Algorithme de Grover

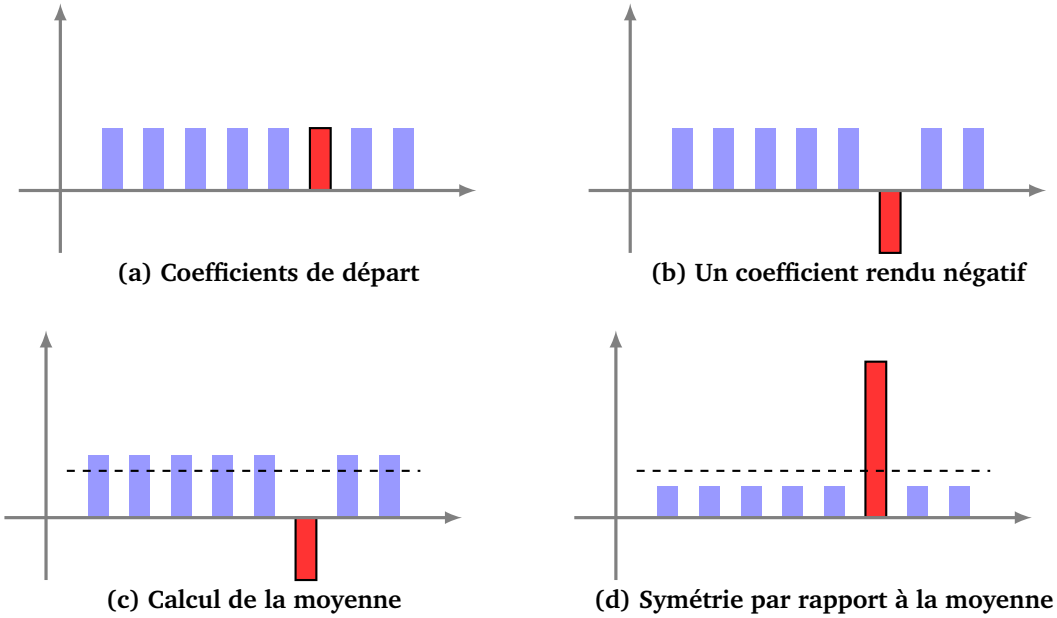
Vidéo ■ partie 11. Algorithme de Grover

L'algorithme de Grover est un algorithme de recherche d'un élément dans une liste qui est plus efficace que les algorithmes classiques. Son principe est simple, même si sa mise en œuvre est un peu complexe. L'algorithme de Grover ne fournit pas un résultat sûr à 100 *%, mais une réponse qui a de grandes chances d'être la bonne.*

1. Recherche dans une liste

1.1. Idée de l'algorithme

Expliquons l'algorithme de Grover avec des dessins.



(a) Coefficients de départ

(b) Un coefficient rendu négatif

(c) Calcul de la moyenne

(d) Symétrie par rapport à la moyenne

Figure (a). Il s'agit de distinguer un rang parmi les autres, ici le rang du rectangle rouge. On considère que les hauteurs des rectangles représentent les coefficients d'un qubit : ici il y a 8 coefficients pour l'expression d'un 3-qubit comme somme des 8 états de base. Une mesure de ce qubit ne donnerait aucune information, chacun des rangs s'obtenant avec la même probabilité, car les hauteurs des rectangles sont égales.

Figure (b). On rend le coefficient du rang qui nous intéresse négatif. (Cela peut se faire sans connaître le rang : je regarde la couleur du rectangle, s'il est rouge je change le signe). Une mesure de nouveau qubit ne donnerait toujours aucune information, car en valeur absolue les hauteurs des rectangles sont encore égales.

Figure (c). On calcule la moyenne des coefficients

Figure (d). On effectue une symétrie par rapport à la moyenne. Les rectangles bleus ont maintenant des hauteurs petites alors que le rectangle rouge a une grande hauteur. Que donne une mesure de ce nouveau qubit ? Il y a beaucoup plus de chances d'obtenir l'état de base correspondant au rectangle rouge et donc d'obtenir le rang souhaité.

L'algorithme de Grover est l'itération de ce procédé : à partir du dernier état obtenu avant mesure, on recommence les étapes (b), (c) et (d). Le rectangle rouge devient de plus en plus grand et les rectangles bleus de plus en plus petits. Ainsi après plusieurs itérations, une mesure donne avec une très forte probabilité, le rang du rectangle rouge.

1.2. Recherche dans une liste ordonnée

On dispose d'une liste et on nous donne un élément. Il s'agit de trouver s'il est présent dans la liste et de déterminer son rang. Le problème est donc : trouver i tel que liste$[i]$ = mon-élément.
Supposons d'abord que les éléments sont classés par ordre (par exemple les mots d'un dictionnaire par ordre alphabétique, ou bien les numéros de cartes d'étudiants classés du plus petit au plus grand). Alors un algorithme de recherche classique est la recherche par dichotomie (on coupe au milieu, on regarde si l'élément cherché est avant ou après et on recommence). C'est une méthode très efficace : si la liste comporte N éléments alors la complexité est $O(\ln_2(N))$. La complexité est mesurée comme le nombre de comparaisons entre l'élément au rang i et l'élément recherché. Par exemple si $N = 1024 = 2^{10}$, alors il faut environ moins de 10 comparaisons pour trouver l'élément et si la liste contient un milliard de données, il faut moins de 30 comparaisons pour conclure !

1.3. Recherche dans une liste non ordonnée

Pour certaines listes, il n'est pas possible d'ordonner des éléments ou bien on ne souhaite pas le faire car ordonner une liste est assez long. Comment chercher un élément dans une liste non ordonnée ? Il n'y a pas d'autre choix que de parcourir la liste ! On peut par exemple parcourir la liste en partant du premier élément, ou bien en choisissant les éléments au hasard. Dans les deux méthodes, la complexité dans le pire des cas est N (si l'élément cherché est le dernier à être testé). En moyenne, on trouvera l'élément cherché au bout de $\frac{N}{2}$ tests, mais cela n'améliore pas l'ordre de grandeur de la complexité qui est donc $O(N)$ (car $O(N) = O(\frac{N}{2})$).

1.4. Complexité de l'algorithme de Grover

L'algorithme de Grover qui sera étudié dans ce chapitre a une complexité d'ordre $O(\sqrt{N})$, c'est donc un gros progrès par rapport aux algorithmes classiques. Par exemple, dans une liste de $N = 1024$ personnes, il suffira d'environ 30 tests ; pour une liste d'un milliard de données, la complexité est d'environ 30 000. C'est évidemment beaucoup plus que l'algorithme de la dichotomie sur une liste ordonnée, mais beaucoup moins que la recherche séquentielle qui est de complexité N.

1.5. Algorithmes probabilistes

L'algorithme de Grover est rapide mais ne renvoie malheureusement pas toujours le bon résultat ! C'est un algorithme probabiliste. L'algorithme de Grover renvoie le bon résultat dans la plupart des cas (on verra que pour une liste de longueur N, l'algorithme se trompe avec une probabilité inférieure à $\frac{4}{N}$).
Pourquoi un algorithme probabiliste, ne donnant donc pas toujours la réponse attendue, peut quand même être un bon algorithme ? Tout d'abord, pour certains problèmes, ne pas avoir la bonne réponse n'est pas trop grave. Par exemple, si un algorithme vous fournit le plus court chemin dans 99 cas sur 100 et que vous faîtes quelques kilomètres en plus de temps en temps, cela peut vous convenir. D'autre part il est souvent

facile de vérifier si la réponse donnée est correcte, donc si la réponse obtenue ne vous convient pas, vous relancez l'algorithme. Même un algorithme qui ne donne la bonne réponse qu'une fois sur deux peut être utile ! Imaginez un algorithme qui donne tous les bons numéros du loto seulement une fois sur deux, est-ce que cela vous intéresserait ?

