Autour de la transformation de Fourier quantique

Sylvain Perifel

Le 25 août 2002

Introduction

L'informatique quantique est une alternative à l'informatique classique utilisant les effets quantiques pour effectuer certains calculs plus efficacement. L'idée que la physique quantique pourrait accélérer certaines tâches est dûe à Richard Feynman au début des années 1980, mais l'essor de la discipline n'a débuté qu'en 1994 lorsque Peter Shor a trouvé un algorithme quantique polynomial pour décomposer un nombre en produits de nombres premiers ([Sho96]). Bien que personne n'ait montré que ce problème soit NP-complet, il est en général considéré difficile classiquement. Sa résolution quantique efficace a des conséquences en cryptographie notamment.

Le problème du sous-groupe caché est une généralisation de la factorisation. Sa résolution fournirait notamment un algorithme polynomial pour le problème de l'isomorphisme de graphes, dont on ne sait pas s'il est dans P (pour des machines de Turing classiques). Des algorithmes quantiques polynomiaux pour certaines instances du sous-groupe caché (par exemple lorsque le groupe est abélien, et pour certains groupes dihédraux, *cf* [FMSS02]) ont été trouvés ; ils s'appuient sur la transformation de Fourier quantique. Il s'agit d'effectuer la transformation de Fourier discrète sur un groupe fini, ce qu'on sait faire quantiquement en temps logarithmique en fonction du cardinal du groupe, lorsque celui-ci est abélien.

Nous introduisons dans un premier temps la notion de calcul quantique, puis verrons comment effectuer la transformation de Fourier quantique sur un groupe abélien fini. Nous étudierons ensuite le problème du sous-groupe caché pour des groupes abéliens. Notre parcours nous amènera à nous demander comment réaliser efficacement la transformation de Fourier quantique exacte, et nous nous intéresserons également à la façon de trouver la période d'une fonction que l'on ne peut pas prolonger. Nous ne nous intéressons pas dans ce rapport à la mise en œuvre physique d'un ordinateur quantique : c'est une tâche très difficile et nous nous contentons d'examiner la théorie du calcul quantique.

1 Modèle de calcul quantique

1.1 Le qubit

Définition 1.1 (Norme)
Soit U un $\mathbb{C}$-espace vectoriel de dimension finie d_U. On travaillera avec la norme euclidienne sur cet espace : soit $(\mathbf{u_1}, \ldots, \mathbf{u_{d_U}})$ une base orthonormée. La norme du vecteur $\mathbf{u} = \sum_{i=1}^{d_U} \alpha_i \mathbf{u_i}$ est alors

$$\| \mathbf{u} \| = \sum_{i=1}^{d_U} |\alpha_i|^2$$

On rappelle que le produit scalaire de $\mathbf{u} = \sum_{i=1}^{d_U} \alpha_i \mathbf{u_i}$ et $\mathbf{v} = \sum_{i=1}^{d_U} \beta_i \mathbf{u_i}$ est

$$\langle \mathbf{u} | \mathbf{v} \rangle = \sum_{i=1}^{d_U} \overline{\alpha_i} \beta_i$$

où, pour $x \in \mathbb{C}$, $\overline{x}$ est le conjugué de x.

Définition 1.2 (Qubit)
On se fixe une base orthonormée de $\mathbb{C}^2$, dont les deux vecteurs unitaires sont notés $|0\rangle$ et $|1\rangle$.[1] L'espace vectoriel $\mathbb{C}^2$ est appelé *registre de taille 1*. Un registre de taille 1 a "une valeur" : un qubit. Un *qubit* est un

[1] Cette notation est dûe à Paul Dirac.

vecteur unitaire de $\mathbb{C}^2$. Il s'écrit donc sous la forme

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \text{ avec } \alpha,\beta \in \mathbb{C} \text{ et } |\alpha|^2 + |\beta|^2 = 1$$

On dit alors que le registre contient le qubit $|\psi\rangle$.

Exemple $|0\rangle$, $|1\rangle$, $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ sont des qubits, tous dans des registres de taille 1.

1.2 Produit tensoriel, registre

Définition 1.3 (Produit tensoriel)
Soient U et V deux $\mathbb{C}$-espaces vectoriels de dimensions respectives d_U et d_V. Soient deux bases fixées $(\mathbf{u_1},\dots,\mathbf{u_{d_U}})$ et $(\mathbf{v_1},\dots,\mathbf{v_{d_V}})$ de U et V respectivement. Soit W un $\mathbb{C}$-espace vectoriel de dimension $d_W = d_U \times d_V$, avec une base fixée $(\mathbf{w_1},\dots,\mathbf{w_{d_W}})$. Le *produit tensoriel* $\otimes$ est une application $U \times V \longrightarrow W$. Elle est définie sur les vecteurs de base par $\mathbf{u_1}\otimes\mathbf{v_1} = \mathbf{w_1}$, $\mathbf{u_1}\otimes\mathbf{v_2} = \mathbf{w_2}$, …, $\mathbf{u_1}\otimes\mathbf{v_{d_V}} = \mathbf{w_{d_V}}$, $\mathbf{u_2}\otimes\mathbf{v_1} = \mathbf{w_{d_V+1}}$, …, $\mathbf{u_{d_U}}\otimes\mathbf{v_{d_V}} = \mathbf{w_{d_W}}$. Le produit tensoriel sur des vecteurs quelconques est défini par distributivité :

$$(\alpha_1\mathbf{u_1} + \cdots + \alpha_{d_U}\mathbf{u_{d_U}}) \otimes (\beta_1\mathbf{v_1} + \cdots + \beta_{d_V}\mathbf{v_{d_V}}) = \alpha_1\beta_1\mathbf{u_1}\otimes\mathbf{v_1} + \alpha_1\beta_2\mathbf{u_1}\otimes\mathbf{v_2} + \cdots + \alpha_{d_U}\beta_{d_V}\mathbf{u_{d_U}}\otimes\mathbf{v_{d_V}}$$

On voit ainsi que le produit tensoriel dépend des bases que l'on fixe. On note $W = U \otimes V$ et on dit que W est le produit tensoriel de U et V.

Définition 1.4 (Registre)
Un *registre* de taille n est le produit tensoriel de n registres de taille 1. Rappelons que l'on note $|0\rangle$ et $|1\rangle$ les vecteurs de la base du registre de taille 1. Ainsi, en effectuant le produit tensoriel n fois, on obtient 2^n vecteurs de base

$$(|0\rangle\,|0\rangle \cdots |0\rangle) \quad (|0\rangle\,|0\rangle \cdots |1\rangle) \quad \dots \quad (|1\rangle\,|1\rangle \cdots |1\rangle)$$ [2]

On écrit également, pour simplifier, $|00\cdots0\rangle$, …, $|11\cdots1\rangle$. Il s'agit toujours d'un registre de taille n, produit tensoriel de n registres de taille 1. Lorsque l'on connaît la taille du registre, on peut utiliser la représentation décimale des nombres binaires ainsi formés, et on note alors, de manière condensée

$$|0\rangle, |1\rangle, |2\rangle, \dots, |2^n - 1\rangle$$

Un registre de taille n n'est autre qu'un $\mathbb{C}$-espace vectoriel de dimension 2^n, dont une base orthonormée est $(|0\rangle, \dots, |2^n - 1\rangle)$. Parfois on précisera la taille d'un registre en indice pour lever toute ambiguité : $|i\rangle_n$ signifie que l'on travaille dans un registre de taille n.

Définition 1.5 (n-qubit)
Un *n-qubit* est un vecteur unitaire d'un registre de taille n. Ainsi, un n-qubit s'écrit sous la forme

$$\sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle \text{ avec } \forall i, \alpha_i \in \mathbb{C} \text{ et } \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

Exemple Pour tout $0 \leqslant i < 2^n$, $|i\rangle$ est un n-qubit. De même,

$$\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$$

est un n-qubit, appelé superposition uniforme d'états.

Remarque Un registre de taille n est le produit tensoriel de n registres de taille 1 ; cependant, un n-qubit n'est pas toujours le produit tensoriel de n qubits. En effet, lorsque $n = 2$ par exemple, le 2-qubit $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ n'est pas le produit tensoriel de deux qubits, car $(\alpha\,|0\rangle + \beta\,|1\rangle) \otimes (\gamma\,|0\rangle + \delta\,|1\rangle) = \alpha\gamma\,|00\rangle + \alpha\delta\,|01\rangle + \beta\gamma\,|10\rangle + \beta\delta\,|11\rangle$ ce qui ne peut pas être égal à $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Lorsque l'on ne peut pas exprimer un n-qubit comme produit tensoriel de n qubits, on dit que le n-qubit est *enchevêtré*.

[2] On omet souvent le symbole $\otimes$ lorsqu'il n'y a pas ambiguité : on écrit alors $|x\rangle\,|y\rangle$ au lieu de $|x\rangle \otimes |y\rangle$.

1.3 Opérations sur les qubits

Définition 1.6 (Unitaire)
Soit U une transformation linéaire. Si l'on note U^* l'adjoint de U, alors U unitaire $\Longleftrightarrow UU^* = \mathrm{id}$.

Une opération quantique doit être unitaire, *i.e.* elle doit être linéaire et préserver la norme (un vecteur unitaire est envoyé sur un vecteur unitaire). Ainsi elle est bijective. Le caractère unitaire des opérations quantiques nous assure de ne travailler qu'avec des vecteurs unitaires.
Remarque Le caractère unitaire des opérations quantiques est imposé par la mécanique quantique (il n'a pas été introduit seulement pour la commodité de l'informaticien), lorsque le système est isolé.

Exemple La transformation d'Hadamard H, sur un qubit, est la transformation unitaire suivante :

$$\begin{array}{lcl} |0\rangle & \mapsto & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle & \mapsto & \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array}$$

Donc dans la base $(|0\rangle, |1\rangle)$, elle a pour matrice

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

Considérons maintenant un registre de taille n, initialement dans l'état $|0\rangle$, et appliquons la transformation d'Hadamard à chaque sous-registre de taille 1 :

$$|0\rangle \xrightarrow{H_n} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$$

Nous avons obtenu la superposition uniforme en n opérations.

Exemple L'opération "Non-contrôlé", notée CNot, agit sur un registre de taille 2, et a pour matrice dans la base $(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$:

$$\mathrm{CNot} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Cette transformation inverse le second qubit si et seulement si le premier est à 1.

Partant de $|00\rangle$, et appliquant la transformation d'Hadamard au premier sous-registre de taille 1 (le "premier 0"), on obtient :

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

De là, si on applique le CNot, on a :

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |3\rangle)$$

ce qui correspond à l'état enchevêtré dans la remarque du paragraphe 1.2. Ainsi, la transformation CNot peut créer de l'enchevêtrement.

En informatique quantique, nous sommes restreints aux applications unitaires. Cependant, nous aimerions parfois appliquer une fonction qui n'est pas unitaire, ni même bijective.

Proposition 1.1 *Soit $f : \{0, 1, \ldots, 2^n - 1\} \longrightarrow \{0, 1, \ldots, 2^m - 1\}$ une fonction quelconque. L'application*

$$U_f : |x\rangle_n\, |y\rangle_m \mapsto |x\rangle_n\, |y \oplus f(x)\rangle_m$$

où $\oplus$ désigne l'addition dans $\mathbb{Z}_{2^m}$, est unitaire (nous ne définissons U_f que sur les vecteurs de base, et la complétons par linéarité).

