Hugo Cordon

Robin Wagner

Algorithme de Shor

Sommaire :

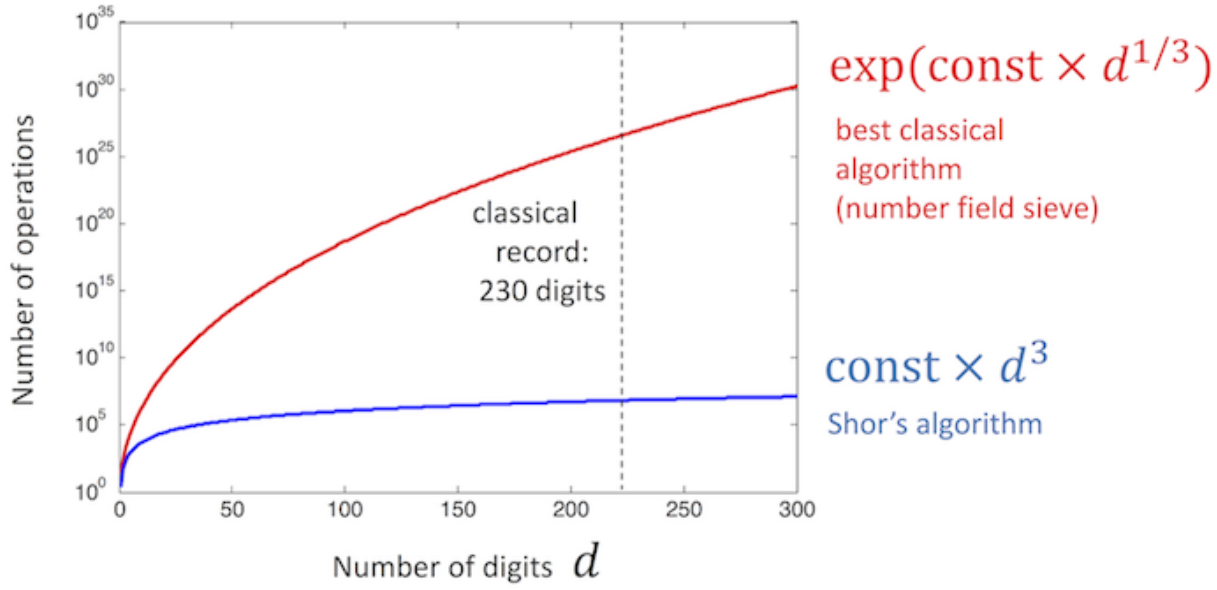
Introduction :

L'algorithme de Shor (implémenté par Peter Shor en 1994) est un algorithme quantique. Il permet de factoriser un nombre, c'est-à-dire qu'à partir de deux entiers p et q tel que p * q = N, on peut retrouver ces deux entiers en connaissant N.

A savoir : Factoriser N est équivalent à retrouver un nombre de (1%N)^2

La complexité sur un ordinateur quantique d'un tel algorithme est de O((log N)^2*(log log N) *(log log log N))

La complexité sur un ordinateur classique d'un tel algorithme est de O(e^(1.9(log N)^1/3*(log log N)^2/3))



On peut donc remarquer qu'un algorithme de Shor compilé sur un ordinateur quantique est bien plus rapide qu'un algorithme de Shor compilé sur un ordinateur classique.

Histoire :

En 2001, IBM arrivait à factoriser 15 en utilisant 7 qubits

En 2012, on arrivait à factoriser 15 avec qubits d'états solides, mais aussi à factoriser 21

EN 2018, on était capable de factoriser 15, 143, 59989 et 376289 tout cela en utilisant respectivement 4, 12, 59 et 94 qubits logiques.

A savoir :

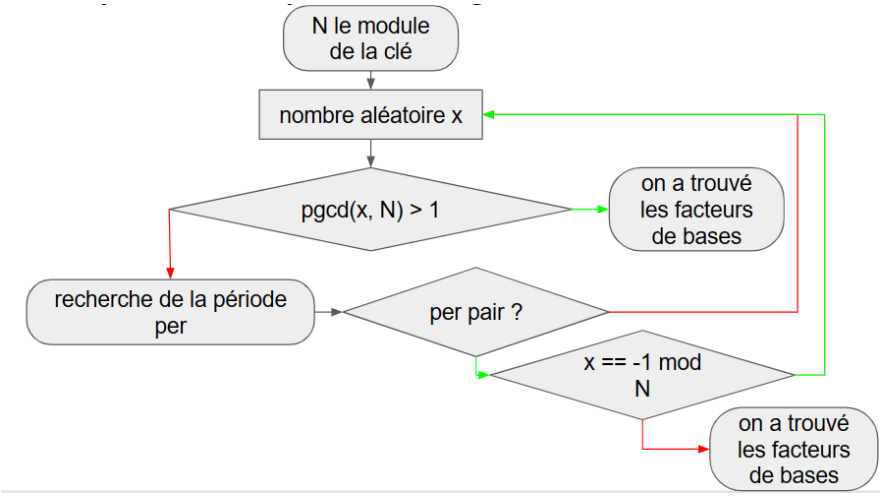
Deux problèmes venant de la théorie du quantum :