1.6. Application au hachage

Certaines sécurités informatiques sont basées sur des fonctions de hachage. Par exemple une fonction de hachage permet de vérifier qu'un fichier téléchargé n'a pas été compromis (*checksum*). D'autres exemples sont les *bitcoins* qui utilisent une « preuve de travail », de même que certaines méthodes de cryptographie (par exemple pour ne pas sauvegarder vos mots de passe en clair sur votre disque dur).
Considérons l'exemple d'une fonction de hachage qui à un entier k codé sur n bits (la clé) associe un entier $h(k)$ (le *hash*). J'utilise cette fonction ainsi :

- je choisis une clé secrète par exemple $k_0 = 1.0.1.0.1$ (avec ici $n = 5$),
- je calcule $h(k_0)$, par exemple $h(k_0) = 12\,575\,302$,
- $h(k_0)$ est mon mot de passe.

Imaginons un pirate qui voudrait attaquer mon compte. Il n'a pas d'autre choix que de tester toutes les clés possibles 0.0.0.0.0, puis 0.0.0.0.1,... afin d'obtenir la bonne clé, donc le bon mot de passe. Il y a en tout $N = 2^n$ (ici $n = 5$) clés possibles à tester dans le pire des cas avec l'informatique classique.
Mais trouver cette clé revient à trouver le bon élément parmi une liste de $N = 2^n$ éléments, ce que fait l'algorithme de Grover avec une complexité d'ordre $O(\sqrt{N})$, c'est-à-dire $O(2^{n/2})$, ce qui est beaucoup plus rapide que la solution classique.
À la suite de la découverte de Grover, il a été recommandé de doubler la longueur des clés de certains protocoles (par exemple passer de AES-128 à AES-256). En effet imaginons une clé de longueur n bits, un algorithme classique nécessite de l'ordre de $N = 2^n$ tests et l'algorithme quantique seulement $\sqrt{N} = 2^{n/2}$. Si la clé est doublée à une longueur $2n$, alors l'algorithme de Grover nécessite maintenant $\sqrt{2^{2n}} = 2^n$ tests et donc la niveau de sécurité initial est maintenu.

1.7. Image réciproque

Élargissons la situation précédente à un problème plus général. Soit $f : E \to F$ une fonction. Étant donné $x \in E$, il est généralement facile de calculer son image $y = f(x)$. Par contre le problème inverse de trouver un antécédent est souvent délicat : étant donné $y \in F$, trouver $x \in E$ tel que $y = f(x)$. Parfois la seule solution est de tester tous les x possibles, et là encore l'algorithme de Grover permet de le faire plus rapidement.
Voici un exemple de fonction où il est difficile de calculer un antécédent. Soit p un (grand) nombre premier et $f : \mathbb{Z} \to \mathbb{Z}$ la fonction définie par $f(x) = x^2 \pmod p$. Bien sûr, pour x donné, il est facile de calculer $y = f(x)$. Par contre y étant fixé, il est difficile de trouver un antécédent, c'est-à-dire un x tel que $x^2 \pmod p = y$. On doit alors se rabattre sur des techniques de force brute et tester $x = 0$, $x = 1$,..., $x = p-1$.

2. Principe et circuit

2.1. Problème

Nous modélisons la recherche dans une liste non ordonnée à l'aide d'une fonction mathématique. Soit N un entier fixé et soit k_0 un entier avec $0 \leqslant k_0 \leqslant N-1$. Définissons alors la fonction $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ par

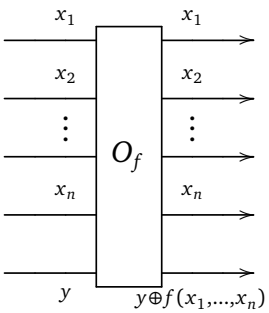
$$f(k_0) = 1 \qquad \text{et} \qquad f(k) = 0 \quad \text{pour tout } k \neq k_0.$$

Problème. Étant donnée une telle fonction f, trouver la valeur k_0 telle que $f(k_0) = 1$.
Remarques.

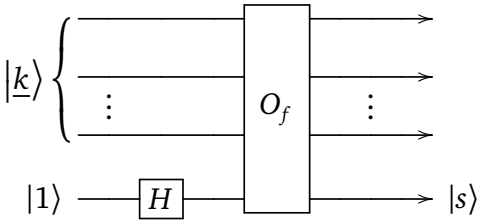
- Il s'agit donc de trouver l'antécédent de 1 par f.
- On peut identifier $\{0,1,\dots,N-1\}$ à $\mathbb{Z}/N\mathbb{Z}$ et $\{0,1\}$ à $\mathbb{Z}/2\mathbb{Z}$ et donc considérer $f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/2\mathbb{Z}$.

2.2. Oracle

On se place dans le cas où N est une puissance de 2 : $N = 2^n$. On rappelle que pour $0 \leqslant k \leqslant N-1$, alors $\underline{k}$ est l'écriture binaire de k sur n bits. Ainsi $\left|\underline{k}\right\rangle$, pour $k = 0,\dots,2^n-1$, désigne les n-qubits de la base canonique : $\left|\underline{0}\right\rangle = |0.0\dots0\rangle$, $\left|\underline{1}\right\rangle = |0.0\dots1\rangle$, …, $\left|\underline{2^n-1}\right\rangle = |1.1\dots1\rangle$.
Nous allons utiliser l'oracle O_f associé à la fonction f. Pour $x \in \mathbb{Z}/N\mathbb{Z}$ et $y \in \mathbb{Z}/2\mathbb{Z}$, l'oracle réalise une fonction $F(x,y) = (x, y \oplus f(x))$. On préfère écrire l'entier x à l'aide de son écriture binaire $\underline{x} = x_1.x_2\dots x_n$, ce qui permet de récrire la fonction F sous la forme $F(x,y) = (x_1,\dots,x_n, y \oplus f(x_1,\dots,x_n))$. On rappelle que $y \oplus y'$ est l'addition dans $\mathbb{Z}/2\mathbb{Z}$ (qui vérifie $1 \oplus 1 = 0$).



Utilisons notre oracle pour réaliser le petit circuit suivant :



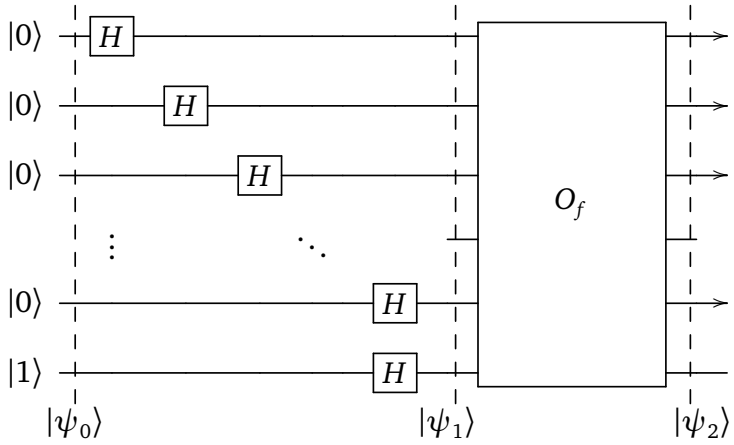
Que se passe-t-il pour notre f particulier qui vaut 1 seulement en k_0 ? L'entrée sur la dernière ligne avant l'oracle est $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ et la sortie est

$$s = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \oplus f(k) = (-1)^{f(k)}\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \text{si } k \neq k_0 \\ -\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \text{si } k = k_0 \end{cases}$$

Ainsi l'oracle permet de détecter si l'entier en entrée est k_0 ou pas. Malheureusement nous n'avons fait aucun progrès car il faut de nouveau tester tous les k de 0 à $N-1$ pour pouvoir conclure. C'est là qu'entre en jeu la magie de l'informatique quantique et la superposition des états : il est possible de tester toutes ces valeurs en même temps !

