ALGO2 – Algorithmique quantique

François Schwarzentruber

3 mai 2022

"I think I can safely say that nobody understands quantum mechanics."
Richard Feynman

`https://francoisschwarzentruber.github.io/quantumalgowithcats/`

1 Motivation

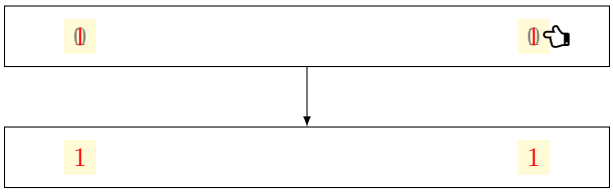
1.1 Explosion exponentielle

- Problème de factorisation d'un nombre entiers : Algorithme de Shor [Sho94]
- Problème de recherche d'une solution : Algorithme de Grover [Gro96]
- Simulation numérique pour développer de nouveaux médicaments [LWG+10]
- Utilisation d'un ordinateur quantique pour comprendre la matière et la physique quantique [PODDB+12] [SKPK19]
- Apprentissage automatique quantique [BWP+17]

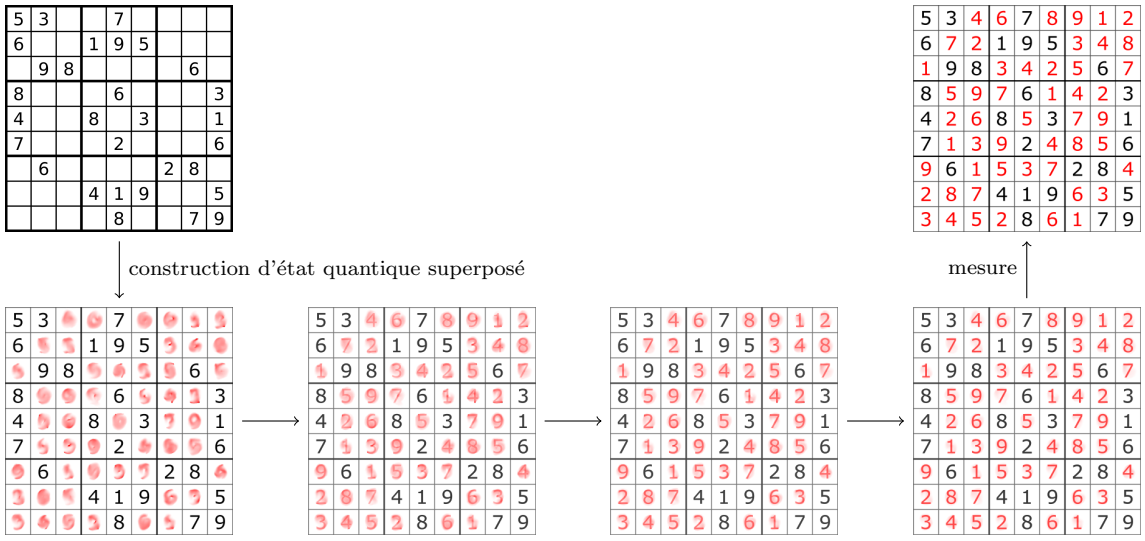
1.2 Avantage de la physique quantique

Superposition : le système peut être dans plusieurs états physiques

Il peut y avoir intrication : changer un qubit en affecte un autre



Archétype d'un algorithme quantique qui parallélise le calcul



1.3 Un effort industriel

— Chez IBM : `https://quantum-computing.ibm.com`
Librairie Python3 `qiskit` pour manipuler des algorithmes quantiques
Langage assembleur de description de circuits quantiques : Open Quantum Assembly Language (OpenQASM)
— Chez Atos : Librairie Python3 `pyAQASM` Assembleur : aQASM
— Chez Microsoft : Q#
— Chez Google :`https://quantumai.google/`

Défi : conserver un état quantique, i.e. éviter les interactions
— Architecture en circuit où données et opérations sont ensemble pour réduire les déplacements des électrons (surtout pas d'aller retour RAM $\leftrightarrow$ CPU !)
— Superconducteurs, froid

	ordinateur classique	ordinateur quantique
unité d'information	bit $\in \{0,1\}$	q-bit $\in \mathbb{C}^2$
mémoire	n bits	n qubits
état d'un programme	état physique $s \in \{0,1\}^n$	superposition d'états physiques $s \in \mathbb{C}^{2^n}$
calcul	déterministe	déterministe
mesure	déterministe	probabiliste

Comme les q-bits peuvent être intriqués, l'état d'un programme quantique est $\mathbb{C}^{2^n} \equiv \mathbb{C}^{\{0,1\}^n}$ et non pas $\mathbb{C}^{2n}$.

2 États quantiques

Définition 1 (état quantique) Un état quantique [1] sur n q-bits est un vecteur de $\mathbb{C}^{\{0,1\}^n}$ de norme 1. Les coordonnées sont repérées par des suites de bits de longueur n.

Notation 2 (notation de Dirac - bra-ket) Les vecteurs unités, où seule une coordonnée pour un certain mot $u \in \{0,1\}^n$ vaut 1 et les autres coordonnées valent 0 se note $|u\rangle$. Un tel vecteur représente un *état classique*.

2.1 Exemple d'un qu-bit ($n = 1$)

Notation 3 $|0\rangle$ est le vecteur $\begin{pmatrix}1\\0\end{pmatrix}$. $|1\rangle$ est le vecteur $\begin{pmatrix}0\\1\end{pmatrix}$.

Exemple 1 $|0\rangle$ désigne le qu-bit avec spin up.

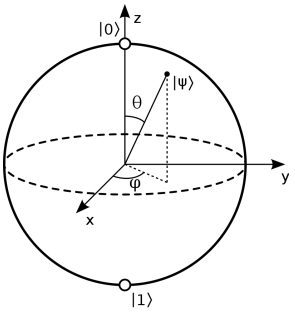
Exemple 2 $|1\rangle$ désigne le qu-bit avec spin down.

Notation 4 L'état quantique d'un qu-bit est un vecteur $\begin{pmatrix}a\\b\end{pmatrix}$ de $\mathbb{C}^2$ que l'on note

$$a\,|0\rangle + b\,|1\rangle$$

où $a, b \in \mathbb{C}$ avec $|a|^2 + |b|^2 = 1$.

Remarque 5 (sphère de Bloch) L'état d'un qubit est un point sur la sphère de Bloch :



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

1. En réalité, c'est modulo un décalage de phase donc il faut quotienter l'ensemble des vecteurs de norme 1 de $\mathbb{C}^{\{0,1\}^n}$ par la relation $\sim$ définie par $|x\rangle \sim |y\rangle$ ssi il existe $\alpha \in \mathbb{C}$ de norme 1 avec $|x\rangle = \alpha|v\rangle$. Mais, nous n'avons pas besoin de cette lourdeur administrative dans ce cours.

