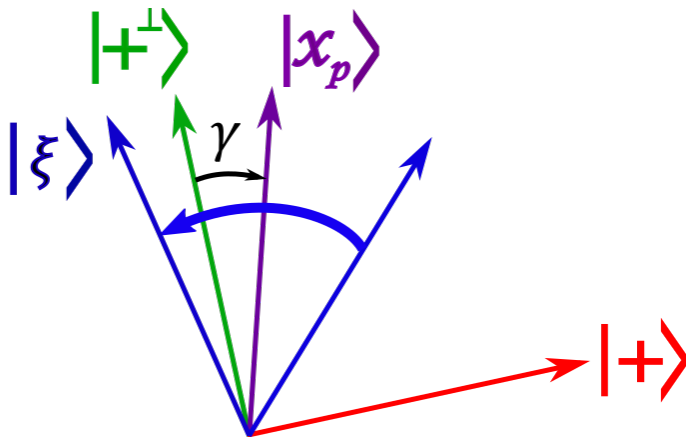
III.4. Geometrical interpretation of Grover's algorithm



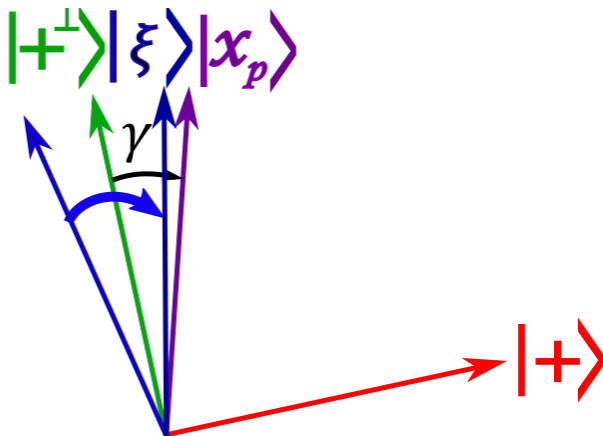
III.4. Geometrical interpretation of Grover's algorithm



III.4. Geometrical interpretation of Grover's algorithm



III.4. Geometrical interpretation of Grover's algorithm



III.4. Geometrical interpretation of Grover's algorithm

After each iteration, $|\xi\rangle$ moves closer to $|x_p\rangle$. In fact, the composition of two reflections around $|x_p\rangle$ and $|+\perp\rangle$ is a rotation. Let not θ the angle between $|\xi\rangle$ and $|x_p\rangle$, and γ the angle between $|x_p\rangle$ and $|+\perp\rangle$.

After $\hat{R}_{|x_p\rangle}$, $|\xi\rangle$ rotates by an angle 2θ anticlockwise. After $\hat{R}_{|+\perp\rangle}$, $|\xi\rangle$ rotates by an angle $2(\theta - \gamma)$ clockwise. Thus, after $\hat{R}_{|+\perp\rangle}\hat{R}_{|x_p\rangle}$, $|\xi\rangle$ rotates by an angle of

$$\Delta\theta = 2\theta - 2(\theta - \gamma) = 2\gamma.$$

$$\boxed{\Delta\theta = 2\gamma}.$$

After each iteration, the state has rotates within the plane defined by $|x_p\rangle$ and $|+\perp\rangle$ by an angle of 2γ .

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm**
 - Grover's problem - unstructured search
 - Grover's algorithm
 - Analysis of Grover's algorithm
 - Geometrical interpretation of Grover's algorithm
 - Number of iterations**
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation

III.5. Number of iterations

The iteration has to be stopped when $|\xi\rangle$ is close as much as possible to $|x_p\rangle$. We start with $|\xi\rangle = |+\rangle$, so the initial angle between $|\xi\rangle$ and $|x_p\rangle$ is $\frac{\pi}{2} - \gamma$. We can calculate γ as follow

$$\begin{cases} \cos \gamma = \langle x_p | +^\perp \rangle, \\ \sin \gamma = \langle x_p | + \rangle = \frac{1}{\sqrt{N}}, \end{cases} \quad (2)$$

because

$$|+\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

For large N ,

$$\sin \gamma \approx \gamma \approx \frac{1}{\sqrt{N}}.$$

III.5. Number of iterations

So the number of iterations M required to move from an angle $\frac{\pi}{2} - \gamma$ down to approximately 0 is

$$M \approx \frac{\frac{\pi}{2} - \gamma}{2\gamma} = \frac{\pi}{4\gamma} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{N} - \frac{1}{2}.$$

So in the limit where $N \gg \gg 1$,

$$M \approx \frac{\pi}{4}\sqrt{N}.$$

The number of iteration with a quantum algorithm scales as \sqrt{N} while with a classical algorithm, it scales as N .

