Projet de Fin d'Études

Étudiante : Roxane LEDUC

Encadrants : Stéphane LOUISE et Arnaud KNIPPEL

16 janvier 2024

Étude de factorisation sur QPU

Résumé

Ce projet a été réalisé dans le cadre de ma dernière année de formation à l'INSA Rouen Normandie, en Mathématiques Appliquées, sous la supervision de messieurs Stéphane LOUISE (directeur de recherche au CEA Paris-Saclay) et Arnaud KNIPPEL (enseignant chercheur en mathématiques appliquées à l'INSA Rouen Normandie). Il traîte des problèmes de factorisation par méthodes quantiques.

La première partie de ce rapport impliquera une exploration détaillée de la littérature sur les fondamentaux de l'algorithmique quantique. Ensuite, dans un second temps, nous nous concentrerons sur la compréhension de l'algorithme de Shor et sur la manière de l'implémenter sur des QPUs à portes logiques.

Table des matières

1 Introduction

L'informatique quantique, véritable révolution technologique en gestation, émerge comme l'une des avancées les plus prometteuses du XXIe siècle. Alors que l'informatique classique a fait d'immenses progrès au fil des décennies, elle atteint progressivement les limites de sa capacité à résoudre certains problèmes complexes. C'est dans ce contexte que l'informatique quantique se profile comme une lueur d'espoir, offrant des perspectives radicalement nouvelles pour résoudre des défis informatiques insurmontables jusqu'à présent.

Le concept d'informatique quantique repose sur la mécanique quantique, une théorie fondamentale de la physique qui décrit le comportement des particules subatomiques. Cette théorie a révolutionné notre compréhension du monde physique, et elle est désormais en passe de bouleverser notre approche de l'information et du calcul. Contrairement à l'informatique classique, qui utilise des bits comme unités de données pouvant être soit 0, soit 1, l'informatique quantique exploite des qubits, des unités de données quantiques qui peuvent être à la fois 0 et 1 en même temps, grâce au phénomène de superposition. Cette propriété unique permet aux ordinateurs quantiques de traiter simultanément un grand nombre de solutions potentielles, ouvrant ainsi la voie à des calculs beaucoup plus rapides pour des problèmes complexes tels que la factorisation d'entiers, la simulation de molécules ou la recherche optimale.

Dans un monde de plus en plus dépendant de l'informatique, l'informatique quantique promet de résoudre des problèmes insolubles jusqu'à présent, d'accélérer les découvertes scientifiques et de révolutionner les industries clés telles que la finance, la santé, la logistique et bien d'autres. Cependant, cette révolution technologique suscite également des questions éthiques, légales et de sécurité qui méritent une attention particulière.

La première phase de notre projet consistera en une étude bibliographique approfondie des principes généraux de l'algorithmique quantique.

La seconde, quant à elle, impliquera la découverte de l'algorithme de Shor, suivi de sa méthode d'implémentation sur un processeur quantique universel (QPU) à base de portes logiques. Nous analyserons les principes mathématiques sous-jacents qui permettent à cet algorithme révolutionnaire de factoriser rapidement des nombres entiers, ce qui a des implications majeures en cryptographie et en sécurité informatique. De plus, nous mettrons en œuvre un prototype simple de l'algorithme de Shor en utilisant un simulateur quantique.

2 Principes de l'algorithmie quantique

2.1 Qu'est ce qu'un ordinateur quantique ?

Un peu de physique... Dans un ordinateur quantique, il est essentiel de maintenir un contrôle strict sur le système physique où les bits logiques sont encodés. Toute interaction non contrôlée, même minime, peut entraîner des erreurs potentiellement préjudiciables pour le bon fonctionnement de l'ordinateur quantique. Ces interactions indésirables incluent des influences extérieures, telles que des particules d'air perturbant le système porteur des bits, ou l'absorption de petites quantités d'énergie thermique provenant de l'environnement. Il peut également y avoir des interactions perturbatrices entre les composants internes du système physique, entre ceux qui sont essentiels pour l'encodage des bits et le calcul, et ceux qui ne le sont pas. Cette dégradation de l'importance du calcul par des interactions nuisibles est ce que l'on appelle la **décohérence**, ce qui peut être dévastateur pour les calculs quantiques.

Malgré ces défis, divers facteurs suscitent de l'espoir quant à la réalisation d'un ordinateur quantique robuste et puissant. Par exemple, la séparation significative entre les niveaux d'énergie discrets à l'échelle atomique rend l'isolation dynamique d'un système atomique relativement réalisable. Par ailleurs, il a été découvert que les erreurs induites par des interactions extérieures peuvent être corrigées efficacement si elles restent rares.

Modèle de calcul. Il est important de noter que le dispositif de calcul quantique est similaire à un dispositif de calcul classique : il a un état et cet état évolue en appliquant certaines opérations. Le modèle de calcul que nous considérerons ici est le modèle du **circuit quantique**, qui fonctionne comme suit :

— L'ordinateur quantique a un état qui est contenu dans un registre quantique et est initialisé d'une manière prédéfinie.
— L'état évolue en appliquant des opérations spécifiées à l'avance sous la forme d'un algorithme.
— À la fin du calcul, des informations sur l'état du registre quantique sont obtenues au moyen d'une opération spéciale, appelée mesure.

Nous verrons dans la suite de cette section comment fonctionnent les systèmes à 1 puis à 2-qubits afin de généraliser, par la suite, l'étude à un circuit à n-qubits.

2.2 Système 1-qubit

Un ordinateur classique fonctionne en utilisant des séquences de 0 et de 1, que l'on appelle des "*bits*". En informatique quantique, les éléments de base sont les "*qubits*", qui ont la particularité de pouvoir exister dans un état de superposition. Cela signifie qu'ils peuvent simultanément représenter à la fois les valeurs 0 et 1.

2.2.1 Qubit et états quantiques

Pour comprendre les qubits, nous partons de deux **états quantiques** fondamentaux, que nous notons $|0\rangle = \binom{1}{0}$ et $|1\rangle = \binom{0}{1}$. Ces états forment une base orthonormée d'un espace vectoriel de Hilbert (on les lit "ket 0" et "ket 1").

Un qubit est un état quantique qui peut être obtenu en combinant linéairement ces deux états de base, en suivant le **principe de superposition** : $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. Ainsi, un qubit peut être représenté par un vecteur comme suit : $\binom{\alpha}{\beta}$, où α et β (appelés communément "coefficients d'amplitudes") sont des nombres complexes soumis à la contrainte $|\alpha|^2 + |\beta|^2 = 1$.

Remarque : Le dual du vecteur $|\psi\rangle$ est noté $\langle\psi| = (\alpha^*\beta^*)$. On le lit "bra psi". Le produit de deux qubits $\langle\psi| \times |\phi\rangle$ correspond au produit scalaire hermitien $\langle\psi|\phi\rangle$ (que l'on lit "braket").

Une manière courante de représenter l'état d'un qubit consiste à utiliser un point de la surface de la **sphère de Bloch**.

Avant d'introduire cette nouvelle représentation, il est important de comprendre la notion d'**équivalence**. On dit que deux qubits $|\psi\rangle$ et $|\phi\rangle$ sont équivalents si $\exists z \in \mathbf{C} / \, |\psi\rangle = z\,|\phi\rangle$.

Ainsi, un qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ est équivalent à un qubit de la forme : $|\psi'\rangle = cos(\frac{\theta}{2})\,|0\rangle + sin(\frac{\theta}{2})e^{i\varphi}\,|1\rangle$, avec θ et φ vérifiant : $0 \leq \theta \leq \pi$ et $-\pi < \varphi \leq \pi$ (l'idée de la démonstration est la suivante : poser $\alpha = e^{-i\theta}$ et $z = \frac{e^{-i\theta}}{\|\psi\|}$). Cela permet de représenter un qubit sur la sphère de Bloch, par un un vecteur de colatitude θ et longitude φ, et un rayon 1 (voir Figure 1).

Les formules pour obtenir les coordonnées $(x, y, z) \in \mathbf{R^3}$ de ce point sont :

$$\begin{cases} x = sin(\theta)cos(\varphi) \\ y = sin(\theta)sin(\varphi) \\ \quad z = cos(\theta) \end{cases}$$

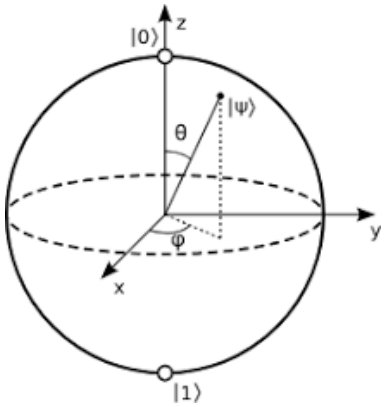

FIGURE 1 – Sphère de Bloch (Source : *Wikipedia - Bloch sphere*).

2.2.2 Mesure quantique

L'un des concepts fondamentaux de la physique quantique est que les coefficients α et β de l'état quantique $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ ne peuvent pas être mesurés directement.

La **mesure d'un qubit** $|\psi\rangle$ en état de superposition provoque sa "réduction" à une valeur binaire spécifique, avec une probabilité liée aux coefficients d'amplitude quantiques (on obtient soit le résultat 0 avec une probabilité $|\alpha|^2$, soit le résultat 1 avec une probabilité $|\beta|^2$). La mesure est un processus probabiliste qui extrait une réponse définitive du système quantique. Si la mesure renvoie 0, alors l'état $|\psi\rangle$ est instantanément modifié en $|0\rangle$, tandis que si la mesure renvoie 1, l'état $|\psi\rangle$ devient $|1\rangle$.

