ÉCOLE
NORMALE
SUPÉRIEURE

DÉPARTEMENT DE
MATHÉMATIQUES ET
APPLICATIONS

Factoriser 21

Gaspard DOUSSON LYS, Eliott GALLOIS & Giacomo SPRIANO,
encadrés par David GONTIER & Phong NGUYEN

Avec l'émergence de l'écriture, les communications humaines ont pu s'étendre à large échelle, libérées des contraintes physiques de l'oral. Dès lors, l'apparition d'intermédiaires a suscité la question de la sécurité de ces communications, donnant naissance à la cryptographie. Cette discipline a particulièrement révélé l'importance militaire de la capacité de calcul d'un État au cœur de la Seconde guerre mondiale. L'apparition des ordinateurs et le développement exponentiel de leur efficacité a mis en lumière, plus que jamais auparavant, la nécessité de la cryptographie. Bien que des protocoles sécurisés existent à l'heure actuelle, l'émergence d'une informatique nouvelle, dite quantique, semble bousculer cet équilibre. On cherche à comprendre ce changement inédit, dont l'efficacité ne réside pas dans un calcul rapide mais dans un calcul différent.
Dans un premier temps, on expose des postulats et principes fondamentaux de la physique quantique dans le cadre de l'informatique quantique. Il s'agit ensuite d'étudier en quoi les nouveaux paradigmes introduits permettent de résoudre très efficacement des problèmes spécifiques, et ce par l'exemple de la factorisation d'un entier. Des généralisations vers le problème du sous-groupe caché sont également exposées. Enfin, une annexe expose un certain nombre de compléments n'ayant pas trouvé leur place dans le rythme du texte principal.

I. Physique et informatique quantique.

La physique quantique est fondée sur un ensemble de postulats dont un certain nombre, simplifiés ici, suffisent à distinguer très clairement l'informatique quantique de l'informatique classique. Déjà, un système quantique isolé est décrit, en tout instant, par un vecteur normalisé d'un espace de Hilbert appelé *espace des états*. On se restreint, dans le cadre de

l'informatique quantique, à un ensemble de *qubits*, c'est-à-dire au produit tensoriel d'un nombre fini d'espaces canoniques $\mathbf{C}^2$, chacun muni d'une base orthonormée notée $(|0\rangle, |1\rangle)$. On peut alors encoder toute information classique $x = \sum x_k 2^k = x_n \dots x_0$ dans un état quantique défini par $|x\rangle = \left|\sum x_k 2^k\right\rangle = |x_n \dots x_0\rangle = |x_n\rangle \otimes \cdots \otimes |x_0\rangle$. On en déduit des superpositions d'informations classiques, à l'image de $\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$. Les états particuliers ne correspondant pas à la donnée d'une information classique mais à la superposition d'au moins deux sont dits *intriqués*. Par exemple, $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ est intriqué, mais pas $|00\rangle = |0\rangle \otimes |0\rangle = |0\rangle|0\rangle$.

Postulat (de Schrödinger)**.** *Toute évolution d'un système quantique isolé correspond à une transformation linéaire unitaire de l'espace des états.*

Bien qu'un ordinateur quantique puisse profiter du principe de superposition pour réaliser une même opération sur plusieurs jeux de données classiques d'un seul coup, il est limité à des transformations unitaires, et en particulier réversibles. Il est notamment impossible de calculer l'image d'un ensemble de qubits $|x\rangle = |x_n \dots x_0\rangle$ par une fonction trop quelconque $f : \{0,1\}^{n+1} \to \{0,1\}^{n+1}$, contrairement au cas classique.
Une solution souvent nécessaire est d'allouer un certain nombre de qubits à un travail d'artefact dans les algorithmes quantiques. En l'occurence, n autres qubits $\left|y\right\rangle = \left|y_n \dots y_0\right\rangle$ suffisent pour se ramener à l'opérateur unitaire appelée *oracle* $O_f : \left|y\right\rangle|x\rangle \mapsto \left|y \oplus f(x)\right\rangle|x\rangle$, où $\oplus$ désigne l'addition modulo 2. Ainsi, pour $f : \{0,1\} \to \{0,1\}$, on peut calculer d'un seul coup $O_f\left(|0\rangle\left(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right)\right) = \frac{1}{2}\left|0 \oplus f(0)\right\rangle|0\rangle + \frac{\sqrt{3}}{2}\left|0 \oplus f(1)\right\rangle|1\rangle = \frac{1}{2}\left|f(0)\right\rangle|0\rangle + \frac{\sqrt{3}}{2}\left|(f(1)\right\rangle|1\rangle$.

Postulat (de quantification)**.** *La mesure d'un qubit dans l'état (normalisé)* $\lambda|0\rangle + \mu|1\rangle$ *renvoie* 0*, avec probabilité* $|\lambda|^2$*, ou* 1*, avec probabilité* $|\mu|^2$*.*

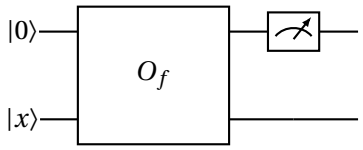
Postulat (de réduction du paquet d'onde)**.** *La mesure d'un qubit projette son état dans le sous-espace associé au résultat de la mesure.*

En plus de la restriction à des transformations unitaires, qu'on a pu contourner au prix d'une augmentation de la complexité spatiale, il n'est pas possible d'accéder à l'ensemble des résultats après un calcul. En effet, lorsqu'on mesure des qubits, on obtient un seul état de la superposition, et tous les autres disparaissent définitivement. Bien qu'on ait pu calculer simultanément $f(0)$ et $f(1)$ par le biais de l'oracle de f, on ne peut observer directement le résultat intriqué $\frac{1}{2}|0\rangle\left|f(0)\right\rangle + \frac{\sqrt{3}}{2}|1\rangle\left|f(1)\right\rangle$ mais seulement l'information classique correspondant à $|0\rangle\left|f(0)\right\rangle$, avec probabilité $\frac{1}{4}$, ou à $|1\rangle\left|f(1)\right\rangle$, avec probabilité $\frac{3}{4}$.
Même si on mesure seulement le second qubit de $\frac{1}{2}|0\rangle\left|f(0)\right\rangle + \frac{\sqrt{3}}{2}|1\rangle\left|f(1)\right\rangle$, on annihile tout état incompatible. Par exemple, si on extrait l'information $f(0)$ sur un bit classique, l'état est irrémédiablement transformé en $|0\rangle\left|f(0)\right\rangle$, car l'état $|1\rangle\left|f(1)\right\rangle$ n'appartient pas au sous-espace associé à $\mathbf{C}^2 \otimes \mathbf{C}\left|f(0)\right\rangle$. Cet aléatoire, introduit par la mesure d'un système quantique et la disparition définitive des états incompatibles, impose souvent d'appeler plusieurs fois un même algorithme avant de pouvoir réellement exploiter ses résultats. On constatera cependant qu'on peut exploiter la réduction du paquet d'onde à notre avantage dans certains cas.

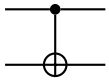
Théorème 1 (de non-clonage)**.** *Il est impossible de cloner un état quantique.*

Démonstration. Supposons qu'il existe une transformation unitaire de clonage $C : |x\rangle |0\rangle |0\rangle \mapsto |x\rangle |x\rangle \left|y\right\rangle$, où on autorise même l'usage d'un artefact pour une preuve plus générale. Considérons alors deux états $|x_1\rangle$ et $|x_2\rangle$ tels que $0 < \langle x_1|x_2\rangle < 1$. Le clonage étant unitaire, on vérifie $\langle x_1|x_2\rangle = (\langle x_1| \langle 0| \langle 0|)(|x_2\rangle |0\rangle |0\rangle) = C(\langle x_1| \langle 0| \langle 0|)C(|x_2\rangle |0\rangle |0\rangle) = \langle x_1|x_2\rangle\langle x_1|x_2\rangle\langle y_1|y_2\rangle$, d'où $\langle x_1|x_2\rangle\langle y_1|y_2\rangle = 1$, car $\langle x_1|x_2\rangle > 0$, ce qui contredit la normalisation de $\left|y\right\rangle_1$ et $\left|y\right\rangle_2$, car $\langle x_1|x_2\rangle < 1$. □

On aurait pu espérer réaliser des calculs sur une superposition d'états, puis contourner le problème induit par la réduction du paquet d'onde en clonant le résultat obtenu pour en réaliser autant de mesures que souhaitées, et ce sans jamais réitérer le calcul. Cette tactique est cependant interdite par la physique quantique, qui refuse l'existence d'une transformation unitaire de la forme $|x\rangle |0\rangle \mapsto |x\rangle |x\rangle$.
On représente les algorithmes quantiques sous la forme de *circuits quantiques* où les qubits utilisés sont des fils horizontaux traversant, de gauche à droite, les mesures et opérateurs symbolisés par des boîtes. En guise d'illustration, le calcul de f sur un qubit $|x\rangle$ suivi de la mesure d'une des images obtenues s'écrit comme suit.

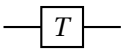


De la même manière qu'un ordinateur classique se décompose en un ensemble de portes logiques qu'on assemble pour former des algorithmes complexes, la construction d'un algorithme quantique se base sur un ensemble de transformations unitaires élémentaires. De plus, étant donné une porte U, on introduit la porte CU, lue *controlled-U*. Celle-ci utilise un qubit supplémentaire déterminant si l'opérateur U est appliquée ou non. Autrement dit, $CU(|0\rangle |x\rangle) = |0\rangle |x\rangle$ et $CU(|1\rangle |x\rangle) = |1\rangle U(|x\rangle)$. On la représente par la boîte U d'où sort un fil vertical vers le qubit de contrôle.



CNOT

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



$$T = \begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{pmatrix}$$

Dans le cadre de ce travail, on se munit des portes quantiques {CNOT, H, T} représentées ci-dessus. La porte CNOT correspond à la négation binaire contrôlée, avec $\text{NOT}|0\rangle = |1\rangle$ et $\text{NOT}|1\rangle = |0\rangle$. La porte H, dite *d'Hadamard*, permet de former une superposition de *parité* celle du qubit d'entrée. Enfin, la porte T s'interprète comme la rotation de $\frac{\pi}{4}$ radians de la *phase*. On peut montrer que l'ensemble de ces trois portes est *universel*, c'est-à-dire suffisant pour approcher aussi précisément [1] que souhaité toute transformation unitaire. Par exemple,

1. Au sens de la norme d'opérateur subordonnée à la norme euclidienne pour la suite, celle-ci étant naturelle pour travailler sur des vecteurs systématiquement normalisés. Pour autant, n'importe quelle norme sur les transformations linéaires unitaires de $\mathbf{C}^{2^n}$ convient, celles-ci étant toutes équivalentes.

il existe une manière d'implémenter, à partir des portes CNOT, H et T, le *décalage de phase* de $\frac{\pi}{2^n}$ radians défini par $P_n|0\rangle = |0\rangle$ et $P_n|1\rangle = e^{i\frac{\pi}{2^n}}|1\rangle$.

