2023-02

Robustesse des algorithmes de la cryptographie classique face à l'informatique quantique

HARERIMANA, Elie

UB

UNIVERSITE DU BURUNDI
FACULTE DES SCIENCES
DEPARTEMENT DE PHYSIQUE
CENTRE DE RECHERCHE EN MATHEMATIQUES ET PHYSIQUE

Robustesse des algorithmes de la cryptographie classique face à l'informatique quantique

Par:

HARERIMANA Elie

Mémoire présenté et défendu publiquement en vue de l'obtention du Diplôme de Master en Physique fondamentale et appliquée.
Option: Physique fondamentale

Sous la direction de:
Prof. Rachel AKIMANA

Bujumbura, février 2023

Composition du Jury

Prof. Hippolyte NYENGERI: *Président*
Dr Thaddée BARANCIRA: *Secrétaire*
Dr Pierre Célestin KARANGWA: *Membre*
Prof. Rachel AKIMANA: *Directeur*

Dédicace

A mes parents;
A mes frères et sœurs;
A toute ma famille;
A mes amis.

Remerciements

C'est un grand honneur pour moi de m'adresser à mon directeur de mémoire Professeur Rachel AKIMANA en la remerciant pour tous ses conseils, son encouragement et toutes ses idées sans lesquelles ce mémoire n'aurait pu aboutir.

Je tiens également à remercier les membres du jury pour leurs conseils scientifiques pendant la rédaction de ce mémoire.

Mes sincères remerciements sont adressés à tous les enseignants de la faculté des sciences en général et en particulier ceux du département de Physique, pour la formation scientifique dont ils m'ont fait bénéficier.

Je me dois également de remercier mes parents qui m'ont ouvert les portes du savoir et de la sagesse. Je leur dois un grand respect pour m'avoir éduqué, accompagné et aidé durant toutes ces années. Je remercie également toute ma famille pour son soutien tant moral que matériel.

Je dis enfin merci à toutes les personnes qui m'ont accordé un soutien moral ou matériel pour ma formation depuis l'école primaire jusqu'à la fin de ce mémoire.

Résumé

Ce travail met l'accent sur la faiblesse des algorithmes de la cryptographie classique face à la puissance des calculateurs quantiques. Les problèmes qui ne peuvent pas être résolus en temps raisonnable, notamment la factorisation d'un entier composé de plusieurs chiffres et en général tous les problèmes NP-complets, avec l'arrivée des calculateurs quantiques, pourraient être résolus en un temps relativement court. Cela veut dire que tous les problèmes dont le nombre d'opérations augmente rapidement avec la taille des données, risquent d'avoir une complexité en un temps polynomial quantique. Ainsi, partant du problème de factorisation d'un entier en deux nombres premiers, nous avons utilisé l'algorithme de Grover qui cherche un élément dans une liste des valeurs $([\sqrt{n}] + u)^2 - n$ avec $u = 1, 2, 3, ...,$ pour y chercher un carré puis factoriser le nombre n sous la forme d'un produit de facteurs premiers p et q. Ainsi, il devient possible de casser le système RSA qui est un algorithme basé sur la factorisation.

Mots clés: cryptographie, factorisation, Problème NP-Complet, Algorithme de Grover, factorisation quantique.

Abstract

This work focuses on the weakness of classical cryptography algorithms in the face of the power of quantum computers. Problems that cannot be solved in a reasonable time, such as the factorization of an integer composed of several digits and in general all NP-complete problems, with the arrival of quantum computers, would be solved in a short time. This means that problems whose number of operations increase fastly with the size of the data, could be solved in a quantum polynomial time. Thus, starting from the problem of factorizing an integer into two primes, we used Grover's algorithm which searches for an element in a list of values $([\sqrt{n}] + u)^2 - n$, to search for a square and then factor the number n as a product of prime factors p and q.

Thus, it becomes possible to break the RSA system which is an algorithm based on factoring.

Key words: cryptography, factorization, NP-Complet, Grover's Algorithm, quantum factorization.

Table des matières

Liste des tableaux

Liste des figures

Liste des symboles, sigles et Abréviations

Symboles

$\mathscr{H}$: Espace de Hilbert

$\mathbb{C}$: Ensemble des nombres complexes

$\hbar$: Constante de planck réduite

$|\psi\rangle$: Vecteur d'état (ket)

$\langle\psi|$: Dual du vecteur d'état(bra)

$\langle\varphi|\psi\rangle$: produit scalaire

$\otimes$: Produit tensoriel

$\dagger$: Opérateur de Transposition

$*$: Opérateur de conjugaison

$\simeq$: sensiblement égal

mod: modulo

$|\psi\rangle^{\otimes n}$: Vecteur appartenant dans l'espace vectoriel

$\mathscr{H}^{\otimes n}$: Espace de Hilbert de dimension n

H: Opérateur d'Hadmard

O_f: Oracle associé à la fonction f

G: Porte de Grover

$|\underline{k}\rangle$: Vecteur d'état écrit dans une base donnée

Θ: thêta

O: Grand O

$\mathbf{\Omega}$: oméga

Sigles et abréviations

AES: Advanced Encryptions Standarded

C-NOT: Controlled-NOT

DES: Data Encryption Standard

MMP: Minimum Matching Problem

MST: Minimum Spanning Tree

NP: Non deterministic polynomial

RSA: Ravist-Shami-Adleman

SWAP: Shared Wireless Access Protocol

XOR: Exclusive OR

Avant-propos

Le présent travail de mémoire a été réalisé dans le cadre de l'obtention du diplôme de fin d'étude du deuxième cycle des enseignements de l'Université du Burundi.