2.3. Circuit

Voici le début du circuit de l'algorithme de Grover qui permet de comprendre l'essentiel de son fonctionnement (le circuit complet sera étudié plus tard).



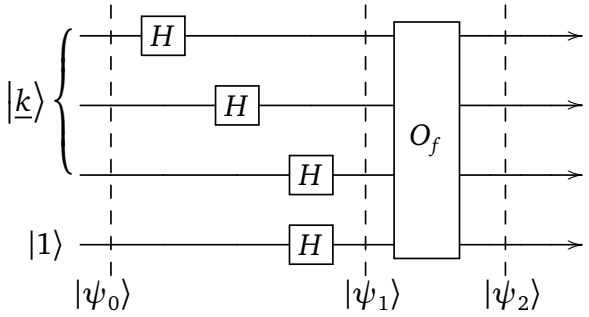
- Initialisation. Le qubit en entrée est le $(n+1)$-qubit :
$$|\psi_0\rangle = |0\dots0\rangle \cdot |1\rangle = \left|\underline{0}\right\rangle \cdot |1\rangle .$$
- Transformation de Hadamard. On s'intéresse d'abord seulement aux n premières lignes. Après la transformation de Hadamard (une porte de Hadamard sur chacune des n premières lignes) alors le n-qubit est
$$\left|\underline{0}\right\rangle + \left|\underline{1}\right\rangle + \cdots + \left|\underline{k_0}\right\rangle + \cdots + \left|\underline{2^n-1}\right\rangle$$
(à un facteur multiplicatif près). Ainsi tous les qubits $\left|\underline{k}\right\rangle$ se retrouvent simultanément en entrée de l'oracle !
Voici les calculs complets, en intégrant tous les qubits :
$$\begin{aligned} |\psi_1\rangle &= H^{\otimes n+1} |\psi_0\rangle \\ &= H^{\otimes n} \left|\underline{0}\right\rangle \cdot H |1\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \left|\underline{k}\right\rangle \cdot \tfrac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \end{aligned}$$

- Oracle. Nous avons vu que l'oracle fait apparaître un signe « $-$ » devant le terme correspond à k_0. Ainsi la sortie de l'oracle est
$$\left|\underline{0}\right\rangle + \left|\underline{1}\right\rangle + \cdots - \left|\underline{k_0}\right\rangle + \cdots + \left|\underline{2^n-1}\right\rangle$$
Noter le signe « $-$ » devant $\left|\underline{k_0}\right\rangle$ uniquement.
En détail, sachant que $(-1)^{f(k)} = 1$, sauf $(-1)^{f(k_0)} = -1$:
$$\begin{aligned} |\psi_2\rangle &= O_f |\psi_1\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{f(\underline{k})} \left|\underline{k}\right\rangle \cdot \tfrac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\ &= \frac{1}{\sqrt{2^n}} \left(\left|\underline{0}\right\rangle + \left|\underline{1}\right\rangle + \cdots - \left|\underline{k_0}\right\rangle + \cdots + \left|\underline{2^n-1}\right\rangle \right) \cdot \tfrac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \end{aligned}$$

- Bilan. En une seule évaluation de l'oracle, on arrive à distinguer le terme de rang k_0 des autres termes. L'idée essentielle est ici. Cependant on n'a pas complètement terminé : il reste à déterminer précisément ce rang, connaissant la somme. Ce sera le travail assez technique du reste de ce chapitre.

2.4. Exemple

Exemple.
Prenons $n = 3$ et $k_0 = 5$ qui caractérisent la fonction $f : \mathbb{Z}/8\mathbb{Z} \to \mathbb{Z}/2\mathbb{Z}$ définie par $f(k) = 0$ pour tout $k \neq 5$ et $f(5) = 1$.



Reprenons les calculs dans ce cas :

- Initialisation. $|\psi_0\rangle = |0.0.0\rangle \cdot |1\rangle = \left|\underline{0}\right\rangle \cdot |1\rangle$.
- Transformation de Hadamard.

$$\begin{aligned}|\psi_1\rangle &= \tfrac{1}{4}\,|(0+1).(0+1).(0+1)\rangle\,|0-1\rangle \\ &= \tfrac{1}{4}\big(\,|0.0.0\rangle + |0.0.1\rangle + |0.1.0\rangle + |0.1.1\rangle + |1.0.0\rangle + |1.0.1\rangle + |1.1.0\rangle + |1.1.1\rangle\,\big)\,|0-1\rangle \\ &= \tfrac{1}{4}\big(\,\left|\underline{0}\right\rangle + \left|\underline{1}\right\rangle + \left|\underline{2}\right\rangle + \left|\underline{3}\right\rangle + \left|\underline{4}\right\rangle + \left|\underline{5}\right\rangle + \left|\underline{6}\right\rangle + \left|\underline{7}\right\rangle\,\big)\,|0-1\rangle\end{aligned}$$

- Oracle. Comme $f(5) = 1$ alors nous avons vu que l'oracle fait apparaître un signe « $-$ » devant le terme correspond à $\left|\underline{5}\right\rangle = |1.0.1\rangle$.

$$\begin{aligned}|\psi_2\rangle &= \tfrac{1}{4}\big(\,|0.0.0\rangle + |0.0.1\rangle + |0.1.0\rangle + |0.1.1\rangle + |1.0.0\rangle - |1.0.1\rangle + |1.1.0\rangle + |1.1.1\rangle\,\big)\,|0-1\rangle \\ &= \tfrac{1}{4}\big(\,\left|\underline{0}\right\rangle + \left|\underline{1}\right\rangle + \left|\underline{2}\right\rangle + \left|\underline{3}\right\rangle + \left|\underline{4}\right\rangle - \left|\underline{5}\right\rangle + \left|\underline{6}\right\rangle + \left|\underline{7}\right\rangle\,\big)\,|0-1\rangle\end{aligned}$$

- Bilan. On trouve bien une somme des qubits de base avec un signe « $-$ » au rang $k_0 = 5$ (on commence à compter au rang 0).

3. Transformations géométriques

Le reste du chapitre est dédié à la détection du rang k_0. C'est une partie assez technique, mais on comprend mieux les calculs à l'aide d'une interprétation géométrique un peu plus sophistiquée que celle de l'introduction de ce chapitre.

3.1. Symétrie de l'oracle

Que fait l'oracle sur les n-qubits des n premières lignes ? L'oracle change $\left|\underline{k_0}\right\rangle$ en $-\left|\underline{k_0}\right\rangle$, et laisse inchangé $\left|\underline{k}\right\rangle$ pour $k \neq k_0$:

$$\begin{cases} \left|\underline{k_0}\right\rangle \overset{O_f}{\longmapsto} -\left|\underline{k_0}\right\rangle \\ \left|\underline{k}\right\rangle \overset{O_f}{\longmapsto} \left|\underline{k}\right\rangle \quad \text{si } k \neq k_0. \end{cases}$$

On va isoler le qubit de base $\left|\underline{k_0}\right\rangle$ d'un côté et regrouper tous les autres qubits de base, ainsi n'importe quel

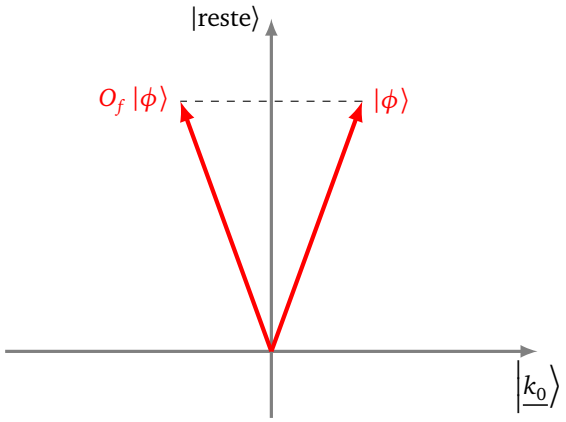
n-qubit $|\phi\rangle$ s'écrit :

$$|\phi\rangle = \alpha \left|\underline{k_0}\right\rangle + \sum_{k \neq k_0} \alpha_k \left|\underline{k}\right\rangle$$

L'action de l'oracle O_f donne :

$$O_f |\phi\rangle = -\alpha \left|\underline{k_0}\right\rangle + \sum_{k \neq k_0} \alpha_k \left|\underline{k}\right\rangle$$

Géométriquement cette transformation est une symétrie par rapport à l'axe formé des qubits de base autres que $\left|\underline{k_0}\right\rangle$ que l'on regroupe schématiquement par l'axe $|\text{reste}\rangle$ ci-dessous. Sur la figure, la transformation est représentée comme une symétrie par rapport à une droite (mais en réalité c'est une symétrie par rapport à un hyperplan de dimension $2^n - 1$).