III.5. Number of iterations

After T iterations, the angle between $|\xi\rangle$ and $|x_p\rangle$ is

$$\gamma_T = \frac{\pi}{2} - (2T + 1) \arcsin\left(\frac{1}{\sqrt{N}}\right),$$

so the probability of obtaining the outcome $|x_p\rangle$ when we measure it is precisely

$$|\langle \xi | x_p \rangle|^2 = \cos^2 \gamma_T = \sin^2\left((2T + 1) \arcsin\left(\frac{1}{\sqrt{N}}\right)\right).$$

Maximising this by taking T as the nearest integer to

$$\frac{\pi}{4 \arcsin\left(\frac{1}{\sqrt{N}}\right)} - \frac{1}{2} = \frac{\pi}{4} \sqrt{N} - \frac{1}{2} - \mathcal{O}\left(\frac{1}{N}\right).$$

We have access to x_p with a probability $1 - \mathcal{O}\left(\frac{1}{N}\right)$ using $\mathcal{O}\left(\sqrt{N}\right)$ queries (for small x , $\arcsin x \approx a + \mathcal{O}(x^3)$).

III.5. Number of iterations

Remark:

the optimum number of iteration is independent of x_p .

A particular nice case, where we can determine an exact solution for T , is for $N = 4$. Indeed,

$$\arcsin \frac{1}{2} = \frac{\pi}{6},$$

so if we plug in $T = 1$, the probability of getting x_p at the outcome is $\sin^2 \frac{\pi}{2} = 1$.

So we get the right answer only after 1 query for 4 possibilities of x_p !

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
 - Number of marked elements known
 - Number of marked elements unknown
 - Amplitude amplification
- 5 Phase estimation
- 6 Shor's algorithm

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
 - Number of marked elements known
 - Number of marked elements unknown
 - Amplitude amplification
- 5 Phase estimation
- 6 Shor's algorithm

IV.1. Number of marked elements known

Grover's algorithm can also be used when there are $M > 1$ marked elements. In this setting, the operator \hat{U}_f inverts the phase of inputs elements $x \in S$, for S an unknown subset of $\{0, 1\}^n$, and $\text{Card}(S) = M$.

\hat{U}_f is still related to a reflection operator, but now an inversion around a M -dimensional subspace

$$\hat{U}_f = \mathbb{I} - 2\hat{\Pi}_S,$$

where

$$\hat{\Pi}_S = \sum_{x \in S} |S\rangle\langle S|,$$

is the projector on the subspace generated by S .

IV.1. Number of marked elements known

Let's define the state $|S\rangle$ as follow

$$|S\rangle = \frac{1}{\sqrt{M}} \sum_{x \in S} |x\rangle.$$

Then,

$$\begin{aligned} \hat{I}_{|S\rangle} &= (\mathbb{I} - 2|S\rangle\langle S|)|+\rangle \\ &= |+\rangle - 2 \left(\frac{1}{M} \sum_{x,y \in S} |x\rangle\langle y| \right) \left(\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle \right). \end{aligned}$$

So

$$\hat{I}_{|S\rangle} = |+\rangle - \frac{2}{\sqrt{N}} \sum_{x \in S} |x\rangle = (\mathbb{I} - 2\hat{\Pi}_S)|+\rangle = \hat{U}_f|+\rangle.$$

IV.1. Number of marked elements known

Similarly

$$\hat{I}_{|S\rangle} |S\rangle = -|S\rangle = (\mathbb{I} - 2\hat{\Pi}_S) |S\rangle = \hat{U}_f |S\rangle.$$

Then, \hat{U}_f operation behaves like a reflection around $|S\rangle$ operator for any states in the subspace spanned by $|+\rangle$ and $|S\rangle$.

The whole of the previous analysis goes through, except that now the angle γ moved at each step satisfies

$$\sin \gamma = \langle S|+\rangle = \sqrt{\frac{M}{N}}.$$

IV.1. Number of marked elements known

Thus, after T iterations, we have

$$|\langle \xi | S \rangle|^2 = \cos^2 \gamma_T = \sin^2 \left((2T + 1) \arcsin \sqrt{\frac{M}{N}} \right).$$

To obtain an overlap with $|S\rangle$ close to 1, it requires T iterations with

$$T \approx \frac{\pi}{4} \sqrt{\frac{N}{M}}.$$

At the end of the algorithm, one get an element of the subset S at the measurement (a uniformly random distribution of elements of S) with a probability $|\langle \xi | S \rangle|^2$.

For $M = \frac{N}{4}$, we again measure an element of S with certainty using only one query.

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
 - Number of marked elements known
 - Number of marked elements unknown
 - Amplitude amplification
- 5 Phase estimation
- 6 Shor's algorithm

IV.2. Number of marked elements unknown

Now the number of marked elements is not known (noted M'). In that case, one first runs the algorithm assuming there is only 1 marked element. If it fails, try again assuming there are 2 marked elements. Then 4, 8, etc... The total number of queries used is roughly

$$\sum_{k=0}^{\log_2 N} \frac{\pi}{4} \sqrt{\frac{N}{2^k}} = \frac{\pi}{4} \sqrt{N} \sum_{k=0}^{\log_2 N} 2^{-k/2} = \mathcal{O}(\sqrt{N}).$$

If the number of marked elements is M' , at least one of the iterations must choose a guess M for M' such that

$$\frac{M'}{2} \leq M \leq 2M'.$$

This corresponds to a value of T which is within a factor of about $\sqrt{2}$ of the optimal value $T' \approx \frac{\pi}{4} \sqrt{\frac{N}{M'}}$.