L'idée du mémoire est de montrer l'avantage de l'informatique quantique par rapport à l'informatique classique et de tester la faiblesse des algorithmes de la cryptographie classique face à la puissance des calculateurs quantiques.

En effet, la suprematie des calculateurs quantiques réside dans certains phénomènes quantiques, notamment le principe de superposition et le parallélisme quantiques. Pour cela, nous avons consideré le problème de la factorisation d'un entier en deux nombres premiers qui est un des problèmes NP-complets (problèmes dont le nombre d'opérations augmente rapidement avec la taille des entrées) et nous avons fait son approche quantique avec la méthode de Grover; cela nous a conduit à conclure qu'on peut casser RSA, un algorithme cryptographique basé sur la factorisation.

Introduction générale

Depuis la création de la vie sur la Terre, l'homme cherche à trouver une solution aux problèmes qu'il rencontre. En même temps, il cherche une compréhension et une explication des phénomènes qu'il observe dans le monde qui l'entoure.

L'émergence de nouvelles idées et concepts qui sont stimulés par des expériences sur terrain, des résultats théoriques publiés dans des livres, articles etc., fait que l'homme se lance de plus en plus dans la réalisation d'une théorie moderne et adéquate pour arriver à surmonter certains problèmes qu'il ne résolvait pas dans les temps antérieurs. Dans la période du $20^{ème}$ siècle, la discipline élaborée, moderne et la plus bouleversante est la Mécanique Quantique (MQ); celle-ci consiste à étudier et à décrire les phénomènes physiques fondamentaux à l'échelle atomique et subatomique [Bernstein and Vazirani, 1993].

En 1985, David Deutsch a reconnu qu'un ordinateur quantique avait des capacités bien supérieures à celles d'un ordinateur classique et a suggéré que de telles capacités pouvaient être exploitées par des algorithmes astucieux [Deutsch, 1985a].

En 1992, Deutsch et Jozsa [Deutsch and Jozsa, 1992] ainsi que Berthiaume et Brassard [Berthiaume and Brassard, 1992] [Berthiaume and Brassard, 1994a] ont montré que certains problèmes qui ne peuvent être résolus qu'avec une forte probabilité à l'aide d'un générateur de nombres aléatoires, peuvent être résolus exactement avec des ordinateurs quantiques. En 1994, Peter Shor [Shor, 1994] a trouvé un algorithme en temps polynomial pour factoriser des nombres de n bits sur des ordinateurs quantiques et a suscité une vague d'enthousiasme pour l'informatique quantique.

Certains phénomènes quantiques comme la superposition quantique et le parallélisme quantique sont à l'origine de la suprématie des calculateurs quantiques vis à vis de ceux classiques [Brune and Raimond, 2005]. La résolution des problèmes dont le nombre d'opérations croît très rapidement avec la taille des entrées risque d'être facile. Certains problèmes NP-complets notamment la factorisation d'un entier composé de plusieurs chiffres, en deux nombres premiers, se basent par ailleurs sur des considérations liées au temps et non sur des démonstrations ou de la difficulté de calcul. Si les calculateurs quantiques sont mis en place, on pourra observer une accélération du temps de calcul notamment passer de la complexité exponentielle à une complexité sous-exponentielle. L'objectif général de ce travail est de tester la robustesse des algorithmes cryptographiques classiques.

Nous avons formulé les objectifs spécifiques suivants:

1 Montrer qu'on arrive à trouver une complexité en temps polynomial quantique pour les problèmes NP-Complets: cas de la factorisation des entiers.

2 Montrer qu'il est possible de casser RSA

Nous utilisons la méthode de Fermat en factorisant un entier composé en deux nombres premiers que nous notons p et q.

Par la suite, nous utilisons la méthode numérique dans le but de tracer à l'aide d'un logiciel gnuplot et d'un code en langage python la courbe de variation du nombre d'opérations.

Enfin, nous utilisons la méthode de Grover pour faire une approche quantique de la factorisation avec la méthode de Fermat.

Ce mémoire s'articule autour de quatre chapitres précédés par une introduction générale et sanctionnés par une conclusion générale.

Dans le premier chapitre, nous donnons une introduction générale sur la cryptographie, l'évolution de cette dernière en donnant quelques repères historiques et quelques types de chiffrement.

Dans le deuxième chapitre, nous traitons les problèmes NP-complets qui sont des problèmes dont le nombre d'opérations augmente rapidement avec la taille des données, car notre volonté est de montrer l'avantage de l'informatique quantique par rapport à l'informatique classique. Donc nous avons dû considérer les problèmes dont le nombre d'opérations croît rapidement avec la taille des entrées.

Dans le troisième chapitre, nous présentons quelques concepts clés de la théorie de l'information quantique. Nous générons également les portes logiques quantiques à un seul qubit, à deux qubits, et à trois qubits à l'aide d'un logiciel qiskit [Wille et al., 2019a] associé au langage python.

Dans le quatrième chapitre, nous présentons la méthode classique de Fermat pour la factorisation d'un entier n, nous montrons que sa complexité augmente rapidement avec la méthode classique de Fermat. Nous utilisons la méthode de Grover pour faire une approche quantique de la factorisation de Fermat.