Preuve
Rappelons que, conformément à notre notation, $|y \oplus f(x)\rangle$ est le $(y \oplus f(x))$-ème vecteur de base. Une application unitaire est exactement une application qui envoie deux vecteurs unitaires orthogonaux sur deux vecteurs unitaires orthogonaux. Pour U_f, il suffit de montrer ce résultat sur les vecteurs de base du produit

tensoriel des deux registres ; on pourra conclure par linéarité. Soient deux vecteurs orthogonaux $|x\rangle\,|y\rangle$ et $|x'\rangle\,|y'\rangle$. Si $|x\rangle \neq |x'\rangle$, alors $|x\rangle\,|y \oplus f(x)\rangle$ et $|x'\rangle\,|y' \oplus f(x')\rangle$ sont orthogonaux (ce sont deux vecteurs de base distincts dans le produit tensoriel des deux registres). On peut donc considérer $x = x'$ (et $y \neq y'$, sinon les deux vecteurs seraient égaux). Il suffit donc de montrer que $y \oplus f(x) \neq y' \oplus f(x)$, ce qui est vrai car $y \neq y'$. $\square$

Remarque Lorsque l'on sait effectuer f efficacement d'une manière classique, alors on peut également effectuer U_f efficacement en quantique (*cf* [RP00]). Souvent, la transformation U_f nous sera donnée sous la forme d'une boîte noire, à laquelle nous fournirons une entrée $|x\rangle\,|y\rangle$. Elle nous donnera le résultat $|x\rangle\,|y \oplus f(x)\rangle$. Notre but sera de trouver des propriétés de f lorsque l'on a accès à f seulement *via* la boîte noire.

Exemple Soit une fonction $f : \{0, 1, \ldots, 2^n - 1\} \longrightarrow \{0, 1, \ldots, 2^m - 1\}$. On peut alors appliquer U_f à un vecteur de base :

$$U_f(|x\rangle\,|0\rangle) = |x\rangle\,|f(x)\rangle$$

Mais on peut aussi appliquer U_f à une superposition d'états : par exemple, pour la superposition uniforme, on obtient (par linéarité)

$$U_f\left(\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle\,|0\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle\,|f(i)\rangle$$

Ainsi, on a calculé f en 2^n points en un seul appel à la boîte noire U_f.

Mais que peut-on faire de ce résultat ? Comment extraire l'information que contient ce registre ?

1.4 Mesure

La mesure est une opération particulière en physique quantique. C'est la seule transformation qui ne soit pas unitaire : on perd de l'information (la mesure n'est pas réversible). La mécanique quantique autorise une opération non unitaire car le système n'est plus isolé : on intéragit avec lui pour le mesurer.

Définition 1.7 (Mesure)
Soit $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle$ un n-qubit. Lorsque l'on mesure le registre contenant $|\psi\rangle$, on obtient une valeur classique entre 0 et $2^n - 1$. Si l'on obtient la valeur i_0, alors le registre est modifié et prend la valeur $|i_0\rangle$. Le registre a subi une projection (et une renormalisation), et nous avons récupéré une valeur classique. La valeur i_0 est obtenue avec la probabilité $|\alpha_{i_0}|^2$.

Exemple Si l'on mesure un registre contenant $|0\rangle$, on obtient toujours la valeur 0 (avec probabilité 1), et le registre reste à $|0\rangle$. Si l'on mesure $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$, alors on obtient i_0 quelconque avec probabilité uniforme, et le registre contient maintenant $|i_0\rangle$.

On peut également effectuer une mesure partielle d'un registre, en décidant de ne mesurer qu'un sous-registre.

Définition 1.8 (Mesure partielle)
Soit un registre de taille n, et soit $m < n$. Le produit tensoriel étant associatif, ce registre est le produit tensoriel d'un registre de taille m et d'un regitre de taille $n - m$. Nous voulons mesurer seulement le premier registre de taille m.

Lors de la mesure de ce sous-registre, on obtient une valeur classique comprise entre 0 et $2^m - 1$. Le registre est alors projeté (puis normalisé) sur la valeur i_0 trouvée : il contient $|i_0\rangle_m$. Le registre complet de taille n subit en fait une projection orthogonale sur l'espace $(|i_0\rangle_m \otimes \mathbb{C}^{n-m})$. L'autre sous-registre de taille $n - m$ peut en être modifié (si les deux registres sont enchevêtrés) : il est projeté sur tous les vecteurs dont la restriction au premier registre était $|i_0\rangle_m$ (on ne "conserve" que les vecteurs "commençant" par i_0). La probabilité de mesurer i_0 est la somme des modules au carré des coefficients des vecteurs "commençant" par i_0.

Exemple Soit le registre de taille 3 contenant $\frac{1}{\sqrt{7}}(|0\rangle + 2\,|3\rangle - |6\rangle + |7\rangle) = \frac{1}{\sqrt{7}}(|000\rangle + 2\,|011\rangle - |110\rangle + |111\rangle)$. Supposons que l'on mesure le premier sous-registre de taille 1 (correspondant au bit de poids fort : le premier chiffre dans l'écriture des vecteurs de base, par exemple 0 dans $|011\rangle$). Nous obtenons alors le résultat 0 avec probabilité $(\frac{1}{\sqrt{7}})^2 + (\frac{2}{\sqrt{7}})^2 = \frac{5}{7}$, et le registre est projeté dans l'état $\frac{1}{\sqrt{5}}(|000\rangle + 2\,|011\rangle)$ (on a "gardé" seulement les vecteurs "commençant" par 0). De même, nous obtenons le résultat 1 avec probabilité $(-\frac{1}{\sqrt{7}})^2 + (\frac{1}{\sqrt{7}})^2 = \frac{2}{7}$, et le registre est projeté dans l'état $\frac{1}{\sqrt{2}}(-\,|110\rangle + |111\rangle)$.

Exemple Soit $f : \{0,1,\dots,2^n-1\} \longrightarrow \{0,1,\dots,2^m-1\}$ une fonction T-périodique, avec T divisant 2^n. Supposons de plus que f est injective sur $\{0,\dots,T-1\}$. Appliquons U_f à la superposition uniforme :

$$U_f(\frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}|i\rangle|0\rangle_m) = \frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}|i\rangle|f(i)\rangle = \frac{1}{\sqrt{2^n}}\sum_{i=0}^{T-1}\sum_{j=0}^{\frac{2^n}{T}-1}|i+jT\rangle|f(i)\rangle$$

Maintenant, si nous mesurons le second registre, nous obtenons $f(i_0)$ pour un certain i_0 (avec probabilité uniforme). De plus, l'état du registre est projeté sur

$$\sqrt{\frac{T}{2^n}}\sum_{j=0}^{\frac{2^n}{T}-1}|i_0+jT\rangle$$

Remarque Ceux qui connaissent le modèle de calcul probabiliste se demandent peut-être d'où vient la puissance quantique. La différence fondamentale entre les deux modèles est la possibilité d'utiliser des nombres négatifs en quantique. En effet, les calculs peuvent ainsi créer des interférences destructives, alors qu'elles ne sont que constructives dans le cas probabiliste. Nous verrons par exemple que cela est nécessaire pour la transformation de Fourier quantique.

En outre, une autre différence majeure est le rôle de la mesure. Dans le modèle quantique, la mesure ne commute pas avec les autres opérations unitaires, donc on ne peut pas faire la mesure à n'importe quel moment de l'algorithme. Dans le cas probabiliste, en revanche, la mesure ne perturbe pas le calcul.

1.5 Circuits quantiques

Définition 1.9 (Porte quantique)
Une porte quantique est une opération unitaire, ou une opération de mesure, sur un registre de taille n quelconque. Physiquement, on peut réaliser plus ou moins facilement une telle porte agissant sur de vrais qubits.

Exemple La transformation d'Hadamard est une porte quantique agissant sur un registre de taille 1 ; la transformation CNot est une porte quantique agissant sur un registre de taille 2.

Définition 1.10 (Portes universelles)
- Un ensemble U de portes quantiques est *universel* pour le calcul quantique si toute opération unitaire peut être obtenue en composant des portes de U uniquement, *i.e.* la clôture par composition finie de U est exactement l'ensemble des transformations unitaires.
- Un ensemble U' de portes quantiques est *universel approché* pour le calcul quantique si toute opération unitaire peut être approchée aussi près que l'on veut en composant des portes de U' uniquement, *i.e.* l'adhérence de la clôture par composition finie de U' est exactement l'ensemble des transformations unitaires.

On peut trouver la démonstration de la proposition suivante dans les notes de cours de John Preskill, [Pre98].

Proposition 1.2 – *Soit U l'ensemble des portes quantiques agissant sur des registres de taille 1. $U \cup \{CNot\}$ est universel pour le calcul quantique.*
– *Soit $U' = \{H, A, CNot\}$ l'ensemble des portes de Hadamard, CNot et*

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

U' est universel approché pour le calcul quantique.

Définition 1.11 (Circuit)
Se donner un ensemble universel (approché ou non) de portes quantiques de base revient à se donner théoriquement un ordinateur quantique. Un algorithme quantique est alors décrit comme la succession de portes de base. On appelle cette succession de portes de base un *circuit quantique*.

2 Transformation de Fourier quantique

2.1 Motivation

La transformation de Fourier quantique joue un grand rôle en informatique quantique, car elle est utilisée dans de nombreux algorithmes connus. Son importance vient du fait qu'on peut la réaliser très efficacement : en temps polynomial en $\log N$, alors que la FFT (Fast Fourier Transform) classique nécessite $\mathcal{O}(N \log N)$ opérations.

Dans son algorithme de factorisation des nombres, Shor se ramène à trouver la période d'une fonction. Plus généralement, le problème du sous-groupe caché est lui aussi un problème de période. Et c'est là l'atout de la transformation de Fourier quantique. Lorsque l'on a une boîte noire calculant une fonction périodique, l'appel à cette boîte noire puis l'utilisation de la transformation de Fourier quantique amène le registre sur une valeur présentant un nombre de pics égal à la période, comme sur la figure 1. Ainsi, on peut retrouver la période en mesurant le registre.

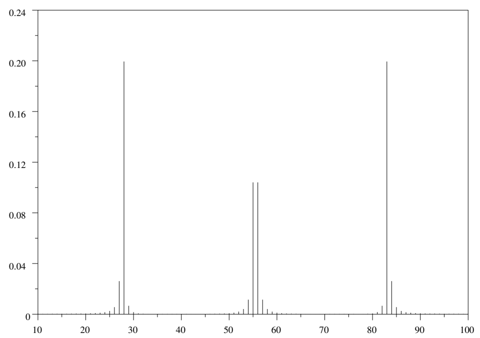

FIG. 1 – Exemple de transformation de Fourier discrète

2.2 Transformation de Fourier discrète sur groupe abélien fini

Dans toute la suite, $(G, +)$ représente un groupe abélien fini de cardinal N, on note 0 l'élément neutre de G. On note $\mathbb{C}^*$ l'ensemble des nombres complexes non nuls. $(\mathbb{C}^*, \cdot)$ forme un groupe pour le produit usuel sur les nombres complexes. On note $\mathbb{Z}_N$ le groupe des entiers modulo N, muni de l'addition.

La transformation de Fourier discrète utilise la notion de caractère d'un groupe.

2.2.1 Caractère

Définition et propriétés

DÉFINITION 2.1 (CARACTÈRE)
Un *caractère* du groupe abélien G est un morphisme de groupes $\chi : G \longrightarrow \mathbb{C}^*$, *i.e.* une fonction de G dans $\mathbb{C}^*$ telle que $\forall g, g' \in G, \chi(g + g') = \chi(g)\chi(g')$.

On note $\widehat{G}$ l'ensemble des caractères du groupe G.

Lemme *Soit χ un caractère du groupe abélien G (de cardinal N). $\forall g \in G$, $\chi(g)$ est une racine N-ième de l'unité, i.e. $\exists k \in \mathbb{N}$ tel que $\chi(g) = e^{\frac{2i\pi k}{N}}$.*

Preuve
χ étant un morphisme, on a $\chi(g)^N = \chi(Ng) = \chi(0)$, d'après le théorème de Lagrange ($Ng = 0$ car $N = |G|$). Or $\chi(0) = 1$: $\forall g' \in G$, $\chi(g') = \chi(g' + 0) = \chi(g')\chi(0)$, et $\chi(g') \neq 0$ car $\chi(g') \in \mathbb{C}^*$, donc $\chi(0) = 1$. $\square$

Proposition 2.1 *L'ensemble $\widehat{G}$ des caractères de G est un groupe pour la multiplication $(\chi\chi')(g) = \chi(g)\chi'(g)$.*

Preuve
Soient χ et $\chi' \in \widehat{G}$ deux caractères de G. Montrons que le produit $\chi\chi'$ est encore un caractère de G : c'est

évidemment un morphisme de G dans $\mathbb{C}^*$. Soit le caractère trivial $\mathbf{1}$ tel que $\forall g \in G$, $\mathbf{1}(g) = 1$. $\mathbf{1}$ est un élément neutre pour $\widehat{G}$, car $\forall \chi \in \widehat{G}$, $\mathbf{1}\chi = \chi\mathbf{1} = \chi$.