2.2.3 Portes quantiques

Dans un ordinateur quantique, des qubits sont produits et soumis à des transformations, appelées "portes". Ces portes sont l'équivalent des opérations logiques dans l'informatique classique. Elles modifient l'état des qubits (rotations sur la sphère de Bloch), permettant ainsi la manipulation de l'information quantique. Leurs représentations matricielles sont des matrices unitaires U ($U^t = U^{-1}$), car une opération effectuée sur des états quantiques nécessite de préserver la norme à 1. Un circuit est composé d'une succession de portes et se lit de gauche à droite.

Porte X. La porte X (ou NOT) est la transformation qui échange les deux états quantiques de base : $|0\rangle \overset{X}{\longmapsto} |1\rangle$ et $|1\rangle \overset{X}{\longmapsto} |0\rangle$. Cette transformation est linéaire et échange les deux coefficients d'un état quantique : $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \overset{X}{\longmapsto} \beta\,|0\rangle + \alpha\,|1\rangle$. En notation matricielle on a donc que : $\mathrm{X}=\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$.

Porte Y. La porte Y est la transformation définie par : $|0\rangle \overset{Y}{\longmapsto} i\,|1\rangle$ et $|1\rangle \overset{Y}{\longmapsto} -i\,|0\rangle$. En notation matricielle on a donc que : Y=$\left(\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right)$.

Porte Z. La porte Z est la transformation définie par : $|0\rangle \overset{Z}{\longmapsto} |0\rangle$ et $|1\rangle \overset{Z}{\longmapsto} -|1\rangle$. En notation matricielle on a donc que : Z=$\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$.

Remarque : Les portes X, Y et Z appartiennent à la même famille de portes, appelée **famille de Pauli**. Elles ont chacune une interprétation géométrique simple lorsque l'on observe leur action sur la sphère de Bloch. Géométriquement, $X(|\psi\rangle)$ correspond à une rotation de la sphère de Bloch autour de l'axe O_x d'un angle de π radians (soit une demi-tour). La porte Y réalise une rotation similaire autour de l'axe O_y d'un angle de π, tandis que la porte Z effectue une rotation autour de l'axe O_z d'un angle de π.

Porte H. La porte H (ou de Hadamard) est la transformation linéaire définie par : $|0\rangle \overset{H}{\longmapsto} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ et $|1\rangle \overset{H}{\longmapsto} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Ainsi, si $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, alors, $|\psi\rangle \overset{H}{\longmapsto} \frac{\alpha+\beta}{\sqrt{2}}\,|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}\,|1\rangle$.En notation matricielle on a donc que : H=$\frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$.

Remarque : Il est fréquent de rencontrer les notations suivantes : $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ et $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

2.3 Système 2-qubit

Deux qubits réunis sont dans un état quantique $|\psi\rangle$, appelé 2-qubit, défini par la superposition : $|\psi\rangle = \alpha\,|0.0\rangle + \beta\,|0.1\rangle + \gamma\,|1.0\rangle + \delta\,|1.1\rangle$, avec α, β, γ et δ complexes et la convention de normalisation, $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$.

2.3.1 Opérations et produit tensoriel

Addition. L'addition de deux qubits se fait coefficient par coefficient, il s'agit donc d'additionner des paires de nombres complexes.

Multiplication. On peut multiplier deux 1-qubits pour obtenir un 2-qubit. Les calculs se font comme des calculs algébriques à l'aide des règles de bases ($|0\rangle\,.\,|1\rangle = |0.1\rangle$, etc.). Pour les coefficients, on utilise la multiplication des nombres complexes.

Produit tensoriel. Les vecteurs nouvellement introduits ($|0.0\rangle$, $|0.1\rangle$, etc.) sont de nouveaux vecteurs d'une base mais cette fois en quatre dimension. De manière générale, le produit $|\psi.\phi\rangle$ est

défini par le produit tensoriel $|\psi.\phi\rangle = |\psi\rangle \otimes |\phi\rangle$. Le produit tensoriel de deux vecteurs $u \in \mathbf{K}^n$ et $v \in \mathbf{K}^m$, se détermine comme suit :

$$u \otimes v = \begin{pmatrix} u_1 \begin{pmatrix} v_1 \\ ... \\ v_m \end{pmatrix} \\ ... \\ u_n \begin{pmatrix} v_1 \\ ... \\ v_m \end{pmatrix} \end{pmatrix} = \begin{pmatrix} u_1 v_1 \\ ... \\ u_1 v_m \\ ... \\ u_n v_1 \\ ... \\ u_n v_m \end{pmatrix} \in \mathbf{K^{nm}}$$

2.3.2 Mesure classique et mesure partielle

Mesure classique. La mesure d'un 2-qubit renvoie :

— 0.0 avec probabilité $|\alpha|^2$,

— 0.1 avec probabilité $|\beta|^2$,

— 1.0 avec probabilité $|\gamma|^2$,

— 1.1 avec probabilité $|\delta|^2$.

Il s'agit d'une situation analogue à celle que nous avions avec un qubit, mais maintenant avec quatre possibilités.

Mesure partielle. On peut aussi choisir de ne mesurer qu'un seul qubit. Si on choit de ne mesurer que le premier qubit (cela fonctionne de manière analogue avec le second), nous obtiendrons 0 avec une probabilité $|\alpha|^2 + |\beta|^2$ (et dans ce cas là, le nouvel état de ψ sera $\frac{\alpha|0.0\rangle + \beta|0.1\rangle}{\sqrt{|\alpha|^2+|\beta|^2}}$), et, 1 avec une probabilité $|\gamma|^2 + |\delta|^2$.

2.3.3 Porte CNOT et état de Bell

La **porte CNOT** (ou Controlled-NOT) prend en entrée deux qubits et renvoie deux qubits en sortie. Si le premier qubit est $|0\rangle$, rien ne change. S'il est $|1\rangle$, le deuxième bit est inversé (et le premier reste inchangé). Elle est ainsi donnée par la matrice unitaire suivante : $M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.

La transformation associée à cette porte s'écrit aussi : $|x.y\rangle \overset{CNOT}{\longmapsto} |x.y \oplus x\rangle$, où x et y ont pour valeurs 0 ou 1 et où $\oplus$ représente l'addition usuelle sur un bit (comme une porte XOR).

À l'aide de cette porte, nous allons obtenir quatre états fondamentaux : les **états de Bell**.

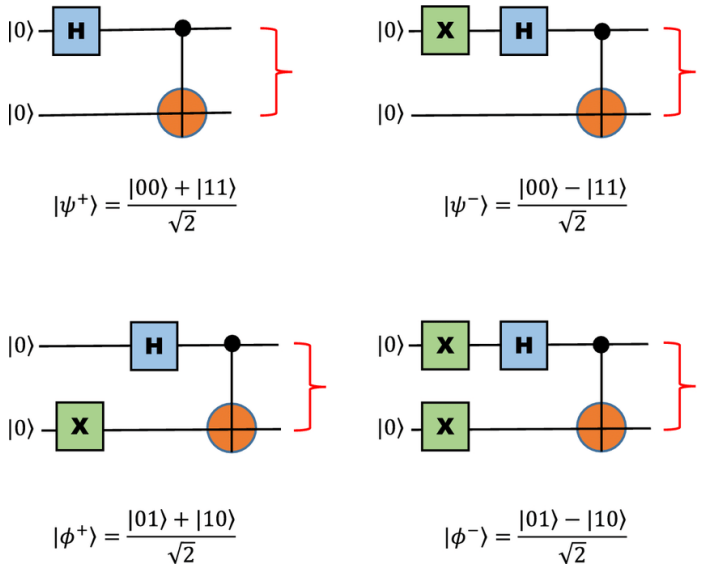

FIGURE 2 – États de Bell.

Une mesure de l'état $|\psi^+\rangle = \frac{1}{\sqrt{2}}|0.0\rangle + \frac{1}{\sqrt{2}}|1.1\rangle$ conduit par exemple à :

— 0.0 avec une probabilité 0.5,

— 1.1 avec une probabilité 0.5,

— les deux autres sorties 0.1 et 1.0 ayant une probabilité nulle.

Remarque : Ces états de Bell sont dits "intriqués", de telle sorte que la connaissance de l'état d'une particule donne immédiatement de l'information sur l'état de l'autre, quelle que soit la distance qui les sépare. Ce phénomène s'appelle "l'**intrication quantique**". Un état intriqué est un état qui n'est pas un état produit (un état $|\psi\rangle$ est un état produit s'il peut être écrit sous la forme $|\psi_1\rangle \,.\, |\psi_2\rangle$).

Il existe par ailleurs un grand nombre d'autres portes. On a par exemple : la **porte SWAP**, qui intervertit deux qubits ou encore, une porte 3-bits, la **porte de Toffoli**.

2.4 Circuit quantique

D'une façon générale, le regroupement de plusieurs qubits conduit à un "n-qubit". Voici le schéma de principe d'un circuit quantique (voir Figure 3) :

— en entrée : n qubits dont la superposition représente un n-qubit,

— une succession de portes quantiques, chacune agissant sur un ou plusieurs qubits,

— le circuit est terminé par un certain nombre de mesures, qui renvoient des bits classiques.

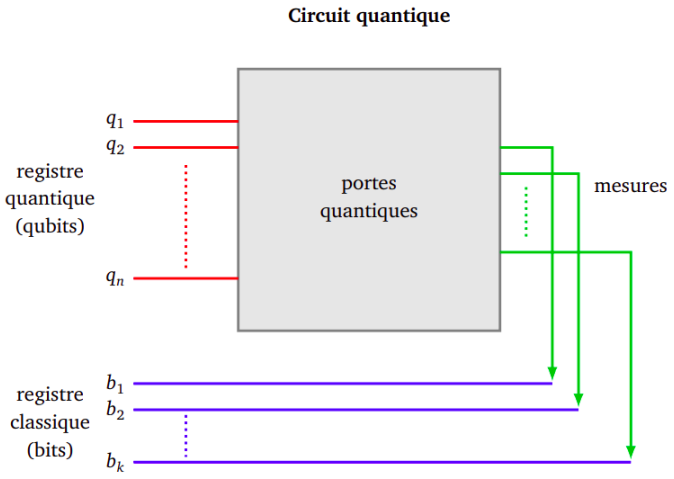

FIGURE 3 – Circuit quantique.