Chapitre 1

Introduction générale à la cryptographie

1.1 Introduction

La cryptographie joue un rôle important dans la protection d'informations échangées entre deux interlocuteurs lors d'une communication. La cryptographie comme science a évolué pour répondre à la demande des utilisateurs et aussi avec les progrès technologiques.

1.2 Quelques définitions

La cryptographie: étymologiquement le mot **cryptographie** vient du mot grec **kryptos** qui signifie caché et **graphein** qui signifie écrire [Stern, 1998] et est l'ensemble des techniques permettant de dissimuler une information à l'aide d'un code secret, c'est-à-dire permettant de rendre un message transmis incompréhensif sans qu'on le décode.

La cryptanalyse [harizaka Chrystelle, 2019]: c'est l'étude des procédés cryptographiques permettant de trouver des faiblesses, en particulier, de pouvoir décrypter des messages chiffrés. Le décryptement est l'action consistant à trouver le message en clair sans connaître la clé de déchiffrement.

La cryptologie [Stern, 1998]: est une science mathématique qui englobe la cryptographie et la cryptanalyse.

Un crypto-système [harizaka Chrystelle, 2019]: est un terme utilisé en cryptographie pour désigner un ensemble composé d'algorithmes cryptographiques et de tous les textes en clair, textes chiffrés et clés possibles.

Le chiffrement [Stern, 1998]: est le fait de coder un message de façon à le rendre secret.

Le déchiffrement [Stern, 1998]: est la méthode consistant à retrouver le message codé.

1.3 Objectifs de la cryptographie

L'objectif de la cryptographie est d'assurer la protection de l'information, autrement dit, elle permet d'assurer:

· la confidentialité : visant à garder les informations secrètes de tous sauf les personnes autorisées;

· l'authentification : permettant de prouver l'authenticité par la confirmation de l'identité d'une entité;

· l'intégrité des informations : permettant de garantir que les données n'ont pas été modifiées (par des utilisateurs non autorisés) lors du stockage ou du transfert;

· la non répudiation : cela consiste à garantir qu'aucun des partenaires ne puisse nier la transaction effectuée;

· la disponibilité : pour garantir l'accès à un service ou une donnée;

· contrôle d'accès : pour limiter l'accès aux ressources aux personnes privilégiées;

· message d'authentification : permettant de confirmer la source de l'information;

· signature : permettant de lier l'information à une entité.

1.4 Quelques repères historiques

La cryptographie est une science relativement jeune, qui a beaucoup d'applications dans le monde actuel. Elle se croise actuellement aux mathématiques et nous avons l'espoir qu'elle se croisera à la physique quantique, avec la probabilité grandissante de voir un jour arriver un ordinateur quantique.

Voici quelques repères historiques qui montrent que la cryptographie évolue dans le temps et s'adapte en fonction des besoins des utilisateurs:

1.4.1 Principe de Kerckhoffs.

En 1883, Auguste Kerckhoffs énonce un principe fondamental de la cryptographie: la sécurité d'un cryptosystème doit reposer exclusivement sur le secret de la clé [Fournaise, 2022]. Ce principe de base de la cryptographie s'oppose à une sécurité par obscurité. Il convient alors de faire en sorte que la partie secrète de la clé, soit la plus facile et la moins coûteuse à changer. En effet, si le système entier est compromis à un moment, il faut le remplacer dans sa totalité. Il est important de noter que ce principe a été énoncé à la fin du XIX ème siècle à une époque où les algorithmes de chiffrement étaient mécaniques [Rotella, 2018].

1.4.2 Enigma et les "bombes" de Turing

Pendant la seconde guerre mondiale, les Allemands utilisaient une machine à chiffrer appelée Enigma dont le nombre de clés possibles était de l'ordre de 10^{16} c'est-à-dire environ 10 millions de milliards, à une époque où les ordinateurs existaient mais étaient peu performants. La taille de l'espace des clés possibles de la machine Enigma rendait toute recherche exhaustive manuelle hors de portée. C'est avec les "bombes" d'Alan Turing (machines automatisant les calculs) que ce dernier a pu "casser" Enigma: à partir de messages chiffrés interceptés par les Alliés, ils pouvaient retrouver la clé et donc déchiffrer les messages [Courtois, 2019].

1.4.3 Claude Shannon et la notion de sécurité

Claude Shannon a prouvé que le chiffrement de Vernam (le chiffrement de Vernam utilise une clé secrète et un message; pour coder un message, on effectue un « ou exclusif » XOR entre le message à coder et la clé secrète; pour le décoder, on effectue à nouveau un « ou exclusif » entre le message à décoder et la clé secrète [harizaka Chrystelle, 2019]) était inconditionnellement sûr. Ce dernier étant impraticable, Shannon a théorisé la notion de sécurité des cryptosystèmes et a introduit la notion de sécurité pratique [Vaudenay, 1995].

Sécurité inconditionnelle: la connaissance du message chiffré n'apporte aucune information sur le message clair. Pour avoir une sécurité inconditionnelle, il faut que la clé soit au moins aussi longue que le message. Cela est évidemment impossible à mettre en place sauf dans quelques applications particulières [Vaudenay, 1995].

Sécurité pratique: la connaissance du message chiffré et de certains couples clairs/ chiffrés ne permet de retrouver ni la clé ni le message en un temps raisonnable.