Théorème 2. *Pour toute transformation linéaire unitaire U et réel $\epsilon > 0$, il existe un circuit composé exclusivement de portes* CNOT, *H et T réalisant une transformation linéaire unitaire $\widetilde{U}$ telle que tout état (normalisé) $|x\rangle$ vérifie $\left\| U|x\rangle - \widetilde{U}|x\rangle \right\| \leq \epsilon$.*

La *complexité temporelle* d'un algorithme quantique est alors définie comme le nombre de portes quantiques élémentaires nécessaires pour l'implémenter en fonction de la taille de l'entrée, c'est-à-dire en fonction du nombre de qubits. Cette notion permet, dans une certaine mesure, de comparer l'efficacité des versions classiques et quantiques d'un même algorithme.

II. Factorisation d'un entier : réduction générale.

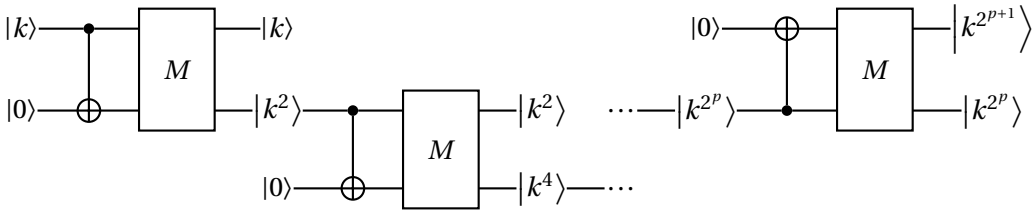
La cryptographie est fondée sur un ensemble de problèmes complexes permettant de sécuriser les communications informatiques. Le chiffrement RSA utilise par exemple la difficulté de la factorisation d'un entier. L'idée est qu'un ordinateur choisisse deux nombres premiers p et q grands, puis annonce publiquement leur produit $N = pq$. Une communication lui étant adressée peut alors être encodée à l'aide de N, et de sorte qu'elle ne puisse plus être déchiffrée qu'en connaissant p et q. Seul l'ordinateur destinataire pourra alors la comprendre.
On va désormais étudier comment ce problème de factorisation, difficile pour un ordinateur classique, peut être résolu efficacement, c'est-à-dire en complexité polynomiale, à l'aide d'un ordinateur quantique. Il s'agit donc de trouver un algorithme renvoyant un facteur non trivial de N. Pour cela, on propose la méthode suivante :

1. choisir $k < N$ pseudo-aléatoirement ;
2. calculer $p = \text{pgcd}(k, N)$;
3. si $p \neq 1$, alors il s'agit d'un facteur non trivial de N ;
4. sinon, calculer l'ordre r de k dans $(\mathbf{Z}/N\mathbf{Z})^\times$;
5. si r est impair, repartir à l'étape 1 ;
6. si r est pair, calculer $q = \text{pgcd}(k^{\frac{r}{2}} - 1, N)$;
7. si $q \neq N$, alors il s'agit d'un facteur non trivial de N ;
8. sinon, repartir à l'étape 1.

Le processus, en dehors de l'étape 4, consiste simplement en une succession d'algorithmes d'Euclide et de vérifications pouvant mener à des redémarrages. La seconde partie repose sur le fait que, si r est pair et si $k^{\frac{r}{2}} - 1$ et $k^{\frac{r}{2}} + 1$ ne sont pas divisibles par N, alors le résultat est un facteur non trivial de N, car $(k^{\frac{r}{2}} - 1)(k^{\frac{r}{2}} + 1) \equiv k^r - 1 \equiv 0 \pmod{N}$.

Théorème 3. *Si N n'est pas pair ou une puissance d'un premier, alors la probabilité pour que $k \in (\mathbf{Z}/N\mathbf{Z})^\times$ soit d'ordre r pair et que N ne divise ni $\left(k^{\frac{r}{2}} - 1\right)$ ni $\left(k^{\frac{r}{2}} + 1\right)$ est d'au moins $\frac{1}{2}$.*

La complexité temporelle de l'algorithme d'Euclide est quadratique. Le choix initial et les vérifications sont réalisées en $O(1)$. La procédure est réitérée autour de $O(1)$ fois du fait du théorème 3. Finalement, la complexité temporelle totale est $O\left(\max(\log N, C(N))\log^2 N\right)$, avec C la complexité de la quatrième étape.
On s'est ramené au problème de recherche d'un ordre en temps polynomial. On dispose de $k \in (\mathbf{Z}/N\mathbf{Z})^\times$, et on pose $f : x \mapsto k^x$. Il s'agit d'une fonction périodique, de période r l'ordre de k, et injective sur $[\![1, r]\!]$. C'est pour ce problème de recherche de période qu'on va utiliser l'informatique quantique. En effet, on peut implémenter O_f en temps polynomial par un analogue de l'exponentiation rapide classique.
Pour commencer, on construit la multiplication modulaire $M : |x\rangle\left|y\right\rangle \mapsto \left|xy \pmod N\right\rangle$ à partir de $O(n)$ sommes modulaires constituée de $O\left(n^2\right)$ portes CNOT. On peut alors calculer O_f sur les états de la forme $|2^p\rangle$ à l'aide de copies par CNOT suivies de multiplications par M, comme représenté ci-dessous. On en déduit O_f en $O\left(n^4\right)$ grâce à $f(x+y) = f(x)f(y)$. En effet, dans l'état $|x\rangle|1\rangle$, il suffit de multiplier le second membre par $\left|k^{2^p}\right\rangle$ sous réserve que $\left|x_p\right\rangle = |1\rangle$ à l'aide de CM pour chaque $p \in [\![0, n-1]\!]$, avant d'atteindre $|x\rangle|k^x\rangle = |x\rangle\left|f(x)\right\rangle$.



Dans le cas général d'un entier quelconque N, on peut directement lire sa valuation dyadique dans son écriture binaire. Par ailleurs, le calcul des racines n-èmes, pour $n \in [\![2, \log_2 N]\!]$, éliminant le cas d'une puissance d'un premier, s'implémente en temps quadratique. On se ramène alors au cas d'application du théorème et de l'algorithme précédents, qui permettent d'obtenir un facteur non trivial de N. Il suffit de réitérer la méthode jusqu'à obtenir sa décomposition en facteurs premiers.

III. Recherche d'une période : cas particulier.

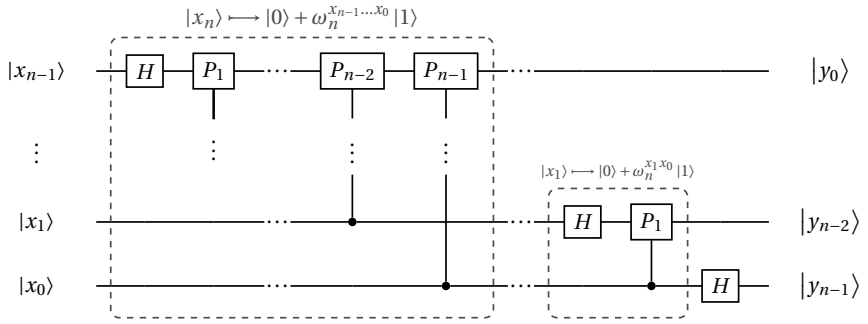
L'algorithme de Shor pour la recherche de période, résolvant le problème auquel on a ramené la factorisation, utilise l'opérateur quantique $\mathscr{F}_n$ de la transformée de Fourier. En notant $\omega_n = e^{i\frac{2\pi}{2^n}}$, il est défini sur $(\mathbf{C}^2)^{\otimes n}$ par

$$\mathscr{F}_n|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}\omega_n^{xy}\left|y\right\rangle.$$

On reconnaît la transformée de Fourier discrète, bien que les objets sur lesquels $\mathscr{F}_n$ opère sont de nature très différente. Par périodicité des racines de l'unité, on a que

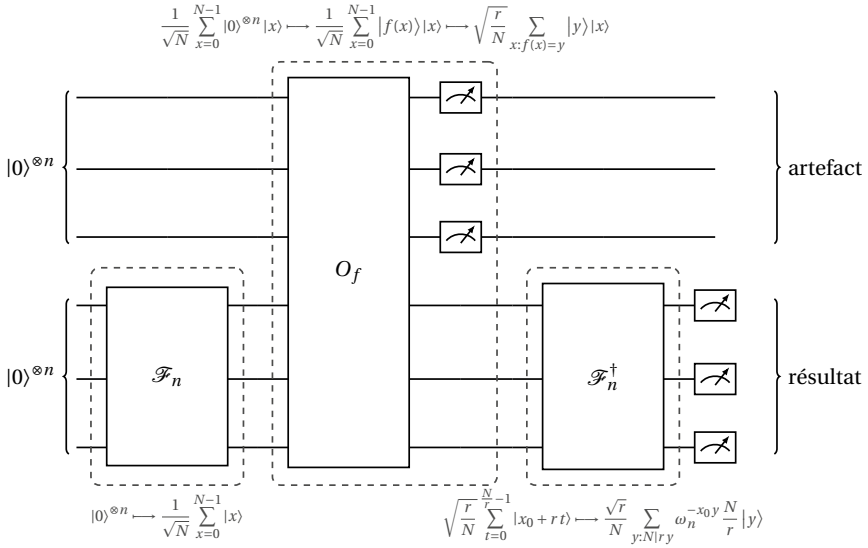
$$\sqrt{2^n}\mathscr{F}_n|x\rangle = \sum_{y=0}^{2^n-1}\omega_n^{x\left(\sum y_{n-k}2^{n-k}\right)}\left(\bigotimes_{k=1}^{n}\left|y_{n-k}\right\rangle\right) = \bigotimes_{k=1}^{n}\left(\sum_{y_{n-k}=0}^{1}\omega_k^{xy_{n-k}}\left|y_{n-k}\right\rangle\right) = \bigotimes_{k=1}^{n}\left(|0\rangle + \omega_k^{x_{k-1}\cdots x_0}|1\rangle\right)$$

d'où on déduit le circuit de l'algorithme quantique implémentant $\mathscr{F}_n$ ci-après. Notons qu'on peut utiliser $\left\lfloor \frac{n}{2} \right\rfloor$ opérations SWAP échangeant deux qubits pour ordonner la sortie en cohérence avec l'entrée.



Classiquement, la transformée de Fourier rapide s'implémente en $O(N\log N)$. Ici, on constate que la transformée de Fourier quantique est exponentiellement plus efficace, avec pour complexité $O\left(n^2\right) = O\left(\log^2 N\right)$. La comparaison est cependant injuste en cela que les deux opérateurs agissent sur des objets de nature très différente, et ne fournissent donc pas du tout les mêmes types de résultats.

Revenons désormais au problème de recherche de fréquence. On cherche la période r de f telle que f se restreint sur $[\![1,r]\!]$ en une fonction injective. Dans un premier temps, on suppose que $N = 2^n$ et que $r|N$ pour exposer clairement l'essentiel de l'algorithme. L'idée est de commencer par construire une superposition équiprobable des états encodant les éléments de $[\![0,N-1]\!]$. On y applique l'oracle O_f pour lier ces nombres à leurs images par f, avant de mesurer une de ces images pour réduire le paquet d'onde [2]. La mesure après transformée de Fourier quantique réciproque permet alors d'obtenir un entier y tel que $N|r\,y$.



2. En pratique, la mesure de l'artefact n'est pas nécessaire, car la projection induite commute avec $\mathscr{F}_n^\dagger$.