IV.2. Number of marked elements unknown

Since

$$(2T' + 1) \arcsin \sqrt{\frac{M'}{N}} = \frac{\pi}{2} + \mathcal{O}\left(\sqrt{\frac{M'}{N}}\right),$$

then

$$\begin{aligned} \sin^2\left((2T + 1) \arcsin \sqrt{\frac{M'}{N}}\right) &= \sin^2\left(\frac{2T+1}{2T'+1} (2T' + 1) \arcsin \sqrt{\frac{M'}{N}}\right) \\ &= \sin^2\left(\frac{2T+1}{2T'+1} \left(\frac{\pi}{2} + \mathcal{O}\left(\sqrt{\frac{M'}{N}}\right)\right)\right), \end{aligned}$$

which is lower-bounded by a strictly positive constant of M is small with respect to N .

IV.2. Number of marked elements unknown

Repeating the whole algorithm $\mathcal{O}(1)$ times, and checking each time whether the returned element is marked, allows to achieve an arbitrary high success probability.

This algorithm might still have a high probability of failing in the case where $M = \mathcal{O}(N)$. To find a marked element in this case, we can just sample $\mathcal{O}(1)$ random values of $f(x)$ classically ; we will find a marked element with high probability.

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
 - Number of marked elements known
 - Number of marked elements unknown
 - **Amplitude amplification**
- 5 Phase estimation
- 6 Shor's algorithm

IV.3. Amplitude amplification

The idea of Grover's algorithm might be generalized to an algorithm for finding heuristic solutions to any problems. This algorithm is known as **amplitude amplification**.

Imagine we have $N = 2^n$ possible solutions, of which a subset S are "good", and we would like to find a good solution. As well as having access to a "checking" algorithm f as before, where $f(x) = 1$ if and only if x is marked, we now have access to a "guessing" algorithm \hat{A} , which has the job of producing potential solution to the problem.

It performs the map

$$\hat{A}|0\rangle^{\otimes n} = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle,$$

with $\alpha_x \in \mathbb{C}$.

IV.3. Amplitude amplification

After applying \hat{A} , the probability that we would obtain a good solution after measurement is

$$p = \sum_{x \in S} |\alpha_x|^2.$$

We may consider \hat{A} as an heuristic try for output of a good solution. We can use f afterwards to check whether a claimed solution is actually good. If we repeated the algorithm \hat{A} until we got a good solution, the expected number of trials we would need is $\mathcal{O}\left(\frac{1}{p}\right)$.

Amplitude amplification algorithm:

We are given access to \hat{A} and \hat{U}_f .

- 1 Apply \hat{A} to initial state $|0\rangle^{\otimes n}$.
- 2 Repeat the following operations T times, for some T to be determined
 - 1 Apply \hat{U}_f .
 - 2 Apply $-\hat{A}\hat{U}_0\hat{A}^{-1}$.
- 3 Measure all the qubits and output the result.

IV.3. Amplitude amplification

Let introduce

$$|\psi\rangle = \hat{\mathcal{A}}|0\rangle^{\otimes n},$$

and

$$|G\rangle = \frac{\hat{\Pi}_S |\psi\rangle}{\|\hat{\Pi}_S |\psi\rangle\|}, \quad \text{with} \quad \hat{\Pi}_S = \sum_{x \in S} |x\rangle\langle x|.$$

The previous analysis is still valid, replacing $|+\rangle$ with $|\psi\rangle$ and $|S\rangle$ with $|G\rangle$. The first operation applied is equivalent to $\hat{I}_{|G\rangle}$ and the second is equivalent to $-\hat{I}_{|\psi\rangle}$.

IV.3. Amplitude amplification

We start with the state $|\psi\rangle$ and rotate it toward $|G\rangle$. The angle γ moved at each step is such that

$$\sin \gamma = \langle \psi | G \rangle = \|\hat{\Pi}_G |\psi\rangle\| = \sqrt{p},$$

so the number of iterations required to move from $|\psi\rangle$ to $|G\rangle$ is $\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$, which is a quadratic improvement.

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation**
 - Quantum Fourier Transformation
 - Periodicity determination with QFT
 - Phase estimation
- 6 Shor's algorithm

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation**
 - Quantum Fourier Transformation
 - Periodicity determination with QFT
 - Phase estimation
- 6 Shor's algorithm

V.1. Quantum Fourier Transformation

We now introduce an important unitary transformation which is used in a number of different contexts in quantum information theory

the quantum Fourier transform (QFT) over \mathbb{Z}_N ,

where \mathbb{Z}_N is the ensemble of integer modulo N .

QFT might be seen as a generalization of the Hadamard gate, which has the following map

$$\hat{H}^{\otimes n} = \frac{1}{\sqrt{2^n}} (|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|)^{\otimes n}.$$

The QFT map is the following

$$\hat{Q}_N |x\rangle = \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{x \cdot y} |y\rangle,$$

where $\omega_N = e^{\frac{2i\pi}{N}}$, but $x \cdot y$ is the product of x and y as integer of \mathbb{Z}_N .