Montrons que tout caractère χ admet un inverse par rapport à $\mathbf{1}$: soit χ' défini par $\chi'(g) = \frac{1}{\chi(g)}$, alors χ' est un morphisme de G dans $\mathbb{C}^*$, et $\chi\chi' = \mathbf{1}$, donc $\chi' = \chi^{-1}$.

Donc $\widehat{G}$ est un groupe. □

Proposition 2.2 (Caractères d'un groupe cyclique) *L'ensemble des caractères de $\mathbb{Z}_N$ est isomorphe à $\mathbb{Z}_N$.*

Preuve
Rappelons que tout groupe cyclique d'ordre N est isomorphe à $\mathbb{Z}_N$. Donc si G est cyclique, alors $\widehat{G} \approx G$. On a vu que $\widehat{\mathbb{Z}_N}$ est un groupe ; montrons qu'il est cyclique et d'ordre N. On note $k_N = k + N\mathbb{Z}$; 1_N est un élément générateur de $\mathbb{Z}_N$. $\forall \chi \in \widehat{\mathbb{Z}_N}$, χ est défini par l'image de 1_N, puisque c'est un morphisme. Or $\chi(1_N)$ peut avoir au plus N valeurs, qui sont les racines N-ièmes de l'unité. Donc $|\widehat{\mathbb{Z}_N}| \leqslant N$.

Soit χ_1 le caractère défini par $\chi_1(1_N) = e^{\frac{2i\pi}{N}}$. Montrons que χ_1 est bien défini : $\chi_1(k_N) = e^{\frac{2i\pi k}{N}}$. Montrons de plus que l'ordre de χ_1 est N : $\chi_1^k(1_N) = e^{\frac{2i\pi k}{N}} \neq \chi_1(1_N)$ pour $1 < k < N$. Or l'ordre de $\widehat{\mathbb{Z}_N}$ est au plus N, et nous avons trouvé un élément d'ordre N, donc $\widehat{\mathbb{Z}_N}$ est cyclique et d'ordre N. Donc $\widehat{\mathbb{Z}_N} \approx \mathbb{Z}_N$. □

Corollaire 2.1 *Soit G un groupe abélien fini. Alors $G \approx \widehat{G}$.*

Preuve
Rappelons que tout groupe abélien fini G de cardinal N peut être décomposé en somme directe (ou produit cartésien) de groupes cycliques : $G \approx \mathbb{Z}_{d_1} \oplus \cdots \oplus \mathbb{Z}_{d_k}$, avec $N = d_1 d_2 \cdots d_k$.

Soit $\chi \in \widehat{G}$. On considère l'application $\eta : \widehat{G} \longrightarrow \prod_{i=1}^k \widehat{\mathbb{Z}_{d_i}}$ telle que $\eta(\chi) = (\chi_1, \ldots, \chi_k)$, où les χ_i vérifient : $\forall i$, $1 \leqslant i \leqslant k$, $\forall g \in \mathbb{Z}_{d_i}$, $\chi_i(g) = (\chi(g))_i$, cela en considérant la décomposition $G \approx \mathbb{Z}_{d_1} \oplus \cdots \oplus \mathbb{Z}_{d_k}$ (on prend la i-ème composante dans la décomposition de G : χ_i est la restriction de χ à la i-ème composante).

Montrons que η est bien défini : chaque χ_i est la "restriction" de χ à $\mathbb{Z}_{d_i}$: si $g \in \mathbb{Z}_{d_i}$, alors $\chi_i(g) = \chi(g)$, ce qui définit bien un morphisme de $\mathbb{Z}_{d_i}$ dans $\mathbb{C}^*$ puisque χ en est un. La somme étant directe, les χ_i sont uniquement déterminés. Montrons que η est un morphisme de groupe ($\prod_{i=1}^k \widehat{\mathbb{Z}_{d_i}}$ est un groupe en tant que produit cartésien de groupes). Il suffit de montrer que $\eta(\chi\chi') = \eta(\chi)\eta(\chi')$: la restriction de $\chi\chi'$ à $\mathbb{Z}_{d_i}$ est le produit des restrictions de χ et χ' à $\mathbb{Z}_{d_i}$.

Montrons de plus que η est un isomorphisme. Il suffit maintenant de montrer qu'il est bijectif. L'injectivité découle du fait que si $\chi \neq \chi'$, alors $\exists g \in G$ tel que $\chi(g) \neq \chi'(g)$, donc $\exists g_i \in \mathbb{Z}_{d_i}$ tel que $\chi_i(g_i) \neq \chi'_i(g_i)$. Pour la surjectivité : si on considère $(\chi_1, \ldots, \chi_k)$, on définit $\chi \in \widehat{G}$ comme suit. Soit $g \in G$, $g = g_1 + \cdots + g_k$ par la somme directe, et de façon unique ; on définit $\chi(g) = \prod_{i=1}^k \chi_i(g_i)$. χ est bien un morphisme, puisque les χ_i en sont, et $\eta(\chi) = (\chi_1, \ldots, \chi_k)$.

Ainsi, on a un isomorphisme $\widehat{G} \approx \prod_{i=1}^k \widehat{\mathbb{Z}_{d_i}}$; or $\forall i$, $\widehat{\mathbb{Z}_{d_i}} \approx \mathbb{Z}_{d_i}$ d'après la proposition 2.2, d'où le résultat. □

Espace des fonctions de G dans $\mathbb{C}$

Soit $V = \{f : G \longrightarrow \mathbb{C}\}$ l'ensemble des fonctions (quelconques) de G dans $\mathbb{C}$. C'est un espace vectoriel sur le corps $\mathbb{C}$, avec l'addition : $f, f' \in V$, $g \in G$, $(f + f')(g) = f(g) + f'(g)$; et avec la multiplication par un scalaire : $\lambda \in \mathbb{C}$, $f \in V$ et $g \in G$, $(\lambda f)(g) = \lambda f(g)$. Cet espace vectoriel est de dimension $N = |G|$, car chaque fonction $f : G \longrightarrow \mathbb{C}$ peut être vue comme un vecteur $(f(g_1), \ldots, f(g_N))$ si $G = \{g_i | 1 \leqslant i \leqslant N\}$.

On munit l'espace vectoriel V du produit scalaire :

$$\langle f | f' \rangle = \sum_{g \in G} \overline{f(g)} f'(g)$$

où $\overline{f(g)}$ est le conjugué complexe de $f(g)$.

Proposition 2.3 *$\widehat{G}$ est une base orthogonale de V.*

Preuve
Il suffit de montrer que les caractères sont deux à deux orthogonaux. En effet, il y en a N d'après le corollaire 2.1, donc s'ils sont orthogonaux, ils forment une famille libre de cardinal $N = \dim V$, ce qui est une base.

Soient χ et χ' deux caractères de G différents. Il suffit de montrer que $S = \sum_{g\in G} \overline{\chi(g)}\chi'(g) = 0$. Or $\overline{\chi}$ est encore un caractère de G, car $\overline{\chi} = \chi^{-1}$, donc $\overline{\chi}\chi'$ est aussi un caractère de G. Soit $\varphi = \overline{\chi}\chi' \in \widehat{G}$; φ n'est pas le caractère trivial $\mathbf{1}$, sinon $\chi^{-1}\chi' = \mathbf{1}$, soit $\chi = \chi'$, ce qui est faux. Donc $\exists g_0 \in G$ tel que $\varphi(g_0) \neq 1$. G étant un groupe, $g \mapsto g_0 + g$ est une permutation de G ; φ étant un morphisme, on a :

$$S = \sum_{g\in G} \varphi(g_0 + g) = \varphi(g_0)S$$

donc $S = 0$. □

Corollaire 2.2 *Soit $\mathcal{B} = \{\frac{1}{\sqrt{N}}\chi | \chi \in \widehat{G}\}$. Alors $\mathcal{B}$ est une base orthonormée de V.*

Preuve
Il suffit de montrer que $\| \chi \| = \sqrt{N}$ lorsque $\chi \in \widehat{G}$, ce qui équivaut à $\langle\chi|\chi\rangle = N$. Or cela découle de $\overline{\chi}\chi = \mathbf{1}$: en effet, $\langle\chi|\chi\rangle = \sum_{g\in G} \mathbf{1}(g) = N$. □

Dans la suite, puisque G et $\widehat{G}$ sont isomorphes, on notera $\mathcal{B} = \{\mathbf{b_g} | g \in G\}$. De même, on utilisera la notation χ_g.

2.2.2 Transformation de Fourier discrète

Définition et propriétés

DÉFINITION 2.2 (TRANSFORMATION DE FOURIER DISCRÈTE)
Soit $f \in V$ une fonction de G dans $\mathbb{C}$. $\mathcal{B}$ étant une base, f s'écrit : $f = \sum_{g\in G} \widehat{f}_g \mathbf{b_g}$. On appelle *transformée de Fourier discrète* de f la fonction $\widehat{f} : \widehat{G} \longrightarrow \mathbb{C}$ définie par $\widehat{f}(\chi_g) = \widehat{f}_g$. La *transformation de Fourier discrète* est l'application qui à f associe $\widehat{f}$.

On a donc, $\mathcal{B}$ étant une base orthonormée, $\widehat{f}(\chi_g) = \langle \mathbf{b_g} | f\rangle = \frac{1}{\sqrt{N}} \sum_{g'\in G} \overline{\chi_g(g')} f(g')$.

Puisque $\widehat{G} \approx G$ d'après le corollaire 2.1, on peut considérer $\widehat{f} : G \longrightarrow \mathbb{C}$, avec $\widehat{f}(g) = \widehat{f}_g$.

Proposition 2.4 (Identité de Parseval) *Soit $f \in V$. Alors $\| f \| = \| \widehat{f} \|$.*

Preuve

$$\| \widehat{f} \|^2 = \langle\widehat{f}|\widehat{f}\rangle = \sum_{g\in G} \overline{\widehat{f}(g)}\widehat{f}(g) = \sum_{g\in G} |\langle\mathbf{b_g}|f\rangle|^2 = \| f \|^2$$

□

Corollaire 2.3 *La transformation de Fourier discrète est une transformation unitaire.*

Ainsi, puisque la transformation de Fourier discrète est unitaire, on va pouvoir l'utiliser en informatique quantique.

Exemples de transformation de Fourier discrète sur certains groupes abéliens

Exemple Transformation de Fourier discrète sur $\mathbb{Z}_N$

On a déjà trouvé tous les caractères de $\mathbb{Z}_N$ dans la proposition 2.2. On a un isomorphisme $\eta : \widehat{\mathbb{Z}_N} \longrightarrow \mathbb{Z}_N$ défini par $\eta(\chi) = \chi(1)$, donc on peut indexer les caractères selon leur valeur en 1. On a : $\widehat{\mathbb{Z}_N} = \{\chi_k | k \in \mathbb{Z}_N \text{ et } \chi_k(1) = e^{\frac{2i\pi k}{N}}\}$. Ainsi, $\chi_k(l) = e^{\frac{2i\pi lk}{N}}$. Cet isomorphisme nous permet de considérer $\widehat{f} : G \longrightarrow \mathbb{C}$ comme précédemment.