Bien que la transformée de Fourier quantique apparaissent deux fois dans ce circuit, son premier usage n'est qu'une manière détournée de créer la superposition équiprobable initiale. Seule la seconde utilisation correspond réellement à une étude de période. Finalement, il suffit de relancer l'algorithme jusqu'à obtenir deux entiers $y_1 = k_1 \frac{N}{r}$ et $y_2 = k_2 \frac{N}{r}$ avec k_1 et k_2 premiers entre eux pour retrouver r.

Théorème 4. *Si k_1 et k_2 sont non nuls et premiers entre eux, alors $r = \text{ppcm}(b_1, b_2)$, où b_1 et b_2 sont les dénominateurs irréductibles des fractions $\frac{y_1}{N}$ et $\frac{y_2}{N}$.*

Démonstration. On note $\frac{a_1}{b_1}$ la fraction irréductible égale à $\frac{y_1}{N} = \frac{k_1}{r}$. Par lemme de Gauss, puisque les entiers a_1 et b_1 sont premiers entre eux, on a que $b_1 | r$. De même $b_2 | r$, d'où on tire que $\text{ppcm}(b_1, b_2)$ divise r. En décomposant $r = \prod p_i^{\nu_i}$, on a que tout $p_i^{\nu_i}$ divise $b_1 k_2$ et $b_2 k_2$. Or k_1 et k_2 sont premiers entre eux, donc $p_i^{\nu_i}$ divise b_1 ou b_2, et r divise $\text{ppcm}(b_1, b_2)$. □

Vérifier que k_1 et k_2 sont convenables et calculer r ne nécessite qu'un temps quadratique, toujours par l'algorithme d'Euclide. Pour obtenir k_1 et k_2, un appel de l'algorithme de recherche de période de Shor consiste en deux transformées de Fourier et un oracle, d'où une exécution en $O\big(\text{poly}(n)\big) = O\big(\text{poly}(\log N)\big)$, et il faut seulement $O(1)$ tels appels pour vérifier les conditions de primalité du théorème précédent. En effet, la probabilité de succès tend exponentiellement vite vers 1, par le théorème 5, d'où une probabilité de conclusion en $1 - O\big(\frac{1}{k}\big)$ après $O(\log k)$ appels.

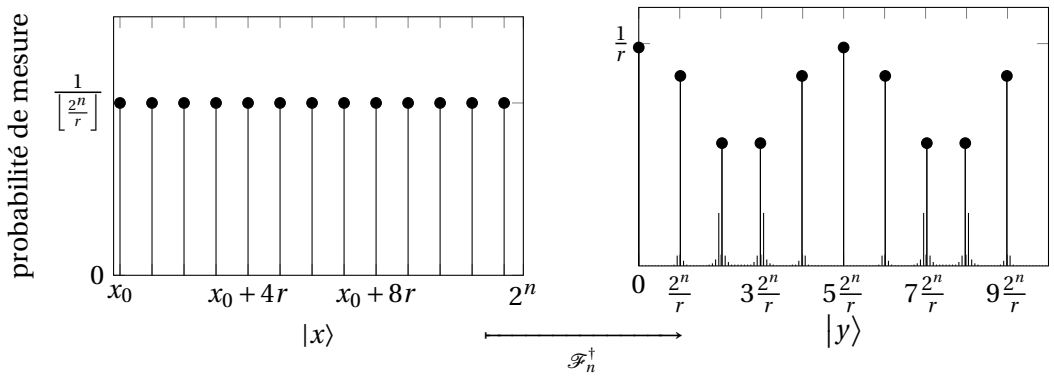
Théorème 5. *Pour deux appels indépendants de l'algorithme, la probabilité d'obtenir k_1 et k_2 premiers entre eux est d'au moins* 35%.

Démonstration. En considérant l'évènement contraire, on a $\mathbf{P}\big[k_1 \text{ et } k_2 \text{ non copremiers}\big] \leq \sum_{p \text{ premier}} \mathbf{P}\big[p|k_1 \text{ et } p|k_2\big] = \sum_{p \text{ premier}} \mathbf{P}\big[p|k_1\big]\mathbf{P}\big[p|k_2\big] = \sum_{p \text{ premier}} \frac{1}{p}\frac{1}{p} \leq \sum_{n \geq 2} \frac{1}{n^2} \leq \frac{\pi^2}{6} - 1 \leq 0{,}65.$ □

IV. Recherche d'une période : cas général.

La méthode exposée jusqu'à présent repose sur deux hypothèses simplificatrices qu'on souhaite désormais lever : $N = 2^n$ et $r | N$. Déjà, si N n'est pas une puissance de 2, alors on peut simplement étendre le domaine de f de $[\![0, N-1]\!]$ à $[\![0, 2^n - 1]\!]$, avec $2^n \geq N$. L'algorithme précédent continue de fonctionner sous réserve que $r | 2^n$. Pour la suite, on fixe plus particulièrement n tel que $2^n > N^2$.
Lorsque r ne divise pas 2^n, la périodicité ne tombe pas parfaitement sur la fin du domaine et l'application de $\mathscr{F}_n^\dagger$ laissera transparaître du bruit, comme ci-après [3] avec $r = 10$, $N = 77$ et $2^n = 128$. En conséquence, il existe une probabilité non nulle que le résultat de l'algorithme de Shor ne soit pas voisin d'un multiple de $\frac{2^n}{r}$. Cette probabilité peut cependant être majoré par une constante, de sorte qu'un nombre suffisant d'appels permet de converger vers un entier y tel qu'il existe $k \in \mathbf{N}$ vérifiant $\left| y - k\frac{2^n}{r} \right| \leq \frac{1}{2}$.

3. On ne respecte exceptionnellement pas la condition $2^n > N^2$ pour rendre plus visible ce bruit apparent.



Théorème 6. *La probabilité que y soit à une distance au plus $\frac{1}{2}$ d'un multiple de $\frac{2^n}{r}$ est d'au moins* 40%.

On en déduit $\frac{k}{r}$ en calculant $\frac{y}{2^n}$ en vertu du théorème ci-après, car $\left|\frac{y}{2^n} - \frac{k}{r}\right| \leq \frac{1}{2\times 2^n} < \frac{1}{2N^2} \leq \frac{1}{2r^2}$: c'est cette étape qui impose de choisir n de sorte que $2^n > N^2$. Le résultat s'obtient après un développement en fraction continue de complexité cubique.

Théorème 7. *Étant donné un nombre rationnel $x \in \mathbf{Q}$, il existe une unique fraction $\frac{p}{q} \in \mathbf{Q}$ tel que $\left|x - \frac{p}{q}\right| < \frac{1}{2q^2}$.*

Pour en déduire r, il faut que ce dernier soit premier avec k. Cet évènement a une probabilité de réalisation minorée par une constante, par le même argument que le théorème 5, et on peut ainsi se permettre de réitérer le processus si le résultat obtenu ne correspond pas à une période [4] de f.

V. Factorisation d'un entier : exemple particulier.

En résumé, on a pu ramener le problème de factorisation à un problème de recherche de période sur une machine classique, puis on a utilisé un ordinateur quantique pour résoudre ce nouveau problème en temps polynomial. On a donc établi une méthode théorique pour la factorisation des entiers en $O\big(\text{poly}(\log N)\big) = O\big(\text{poly}(n)\big)$, alors même que les algorithmes classiques actuels requièrent une complexité a minima sous-exponentielle.
Pour fixer les idées, détaillons l'algorithme complet dans le cas $N = 21$, avec $n = 9$. On se munit, par exemple, de $k = 5$, inférieur et premier avec 21. On cherche alors la période de $f : x \mapsto 5^x \pmod{21}$. Pour cela, on applique l'oracle O_f pour transformer la superposition comme suit.

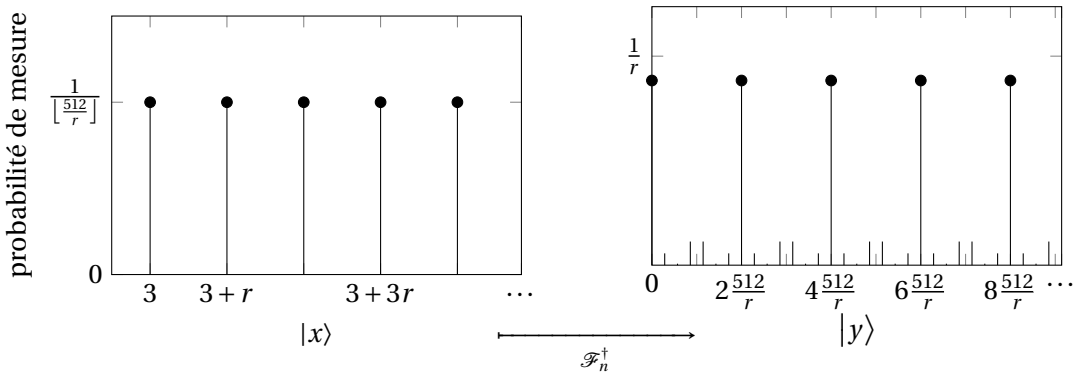
$$\frac{1}{\sqrt{512}} \sum_{x=0}^{511} |0\rangle^{\otimes 9} |x\rangle \longmapsto \frac{1}{\sqrt{512}} \sum_{x=0}^{511} \big|f(x)\big\rangle |x\rangle$$

4. L'algorithme déduisant r par le théorème 7 renvoie un résultat strictement inférieur à r en cas d'erreur, c'est-à-dire en présence d'une fraction $\frac{k}{r}$ n'étant pas irréductible. Ainsi, si le résultat correspond à une période de f, il ne peut s'agir que de r, par minimalité. Par injectivité sur une période, cette vérification se résume au calcul de $f(r)$, supposé donner 1.

On mesure, par hasard, l'image 16, ce qui projette la superposition du domaine de f sur la superposition des antécédents de 16 par f. Enfin, la transformée de Fourier réciproque réalise

$$\sqrt{\frac{r}{512}}\sum_{f(x)=16}|x\rangle=\sqrt{\frac{r}{512}}\sum_{t=0}^{\lfloor\frac{512}{r}\rfloor-1}|4+r\,t\rangle\longmapsto\frac{1}{\sqrt{512}}\sum_{y=0}^{511}\sqrt{\frac{r}{512}}\sum_{t=0}^{\lfloor\frac{512}{r}\rfloor-1}\omega_9^{-(4+r\,t)\,y}\,|y\rangle$$

On obtient, par une mesure aléatoire simulée en annexe, l'entier $y=426$, de fraction associée $\frac{y}{2^n}=\frac{426}{512}$. Un algorithme de développement en fraction continu, admis ici, permettrait de calculer la fraction $\frac{5}{6}$ vérifiant $\left|\frac{426}{512}-\frac{5}{6}\right|<\frac{1}{6^2}$, et donc égale à $\frac{k}{r}$ par théorème. On aurait ainsi $r=6$, sous réserve que k est premier avec r. Or, si ce n'était pas le cas, on aurait $6<r$, d'où $f(6)\neq f(0)$ par injectivité de f sur une période. Il suffit de la simple vérification $f(6)=1$ pour conclure qu'effectivement $r=6$. Il s'agit bien d'un ordre pair et $10^{\frac{6}{2}}-1=999$ n'est pas un multiple de 21, d'où le facteur non trivial $\text{pgcd}(999,21)=3$, et $21=3\times7$.