De plus, Shannon a introduit les notions de confusion et de diffusion, nécessaires à la sécurité d'un algorithme de chiffrement. La confusion signifie que la relation entre les bits du texte clair, les bits du chiffré et les bits de la clé doit être suffisamment compliquée. La diffusion exprime , elle, le fait que changer un bit dans le texte clair doit changer un grand nombre de bits dans le texte chiffré. En d'autres termes, un bit en entrée du chiffrement doit impacter plusieurs bits en sortie, et la relation entre les bits doit être suffisamment compliquée [Vaudenay, 1995].

1.4.4 Norme de chiffrement des données (DES - 1973)

L'algorithme de chiffrement DES est un chiffrement symétrique standardisé qui souffre principalement de la petite taille des clés (56 bits). Il est actuellement complètement cassé, principalement à cause de la croissance de la puissance des ordinateurs. Pour donner quelques ordres de grandeur, en juin 1997, en utilisant 78000 ordinateurs, le DES était cassé en 96 jours ; 6 mois plus tard, en distribuant les calculs sur plusieurs millions

de processeurs, récupérer une clé secrète utilisée dans le DES prenait 39 jours. Encore 6 mois plus tard, pour un coût de 250000 dollars , on arrivait à casser le DES en une dizaine d'heures.

Les tailles de clés actuelles sont, en cryptographie symétrique, souvent de 128 bits ou 256 bits. Cependant, tous les systèmes décrits précédemment sont des algorithmes de chiffrement symétrique, ce qui impose que deux entités qui veulent communiquer en utilisant ces algorithmes doivent se mettre d'accord au préalable sur le secret commun à utiliser, sans que celui-ce ne soit compromis [Aumeunier, 1977].

1.5 Quelques méthodes de chiffrement

Nous pouvons distinguer les différents types de cryptographie: sur base de la gestion des clés y relatives ou sur base de la méthode de chiffrement.

1.5.1 Cryptographie basée sur la gestion des clés

1.5.1.1 Chiffrement symétrique

Dans le chiffrement symétrique, les acteurs que nous appelons Alice et Bob (qui sont des personnages virtuels couramment utilisés en cryptographie), se conviennent et au besoin échangent une clé qui doit être utilisée à la fois pour chiffrer et déchiffrer les messages. Dans ce cas, Alice chiffre le message à l'aide de la clé privée, transmet ce message à Bob qui, à son tour le déchiffre en utilisant la même clé d'Alice.

La cryptographie à clé privée est une méthode très confidentielle dans la mesure où, celui qui intercepterait la communication ne pourra pas lire les données échangées tant qu'il n'aura pas pu se procurer la clé. Il n'y a aucune authentification de fait sur l'émetteur comme sur le récepteur, sauf si deux personnes seulement disposent de la clé. Le principal souci avec cette méthode, c'est qu'il faut s'échanger la clé et lors de cet échange, il faut prendre des précautions particulières. Si Alice et Bob décident d'utiliser pour communiquer un principe de chiffrement symétrique, ils doivent partager la même clé k, tenue secrète et qui servira pour le chiffrement et le déchiffrement. Un des exemples les plus connus de la cryptographie à clé secrète est le chiffrement de Vernam, inventé par ce dernier en 1917. C'est un des seuls chiffrements inconditionnellement sûrs (c'est-à-dire pour lequel la connaissance du chiffré ne permet d'obtenir aucune information sur le texte clair). Si Alice veut envoyer un message m de longueur n à Bob en utilisant cette méthode, elle doit générer une clé aléatoire k de longueur n et calculer le chiffré à envoyer donné par la formule :

$$c = m \oplus k \tag{1.1}$$

où $\oplus$ représente le ou exclusif bit à bit.

Pour déchiffrer le message c, Bob qui possède la même clé k qu'Alice n'a alors qu'à calculer

$$m = c \oplus k \tag{1.2}$$

L'inconvénient majeur de cet algorithme est la longueur de la clé. On doit avoir k $\geq$ taille(m) [harizaka Chrystelle, 2019]. Pour que ce chiffrement soit inconditionnellement sûr, il ne faut utiliser la clé k qu'une seule fois, d'où le deuxième nom de chiffrement de Vernam : « One Time Pad ».

◇ **Type de chiffrement de la cryptographie à clé secrète**

On distingue deux types de chiffrement:

- Le chiffrement par bloc: dans ce chiffrement, il y a une séparation du texte clair en blocs d'une longueur fixe et un algorithme chiffre un bloc à la fois. La taille de ces blocs est essentielle pour la sécurité des communications, c'est-à-dire les grands blocs sont plus sécuritaires mais aussi plus lourds à transférer.

- Le chiffrement par flux: dans ces algorithmes les messages sont chiffrés bit par bit, quelle que soit la longueur du message à coder sans besoin de les découper [harizaka Chrystelle, 2019].

◇ **Avantages et inconvénients du chiffrement symétrique**

a. Avantages du chiffrement symétrique

– il est facile à mettre en place;

– il est assez simple ceci veut dire que tous les âges et tous les milieux peuvent l'utiliser;
– – il est Simple et facile à implémenter;

– il est adapté au grand flux de données à chiffrer;

– il nécessite moins de ressources de calcul;

– il nécessite moins de bande passante.

b. Inconvénients du chiffrement symétrique

– la clé secrète doit être partagée avec le destinataire;

– nécessite un canal sécurisé;

– toute personne interceptant la clé lors d'un transfert peut ensuite lire ou même modifier ou falsifier toutes les informations cryptées.