Soit f une fonction de $\mathbb{Z}_N$ dans $\mathbb{C}$. On a, pour $k \in \mathbb{Z}_N$:

$$\widehat{f}(k) = \frac{1}{\sqrt{N}} \sum_{l\in\mathbb{Z}_N} \overline{\chi_k(l)} f(l) = \frac{1}{\sqrt{N}} \sum_{l\in\mathbb{Z}_N} e^{-\frac{2i\pi kl}{N}} f(l)$$

Exemple Transformation de Fourier discrète sur le groupe $(\mathbb{Z}_2)^n$

Les 2 caractères sur $\mathbb{Z}_2$ sont $\mathbf{1}$ et χ_{-1} (tel que $\chi_{-1}(0) = 1$ et $\chi_{-1}(1) = -1$) ; les caractères de $(\mathbb{Z}_2)^n$ sont les "produits" de ces deux caractères, comme on a pu le voir dans la démonstration de la proposition 2.1 : si on note $\mathbf{x}, \mathbf{y} \in (\mathbb{Z}_2)^n$ et leur produit scalaire $\mathbf{x} \cdot \mathbf{y}$ (produit scalaire usuel de deux vecteurs de dimension n : somme des produits composante par composante), on obtient tous les caractères de $(\mathbb{Z}_2)^n$:

$$\chi_{\mathbf{y}}(\mathbf{x}) = (-1)^{\mathbf{x}\cdot\mathbf{y}} \text{ pour } \mathbf{y} \in (\mathbb{Z}_2)^n$$

Ainsi, la transformée de Fourier discrète de $f : (\mathbb{Z}_2)^n \longrightarrow \mathbb{C}$ est $\widehat{f} : (\mathbb{Z}_2)^n \longrightarrow \mathbb{C}$ définie par :

$$\widehat{f}(\mathbf{x}) = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{y}\in(\mathbb{Z}_2)^n} (-1)^{\mathbf{x}\cdot\mathbf{y}} f(\mathbf{y})$$

2.3 Transformation de Fourier quantique sur groupe abélien fini

$G = \{g_1, \ldots, g_N\}$ représente toujours un groupe abélien fini de cardinal N. On choisit une base orthonormée de notre système quantique représentant les éléments de G, *i.e.* $\{|g_1\rangle, \ldots, |g_N\rangle\}$ est une base orthonormée. Ainsi, tout état quantique s'écrit

$$|\psi\rangle = \sum_{i=1}^{N} \alpha_i |g_i\rangle$$

où $\forall i$, $\alpha_i \in \mathbb{C}$ et $\sum_{i=1}^{N} |\alpha_i|^2 = 1$. $|\psi\rangle$ peut donc être considéré comme une fonction f de G dans $\mathbb{C}$: $f(g_i) = \alpha_i$.

Définition 2.3 (Transformation de Fourier quantique)
Soit $G = \{g_1, \ldots, g_N\}$ un groupe abélien fini de cardinal N. La *transformation de Fourier quantique* sur G est la fonction QFT_G suivante :

$$\mathrm{QFT}_G : \sum_{i=1}^{N} \alpha_i |g_i\rangle \longmapsto \sum_{i=1}^{N} \widehat{\alpha_i} |g_i\rangle$$

où $\widehat{\alpha_i} = \widehat{f}(g_i)$, avec $f : g_i \mapsto \alpha_i$.

Lemme *Soit G un groupe abélien fini, somme directe de deux sous-groupes U et V : $G = U \oplus V$. Soit un système quantique représentant G sous cette forme de somme directe, i.e. $\{|u\rangle |v\rangle \,|u \in U, v \in V\}$ est une base orthonormée. Alors la transformation de Fourier quantique sur G est le produit des transformations de Fourier quantiques sur U et sur V :*

$$QFT_G : |u\rangle |v\rangle \longmapsto (QFT_U(|u\rangle)) \otimes (QFT_V(|v\rangle))$$

Preuve
Soient $u \in U$, $v \in V$, $\chi_u \in \widehat{U}$, $\chi_v \in \widehat{V}$. On considère la fonction $f : G \longrightarrow \mathbb{C}$ nulle partout sauf en $u + v$ où elle vaut 1.

$$\mathrm{QFT}_G(|u\rangle |v\rangle) = \frac{1}{\sqrt{N}} \sum_{g\in G} \widehat{f}(g) |g\rangle$$

Or

$$\widehat{f}(g) = \frac{1}{\sqrt{N}} \sum_{g'\in G} \overline{\chi_g(g')} f(g') = \frac{1}{\sqrt{N}} \overline{\chi_g(u+v)}$$

donc on obtient

$$\mathrm{QFT}_G(|u\rangle |v\rangle) = \frac{1}{N} \sum_{g\in G} \overline{\chi_g(u+v)} |g\rangle$$

De même, on a :

$$\mathrm{QFT}_U(|u\rangle) = \frac{1}{|U|} \sum_{u'\in U} \overline{\chi_{u'}(u)} |u'\rangle \text{ et } \mathrm{QFT}_V(|v\rangle) = \frac{1}{|V|} \sum_{v'\in V} \overline{\chi_{v'}(v)} |v'\rangle$$

donc en faisant le produit, on obtient

$$\mathrm{QFT}_U(|u\rangle)\mathrm{QFT}_V(|v\rangle) = \frac{1}{|U||V|} \sum_{u',v'} \overline{chi_{u'}(u)\chi_{v'}(v)} |u'\rangle |v'\rangle = \frac{1}{N} \sum_{g\in G} \overline{\chi_g(u+v)} |g\rangle = \mathrm{QFT}_G(|u\rangle |v\rangle)$$

car G est la somme directe de U et V (*cf* la décomposition du corollaire 2.1). □

Avec ce lemme, on voit que l'on peut appliquer la transformation de Fourier quantique localement lorsque le groupe G est décomposable en somme directe. Cependant, on n'a pas de résultat sur la transformation de Fourier quantique sur les groupes cycliques, donc ce résultat ne peut pas toujours servir pour les groupes abéliens. On peut toutefois l'appliquer pour implémenter la transformation de Fourier quantique sur le groupe $(\mathbb{Z}_2)^m$, par exemple.

2.4 Transformation de Fourier quantique sur $\mathbb{Z}_N$

Pour simplifier, on notera F_N la transformation de Fourier quantique sur $\mathbb{Z}_N$.

2.4.1 Groupe $\mathbb{Z}_{2^n}$

D'après le théorème chinois, si $N = pq$ avec $\text{pgcd}(p, q) = 1$, alors $\mathbb{Z}_N \approx \mathbb{Z}_p \oplus \mathbb{Z}_q$. Ainsi, on peut appliquer le lemme précédent pour calculer la transformation de Fourier quantique sur $\mathbb{Z}_N$ récursivement. Cependant, souvent de tels facteurs p et q n'existent pas : par exemple, si $N = 2^n$, tous les diviseurs non triviaux de N ont pour pgcd 2, et on ne peut pas appliquer le théorème chinois. Mais on peut trouver mieux dans le cas où $N = 2^n$.

Nous cherchons donc à effectuer la transformation de Fourier quantique sur $\mathbb{Z}_{2^n}$:

$$\text{QFT}_{\mathbb{Z}_{2^n}} : |x\rangle \longmapsto \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{-\frac{2i\pi xy}{2^n}} |y\rangle$$

Lemme

$$\sum_{y=0}^{2^n-1} e^{-\frac{2i\pi xy}{2^n}} |y\rangle = (|0\rangle + e^{-\frac{2i\pi x}{2^1}} |1\rangle)(|0\rangle + e^{-\frac{2i\pi x}{2^2}} |1\rangle) \cdots (|0\rangle + e^{-\frac{2i\pi x}{2^n}} |1\rangle) \tag{2.1}$$

<u>Preuve</u>
On raisonne par récurrence, et on se ramène à l'hypothèse de récurrence en différenciant les cas où y est pair et où y est impair :

$$\sum_{y=0}^{2^n-1} e^{-\frac{2i\pi xy}{2^n}} |y\rangle = \sum_{y'=0}^{2^{n-1}-1} e^{-\frac{2i\pi x(2y')}{2^n}} |y'0\rangle + \sum_{y'=0}^{2^{n-1}-1} e^{-\frac{2i\pi x(2y'+1)}{2^n}} |y'1\rangle = \sum_{y=0}^{2^{n-1}-1} e^{-\frac{2i\pi xy}{2^{n-1}}} |y\rangle \left(|0\rangle + e^{-\frac{2i\pi x}{2^n}} |1\rangle\right)$$

□

Dans la décomposition du lemme précédent, la phase du k-ième $|1\rangle$, $e^{-\frac{2i\pi x}{2^k}}$, ne dépend que des $k-1$ premiers bits de x :

$$\exp\left(-2i\pi \frac{2^0 x_0 + 2^1 x_1 + \cdots + 2^n x_n}{2^k}\right) = \exp\left(-2i\pi \frac{2^0 x_0 + 2^1 x_1 + \cdots + 2^{k-1} x_{k-1}}{2^k}\right) = \prod_{l=1}^{k-1} e^{-\frac{2i\pi x_l}{2^{k-l+1}}}$$

À partir de cette décomposition, on va pouvoir implémenter la transformation de Fourier quantique efficacement.

En effet, en partant de l'état $|x\rangle$, *i.e.* le k-ième qubit à $|x_k\rangle$, on applique la transformation d'Hadamard sur le dernier qubit, afin d'obtenir $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_n} |1\rangle)$. Puis on corrige la phase de $|1\rangle$ en appliquant les rotations conditionnelles suivantes sur le k-ième et sur le dernier qubit, pour k variant de 0 à $n-1$ (dans l'écriture qui suit, le premier registre correspond donc au k-ième qubit, et le second registre au dernier qubit) :

$$\begin{array}{ll} |0\rangle|0\rangle \mapsto |0\rangle|0\rangle & \quad |0\rangle|1\rangle \mapsto |0\rangle|1\rangle \\ |1\rangle|0\rangle \mapsto |1\rangle|0\rangle & \quad |1\rangle|1\rangle \mapsto e^{\frac{2i\pi}{2^{n-k+1}}} |1\rangle|1\rangle \end{array}$$

Ainsi, on obtient la bonne phase pour le dernier qubit. On fait ensuite la même transformation (que pour le qubit n) sur les qubits $n-1$ à 0, et on a implémenté la transformation de Fourier quantique sur $\mathbb{Z}_{2^n}$. L'algorithme nécessite $\mathcal{O}(n^2)$ étapes, soit $\mathcal{O}(\log^2 N)$.

2.4.2 Transformation de Fourier quantique approchée sur $\mathbb{Z}_N$

Nous aimerions maintenant pouvoir effectuer la transformation de Fourier quantique sur tout groupe cyclique, car tout groupe abélien se décompose en somme directe de groupes cycliques. Les algorithmes efficaces connus effectuent une transformation de Fourier quantique approchée sur $\mathbb{Z}_N$, aussi proche que l'on veut de la transformation de Fourier quantique exacte.