Un n-qubit est un état quantique caractérisé par son vecteur : $\begin{pmatrix} \alpha_0 \\ \cdots \\ \alpha_{2^n-1} \end{pmatrix} \in \mathbb{C}^{2^n}$, avec la condition de normalisation, $\sum |\alpha_i|^2 = 1$. Un n-qubit contient donc un total de 2^n coefficients.

La combinaison de n qubits entraîne une superposition de 2^n états de base distincts. Travailler avec un n-qubit signifie effectuer des opérations sur les 2^n n-bits classiques simultanément, c'est-à-dire sur des séquences telles que 0.0 . . . 0.0, 0.0 . . . 0.1, . . ., 1.1 . . . 1.1, tandis que l'informatique classique ne traite qu'un seul n-bit à la fois.

2.5 Outils de simulation quantique

Nous utilisons, dans cette section, le langage de programmation Python et la librairie *qiskit* fournie par IBM. Nous introduisons les concepts de base permettant de créer des circuits simples dans un premier temps.

Dans l'exemple ci-dessous, on part donc de l'état initial $|0\rangle$ (valeur par défaut), on applique une porte de Hadamard, puis on termine par une mesure qui renvoie un bit classique 0 ou 1 avec ici chacun la probabilité $\frac{1}{2}$.

```
# Import des modules
from qiskit import *
from qiskit.visualization import *
from qiskit.tools.monitor import *
```

```
# Creation du circuit (1 qubit et 1 bit pour le resultat de la mesure)
circ = QuantumCircuit(1,1)
# Ajout d une porte de Hadamard
circ.h(0)
# Mesure
circ.measure(range(1),range(1))
# Affichage du code quasm
print(circ.qasm())

# Execution sur un simulateur
backend_sim = Aer.get_backend('qasm_simulator')
job_sim = execute(circ, backend_sim, shots=1024) # 1024 simulations (une
# seule valeur ne permet pas de conclure sur la nature du circuit)

# Collecte des resultats et visulisation
result_sim = job_sim.result()
counts = result_sim.get_counts(circ)
print(counts)
plot_histogram(counts)
```

On obtient ainsi le type de résultats suivant : {'0' : 524, '1' : 500}, que l'on peut choisir de représenter par un histogramme de fréquences.

Remarque : On peut choisir d'initialiser l'entrée par un qubit $|\psi\rangle$ quelconque (et non plus $|0\rangle$). Supposons que l'on souhaite comme qubit initial : $|\psi\rangle = (3+i)|0\rangle + (1-2i)|1\rangle$. Pour cela on définit α_0 et α_1, puis, afin que l'entrée soit acceptée, on normalise le qubit. Ce qui nous permet alors de définir le qubit initial à l'aide de la commande Initialise([alpha,beta]) :

```
# Construction du circuit
circuit = QuantumCircuit(1)
# Initialisation a la main
alpha0 = 3+1j
beta0 = 1-2j
```

```
norme = np.sqrt(abs(alpha0)**2 + abs(beta0)**2)
alpha, beta = alpha0/norme, beta0/norme
qubit_initial = extensions.Initialize([alpha,beta])
circuit.append(qubit_initial,[0])
```

On peut bien évidement choisir de complexifier les circuits en ajoutant d'autres qubits (ainsi que d'autres bits pour les mesures) et d'autres portes (il faut préciser le numéro de ligne du circuit lorsque l'on en ajoute une).

Par ailleurs, il convient de noter qu'il y a peu de temps, l'un des avantages inhérents à l'utilisation de Qiskit résidait dans la capacité d'exécuter ses programmes sur des calculateurs quantiques authentiques. Toutefois, en novembre dernier, le programme QPU public d'IBM a été interrompu, bien que l'on nourrisse l'espoir d'une éventuelle reprise de cette opportunité. Précédemment, IBM avait généreusement mis à disposition un accès aux ressources de calcul quantique, permettant l'utilisation du temps de traitement sur de véritables ordinateurs quantiques.

2.6 Codage super-dense et téléportation quantique

2.6.1 Codage super-dense

Le **codage super-dense** est un protocole quantique qui facilite l'échange d'informations entre deux parties.

Alice souhaite ici transmettre de façon sécurisée à Bob une information constituée de deux bits classiques, en envoyant un seul qubit.

Ce processus se déroule en 3 étapes : préparation de l'état de Bell, codage de l'information par Alice, et enfin, pour finir, décodage par Bob.

Le protocole commence donc par un travail de préparation externe : une troisième personne, prépare l'état de Bell suivant : $|\Psi^+\rangle = \frac{1}{\sqrt{2}}|0_A.0_B\rangle + \frac{1}{\sqrt{2}}|1_A.1_B\rangle$ (cf. Section 2.3.3). Il envoie ensuite, un premier qubit $|\psi_A\rangle = \frac{1}{\sqrt{2}}|0_A\rangle + \frac{1}{\sqrt{2}}|1_A\rangle$, à Alice et un second, $|\psi_B\rangle = \frac{1}{\sqrt{2}}|O_B\rangle + \frac{1}{\sqrt{2}}|1_B\rangle$, à Bob.

Alice applique alors l'une des quatre transformations suivantes en fonction de l'information qu'elle souhaite transmettre :

— L'identité si elle souhaite transmettre l'information 0.0 ;

— La porte X si elle souhaite transmettre l'information 0.1 ;

— La porte Z si elle souhaite transmettre l'information 1.0 ;

— La porte X puis la porte Z elle souhaite transmettre l'information 1.1.

Elle transmet ensuite le qubit transformé $|\psi'_A\rangle$ à Bob. Ce dernier reçoit donc deux qubits ($|\psi'_A\rangle$ et $|\psi_B\rangle$), toujours liés par intrication (ces états quantiques ne peuvent pas être séparés en produits tensoriels des états des qubits individuels). Pour finir, Bob applique une porte CNOT suivi d'une porte H, puis, mesure les deux qubits.

2.6.2 Téléportation quantique

La **téléportation quantique** est une procédure permettant de déplacer un qubit depuis un emplacement initial (point A) vers une destination (point B).

À la différence du codage super-dense, où Bob reçoit un message d'Alice offrant uniquement des options limitées, la téléportation quantique offre à Bob la réception d'un qubit représenté par $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, avec une gamme infinie de possibilités. Cette infinie variabilité découle du fait que les coefficients α et β sont des nombres complexes.

L'idée de ce processus est la suivante : d'un côté, Alice possède un 1-qubit $|\psi\rangle$. De l'autre, on a préparé un 2-qubit à l'état de Bell $|\Psi^+\rangle$ (cf. Section 2.3.3). On réalise ensuite un 3-qubit $|\phi\rangle = |\psi\rangle \otimes |\Psi^+\rangle$. Alice réalise alors une mesure partielle des deux premiers qubits de ce 3-qubit dans la base de Bell et obtient une information classique (on applique sur les deux premiers qubits une porte CNOT, suivie d'une porte de Hadamard, suivie de deux mesures). Pour finir, Bob reconstitue le qubit initial à partir de cette information que lui transmet Alice et du troisième qubit du 3-qubit. La solution finale est le troisième qubit du 3-qubit transformé :

— Si l'information reçue est 0.0, il applique l'identité I au troisième qubit de $|\phi\rangle$;

— Si l'information reçue est 0.1, il applique une porte X au troisième qubit de $|\phi\rangle$;

— Si l'information reçue est 1.0, il applique une porte Z au troisième qubit de $|\phi\rangle$;

— Si l'information reçue est 1.1, il applique une porte X puis une porte Z au troisième qubit de $|\phi\rangle$.

2.7 Algorithmes quantiques

Les algorithmes de Deutsch et de Grover, bien que distincts dans leurs objectifs et leurs domaines d'application, incarnent tous deux des percées significatives dans le domaine de l'informatique

quantique.

L'algorithme de Deutsch, tout d'abord, se penche sur la question de la détermination de la nature d'une fonction booléenne, à savoir si elle est constante ou équilibrée. Bien que sa pertinence apparente puisse sembler limitée, il détient la clé pour comprendre comment les avantages quantiques peuvent être exploités pour résoudre des problèmes plus complexes de manière exponentiellement plus efficace que les ordinateurs classiques.

De même, l'algorithme de Grover se distingue par sa capacité à accélérer de manière spectaculaire la recherche dans une base de données non triée. Il offre un gain substantiel en termes de recherche par rapport aux méthodes classiques, ce qui le rend particulièrement intéressant.

Dans cette section, nous explorerons de manière générale les concepts sous-jacents à ces deux algorithmes. Nous avons fait le choix ne pas développer certains détails de calculs trop complexes, car ces algorithmes ne constituent pas le cœur de notre étude. Nous laissons au lecteur la possibilité de se référer à [4] s'il souhaite explorer plus en détail certains aspects techniques pour l'algorithme de Grover.

2.7.1 Algorithme de Deutsch

L'**algorithme de Deutsch** résout un problème binaire simple. Il s'agit du problème de déterminer si une fonction booléenne $f(x)$ est constante (la sortie est 0 ou 1 pour toutes les entrées) ou équilibrée (la sortie est 0 dans la moitié des cas, 1 dans les autres). Cet algorithme résout ce problème en utilisant un seul appel quantique à la fonction $f(x)$ (contrairement au cas classique qui devrait réaliser deux appels afin de calculer les deux valeurs de la fonction) et en fournissant une réponse déterministe en une seule étape.