V.1. Quantum Fourier Transformation

Exemple:

$$Q_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad Q_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{\frac{2i\pi}{3}} & e^{-\frac{2i\pi}{3}} \\ 1 & e^{-\frac{2i\pi}{3}} & e^{\frac{2i\pi}{3}} \end{pmatrix},$$

$$Q_4 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}.$$

Note that the QFT is unitary.

Demonstration: Let consider the inner product of rows x and z

$$\frac{1}{N} \sum_{y \in \mathbb{Z}_N} (\omega_N^{x \cdot y})^* \omega_N^{z \cdot y} = \frac{1}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{(z-x) \cdot y}.$$

V.1. Quantum Fourier Transformation

Or

$$\sum_{k=0}^{r-1} x^k = \begin{cases} \frac{1-x^r}{1-x} & \text{if } x \neq 1, \\ r & \text{if } x = 1. \end{cases} \quad (3)$$

Given that $\omega_N^N = 1$, the inner product is then 0 if $z \neq x$, and 1 otherwise ($z = x$). More generally, for any integer j ,

$$\frac{1}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{j \cdot y} = \begin{cases} 0 & \text{if } j \neq 0 [N], \\ 1 & \text{if } j = 0 [N]. \end{cases} \quad (4)$$

Then the QFT is unitary.

The QFT is a similar transformation than the Discrete Fourier Transformation (DFT) used for classical computation and signal processing, up to the non standard normalization of $1/\sqrt{N}$.

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation**
 - Quantum Fourier Transformation
 - Periodicity determination with QFT**
 - Phase estimation
- 6 Shor's algorithm

V.2. Periodicity determination with QFT

Let consider a function

$$f : \mathbb{Z}_N \longrightarrow \mathbb{Z}_M,$$

for $(N, M) \in \mathbb{N}^2$ such that

- 1 f is periodic: there is a r such that

$$\forall x \in \mathbb{Z}_N, f(x + r) = f(x),$$

- 2 f is one-to-one on each period

$$\forall (x, y) \in \mathbb{Z}_N \text{ such that } |x - y| < r, f(x) \neq f(y).$$

The goal is to determine r .

V.2. Periodicity determination with QFT

The periodicity determination algorithm is the following. We start in the state $|0\rangle^{\otimes N} |0\rangle^{\otimes M}$.

- 1 Apply \hat{Q}_N to the first register.
- 2 Apply \hat{O}_f to the two registers (the Oracle).
- 3 Measure the second register.
- 4 Apply \hat{Q}_N to the first register.
- 5 Measure the first register ; let the answer be k .
- 6 Simplify the fraction $\frac{k}{N}$ as far as possible and return the denominator.

V.2. Periodicity determination with QFT

$$|0\rangle^{\otimes N} |0\rangle^{\otimes M} \xrightarrow{1)} \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle |0\rangle^{\otimes M} \xrightarrow{2)} \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle |f(x)\rangle.$$

When the second register is measured, we receive an answer, say z . Since f is periodic and one-to-one,

$$\exists x_0 \text{ such that } f(x_0) = z.$$

Consequently

$$\forall x \in \mathbb{Z}_N \text{ such that } f(x) = z, \exists j \in \mathbb{Z} \text{ such that } x = x_0 + jr.$$

The state collapses then to something of the following form

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |x_0 + jr\rangle,$$

which means there is N/r states in a period.

V.2. Periodicity determination with QFT

After we apply the QFT, we get the state

$$\frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \left(\sum_{y \in \mathbb{Z}_N} \omega_N^{y \cdot (x_0 + jr)} |y\rangle \right) = \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{y \cdot x_0} \left(\sum_{j=0}^{\frac{N}{r}-1} \omega_N^{j \cdot r \cdot y} \right) |y\rangle$$

Observe that, as r divides N ,

$$\omega_N^r = e^{\frac{2i\pi r}{N}} = \omega_{\frac{N}{r}}.$$

This state is then equivalent to

$$\frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{y \cdot x_0} \left(\sum_{j=0}^{\frac{N}{r}-1} \omega_{\frac{N}{r}}^{j \cdot y} \right) |y\rangle$$

V.2. Periodicity determination with QFT

$$\sum_{j=0}^{\frac{N}{r}-1} \omega_N^{j \cdot y} = 0 \quad \text{unless } y \equiv 0 \left[\frac{N}{r} \right], \quad \text{in other words, if } y = l \frac{N}{r}, l \in \mathbb{Z}.$$

This state might be rewrite as follow

$$\frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} \omega_N^{l \cdot x_0 \cdot \frac{N}{r}} \left| l \frac{N}{r} \right\rangle.$$

When me perform the final measurement, we receive an outcome

$$k = \frac{l_0 N}{r},$$

for some l_0 picked uniformly at random from $\{0, \dots, r-1\}$ so

$$\frac{k}{N} = \frac{l_0}{r}.$$

If l_0 is coprime to r , we could cancel the fraction $\frac{k}{N}$ and output the denominator. For a integer, b picked-up uniformly at random from 0 to a the probability that b is coprime to a is

$$\mathcal{O}\left(\frac{1}{\log(\log a)}\right).$$

The, if we repeat the procedure $\mathcal{O}(\log(\log r)) = \mathcal{O}(\log(\log N))$ times, we are likely to find the period r .