- ⇨ Mesure : l'état d'un q-qubit de registre quantique est décrite par un vecteur de dimension 2^q. Il devient exponentiellement large par rapport à la dimension du vecteur requis pour décrire q bits classique. Cependant, sur un ordinateur classique, nous pouvons juste lire l'état des différents bits tandis que dans un ordinateur quantique, nous n'avons pas d'accès direct ni illimité à un état quantique. Les informations du quantique sont obtenus en appliquant une fourchette de différentes mesures, après quoi l'état quantique d'origine n'est plus disponible, car il s'effondre

en une combinaison linéaire des seuls états de base qui sont cohérents avec le résultat de la mesure.

⇨ Pas de théorème pour cloner : depuis que les mesures détruisent l'état des quantums, il faut nécessairement créer une copie de l'état quantique. Cela nous permet de créer une fourchette de mesures dessus, pour laisser l'état original intouchable. Malheureusement, le clonage est impossible car les fourchettes de mesures sont essentiellement des matrices

Structure de l'algorithme :



L'algorithme de Shor consiste en deux partie :

Une partie algorithmique simple : le but est de résoudre le problème de factorisation grâce à la résolution de problème de order-finding.

Une partie quantique : pour résoudre le problème de order-finding.

L'algorithme de Shor repose sur un résultat de la théorie des nombres. Ce résultat est :

La fonction F(a) = x^a mod n est une fonction périodique, où x est un entier premier avec n. Dans le contexte de l'algorithme de Shor, n sera le nombre que nous souhaitons factoriser. Lorsque deux nombres sont premiers entre eux, cela signifie que leur plus grand diviseur commun est 1.

La raison pour laquelle cette fonction est utile pour factoriser de grands nombres est la suivante :

Depuis que F(a) est une fonction périodique, elle a une période r. Nous savons que
x^0 mod n = 1 , donc x^r mod n = 1, et x^2*r mod n = 1, et ainsi de suite puisque la fonction est périodique.

$$x^r \equiv 1 \mod n$$

$$(x^{r/2})^2 \equiv x^r \equiv 1 \mod n$$

$$(x^{r/2})^2 - 1 \equiv 0 \mod n$$

Et si n est pair, on a donc :

$$(x^{r/2} - 1)(x^{r/2} + 1) \equiv 0 \mod n$$

Cela signifie que le dernier produit est un multiple entier de n, le nombre à factoriser. Tant que x^(r/2) n'est pas égal à 1 alors au moins un parmi les deux facteurs ci-dessus doit avoir un facteur en commun avec n. Donc en calculant gcd(x^(r/2)-1) et gcd(x^(r/2)+1) , nous obtiendrons un facteur de n, où gcd est la fonction du plus grand dénominateur commun.

Exemple :

Voyons tout cela avec un exemple :

Soit N = 21 le nombre que nous voulons factoriser. L'essentiel est de résoudre cette équation :

$$x^2 \equiv 1 \ (mod\ 21)$$

Nous trouvons x = 1 et x = -1 (par le carré)

Nous pouvons trouver aussi :

$$x = 8$$
$$x^2 = 64 \ (mod\ 21) \equiv 1 \ (mod\ 21)$$

21 divise (8+1)(8-1), mais ne divise ni l'un ni l'autre. Or 21 = 3 x 7 avec 3 qui divise (8+1) et 7 qui divise (8-1).
On peut donc récupérer les facteurs premiers en calculant gcd(21, 8 + 1) = 3 et gcd(21, 8 – 1) = 7

Qu'est-ce qu'un bit quantique ?

Dans un ordinateur classique, l'information est stockée dans un ensemble de cases mémoires, les bits, dont la valeur est soit 0, soit 1. Un bit quantique (qubit) a, quant à lui, deux états quantiques |0> et |1>, séparés par une différence d'énergie définissant sa fréquence ,et peut être à la fois dans ces deux états. Au cours d'un algorithme, le registre de qubits se trouve dans une superposition quantique de tous ses états possibles (|00...0>, |10...0>, |11...1>, |10...1>), permettant un calcul massivement parallèle.

Transformée de Fourier quantique

La transformée de Fourier quantique est une transformation linéaire sur des bits quantiques, et est l'analogue quantique de la transformée de Fourier discrète.