Cet algorithme quantique est mis en œuvre par un circuit quantique représenté dans la Figure 4. Il fait usage de portes de Hadamard H ainsi que d'un sous-circuit O_f appelé "**oracle**". L'oracle d'une fonction f est un circuit quantique dont on explicite seulement l'entrée et la sortie.

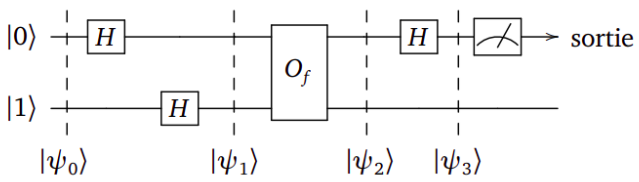

FIGURE 4 – Algorithme de Deutsch.

L'oracle prend le bit x en entrée sur la première ligne du circuit et renvoie la même valeur x en sortie. En revanche, sur la seconde ligne, l'oracle reçoit le bit y, mais sa sortie dépend à la fois des valeurs de x, y, et de la fonction f. La sortie de l'oracle est soit le bit 0, soit le bit 1, en accord avec la formule $y \oplus f(x)$, qui équivaut à l'opération XOR. Il est ainsi possible de définir une fonction sur les 2-qubits à travers l'oracle associé à f.

Voyons maintenant le détail des calculs en suivant l'évolution des qubits au fil du circuit.

Qubit $|\psi_0\rangle$ initial. $|\psi_0\rangle = |0.1\rangle$

Qubit $|\psi_1\rangle$ obtenu après transformation de Hadamard. On applique la porte H sur la première ligne ($H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$) et sur la seconde ($H|1\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$). On obtient alors :

$$|\psi_1\rangle = \tfrac{1}{2}(|0.0\rangle - |0.1\rangle + |1.0\rangle - |1.1\rangle) \equiv (|0.0\rangle - |0.1\rangle + |1.0\rangle - |1.1\rangle)$$

Qubit $|\psi_2\rangle$ obtenu après l'oracle.

$$\begin{aligned}|\psi_2\rangle &= (|0.(0 \oplus f(0))\rangle - |0.(1 \oplus f(0))\rangle + |1.(0 \oplus f(1))\rangle - |1.(1 \oplus f(1))\rangle) \\ &= (-1)^{f(0)}(|0.0\rangle - |0.1\rangle) + (-1)^{f(1)}(|1.0\rangle - |1.1\rangle)\end{aligned}$$

Remarque : Ce résultat est obtenu en déterminant ce qui se passe à chaque fois en fonction de la valeur de $f(0)$ et $f(1)$ (valeurs à 0 ou 1).

Qubit $|\psi_3\rangle$ obtenu après application d'une porte de Hadamard sur la première ligne.

$$\begin{aligned}|\psi_3\rangle &\equiv (-1)^{f(0)}(|(0+1).0\rangle - |(0+1).1\rangle) + (-1)^{f(1)}(|(0-1).0\rangle - |(0-1).1\rangle) \\ &\equiv ((-1)^{f(0)} + (-1)^{f(1)})\,|0.0\rangle + (-(-1)^{f(0)} - (-1)^{f(1)})\,|0.1\rangle + ((-1)^{f(0)} - (-1)^{f(1)})\,|1.0\rangle \\ &+ (-(-1)^{f(0)} + (-1)^{f(1)})\,|1.1\rangle\end{aligned}$$

Remarque : Le coefficient que l'on a omis devant tous les qubits est $\frac{1}{2\sqrt{2}}$ et correspond aux trois portes de Hadamard.

Conclusion. Si f est constante ($f(0) = f(1)$), on a : $|\psi_3\rangle = \pm\frac{1}{\sqrt{2}}(|0.0\rangle - |0.1\rangle)$ et donc la mesure du premier qubit donne 0. Si f est équilibrée ($f(0) \neq f(1)$), on a : $|\psi_3\rangle = \pm\frac{1}{\sqrt{2}}(|1.0\rangle - |1.1\rangle)$ et cette fois, la mesure donne 1.

Remarque : L'algorithme de Deutsch-Jozsa (fréquent dans la littérature) généralise ce problème

pour inclure des fonctions constantes ou équilibrées de tailles arbitraires.

2.7.2 Algorithme de Grover

L'**algorithme de Grover** est un algorithme de recherche d'un élément dans une liste qui est plus efficace (complexité $\mathcal{O}(\sqrt{N})$) que les algorithmes classiques (complexité $\mathcal{O}(N)$ pour une liste non-ordonnée). Il s'agit cependant d'un algorithme probabiliste, qui peut donc parfois renvoyer une solution erronée.

Comme illustré dans la Figure 5, cet algorithme repose sur le concept d'une inversion autour de la moyenne. L'objectif, dans cet exemple, est de discriminer le rang du rectangle rouge. Le processus débute en rendant négatif le coefficient du rang recherché, puis en calculant la moyenne des coefficients, suivie d'une symétrie par rapport à cette moyenne. Ces trois étapes sont répétées de manière itérative. Progressivement, le rectangle rouge augmente en taille tandis que les rectangles bleus diminuent. Après plusieurs itérations, une mesure présente très probablement le rang du rectangle rouge.

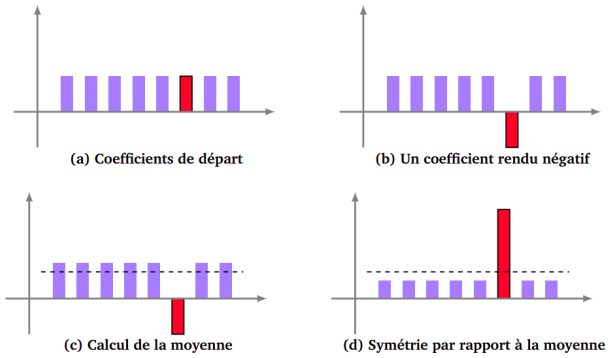

FIGURE 5 – Algorithme de Grover.

Nous pouvons par ailleurs modéliser cette recherche dans une liste (non-ordonnée) à l'aide d'une fonction mathématique $f : \{0, ..., N-1\} \to \{0, 1\}$ avec $f(k_0) = 1$ et $f(k) = 0$, $\forall k \neq k_0$. L'objectif étant donc de déterminer la valeur k_0 telle que $f(k_0) = 1$.

Remarque : On peut identifier $\{0, ..., N-1\}$ à $\mathbf{Z}/N\mathbf{Z}$ (ensemble de toutes les classes de résidus modulo N) et $\{0, 1\}$ à $\mathbf{Z}/2\mathbf{Z}$. On se place dans le cas ou $N = 2^n$.

Cet algorithme quantique est mis en œuvre par un circuit quantique dont la première partie est représentée en Figure 6.

Pour $x \in \mathbf{Z}/N\mathbf{Z}$ et $y \in \mathbf{Z}/2\mathbf{Z}$, l'oracle O_f (voir Figure 7) réalise une fonction $F(x, y) =$

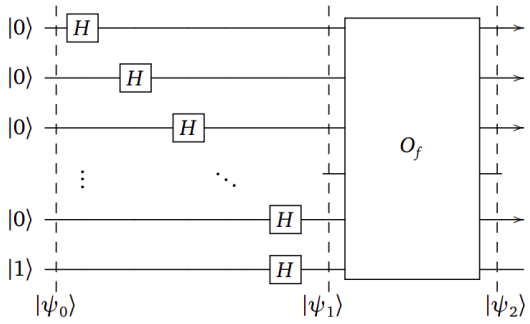

FIGURE 6 – Début de l'algorithme de Grover.

$(x_1...x_n, y \oplus f(x_1...x_n))$, avec $x_1...x_n$ l'écriture binaire de x (que l'on note dans la suite $\underline{x}$).

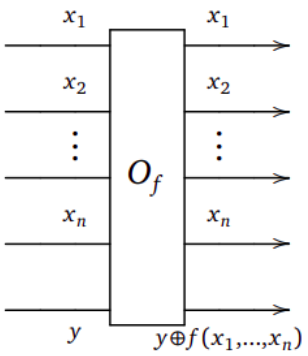

FIGURE 7 – Oracle O_f associé à la fonction f.

Voyons maintenant plus en détail l'évolution des qubits au fil du circuit.

(n+1)-qubit $|\psi_0\rangle$ initial. Le qubit en entrée est le (n+1)-qubit $|\psi_0\rangle = |0...0\rangle . |1\rangle = |\underline{0}\rangle . |1\rangle$.

(n+1)-qubit $|\psi_1\rangle$ obtenu après transformations de Hadamard.

$$|\psi_1\rangle = H^{\otimes n+1} |\psi_0\rangle = H^{\otimes n} |\underline{0}\rangle H |1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle . \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

car $|\underline{0}\rangle = |0...0\rangle, |\underline{1}\rangle = |0...0.1\rangle, ..., |\underline{2^n - 1}\rangle = |1...1\rangle$ (désignent les n-qubits de la base canonique).

Remarque : Pour la suite de l'étude, on notera $|\psi_H\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle$.

(n+1)-qubit $|\psi_2\rangle$ obtenu après l'oracle. (Détails des calculs en Annexe.)

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{f(\underline{k})} |\underline{k}\rangle . \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2^n}}(|\underline{0}\rangle ... - |\underline{k_0}\rangle ... + |\underline{2^n - 1}\rangle) . \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Remarque : On arrive bien, en une seule évaluation de l'oracle, à distinguer le terme de rang

k_0 des autres termes. L'oracle "**marque**" les états qui vérifient la condition (coefficients rendus négatifs). Il reste maintenant à déterminer précisément ce rang.