VI. Problème du sous-groupe caché : cas discret.

Le problème de recherche d'une période minimale se prolonge en un problème plus général dit du sous-groupe caché. On se donne une application f d'un groupe $\mathbf{G}$ dans un ensemble fini X et on cherche le sous-groupe $\mathbf{H}=\{h\in\mathbf{G}|\forall g\in\mathbf{G},f(gh)=f(g)\}$, en admettant que l'application factorisée $\tilde{f}:\mathbf{G}/\mathbf{H}\to X$ est injective. On retrouve bien notre problème précédent avec $f:x\in\mathbf{Z}/\phi(N)\mathbf{Z}\mapsto k^x\in\mathbf{Z}/N\mathbf{Z}$ cachant le sous-groupe $r\mathbf{Z}/\phi(N)\mathbf{Z}$.

Théorème 8 (de Shor)**.** *Sur un ordinateur quantique, le problème du sous-groupe caché se résout en temps polynomial en* $\log(\text{card}\,\mathbf{G})$ *pour les groupes abéliens finis.*

En particulier, le théorème de Shor indique à nouveau que le problème de recherche de période se résout en $O\big(\text{poly}(\log\phi(N))\big)=O\big(\text{poly}(\log N)\big)$. L'essentiel de cette section est dédiée à la démonstration de ce théorème, par le biais d'une généralisation de la transformée de Fourier. Pour cela, on introduit, étant donné $\mathbf{G}$, le groupe $\widehat{\mathbf{G}}$ des morphismes de $\mathbf{G}$ dans $\mathbf{C}^\times$, appelés *caractères* et munis du produit $\times$ induit par $\mathbf{C}^\times$.

Théorème 9. $\widehat{\mathbf{G}}$ *est isomorphe à* $\mathbf{G}$.

On dispose d'un isomorphisme $g \in \mathbf{G} \mapsto \chi_g \in \widehat{\mathbf{G}}$. De plus, on note $(|g\rangle)$ une base orthonormée de $\mathbf{C}^{\mathbf{G}}$ et $(\langle g|)$ la base duale associée. On définit alors la transformée de Fourier $\mathscr{F}$ comme l'endomorphisme sur $\mathbf{C}^{\mathbf{G}}$ défini par $\mathscr{F}|g\rangle = \frac{1}{\sqrt{\operatorname{card}\mathbf{G}}}\chi_g$.

Théorème 10. *$\mathscr{F}$ est un opérateur unitaire.*

Démonstration. Il s'agit de montrer que la famille $\left(\frac{1}{\sqrt{\operatorname{card}\mathbf{G}}}\chi_g\right)$ est une base orthonormée de $\mathbf{C}^{\mathbf{G}}$. On vérifie que

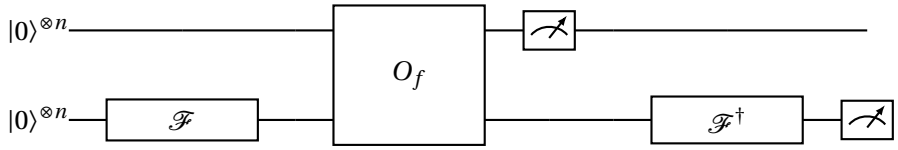
$$\langle \chi_{g_1} | \chi_{g_2} \rangle = \sum_{x\in\mathbf{G}} \chi_{g_1}(x)\overline{\chi_{g_2}(x)} = \sum_{x\in\mathbf{G}} \chi_{g_1}(x)\chi_{g_2}(x)^{-1} = \sum_{x\in\mathbf{G}} \chi_{g_1 g_2^{-1}}(x)\,.$$

Si $g_1 = g_2$, alors $\chi_{g_1 g_2^{-1}} = \tilde{1}$ et la somme est égale à $\operatorname{card}\mathbf{G}$. Sinon, il existe $y \in \mathbf{G}$ tel que $\chi_{g_1 g_2^{-1}}(y) \neq 1$, et alors $\langle \chi_{g_1} | \chi_{g_2} \rangle = 0$, étant donné qu'on vérifie

$$\chi_{g_1 g_2^{-1}}(y)\langle \chi_{g_1} | \chi_{g_2} \rangle = \chi_{g_1 g_2^{-1}}(y) \sum_{x\in\mathbf{G}} \chi_{g_1 g_2^{-1}}(x) = \sum_{x\in\mathbf{G}} \chi_{g_1 g_2^{-1}}(yx) = \sum_{x\in\mathbf{G}} \chi_{g_1 g_2^{-1}}(x) = \langle \chi_{g_1} | \chi_{g_2} \rangle\,.$$

□

On introduit alors l'orthogonal $\mathbf{H}^{\perp} = \{g \in \mathbf{G} | \forall h \in \mathbf{H}, \chi_g(h) = 1\}$ du sous-groupe caché $\mathbf{H}$. C'est cet ensemble qu'on va pouvoir calculer à l'aide d'un circuit quantique. Pour cela, on reproduit l'algorithme de Shor déjà vu en produisant une première superposition $\frac{1}{\sqrt{\operatorname{card}\mathbf{G}}}\sum_{\mathbf{G}} |0\rangle |g\rangle$, pour appliquer l'oracle de f et obtenir $\frac{1}{\sqrt{\operatorname{card}\mathbf{G}}}\sum_{\mathbf{G}} |f(g)\rangle |g\rangle$. Une mesure de l'image permet de se projeter sur l'état $\frac{1}{\sqrt{\operatorname{card}\mathbf{H}}}\sum_{\mathbf{H}} |g_0 h\rangle$, où on applique une transformée de Fourier réciproque $\mathscr{F}^{\dagger}$ pour obtenir l'état final $\frac{1}{\sqrt{\operatorname{card}\mathbf{H}^{\perp}}}\sum_{\mathbf{H}^{\perp}} \chi_k(g_0)|k\rangle$. En le mesurant, on obtient un élément k de $\mathbf{H}^{\perp}$.



En réitérant le procédé autant de fois que nécessaire, on finit par obtenir l'ensemble $\mathbf{H}^{\perp}$, d'où on peut déduire $\mathbf{H}$. Par exemple, si $\mathbf{G}$ est un espace vectoriel sur $\mathbf{Z}/p\mathbf{Z}$, la notion d'orthogonal coïncide avec celle de l'espace vectoriel. On doit calculer $\dim \mathbf{H}^{\perp} = \log_p \operatorname{card}\mathbf{H}^{\perp} = O(\log \operatorname{card}\mathbf{G})$ vecteurs linéairement indépendants pour déterminer $\mathbf{H}^{\perp}$. Un ordinateur classique, avec l'algorithme de Gram-Schmidt, peut finalement calculer $\mathbf{H}$.
Cette généralisation du problème de recherche de période n'entraîne pas de difficulté supplémentaire autant d'un point de vue théorique que pratique. Elle permet cependant de traiter des problèmes plus généraux comme celui du logarithme discret. L'idée est de retrouver y connaissant $g \in \mathbf{G}$ et $x = g^y$, sous réserve que $y < \operatorname{ord}(g)$. Ce problème n'admet pas de solution générale en temps polynomial sur un ordinateur classique.
Le problème de la factorisation de $N = pq$ s'y ramène en choisissant $g \in (\mathbf{Z}/N\mathbf{Z})^{\times}$ et en calculant $x = g^N$. L'ordre du groupe étant $\phi(N) = pq - (p+q-1)$, on a en fait $x = g^{p+q-1}$. En

résolvant le problème du logarithme discret, on obtiendrait $p+q$, d'où on peut déduire p et q connaissant $pq = N$. Plus généralement, le problème du logarithme discret est utilisé en cryptographie dans le groupe des courbes elliptiques.

Corollaire. *Sur un ordinateur quantique, le problème du logarithme discret se résout en temps polynomial.*

Démonstration. On admet pouvoir se ramener au cas du logarithme discret sur un groupe $\mathbf{G}$ d'ordre premier p par le biais de l'algorithme classique de Pohlig-Hellman. On pose alors $f : \left(\mathbf{Z}/p\mathbf{Z}\right)^2 \to \mathbf{G}$ réalisant l'opération $(a,b) \mapsto g^a x^{-b}$. Le sous-groupe $\mathbf{H}$ caché par f est clairement engendré par $(y,1)$.
Les caractères de $\left(\mathbf{Z}/p\mathbf{Z}\right)^2$ s'écrivent $\chi_{(x_1,x_2)} : (y_1,y_2) \mapsto \exp\left(i2\pi\frac{x_1y_1+x_2y_2}{p}\right)$. En conséquence, l'orthogonal $\mathbf{H}^\perp$ est exactement l'ensemble $\left\{(x_1,x_2) | x_1 y + x_2 \equiv 0 \pmod p\right\}$ engendré par le vecteur $(y,1)$. L'algorithme de Shor permet finalement de trouver un élément de $\mathbf{H}^\perp$, ce qui suffit à déterminer y. □

Lorsqu'on ne suppose plus $\mathbf{G}$ abélien, on ne dispose plus d'un isomorphisme entre $\mathbf{G}$ et $\widehat{\mathbf{G}}$. Pour définir la transformée de Fourier, il s'agit de généraliser la notion de caractères en s'appuyant sur les représentations. Une *représentation* de $\mathbf{G}$ est un morphisme $\rho : \mathbf{G} \to GL_d(\mathbf{C})$, où d est appelé *degré* de la représentation. On dit qu'une représentation est *irréductible* si les seuls sous-espaces vectoriels de $\mathbf{C}^d$ stabilisé par l'action de $\rho(\mathbf{G})$ sont triviaux, c'est-à-dire $\{0\}$ et $\mathbf{C}^d$. On note $\widehat{\mathbf{G}}$ l'ensemble des représentations irréductibles [5]. On définit finalement la transformée de Fourier par

$$\mathscr{F} : |g\rangle \longmapsto \sum_{\rho\in\widehat{\mathbf{G}}} \sqrt{\frac{\deg\rho}{\operatorname{card}\mathbf{G}}} |\rho\rangle \sum_{i,j=1}^{\deg\rho} \rho(g)_{ij} |i,j\rangle \ .$$

On peut toujours identifier l'espace d'arrivée $\left\langle |\rho,i,j\rangle \right\rangle$ à $\mathbf{C}^{\mathbf{G}}$ par le théorème de Frobenius ci-dessous. De plus, la transformée de Fourier est toujours un opérateur unitaire, comme conséquence de la formule $\sum \deg\rho \operatorname{tr}\rho(g) = \mathbf{1}_{[g=1]} \operatorname{card}\mathbf{G}$ issue de la théorie des représentations.

Théorème 11 (de Frobenius)**.** $\operatorname{card}\mathbf{G} = \sum_{\rho\in\widehat{\mathbf{G}}} (\deg\rho)^2$.