2.2 Exemple de deux qubits

Notation 6 (état classique) Avec 2 qubits, les quatre états classiques sont :

$$|00\rangle \qquad |01\rangle \qquad |10\rangle \qquad |11\rangle$$

qui dans une notation plus standard en maths se notent :

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

Notation 7 (état quantique quelconque) Avec 2 qubits, un état quantique est un vecteur $\begin{pmatrix}a\\b\\c\\d\end{pmatrix}$ de $\mathbb{C}^4$ que l'on note

$$a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$$

où $a, b, c, d \in \mathbb{C}$ avec $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$.

Exemple 3 (état non intriqué) L'état suivant est non intriqué :

$$\frac{1}{2}\,|00\rangle + \frac{1}{2}\,|01\rangle + \frac{1}{2}\,|10\rangle + \frac{1}{2}.$$

Exemple 4 (état de Bell) L'état de Bell est l'état d'intrication suivant :

$$\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|11\rangle\,.$$

2.3 Exemple de trois qu-bits

Exemple 8 Avec 3 qubits, les états physiques sont :

$$|000\rangle \qquad |001\rangle \qquad |010\rangle \qquad |011\rangle \qquad |100\rangle \qquad |101\rangle \qquad |110\rangle \qquad |111\rangle\,.$$

Ce sont :

$$|000\rangle := \begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} \; |001\rangle := \begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix} \; |010\rangle := \begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{pmatrix} \; |011\rangle := \begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{pmatrix} \; |100\rangle := \begin{pmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix} \; |101\rangle := \begin{pmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{pmatrix} \; |110\rangle := \begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix} \; |111\rangle := \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}.$$

Exemple 5 Un état quantique de 3 q-bits est un vecteur $\sum_{x\in\{0,1\}^3} \alpha_x\,|x\rangle$ dans $\mathbb{C}^{\{0,1\}^3}$ avec $\sum_{x\in\{0,1\}^3} |\alpha_x|^2 = 1$:

$$\alpha_{000}\,|000\rangle + \alpha_{001}\,|001\rangle + \alpha_{010}\,|010\rangle + \alpha_{011}\,|011\rangle + \alpha_{100}\,|100\rangle + \alpha_{101}\,|101\rangle + \alpha_{110}\,|110\rangle + \alpha_{111}\,|111\rangle$$

2.4 Notations générales

Notation 9 On utilise aussi la notation $|\varphi\rangle$ pour désigner un vecteur $\varphi \in \mathbb{C}^{\{0,1\}^n}$.

Notation 10 Si $|\varphi\rangle \in \mathbb{C}^{\{0,1\}^n}$, on note $|\varphi\rangle = \sum_{x\in\{0,1\}^n} \varphi_x\,|x\rangle$.

3 Mesure

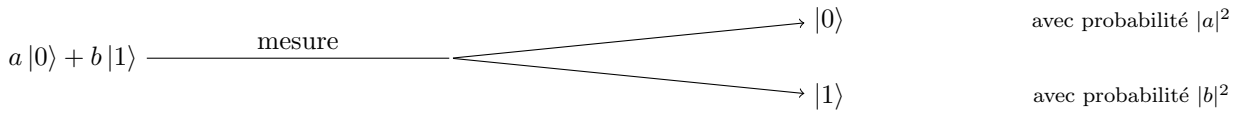
3.1 Mesure complète d'un état

Définition 11 (mesure complète) Considérons l'état $\sum_{x\in\{0,1\}^n} \alpha_x\,|x\rangle$ où $\sum\limits_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$.

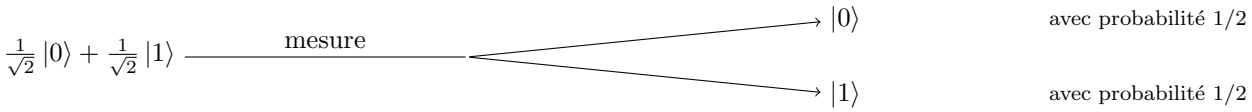
La mesure de cet état s'effectue comme suit :

1. le système choisit $x \in \{0,1\}^n$ avec une probabilité de $|\alpha_x|^2$;
2. l'utilisateur lit cette valeur de x ;
3. le nouveau état devient l'état physique $|x\rangle$.

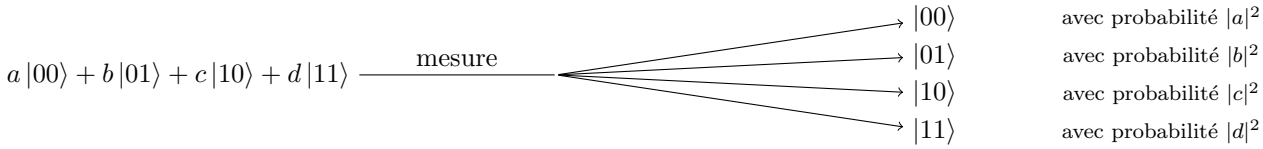
3.1.1 Mesure d'un qubit



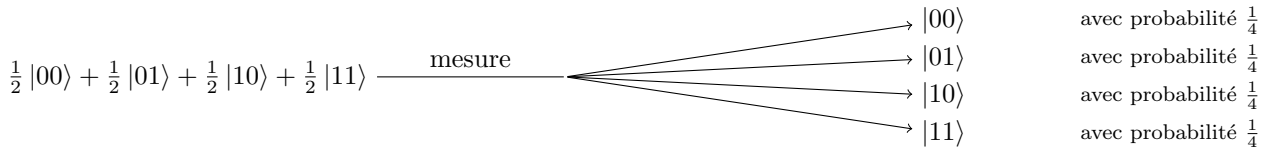
Exemple 6



3.1.2 Mesure de deux qubits

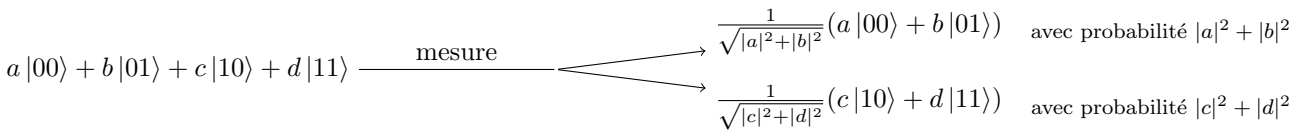


Exemple 7

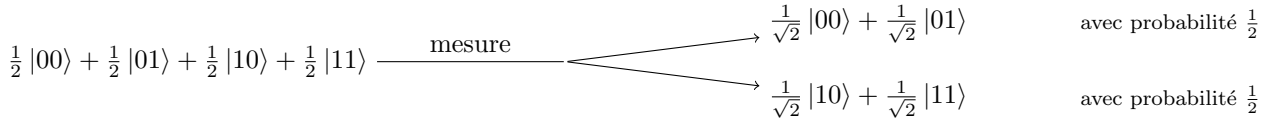


3.2 Mesure partielle

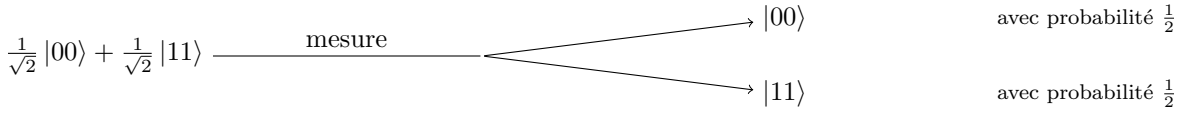
3.2.1 Mesure partielle du premier bit parmi deux qubits



Exemple 8



Exemple 9



3.3 En général

Exemple 10 (mesure du premier qubit) Considérons l'état $\sum_{x\in\{0,1\}^3} \alpha_x |x\rangle$ avec $\sum_{x\in\{0,1\}^3} |\alpha_x|^2 = 1$.