Opérateur de diffusion. On repart du qubit $|\psi_H\rangle$ et on isole le qubit $|\underline{k_0}\rangle$ (on rappelle que l'oracle change $\underline{k_0}$ en $-\underline{k_0}$ et laisse inchangé $\underline{k}$, $\forall k \neq k_0$) : $|\psi_H\rangle = \sqrt{\frac{N-1}{N}}\,|\chi\rangle + \frac{1}{\sqrt{N}}\,|\underline{k_0}\rangle$. On note : $|\chi\rangle = \frac{1}{\sqrt{N-1}} \sum_{k \neq k_0} |\underline{k}\rangle$.

On réécrit $|\psi_H\rangle$ sous une forme trigonométrique : $|\psi_H\rangle = \cos\frac{\theta}{2}\,|\chi\rangle + \sin\frac{\theta}{2}\,|\underline{k_0}\rangle$ (voir Figure 8).

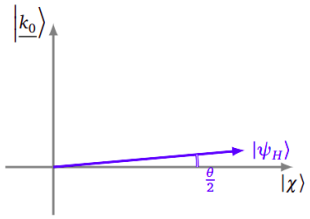

FIGURE 8 – Écriture trigonométrique de $|\psi_H\rangle$.

On applique alors un opérateur de diffusion S_{ψ_H}, qui réalise une **réflexion autour de la moyenne des amplitudes**. Il est construit à partir d'une symétrie par rapport à l'axe défini par la superposition $|\psi_H\rangle$ ($S_{\psi_H} = 2\,|\psi_H\rangle\langle\psi_H| - I$). Cet opérateur peut être décomposé de la manière suivante : $S_{\psi_H} = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n}$, ou S_0 représente une symétrie par rapport à l'axe $|0\rangle$. Le circuit associé est représenté en Figure 10.

Transformation de Grover. L'opération combinée de l'oracle O_f et de l'opérateur de diffusion S_{ψ_H} se nomme "Transformation de Grover" (porte G, qui conduit au qubit $G\,|\psi_H\rangle$). Cette transformation va en fait nous permettre de modifier le qubit $|\psi_H\rangle$ en un qubit très proche du qubit de base $|\underline{k_0}\rangle$. On va chercher à chaque étape à **amplifier** les amplitudes des états marqués. Nous terminerons par effectuer une mesure qui nous fournira, avec une forte probabilité, la valeur de k_0.

Pour représenter de manière concise la Transformation de Grover, nous utilisons en Figure 9 une représentation en deux lignes : la première correspond à un n-qubit (représenté par "/"), et la seconde à un 1-qubit.

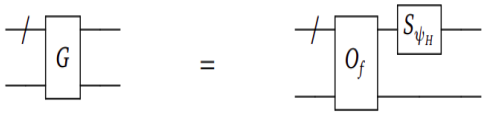

FIGURE 9 – Transformation de Grover.

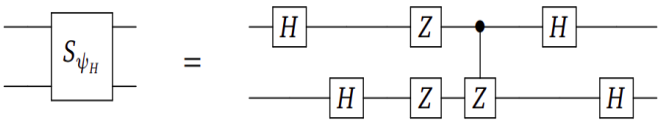

FIGURE 10 – Circuit associé à S_{ψ_H}

Cette transformation de Grover correspond en fait, géométriquement, à une rotation d'angle θ.

Elle va être répétée l fois. On obtient finalement le n-qubit $G^l |\psi_H\rangle = \cos\theta_l |\chi\rangle + \sin\theta_l |\underline{k_0}\rangle$, avec $\theta_l = \frac{\theta}{2} + l\theta$. On cherche alors à ce que $\theta_l \approx \frac{\pi}{2}$ (voir Figure 11) et ainsi, l est défini comme l'entier le plus proche de $\frac{\pi}{2\theta} - \frac{1}{2}$.

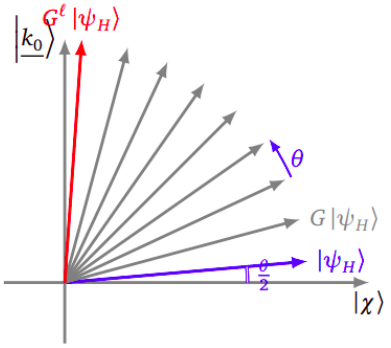

FIGURE 11 – Itérations de la transformation de Grover.

À la fin de l'algorithme, une mesure est donc effectuée sur n-qubit. Cette mesure conduit (avec de fortes chances) au n-bit $\underline{k_0}$ et permet alors d'identifier le rang k_0.

3 Algorithme de Shor

La sécurité des communications en ligne repose sur des principes mathématiques, notamment sur le système de cryptographie RSA (algorithme détaillé en annexe), qui tire sa robustesse de la complexité de la factorisation de nombres entiers très grands par des ordinateurs classiques.

Nous introduisons dans cette section l'**algorithme de Shor** qui permet de décomposer un entier N en un produit de facteurs premiers, en exploitant la puissance des ordinateurs quantiques.

Remarque : Actuellement, l'algorithme classique le plus connu (l'algorithme du crible du champ de nombres) s'exécute en un temps de $e^{\mathcal{O}(n^{1/3}log(n^{2/3}))}$, c'est-à-dire, qu'il n'existe pas d'algorithme classique algorithme classique en temps polynomial pour cette tâche. L'algorithme de Shor s'exécutera quant-à-lui en temps polynomial $\mathcal{O}(n^3)$.

3.1 Méthode

Avant d'approfondir les aspects techniques de cet algorithme, nous présentons brièvement ci-dessous les grandes étapes :

1. Vérifier que N n'est pas un nombre pair, premier ou une puissance d'un nombre premier.
2. Choisir aléatoirement $1 < a < N$.
3. Si $b = pgcd(a, N) > 1$, renvoyer b (facteur non trivial).
4. Sinon, trouver l'ordre de a modulo N... nécessité d'user d'un circuit quantique !
5. Si r est impair, retourner à l'étape 1.
6. Sinon, calculer $x \equiv a^{r/2} + 1[N]$ et $y \equiv a^{r/2} - 1[N]$. Si $x \equiv 0[N]$, retourner à l'étape 2.
7. Calculer : $p = pgcd(x, N)$ et $q = pgcd(y, N)$. Au moins l'un d'entre eux sera un facteur non trivial de N.

Remarque : On désigne par **ordre** d'un entier a modulo N, le plus petit entier r strictement positif tel que $a^r \equiv 1[N]$. Par ailleurs, cet ordre r représente la plus petite période de la fonction $k \to a^k[N]$.

Étape 1. On vérifie que N soit impair autrement nous avons immédiatement 2 comme facteur trivial et le travail est terminé. Ensuite, N doit être non-premier. On peut s'en assurer à l'aide notamment du **test de primalité probabiliste de Miller-Rabin** ou encore du **test de primalité AKS - Agrawal-Kayal-Saxena**. Nous devons aussi contrôler que N ne soit pas une puissance

d'un nombre premier car il existe un algorithme classique qui peut déterminer ce nombre en temps polynomial.

Étape 2. Au début de l'algorithme, on commence par choisir au hasard un entier a avec $1 < a < N$.

Étape 3. On calcule ensuite le *pgcd* de a et de N à l'aide de l'**algorithme d'Euclide**. Si ce pgcd est différent de 1, b est un facteur non-trivial de N et l'algorithme est terminé.

Étape 4. Sinon, on sait d'après le **théorème d'Euler** que si a et N sont premiers entre eux, alors il existe une puissance minimale $1 < r < N$ telle que $a^r \equiv 1[N]$ (ordre de a modulo N). C'est cet ordre que nous allons maintenant déterminer.

En informatique classique, cela nécessiterait de calculer séquentiellement $a^1, a^2...$ modulo N jusqu'à trouver r tel que $a^r \equiv 1[N]$. Cela impliquerait donc environ $\mathcal{O}(N)$ calculs du type a^k modulo N. C'est la raison pour laquelle nous allons ici faire appel à un circuit quantique afin de réaliser l'évaluation simultanée de toutes les valeurs.

Pour ce faire, nous allons considérer la fonction $f : k \to a^k[N]$ à laquelle nous allons associer l'oracle $F : (k, y) \to (k, y \oplus a^k[N])$. Ici, le circuit est initialisé avec $y = 0$ et est composé de deux registres. En entrée le premier registre reçoit l'entier k, codé sur n bits et même chose pour le second registre qui correspond à 0. Nous avons également deux registres en sortie, le premier renvoie k et le second $a^k[N]$.

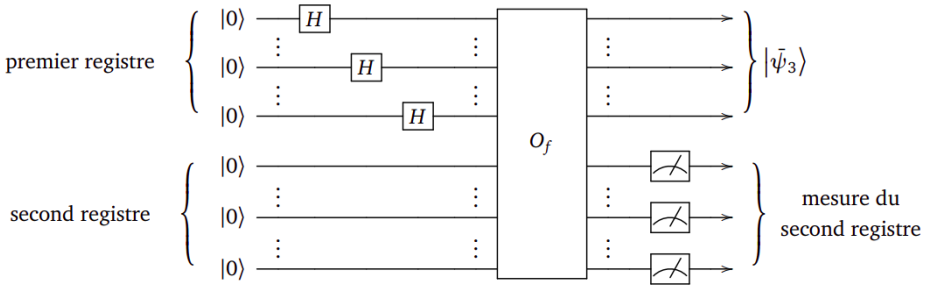

FIGURE 12 – Circuit quantique.

En effet, le circuit est dans un premier temps initialisé par des qubits égaux à $|0\rangle$. On a : $|\psi_0\rangle = |\underline{0}\rangle \otimes |\underline{0}\rangle$.

Ensuite, on applique une transformation de Hadamard sur le premier registre. On obtient : $|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes |\underline{0}\rangle$.

Après l'oracle, on a alors : $|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes \left|\underline{a^k[N]}\right\rangle$. Nous allons réécrire $|\psi_2\rangle$ en utilisant le fait que la fonction f soit périodique de période r.