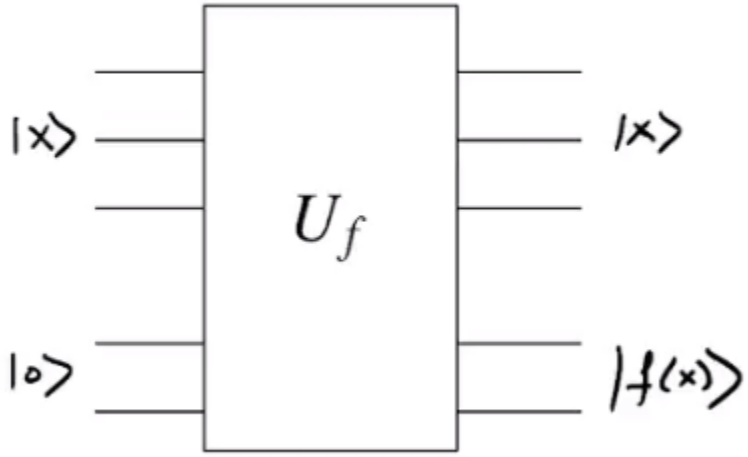
Comme la transformée de Fourier classique, la transformée de Fourier quantique prend les données de la représentation du signal d'origine vers la représentation du domaine fréquentiel. Elle diffère de la transformée de Fourier classique en ce qu'il fonctionne sur un état de superposition et produit un état de superposition différent en sortie.

La transformée de Fourier quantique peut être effectuée efficacement sur un ordinateur quantique, avec une décomposition particulière en un produit de matrices unitaires plus simples. A l'aide d'une décomposition simple, la transformée de Fourier discrète sur 2^n amplitudes peuvent être mises en œuvre sous la forme d'un circuit quantique. Cela peut être comparé à la transformée de Fourier discrète classique. Cependant, la transformée de Fourier quantique agit sur un état quantique, alors que la transformée de Fourier classique agit sur un vecteur, de sorte que toutes les tâches qui utilisent la transformée de Fourier classique ne peuvent pas tirer parti de cette accélération exponentielle.

Période :

Supposons que M est un nombre à 1000 chiffres et que r environ égale à M/2

Solution quantique :

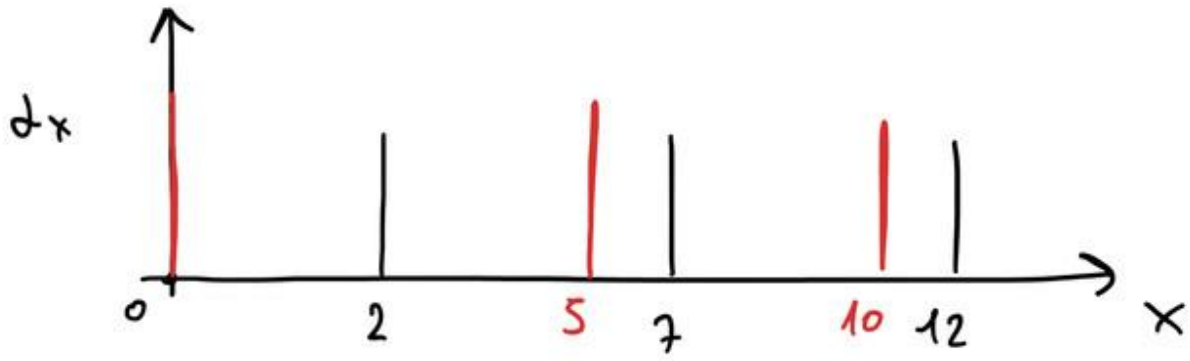


Nous posons une superposition uniforme sur quantum(x)
, et la sortie sera :

$$\frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle|f(x)\rangle \xrightarrow{measure} \sum_{x=0}^{M-1} \alpha_x |x\rangle$$

Maintenant on mesure f(quantum(x)) et on remarque que pour plusieurs valeurs de x, on a f(quantum(x)) = 4
La propriété de ces valeurs est qu'elles sont également éloignées les unes des autres.

Ce que nous pouvons faire maintenant, c'est l'échantillonnage de Fourier, en tirant parti de la propriété suivante : si nous décalons la superposition d'entrée, la sortie de l'échantillonnage de Fourier ne change pas



C'est bien parce que nous avons maintenant une superposition périodique où les amplitudes non nulles sont exactement les multiples de la période.

On obtient donc :

$$\sqrt{\frac{r}{M}} \sum_{j=0}^{\frac{M}{r}-1} |jr\rangle$$

et la sortie sera un multiple de M/r

En exécutant ceci plusieurs fois, nous obtiendrons ces multiples de M/r, donc gcd(m1, m2) = M/r, en connaissant M => r