V.2. Periodicity determination with QFT

We have a probabilistic procedure which succeeds with probability p ; the probability that it fails every time over R repetitions is exactly

$$(1 - p)^R \leq e^{-pR},$$

so it suffices to take $R = \mathcal{O}\left(\frac{1}{p}\right)$ to achieve $\sim 99\%$ success probability.

Each time the algorithm returns a claimed period, we can check whether it is really a period of the function using two additional queries of the Oracle. Each use of the quantum algorithm therefore makes 3 queries of \hat{O}_f so it makes $\mathcal{O}(\log(\log N))$ queries in total.

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation**
 - Quantum Fourier Transformation
 - Periodicity determination with QFT
 - Phase estimation**
- 6 Shor's algorithm

Phase estimation is an important quantum computing primitive routine. Often used as an ingredient of more complex algorithms:

- integer factorisation ;
- matrix inversion ;
- quantum counting ;
- quantum walks.

Let consider an operator \hat{A} , of eigenvectors \vec{x} and eigenvalues λ

$$\hat{A}\vec{x} = \lambda\vec{x}.$$

In the case of a unitary matrix

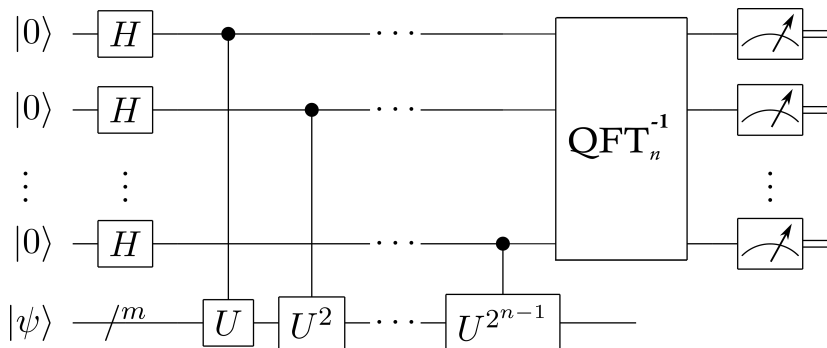
$$\hat{U}|x\rangle = e^{2i\pi\theta}|x\rangle,$$

with $|x\rangle$ an eigenvector and θ the phase of the eigenvalue $e^{2i\pi\theta}$.

Phase estimation algorithm: given \hat{U} and $|x\rangle$, estimate θ .

V.3. Phase estimation

The circuit diagram implementation of the phase estimation algorithm is the following



- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm**
 - Factoring
 - Shor's algorithm

7 Hamiltonian simulation

Quantum algorithms

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm**
 - Factoring
 - Shor's algorithm
- 7 Hamiltonian simulation

The most famous application of quantum computers

$$N = a \times b \longrightarrow \text{find } a \text{ and } b \text{ given } N,$$

where a and b are prime numbers.

Important modern crypto-systems (e.g. RSA) rely on this problem being intractable for computers. But a quantum computer can solve it quickly !

Example: the recommended key size for RSA is 2048 bits.

- Best known classical algorithm \sim 1 billion years.
- Shor's algorithm \sim 100 seconds !!!! (QC at 1 GHz).

Reference:

A compare between Shor's quantum factoring algorithm and general Number Field Sieve, Hamdi *et al.*

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm**
 - Factoring
 - **Shor's algorithm**

7 Hamiltonian simulation

VI.2. Shor's algorithm

Shor's algorithm consists in reducing the problem of factoring to the problem of period finding.

It uses a quantum algorithm for fast period finding.

Shor's algorithm

- 1 If N is even, return $f = 2$.
- 2 If $N = p^k$ for p prime, return p .
- 3 Randomly choose $1 < q < N - 1$.
If $f = \gcd(q, N) > 1$, return f
- 4 Determine the order k of q modulo N . (Phase estimation).
If k is odd, repeat from step 3.
- 5 Write $k = 2l$ and determine $q^l \bmod N$ with $1 < r < N$.
 - 1 If $1 < f = \gcd(r - 1, N) < N$, return f .
 - 2 If $1 < f = \gcd(r + 1, N) < N$, return f .
 - 3 Else repeat step 3.

VI.2. Shor's algorithm

All steps, excepted step 4, can be performed efficiently by a classical computer.

Given an n -bit integer

- classical number field sieve: $\mathcal{O}\left(2^{n^{1/3}}\right)$.
- Shor's algorithm: $\mathcal{O}\left(n^3\right)$.

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation**
 - Context
 - HHL

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation**
 - Context
 - HHL

VII.1. Shor's algorithm

In quantum mechanics, physical systems are described by Hamiltonians. The evolution is given by Schrödinger's equation

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = \hat{H}(t) |\psi(t)\rangle.$$

For a stationary hamiltonian, $|\psi(t)\rangle = e^{-i\frac{\hat{H}t}{\hbar}} |\psi(0)\rangle$

Hamiltonian simulation:

Given a Hamiltonian \hat{H} , construct a quantum circuit that approximates $e^{-i\frac{\hat{H}t}{\hbar}}$.

There are a number of quantum algorithm that can do this efficiently for certain type of Hamiltonian.