1.5.1.2 Chiffrement asymétrique

Les acteurs Alice et Bob ont chacun deux clés, la clé publique et la clé privée. La clé publique peut être diffusée sans contrainte, tandis que la seconde ne peut en aucun cas être divulguée par son propriétaire.

Si Alice veut envoyer un message confidentiel à Bob, elle se procure (par exemple dans un annuaire public) la clé publique "k_p" de Bob et chiffre le message m à l'aide d'un algorithme de chiffrement asymétrique. Elle envoie ensuite sur le canal non sûr le message chiffré c que seul Bob peut déchiffrer grâce à la clé secrète correspondante "k_s".

1.5.2 Cryptographie basée sur la méthode de chiffrement

1.5.2.1 Chiffrements qui remplacent des lettres par des symboles

Il s'agit des chiffrements les plus anciens et qui remplacent chaque lettre de l'alphabet par un symbole. Ce sont des chiffrement mono-alphabétiques. Nous pouvons donner quelques uns de ces chiffrements.

◇ Code Morse:

Créé au début des années 1840 par Samuel F. B. Morse, le code morse est un mode de communication international qui remplace les lettres de l'alphabet par des points et des tirets, sonores ou lumineux(voir le tableau de Morse ci-dessous (Figure 1.1)).

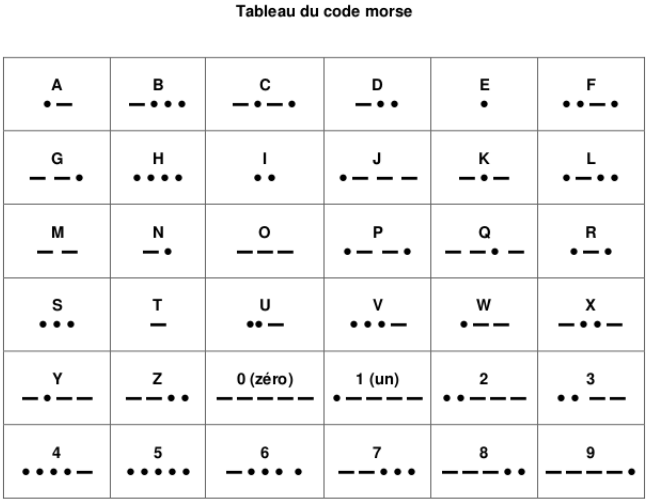

Figure 1.1: Code Morse

Le message en morse peut être transmis au moyen de faisceaux lumineux ou du télégraphe (système utilisant des fils de fer tels que des lignes téléphoniques pour transmettre des signaux) ou d'un émetteur radio [Bruno et al., 2016].

◇ Alphabet des templiers:

Le chiffrement des chevaliers templiers est un chiffre par substitution qui utilise 25 symboles pour représenter les lettres de l'alphabet (il manque la lettre J, qui est généralement remplacée par un I (Figure 1.2)). Les symboles utilisés par l'alphabet templier sont des morceaux de la Croix de Malte [Carmi, 1977].

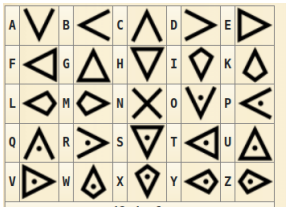

Figure 1.2: Chiffrement des chevaliers templiers

1.5.2.2 Chiffrements qui décalent les lettres

◇ Chiffrement de César

Il s'agit de décaler l'ensemble des valeurs des caractères du message d'un certain nombre de positions, c'est-à-dire de substituer chaque lettre par une autre. Pour utiliser le code de César, il suffit d'aligner toutes les lettres de l'alphabet, et de remplacer chaque lettre du message clair par une autre lettre à une distance fixe appelée « n ». Par exemple, avec n = 3, a devient d, b devient e, etc. (Figure 1.3: Chiffrement de César).

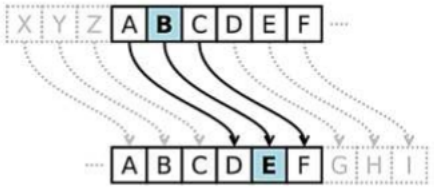

Figure 1.3: Chiffrement de César [harizaka Chrystelle, 2019]

◇ ROT13 : Le ROT13 (rotate by 13 places) est un cas particulier du chiffre de César, un algorithme simpliste de chiffrement de texte. Comme son nom l'indique, il s'agit d'un décalage de 13 caractères de chaque lettre du texte à chiffrer. Son principal aspect pratique est que le codage et le décodage se font exactement de la même manière.

Le chiffrement avec ROT13 est de même que le chiffrement de César mais chaque lettre de l'alphabet est décalée de 13 places (par exemple, la lettre « A » est remplacée par la lettre « N », la lettre « B » est remplacé par la lettre "O", et ainsi de suite) [Manullang et al., 2020].

1.5.2.3 Chiffrement par substitution

La technique, appelée chiffrement par substitution, consiste à changer l'alphabet pour chiffrer un message. Elle était déjà utilisée du temps des romains sous le nom de "chiffrement de César". Pour chiffrer un message, il faut décaler de trois lettres dans l'alphabet chaque lettre du message à transmettre. Pour décoder un message chiffré, il suffit de décaler chacune des lettres de trois positions dans le sens inverse de l'alphabet. Nous présentons un exemple d'un message chiffré par cette méthode : YHQL YLGL YLFL. En

appliquant la méthode exposée ci-dessus pour déchiffrer ce message, nous retrouvons la célèbre phrase que prononça Jules César après sa victoire sur Pharnace roi du Borsphore à Zéla : "VENI VIDI VICI" [Giraud et al., 2019].