Mesurons le premier qubit :
— $\mathbb{P}(\text{1er bit} = 0) = \sum_{x\in\{0,1\}^3 | \text{le 1er bit de } x \text{ est } 0} |\alpha_x|^2$.
— $\mathbb{P}(\text{1er bit} = 1) = \sum_{x\in\{0,1\}^3 | \text{le 1er bit de } x \text{ est } 1} |\alpha_x|^2$.

Le nouvel état s'obtient en supprimant les termes inconsistants avec la mesure. Par exemple, si on mesure le premier qubit à 0, on ne garde que les termes dont le premier bit est 0, puis on renormalise. Le nouvel état est :

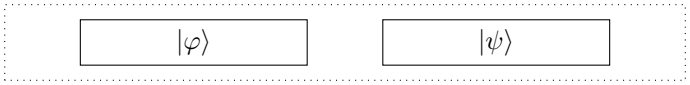
$$\frac{1}{\sqrt{|\alpha_{000}|^2+|\alpha_{001}|^2+|\alpha_{010}|^2+|\alpha_{011}|^2}}\alpha_{000}\,|000\rangle + \alpha_{001}\,|001\rangle + \alpha_{010}\,|010\rangle + \alpha_{011}\,|011\rangle\,.$$

Définition 12 (mesure d'un sous-ensemble de qubits) Soit l'état $\sum_{x\in\{0,1\}^n}\alpha_x\,|x\rangle$ où $\sum_{x\in\{0,1\}^n}|\alpha_x|^2{=}1$. La mesure des qubits $J\subseteq\{1,\ldots,n\}$ s'effectue comme suit :

1. le système choisit $(y_j)_{j\in J}\in\{0,1\}^J$ avec une probabilité de $\sum_{x\in\{0,1\}^n|x_J=y_J}|\alpha_x|^2$.
2. L'utilisateur lit la valeur choisie $(y_j)_{j\in J}\in\{0,1\}^J$.
3. Le nouvel état est :
$$\frac{1}{\sqrt{\sum_{x\in\{0,1\}^n|x_J=y_J}|\alpha_x|^2}}\sum_{x\in\{0,1\}^n|x_J=y_J}\alpha_x\,|x\rangle\,.$$

4 Intrication et produit tensoriel

Soit deux systèmes indépendants (*non intriqués*) d'états quantiques respectifs $|\varphi\rangle$ et $|\psi\rangle$. Le produit tensoriel $|\varphi\rangle\otimes|\psi\rangle$ représente l'état quantique du système global.



Exemple 11 Considérons deux qubits non intriqués

$$|\varphi\rangle = a\,|0\rangle + b\,|1\rangle \qquad \text{et} \qquad |\psi\rangle = c\,|0\rangle + d\,|1\rangle.$$

L'état quantique global superposé est le *produit tensoriel*

$$|\varphi\rangle\otimes|\psi\rangle = ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle\,.$$

Définition 13 (produit tensoriel) Soit $|\varphi\rangle=\sum_{x\in\{0,1\}^n}\varphi_x\,|x\rangle$ dans $\mathbb{C}^{2^n}$ et $|\psi\rangle=\sum_{y\in\{0,1\}^k}\psi_y\,|y\rangle$ dans $\mathbb{C}^{2^k}$. Le produit tensoriel de $|\varphi\rangle$ et $|\psi\rangle$ est le vecteur $\mathbb{C}^{2^{n+k}}$ défini par :

$$|\varphi\rangle\otimes|\psi\rangle = \sum_{x\in\{0,1\}^n,y\in\{0,1\}^k}\varphi_x\psi_y\,|xy\rangle\,.$$

Exemple 12 (état non intriqué) L'état

$$\frac{1}{2}\,|00\rangle + \frac{1}{2}\,|01\rangle + \frac{1}{2}\,|10\rangle + \frac{1}{2}\,|11\rangle$$

est non intriqué car il s'écrit comme

$$(\frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle)\otimes(\frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle).$$

Exemple 13 (état de Bell) L'état de Bell est

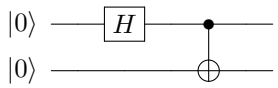
$$\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|11\rangle$$

est un état d'intrication : il ne peut pas s'exprimer comme $|\varphi\rangle\otimes|\psi\rangle$ où $|\varphi\rangle$ et $|\psi\rangle$ sont deux qubits.

5 Circuits quantiques

Définition 14 (informelle) Un *circuit quantique* à n qubits est une suite finie de *portes*, chacune opérant sur un sous-ensemble de qubits.

Exemple 15



5.1 Quelques portes

Porte	Effet	Matrice
Négation	$\|0\rangle$ —[NOT]— $\|1\rangle$; $\|1\rangle$ —[NOT]— $\|0\rangle$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Hadamard	$\|0\rangle$ —[H]— $\frac{1}{\sqrt{2}}\|0\rangle + \frac{1}{\sqrt{2}}\|1\rangle$; $\|1\rangle$ —[H]— $\frac{1}{\sqrt{2}}\|0\rangle - \frac{1}{\sqrt{2}}\|1\rangle$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Non contrôlé	$\|00\rangle \to \|00\rangle$; $\|01\rangle \to \|01\rangle$; $\|10\rangle \to \|11\rangle$; $\|11\rangle \to \|10\rangle$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

Remarque 16 Les circuits sont linéaires. Par exemple :

$$a|0\rangle + b|1\rangle \text{ —[}NOT\text{]— } a|1\rangle + b|0\rangle$$

Proposition 17 $H^2 = Id_2$.

5.2 Fonction linéaire unitaire (*)

Définition 18 (matrice adjointe) Soit U une matrice carrée de taille $N \times N$. La matrice adjointe de U, notée U^*, est la matrice transposée de la matrice conjuguée, i.e. :

$$U^*_{i,j} = \overline{U_{j,i}}$$

pour tout $i, j \in \{1, \ldots, N\}$.

Exemple 14 $\begin{pmatrix} 4+i & 5 \\ 2+2i & i \end{pmatrix}^* = \begin{pmatrix} 4-i & 2-2i \\ 5 & -i \end{pmatrix}$.