Soit 2^n, la plus petite puissance de 2 supérieure à N^2 et telle que $2^n = Br + b$. Écrivons k sous la forme $\alpha r + \beta$ et posons $A = \begin{cases} \lfloor \frac{2^n}{r} \rfloor + 1, \beta < b \\ \lfloor \frac{2^n}{r} \rfloor, \beta \geq b \end{cases}$. On a alors :

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{\beta=0}^{r-1} (\sum_{\alpha=0}^{A-1} \left|\underline{\alpha r + \beta}\right\rangle) \otimes \left|\underline{a^\beta}\right\rangle$$

Nous effectuons alors une mesure du second registre, ce qui nous permet d'obtenir la mesure d'un $\left|\underline{a^{\beta_0}}\right\rangle$. Après cette mesure, le qubit du premier registre est : $\left|\bar{\psi_3}\right\rangle = \frac{1}{\sqrt{A}} \sum_{\alpha=0}^{A-1} \left|\underline{\alpha r + \beta_0}\right\rangle$.

La mesure du premier registre ne permet pas de conclure. Ainsi, après la mesure du second registre, nous allons faire agir sur le premier registre la transformée de Fourier discrète inverse $\hat{F}^{-1}$.

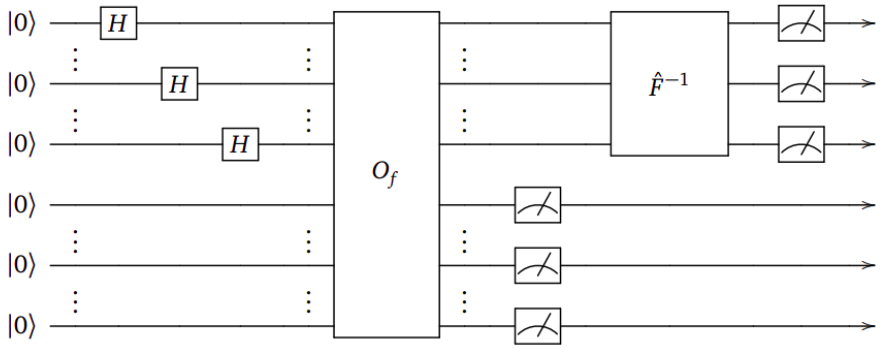

FIGURE 13 – Circuit quantique.

Remarque : La transformée de Fourier discrète $\hat{F}$ transforme un n-qubit en une somme de n-qubits selon la formule : $\hat{F}\,|\underline{k}\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{2i\pi \frac{kj}{2^n}} \left|\underline{j}\right\rangle$. Cette transformation agit en fait comme un changement de base. On passe de $\{|0\rangle, |1\rangle\}$ à $\{|+\rangle, |-\rangle\}$. Cette application est unitaire et donc inversible, et ainsi, on peut déterminer la transformée de Fourier inverse : $\hat{F}^{-1}\,|\underline{k}\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2i\pi \frac{kj}{2^n}} \left|\underline{j}\right\rangle$.

Nous allons maintenant calculer le qubit $\left|\bar{\psi_4}\right\rangle$, obtenu après l'action de la transformée de Fourier

inverse :

$$\begin{aligned}
\left|\bar{\psi}_4\right\rangle &= \hat{F}^{-1}\left|\bar{\psi}_3\right\rangle \\
&= \hat{F}^{-1}(\frac{1}{\sqrt{A}}\sum_{\alpha=0}^{A-1}\left|\underline{\alpha r+\beta_0}\right\rangle) \\
&= \frac{1}{\sqrt{A}}\sum_{\alpha=0}^{A-1}\hat{F}^{-1}(\left|\underline{\alpha r+\beta_0}\right\rangle) \\
&= \frac{1}{\sqrt{A}}\sum_{\alpha=0}^{A-1}\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}e^{-2i\pi\frac{(\alpha r+\beta_0)j}{2^n}}\left|\underline{j}\right\rangle \\
&= \frac{1}{\sqrt{A}\sqrt{2^n}}\sum_{j=0}^{2^n-1}(\sum_{\alpha=0}^{A-1}e^{-2i\pi\frac{\alpha j}{2^n/r}})e^{-2i\pi\frac{\beta_0 j}{2^n}}\left|\underline{j}\right\rangle
\end{aligned}$$

Supposons pour le moment que $r \mid 2^n$. Alors,

$$\frac{1}{2^n/r}\sum_{\alpha=0}^{2^n/r-1}e^{-2i\pi\frac{\alpha j}{2^n/r}} = \begin{cases} 1, & \text{si } \frac{j}{2^n/r} \text{ est entier,} \\ 0, & \text{sinon.} \end{cases}$$

Remarque : On utilise ici propriétés de somme d'une suite géométrie. Pour $z \in \mathbb{C}$,

$$\sum_{k=0}^{n-1}z^k = \begin{cases} n, & \text{si } z=1, \\ \frac{1-z^n}{1-z}, & \text{sinon.} \end{cases}$$

On en déduit donc que :

$$\frac{1}{n}\sum_{k=0}^{n-1}e^{2i\pi\frac{kj}{n}} = \begin{cases} 1, & \text{si } \frac{j}{n} \text{ est entier,} \\ 0, & \text{sinon.} \end{cases}$$

En effet, il suffit de poser $\omega = e^{\frac{2i\pi}{n}}$ et $z = \omega^j$. Alors, si $\frac{j}{n}$ est un entier, $z = 1$ et donc la somme divisé par n est elle-même égale à 1. Sinon, on a $\frac{1-z^n}{n(1-z)}$, mais $z^n = 1$ et on obtient bien 0.

On peut alors reprendre notre calcul de $\left|\bar{\psi}_4\right\rangle$:

$$\left|\bar{\psi}_4\right\rangle = \frac{1}{\sqrt{r}}\sum_{\frac{j}{2^n/r}\text{entier},j=0}^{2^n-1}e^{-2i\pi\frac{\beta_0 j}{2^n}}\left|\underline{j}\right\rangle = \frac{1}{\sqrt{r}}\sum_{l=0}^{r-1}e^{-2i\pi\frac{\beta_0 l}{t}}\left|\underline{\frac{2^n l}{r}}\right\rangle$$

La mesure du premier registre fournit un entier $m = \frac{2^n l}{r}$ correspondant à l'un des états $\left|\underline{\frac{2^n l}{r}}\right\rangle$

du qubit $|\bar{\psi}_4\rangle$.

On va chercher à déduire la période r. Commençons par diviser par 2^n ($x = \frac{m}{2^n} = \frac{l}{r}$).

— Si la valeur de x est entière, alors on n'a aucune information sur r. Il faudra donc recommencer l'exécution du circuit quantique.

— Si $pgcd(l, r) = 1$, alors d'une part $x = \frac{m}{2^n}$ est connu et d'autre part, $\frac{l}{r}$ est son écriture sous forme irréductible. Afin de déterminer r, il suffit donc de réduire $\frac{m}{2^n}$ en une fraction irréductible.

— Si $pgcd(l, r) \neq 1$, alors l'écriture irréductible de x est $\frac{l'}{r'}$. Ainsi, $r'l = l'r$ donc $r' \mid rl'$, mais, comme $pgcd(r', l') = 1$ par le lemme de Gauss $r' \mid r$. Nous n'obtenons par directement la période de r mais un facteur r'. On recommence donc l'algorithme mais avec un choix initial de $a^{r'}$. En effet, comme la fonction $a \to a^k$ est périodique de période r, alors la fonction qui a $k \to (a^{r'})^k$ est de période r/r'. Nous sommes certains que ce processus se termine car r n'a qu'un nombre fini de facteurs.

Supposons maintenant que nous nous trouvons dans le cas général (r ne divise pas 2^n). Il est possible de montrer que la mesure du premier registre donnera, dans ce cas là, une valeur entière proche d'un multiple de $\frac{2^n}{r}$ avec une probabilité suffisamment élevée. En effet, ces probabilités sont presque nulles sauf pour les valeurs de j proches des réels $\frac{2^n l}{r}$ (qui ne sont pas des entiers).

On va obtenir l'ordre r, ou l'un de ses facteurs, à partir du développement en fractions continues. À part cela, les conclusions sont les mêmes que précédemment.

Remarque : Une fraction continue est une expression mathématique de la forme : $a_0 + \frac{1}{a1 + \frac{1}{a_2 + \frac{1}{\ddots}}}$ où a_0 est une constante, et les $a_1, a_2...$ sont des coefficients entiers (on note cette fraction par la liste $[a_0, a_1, ...]$). Étudions un cas concret ($N = 21$). Pour $2^n = 512$, si la mesure donne un entier j proche de 427, alors le développement en fraction continue de $\frac{j}{512}$ est par exemple $[0, 1, 5, 42, 2]$. Les dénominateurs des fractions issues des différentes listes sont les candidats pour l'ordre r, mais on sait que l'ordre r cherché est inférieur à l'entier N. Ici, la meilleure fraction ayant un dénominateur inférieur à N est $\frac{5}{6}$. On trouve ainsi $r = 6$.

Étape 5. Si r est impair, on va voir dans l'étape 6 que cela va poser problème. On doit donc reprendre l'algorithme du début et choisir une autre valeur aléatoire de a.

Étape 6. On a que $a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1) \equiv 0[N]$. Ainsi, $N \mid (a^{r/2} - 1)(a^{r/2} + 1)$. On sait dans un premier temps que N ne divise pas $(a^{r/2} - 1)$, puisque r est la plus petite puissance

k telle que $a^k - 1$ soit divisible par N. Ainsi, dans un second temps, si N ne divise pas $a^{r/2} + 1$ (i.e. $a^{r/2} \not\equiv -1[N]$ ou $x \neq 0$), alors N peut être partiellement divisé en $(a^{r/2} - 1)$ et $(a^{r/2} + 1)$. Évidemment, si N divise $a^{r/2} + 1$, alors, on a pas d'autre choix que de recommencer l'algorithme en choisissant une nouvelle valeur de a.