3.2. Symétrie S_0

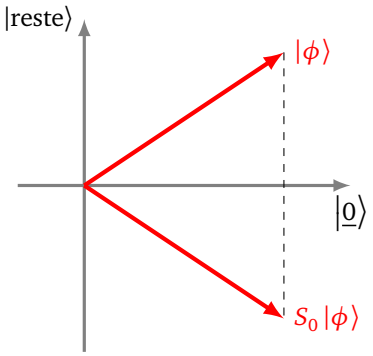
Considérons la transformation S_0 définie sur les n-qubits de la base canonique par :

$$\begin{cases} \left|\underline{0}\right\rangle \overset{S_0}{\longmapsto} \left|\underline{0}\right\rangle \\ \left|\underline{k}\right\rangle \overset{S_0}{\longmapsto} -\left|\underline{k}\right\rangle \quad \text{si } k \neq 0 \end{cases}$$

Cette fois c'est seulement le qubit de base $\left|\underline{0}\right\rangle$ qui reste inchangé.
Par exemple : pour $n = 2$, on a $S_0 |0.0\rangle = |0.0\rangle$ alors que $S_0 |0.1\rangle = -|0.1\rangle$, $S_0 |0.1\rangle = -|0.1\rangle$, $S_0 |1.1\rangle = -|1.1\rangle$. Comme d'habitude on étend S_0 par linéarité à tous les n-qubits. Ainsi :

$$S_0 (\alpha |0.0\rangle + \beta |0.1\rangle + \gamma |1.0\rangle + \delta |1.1\rangle) = \alpha |0.0\rangle - \beta |0.1\rangle - \gamma |1.0\rangle - \delta |1.1\rangle .$$

Géométriquement S_0 est une symétrie par rapport à l'axe $\left|\underline{0}\right\rangle$ (attention les axes ne sont pas les mêmes que dans la figure précédente).



Voici l'écriture algébrique de S_0.

Lemme 1.

$$\boxed{S_0 = 2\left|\underline{0}\right\rangle\left\langle\underline{0}\right| - I}$$

I désigne l'application identité. Ainsi cette formule signifie que pour un qubit $|\phi\rangle$ on a :

$$S_0|\phi\rangle = 2\left|\underline{0}\right\rangle\left\langle\underline{0}|\phi\right\rangle - |\phi\rangle$$

L'écriture $\left|\underline{0}\right\rangle\left\langle\underline{0}|\phi\right\rangle$ est bien qubit car $\left\langle\underline{0}|\phi\right\rangle$ est un scalaire (i.e. un nombre complexe).

Démonstration. Il suffit de vérifier que cette formule est vraie pour les $|\phi\rangle$ parcourant les qubits de base.
Pour $|\phi\rangle = \left|\underline{0}\right\rangle$, alors

$$\left(2\left|\underline{0}\right\rangle\left\langle\underline{0}\right| - I\right)\left|\underline{0}\right\rangle = 2\left|\underline{0}\right\rangle\left\langle\underline{0}|\underline{0}\right\rangle - \left|\underline{0}\right\rangle = 2\left|\underline{0}\right\rangle - \left|\underline{0}\right\rangle = \left|\underline{0}\right\rangle,$$

car $\left\langle\underline{0}|\underline{0}\right\rangle = 1$.
Pour $|\phi\rangle$ vérifiant $\left\langle\underline{0}|\phi\right\rangle = 0$, alors

$$\left(2\left|\underline{0}\right\rangle\left\langle\underline{0}\right| - I\right)|\phi\rangle = 2\left|\underline{0}\right\rangle\left\langle\underline{0}|\phi\right\rangle - |\phi\rangle = 2\left|\underline{0}\right\rangle \cdot 0 - |\phi\rangle = -|\phi\rangle.$$

□

Notons que la matrice de S_0 est une matrice diagonale, avec $+1$ comme premier élément et des -1 ailleurs.

$$S_0 = \begin{pmatrix} 1 & & & & \\ & -1 & & & \\ & & -1 & & \\ & & & \ddots & \\ & & & & -1 \end{pmatrix}$$

Il est clair que cette matrice est unitaire.

3.3. Transformation S_ψ

Nous allons généraliser la transformation S_0. Fixons un n-qubit $|\psi\rangle$ de norme 1. Nous définissons la transformation S_ψ sur les n-qubits par la formule

$$S_\psi = 2|\psi\rangle\langle\psi| - I$$

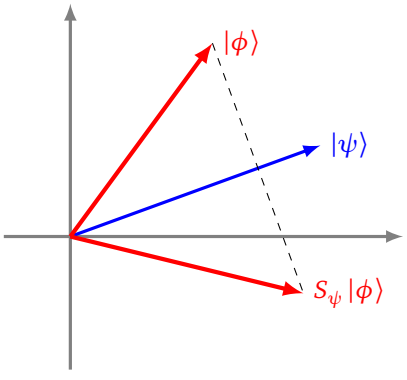
Autrement dit, pour n'importe quel n-qubit $|\phi\rangle$, on a :

$$S_\psi|\phi\rangle = 2|\psi\rangle\langle\psi|\phi\rangle - |\phi\rangle.$$

Cette transformation vérifie :

$$\begin{cases} |\psi\rangle \overset{S_0}{\longmapsto} |\psi\rangle \\ |\phi\rangle \overset{S_0}{\longmapsto} -|\phi\rangle \quad \text{si } |\phi\rangle \text{ est orthogonal à } |\psi\rangle. \end{cases}$$

Rappelons que « $|\phi\rangle$ est orthogonal à $|\psi\rangle$ » signifie « $\langle\psi|\phi\rangle = 0$ ».
Géométriquement S_ψ est une symétrie par rapport à l'axe dirigé par $|\psi\rangle$.



3.4. Transformation S_{ψ_H}

Notons $|\psi_H\rangle$ le n-qubit formé par la somme de tous les qubits de la base canonique (normalisé de façon à avoir une norme 1) :

$$|\psi_H\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \left|\underline{k}\right\rangle.$$

Ce qubit $|\psi_H\rangle$ est aussi l'image du qubit $|0\dots0\rangle$ par la transformation de Hadamard :

$$|\psi_H\rangle = H^{\otimes n} \left|\underline{0}\right\rangle.$$

Proposition 1.
La transformation S_{ψ_H} est définie par l'une des caractérisations équivalentes suivantes :

(i) $S_{\psi_H} = 2|\psi_H\rangle\langle\psi_H| - I$, *c'est-à-dire* $S_{\psi_H}|\phi\rangle = 2|\psi_H\rangle\langle\psi_H|\phi\rangle - |\phi\rangle$, *pour tout qubit* $|\phi\rangle$.