To simulate a classical system like a plane, a classical computer is appropriated. But to simulate a quantum system like a molecule, a quantum computer is better suited.

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation**
 - Context
 - HHL**

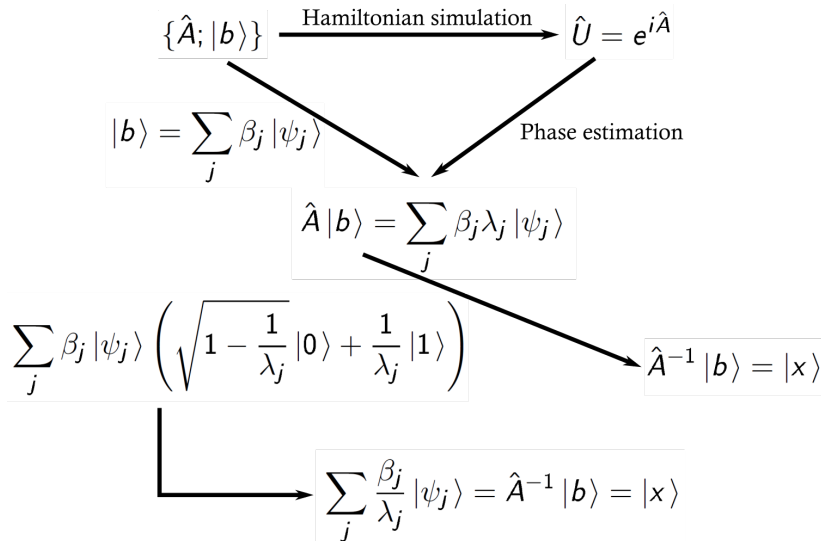
Named after Harrow, Hassidim and Lloyd, who invented it in 2008. It attacks one of the most fundamental tasks in science: solving systems of linear equations

$$A\vec{x} = \vec{b}, \text{ solve for } \vec{x}.$$

Classically: it takes polynomial time in the size of the matrix, whereas HHL "solves" this problem in logarithmic time.

Quantum algorithm HHL: inputs $|b\rangle$ and \hat{A} , outputs quantum state $|x\rangle$.

HHL algorithm outline



- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation**
 - Context
 - HHL

- Solving systems of differential equations (finite element method, FEM).
- Data fitting.
- Various tasks in machine learning (clustering, support-vector machines, principal component analysis).

Run time: for a system of n equations.

- classical: $\mathcal{O}(n^3)$;
- quantum: $\mathcal{O}\left(\kappa s \frac{\log n}{\varepsilon}\right)$, with κ the condition number, s the sparsity and ε the accuracy.

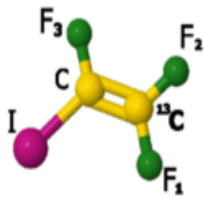
- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation**
 - Context
 - HHL

VII.4. Applications

- 1** Experimental Quantum Computing to Solve Systems of Linear Equations,
X.-D. Cai *et al.*, Phys. Rev. Lett. **110**, 230501 (2013).
Problem solved: 2×2 linear equations.
Qubits: photons (polarization).
- 2** A two-qubit photonic quantum processor and its application to solving systems of linear equation,
S. Barz *et al.*, Sci. Rep. **4**, 6115 (2014).
Problem solved: 2×2 linear equations.
Qubits: photons (polarization).
- 3** Experimental realization of quantum algorithm for solving linear systems of equations,
J. Pan *et al.*, Phys. Rev. A **89**, 022313 (2014).
Problem solved: 2×2 linear equations.
Qubits: NMR type qubits in a molecule of iodotrifluoroethylene $^{12}\text{C}^{13}\text{CF}_3\text{I}$.

VII. Introduction

	^{13}C	F_1	F_2	F_3
^{13}C	15479.7 Hz			
F_1	-297.7 Hz	-33122.4 Hz		
F_2	-275.7 Hz	64.6 Hz	-42677.7 Hz	
F_3	39.1 Hz	51.5 Hz	-129.0 Hz	-56445.8 Hz
T_2^*	1.22 s	0.66 s	0.63 s	0.61 s
T_2	7.9 s	4.4 s	6.8 s	4.8 s



J. Pan *et al.*, Phys. Rev. A 89, 022313 (2014)

Quantum algorithms

- 1 Introduction
- 2 Bernstein-Vazirani algorithm
- 3 Grover's algorithm
- 4 Grover's algorithm in the case of multiple marked elements
- 5 Phase estimation
- 6 Shor's algorithm
- 7 Hamiltonian simulation

VIII. Quantum error correction

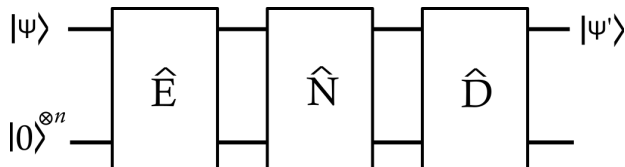
In classical computing, error correcting codes preserve classical bits. Quantum error correcting code will preserve a qubit $|\psi\rangle$ in quantum computing.