Remarque : On arrive donc à cette étape si r est pair et si $a^{r/2} \not\equiv -1[N]$. Quelle est la probabilité que cela se produise, étant donné que a a été choisi au hasard ? Nous citons le théorème suivant : "Supposons que N soit impair et ne soit pas une puissance d'un nombre premier. Si $a < N$ est choisi uniformément au hasard avec $pgcd(a, N) = 1$ alors $P(r \equiv 0[2] \wedge a^{r/2} \not\equiv -1[N]) = \frac{1}{2}$."

Étape 7. Finalement on peut déterminer (à l'aide de l'algorithme d'Euclide), $p = pgcd(a^{r/2} - 1, N)$ et $q = pgcd(a^{r/2} + 1, N)$. Au moins l'un d'entre eux sera un facteur non trivial de N.

3.2 Application au cas N = 21

Considérons dans cette section le cas particulier $N = 21$. Supposons que $n = 9$ (on a bien $2^9 = 512 \geq 21^2 = 441$).

Étape 1. N n'est ni pair, ni premier.

Étape 2. Prenons a = 2.

Étape 3. a est bien premier avec N.

Étape 4. On va chercher à déterminer l'ordre de a modulo N. Reprenons les calculs du circuit de Shor :

$$|\psi_0\rangle = |\underline{0}\rangle \otimes |\underline{0}\rangle$$
$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes |\underline{0}\rangle$$
$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes \left|\underline{a^k[N]}\right\rangle$$

Nous devons maintenant retrouver l'ordre de a modulo N qui est ici $r = 6$. Réordonnons les éléments de $|\psi_2\rangle$ en regroupant les termes selon le second facteur qui est l'un des $\left|\underline{a^k}\right\rangle$, pour k

variant de 0 à 5 :

$$\begin{aligned}|\psi_2\rangle =& \frac{1}{\sqrt{512}}(|\underline{0}\rangle + |\underline{6}\rangle + ...)\,|\underline{1}\rangle \\ &+ \frac{1}{\sqrt{512}}(|\underline{1}\rangle + |\underline{7}\rangle + ...)\,|\underline{2}\rangle \\ &+ \frac{1}{\sqrt{512}}(|\underline{2}\rangle + |\underline{8}\rangle + ...)\,|\underline{4}\rangle \\ &+ \frac{1}{\sqrt{512}}(|\underline{3}\rangle + |\underline{9}\rangle + ...)\,|\underline{8}\rangle \\ &+ \frac{1}{\sqrt{512}}(|\underline{4}\rangle + |\underline{10}\rangle + ...)\,|\underline{16}\rangle \\ &+ \frac{1}{\sqrt{512}}(|\underline{5}\rangle + |\underline{11}\rangle + ...)\,|\underline{11}\rangle\end{aligned}$$

On effectue ensuite une mesure du second registre. Supposons que l'on obtienne $|\underline{2}\rangle$. Alors le premier registre, une fois normalisé, contient le qubit : $\left|\bar{\psi_3}\right\rangle = \frac{1}{\sqrt{86}}(|\underline{1}\rangle + |\underline{7}\rangle + ...)$.

Il nous reste à appliquer la transformée de Fourier inverse et à effectuer une mesure sur le premier registre. On a :

$$\begin{aligned}\left|\bar{\psi_4}\right\rangle &= \hat{F}^{-1}\left|\bar{\psi_3}\right\rangle \\ &= \hat{F}^{-1}(\frac{1}{\sqrt{86}}\sum_{\alpha=0}^{85}|\underline{6\alpha+1}\rangle) \\ &= \frac{1}{\sqrt{86}}\sum_{\alpha=0}^{85}\frac{1}{\sqrt{512}}\sum_{j=0}^{511}e^{-2i\pi\frac{(6\alpha+1)j}{512}}\left|\underline{j}\right\rangle \\ &= \frac{1}{\sqrt{512}}\sum_{j=0}^{511}(\frac{1}{\sqrt{86}}\sum_{\alpha=0}^{85}e^{-2i\pi\frac{6\alpha j}{512}})e^{-2i\pi\frac{j}{512}}\left|\underline{j}\right\rangle\end{aligned}$$

La somme $\Sigma(j) = \frac{1}{\sqrt{86}}\sum_{\alpha=0}^{85}e^{-2i\pi\frac{6\alpha j}{512}}$ représente un nombre complexe, pouvant adopter des valeurs autres que 0 et 1. La mesure du premier registre conduit à la valeur j avec la probabilité $p_j = \frac{1}{512}\left|\Sigma(j)\right|^2$. Ces probabilités sont presque nulles, à l'exception des valeurs de j qui sont proches des réels $\frac{2^n l}{r}$, avec $l = 0, 1, \ldots, 5$. Ainsi, les probabilités sont significatives principalement autour des entiers $0, 85, 171, 256, 341$, et 427.

Il devient alors possible de déterminer l'ordre, ou l'un de ses facteurs, en exploitant le développement en fractions continues. Si la mesure donne par exemple un entier j proche de 427, alors le développement en fraction continue de $\frac{j}{512}$ révèle l'ordre $r = 6$ (cf. Remarque sur les fractions continues).

Étape 5. $r = 6$ est pair, on peut donc poursuivre à l'étape 6.

Étape 6. On a $2^3 + 1 = 9$ et $2^3 - 1 = 7$.

Étape 7. Finalement, $p = pgcd(9, 21) = 3$ et $p = pgcd(7, 21) = 7$.

3.3 Implémentation

Dans cette partie on va chercher à implémenter le circuit quantique permettant de déterminer l'ordre (code fourni en Annexe). Dans ce scénario particulier, nous choisissons $N = 15$, $n = 8+4$, et $a = 7$. On peut noter que, dans ce contexte, les conditions $pgcd(7, 15) = 1$ et $2^{12} = 4096 \geq 15^2 = 225$ sont respectées.

Dans un premier temps, on applique sur le premier registre (8 premiers qubits), une transformation de Hadamard.

Dans un second temps on va se servir de *qc.append*, fonction dont le premier argument correspond à la transformation quantique que l'on souhaite ajouter au circuit et le second spécifie les qubits sur lesquels l'opération sera appliquée.

On s'en sert une première fois pour appliquer l'opération de multiplication modulaire, **apmod15** sur les n_2 qubits du second registre, puis une seconde fois pour appliquer **tfi**, l'opération de transformée de Fourier quantique inverse sur les n_1 qubits du premier registre.

Remarque : Ces deux opérations sont détaillées ci-dessous dans les Sections 3.3.2 et 3.3.1.

Pour finir, on mesure les n_1 premiers qubits et on stocke les résultats dans les premiers n_1 bits classiques du circuit quantique.

Remarque : Lorsque nous examinons la partie quantique de l'algorithme de Shor, nous pouvons réaliser que son cœur repose en réalité sur l'algorithme bien connu d'**Estimation Quantique de Phase** (voir Figure 14). Cette estimation de phase est essentielle pour extraire des informations cruciales sur la période d'une fonction modulaire.

3.3.1 Construction de *tfi*

On cherche ici à appliquer une Transformée de Fourier Inverse sur les n_1 qubits du premier registre.

Avant toute chose, nous allons commencer par introduire le circuit quantique correspondant à une **Transformation de Fourier**. On rappelle que : $\hat{F}\left|\underline{k}\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{2i\pi \frac{kj}{2^n}} \left|\underline{j}\right\rangle$. On va

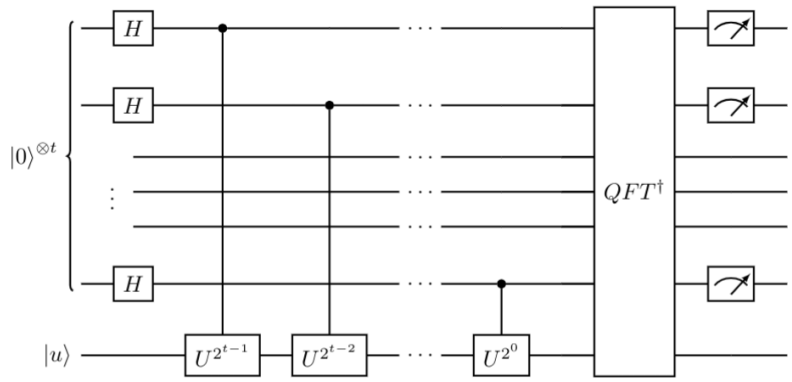

FIGURE 14 – Circuit associé à l'algorithme d'Estimation Quantique de Phase.

commencer par modifier légèrement ce résultat afin de comprendre comment construire le circuit associé :

$$\begin{aligned}
\hat{F}\,|\underline{k}\rangle &= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{2i\pi \frac{kj}{2^n}} \,|\underline{j}\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \prod_{l=1}^{n} e^{2i\pi \frac{kj_l}{2^l}} \,|j_1 j_2 ...\rangle \\
&= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2i\pi \frac{k}{2^1}} \,|1\rangle) \otimes (|0\rangle + e^{2i\pi \frac{k}{2^2}} \,|1\rangle) \otimes ...(|0\rangle + e^{2i\pi \frac{k}{2^n}} \,|1\rangle)
\end{aligned}$$

Remarque : On a réécrit j sous sa notation binaire $j = \sum_{l=1}^{n} j_l 2^{n-k}$.

On obtient une expression dans laquelle on reconnaît quelque chose de très similaire à ce que l'on obtient à l'aide de portes de Hadmard. En effet (pour k_m égal à 0 ou 1), $H\,|k_m\rangle = \frac{|0\rangle + e^{2i\pi \frac{k_m}{2}} |1\rangle}{\sqrt{2}}$.