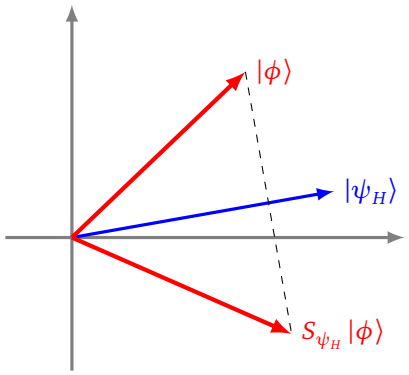
(ii) $\begin{cases} |\psi_H\rangle \overset{S_{\psi_H}}{\longmapsto} |\psi_H\rangle \\ |\phi\rangle \overset{S_{\psi_H}}{\longmapsto} -|\phi\rangle \quad \text{si } |\phi\rangle \text{ est orthogonal à } |\psi_H\rangle \end{cases}$

(iii) $S_{\psi_H} = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n}$

(iv) S_{ψ_H} *a pour matrice*

$$\frac{2}{2^n}U - I \qquad \text{où} \qquad U = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \ddots & & 1 \\ \vdots & & \ddots & \vdots \\ 1 & \cdots & 1 & 1 \end{pmatrix} \in M_{2^n}$$

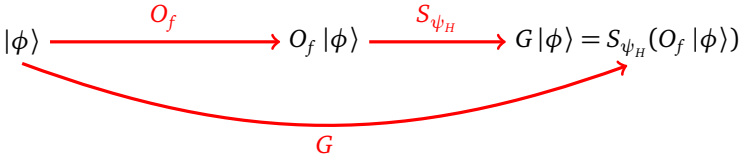
Nous prouverons cette proposition un peu plus loin. On retient que S_{ψ_H} est une symétrie par rapport à l'axe dirigé par $|\psi_H\rangle$.



3.5. Transformation de Grover

La ***transformation de Grover*** est l'application

$$G = S_{\psi_H} \circ O_f.$$



Nous allons voir quelle est l'action géométrique de G sur les qubits. Reprenons le qubit $|\psi_H\rangle$ obtenu comme la somme de tous les qubits de base :

$$|\psi_H\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \left|\underline{k}\right\rangle.$$

Dans cette somme nous mettons à part le qubit correspondant à l'indice k_0, qui est le rang que l'on doit déterminer :

$$|\psi_H\rangle = \sqrt{\frac{N-1}{N}}\,|\chi\rangle + \frac{1}{\sqrt{N}}\left|\underline{k_0}\right\rangle$$

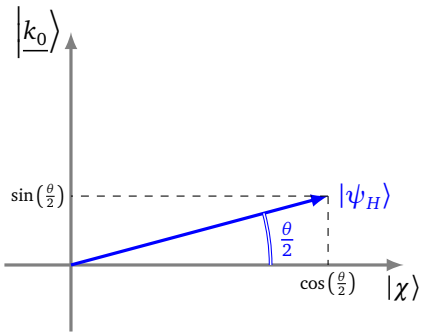
où

$$N = 2^n \qquad \text{et} \qquad |\chi\rangle = \frac{1}{\sqrt{N-1}}\sum_{k\neq k_0}\left|\underline{k}\right\rangle.$$

Nous récrivons maintenant ψ_H à l'aide d'une écriture trigonométrique :

$$|\psi_H\rangle = \cos\left(\tfrac{\theta}{2}\right)|\chi\rangle + \sin\left(\tfrac{\theta}{2}\right)\left|\underline{k_0}\right\rangle$$

où $\frac{\theta}{2}$ est l'angle entre $|\chi\rangle$ et $|\psi_H\rangle$.

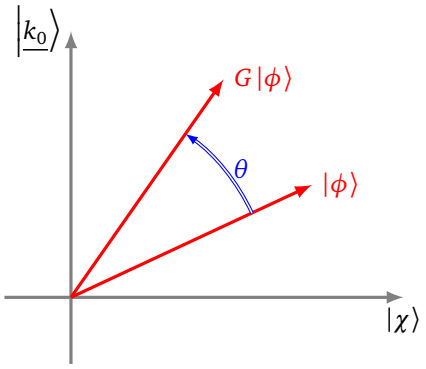


Comme dans la pratique $N = 2^n$ est grand, alors l'angle $\frac{\theta}{2}$ est petit. Pour plus de lisibilité le dessin ne reflète pas à quel point $\frac{\theta}{2}$ est petit. Ainsi $\frac{\theta}{2}$ est l'angle défini par :

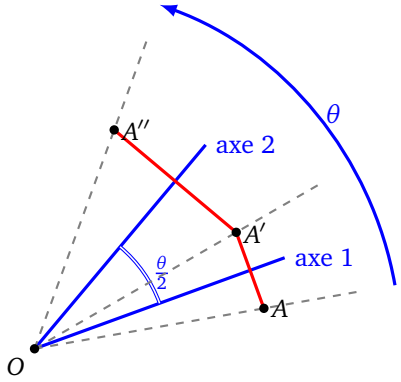
$$\cos\left(\tfrac{\theta}{2}\right) = \sqrt{\frac{N-1}{N}} \qquad \text{et} \qquad \sin\left(\tfrac{\theta}{2}\right) = \frac{1}{\sqrt{N}}.$$

Proposition 2.
La transformation de Grover est une rotation d'angle θ (centrée à l'origine).

Autrement dit, pour tout qubit $|\phi\rangle$, $G\,|\phi\rangle$ est obtenu à partir de $|\phi\rangle$, par une rotation d'angle θ (encore une fois l'angle θ est en réalité beaucoup plus petit que sur le dessin ci-dessous).



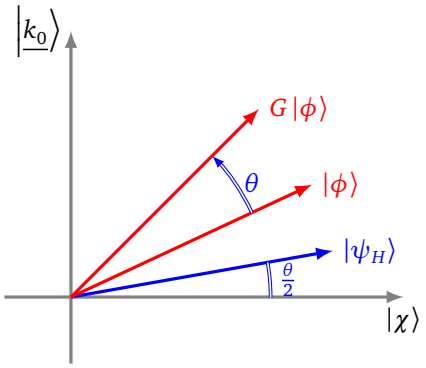
Démonstration. Un résultat géométrique dit que la composition de deux symétries axiales est une rotation, l'angle θ de cette rotation étant le double de l'angle entre les axes.



Ici G est la composition de deux symétries :

- la symétrie O_f d'axe $|\chi\rangle$,
- la symétrie S_{ψ_H} d'axe $|\psi_H\rangle$,
- l'angle entre $|\chi\rangle$ et $|\psi_H\rangle$ est $\frac{\theta}{2}$.

Ainsi $G = S_{\psi_H} \circ O_f$ est la rotation d'angle θ (centrée à l'origine).



□

3.6. Idée de l'algorithme

Le but de l'algorithme de Grover est de déterminer le rang k_0. Ce rang est repérable après l'application de l'oracle O_f. En effet partons de

$$|\psi_H\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \left|\underline{k}\right\rangle = \sqrt{\frac{N-1}{N}}\,|\chi\rangle + \frac{1}{\sqrt{N}}\left|\underline{k_0}\right\rangle,$$

alors

$$O_f\,|\psi_H\rangle = \sqrt{\frac{N-1}{N}}\,|\chi\rangle - \frac{1}{\sqrt{N}}\left|\underline{k_0}\right\rangle.$$

Mais attention ceci est une écriture quantique qui n'est pas mesurable. Souvenez-vous que nous n'avons pas accès aux coefficients d'un qubit.