Définition 19 (fonction linéaire unitaire) Une fonction linéaire unitaire de $\mathbb{C}^{2^n}$ dans $\mathbb{C}^{2^n}$ est une fonction

$$\begin{array}{ccc} \mathbb{C}^{2^n} & \to & \mathbb{C}^{2^n} \\ |\varphi\rangle & \mapsto & U|\varphi\rangle \end{array}$$

où U est une matrice à coefficients complexes de taille $2^n \times 2^n$ telle que $UU^* = I_{2^n}$.

Définition 20 Une porte sur k bits est une partie d'un circuit, munie d'une fonction linéaire unitaire de $\mathbb{C}^{2^k}$ dans $\mathbb{C}^{2^k}$.

5.3 Hadamard en parallèle sur $|0 \ldots 0\rangle$

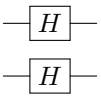
$$\begin{array}{l} |0\rangle \text{ —[}H\text{]—} \\ \quad\vdots \\ |0\rangle \text{ —[}H\text{]—} \end{array}$$

Chaque porte H transforme sont qu-bits $|0\rangle$ en : $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
L'état global s'obtient par produit tensoriel :

$$\tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \ldots \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{u \in \{0,1\}^n} |u\rangle$$

5.4 Portes en parallèle (*)

5.4.1 Deux portes Hadamard en parallèle



Supposons que l'état quantique initial soit le tenseur de $a\,|0\rangle + b\,|1\rangle$ et $c\,|0\rangle + d\,|1\rangle$, i.e. :

$$ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle\,.$$

La première porte Hadamard transforme le premier qubit en :

$$\frac{1}{\sqrt{2}}(a+b)\,|0\rangle + \frac{1}{\sqrt{2}}(a-b)\,|1\rangle$$

La deuxième porte Hadamard transforme le deuxième qubit en :

$$\frac{1}{\sqrt{2}}(c+d)\,|0\rangle + \frac{1}{\sqrt{2}}(c-d)\,|1\rangle$$

L'état quantique final global s'obtient en prenant le tenseur :

$$\frac{1}{2}(a+b)(c+d)\,|00\rangle + \frac{1}{2}(a+b)(c-d)\,|01\rangle + \frac{1}{2}(a-b)(c+d)\,|10\rangle + \frac{1}{2}(a-b)(c-d)\,|11\rangle\,.$$

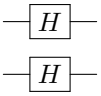
5.4.2 Cas général (*)

En général, voici comment combiner deux portes. Considérons un système avec k premiers qubits suivis de m qubits. Soit $A = (a_{ij})_{i,j\in\{0,1\}^k}$ et B une matrice de taille 2^m.

Alors l'exécution parallèle de l'application de A qui opère sur les k premiers qubits qui et B qui opère sur les m derniers qubits, est donnée par la matrice de taille 2^{k+m} suivante :

$$\begin{pmatrix} a_{00\ldots0,00\ldots0}B & a_{00\ldots0,00\ldots1}B & \ldots & a_{00\ldots0,11\ldots1}B \\ a_{00\ldots1,00\ldots0}B & a_{00\ldots1,00\ldots1}B & \ldots & a_{00\ldots1,11\ldots1}B \\ \vdots & \vdots & \ldots & \vdots \\ a_{11\ldots1,00\ldots0}B & a_{11\ldots1,00\ldots1}B & \ldots & a_{11\ldots1,11\ldots1}B \end{pmatrix}$$

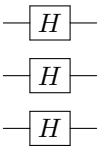
Exemple 15



L'exécution parallèle de deux portes Hadamard est :

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

Exemple 16 L'exécution parallèle de trois portes Hadamard est :



$$H_3 = \frac{1}{2^{\frac{3}{2}}}\begin{pmatrix} 1&1&1&1&1&1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&1&1&1&-1&-1&-1&-1\\1&-1&1&-1&-1&1&-1&1\\1&1&-1&-1&-1&-1&1&1\\1&-1&-1&1&-1&1&1&-1 \end{pmatrix}$$

Proposition 21 Le résultat de n portes Hadamard en parallèle sur n qubits $|0\rangle$ est :

$$\sum_{x\in\{0,1\}^n} \frac{1}{2^{\frac{n}{2}}} |x\rangle .$$

DÉMONSTRATION. Par récurrence sur n. ■

Définition 22 Soit $i, j \in \{0,1\}^n$. $i \bullet j$ est le produit scalaire des vecteurs de bits correspondant à i et j.

Exemple 17 $011 \bullet 100 = 0 \times 1 + 1 \times 0 + 1 \times 0 = 0$.

Proposition 23 La matrice $H^{\otimes n}$ de n portes d'Hadamard en parallèle est

$$H^{\otimes n}_{i,j} = \frac{1}{2^{\frac{n}{2}}}(-1)^{i\bullet j}$$

DÉMONSTRATION. Par récurrence sur n. ■

6 Algorithme de Grover

Dans toute cette section, n désigne le nombre de qubits et $M = 2^n$ le nombre d'états physiques possibles.

6.1 Problème de recherche d'une solution

Recherche d'une solution

entrée : une boîte noire implémentant une fonction $f : \{0,1\}^n \to \{0,1\}$ telle qu'il existe un unique $\sigma \in \{0,1\}^n$ tel que $f(\sigma) = 1$.
sortie : l'élément σ avec une probabilité $\geq \frac{1}{2}$.

Exemple 24 Considérons une grille de Sudoku qui admet une unique solution σ.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

solution σ

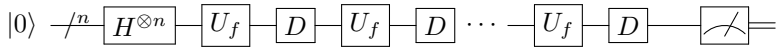
On considère la fonction

$$f : \begin{array}{ccc} \{0,1\}^n & \to & \{0,1\} \\ \text{complétion } x \text{ de la grille} & \mapsto & \begin{cases} 1 & \text{si } x \text{ respecte les contraintes du Sudoku} \\ 0 & \text{sinon} \end{cases} \end{array}$$

Proposition 25 L'algorithme déterministe naïf pour décider la recherche d'une solution est en temps $\Theta(2^n)$ dans le pire cas, si on suppose que la boîte noire est en $O(1)$.

6.2 Principe de l'algorithme

Algorithme de Grover



où

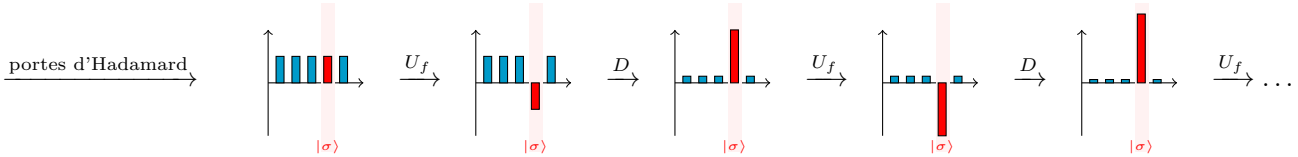
— $H^{\otimes n}$ représente n portes Hadamard en parallèle, une sur chaque qubit ;

— U_f est un circuit qui représente la fonction f de la façon suivante :

$$U_f(x) = \begin{cases} -x & \text{si } x = \sigma \\ x & \text{sinon} \end{cases}$$

— D est appelé l'opérateur de diffusion de Grover défini par la matrice :

$$\begin{pmatrix} 2/M-1 & 2/M & \cdots & 2/M \\ 2/M & 2/M-1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 2/M \\ 2/M & \cdots & 2/M & 2/M-1 \end{pmatrix}$$



Remarque 26 (implémentation de D) On a :

$$D = H^{\otimes n} R H^{\otimes n}$$

où $H^{\otimes n}$ est la matrice pour n portes Hadamard en parallèle et $R = \begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & -1 & \dots & 0 \\ \vdots & & \dots & & \\ 0 & 0 & \dots & 0 & -1 \end{pmatrix}$.