L'idée est de faire la transformation de Fourier quantique sur un groupe $\mathbb{Z}_M$ avec M suffisamment grand devant N, en choisissant M correctement (une puissance de 2) pour pouvoir faire efficacement la transformation de Fourier quantique sur $\mathbb{Z}_M$. Si l'on effectue une homothétie de rapport $\frac{M}{N}$ à la fonction de départ, la transformation de Fourier quantique sur $\mathbb{Z}_M$ sera sensiblement la même que celle de la fonction initiale sur $\mathbb{Z}_N$. Si M est raisonnable (grand devant N mais pas trop), on va pouvoir effectuer la transformation de Fourier quantique sur $\mathbb{Z}_M$ en $\mathcal{O}(\log^2 M) = \mathcal{O}(\log^2 M)$. L'article [Hø00] montre que la transformation de Fourier quantique sur $\mathbb{Z}_M$ approche effectivement celle sur $\mathbb{Z}_N$ pour M assez grand. La transformation de Fourier quantique sur $\mathbb{Z}_N$ est une "restriction" de la transformation de Fourier quantique sur $\mathbb{Z}_M$: rappelons que F_M et F_N sont les transformation de Fourier quantique sur $\mathbb{Z}_M$ et $\mathbb{Z}_N$ respectivement. On définit alors $F'_N = cRF_M$, où c est une constante de normalisation, et R "choisit" les bons éléments : R est défini comme

$$R(|i'\rangle) = |i\rangle$$

avec pour tout $0 \leqslant i < N, i' = \left\lfloor i\frac{M}{N} + \frac{1}{2} \right\rfloor$. C'est l'homothétie de rapport $\frac{M}{N}$ dont nous parlions précédemment, qui s'implémente facilement à la manière classique, *i.e.* en calculant $i' = \left\lfloor i\frac{M}{N} + \frac{1}{2} \right\rfloor$. Ainsi, F'_N effectue la transformation de Fourier quantique sur $\mathbb{Z}_M$ et prend ensuite les N éléments i' équirépartis. Dans [Hø00], il est montré que de cette façon, on approche la transformation de Fourier quantique sur $\mathbb{Z}_N$. Plus précisément :

Proposition 2.5 *Il suffit de prendre* $M \geqslant \frac{15}{\varepsilon} N \log N$ *pour avoir* $\| \; F_N |u\rangle - F'_N |u\rangle \; \| \leqslant \varepsilon$. *De plus, les distributions de probabilités (la norme au carré des amplitudes) de* F'_N *et de* F_N *sont proches également.*[3]

Maintenant, si on se fixe $\varepsilon > 0$ petit comme marge d'erreur, on choisit

$$M = 2^{\left\lceil \log\left(\frac{15}{\varepsilon} N \log N\right)\right\rceil}$$

et on applique la transformation de Fourier quantique sur $\mathbb{Z}_M$, que l'on sait faire en $\mathcal{O}(\log^2 M)$ comme on a vu précédemment, puis on restreint grâce à l'opérateur R, pour se ramener à $\mathbb{Z}_N$.

La complexité du calcul de F'_N est alors de $\mathcal{O}(\log^2 M) = \mathcal{O}((\log(N \log N))^2) = \mathcal{O}(\log^2 N)$, donc on a implémenté la transformation de Fourier quantique approchée efficacement.

2.4.3 Transformation de Fourier quantique exacte sur $\mathbb{Z}_N$

On sait faire efficacement la transformation de Fourier quantique approchée sur $\mathbb{Z}_N$, et cela suffit pour les applications que l'on en fait. D'un point de vue théorique, on peut tout de même se demander comment faire la transformation de Fourier quantique exacte sur $\mathbb{Z}_N$. Il nous faut alors changer de modèle, et travailler avec des circuits non-uniformes, puisque l'on veut réaliser des transformations exactes. On se donne un ensemble de portes quantiques de base spécifique pour le problème de la transformation de Fourier quantique sur $\mathbb{Z}_N$, et l'on ne travaille plus avec un ensemble universel approché de portes quantiques.

L'algorithme que l'on a vu pour effectuer la transformation de Fourier quantique sur $\mathbb{Z}_{2^n}$ est exact. On peut généraliser facilement la méthode pour $N = p^\alpha$ pour p petit devant N. En effet, de la même façon que lorsque $p = 2$, la décomposition (2.1) (p. 10) devient

$$\begin{aligned}\textstyle\sum_{y=0}^{N-1} e^{-\frac{2i\pi xy}{N}} |y\rangle = \quad & (|0\rangle + e^{-\frac{2i\pi x}{p^1}} |1\rangle + e^{-\frac{2i\pi x}{2p^1}} |2\rangle + \cdots + e^{-\frac{2i\pi x}{(p-1)p^1}} |p-1\rangle) \otimes \\ & (|0\rangle + e^{-\frac{2i\pi x}{p^2}} |1\rangle + e^{-\frac{2i\pi x}{2p^2}} |2\rangle + \cdots + e^{-\frac{2i\pi x}{(p-1)p^2}} |p-1\rangle) \otimes \cdots \\ & \otimes (|0\rangle + e^{-\frac{2i\pi x}{p^\alpha}} |1\rangle + e^{-\frac{2i\pi x}{2p^\alpha}} |2\rangle + \cdots + e^{-\frac{2i\pi x}{(p-1)p^\alpha}} |p-1\rangle)\end{aligned}$$

ce qui nous permet d'utiliser la même méthode de rotations conditionnelles que pour $p = 2$.

De même, on peut facilement effectuer la transformation de Fourier quantique exacte sur $\mathbb{Z}_N$ lorsque tous les diviseurs premiers de N sont petits. Mais que faire lorsque N est lui-même premier, par exemple ? La fin du stage (trop courte) a été consacrée à étudier ce cas ; cependant, je n'ai trouvé aucun algorithme vraiment intéressant. Une étude du cas classique aurait pu aider dans la résolution du cas quantique, c'est

[3] Quelques erreurs mineures dans le papier de Høyer (notamment dans le lemme 5 page 7) font que les constantes du papier initial sont erronnées.

pourquoi j'ai commencé par l'étude de l'algorithme de la FFT pour N quelconque. Présentons cet algorithme classique, qui à partir de la fonction f calcule $\widehat{f}$ définie par

$$\widehat{f}(k) = \sum_{j=0}^{N-1} e^{\frac{2i\pi kj}{N}} f(j)$$

On va utiliser une fonction intermédiaire, FFT', qui prend en entrée un nombre complexe α et un tableau T de taille 2^n de valeurs complexes : il s'agit des valeurs de la fonction f dont on cherche la transformée de Fourier. En sortie, on retourne un tableau $\widehat{T}$ de taille 2^n correspondant aux valeurs de $\widehat{f}$. La fonction FFT' réalise la transformation suivante :

$$\widehat{T}(k) = \sum_{j=0}^{2^n-1} \alpha^{jk} T(j)$$

FFT'$(\alpha, T) =$

1. Pour i de 0 à $2^{n-1}-1$ faire
   - $T_0(i) \leftarrow T(2i)$;
   - $T_1(i) \leftarrow T(2i+1)$;
2. $\widehat{T_0} \leftarrow$ FFT'(α^2, T_0) ;
3. $\widehat{T_1} \leftarrow$ FFT'(α^2, T_1) ;
4. Pour i de 0 à $2^{n-1}-1$ faire
   - $\widehat{T}(i) \leftarrow \widehat{T_0}(i) + \alpha^i \widehat{T_1}(i)$;
   - $\widehat{T}(2^{n-1}+i) \leftarrow \widehat{T_0}(i) + \alpha^{2^{n-1}+i} \widehat{T_1}(i)$;
5. Retourner $\widehat{T}$.

Cet algorithme est justifié par le fait que

$$\widehat{T}(k) = \sum_{j=0}^{2^n-1} \alpha^{jk} T(j) = \sum_{j=0}^{2^{n-1}-1} (\alpha^2)^{jk} T_0(j) + \alpha^k \sum_{j=0}^{2^{n-1}-1} (\alpha^2)^{jk} T_1(j)$$

La complexité $C(2^n)$ de cet algorithme vérifie $C(2^n) = 2C(2^{n-1}) + \mathcal{O}(2^n)$ ce qui mène à $C(2^n) = \mathcal{O}(n2^n)$, soit, si $N = 2^n$, $C(N) = \mathcal{O}(N \log N)$.

Pour obtenir la FFT de notre fonction f à N valeurs, il suffit maintenant de compléter le tableau des valeurs de f par des zéros, jusqu'à obtenir un tableau T de taille 2^n, puis d'appeler FFT'$(e^{\frac{2i\pi}{N}}, T)$. On obtient ainsi un algorithme de calcul de la FFT de complexité $\mathcal{O}(N \log N)$.

Cependant, je n'ai pas réussi à utiliser cet algorithme classique pour résoudre le cas quantique, notamment car la transformation n'est pas unitaire. On peut noter toutefois le résultat suivant, qui effectue la transformation de Fourier quantique exacte, mais l'algorithme est d'une complexité $\mathcal{O}(\sqrt{N})$ qui le rend inutilisable, mais qui est mieux qu'en classique.

Proposition 2.6 *Soit N quelconque (N peut être premier). La transformation de Fourier quantique exacte sur $\mathbb{Z}_N$ peut être faite en $\mathcal{O}(\sqrt{N})$ étapes.*

<u>Preuve</u>
Présentons l'algorithme pour effectuer la transformation de Fourier quantique exacte en $\mathcal{O}(\sqrt{N})$. On rappelle que la transformation de Fourier quantique sur $\mathbb{Z}_N$ est la transformation unitaire

$$F_N : |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} \omega_N^{ix} |i\rangle$$

où $\omega_N = \exp\left(\frac{2i\pi}{N}\right)$. On utilisera la notation ω_q pour tout $q \in \mathbb{Q}$, et pas seulement pour les entiers.

Soit $q \in \mathbb{Q}$, et $m_q \in \mathbb{N}$ le plus petit entier tel que $q \leqslant 2^{m_q}$. Considérons la transformation

$$U_q : |x\rangle |0\rangle \mapsto \frac{1}{\sqrt{2^{m_q}}} \sum_{i=0}^{2^{m_q}-1} \omega_q^{ix} |x\rangle |i\rangle$$

Cette transformation peut-être completée en une transformation unitaire, car deux vecteurs unitaires orthogonaux sont envoyés sur deux vecteurs unitaires orthogonaux.

On peut alors décomposer la transformation U_q comme suit :

$$U_q(|x\rangle|0\rangle) = \left(\frac{1}{\sqrt{2^{m_{q/2}}}}\sum_{i=0}^{2^{m_{q/2}}-1}\omega_{q/2}^{ix}|x\rangle|i\rangle\right)\otimes\frac{1}{\sqrt{2}}(|0\rangle+\omega_q^x|1\rangle) = U_{q/2}(|x\rangle|0\rangle)\otimes\frac{1}{\sqrt{2}}(|0\rangle+\omega_q^x|1\rangle)$$

Ainsi, on peut appeler récursivement $U_{q'}$ pour $q' = \frac{q}{2}$. Le nombre d'appels à la transformation U_q est logarithmique, si l'on sait la faire pour $m_q = 1$. Or il s'agit de la transformation

$$|x\rangle|0\rangle \mapsto \frac{1}{\sqrt{2}}|x\rangle(|0\rangle+\omega_q^x|1\rangle)$$

qui est simplement $\mathcal{O}(m)$ rotations conditionnelles. De même, on réalise efficacement les rotations qui permettent de passer de U_q à $U_{q/2}$.

Ainsi, la transformation U_q complète est réalisée efficacement (complexité polynomiale en $\log q$). Maintenant, soit φ_q la tranformation suivante

$$\varphi_q : |i\rangle|0\rangle \mapsto |i\rangle|\lfloor i/q\rfloor\rangle$$

Si l'on applique φ_q au résultat obtenu après U_q, sur le deuxième registre, on a

$$\varphi_q\left(\frac{1}{\sqrt{2^{m_q}}}\sum_{i=0}^{2^{m_q}-1}\omega_q^{ix}|x\rangle|i\rangle|0\rangle\right) = \frac{1}{\sqrt{2^{m_q}}}\sum_{i=0}^{2^{m_q}-1}\omega_q^{ix}|x\rangle|i\rangle|\lfloor i/q\rfloor\rangle$$

Le troisième registre vaut $|0\rangle$ si et seulement si $i < q$. Supposons $q = N$ entier. Lorsque l'on mesure le troisième registre, on obtient alors, avec probabilité plus grande que $\frac{1}{2}$

$$\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}\omega_n^{ix}|x\rangle|i\rangle$$

De plus, le résultat de la mesure nous informe si nous sommes bien dans cet état, selon que 0 ou 1 est mesuré.