Voici la seule certitude qu'une mesure puisse nous donner : si je sais par avance qu'un qubit $|\phi\rangle$ est un des qubits de base $|0\ldots0.0\rangle$, $|0\ldots0.1\rangle$,..., $|1\ldots1.1\rangle$, alors la mesure de ce n-qubit permet d'identifier ce qubit de base $|\phi\rangle$.
Prenons l'exemple des 1-qubits : si mon qubit $|\phi\rangle$ est $|0\rangle$ ou $|1\rangle$, alors une mesure permet d'identifier si on avait $|\phi\rangle = |0\rangle$ ou bien $|\phi\rangle = |1\rangle$. Noter que ceci ne fonctionnerait pas pour un état superposé $|\phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. De même avec un 2-qubit $|\phi\rangle$ parmi $|0.0\rangle$, $|0.1\rangle$, $|1.0\rangle$, $|1.1\rangle$ par une double mesure.
La transformation de Grover va nous permettre de transformer le qubit $|\psi_H\rangle$ (obtenu par transformation de Hadamard de $|0\ldots0.0\rangle$) en un qubit très proche du qubit de base $\left|\underline{k_0}\right\rangle$. Il ne reste plus qu'a effectuer une mesure pour obtenir (presque à coup sûr) la valeur de k_0.

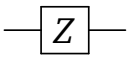
Idée de l'algorithme de Grover.

- La transformation de Hadamard envoie l'état initial $|0.0\ldots0\rangle$ sur $|\psi_H\rangle$.
- On part du qubit $|\psi_H\rangle$ qui est la superposition de tous les qubits de base.
- Ce qubit forme un angle $\frac{\theta}{2}$ avec l'axe $|\chi\rangle$. (L'angle $\frac{\theta}{2}$ est petit car $N = 2^n$ est grand.)
- La transformation de Grover est une rotation d'angle θ et conduit donc au qubit $G|\psi_H\rangle$ qui forme un angle $\frac{\theta}{2} + \theta$ avec l'axe $|\chi\rangle$.
- On itère la transformation de Grover jusqu'à obtenir un qubit $G^\ell|\psi_H\rangle$ qui forme un angle d'environ $\frac{\pi}{2}$ avec l'axe $|\chi\rangle$. (Ce nombre d'itérations ℓ est environ $\frac{\pi}{2\theta}$.)
- Le qubit $G^\ell|\psi_H\rangle$ obtenu est proche de $\left|\underline{k_0}\right\rangle$.
- La mesure de ce qubit conduit très probablement à $\underline{k_0}$ (avec une probabilité d'erreur très petite, d'ordre $\frac{4}{N}$).

3.7. Portes quantiques

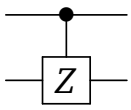
La transformation de Grover est la composition de l'oracle O_f et de la transformation S_{ψ_H}. La transformation de l'oracle est réalisable par un circuit quantique (voir le chapitre « Portes quantiques »). Nous montrons ici comment réaliser le circuit pour la transformation S_{ψ_H}, en nous limitant au cas des 2-qubits.

Porte Z. Tout d'abord rappelons l'action de la porte Z et sa matrice :



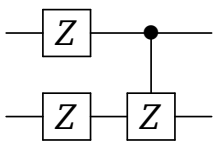
$$\begin{cases} |0\rangle \mapsto |0\rangle \\ |1\rangle \mapsto -|1\rangle \end{cases} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Porte CZ. La porte CZ (*Controlled Z*) fonctionne sur le même principe qu'une porte $CNOT$: si l'entrée de la première ligne est $|0\rangle$, alors la seconde ligne est inchangée, par contre si l'entrée de la première ligne est $|1\rangle$, alors on fait agir une porte Z sur la seconde ligne.



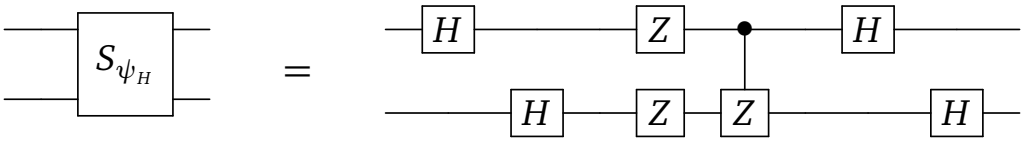
$$\begin{cases} |0.0\rangle \mapsto |0.0\rangle \\ |0.1\rangle \mapsto |0.1\rangle \\ |1.0\rangle \mapsto |1.0\rangle \\ |1.1\rangle \mapsto -|1.1\rangle \end{cases} \qquad CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Circuit pour S_0.
Voici un circuit qui permet de réaliser la transformation S_0, dans le cas des 2-qubits. Noter bien que la partie droite du circuit est un porte CZ.



$$\begin{cases} |0.0\rangle \mapsto |0.0\rangle \\ |0.1\rangle \mapsto -|0.1\rangle \\ |1.0\rangle \mapsto -|1.0\rangle \\ |1.1\rangle \mapsto -|1.1\rangle \end{cases} \qquad S_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Circuit pour S_{ψ_H}.
On sait par la proposition 1 que $S_{\psi_H} = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n}$, il suffit juste d'appliquer la transformation de Hadamard avant et après le circuit précédent.



3.8. Preuve autour de la transformation S_{ψ_H}

Cette section peut être passée lors d'une première lecture. Il s'agit de prouver la proposition 1 énoncée auparavant et que l'on rappelle ci-dessous.

Proposition 3.
La transformation S_{ψ_H} est définie par l'une des caractérisations équivalentes suivantes :

(i) $S_{\psi_H} = 2|\psi_H\rangle\langle\psi_H| - I$, *c'est-à-dire* $S_{\psi_H}|\phi\rangle = 2|\psi_H\rangle\langle\psi_H|\phi\rangle - |\phi\rangle$, *pour tout qubit* $|\phi\rangle$.

(ii) $\begin{cases} |\psi_H\rangle \xmapsto{S_{\psi_H}} |\psi_H\rangle \\ |\phi\rangle \xmapsto{S_{\psi_H}} -|\phi\rangle \quad \text{si } |\phi\rangle \text{ est orthogonal à } |\psi_H\rangle \end{cases}$

(iii) $S_{\psi_H} = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n}$

(iv) S_{ψ_H} *a pour matrice*

$$\frac{2}{2^n}U - I \qquad \text{où} \qquad U = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \ddots & & 1 \\ \vdots & & \ddots & \vdots \\ 1 & \cdots & 1 & 1 \end{pmatrix} \in M_{2^n}$$

Démonstration. Définissons S_{ψ_H} par $S_{\psi_H} = 2|\psi_H\rangle\langle\psi_H| - I$. On rappelle que $|\psi_H\rangle$ est de norme 1.

- Preuve que (i) $\iff$ (ii).
  On a $S_{\psi_H}|\psi_H\rangle = 2|\psi_H\rangle\langle\psi_H|\psi_H\rangle - |\psi_H\rangle = 2|\psi_H\rangle - |\psi_H\rangle = |\psi_H\rangle$ et $S_{\psi_H}|\phi\rangle = 2|\psi_H\rangle\langle\psi_H|\phi\rangle - |\phi\rangle = 2|\psi_H\rangle \cdot 0 - |\phi\rangle = -|\phi\rangle$ pour tout $|\phi\rangle$ orthogonal à $|\psi_H\rangle$, c'est-à-dire $\langle\psi_H|\phi\rangle = 0$.
  Réciproquement si on complète le vecteur $|\psi_H\rangle$ en une base orthogonale, alors la relation (ii) définit une unique application linéaire, qui est donc S_{ψ_H}.
- Preuve que (ii) $\iff$ (iii).
  Notons $T = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n}$. On va montrer que $T = S_{\psi_H}$ en vérifiant que T et S_{ψ_H} vérifient les mêmes relations que celles vues en (ii).
  On sait d'une part que $|\psi_H\rangle = H^{\otimes n}|\underline{0}\rangle$, mais $H^{\otimes n}$ est unitaire alors $(H^{\otimes n})^{-1} = (H^{\otimes n})^* = H^{\otimes n}$, d'où $|\underline{0}\rangle = H^{\otimes n}|\psi_H\rangle$. Ainsi :

$$\begin{aligned} T|\psi_H\rangle &= H^{\otimes n} \cdot S_0 \cdot H^{\otimes n}|\psi_H\rangle \\ &= H^{\otimes n} \cdot S_0|\underline{0}\rangle \\ &= H^{\otimes n}|\underline{0}\rangle \qquad \text{car } S_0|\underline{0}\rangle = |\underline{0}\rangle \\ &= |\psi_H\rangle \end{aligned}$$