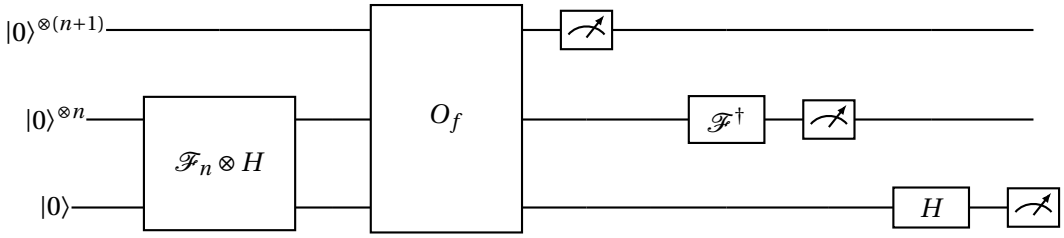
On s'intéresse plus particulièrement au cas du groupe diédral $\mathbf{D}_{2n}$, lié au *Shortest Vector Problem* de la théorie des réseaux euclidiens, utilisée aujourd'hui pour fonder la cryptographie post-quantique. Géométriquement, il s'agit du groupe d'isométrie d'un polygone régulier à n côtés. Algébriquement, il s'identifie au produit semi-direct $\mathbf{Z}/n\mathbf{Z} \rtimes_\phi \mathbf{Z}/2\mathbf{Z}$, avec $\phi : k \mapsto \left(x \mapsto (-1)^k x\right)$.
Géométriquement, les sous-groupes stricts de $\mathbf{D}_{2n}$ sont d'ordre 2 engendrés par une symétrie, cycliques engendrés par une rotation, ou diédraux. Algébriquement, ils sont, respectivement, de la forme $\langle (y,1)\rangle$, $\langle (x,0)\rangle$ ou $\langle (y,1),(x,0)\rangle$. Pour une fonction $f : \mathbf{D}_{2n} \to X$ cachant $\langle (x,0)\rangle$ ou $\langle (y,1),(x,0)\rangle$, sa restriction f au sous-groupe abélien $\mathbf{Z}/n\mathbf{Z} \rtimes \{0\}$ des rotations du

5. Cette notation coïncide, dans le cadre abélien, avec celle donnée pour les caractères.

polygone cache le sous-groupe $\langle (x,0)\rangle$. On peut donc toujours retrouver $\langle (x,0)\rangle$ par le biais du cas abélien précédent. Or $\langle (x,0)\rangle$ étant distingué, on peut ramener le cas du sous-groupe caché $\langle (y,1),(x,0)\rangle$ au cas de $\langle (y,1)\rangle$ en factorisant f en $\bar{f} : \mathbf{D}_{2n} / \langle (x,0)\rangle \cong \mathbf{D}_{2m} \to X$.
En réalité, seul le cas d'une fonction $f : \mathbf{D}_{2n} \to X$ cachant $\langle (y,1)\rangle$ ne pouvait donc être traité avec les outils précédents. On reproduit à nouveau l'algorithme de Shor en préparant une superposition initiale afin d'appliquer l'oracle de f, puis la transformée réciproque de Fourier sur les antécédents d'un élément de X.

$$O_f : \frac{1}{\sqrt{2n}} \sum_{x \in \mathbf{Z}/n\mathbf{Z}} \sum_{k \in \mathbf{Z}/2\mathbf{Z}} |0\rangle |x,k\rangle \longmapsto \frac{1}{\sqrt{2n}} \sum_{x \in \mathbf{Z}/n\mathbf{Z}} \sum_{k \in \mathbf{Z}/2\mathbf{Z}} \left| f(x,k) \right\rangle |x,k\rangle$$

$$\mathscr{F}^\dagger : \frac{1}{\sqrt{2}} \left(|x,0\rangle + \left| x+y,1 \right\rangle \right) \longmapsto \frac{1}{\sqrt{2n}} \left(\sum_{k=0}^{n-1} \left(e^{\frac{i2\pi kx}{n}} |k\rangle |0\rangle + e^{\frac{i2\pi k(x+y)}{n}} |k\rangle |1\rangle \right) \right)$$



Un état quantique ne dépendant pas d'un facteur de phase globale, on obtient finalement un résultat $k \in [\![0, n-1]\!]$ et un état $\frac{1}{\sqrt{2}} \left(|0\rangle + e^{\frac{i2\pi ky}{n}} |1\rangle \right)$. En appliquant la porte de Hadamard, l'état devient $e^{\frac{i\pi ky}{n}} \left(\cos \frac{i\pi ky}{n} |0\rangle - i \sin \frac{i\pi ky}{n} |1\rangle \right)$ pour la dernière mesure. Un algorithme, de Ettinger et Hoyer et évoqué dans [10], permet finalement d'en déduire y en temps polynomial, avec $O(\log n)$ utilisations du circuit quantique ci-dessus.

VII. Problème du sous-groupe caché : cas continu.

Un autre problème contenu dans celui du sous-groupe caché est le calcul du groupe des unités d'un anneau d'entiers algébriques. On se munit d'un corps de fractions $\mathbf{Q}(\theta)$, où $\theta \in \mathbf{C}$ est racine d'un polynôme à coefficients rationnels. L'anneau des entiers de $\mathbf{Q}(\theta)$, noté O, est alors l'ensemble des éléments de $\mathbf{Q}(\theta)$ racines d'un polynôme unitaire de $\mathbf{Z}[X]$. On souhaite déterminer l'ensemble $O^\times$ des inversibles de O.
Pour mieux comprendre $\mathbf{Q}(\theta)$, on peut s'intéresser aux morphismes de corps de $\mathbf{Q}(\theta)$ dans $\mathbf{R}$ ou dans $\mathbf{C}$. Un tel morphisme τ est entièrement déterminé par l'image qu'il donne de θ. Par ailleurs, en tant que morphisme de corps, τ correspond à l'identité sur $\mathbf{Q}$. On en déduit que $\tau(\theta)$ est racine du polynôme minimal de θ, racines qui sont en nombre fini. De plus, il en existe un nombre pair à valeurs complexes pas seulement réelles, en appareillant chacun d'entre eux à son conjugué. Il existe donc m morphismes réels et $2n$ autres morphismes complexes. On dispose alors du théorème de Dirichlet pour étudier $O^\times$.

Théorème 12 (des unités de Dirichlet)**.** *Pour un corps* $\mathbf{K}$*, le groupe* $O^\times(\mathbf{K})$ *des unités de l'anneau de ses entiers* $O(\mathbf{K})$ *est isomorphe au produit du groupe des racines de l'unité* $\mathbf{U}(\mathbf{K})$ *et du groupe abélien* $\mathbf{Z}^{m+n-1}$*.*

Le groupe $O^\times$ est discret dans $\mathbf{C}$, et donc l'ensemble $\mathbf{U}(\mathbf{Q}(\theta)) \subset O^\times$ est fini. Il s'agit finalement de déterminer le quotient $O^\times / \mathbf{U}(\mathbf{Q}(\theta))$, qui est un réseau de $\mathbf{R}^{m+n-1}$. En notant $\tau_1, \dots, \tau_n$ les morphismes de $\mathbf{Q}(\theta)$ dans $\mathbf{R}$ et $\tau_{n+1}, \dots, \tau_{n+m}$ des morphismes non conjugués de $\mathbf{Q}(\theta)$ dans $\mathbf{C}$, chaque élément $x \in O$ est caractérisé par le vecteur $(\tau_1(x), \dots, \tau_{n+m}(x))$ de $\mathbf{R}^n \times \mathbf{C}^m$. On remarque que le produit se fait coordonnée par coordonnée, c'est-à-dire que, pour $x, y \in O$, le produit xy est représenté par le vecteur $\big(\tau_1(x)\tau_1(y), \dots, \tau_{n+m}(x)\tau_{n+m}(y)\big)$.
Lorsque toutes les coordonnées du vecteur associé à $x \in O$ sont non nulles, on écrit $x = (\epsilon_1 e^{u_1}, \dots, \epsilon_{n+m} e^{u_{n+m}})$, avec $u_1, \dots, u_{n+m}$ réels et $\epsilon_1, \dots, \epsilon_{n+m}$ de module 1. On définit alors l'application $N : x = (x_1, \dots, x_{m+n}) \mapsto x_1 \dots x_m |x_{m+1}|^2 \dots |x_{m+n}|^2$. On constate que N est multiplicative, c'est-à-dire que pour $x, y \in O$, $N(xy) = N(x)N(y)$, et en particulier qu'un inversible $x \in O^\times$ vérifie $N(x) = \pm 1$.

Lemme. *Si l'élément* $x \in O$ *est inversible, alors toutes les coordonnées de* x *sont non nulles et* $u_1 + \dots + u_m + 2u_{m+1} + \dots + 2u_{m+n} = 0$*.*

Démonstration. On dispose d'un inverse x^{-1}, de sorte que $\tau_k(x)\tau_k(x^{-1}) = \tau_k(1) = 1$. Ainsi, toute coordonnée de x est inversible, donc non nulle. L'annulation de la somme découle directement de l'égalité $|N(x)| = 1$. □

Par lemme, le noyau $\mathbf{G}$ du morphisme de groupes $(\epsilon_1 e^{u_1}, \dots, \epsilon_{n+m} e^{u_{n+m}}) \mapsto u_1 + \dots + 2u_{m+n}$ contient $O^\times$. Or se donner un élément de $\mathbf{G}$, c'est se donner $m+n-1$ réels, m signes et n phases, d'où $\mathbf{G} \cong \mathbf{R}^{m+n-1} \times (\mathbf{Z}/2\mathbf{Z})^m \times (\mathbf{R}/\mathbf{Z})^n$. On y lit la torsion de $\mathbf{G}$, c'est-à-dire les racines de l'unité $\mathbf{U}(\mathbf{Q}(\theta))$, qui n'est autre que $(\mathbf{Z}/2\mathbf{Z})^m \times (\mathbf{R}/\mathbf{Z})^n$. Finalement, en notant Λ_{m+n-1} l'ensemble des réseaux de $\mathbf{R}^{m+n-1}$, l'application $f : \mathbf{G} \to \Lambda_{m+n-1}$ associant au couple (u, ϵ) le réseau $(\epsilon_k e^{u_k})\, O$ cache le groupe $O^\times$ recherché [6].
Bien qu'on voudrait utiliser les méthodes précédentes, le groupe de travail $\mathbf{G}$ n'est désormais plus fini. On parle de problème du sous-groupe caché continu pour une fonction $f : \mathbf{R}^m \to X$, lipschitzienne et périodique sur un réseau Λ, et des réels $r > 0$ et $\epsilon > 0$ tels que $\forall x, y \in \mathbf{R}^m, \forall v \in \Lambda, (\|(x-y) - \lambda\| \geq r) \implies (\langle (fx) | f(y) \rangle \leq \epsilon)$ [7]. On dit alors que f cache Λ. Il s'agit bien de notre cadre de travail, en généralisant sensiblement $\mathbf{R}^m$ à $\mathbf{G}$ et en se posant la question, éludée ici, de la représentation informatique des réseaux.
La résolution de ce nouveau type de problème du sous-groupe caché ressemble à nouveau à l'algorithme de Shor. On note $\rho_s : x \mapsto e^{-\frac{1}{2}s^2\pi\|x\|^2}$ une gaussienne et $\mathbf{D}_m = \frac{1}{q}\mathbf{Z}^m / \mathbf{Z}^m$ un ensemble, où les paramètres $s > 0$ et $q \in \mathbf{N}$ servent à affiner la complexité. On initialise un premier état sur lequel on applique l'oracle O_f, pour conclure par une transformée de Fourier

6. En pratique, la manière de représenter l'ensemble des réseaux de $\mathbf{R}^{m+n-1}$ sur un ordinateur n'est pas claire. Il faut se munir d'une base et en tirer une application g associant aux réseaux des états quantiques. On s'intéresse alors à la composée $g \circ f$, et pas à f directement.