Let consider an error affecting one or more qubits is simply an arbitrary (unknown) unitary operator \hat{N} applied to those qubits (\hat{N} is a noise operator). The classical bit-flip is an example, and corresponds to the application of the operator \hat{X} .

$$\hat{X} |0\rangle = |1\rangle \text{ and } \hat{X} |1\rangle = |0\rangle.$$

VIII. Quantum error correction

The process of correcting errors in a qubits state $|\psi\rangle$ might be described as follow



- \hat{E} is an **encoding** unitary operator,
- \hat{N} is a **noise** unitary operator,
- \hat{D} is a **decoding** unitary operator,

VIII. Quantum error correction

We encode some qubit state $|\psi\rangle$ in a larger state $|E(\psi)\rangle$ using n ancilla qubits (initially on the state $|0\rangle^{\otimes n}$).

Some noise is applied through \hat{N} , and later we decode the noisy encoded state to produce a state $|\psi'\rangle$.

Goal of the process: $|\psi'\rangle \approx |\psi\rangle$

VIII. Quantum error correction

Non cloning theorem: it is not possible to duplicate a state $|\psi\rangle$ in the general case

$$|\psi\rangle \not\rightarrow |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle \cdots \otimes |\psi\rangle.$$

The error protection can't be performed by cloning the state $|\psi\rangle$.

VIII. Quantum error correction

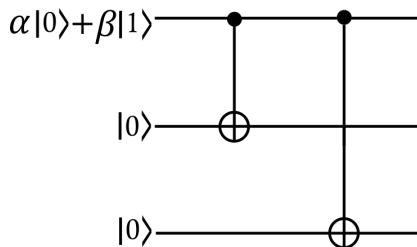
Principle of error correction code: let consider $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Then, encode it as follow

$$|E(\psi)\rangle = \alpha|000\rangle + \beta|111\rangle.$$

Remark: it is not a cloning !

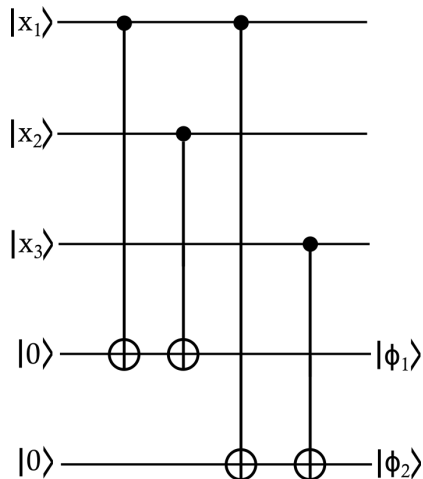
$$|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle = (\alpha|0\rangle + \beta|1\rangle)^{\otimes 3}.$$

The operator \hat{E} might be implemented as follow

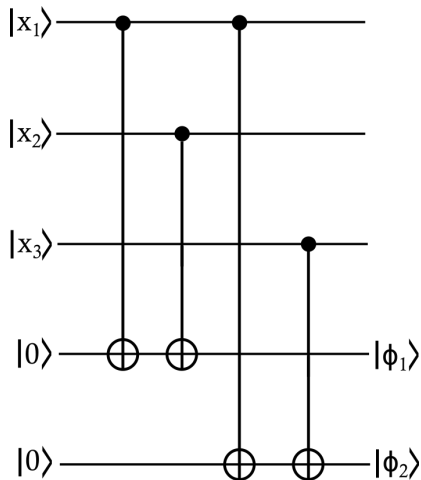


VIII. Quantum error correction

The decoding algorithm for this code will be based on the following circuit



VIII. Quantum error correction



The first three qubits are called input qubits. The last two qubits are called output qubits.

$$|\phi_1\rangle = |x_1 \oplus x_2\rangle,$$

$$|\phi_2\rangle = |x_1 \oplus x_3\rangle.$$

$x_1 \oplus x_2$ and $x_1 \oplus x_3$ are invariant under the flipping of all the bits of x .

VIII. Quantum error correction

After the application of \hat{N}

$$\hat{N} |E(\psi)\rangle = \alpha |x_1 x_2 x_3\rangle + \beta |x_1 x_2 x_3 \oplus 111\rangle.$$

The circuit proposed performs the following map

$$\begin{aligned} & (\alpha |x_1 x_2 x_3\rangle + \beta |x_1 x_2 x_3 \oplus 111\rangle) \otimes |0\rangle \otimes |0\rangle \\ & \longrightarrow (\alpha |x_1 x_2 x_3\rangle + \beta |x_1 x_2 x_3 \oplus 111\rangle) \otimes |x_1 \oplus x_2\rangle \otimes |x_1 \oplus x_3\rangle. \end{aligned}$$

If one measures the two output qubits, we learn both $x_1 \oplus x_2$ and $x_1 \oplus x_3$ without disturbing the input qubits. The encoded state $|\psi\rangle$ is always of this form, even after arbitrary bit-flip errors are applied to $|E(\psi)\rangle$.