  Considérons maintenant un qubit $|\phi\rangle$ orthogonal à $|\psi_H\rangle$, alors $\langle\psi_H|\phi\rangle = 0$ et comme $H^{\otimes n}$ est unitaire alors il préserve le produit scalaire, donc on a aussi $\langle H^{\otimes n}|\psi_H\rangle \,|\, H^{\otimes n}|\phi\rangle\rangle = 0$, donc $|\underline{0}\rangle$ et $H^{\otimes n}|\phi\rangle$ sont orthogonaux. Ainsi

$$\begin{aligned} T|\phi\rangle &= H^{\otimes n} \cdot S_0\left(H^{\otimes n}|\phi\rangle\right) \\ &= H^{\otimes n}\left(-H^{\otimes n}|\phi\rangle\right) \qquad \text{car } |\underline{0}\rangle \text{ et } H^{\otimes n}|\phi\rangle \text{ sont orthogonaux, donc } S_0\left(H^{\otimes n}|\phi\rangle\right) = -H^{\otimes n}|\phi\rangle \\ &= -H^{\otimes n} \cdot H^{\otimes n}|\phi\rangle \\ &= -|\phi\rangle \qquad \text{car } H^{\otimes n} \cdot H^{\otimes n} = \text{id parce que } H^{\otimes n} \text{ est unitaire} \end{aligned}$$

  Conclusion : T et S_{ψ_H} vérifient la même relation vue en (ii). Les applications sont donc égales : $T = S_{\psi_H}$.
- Preuve que (i) $\iff$ (iv). Par construction l'état $|\psi_H\rangle$ est la superposition de tous les états de la base canonique, donc il s'écrit sous forme de vecteur $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ à un facteur de normalisation près. Plus

précisément :

$$|\psi_H\rangle = \frac{1}{\sqrt{2^n}}\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix} \qquad \text{donc} \qquad \langle\psi_H| = |\psi_H\rangle^* = \frac{1}{\sqrt{2^n}}\begin{pmatrix}1 & 1 & \cdots & 1\end{pmatrix}$$

Ainsi :

$$|\psi_H\rangle\langle\psi_H| = \frac{1}{\sqrt{2^n}}\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix} \times \frac{1}{\sqrt{2^n}}\begin{pmatrix}1 & 1 & \cdots & 1\end{pmatrix} = \frac{1}{2^n}\begin{pmatrix}1 & 1 & \cdots & 1\\1 & \ddots & & 1\\\vdots & & \ddots & \vdots\\1 & \cdots & 1 & 1\end{pmatrix} = \frac{1}{2^n}U$$

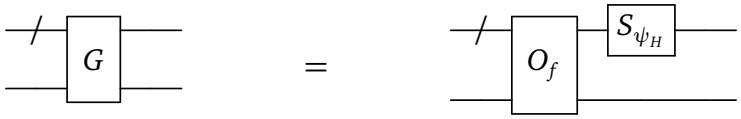
Conclusion : $S_{\psi_H} = 2|\psi_H\rangle\langle\psi_H| - I$ a pour matrice $\frac{2}{2^n}U - I$. (Réciproquement une matrice définit une unique application linéaire.)

□

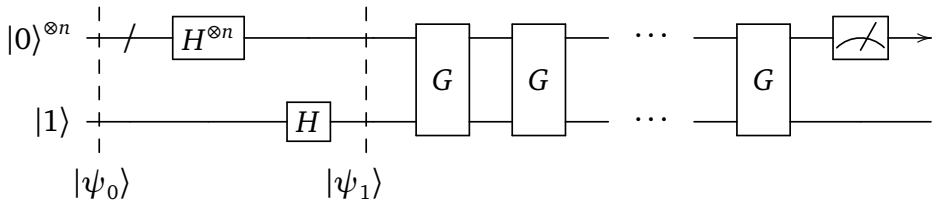
4. Étapes de l'algorithme de Grover

4.1. Circuit

On note G la transformation de Grover, elle prend en entrée un $(n+1)$-qubit, et est formée par la porte O_f de l'oracle, suivie d'une porte associée à la transformation S_{ψ_H}. On représente cette porte G avec 2 lignes seulement, la première ligne correspond à un n-qubit (ligne symbolisée avec « / »), la seconde à un 1-qubit.



Voici le circuit de l'algorithme de Grover.



La porte G est itérée ℓ fois avec $\ell \simeq \frac{\pi}{4}\sqrt{N}$ où $N = 2^n$. La complexité de l'algorithme est d'ordre ℓ, donc d'ordre $O(\sqrt{N})$. La mesure finale est la mesure d'un n-qubit (et correspond donc à n mesures de 1-qubits). Le circuit renvoie donc un n-bit classique $\underline{k}$ avec $0 \leqslant k < 2^n$. Nous allons justifier que cet entier est très probablement le rang k_0 cherché.

4.2. Données

Soit $n \geqslant 1$ et $N = 2^n$. Supposons donné un entier k_0 vérifiant $0 \leqslant k_0 < N$. On considère la fonction $f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/2\mathbb{Z}$, avec $f(k_0) = 1$ et $f(k) = 0$ pour tout $k \neq k_0$.

4.3. Initialisation et transformation de Hadamard

Le circuit quantique est initialisé par le $(n+1)$-qubit

$$|\psi_0\rangle = |0\ldots0\rangle \cdot |1\rangle = \left|\underline{0}\right\rangle \cdot |1\rangle\,.$$

Ensuite on applique la transformation de Hadamard pour obtenir le qubit

$$\begin{aligned}|\psi_1\rangle &= H^{\otimes n+1}|\psi_0\rangle \\ &= H^{\otimes n}\left|\underline{0}\right\rangle \cdot H|1\rangle \\ &= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}\left|\underline{k}\right\rangle \cdot \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= |\psi_H\rangle \cdot \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\end{aligned}$$

Dans la suite on oublie le dernier qubit et on s'intéresse seulement au n-qubit formé par les n premières lignes.

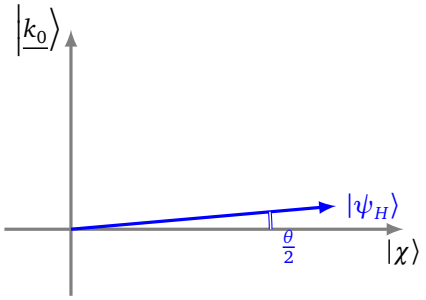
Dans $|\psi_H\rangle$ distinguons le qubit de base $\left|\underline{k_0}\right\rangle$:

$$|\psi_H\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}\left|\underline{k}\right\rangle = \sqrt{\frac{N-1}{N}}|\chi\rangle + \frac{1}{\sqrt{N}}\left|\underline{k_0}\right\rangle$$

que l'on récrit sous forme trigonométrique :

$$|\psi_H\rangle = \cos\left(\tfrac{\theta}{2}\right)|\chi\rangle + \sin\left(\tfrac{\theta}{2}\right)\left|\underline{k_0}\right\rangle$$

où $\frac{\theta}{2}$ est l'angle entre $|\chi\rangle$ et $|\psi_H\rangle$, également défini par la relation $\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}}$.