7. Cette dernière hypothèse doit être comprise comme une généralisation de l'injectivité sur une période dans le cas fini.

réciproque $\mathscr{F}^\dagger$, définie au sens des caractères précédemment.

$$\sum_{x\in\mathbf{D}^m}\sqrt{\rho_s(x)}\,|x\rangle\,|0\rangle \longmapsto \sum_{x\in\mathbf{D}^m}\sqrt{\rho_s(x)}\,|x\rangle\,\big|f(x)\big\rangle \longmapsto \sum_{x\in\mathbf{D}_m}\sum_{\chi\in\widehat{\mathbf{D}}_m}\sqrt{\rho_s(x)}e^{-i2\pi\chi(x)}\,\big|\chi\big\rangle\,\big|f(x)\big\rangle$$

Mesurer un nombre suffisant de fois permet finalement d'obtenir un élément χ du réseau dual $\Lambda^\star$, c'est-à-dire un caractère tel que $\forall x\in\Lambda, \chi(x)\in\mathbf{Z}$.

L'informatique quantique, radicalement différente de l'informatique classique, permet la résolution de problèmes difficiles à la base de la cryptographie. Bien qu'à l'heure actuelle, le nombre de qubits utilisables simultanément est trop faible pour présenter une quelconque menace, la stabilisation d'un nombre toujours plus grands d'entre eux et l'optimisation de la complexité spatiale des algorithmes rend de plus en plus probable l'avènement d'une *suprématie quantique*[8]. En prévision, de nouveaux protocoles de cryptographie *post-quantiques*, c'est-à-dire censés résister à ce nouveau genre de machines, ont été choisis pour une standardisation d'ici 2024.

Références

[1] Philipp Kammerlander. *Quantum Information Processing.* ETH Zurich, February 2022.

[2] Yves Leroyer and Géraud Sénizergues. *Introduction à l'informatique quantique.* ETH ENSEIRB, 2017.

[3] Richard J Lipton and Kenneth W Regan. *Introduction to quantum algorithms via linear algebra.* MIT Press, 2021.

[4] Miklos Santha. Algorithmes quantiques. *Collège de France*, 2021.

[5] Frédéric Magniez. Le problème du sous-groupe caché. *Collège de France*, 2021.

[6] Ronald de Wolf. Quantum computing : Lecture notes, 2019.

[7] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information.* American Association of Physics Teachers, 2002.

[8] Andrew M Childs. Lecture notes on quantum algorithms. *Lecture notes at University of Maryland*, 2017.

[9] Andrew M Childs and Wim Van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1) :23–31, 2010.

[10] Frédéric Wang. The hidden subgroup problem, 2010.

[11] Koen de Boer, Léo Ducas, and Serge Fehr. On the quantum complexity of the continuous hidden subgroup problem. In *Advances in Cryptology–EUROCRYPT 2020 : 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II*, pages 341–370. Springer, 2020.

[12] Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song. A quantum algorithm for computing the unit group of an arbitrary degree number field. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 293–302, 2014.

[13] Vlatko Vedral, Adriano Barenco, and Artur Ekert. Quantum networks for elementary arithmetic operations. *Physical Review A*, 54(1) :147, 1996.

8. L'expression de suprématie quantique désigne le moment critique à partir duquel les ordinateurs quantiques seront suffisamment puissants pour qu'aucun ordinateur classique ne puisse en simuler le fonctionnement en temps raisonnable.

Annexe.

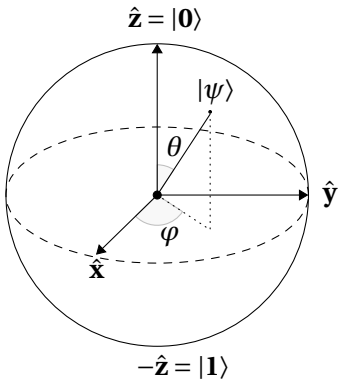
On donne les idées importantes de la démonstration du théorème 2, moins fastidieux à comprendre qu'à rédiger, et on démontre complètement les théorèmes 3 et 6. On admet les théorèmes 7, présenté par Yves Leroyer dans l'exercice 26 section 4.6 de [2], et 12, évoqué dans [12].
On détaille brièvement le code de la partie quantique de l'algorithme de Shor dans le cas $N = 21$, écrit en Python avec le module Qiskit. On présente enfin la construction de la multiplication modulaire en guise d'illustration de l'implémentation des opérations élémentaires sur un ordinateur quantique.

Preuve du théorème 2.

Théorème 2. *L'ensemble* {CNOT, H, T} *est universel.*

Pour montrer un tel résultat, il est d'abord nécessaire de s'intéresser précisément au cas d'un unique qubit isolé. Les postulats de la mécanique quantique indique que tout état d'un qubit est normalisé et ne dépend pas du facteur de phase global. Chaque état s'écrit donc $\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$ avec d'uniques $\theta \in [0,\pi]$ et $\phi \in [0,2\pi[$. On établit ainsi une correspondance bijective entre les états d'un qubit isolé et les points de la sphère $\mathbf{R}^3$ par

$$\big|\psi\big\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \longmapsto \big(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta\big)\ .$$



Cette représentation des états, appelée *sphère de Bloch*, permet de décrire facilement les transformations linéaires unitaires sur un qubit comme des rotations directes sur cette sphère. Par exemple, la porte T agit comme la rotation de $\frac{\pi}{4}$ radians orientée par $\hat{z}$ tandis que la porte HTH correspond à la rotation de $\frac{\pi}{4}$ radians autour de $\hat{x}$.

Lemme. *L'ensemble* $\{H, T\}$ *est dense dans l'ensemble des transformations linéaires unitaires sur un qubit.*

Démonstration. La composition des deux rotations T et HTH correspond à la rotation $R_{\hat{n}}(\theta)$ autour de $\hat{n} = \big(\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8}\big)$ d'un angle θ tel que $\cos\left(\frac{\theta}{2}\right) = \cos^2\left(\frac{\pi}{8}\right)$. Un tel angle θ est incommensurable avec 2π, ce qui entraîne que $\{H, T\}$ est dense dans les rotations autour de

$\hat{n}$. On constate alors que $HR_{\hat{n}}(\theta)H = R_{\hat{m}}(\theta)$ avec $\hat{m} = \left(\cos\frac{\pi}{8}, -\sin\frac{\pi}{8}, \cos\frac{\pi}{8}\right)$. Or $\hat{m}$ n'est pas colinéaire avec $\hat{n}$, ce qui permet de retrouver toute rotation de la sphère de Bloch à l'aide de rotations autour de $\hat{n}$ et $\hat{m}$, d'où la densité de $\{H, T\}$. □

L'ensemble $\{H, T\}$ ne permet que d'approcher des circuits agissant qubit par qubit. L'idée consiste à utiliser la porte CNOT pour obtenir les relations d'interdépendances entre qubits.

Lemme. *L'ensemble des transformations linéaires unitaires sur un qubit muni de la porte* CNOT *est dense dans l'ensemble des transformations linéaires unitaires sur des qubits (quel que soit leur nombre).*

Théorème 2. *L'ensemble* {CNOT, H, T} *est dense dans l'ensemble des transformations linéaires unitaires sur des qubits (quel que soit leur nombre).*

Démonstration. Par conjonction des deux lemmes précédents. □

Preuve du théorème 3.

Théorème 3. *Si N n'est pas pair ou une puissance d'un premier, alors la probabilité pour que $k \in (\mathbf{Z}/N\mathbf{Z})^\times$ soit d'ordre r pair et que N ne divise ni $\left(k^{\frac{r}{2}} - 1\right)$ ni $\left(k^{\frac{r}{2}} + 1\right)$ est d'au moins $\frac{1}{2}$.*

Étant donné un entier naturel $N \in \mathbf{N}^\times$, on note :

— $R_N = \left\{k \in (\mathbf{Z}/N\mathbf{Z})^\times \,|\, k^2 = 1\right\}$ l'ensemble des racines carrées de l'unité ;

— $P_N = \left\{k \in (\mathbf{Z}/N\mathbf{Z})^\times \,|\, \operatorname{ord} k \equiv 0 \pmod 2\right\}$ l'ensemble des éléments d'ordre pair ;

— $F_N = \left\{k \in P_N \,|\, k^{\frac{\operatorname{ord} k}{2}} \neq -1\right\}$ l'ensemble des éléments convenables pour la factorisation.

Il s'agit de montrer que le cas *favorable*, caractérisé par F_N, est également le cas *probable*. Pour se faire, on utilisera librement le théorème des restes chinois et le théorème de structure des unités ci-dessous.

Théorème 13 (des restes chinois)**.** *Pour p et q premiers entre eux,*

$$\left(\mathbf{Z}\Big/pq\mathbf{Z}\right)^\times \cong \left(\mathbf{Z}\Big/p\mathbf{Z}\right)^\times \times \left(\mathbf{Z}\Big/q\mathbf{Z}\right)^\times .$$

Théorème 14 (de structure des unités)**.** *Pour $p \geq 3$ et $\nu \geq 1$,*

$$\left(\mathbf{Z}\Big/p^\nu\mathbf{Z}\right)^\times \cong \left(\mathbf{Z}\Big/(p-1)\mathbf{Z}\right) \times \left(\mathbf{Z}\Big/p^{\nu-1}\mathbf{Z}\right) .$$

Pour commencer, on démontre deux lemmes à propos des cardinaux des ensembles intermédiaires considérés plus haut.

Lemme. *Pour $N = \prod_{i=1}^{n} p_i^{\nu_i}$, on a* $\dfrac{\operatorname{card} P_N}{\operatorname{card}(\mathbf{Z}/N\mathbf{Z})^\times} \geq 1 - \dfrac{1}{2^n}$.