6.3 Effet des itérations

Notation 27 (somme des états physiques autres que σ) $|\rho\rangle := \sum_{x\in\{0,1\}^n, x\neq\sigma} |x\rangle$.

Proposition 28 La k-ème itération est :

$$s_{k-1}|\sigma\rangle + r_{k-1}|\rho\rangle \xrightarrow{U_f} -s_{k-1}|\sigma\rangle + r_{k-1}|\rho\rangle \xrightarrow{D} s_k|\sigma\rangle + r_k|\rho\rangle$$

avec $s_0 = r_0 = \frac{1}{\sqrt{N}}$; $\quad s_{k+1} := \left(\frac{N-2}{N}s_k + 2\frac{N-1}{N}r_k\right)$ et $r_{k+1} := \left(-\frac{2}{N}s_k + \frac{N-2}{N}r_k\right)$.

Démonstration. Montrons par récurrence sur k qu'au début de la k-ème itération est $s_k|\sigma\rangle + r_k|\rho\rangle$.
Initialisation. Juste après les portes d'Hadamard, le système est dans l'état

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

d'où les valeurs de s_0 et r_0.

Hérédité.
Comme U_f ne touche à rien sauf qu'il retourne $|\sigma\rangle$ on a :

$$s_k|\sigma\rangle + r_k|\rho\rangle \xrightarrow{U_f} -s_k|\sigma\rangle + r_k|\rho\rangle$$

Puis on a :

$$-s_k|\sigma\rangle + r_k|\rho\rangle \to^D s_{k+1}|\sigma\rangle + r_{k+1}|\rho\rangle.$$

En effet,

$$D(-s_k|\sigma\rangle + r_k|\rho\rangle) = 2(-s_k P|\sigma\rangle + r_k P|\rho\rangle) + s_k|\sigma\rangle - r_k|\rho\rangle$$

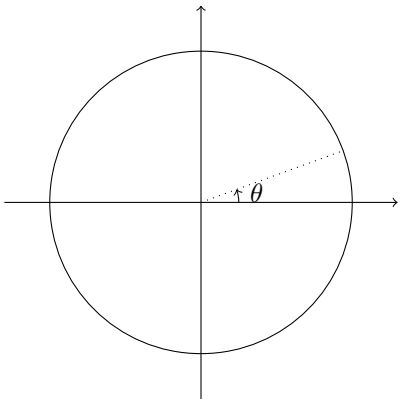
et $P|\sigma\rangle = \frac{1}{N}|\sigma\rangle + \frac{1}{N}|\rho\rangle$ et $P|\rho\rangle = \frac{N-1}{N}|\sigma\rangle + \frac{N-1}{N}|\rho\rangle$. ■

6.4 Calcul du nombre d'itérations

Notation 29 Dans la suite, on note $\theta \in]0, \frac{\pi}{2}[$ l'angle tel que $\sin\theta = \frac{1}{\sqrt{N}}$.

Proposition 30 Pour tout entier $k \geq 0$, on a :

$$s_k = \sin((2k+1)\theta) \qquad r_k = \tfrac{1}{\sqrt{N-1}}\cos((2k+1)\theta).$$



Démonstration. Par récurrence sur k. Laissé en exercice en vénérant l'art de la trigonométrie. ■

Théorème 31 Avec un nombre d'itération en $\ell = O(\sqrt{N})$, on mesure σ avec une probabilité $\geq 1 - \frac{1}{N}$.

Démonstration. Idéalement, on aurait $(2\ell^* + 1)\theta = \frac{\pi}{2}$, i.e. $\ell^* = \frac{\pi}{4\theta} - \frac{1}{2}$. Mais cette valeur n'est pas forcément un entier. Soit ℓ l'entier le plus proche de ℓ^*.

$$\frac{\dfrac{\ell \leq \frac{\pi}{4\theta} \qquad \sin\theta \leq \theta}{\ell \leq \frac{\pi}{4\sin\theta} = \frac{\pi\sqrt{N}}{4} = O(\sqrt{N})}}{}$$

La probabilité de mesurer σ est $|s_\ell|^2 = s_\ell^2$. La probabilité de ne pas mesurer σ est $1 - s_\ell^2$.

$$\begin{aligned} 1 - s_\ell^2 &= 1 - \sin^2((2\ell+1)\theta) \\ &= \cos^2((2\ell+1)\theta)) \\ &= \sin^2((2\ell+1)\theta - \frac{\pi}{2})) \\ &\leq \sin^2\theta \text{ voir plus bas} \\ &= \frac{1}{N} \end{aligned}$$

$$\frac{\dfrac{\dfrac{\dfrac{\ell \text{ l'entier le plus proche de } \ell^*}{|\ell - \ell^*| \leq \frac{1}{2}}}{|(2\ell+1) - (2\ell^*+1)| \leq 1} \qquad \ell^* = \frac{\pi}{4\theta} - \frac{1}{2}}{|(2\ell+1)\theta - \frac{\pi}{2}| \leq \theta}}{\dfrac{\sin^2 x = \sin^2|x| \qquad \sin(|(2\ell+1)\theta - \frac{\pi}{2}|) \leq \sin\theta}{\sin^2((2\ell+1)\theta - \frac{\pi}{2}) \leq \sin^2\theta}}$$

■

7 Transformée de Fourier quantique

7.1 Rappel du problème

Soit $M = 2^n$.

Définition 32 **Transformée de Fourier**

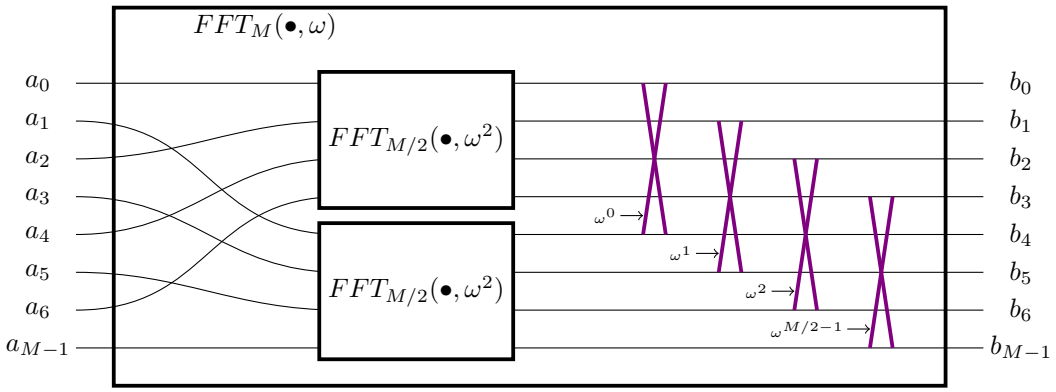
entrée : $(a_0, \ldots, a_{M-1}) \in \mathbb{C}^M$, une racine M-ème de l'unité primitive ω ;
sortie : $A(\omega^0), A(\omega), \ldots, A(\omega^{M-1})$ où $A = \sum_{i=0}^{M-1} a_i X^i$.