Ainsi, on a réalisé la transformation

$$V_n : |x\rangle|0\rangle \mapsto \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}\omega_n^{ix}|x\rangle|i\rangle$$

De cet état, on aimerait se débarrasser de la dépendance vis-à-vis du premier registre. En effet, les deux registres sont enchevêtrés, ce qui empêche d'avoir la transformation de Fourier quantique dans le second registre. Pour cela, on aimerait ramener le premier registre à $|0\rangle$, pour le mettre en facteur, et ainsi avoir la transformation de Fourier quantique dans le second registre. On peut utiliser l'algorithme de Grover (que nous ne présentons pas ici), qui permet de trouver un élément dans une base de données désordonnée de taille N en $\mathcal{O}(\sqrt{N})$ étapes seulement. En appliquant la même itération que dans l'algorithme de Grover, on peut faire converger le premier registre vers $|0\rangle$, en $\mathcal{O}(\sqrt{N})$ étapes. Ainsi, on a réalisé la transformation de Fourier quantique en $\mathcal{O}(\sqrt{N})$ étapes, et ce quelque soit N (décomposable ou non). □

Le problème lors de la recherche d'un algorithme de complexité polynomiale en $\log N$ vient du fait que N n'est pas forcément pair, auquel cas on pourrait utiliser la récursion comme dans le cas $N = 2^n$. De plus, si on essaie de compléter jusqu'à 2^n, comme ce que nous venons de faire, on n'obtient une transformation unitaire que si l'on ajoute un autre registre. Cet autre registre, s'il ne dépend que de x, doit contenir $|x\rangle$ forcément ; s'il ne dépend pas que de x, mais aussi de i, il existe peut-être une valeur permettant de réaliser une transformation unitaire : il faudrait continuer à chercher dans cette direction.

Peut-être peut-on trouver une borne inférieure pour la transformation de Fourier quantique exacte, mais le fait qu'on puisse la mettre en œuvre efficacement dans le cas $N = 2^n$ semble rendre la tâche difficile. De plus, puisque la FFT existe dans le cas classique, on peut conjecturer que la transformation de Fourier quantique exacte existe aussi. Il reste encore à le démontrer.

2.5 Trouver la période, cas facile

La transformation de Fourier quantique sert principalement à trouver la période d'une fonction ; montrons maintenant comment faire (voir aussi [Sho96]). Soit une fonction $f : \{0, 1, \ldots, n-1\} \longrightarrow S$, ($S$ est un ensemble fini quelconque), r-périodique, et injective sur $\{0, 1, \ldots, r-1\}$. On suppose que $r \leqslant \sqrt{n}$. Nous disposons de f grâce à une boîte noire effectuant la transformation

$$U_f : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus f(x)\rangle$$

Dans ce modèle de la boîte noire, on connaît la borne inférieure $\Omega(\sqrt{n})$ en informatique classique pour trouver la période de f (démontrée dans [Sim94]). Nous allons cependant la trouver en temps logarithmique grâce au calcul quantique.

Partant de la superposition uniforme

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle\,|0\rangle$$

on applique une fois U_f pour obtenir

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle\,|f(i)\rangle = \frac{1}{\sqrt{n}} \sum_{i=0}^{r-1} \sum_{j=0}^{\lfloor \frac{n}{r} \rfloor - 1} |i + jr\rangle\,|f(i)\rangle + \frac{1}{\sqrt{n}} \sum_{i=r\lfloor \frac{n}{r} \rfloor}^{n-1} |i\rangle\,|f(i)\rangle$$

Mesurons maintenant le second registre. On obtient une valeur $f(i_0)$, et selon $i_0 \in \{0, \ldots, r-1\}$, les registres sont maintenant :

$$\frac{1}{\sqrt{\lfloor \frac{n}{r} \rfloor}} \sum_{j=0}^{\lfloor \frac{n}{r} \rfloor - 1} |i_0 + jr\rangle\,|f(i_0)\rangle \quad \text{ou} \quad \frac{1}{\sqrt{\lfloor \frac{n}{r} \rfloor + 1}} \sum_{j=0}^{\lfloor \frac{n}{r} \rfloor} |i_0 + jr\rangle\,|f(i_0)\rangle \tag{2.2}$$

L'analyse des deux cas est identique ; supposons que nous soyons dans la première situation. On aimerait connaître r à partir du premier registre. Mais le décalage aléatoire i_0 nous empêche d'effectuer directement une mesure, car nous n'aurions aucune information. Nous appliquons alors la transformation de Fourier quantique sur $\mathbb{Z}_n$ sur le premier registre. On obtient, en ne considérant que le premier registre :

$$\frac{1}{\sqrt{\lfloor \frac{n}{r} \rfloor}} \sum_{j=0}^{\lfloor \frac{n}{r} \rfloor - 1} \left(\frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} e^{\frac{2i\pi(i_0 + jr)k}{n}}\,|k\rangle \right)$$

soit, en intervertissant les sommes :

$$\frac{1}{\sqrt{n \lfloor \frac{n}{r} \rfloor}} \sum_{k=0}^{n-1} e^{\frac{2i\pi i_0 k}{n}} \left(\sum_{j=0}^{\lfloor \frac{n}{r} \rfloor - 1} (e^{\frac{2i\pi rk}{n}})^j \right) |k\rangle \tag{2.3}$$

Si $r|n$ (hypothèse faite dans un premier temps pour simplifier les calculs), l'état précédent s'écrit

$$\frac{1}{\sqrt{\frac{n^2}{r}}} \sum_{k=0}^{n-1} e^{\frac{2i\pi i_0 k}{n}} \left(\sum_{j=0}^{\frac{n}{r} - 1} (e^{\frac{2i\pi rk}{n}})^j \right) |k\rangle$$

Or, par simple évaluation de la somme géométrique,

$$\sum_{j=0}^{\frac{n}{r} - 1} (e^{\frac{2i\pi rk}{n}})^j = \begin{cases} \frac{n}{r} & \text{si } n|rk \\ 0 & \text{sinon} \end{cases}$$

donc le registre contient seulement les k tels que $k = l\frac{n}{r}$, comme on peut le voir sur la figure 2.a :

$$\frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} e^{\frac{2i\pi i_0 ln/r}{n}} \left| l\frac{n}{r} \right\rangle$$

Ainsi, lorsque l'on effectue la mesure du registre, on obtient $k_0 = a_0 \frac{n}{r}$ avec $0 \leqslant a_0 < r$. Si l'on répète la procédure, on obtient $k_1 = a_1 \frac{n}{r}$. Or on avait probabilité uniforme sur les a_i, donc pour n grand, la

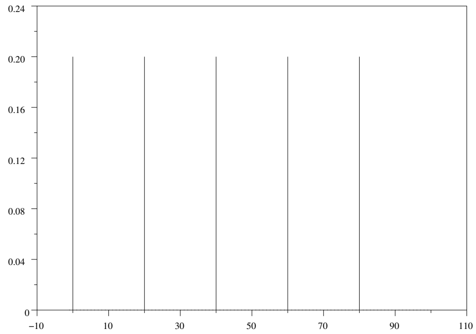

FIG. 2.a – $n = 100$, $r = 5$ (donc r divise n) : on voit 5 pics.

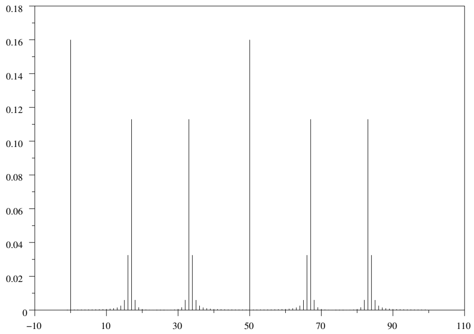

FIG. 2.b – $n = 100$, $r = 6$ (donc r ne divise pas n) : on voit 6 pics.

probabilité que a_0 et a_1 soient premiers entre eux est supérieure à $\frac{1}{2}$.[4] Si a_0 et a_1 sont premiers entre eux, alors $\text{pgcd}(k_0, k_1) = \frac{n}{r}$: on retrouve r. Finalement, avec probabilité constante, on trouve r en 2 mesures.

Revenons maintenant au cas général : r ne divise plus n. En ce qui concerne la distribution de probabilité de mesure, nous obtenons des pics autour des multiples de $\frac{n}{r}$, mais les autres valeurs ne sont plus nulles (*cf* figures 2.a et 2.b).

Lorsqu'on effectue la mesure du registre, intéressons-nous à la probabilité d'obtenir une valeur k_0 telle qu'il existe a_0 vérifiant $a_0\frac{n}{r} - \frac{1}{2} < k_0 \leqslant a_0\frac{n}{r} + \frac{1}{2}$ (dans ce cas, k_0 est "proche" d'un pic). Par périodicité et parité, cela revient à calculer le rapport

$$\left(\int_0^{\frac{1}{2}} f(x)dx\right) \Big/ \left(\int_0^{\frac{n}{2r}} f(x)dx\right), \text{ avec } f(x) = \frac{1}{n\lfloor n \rfloor r}\left|\sum_{j=0}^{\lfloor \frac{n}{r} \rfloor - 1} (e^{\frac{2i\pi r x}{n}})^j\right|^2$$

Nous nous contenterons d'une minoration. On peut visualiser la situation ainsi :

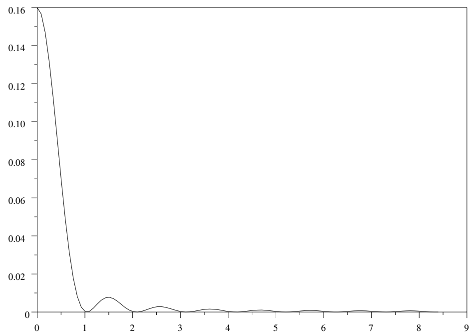

FIG. 3 – Fonction f entre 0 et $\frac{n}{r}$ pour $n = 100$ et $r = 6$.

f est décroissante sur $[0, \frac{1}{2}]$, donc on peut minorer le numérateur par $f(\frac{1}{2})/2$. Le dénominateur est égal à l'intégrale sur tout le domaine $[0, n]$, divisé par $2r$ (du fait de la périodicité et parité de f). Or l'intégrale sur tout le domaine est égale à 1, car f représente une probabilité, donc le dénominateur est égal à $\frac{1}{2r}$. Ainsi, la probabilité est minorée par

$$rf(\frac{1}{2}) = \frac{r}{n\lfloor n \rfloor r}\left|\sum_{j=0}^{\lfloor \frac{n}{r} \rfloor - 1} (e^{\frac{2i\pi r}{2n}})^j\right|^2 = \frac{r}{n\lfloor n \rfloor r}\left|\frac{1 - e^{\frac{2i\pi r \lfloor \frac{n}{r} \rfloor}{2n}}}{1 - e^{\frac{2i\pi r}{2n}}}\right|^2$$

[4]La probabilité pour que deux nombres tirés uniformément entre 1 et n soient premiers entre eux tend vers $\frac{6}{\pi^2}$ lorsque n tend vers l'infini.

En notant que $r \leqslant \sqrt{n}$, et en faisant des calculs asymptotiques, on obtient que

$$\left|\frac{1-e^{\frac{2i\pi r\left\lfloor\frac{n}{r}\right\rfloor}{2n}}}{1-e^{\frac{2i\pi r}{2n}}}\right|^2 \sim \frac{4n^2}{\pi^2 r^2}$$

donc la probabilité de mesurer k_0 vérifiant $a_0\frac{n}{r}-\frac{1}{2}<k_0\leqslant a_0\frac{n}{r}+\frac{1}{2}$ est minorée par

$$\frac{4}{\pi^2} \simeq 0,4$$ [5]

Avec probabilité constante, on obtient alors k_0 tel que

$$\left|\frac{k_0}{n}-\frac{a_0}{r}\right| \leqslant \frac{1}{2n}$$

Nous aimerions trouver $\frac{a_0}{r}$, c'est-à-dire une fraction de dénominateur inférieur à $\sqrt{n}$ et qui est à moins de $\frac{1}{2n}$ de $\frac{k_0}{n}$. Or deux fractions différentes de dénominateur au plus $\sqrt{n}$ sont séparées par au moins

$$\left|\frac{a}{b}-\frac{c}{d}\right| = \left|\frac{ad-bc}{bd}\right| \geqslant \frac{1}{bd} \geqslant \frac{1}{n}$$

Donc il y a au plus une fraction $\frac{a_0}{r}$ convenable. La théorie des fractions continues nous apprend qu'il existe une telle fraction, et qu'on peut la trouver en un nombre d'étapes polynomial en $\log n$. Nous obtenons ainsi une fraction vérifiant les conditions précédentes, donc cette fraction est $\frac{a_0}{r}$. Mais nous l'avons sous forme réduite, donc nous obtenons des diviseurs de r. Cependant, si a_0 et r sont premiers entre eux, on trouve r. La probabilité que a_0 et r soient premiers entre eux est $\frac{\phi(r)}{r} > \frac{\delta}{\log\log r}$ pour une constante δ, où $\phi(r) = |\{k<r ; \mathrm{pgcd}(r,k)=1\}|$ (ce résultat est démontré par Hardy et Wright dans *An introduction to the theory of numbers*).