Nous allons ensuite utiliser des portes de rotation pour retrouver les phases attendues :

$$R_l\,|k_m\rangle = e^{2i\pi \frac{k_m}{2^l}} \,|k_m\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2i\pi}{2^l}} \end{bmatrix}$$

Remarque : Nous obtenons cette matrice 2x2 (matrice unitaire d'un unique qubit) car le premier vecteur colonne correspond à ce que fait cette porte à l'état 0 (renvoie l'état 0) et le second à l'état 1 (applique la phase attendue à l'état 1).

À ce stade, nous approchons de l'expression de $\hat{F}\,|\underline{k}\rangle$. Cependant, nous devons utiliser des portes *Swap* pour échanger les premiers qubits avec les derniers, afin d'obtenir exactement le résultat souhaité.

Maintenant que nous avons identifié les portes nécessaires - la porte de Hadamard, les portes de rotation, et finalement les portes *Swap* - nous sommes prêts à construire le circuit correspondant (cf. Figure 15).

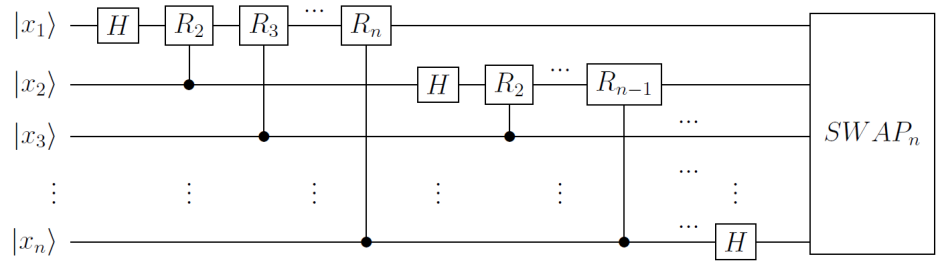

FIGURE 15 – Circuit associé à la Transformée de Fourier quantique.

Les opérations impliquées dans la Transformation de Fourier sont unitaires, ce qui signifie que le circuit est réversible. Par conséquent, la conception de la **Transformée de Fourier Inverse** associée ne devient qu'une tâche inverse intuitive. Pour la construire, il nous suffit d'examiner le circuit présenté dans la Figure 15 en sens inverse, tout en étant attentif à appliquer l'opération conjuguée de chaque étape.

On commence donc par implémenter une boucle, qui effectue une série d'échanges entre les qubits du registre (*Swap*). Les qubits sont échangés de manière symétrique par rapport au milieu du registre. Ensuite, pour chaque qubit dans le registre, une porte de Hadamard est appliquée. Les boucles sur j et m appliquent des portes de rotation conditionnelles autour de l'axe Rz. Les angles de rotation sont déterminés en fonction des positions des qubits, créant ainsi les déphasages nécessaires pour réaliser cette transformation.

3.3.2 Construction de *apmod15*

On cherche ici à appliquer une opération de multiplication modulaire sur les n_2 qubits du second registre.

On peut réécrire $a^x[N]$ à l'aide de la notation binaire de x. On obtient alors $a^{2^{n-1}x_1+\ldots+2^0x_n}[N]$, ce qui est encore égal à $a^{2^{n-1}x_1}\ldots a^{2^0x_n}[N]$. Les x_i en exposants indiquent qu'il s'agit ici d'opérations contrôlées. On retrouve bien une partie de l'algorithme d'Estimation Quantique de Phase, seulement, avec $U^{2^x} = a^{2^x}[N]$.

Cette exponentiation modulaire peut être mise en œuvre dans un circuit quantique, à l'aide de portes *Swap*.

Prenons le cas particulier où $a = 7$ à la puissance 2. En commençant avec l'état $|0001\rangle$ (décision de conception pour notre circuit), nous effectuons une série d'opérations de *Swap* :

1. Après un premier *Swap* entre les qubits 2 et 3, l'état devient $|0010\rangle$.
2. Un second *Swap* entre les qubits 1 et 2 transforme l'état en $|0100\rangle$.
3. Un troisième *Swap* entre les qubits 0 et 1 donne $|1000\rangle$.

En appliquant ensuite une porte X sur chaque qubit, l'état final après une itération est $|0111\rangle$, correspondant à la valeur 7 (car $7^1 \equiv 7[15]$). En poursuivant ce processus une fois de plus, nous obtenons $7^2 \equiv 4[15]$, représenté par l'état $|0100\rangle$, qui est effectivement le reste de la division de 49 par 15. Cette méthode peut être répétée pour calculer les autres opérations modulaires nécessaires.

Ainsi, à chaque itération, notre circuit va donc effectuer des opérations de permutations (*Swap*) et d'inversions (X). Une porte de contrôle est créée pour la porte U (l'opération représentée par la porte U sera appliquée uniquement si un qubit de contrôle est dans l'état $|1\rangle$).

4 Conclusion

Ce rapport présente les principes généraux de l'algorithmique quantique, mettant en lumière les différences clés entre les algorithmes quantiques de leurs homologues classiques. Il introduit des algorithmes quantiques emblématiques tels que l'algorithme de Deutsch et l'algorithme de Grover, soulignant leur puissance dans la résolution de problèmes spécifiques.

L'axe principal du rapport se concentre sur une étude approfondie de l'algorithme de Shor. Une attention particulière est accordée à sa capacité à factoriser des nombres entiers en temps polynomial, dépassant ainsi les capacités des algorithmes classiques. Ce rapport met donc en évidence les implications significatives de cette capacité en matière de cryptographie et de sécurité des systèmes d'information. L'aspect pratique de l'étude ressort à travers une section dédiée à l'implémentation concrète de l'algorithme de Shor.

Il aurait été intéressant d'avoir le temps d'explorer la capacité de réaliser une factorisation en utilisant un annealer quantique, en formulant le problème de factorisation sous la forme d'un problème d'optimisation quantique (QUBO - *Quadratic Unconstrained Binary Optimization*). Cette approche nous aurait permise d'explorer une autre facette de l'informatique quantique, en exploitant les caractéristiques uniques des annealers quantiques pour résoudre des problèmes de nature différente.

5 Annexes

- Oracle.

$$
\begin{aligned}
|\psi_2\rangle &= O_f\,|\psi_1\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{\ell=0}^{2^n-1} \left|\underline{\ell}\right\rangle \cdot \tfrac{1}{\sqrt{2}} \left(\left|0 \oplus f(\underline{\ell})\right\rangle - \left|1 \oplus f(\underline{\ell})\right\rangle\right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{\ell=0}^{2^n-1} (-1)^{f(\underline{\ell})} \left|\underline{\ell}\right\rangle \cdot \tfrac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle\right)
\end{aligned}
$$

On a utilisé :

$$
\left|0 \oplus f(\underline{\ell})\right\rangle - \left|1 \oplus f(\underline{\ell})\right\rangle = \left\{ \begin{array}{ll} |0\rangle - |1\rangle & \text{si } f(\underline{\ell}) = 0 \\ -(|0\rangle - |1\rangle) & \text{si } f(\underline{\ell}) = 1 \end{array} \right\} = (-1)^{f(\underline{\ell})} \left(|0\rangle - |1\rangle\right).
$$

FIGURE 16 – Passage de l'oracle dans l'algorithme de Grover.

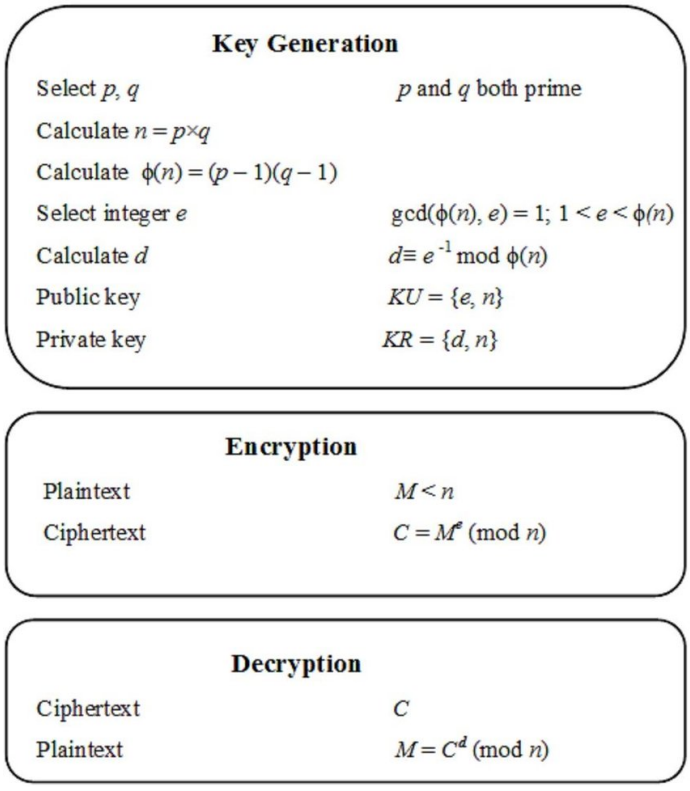

FIGURE 17 – RSA.

6 Références

[1] https ://indico.cern.ch/event/970903/

[2] https ://epubs.siam.org/doi/pdf/10.1137/18M1170650

[3] https ://www.oezratty.net/wordpress/2022/understanding-quantum-technologies-2022/

[4] https ://en.wikipedia.org/wiki/Grover%27s_algorithm

[5] Quantum computation and Quantum Information M. A. Nielsen & L. Chuang

[6] Un peu de Mathématiques pour l'informatique quantique, Arnaud Bodin (2021)

[7] https ://www.nature.com/articles/npjqi201523

[8] Programmation : https ://qiskit.org/learn et https ://myqlm.github.io/

[9] https ://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/shor.ipynb

[10] https ://github.com/mett29/Shor-s-algorithm/blob/master/Shor.ipynb