Démonstration. Par le lemme des restes chinois, on a $(\mathbf{Z}/N\mathbf{Z})^\times \cong \prod_{i=1}^{n} \left(\mathbf{Z}/p_i^{\nu_i}\mathbf{Z}\right)^\times$. Un élément $k = (k_i)$ vérifie $\operatorname{ord} k = \operatorname{ppcm}(\operatorname{ord} k_i)$. En particulier, k est d'ordre pair si et seulement l'un des (k_i) est d'ordre pair. Ainsi, la proportion des éléments k d'ordre impair est inférieur à $\frac{1}{2^n}$. En effet, si $p_i = 2$, alors tout élément de $\left(\mathbf{Z}/p_i^{\nu_i}\mathbf{Z}\right)^\times$ est d'ordre pair et, sinon,

$$\left(\mathbf{Z}/\,p_i^{\nu_i}\mathbf{Z}\right)^\times \cong \left(\mathbf{Z}/\,(p_i-1)\mathbf{Z}\right) \times \left(\mathbf{Z}/\,p_i^{\nu_i-1}\mathbf{Z}\right),$$

où tout élément de la forme $(2x+1, y)$ est d'ordre pair, car si $r\cdot(2x+1,y)=0$, alors $p-1$ divise $r(2x+1)$, et donc 2 divise r par lemme de Gauss. □

Lemme. *Pour $N=\prod_{i=1}^{n} p_i^{\nu_i}$, si $n\geq 1$ et $(p_i)\geq 3$, alors, pour tout entier $r\geq 1$,*

$$\frac{\operatorname{card}\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times \,|\, k^r\in R_N\setminus\{\pm 1\}\right\}}{\operatorname{card}\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times \,|\, k^r\in R_N\setminus\{1\}\right\}} \geq \frac{2^n-2}{2^n-1}\,.$$

Démonstration. On fixe $r\geq 1$ et on suppose qu'il existe $k\in(\mathbf{Z}/\,N\mathbf{Z})^\times$ tel que $k^r=-1$, sans quoi l'énoncé est trivial. Par le théorème des restes chinois et le théorème de structure des unités, on sait que le groupe multiplicatif $(\mathbf{Z}/\,N\mathbf{Z})^\times$ est isomorphe au produit de groupes additifs $\prod_{i=1}^{n}\left(\mathbf{Z}/\,(p_i-1)\mathbf{Z}\right)\times\left(\mathbf{Z}/\,p_i^{\nu_i-1}\mathbf{Z}\right)$.
En conséquence, la classe de -1 s'écrit $\left(\frac{p_1-1}{2},0,\ldots,\frac{p_n-1}{2},0\right)$, et, plus généralement, les éléments de R_N sont de la forme $(\omega_1,0,\ldots,\omega_n,0)$, avec $r_i=\frac{p_i-1}{2}$ ou $r_i=0$. L'hypothèse initiale se traduit par l'existence de (k_i) tel que $(rk_1,0,\ldots,rk_n,0)=\left(\frac{p_1-1}{2},0,\ldots,\frac{p_n-1}{2},0\right)$, d'où on déduit que $\forall\omega\in R_N, \exists k_\omega\in(\mathbf{Z}/\,N\mathbf{Z})^\times, k_\omega^r=x$.
En considérant l'ensemble des racines r-ème de l'unité $U=\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times\,|\,k^r=1\right\}$, la bijection $k\mapsto k_\omega k$, pour $\omega\in R_N$, indique que $\operatorname{card}\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times\,|\,k^r=\omega\right\}=\operatorname{card}U$. Il en vient que

$$\frac{\operatorname{card}\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times \,|\, k^r\in R_N\setminus\{\pm 1\}\right\}}{\operatorname{card}\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times \,|\, k^r\in R_N\setminus\{1\}\right\}} = \frac{\operatorname{card}(R_N\setminus\{\pm 1\})\times\operatorname{card}U}{\operatorname{card}(R_N\setminus\{1\})\times\operatorname{card}U} = \frac{(2^n-2)}{(2^n-1)}\,.$$

□

Pour conclure, il suffit de reformuler le théorème 3 de sorte à pouvoir appliquer les lemmes précédents.

Théorème 3. *Pour $N=\prod_{i=1}^{n} p_i^{\nu_i}$, si $n\geq 2$ et $(p_i)\geq 3$, alors* $\dfrac{\operatorname{card}F_N}{\operatorname{card}(\mathbf{Z}/\,N\mathbf{Z})^\times}\geq\dfrac{1}{2}$.

Démonstration. D'une part, le premier lemme indique que $\dfrac{\operatorname{card}P_N}{\operatorname{card}(\mathbf{Z}/\,N\mathbf{Z})^\times}\geq 1-\dfrac{1}{2^n}\geq\dfrac{3}{4}$.
D'autre part, on dispose des deux décompositions

$$P_N=\bigcup_{r=1}^{N}\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times\,|\,k^r\in R_N\setminus\{1\}\right\} \qquad\text{et}\qquad F_N=\bigcup_{r=1}^{N}\left\{k\in(\mathbf{Z}/\,N\mathbf{Z})^\times\,|\,k^r\in R_N\setminus\{\pm 1\}\right\},$$

de sorte que, par le second lemme, $\dfrac{\operatorname{card}F_N}{\operatorname{card}P_N}\geq\dfrac{2^n-2}{2^n-1}\geq\dfrac{2}{3}$.
En conclusion,

$$\frac{\operatorname{card}F_N}{\operatorname{card}(\mathbf{Z}/\,N\mathbf{Z})^\times}=\frac{\operatorname{card}P_N}{\operatorname{card}(\mathbf{Z}/\,N\mathbf{Z})^\times}\times\frac{\operatorname{card}F_N}{\operatorname{card}P_N}\geq\frac{3}{4}\times\frac{2}{3}=\frac{1}{2}\,.$$

□

Preuve du théorème 6.

On se place à l'étape précédent la transformation de Fourier réciproque dans l'algorithme de Shor. L'état, après projection sur l'image y, subit la transformation

$$\sqrt{\frac{r}{2^n}} \sum_{x:f(x)=y} |x\rangle = \sqrt{\frac{r}{2^n}} \sum_{t=0}^{T} |x_0 + r\,t\rangle \longmapsto \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} \omega_n^{-y(x_0+tr)} \big|y\big\rangle$$

avec $T = \left\lfloor \frac{2^n}{r} \right\rfloor$. On cherche à démontrer le théorème 6 en étudier la probabilité de mesurer y dans l'état final $\mathbf{P}\left[\big|y\big\rangle\right] = \left| \frac{1}{2^n} \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} \omega_n^{-y(x_0+tr)} \right|^2 = \frac{1}{2^n K} \frac{\sin^2\left(K\frac{\pi y r}{2}\right)}{\sin^2\left(\frac{\pi y r}{2}\right)}$.

Lemme. *Si y est à distance au plus $\frac{1}{2}$ d'un multiple de $\frac{2^n}{r}$, alors $\mathbf{P}\left[\big|y\big\rangle\right] \geq \frac{4}{\pi^2}\frac{1}{r}$.*

Démonstration. D'après le lemme précédent, avec $k \in [\![0, r-1]\!]$ tel que $\left| y - k\frac{2^n}{r} \right| \leq \frac{1}{2}$, on a

$$\mathbf{P}\left[\big|y\big\rangle\right] = \frac{1}{2^n K} \frac{\sin^2\left(K\frac{\pi y r}{2}\right)}{\sin^2\left(\frac{\pi y r}{2}\right)} = \frac{1}{2^n K} \frac{\sin^2\left(K\frac{\pi y r}{2}\right)}{\sin^2\left(\frac{\pi y r}{2}\right)} = \frac{1}{2^n K} \frac{\sin^2\left(K\frac{\pi\left(y-k\frac{2^n}{r}\right)r}{2}\right)}{\sin^2\left(\frac{\pi\left(y-k\frac{2^n}{r}\right)r}{2}\right)}\,.$$

Or, par concavité, on sait que $\frac{2}{\pi}x \leq \sin x \leq x$ pour tout $x \in \left[0, \frac{\pi}{2}\right]$, d'où

$$\mathbf{P}\left[\big|y\big\rangle\right] \geq \frac{1}{2^n K} \frac{\frac{4}{\pi^2}\left(K\frac{\pi\left(y-k\frac{2^n}{r}\right)r}{2}\right)^2}{\left(\frac{\pi\left(y-k\frac{2^n}{r}\right)r}{2}\right)^2} = \frac{4}{\pi^2}\frac{K}{2^n} \geq \frac{4}{\pi^2}\frac{1}{r}\,.$$

□

Théorème 6. *Une mesure de l'état final est à une distance au plus $\frac{1}{2}$ d'un multiple de $\frac{2^n}{r}$ avec probabilité au moins $0,4$.*

Démonstration. En sommant sur chaque multiple possible de $\frac{2^n}{r}$ la probabilité de mesurer le voisin le plus proche et en appliquant l'inégalité du lemme ci-dessus, on trouve

$$\mathbf{P}\left[\big|y\big\rangle : \exists k \in [\![0, r-1]\!], \left| y - k\frac{2^n}{r} \right| \leq \frac{1}{2}\right] = \sum_{k=0}^{r-1} \mathbf{P}\left[\big|y\big\rangle : \left| y - k\frac{2^n}{r} \right| \leq \frac{1}{2}\right] \geq \sum_{k=0}^{r-1} \frac{4}{\pi^2}\frac{1}{r} = \frac{4}{\pi^2} \geq 0,4\,.$$

□

Implémentation de l'algorithme de Shor (pour $N = 21$).

On commence par importer, depuis le module Qiskit, l'ensemble des outils nécessaires à la simulation d'un algorithme quantique.

```
from numpy import pi

from qiskit import Aer, execute
from qiskit import QuantumCircuit
```

On initialise les qubits de travail dans une superposition équiprobable et on place les qubits auxiliares dans l'état $|1\rangle$ pour pouvoir implémenter l'oracle plus facilement.

```
def init_qubits(qc, n, m):
    qc.h(range(n))
    qc.x(n+m-1)
```

On décompose l'oracle comme une succession de multiplications modulaires par les puissances de 2 contrôlées par les qubits associés. La multiplication est définie par des opérations SWAP sur les qubits [9] pour les puissances de 2 et leurs opposés modulo 21.

```
def multiplication_mod21(x, y, n):
    if x not in [2,4,5,8,13,16,17,19]:
        raise ValueError("ni (+x mod 21) ni (-x mod 21) ne sont des
    puissances de 2")
    M = QuantumCircuit(n)
    for iteration in range(y):
        if x in [2,19]:
            for k in range(n-1, 0, -1):
                M.swap(k-1, k)
        if x in [4,17]:
            for iteration in range(2):
                for k in range(n-1, 0, -1):
                    M.swap(k - 1, k)
        if x in [8, 13]:
            for iteration in range(3):
                for k in range(n-1, 0, -1):
                    M.swap(k - 1, k)
        if x in [5,16]:
            for iteration in range(4):
                for k in range(n-1, 0, -1):
                    M.swap(k - 1, k)
        if x in [5,13,17,19]:
            for q in range(n):
                M.x(q)
    M = M.to_gate()
    M.name = f"M({x}x{y})mod21"
    CM = M.control()
    return CM

def exponentiation_mod21(qc, n, m, k):
    for i in range(n):
```

9. Cette restriction de l'opération à des puissances de 2 ne sert qu'à gagner du temps. La section suivante détaille l'implémentation complète d'une multiplication modulaire en général.

```
        qc.append(multiplication_mod21(k, 2**i, n),
              [i] + list(range(n, n+m)))
```

On implémente la transformée de Fourier réciproque, qu'on aurait également pu importer depuis Qiskit.

```
def qft_inverse(qc, n, m):
    inv_f = QuantumCircuit(n, 0)
    for k in range(n-1, -1, -1):
        inv_f.h(k)
        for l in range(k):
            inv_f.cp(-pi/2**(k-l), l, k)
    inv_f = inv_f.to_gate()
    inv_f.name = f"QFT inverse"
    qc.append(inv_f, list(range(0, n)))
```

Enfin, on en déduit le circuit général de l'algorithme de recherche de période de Shor.

```
def shor(n, m, k):
    qc = QuantumCircuit(n + m, n)

    init_qubits(qc, n, m)
    qc.barrier()

    exponentiation_mod21(qc, n, m, k)
    qc.barrier()

    qft_inverse(qc, n, m)
    qc.barrier()

    for i in range(n):
        qc.measure(i, i)
    return qc
```

On peut par exemple l'appliquer en choisissant $k = 5$ d'ordre $r = 6$.