Ainsi, il suffit de répéter $\mathcal{O}(\log\log n)$ fois la procédure pour avoir une bonne probabilité de trouver r. En tout, la procédure reste polynomiale en $\log n$. Cette procédure a été programmée avec Scilab et est présentée en annexe.

2.6 Trouver la période, cas difficile

Nous avons considéré dans la partie précédente que la période de la fonction était plus petite que $\sqrt{n}$. Cela nous permettait d'utiliser l'algorithme des fractions continues pour trouver la période. Il est légitime de se demander ce qui se passe lorsque la période est de l'ordre de n (ou au moins supérieure à $\sqrt{n}$). Je n'ai trouvé aucune référence bibliographique sur ce problème : je ne sais toujours pas s'il est possible de trouver la période efficacement dans ces conditions.

J'ai tenté plusieurs approches sans succès. Elles adaptent la méthode précédente à notre problème. Si l'on sait trouver la période lorsque celle-ci est comprise entre $\frac{n}{2}$ et n, alors on pourra résoudre le problème pour une période quelconque. En effet, on cherche d'abord si notre période est comprise entre $\frac{n}{2}$ et n ; si ce n'est pas le cas, on utilise le même algorithme pour la chercher entre $\frac{n}{4}$ et $\frac{n}{2}$, et ainsi de suite, entre $\frac{n}{2^{k+1}}$ et $\frac{n}{2^k}$. En un nombre logarithmique d'étapes on trouve la période. Il nous reste alors à résoudre le problème de la période r lorsque $\frac{n}{2}<r<n$.

En fait, par la même technique, on peut se réduire aux instances pour lesquelles $\alpha n < r < \beta n$ avec $\frac{1}{2}<\alpha<\beta<1$. Si l'on sait que la période est dans cet intervalle, alors avec probabilité constante, on obtient, à la place de l'état 2.2 (p. 14) :

$$\frac{1}{\sqrt{2}}(|i_0\rangle + |i_0+r\rangle)$$

i_0 est aléatoire, et on aimerait trouver r à partir de cet état. En appliquant la transformation de Fourier quantique, nous n'obtenons rien d'intéressant. En effet, l'état 2.3 devient :

$$\frac{1}{\sqrt{2n}}\sum_{j=0}^{n-1} e^{\frac{2i\pi i_0 j}{n}}(1+e^{\frac{2i\pi r j}{n}})\,|j\rangle$$

[5]Une autre version des calculs peut être trouvée dans [Sho96].

donc la probabilité de mesurer j est $\frac{1}{n}(1+\cos\frac{2\pi rj}{n})$. Cette probabilité dépend de r : on a plus de chance de mesurer j tel que $n|rj$, mais cela ne suffit pas pour conclure.

Une autre approche consiste à répéter la fonction : on étudie alors g définie sur $[0, n^2-1]$ par $g(kn+i) = f(i)$. On peut espérer trouver la période, car elle sera suffisamment petite pour appliquer l'algorithme du paragraphe 2.5. Cependant, avec cette méthode, on trouve n et non r. Pour éviter cet inconvénient, on peut à chaque répétition "décaler" la fonction : au lieu d'avoir la nouvelle fonction g vérifiant $g(kn+i) = g(i)$, on aurait $g(kn+i+\delta_k) = g(i)$, avec δ_k aléatoire tel que $\delta_{k+1} \geqslant \delta_k$. Mais cela n'aboutit pas non plus.

Après ces considérations, on peut se demander si notre problème peut se résoudre efficacement. On peut essayer de réduire le problème de la recherche dans une base de donnée désordonnée (Grover, 1996) au notre, car on connaît une borne inférieure à ce problème ($\Omega(\sqrt{n})$ dans une base de données de taille n). Mais cela n'a pas l'air simple. On peut aussi espérer réduire un problème NP-complet, mais ce n'est pas plus évident.

3 Problème du sous-groupe caché

3.1 Problème, motivation

Définition 3.1 (Période)
Soient G un groupe fini, et $H \leqslant G$ un sous-groupe. Soit une fonction $f : G \longrightarrow S$, avec S ensemble quelconque. On dit que f est *H-périodique* si

$$\forall x \in G, \forall h \in H, f(xh) = f(x)$$

Définition 3.2 (Problème du sous-groupe caché)
Le *problème du sous-groupe caché* est le problème suivant : soient G un groupe fini et S un ensemble fini.
- Entrée : une fonction $f : G \longrightarrow S$, H-périodique avec $H \leqslant G$, prenant une valeur différente sur chaque classe de conjugaison xH.
- Sortie : le sous-groupe H.

Le but, à partir de f, est de trouver le sous-groupe H "caché" par f.

Le problème du sous-groupe caché est une généralisation du problème de la période d'une fonction $f : \{0, \ldots, n-1\} \longrightarrow S$. En effet, nous considérons maintenant une fonction partant d'un groupe quelconque, la notion de période étant étendue elle aussi.

Le problème du sous-groupe caché est un problème intéressant, car il est au moins aussi difficile que le problème de l'isomorphisme de graphes. Or ce dernier problème est candidat à être dans NPI (la classe des langages intermédiaires, non vide si P $\neq$ NP) : personne n'a montré qu'il est dans P, et personne n'a montré qu'il est NP-complet. Si en informatique quantique on parvenait à résoudre le problème du sous-groupe caché en un temps polynomial, ce serait encore un argument en faveur de la puissance quantique.

Voyons comment on peut réduire le problème de l'isomorphisme de graphes à notre problème du sous-groupe caché. Le problème de l'isomorphisme de graphes est celui de savoir si deux graphes sont isomorphes, *i.e.* s'il existe une permutation de l'ensemble des sommets du premier dans l'ensemble des sommets du second préservant les arêtes. Ce problème est équivalent à celui de trouver l'ensemble des automorphismes d'un graphe (l'ensemble des permutations le laissant invariant). En effet, lorsque l'on a nos deux graphes G_1 et G_2, chacun de n sommets, on crée le graphe G comme étant l'union disjointe de G_1 (dont les sommets sont numérotés de 1 à n) et G_2 (dont les sommets sont numérotés de $n+1$ à $2n$). Si les deux graphes sont isomorphes, le groupe des automorphismes de G contient une permutation envoyant $\{1, \ldots, n\}$ sur $\{n+1, \ldots, 2n\}$ et $\{n+1, \ldots, 2n\}$ sur $\{1, \ldots, n\}$. On peut facilement vérifier cela, et donc savoir si G_1 et G_2 sont isomorphes.

Or le problème du sous-groupe caché sur le groupe des permutations $\mathcal{S}_n$ peut précisément nous fournir le groupe des automorphismes d'un graphe G : il suffit de trouver le sous-groupe caché par la fonction

$$f : \begin{array}{ccc} \mathcal{S}_n & \longrightarrow & \mathcal{G}_n \\ \sigma & \longmapsto & \sigma(G) \end{array}$$

où $\mathcal{G}_n$ désigne l'ensemble des graphes à n sommets. Ainsi, si nous savons résoudre le problème du sous-groupe caché sur $\mathcal{S}_n$, nous savons aussi résoudre le problème de l'automorphisme de graphe, et donc celui de l'isomorphisme de graphes.

Actuellement, nous savons résoudre le problème du sous-groupe caché en temps quantique polynomial lorsque le groupe est abélien. On sait également le résoudre pour quelques instances particulières de groupes non-abéliens, mais pas dans le cas général (et pas pour le groupe des permutations $\mathcal{S}_n$). Nous allons maintenant voir comment résoudre le problème du sous-groupe caché pour les instances abéliennes. Cette résolution s'appuie sur la transformation de Fourier quantique sur les groupes abéliens. Pour être capable d'effectuer la transformation de Fourier quantique sur tout groupe abélien, nous devons tout d'abord être en mesure de résoudre le problème du sous-groupe caché sur $\mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_k}$. Cet algorithme nécessite la transformation de Fourier quantique sur $\mathbb{Z}_n$ que nous avons vue. Grâce à cet algorithme, nous pourrons décomposer le groupe, puis faire la transformation de Fourier quantique dessus. Enfin, grâce à la transformation de Fourier quantique sur un groupe abélien quelconque, nous pourrons résoudre le problème du sous-groupe caché sur tout groupe abélien.

Nous aurons besoin de la définition suivante.

Définition 3.3 (Orthogonal)
Soit H un sous-ensemble d'un groupe abélien G. *L'orthogonal* de H, noté $H^\perp$, est l'ensemble

$$H^\perp = \{g \in G; \forall h \in H, \chi_h(g) = 1\}$$

3.2 Problème du sous-groupe caché sur groupe abélien fini

Soient G un groupe abélien fini, et S un ensemble fini. Nous avons une fonction $f : G \longrightarrow S$ (donnée sous la forme d'une boîte noire U_f), H-périodique (avec $H \leqslant G$), et nous cherchons H. Ce problème étant une généralisation du problème de la période sur $\{0, \ldots, n-1\}$, raisonnons de la même façon. On note la loi de groupe additivement (groupe abélien). On suppose que l'on sait faire la transformation de Fourier quantique efficacement sur G.

Partant de la superposition uniforme

$$\frac{1}{\sqrt{n}} \sum_{x \in G} |x\rangle\, |0\rangle$$

nous appliquons U_f pour obtenir

$$\frac{1}{\sqrt{n}} \sum_{x \in G} |x\rangle\, |f(x)\rangle = \frac{1}{\sqrt{n}} \sum_{t \in T} \sum_{h \in H} |t+h\rangle\, |f(t)\rangle$$

où T est un ensemble "transversal", *i.e.* contenant exactement un représentant de chaque classe de conjugaison de ${}^{G}/_{H}$. Si nous mesurons $f(t_0)$ dans le second registre, nous obtenons, dans le premier registre :

$$\frac{1}{\sqrt{|H|}} \sum_{h \in H} |t_0 + h\rangle$$

Appliquons maintenant la transformation de Fourier discrète sur G :

$$\frac{1}{\sqrt{n|H|}} \sum_{h \in H} \sum_{y \in G} \chi_{t_0+h}(y)\, |y\rangle = \frac{1}{\sqrt{n|H|}} \sum_{y \in G} \chi_{t_0}(y) \left(\sum_{h \in H} \chi_h(y) \right) |y\rangle$$

Appelons $S(y) = \sum_{h \in H} \chi_h(y)$. Comme dans la démonstration de la proposition 2.3, s'il existe $h_0 \in H$ tel que $\chi_{h_0}(y) \neq 1$, alors on applique la permutation $\begin{array}{ccc} H & \longrightarrow & H \\ h & \longmapsto & h_0 + h \end{array}$, et on a $S(y) = \chi_{h_0}(y) S(y) = 0$.