7.2 Algorithme classique

entrée : des nombres $a_0, \ldots, a_{M-1}$, et une racine M-ème de l'unité primitive ω
sortie : $A(\omega^0), A(\omega), \ldots, A(\omega^{M-1})$ où $A = \sum_{i=0}^{M-1} a_i \mathsf{X}^i$

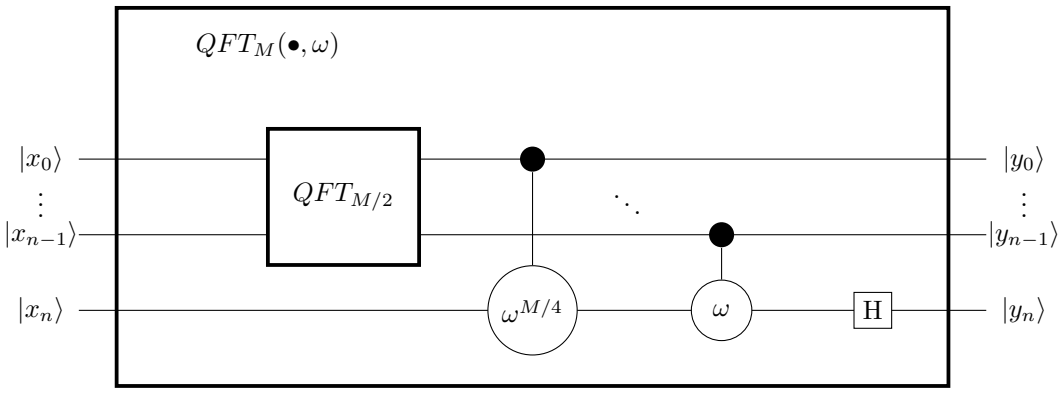
fonction FFT$(a_0, \ldots, a_{M-1},\ \omega)$
  si $M = 1$
    renvoyer a_0
  sinon
    $(s_0, \ldots, s_{M/2-1}) := FFT(a_0, a_2, \ldots, a_{M-2},\ \omega^2)$
    $(s'_0, \ldots, s'_{M/2-1}) := FFT(a_1, a_3, \ldots, a_{M-1},\ \omega^2)$
    pour $k = 0$ à $M/2 - 1$
      $b_k := s_k + \omega^k s'_k$
      $b_{k+M/2} := s_k - \omega^k s'_k$ opération papillon
  renvoyer $b_0, \ldots, b_{M-1}$

Proposition 33 La FFT réalise le calcul en $O(M \log M)$.



7.3 Circuit quantique

Le circuit est composé d'abord de l'appel récursif qui réalise la "FFT en parallèle" sur les $n-1$ premiers qu-bits. Puis on a des *portes de changement de phase contrôlé*. Par exemple, la première porte vient modifier la phase en multipliant par $\omega^{M/4}$ un coefficient d'une coordonnée où $x_0 = x_n = 1$. Pour les autres coefficients, cette porte ne fait rien. Cette suite de changement de phase correspond au calcul de $\omega^j s'_k$ qui est un préalable à l'opération papillon. Enfin, les opérations papillon sont réalisées par une porte d'Hadamard sur le dernier qu-bit (oui oui, ça marche !).



En sortie, on récupère le résultat de la FFT mais normalisée (car il faut que le module du vecteur soit 1).

Proposition 34 QFT est en temps $O(\log^2 M)$.

Démonstration. Soit $T(n)$ la complexité pour le calcul de $QFT(|x\rangle)$ sur une entrée avec n qubits. Cette entrée code bien $M = 2^n$ nombres complexes $a_0, \ldots, a_{M-1}$. On a :

$$T(n) = T(n-1) + O(n).$$

D'où une complexité temporelle $T(n) = O(n^2) = O(\log_2 M)$.
■

Théorème 35 Notons $(b_0, \dots, b_{M-1}) = FFT(a_0, \dots, a_{M-1}, \omega)$.
Soit $|x\rangle := \sum_{u\in\{0,1\}^n} a_u|u\rangle$. L'état quantique après $QFT_{M,\omega}(|x\rangle)$ est

$$|y\rangle := \frac{1}{\sqrt{M}} \sum_{u\in\{0,1\}^n} b_u|u\rangle.$$

Démonstration. On note $\Sigma = \{0,1\}$. Par récurrence sur k, on démontre la propriété suivante $\mathcal{P}(k)$:

$$QFT_{2^k}(|x\rangle) = \frac{1}{\sqrt{2^k}} \sum_{uv\in\Sigma^k\times\Sigma^{n-k}} b_{uv}|uv\rangle$$

où pour tout $v \in \Sigma^{n-k}$, on a $(b_{uv})_{u\in\Sigma^k} = FFT((a_{uv})_{u\in\Sigma^k}, \omega^{2^{n-k}})$.
Dans la propriété précédente, v dénote les bits de poids faibles et u les bits de fort. Le circuit de QFT_{2^k} n'agit que ces k bits de poids fort. La propriété dit que la FFT a été appliqué sur les vecteurs obtenus en fixant les bits de points faibles.
$\boxed{\mathcal{P}(1)}$ C'est ok.
$\boxed{\text{Cas récursif}}$ Supposons $\mathcal{P}(k-1)$ et démontrons $\mathcal{P}(k)$. On a :

$$QFT_{2^{k-1}}(|x\rangle) = \frac{1}{\sqrt{2^{k-1}}} \sum_{uv\in\Sigma^{k-1}\times\Sigma^{n-k+1}} b_{uv}|uv\rangle$$

où pour tout $v \in \Sigma^{n-k+1}$, on a $(b_{uv})_{u\in\Sigma^{k-1}} = FFT((a_{uv})_{u\in\Sigma^{k-1}}, \omega^{2^{n-k}})$.
En particulier, pour tout $v \in \Sigma^{n-k}$, on récupère les valeurs $(b_{u0v})_{u\in\Sigma^{k-1}}$ et $(b_{u1v})_{u\in\Sigma^{k-1}}$.
Effet des portes de déphasage. Les portes de déphasage viennent lire le k bits (le 0 et 1 apparent dans $u0v$ et $u1v$) :
— les valeurs b_{u0v} ne sont pas modifiées ;
— $b_{u0v} := \omega^{M/4} b_{u0v}$ si la première lettre de u est un 1 ;
— $b_{u0v} := \omega^{M/8} b_{u0v}$ si la 2ème lettre de u est un 1 ;
— $\vdots$
— $b_{u0v} := \omega^{1} b_{u0v}$ si la $k-1$-ème lettre (= la dernière lettre) de u est un 1.
Ainsi, si on interprète u comme un nombre (i.e. u est l'écriture binaire d'un nombre que l'on continue à noter u), l'ensemble de l'effet des portes de déphasage correspond à faire