⇒ Limites du chiffrement par substitution

Le chiffrement par substitution se casse facilement, car les lettres sont toujours codées de la même façon. Si l'attaquant ne possède que le message chiffré, il réalisera une cryptanalyse de la fréquence de chaque symbole ou chaque groupe de symboles pour tenter d'en extraire le message en clair. C'est ce que l'on appelle « attaque texte chiffré seul ».

1.5.2.4 Chiffrement par transposition

Les méthodes de chiffrement par transposition consistent à réarranger les données à chiffrer de façon à les rendre incompréhensibles. Il s'agit par exemple de réordonner géométriquement les données pour les rendre visuellement inexploitables.

◇ La technique assyrienne

La technique de chiffrement assyrienne est vraisemblablement la première preuve de l'utilisation de moyens de chiffrement en Grèce dès 600 ans avant Jésus Christ, afin de dissimuler des messages écrits sur des bandes de papyrus. La technique consistait à:

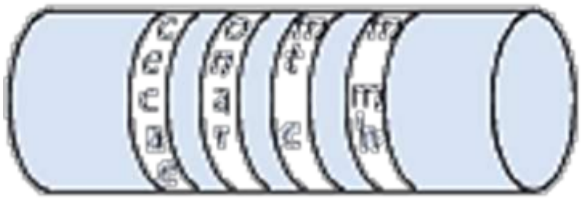

Figure 1.4: Le cylindre de scytale [harizaka Chrystelle, 2019]

• enrouler une bande de papyrus sur un cylindre appelé scytale ;

• écrire le texte longitudinalement sur la bandelette ainsi enroulée (le message dans l'exemple ci-dessus est « comment ça marche »).

Le message, une fois déroulé, n'est plus compréhensible (« cecaeonar mt c m mh »). Il suffit au destinataire d'avoir un cylindre de même diamètre pour pouvoir déchiffrer le message. En réalité un casseur peut déchiffrer le message en essayant des cylindres de diamètre successifs différents, ce qui revient à dire que la méthode peut être cassée statistiquement (il suffit de prendre les caractères un à un éloignés d'une certaine distance).

◇ Machine Enigma

Il s'agit d'une machine à chiffrer et à déchiffrer mécanique qui allie à la fois les méthodes de substitution et de transposition. Enigma possédait un fonctionnement particulièrement

simple : l'objet était équipé d'un clavier pour la saisie du message, de différentes roues pour le codage, et enfin d'un tableau lumineux pour le résultat. A chaque pression d'une touche du clavier, une lettre du panneau lumineux s'illuminait. Il y avait ainsi 3 roues de codage, appelées « Brouilleur Rotor », qui reliait le clavier au panneau lumineux. Par exemple, avec un seul rotor, lorsque l'on appuie sur b le courant passe par le rotor et allume A sur le panneau lumineux [harizaka Chrystelle, 2019]:

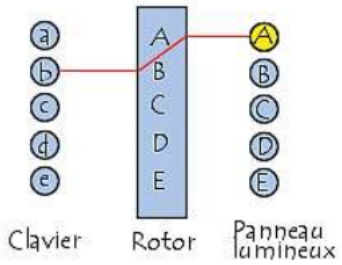

Figure 1.5: Fonctionnement de Enigma

1.6 Quelques fondements de la cryptanalyse

L'attaque à texte chiffré seul : c'est le cas le plus difficile. On ne dispose que d'un ou de plusieurs messages chiffrés, sans avoir d'informations sur leur signification en clair. Ce cas n'est, en réalité, pas si fréquent, car on a souvent une idée du type de message que l'on attend [Kortchinsky, 2004].

L'attaque à texte clair connu: l'attaquant possède plusieurs paires du type message clair/message codé. Cette attaque est plus fréquente qu'on ne pourrait le penser. Par exemple, pendant la Seconde Guerre Mondiale, les Alliés minaient certains ports, sachant que les autorités allemandes envoyaient alors toujours le même formulaire à leur marine [Kortchinsky, 2004].

L'attaque à texte clair choisi : l'attaquant choisit lui-même le message à coder. Cela peut arriver si on a un espion qui fait office d'opérateur dans le camp ennemi [Kortchinsky, 2004]. Les algorithmes à clé publique sont un autre exemple d'attaque à texte clair choisi, puisque l'algorithme pour chiffrer est public.

L'attaque par mot probable [Kortchinsky, 2004] : on ne connaît pas tout le message clair, mais au moins une partie, par exemple, la signature, ou bien le début, etc... Ainsi, le déchiffrement de la machine Enigma pendant la Seconde Guerre Mondiale utilisa beaucoup les bulletins météo envoyés par les Allemands. Ils commençaient en effet toujours par les mêmes mots. Dans ce cas, la rigueur allemande fut bien pénalisante...

En conclusion, nous venons de voir qu'il existait plusieurs moyens de chiffrer un message. Nous avons également vu que les différents moyens qu'on utilisait pour chiffrer un

message possédaient des avantages et des inconvénients. Le plus grand moyen de cacher un message serait alors de chiffrer ce dernier de façon à ce qu'il soit difficile de le déchiffrer. La plupart des algorithmes de chiffrement se construisent sur base des problèmes mathématiques dits difficiles. Le chapitre qui va suivre sera réservé à l'étude des problèmes NP-complets.