VIII. Quantum error correction

$$|E(\psi)\rangle = \alpha |000\rangle + \beta |111\rangle.$$

Effect of bit-flip on $|E(\psi)\rangle$

$$\left(\hat{X} \otimes \mathbb{I} \otimes \mathbb{I}\right) |E(\psi)\rangle = \alpha |100\rangle + \beta |011\rangle,$$

$$\left(\hat{X} \otimes \hat{X} \otimes \hat{X}\right) |E(\psi)\rangle = \alpha |111\rangle + \beta |000\rangle.$$

The result of measuring the output qubits is known as the **syndrome**. What are the syndromes of different noise operators \hat{N} applied to $|E(\psi)\rangle$?

If $\hat{N} = \mathbb{I}$, we always measure 00.

If $\hat{N} = \hat{X} \otimes \mathbb{I} \otimes \mathbb{I}$, we always obtain 11.

VIII. Quantum error correction

Syndrome measured for different bit-flip noise:

\hat{N}	Syndrome
$I \otimes I \otimes I$	00
$I \otimes I \otimes \hat{X}$	01
$I \otimes \hat{X} \otimes I$	10
$\hat{X} \otimes I \otimes I$	11
$I \otimes \hat{X} \otimes \hat{X}$	11
$\hat{X} \otimes \hat{X} \otimes I$	01
$\hat{X} \otimes I \otimes \hat{X}$	10
$\hat{X} \otimes \hat{X} \otimes \hat{X}$	00

If the error occurs on a single qubit, it is possible to detect it, and apply the corresponding bit-flip operation on the corresponding qubit to restore the original encoded state $\alpha |000\rangle + \beta |111\rangle$. On the other hand, if bit flip errors occurs on more than one qubit, one does not detect them.

VIII. Quantum error correction

In the case of \hat{Z} noise,

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

the syndrome measurement always return 00, so the error correction operation does nothing and the \hat{Z} error is not corrected. But

$$\hat{Z} = \hat{H}\hat{X}\hat{H},$$

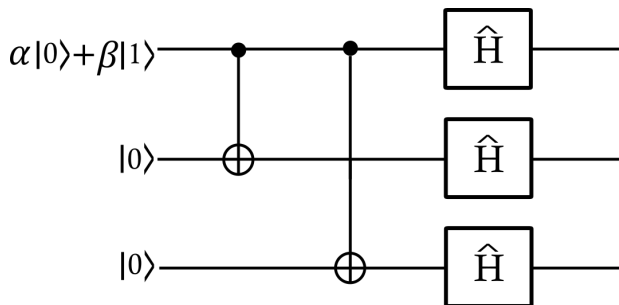
where \hat{H} is the Hadamard gate. Thus \hat{Z} acts in the same way as \hat{X} , up to a change of basis.

If we use the same code as before, but perform this change of basis for each qubit, we obtain a code which corrects against \hat{Z} errors. In other words, we now encode $|\psi\rangle$ as $\alpha|+++ \rangle + \beta|--- \rangle$, with

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{and} \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

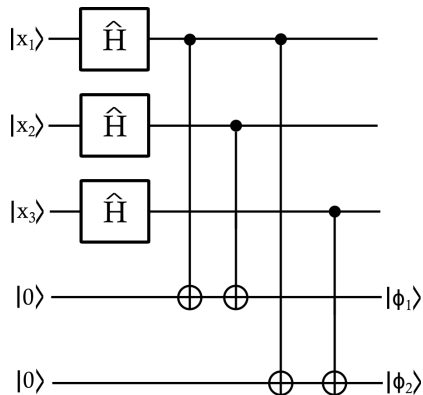
VIII. Quantum error correction

The new encoding circuit is then



VIII. Quantum error correction

and the decoding circuit



VIII. Quantum error correction

But it does no longer protects against \hat{X} errors !!! It is possible to concatenate these two codes.

We first encode $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ using the code protecting against phase flips, and then encode each of the resulting qubits using the code that protects against bit flips. In other words, we perform the following map

$$\begin{aligned} |\psi\rangle &= \alpha|0\rangle + \beta|1\rangle, \\ \longrightarrow \frac{1}{2\sqrt{2}} & (\alpha(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) \\ & + \beta(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)), \end{aligned}$$

VIII. Quantum error correction

$$\begin{aligned} \longrightarrow \frac{1}{2\sqrt{2}} & (\alpha(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \\ & + \beta(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)), \end{aligned}$$

The single qubit $|\psi\rangle$ is now encoded using 9 qubits.

These qubits can naturally be split into three blocks, each of which encodes one qubit of the state

$$\alpha|+++ \rangle + \beta|--- \rangle.$$

To decode this encoded state, first the decoding circuit for the bit-flip code is applied to each block. Assuming at most one bit-flip error has occurred in each block, the result will be the state $\alpha|+++ \rangle + \beta|--- \rangle$, perhaps with a \hat{Z} error applied to one of the qubits. This state can then be mapped back to $\alpha|0\rangle + \beta|1\rangle$ using the decoding algorithm for the phase-flip code.

VIII. Quantum error correction

This quantum error-correcting code was the first such code discovered. It was invented by Peter Shor in 1995, known as Shor's 9 qubit code.

This work is licensed under a Creative Commons “Attribution-NonCommercial-NoDerivatives 4.0 International” license.



<https://creativecommons.org/licenses/by-nc-nd/4.0/>