$$b_{u1v} := \omega^u b_{u1v}.$$

Effet de la porte d'Hadamard. Notons $|y\rangle = \sum_{w\in\Sigma^n} y_w|w\rangle$. La porte d'Hadamard réalise :

$$\begin{cases} y_{u0v} := \frac{1}{\sqrt{2}}(b_{u0v} + \omega^u b_{u1v}) \\ y_{u1v} := \frac{1}{\sqrt{2}}(b_{u0v} - \omega^u b_{u1v}) \end{cases}$$

où les valeurs $b_{\dots}$ sont les valeurs juste après le circuit $QFT_{M/2}$.
Ainsi on a pour tout $v \in \Sigma^{n-k}$, on a

$$(y_{uv})_{u\in\Sigma^k} = \frac{1}{\sqrt{2^k}} FFT((a_{uv})_{u\in\Sigma^k}, \omega^{2^{n-k}}).$$

CQFD. ■

8 Algorithme de Shor

8.1 Problème

Définition 36 **Factorisation**
entrée : un entier N écrit en binaire
sortie : un facteur de N

Définition 37 (problème de décision) **Factorisation**
entrée : un entier N, un entier k, tous deux écrits en binaire
sortie : oui, si N admet un facteur de plus grand que k, non

Proposition 38 **Factorisation** est dans NP ∩ coNP.

Démonstration.

1. Montrons que **FACTORISATION** est dans NP. Voici un algorithme non-déterministe en temps polynomial. Tout d'abord, on choisit de manière non déterministe un entier p dans $\{k, \dots, n-1\}$. On teste si p divise n et si p est premier en temps polynomial.
2. Montrons que **FACTORISATION** est dans coNP. Voici un algorithme non-déterministe pour le problème dual **FACTORISATION**. L'idée est de choisir une factorisation $p_1 \dots p_j$ de n avec $p_i \in \{2, \dots, k-1\}$. Une telle factorisation s'écrit comme un mot de la forme $\langle p_1 \rangle \# \langle p_2 \rangle \dots \langle p_j \rangle$ où $\langle p_i \rangle$ est l'écriture binaire de p_i. Comme $p_1 \dots p_j = n \geq 2^j$ donc $j \leq \log n$. Ensuite, chaque p_i est un mot de longueur au plus $\log k$. Donc une telle factorisation est un mot polynomial en la taille de (n, k).

   Ainsi, on se propose l'algorithme non-déterministe suivant :

   (a) Choisir un mot de la forme $\langle p_1 \rangle \# \langle p_2 \rangle \dots \langle p_j \rangle$ où $j \leq \log n$ et chaque p_i est un mot de longueur au plus $\log k$

   (b) tester si tous les p_i sont premiers avec un algorithme polynomial puis on vérifie que $p_1 \dots p_k = n$.

■

Remarque 39 On ne sait pas si **Factorisation** est dans P.

8.2 Calcul de l'ordre d'un élément

On note n le nombre de bits pour représenter l'entier N à factoriser. On note $Q = 2^n$.

Définition 40 (ordre) L'ordre de x modulo N est le plus petit entier $r > 0$ tel que $x^r \equiv 1[N]$.

entrée : un entier g, un entier N tels que g et N soient premiers entre eux
sortie : l'ordre de g dans $\mathbb{Z}/N\mathbb{Z}^\times$ (ou échec)
fonction quantique ordre(g, N)
- créer deux registres de n qu-bits chacun
- créer de l'état superposé $\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x, 0^n\rangle$ où n est le nombre de bits pour écrire N et $Q = 2^n$
- considérons la fonction

$$\begin{array}{rccc} f: & \mathbb{Z}/N\mathbb{Z} & \to & \mathbb{Z}/N\mathbb{Z} \\ & x & \mapsto & g^x \mod N \end{array}$$

- calcul de l'état superposé $\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x, f(x)\rangle$ en utilisant l'algorithme d'exponentation rapide modulaire
- lire le deuxième registre (celui contenant les valeurs de $f(x)$)
- l'état du premier registre est $|\alpha\rangle =$ coefficient $\times \sum_{j=0}^{Q/r-1} |jr + k\rangle$
- appliquer QFT sur le premier registre
- **renvoyer** la période de f calculée avec l'algo de transformée de Fourier quantique

Explication de comment calculer l'ordre r avec la QFT.

Explication de comment calculer l'ordre r avec la QFT. Par définition de la transformée de Fourier, l'application de QFT sur $|x\rangle$ donne $QFT(|x\rangle) = \frac{1}{\sqrt{Q}} \sum_{y=0}^{Q-1} \omega^{xy} |y\rangle$ où ω est une racine Q de l'unité primitive. L'application de QFT sur le premier registre donne donc l'état quantique :

$$\frac{1}{Q} \sum_{x=0}^{Q-1} \sum_{y=0}^{Q-1} \omega^{xy} |y, f(x)\rangle.$$

Avec un peu de mathématiques, cet état est en fait égal à

$$\frac{1}{Q} \sum_{z=0}^{N-1} \sum_{y=0}^{Q-1} \underbrace{\sum_{x \in \{0,\dots,Q-1\} | f(x)=z} \omega^{xy}}_{\alpha_{yz}} |y, z\rangle.$$

On se retrouve avec un état où le coefficient devant chaque état physique $|x, z\rangle$ est $\sum_{x \in \{0,\dots,Q-1\} | f(x)=z} \omega^{xy}$. On va réécrire ce coefficient. Pour cela, on constate que l'ensemble des x tel que $f(x) = z$ n'est pas un ensemble complétement désordonné ! Il peut se mettre sous la forme :

$$\{x_z + br \mid 0 \leq b \leq m\}$$

où r est le rang de g, x_z est le plus petit entier tel que $f(x_z) = z$ et $m = \lfloor \frac{Q - x_z - 1}{r} \rfloor + 1$. Ainsi ce coefficient s'écrit :

$$\alpha_{yz} := \sum_{x\in\{0,\dots,Q-1\}|f(x)=z} \omega^{xy} = \sum_{b=0}^{m-1} \omega^{(x_z+br)y} = \omega^{x_z y} \sum_{b=0}^{m-1} \omega^{bry} = \omega^{x_z y} \frac{\omega^{mry}-1}{\omega^{ry}-1}$$

La probabilité de lire $|yz\rangle$ vaut $|\alpha_{yz}|^2$ et on a :

$$|\alpha_{yz}|^2 = \frac{1}{Q^2} \frac{\sin^2(\frac{mry}{Q})}{\sin^2(\frac{ry}{Q})}.$$

La probabilité précédente est la plus forte quand le dénominateur est proche de 0, i.e. quand $\frac{ry}{Q}$ est proche d'un entier. Ainsi, on faisant une lecture, on tombera avec une probabilité assez élevé sur un $|y, f(x)\rangle$ avec $\frac{y}{Q}r = c$ entier.