Ainsi, par définition de l'orthogonal, l'état que l'on a obtenu est

$$\frac{1}{\sqrt{\frac{n}{|H|}}} \sum_{y \in H^\perp} \chi_{t_0}(y)\, |y\rangle$$

Donc si nous mesurons le registre, nous obtenons avec probabilité uniforme un élément de $H^\perp$. Or nous savons trouver (classiquement et efficacement) une base de H lorsque l'on a une base de $H^\perp$. Il suffit alors de montrer que l'on obtient rapidement une base de $H^\perp$ lorsqu'on itère la procédure pour avoir plusieurs éléments $y \in H^\perp$. Montrons donc que la probabilité de tirer un élément de $H^\perp$ indépendant des premiers est grande. Plus précisément, supposons que $H^\perp$ ait une base de taille k. Montrons que lorsque l'on a déjà tiré $l < k$ générateurs $(y_1, \ldots, y_l)$, la probabilité de tirer $y_{l+1} \in H^\perp$ qui ne soit pas dans le groupe $\langle y_1, \ldots, y_l \rangle$ engendré par les l premiers est grande. On tire les y uniformément, donc la probabilité d'en tirer un indépendant est $1 - \frac{|\langle y_1, \ldots, y_l \rangle|}{|H^\perp|} \geqslant 1 - \frac{1}{k-l+1} = \frac{k-l}{k-l+1}$. Cette minoration vient du fait que $|\langle y_1, \ldots, y_l \rangle|$

divise $|H^\perp|$, et que si $l' < l$, alors $|\langle y_1, \dots, y_{l'}\rangle| < |\langle y_1, \dots, y_l\rangle|$. Ainsi, le nombre moyen de tirages qu'il faut faire pour trouver y_{l+1} indépendant des l premiers est inférieur à

$$\sum_{n \geqslant 1} \frac{k-l}{k-l+1} \left(\frac{1}{k-l+1}\right)^{n-1} n = \frac{k-l+1}{k-l}$$

et en tout, le nombre moyen pour trouver les k générateurs est inférieur à

$$\sum_{l=0}^{k-1} \frac{k-l+1}{k-l} \leqslant 2k$$

Or le nombre de générateurs de $H^\perp$ est $\mathcal{O}(\log n)$, si $n = |G|$, donc la procédure est en temps logarithmique.

Ainsi, nous avons résolu le problème du sous-groupe caché sur G, groupe abélien fini quelconque, à condition que nous sachions faire la transformation de Fourier quantique sur G.

3.3 Groupe abélien fini

3.3.1 Apprendre un groupe abélien fini

Définition 3.4 (Apprendre)
Soit G un groupe abélien fini. On sait qu'il peut se décomposer sous la forme $G \approx \mathbb{Z}_{d_1} \oplus \dots \oplus \mathbb{Z}_{d_k}$. *Apprendre le groupe* consiste à trouver $d_1, \dots, d_k$, et l'isomorphisme correspondant.

Supposons que G, groupe abélien fini, nous soit donné sous la forme d'un ensemble de générateurs $\{v_1, \dots, v_k\}$, d'une boîte noire permettant de calculer la somme (nous sommes dans un groupe abélien) de deux éléments

$$U_G : |x\rangle\,|y\rangle \mapsto |x\rangle\,|x+y\rangle$$

et que nous ayons une borne supérieure N sur le cardinal de G, *i.e.* $|G| \leqslant N$. Voyons comment nous pouvons apprendre le groupe G.

Remarque Il nous faut faire une remarque sur la façon dont U_G est implémentée. On code les éléments de G sous la forme de nombres binaires. Il se peut donc que certains nombres binaires ne représentent pas un élément de G. Pour palier à cet inconvénient, il nous faut une autre fonction qui permette de savoir si $x \in G$, soit

$$V_G : |x\rangle\,|y\rangle_1 \mapsto |x\rangle\,|y \oplus \delta_G(x)\rangle$$

où δ_G est la fonction caractéristique de G, définie par

$$\delta_G(x) = \begin{cases} 0 & \text{si } x \notin G \\ 1 & \text{si } x \in G \end{cases}$$

De plus, il se peut également que deux nombres binaires représentent un même élément (mais deux éléments ont deux codages binaires distincts, évidemment). Cela n'est pas gênant, si toutefois on a une condition supplémentaire sur U_G : $|\{$ codage de $U_G(a,b) | a$ codage de $|x\rangle$, b codage de $|y\rangle\} = 1$, donc quels que soient les codages de $x, y \in G$, $U_G(|x\rangle\,|y\rangle)$ est toujours codé de la même façon (une sorte de "codage canonique" pour chaque élément de G).

Remarque Le modèle du groupe donné sous la forme d'une boîte noire est le modèle le plus général qui soit. La borne sur le cardinal du groupe est une information nécessaire pour apprendre le groupe de façon quantique, avec l'algorithme présenté ici.

Tout d'abord, trouvons l'ordre des générateurs. Pour cela, il suffit de trouver la période de la fonction

$$\begin{array}{rccc} f_i : & \{0, \dots, N^2-1\} & \longrightarrow & G \\ & j & \longmapsto & jv_i \end{array}$$

En effet, la période de f_i est exactement l'ordre de v_i. Cette période est inférieure à N, donc on peut la trouver par l'algorithme présenté dans le paragraphe 2.5. Maintenant, nous disposons donc des ordres a_i des générateurs v_i.

Considérons la fonction

$$\begin{array}{rccc} \phi : & \mathbb{Z}_{a_1} \times \dots \times \mathbb{Z}_{a_k} & \longrightarrow & G \\ & (m_1, \dots, m_k) & \longmapsto & \sum_{i=1}^k m_i v_i \end{array} \quad [6]$$

[6] Nous considérons G comme $\mathbb{Z}_{a_1} \times \dots \times \mathbb{Z}_{a_k}$-module ; voir un cours d'algèbre élémentaire, comme [Lan93].

ϕ est un morphisme, donc est $(\ker \phi)$-périodique, et prend des valeurs distinctes sur chaque classe de conjugaison. De plus, nous savons faire la transformation de Fourier quantique sur le groupe $\mathbb{Z}_{a_1} \times \cdots \times \mathbb{Z}_{a_k}$, d'après le lemme du paragraphe 2.3. Il suffit de faire la transformation de Fourier quantique sur chaque groupe $\mathbb{Z}_{a_i}$. Ainsi, comme nous l'avons vu précédemment, nous pouvons trouver $\ker \phi$ grâce au problème du sous-groupe caché.

Lorsque l'on a des générateurs de $\ker \phi$, on peut facilement trouver la décomposition cherchée : à partir de la matrice dont les colonnes sont les vecteurs générateurs, on applique l'algorithme de Gauss, pour obtenir sur la diagonale de la matrice les d_i.

3.3.2 Transformation de Fourier quantique sur un groupe abélien fini

Si l'on se donne un groupe abélien fini G quelconque, comme dans le paragraphe précédent (modèle de la boîte noire), on peut apprendre le groupe pour trouver une décomposition $G \approx \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_k}$. À partir de cette décomposition, on peut effectuer la transformation de Fourier quantique, comme au lemme du paragraphe 2.5 : il suffit de faire la transformation de Fourier quantique sur chaque $\mathbb{Z}_{d_i}$. De cette façon, nous avons réalisé la transformation de Fourier quantique sur G.

3.3.3 Problème du sous-groupe caché sur groupe abélien fini

Maintenant que nous savons faire la transformation de Fourier quantique sur un groupe abélien fini quelconque, donné selon le modèle de boîte noire, il suffit d'appliquer l'algorithme du sous-groupe caché présenté dans le paragraphe 3.2 : nous résolvons ainsi le problème du sous-groupe caché sur tout groupe abélien fini.

Théorème 1 Soit G un groupe abélien fini de cardinal n, S un ensemble quelconque et $f : G \longrightarrow S$ une fonction H-périodique, donnée sous la forme d'une boîte noire, pour un $H \leqslant G$. On peut trouver H en temps polynomial en $\log n$, *i.e.* résoudre le problème du sous-groupe caché sur tout groupe abélien fini en temps logarithmique.

Conclusion

Nous avons résolu le problème du sous-groupe caché lorsque le groupe est abélien, en temps quantique polynomial. Aucun algorithme classique n'est connu pour le résoudre en temps polynomial. Cependant, le véritable enjeu du problème du sous-groupe caché sont les instances non-abéliennes, notamment le groupe des permutations $\mathcal{S}_n$. En effet, résoudre le problème du sous-groupe caché sur $\mathcal{S}_n$ permettrait de résoudre l'isomorphisme de graphes, problème candidat à être dans la classe NPI des langages intermédiaires (si elle n'est pas vide).

Des recherches se font dans cette direction ; on sait notamment faire la transformation de Fourier quantique sur $\mathcal{S}_n$ (*cf* [Bea97]), mais dans le cas non-abélien, la transformation de Fourier quantique ne suffit pas à résoudre le problème du sous-groupe caché. Pour d'autres instances non-abéliennes, on sait le résoudre sur les groupes dihédraux de la forme $(\mathbb{Z}_p)^n \rtimes \mathbb{Z}_2$, où p est premier (*cf* [FMSS02]), et peut-être bientôt sur $(\mathbb{Z}_{p^k})^n \rtimes \mathbb{Z}_2$.

Ces avancées démontrent la puissance de l'informatique quantique, discipline théorique apportant une vision nouvelle des algorithmes. Beaucoup d'algorithmes quantiques utilisent la transformation de Fourier quantique, car elle s'effectue très efficacement sur une machine quantique, comme nous l'avons vu. D'autres algorithmes, comme celui de Grover (1996), qui permet de rechercher un élément dans une base de données désordonnée en temps $\mathcal{O}(\sqrt{n})$, apportent également une accélération sur le cas classique.

Références

[Bea97] Robert Beals. Quantum computation of fourier transforms over symmetric groups. Décembre 1997.

[FMSS02] Katalin Friedl, Frédéric Magniez, Miklos Santha, and Pranab Sen. Quantum testers for hidden group properties. Juin 2002. Manuscript.

[Hir01] Mika Hirvensalo. *Quantum Computing*. Natural Computing Series. Springer, 2001.

[Hø00] Peter Høyer. Simplified proof of the fourier sampling theorem. Mai 2000.

[Lan93] Serge Lang. *Algebra*. Addison-Wesley, third edition, 1993.

[Pre98] John Preskill. Lecture notes for physics : Quantum information, 1998.

[RP00] Eleanor Rieffel and Wolfgang Polak. An introduction to quantum computation for non-physicists. *quant-ph/9809016*, Janvier 2000.

[Sho96] Peter Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *quant-ph/9508027*, Janvier 1996.

[Sim94] D. Simon. On the power of quantum computation. *FOCS*, 1994.

Annexes

Programme Scilab pour trouver la période

Voici un programme permettant de trouver la période par l'algorithme de Shor.

```
function [a,b] = fracont(x,t) // Algorithme des fractions continues
 a0 = 1 ; a1 = floor(x) ;  b0 = 0 ; b1 = 1 ;  e = x - a1 ;
 while ((b1 <= t) & (e <> 0))
  q = floor(1/e) ; e = 1/e - q ;
  a2 = a1*q + a0 ; b2 = b1*q + b0 ;
  a0 = a1 ; a1 = a2 ; b0 = b1 ; b1 = b2 ;
 end ;
 if (b1 <= t)
  a = a1 ; b = b1 ;
 else
  a = a0 ; b = b0 ;
 end ;
endfunction ;

// Définition des constantes
N = 1000 ; r = 28 ; M = 2*sqrt(N) ;
n = 2*ceil(log(log(N))) ; s = zeros(1,N) ;

// Simulation de la mesure du registre
i = floor(rand()*(N-r-1))+1 ;
l = floor((N-i)/r) ;
// s contient 1 dans les cases i+j*r
for j = 0:l
 s(j*r+i) = 1 ;
end ;

t = fft(s,1) ; // FFT de s
norme = sum(abs(t)^2) ; // Pour la normalisation du tableau

// log(log(N)) mesures aléatoires
trouve = 0 ;
for k2 = 1:n
 u = rand()*norme ; v = 0 ; i = 1 ;
 while v<u
  v = v+abs(t(i))^2 ; i = i+1 ;
 end ;
 // Trouver la période à partir des mesures
 [c,d] = fracont((i-2)/N,M) ;
 if d==p then trouve = 1 ; end ;
end ;

if trouve==1 then
 printf("Période trouvée !") ; end ;
```

Fixons $N = 1000$ et $r = 28$, et faisons dix essais du programme précédent. On obtient :

```
Période trouvée !
Période trouvée !
Période trouvée !
Période trouvée !
Période trouvée !
Période trouvée 5 fois sur 10
```

On a souvent trouvé la bonne valeur de la période r. Nous ne la trouvons pas à chaque fois car la probabilité de mesurer un k_0 convenable (proche d'un pic) n'est pas 1. Mais nous obtenons la bonne valeur de la période avec probabilité constante.