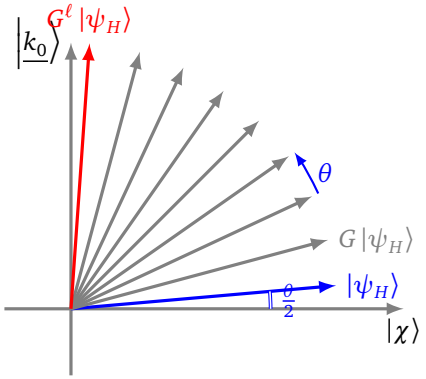
4.4. Itérations de la transformation de Grover

La transformation de Grover $G = S_{\psi_H} \circ O_f$, est une rotation d'angle θ. Donc après ℓ itérations on obtient le n-qubit

$$G^\ell|\psi_H\rangle = \cos(\theta_\ell)|\chi\rangle + \sin(\theta_\ell)\left|\underline{k_0}\right\rangle$$

avec $\theta_\ell = \frac{\theta}{2} + \ell\theta$.

On veut $\theta_\ell \simeq \frac{\pi}{2}$, c'est-à-dire $\frac{\theta}{2} + \ell\theta \simeq \frac{\pi}{2}$. Ainsi ℓ est défini comme l'entier le plus proche de $\frac{\pi}{2\theta} - \frac{1}{2}$.



Donnons une approximation du nombre ℓ d'itérations nécessaires. Pour cela nous considérons que $N = 2^n$ est grand, et donc θ est petit. Comme $\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}}$ alors $\frac{\theta}{2} \simeq \frac{1}{\sqrt{N}}$ (car pour x proche de 0, $\sin(x) \simeq x$). On veut $\ell\theta \simeq \frac{\pi}{2}$ donc $\ell \simeq \frac{\pi}{2\theta}$ et ainsi

$$\boxed{\ell \simeq \frac{\pi}{4}\sqrt{N}}$$

4.5. Mesure

Après ces ℓ itérations nous avons $\theta_\ell \simeq \frac{\pi}{2}$, donc

$$G^\ell |\psi_H\rangle = \cos(\theta_\ell)|\chi\rangle + \sin(\theta_\ell)\left|\underline{k_0}\right\rangle \simeq \left|\underline{k_0}\right\rangle.$$

La mesure de ce n-qubit conduit donc probablement au n-bit $\underline{k_0}$ et permet alors d'identifier le rang k_0. Les détails des probabilités sont donnés ci-dessous.

5. Probabilités

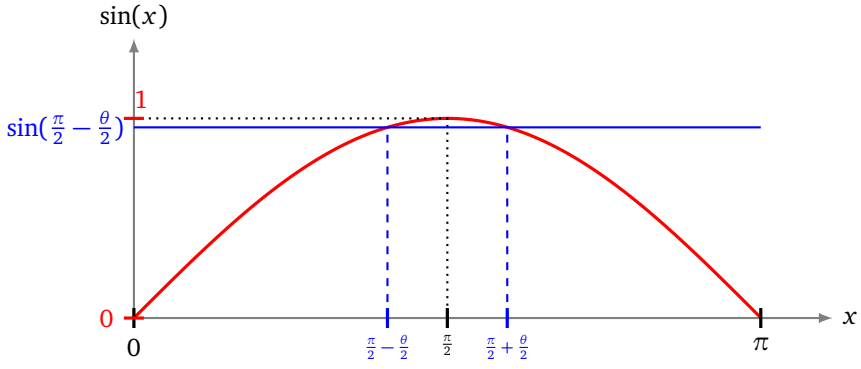
Nous avons construit un qubit $G^\ell |\psi_H\rangle$ qui est proche de $\left|\underline{k_0}\right\rangle$. Ce qubit a donc de grandes chances d'être mesuré en $\underline{k_0}$ et donc on retrouve le rang cherché k_0, mais ce n'est pas une certitude. Avec quelle probabilité obtient-on le résultat correct ? Nous allons calculer cette probabilité d'obtenir le bon résultat.

Proposition 4.
L'algorithme de Grover renvoie le rang correct k_0 avec une probabilité supérieure à $1 - \frac{4}{N}$.

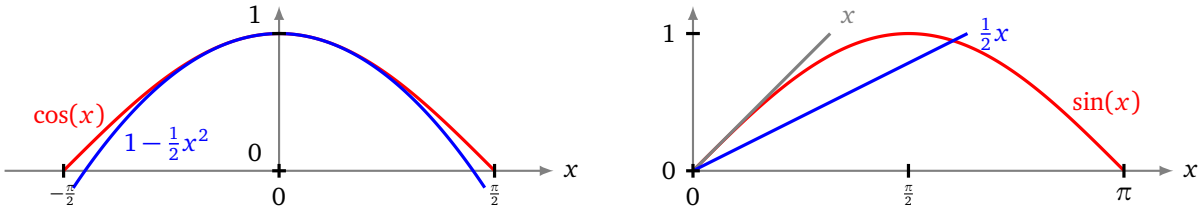
Ainsi la probabilité que l'algorithme renvoie un mauvais résultat est inférieure à $\frac{4}{N}$. Prenons par exemple $n = 10$, alors $N = \frac{1}{2^n} = 1024$ et l'algorithme fournit le résultat correct dans plus de $99,6\%$ des cas.

Démonstration.

- Le qubit final obtenu par l'algorithme de Grover est
$$G^\ell |\psi_H\rangle = \cos(\theta_\ell)|\chi\rangle + \sin(\theta_\ell)\left|\underline{k_0}\right\rangle.$$
Donc, lors de la mesure, la probabilité d'obtenir la bonne réponse $\underline{k_0}$ est $p = |\sin(\theta_\ell)|^2$.
- Nous savons que la transformation de Grover G est une rotation d'angle θ et nous avons itéré cette transformation ℓ fois de façon à construire un angle $\theta_\ell = \frac{\theta}{2} + \ell\theta$ le plus proche possible de l'angle $\frac{\pi}{2}$. Ainsi l'angle θ_ℓ est dans un intervalle d'amplitude θ centré en $\frac{\pi}{2}$:
$$\frac{\pi}{2} - \frac{\theta}{2} < \theta_\ell \leqslant \frac{\pi}{2} + \frac{\theta}{2}.$$
- Ainsi $\sin(\theta_\ell) \geqslant \sin\left(\frac{\pi}{2} - \frac{\theta}{2}\right)$ (voir la figure ci-dessous).



Donc :
$$\sin(\theta_\ell) \geqslant \sin\left(\tfrac{\pi}{2} - \tfrac{\theta}{2}\right) = \cos\left(\tfrac{\theta}{2}\right) \geqslant 1 - \frac{1}{2}\left(\frac{\theta}{2}\right)^2.$$
Pour la dernière inégalité on connaît le développement limité $\cos(x) \simeq 1 - \frac{x^2}{2}$ (pour x proche de 0), mais on a en plus l'inégalité $\cos(x) \geqslant 1 - \frac{x^2}{2}$ (figure de gauche ci-dessous).



- L'angle θ est défini avec la relation $\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}}$. On sait que, pour x proche de 0, on a $\sin(x) \simeq x$, mais on a en plus l'inégalité $\sin(x) \geqslant \frac{x}{2}$ (figure de droite ci-dessus). Ainsi, comme $\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}}$, alors $\frac{1}{\sqrt{N}} \geqslant \frac{\theta}{4}$ donc $\frac{4}{N} \geqslant \left(\frac{\theta}{2}\right)^2$ et alors en reprenant les inégalités ci-dessus :

$$\sin(\theta_\ell) \geqslant 1 - \frac{1}{2}\left(\frac{\theta}{2}\right)^2 \geqslant 1 - \frac{2}{N}.$$

Enfin on a $(1-x)^2 = 1 - 2x + x^2 \geqslant 1 - 2x$ quel que soit x, donc

$$p = |\sin(\theta_\ell)|^2 \geqslant \left(1 - \frac{2}{N}\right)^2 \geqslant 1 - \frac{4}{N}.$$

□