La fraction $\frac{y}{Q}$ est connue (y est donné par la mesure, et Q est connu depuis le début !).

8.3 Calcul d'un facteur

entrée : un entier N composée, qui n'est pas une puissance d'un nombre premier
sortie : une factorisation de N ou échec
fonction shor(N)
    choisir au hasard g dans $\{2, \dots, N-1\}$
    si $pgcd(g, N) \neq 1$ **alors renvoyer** $pgcd(g, N)$
    $r :=$ ordre de g dans $\mathbb{Z}/N\mathbb{Z}^\times$ (obtenu avec l'algorithme quantique ci-dessus)
    si r impair **alors** échec
    si $g^{r/2} \equiv -1 \mod N$ **alors** échec
    renvoyer $pgcd(g^{r/2}+1, N)$

On peut tester si un entier est composé en appliquant un test de primalité. Puis tester qu'il n'est pas de la forme a^b efficacement.

Théorème 41 Si N est un entier composée qui n'est pas une puissance d'un nombre premier, alors shor(N) renvoie un facteur de N avec une probabilité $\geq \frac{1}{3}$.

Démonstration. Si $pgcd(g, N) \neq 1$ alors $pgcd(g, N)$ est un facteur de N.
Sinon, g est dans le groupe multiplicatif $\mathbb{Z}/N\mathbb{Z}^\times$. On calcule l'ordre de g, que l'on note r.

Définition 42 Une racine carrée non triviale x de 1 modulo N est un entier $x \not\equiv \pm 1[N]$ avec $x^2 \equiv 1[N]$.

Lemme 43 Si x est une racine carrée non triviale de 1 modulo N, alors $pgcd(x+1, N)$ est un facteur de N non trivial.

Démonstration. Soit x est une racine carrée non triviale de 1, i.e. $x^2 \equiv 1[N]$. Cela implique que N divise $x^2 - 1 = (x-1)(x+1)$. D'autre part, comme x est une racine carrée non triviale, on a $x \not\equiv \pm 1[N]$. Ainsi N ne divise ni $x+1$ ni $x-1$. N doit alors avoir un facteur non trivial avec $(x+1)$ et $(x-1)$. En particulier $pgcd(x+1, N)$ est non trivial (et $pgcd(x-1, N)$ aussi d'ailleurs. ■

Lemme 44 Soit N composé avec au moins deux facteurs premiers différents. Soit g choisi uniformément dans $\{0, \dots, N-1\}$. Soit r l'ordre de g modulo N. Si $pgcd(g, N) = 1$, alors

$$\mathbb{P}(r \text{ soit pair, et } g^{r/2} \neq -1[N]) \geq \frac{1}{2}$$

.

Démonstration.
Pour simplifier la démonstration, supposons que $N = p_1 p_2$ où p_1 et p_2 premiers. Par le théorème des restes chinois, choisis g choisi uniformément dans Z/NZ^* c'est comme choisis g_1 dans Z/p_1Z^* et g_2 dans Z/p_2Z^* avec $g = g_i[p_i]$. Soit r_i l'ordre de g_i dans Z/p_iZ^*. Soit d_i le plus grand tel que $2_i^d | r_i$; et d le plus grand tel que $2^d | r$. On montre que pour avoir r impair ou $g^{r/2} = -1[N]$, il faut que $d_1 = d_2$. La probabilité de ça est au plus $\frac{1}{2}$.
TODO:
Si r est impair, alors comme $r_1, r_2 | r$, r_1 et r_2 sont aussi impairs. Et donc $d_1 = d_2 = 0$. L'autre cas c'est r pair et $g^{r/2} = -1[N]$. Mais alors $g^{r/2} = -1[p_i]$. TODO: pourquoi ? Donc r_i ne divise pas $r/2$. Comme $r_i | r$, on a $d_1 = d_2$. ■
■

9 Notes bibliographiques

Le dernier chapitre de [DPV06] est plaisant et a été beaucoup utilisé pour fabriquer ce cours. Le seul souci est qu'il présente l'algorithme de Shor : très intéressant mais rien est fait proprement, et pour cause, l'algorithme de Shor requiert un peu de théorie des nombres, et beaucoup de calcul avec la FFT etc. J'ai donc relayé la QFT et Shor à la fin. J'ai préféré présenter au début l'algorithme de Grover, plus simple à comprendre. L'algorithme de Grover a été proposé en 1996 [Gro96], attention la démonstration de correction dans l'article original contient quelque erreurs. L'algorithme de Grover est optimal (dans un certain sens) [BBBV97]. Pour des informations sur l'implémentation, consulter [NO10]. Le lecteur ou la lectrice intéressée pourra se référer aussi à [CEP+18].

Références

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5) :1510–1523, 1997.

[BWP+17] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671) :195–202, 2017.

[CEP+18] Patrick J. Coles, Stephan J. Eidenbenz, Scott Pakin, Adetokunbo Adedoyin, John Ambrosiano, Petr M. Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo N. Djidjev, David Gunter, Satish Karra, Nathan Lemons, Shizeng Lin, Andrey Y. Lokhov, Alexander Malyzhenkov, David Dennis Lee Mascarenas, Susan M. Mniszewski, Balu Nadiga, Dan O'Malley, Diane Oyen, Lakshman Prasad, Randy Roberts, Philip Romero, Nandakishore Santhi, Nikolai Sinitsyn, Pieter Swart, Marc Vuffray, Jim Wendelberger, Boram Yoon, Richard J. Zamora, and Wei Zhu. Quantum algorithm implementations for beginners. *CoRR*, abs/1804.03719, 2018.

[DPV06] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani. Algorithms. 2006.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.

[LWG+10] Benjamin P Lanyon, James D Whitfield, Geoff G Gillett, Michael E Goggin, Marcelo P Almeida, Ivan Kassal, Jacob D Biamonte, Masoud Mohseni, Ben J Powell, Marco Barbieri, et al. Towards quantum chemistry on a quantum computer. *Nature chemistry*, 2(2) :106–111, 2010.

[NO10] Mikio Nakahara and Tetsuo Ohmi. *Quantum computing : from linear algebra to physical realizations*. Taylor & Francis, 2010.

[PODDB+12] Alejandro Perdomo-Ortiz, Neil Dickson, Marshall Drew-Brook, Geordie Rose, and Alán Aspuru-Guzik. Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific reports*, 2(1) :1–7, 2012.

[Sho94] Peter W. Shor. Algorithms for quantum computation : Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994.

[SKPK19] Adam Smith, MS Kim, Frank Pollmann, and Johannes Knolle. Simulating quantum many-body dynamics on a current digital quantum computer. *npj Quantum Information*, 5(1) :1–13, 2019.