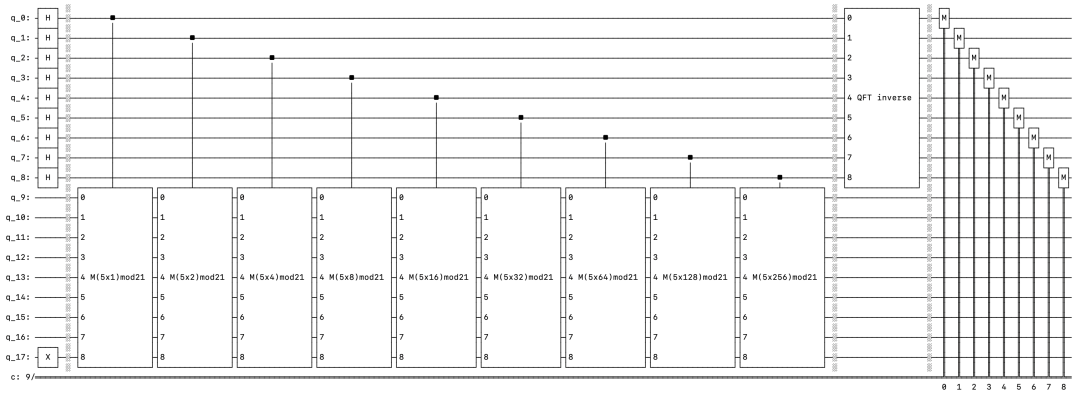
```
    n = 9; m = 9

    qc = recherche_de_periode(n, m, 5)
    qc.draw()

    simulator = Aer.get_backend('qasm_simulator')
    counts = execute(qc, backend=simulator).result().get_counts(qc)

    print(counts.int_outcomes())
```

On obtient alors différentes mesures très bruitées [10], mais contenant notamment l'entier 426 de manière significative, avec 40 occurrences.

```
{313: 47, 341: 36, 213: 8, 482: 7, 454: 54, 226: 46, 426: 40, 113: 29,
    368: 19, 171: 13, 369: 14, 157: 46, 398: 23, 399: 28, 112: 31, 79: 24,
     78: 23, 340: 44, 227: 44, 312: 55, 455: 55, 432: 8, 427: 42, 170: 11,
     1: 58, 156: 45, 0: 54}
```

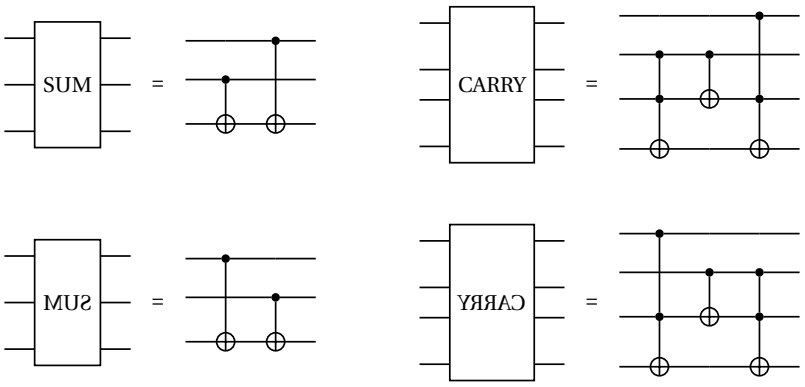
10. Pour raccourcir et rendre le résultat plus lisible, on a éliminé tous les entiers n'ayant été mesurés qu'au plus cinq fois, soit 79 entiers différents sur 106.

Qiskit fournit également un dessin du circuit implémenté.

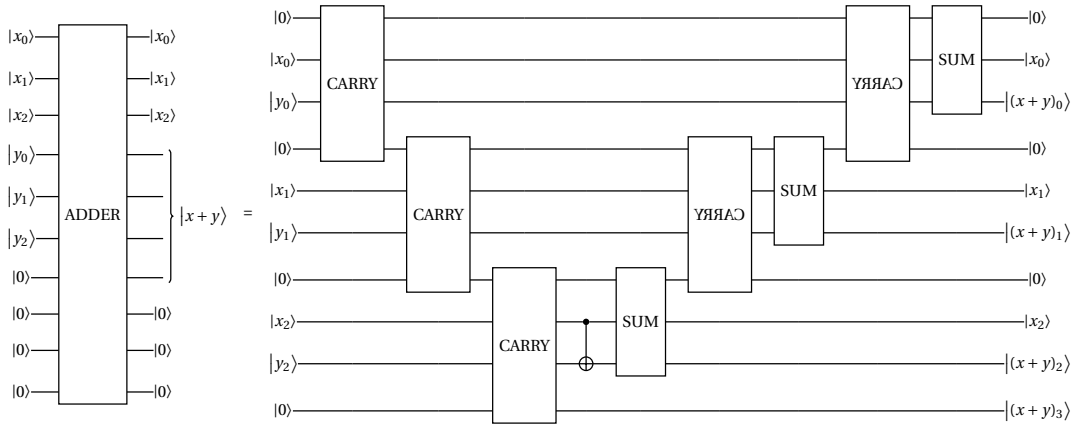


Implémentation de la multiplication modulaire (d'après [13]).

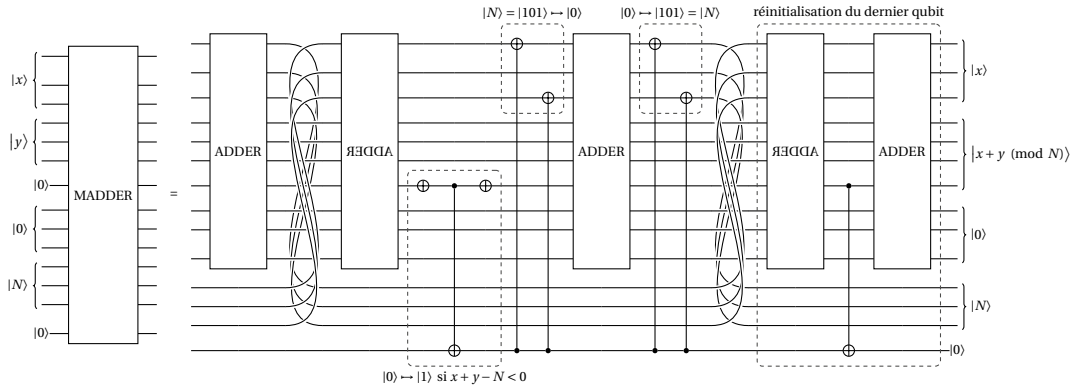
On implémente, à l'aide de portes CNOT, les opérations SUM et CARRY, ainsi que leurs miroirs. Il convient de voir NOT comme l'addition modulo 2 pour comprendre que SUM correspond à la somme de deux qubits sur un troisième et CARRY à l'enregistrement de la retenue de SUM sur un quatrième qubit.



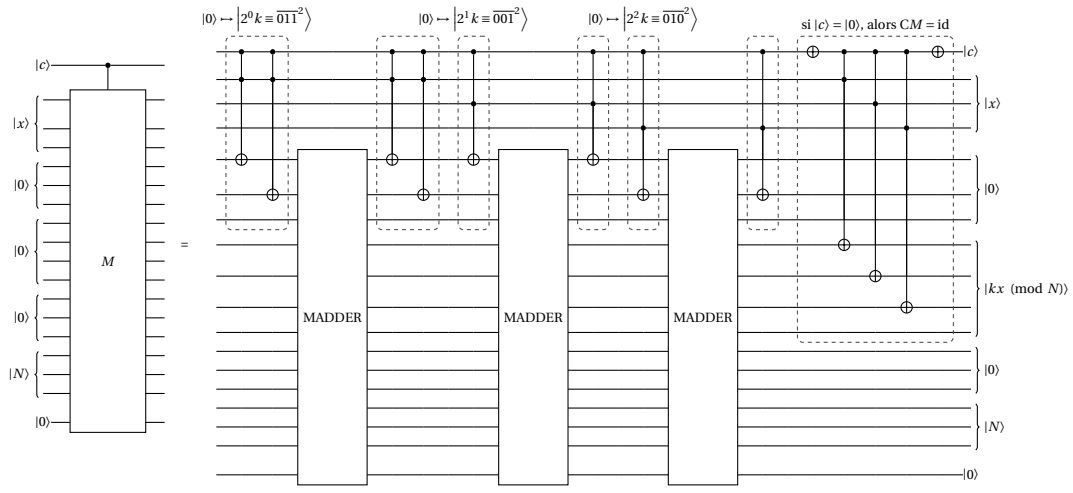
On construit alors l'opérateur ADDER réalisant l'addition de deux entiers de trois bits d'information. On commence par placer toutes les retenues sur des qubits auxiliaires initialement dans l'état $|0\rangle$, avant de sommer chaque bit des deux nombres et la retenue correspondante ensemble. Les opérations YRRAC permettent de rétablir les informations initiales avant d'appliquer SUM, informations initiales que les CARRY correspondants altèrent en présence d'une retenue. On peut naturellement prolonger ADDER à des entiers composés n bits d'information à l'aide de $O(n)$ opérations CARRY, SUM et YRRAC, et donc à l'aide de $O(n)$ portes CNOT.



Il s'agit désormais de construire l'addition modulaire MADDER. On note ЯЭDDA l'opérateur miroir de ADDER permettant de réaliser des soustractions. Il suffit d'ajouter les nombres x et y entre eux, retrancher N puis le rajouter si le résultat est devenu négatif. Notons que N doit être connu à l'avance pour positionner les portes CNOT autour de la seconde addition, de sorte à l'éviter en annulant N si le dernier qubit indique que le résultat n'est pas négatif. Ici on a choisit $N = \overline{101}^2 = 5$. Les derniers calculs ne servent qu'à réinitialiser le dernier qubit sur l'état $|0\rangle$. L'addition modulaire d'entiers écrits avec n qubits utilise ainsi $O(n)$ opérateurs ADDER et ЯЭDDA, donc $O\left(n^2\right)$ portes CNOT.



Finalement, on peut construire la porte CM correspondant à la multiplication par k modulo N contrôlée par un qubit $|c\rangle$, ces deux entiers étant déjà connues à l'avance. Ici on a choisit $N = \overline{101}^2 = 5$ et $k = \overline{011}^2 = 3$. On lit les qubits de l'entrée un à un, en ajoutant le résidu de $2^i k$ modulo N si $x_i = 1$ à chaque étape. Finalement, si le qubit de contrôle est dans l'état $|0\rangle$, alors on recopie $|x\rangle$ à la sortie, car C$M|0\rangle|x\rangle = |x\rangle$ par définition. On vérifie bien que la manipulation de nombres encodés en n qubits nécessitent des portes CM implémentées avec $O(n)$ additions modulaires, pour une complexité finale en $O\left(n^3\right)$.


$|0\rangle \mapsto \left|2^0 k \equiv \overline{011}^2\right\rangle$
$|0\rangle \mapsto \left|2^1 k \equiv \overline{001}^2\right\rangle$
$|0\rangle \mapsto \left|2^2 k \equiv \overline{010}^2\right\rangle$
si $|c\rangle = |0\rangle$, alors $CM = \mathrm{id}$
$|c\rangle$
$|x\rangle$
$|0\rangle$
$|0\rangle$
$|N\rangle$
$|0\rangle$
M
=
MADDER
MADDER
MADDER
$|c\rangle$
$|x\rangle$
$|0\rangle$
$|kx \pmod N\rangle$
$|0\rangle$
$|N\rangle$
$|0\rangle$