Chapitre 2

Problèmes Informatiques

2.1 Intoduction

Dans ce chapitre, nous allons présenter des problèmes d'optimisation combinatoire, problèmes qui ne font pas partie de la classe de calcul classique polynomial, mais qui font partie des problèmes dont le nombre d'opérations élémentaires croit rapidement avec la taille des entrées, Ces problèmes sont considérés comme les plus difficiles dans la classe des problèmes NP. Nous nous sommes référés à ces problèmes, étant donné qu'ils ont une complexité qui augmente vite, pour montrer que nous pouvons transformer leur complexité en une complexité polynomiale quantique.

2.2 Notion de complexité

2.2.1 Définitions

La complexité d'un algorithme ne fait pas référence explicitement au temps de calcul absolu des implémentations de cet algorithme mais de la vitesse avec laquelle ce temps de calcul augmente quand la taille des entrées augmente.

L'efficacité d'un algorithme se mesure par le temps d'exécution en fonction de la taille des données d'entrée et la place mémoire nécessaire à son exécution en fonction de la taille des données d'entrée.

2.2.2 Méthode de calcul de complexité

Il existe deux méthodes pour calculer la complexité des algorithmes:

- La première méthode consiste à prendre un chronomètre et compter le temps qui s'écoule entre le début et la fin d'exécution d'un programme. Cette méthode présente

des résultats relatifs puisque la puissance des machines n'est pas la même. Il s'agit d'une méthode empirique.

- La deuxième méthode consiste à compter le nombre d'opérations élémentaires d'un algorithme. Par **opérations élémentaires** on entend addition ou multiplication pour l'algèbre et si par exemple on a un algorithme de tri, opérations élémentaires signifie comparaison, etc. Il s'agit d'une méthode dite mathématique.

2.3 Ordre de grandeur en complexité

2.3.1 Complexité asymptotique

Soient deux algorithmes A et B résolvant le même problème et de complexité respectives $5000n$ et n^2. On se demande parmi les deux le plus efficace.

- Le rapport des complexités de B à A est égal à n/5000. Donc :
  - ◇ Pour n $<$ 5000, B est plus efficace.
  - ◇ n = 5000, A et B ont la même efficacité.
  - ◇ n $>$ 5000, A est plus efficace.

- Notons que plus n devient grand, plus A est efficace devant B. Si les tailles sont « petites », la plupart des algorithmes résolvant le même problème s'équivalent. C'est le comportement de la complexité d'un algorithme quand la taille des données devient grande qui est important. On appelle ce comportement la complexité asymptotique.

2.3.2 Notations de Landau

Comme dit plus tôt nous ne nous intéressons pas à la complexité exacte d'un algorithme, mais à la vitesse avec laquelle augmente le nombre d'opérations quand la taille n des entrées devient de plus en plus grande. Pour ce faire, nous avons besoin des notations asymptotiques.

Les notations asymptotiques sont définies comme suit [Knuth, 1976]

- O: f = O(g) $\Leftrightarrow \exists n_0, \exists c \geqslant 0 : \ \forall n \geq n_0, f(n) \leqslant c \times g(n)$

- $\Theta : f = \Theta(g) \Leftrightarrow f = O(g)$ et $g = O(f)$

- $\Omega : f = \Omega(g) \Leftrightarrow g = O(f)$

Note

◇ On dit que f est dominée par g et l'on note f= O(g) lorsque $\exists n_0, \exists c \geqslant 0\ \forall n \geq n_0,\ f(n) \leqslant c \times g(n)$

◇ On dit que f est du même ordre de grandeur que g et l'on note $f = \Theta(g)$ lorsque f=O(g) et g=O(f).

◇ Etant donnée une fonction f(n), $\Omega(f(n))$ est l'ensemble des fonctions g(n) telles qu'il existe une constante positive réelle c et un entier non négatif N tel que $\forall n \geq N, g(n) \geqslant c \times f(n)$. En d'autres mots, f(n) est une (forme de) borne inférieure asymptotique pour les fonctions dans $\Omega(f(n))$, alors qu'elle est une borne supérieure asymptotique pour les fonctions dans O(f(n)).

Si une fonction est à la fois dans $\Omega(f(n))$ et dans O(f(n)), alors on peut en conclure que son taux de croissance est équivalent à celui de f(n).

2.4 Différents ensembles de complexité

Avant d'entrer dans les ensembles de complexité, définissons d'abord quelques concepts clés qui vont nous permettre de bien mener l' analyse des problèmes NP-complets.

- **Instance d'un problème** : soit X un problème caractérisé par l'ensemble E de ses données, $E = \{e_1, e_2, e_3, \cdots, e_n\}$, une instance de X serait alors caractérisée par un ensemble concret $E' = \{e_1, e_2, e_3, \cdots, e_n\}$, une deuxième instance serait caractérisée par un deuxième ensemble concret $E" = \{e_1, e_2, e_3, \cdots, e_n\}$ et ainsi de suite. L'invariant suivante est toujours vérifié : $|E| = |E'| = |E"|$ [Ferro et al., 2005] où $|E|,\ |E'|\ et\ |E"|$ sont les cardinaux des ensembles $E,\ E'\ et\ E''$ respectivement.

- **Problème abstrait**: relation binaire entre un ensemble d'instances d'un problème et un ensemble de solutions à ce problème [Ferro et al., 2005].

- **Problème de décision (ou décisionnel)**: un problème décisionnel abstrait fait correspondre à toute instance d'un problème l'ensemble des solutions $\{vrai, faux\}$ [Ferro et al., 2005].

- **Problème d'optimisation**: il s'agit d'un quadruplet < I, S, f, mode > où I représente les donnés du problème, S, le type de collection/ensemble ou le type d'élément qu'on attend comme résultat, f est la fonction qui mesure la qualité du résultat et mode (min ou max) indique s'il faut minimiser ou maximiser la taille de ce résultat [Ferro et al., 2005].

- Pour un problème donné, on dit qu'il est **résoluble en temps polynomial**, s'il existe un algorithme permettant la résolution en $O(n^k)$ pour une constante k et la taille d'entrées n données [Ferro et al., 2005].

- **Réductibilité**: un problème X peut être ramené (réduit) à un autre problème $X^{'}$ si une instance quelconque de X peut être facilement reformulée comme instance de $X^{'}$ dont la solution sera aussi solution pour X [Ferro et al., 2005].

- **Réductibilité en temps polynomial**: la fonction qui réduit le problème X en $X^{'}$ peut être obtenue en temps polynomial [Ferro et al., 2005].

- **Machine de Turing** : modèle abstrait du fonctionnement des appareils de calcul, tel un ordinateur [Bernstein and Vazirani, 1993].

- **Machine de Turing déterministe**: c'est une machine qui produit le même résultat quelle que soit l'entrée en passant toujours par la même séquence d'états [Bernstein and Vazirani, 1993].

- **Machine de Turing non déterministe** [Bernstein and Vazirani, 1993]: elle se différencie de celle déterministe par le fait qu'elle peut avoir plusieurs transitions pour un état donné.

2.4.1 Classement des différents problèmes informatiques

Voici un panorama sur les différents classements de problèmes informatiques comme donnés par Ferro. Comme il le signale, la liste n'est pas exhaustive. Les problèmes décisionnels sont tous rangés dans des collections de complexité comparable : **les classes de complexité**.

Tableau 2.1: Classement des problèmes informatiques [Ferro et al., 2005]

Classe	Description (des problèmes)
$\sharp P$	Comptage des solutions pour un problème NP
P-complete	Les problèmes les plus difficiles dans $\sharp P$. Un problème est dans cette classe si tout autre problème dans $\sharp P$ peut s'y réduire en temps polynomiale
AM	Peuvent être résolus en temps polynomial par un protocole Arthur-Merlin
BPP	Peuvent être résolus en temps polynomial par un algorithme aléatoire (la réponse est correcte la plupart du temps)
BQP	Peuvent être résolus en temps polynomial par un ordinateur quantique
Co-NP	Réponse négative vérifiable en temps polynomial
Co-NP-complete	Les problèmes les plus difficiles dans Co-NP
DSPACE(f(n))	Résolubles avec une machine déterministe et avec un espace en O(f(n))
DTIME(f(n))	Résolubles avec une machine déterministe et avec un temps en O(f (n))
E	Résolubles en temps exponentiel avec un exposant linéaire
ELEMENTARY	L'union des classes dans la hiérarchie exponentielle
ESPACE	Résolubles en espace exponentiel avec un exposant linéaire
EXP	Comme pour EXPTIME
EXPSPACE	Résolubles en espace exponentiel
EXPTIME	Résolubles en temps exponentiel
FNP	Le correspondant de NP pour les problèmes non décisionnels
FP	Le correspondant de P pour les problèmes non décisionnels
FP^{NP}	Le correspondant de P^{NP} pour les problèmes non décisionnels
FPT	Fixed-parameter tractable : tractable avec un paramètre établi
IP	Résolubles en temps polynomial par un système de preuve interactif
L	Peuvent être résolus en espace logarithmique (petit)
NC	Résolubles efficacement (en temps poly-logarithmique) avec un "ordinateur en parallèle" (processeurs en parallèle)
NE	Résolubles en temps exponentiel (avec un exposant linéaire) avec une machine non-déterministe
NESPACE	Résolubles en espace exponentiel (avec un exposant linéaire) avec une machine non-déterministe
NEXP	Comme pour NEXPTIME
NEXPSPACE	Résolubles en espace exponentiel avec une machine non-déterministe

Tableau 2.2: Classement des problèmes informatiques [Ferro et al., 2005]: suite

Classe	Description (des problèmes)
NEXPTIME	Résolubles en temps exponentiel avec une machine non-déterministe
NL	Réponse affirmative vérifiable en espace logarithmique
NP	Une réponse affirmative au problème peut être vérifiée en temps polynomiale
NP-complete	Les problèmes les plus difficiles (et intéressants) dans NP, ceux qui semblent vraiment ne pas faire partie de P
NP-easy	Un autre nom pour FP^{NP}
NP-equivalent	Les problèmes les plus difficiles dans FP^{NP}
NP-hard	Les problèmes NP-complets ou encore plus difficiles
NSPACE(f (n))	Problèmes pouvant être résolus avec une machine non-déterministe en espace O(f (n))
P	Résolubles en temps polynomial
P-complete	Les problèmes les plus difficiles dans P, à résoudre avec un ordinateur en parallèle (processeurs en parallèle)
PCP	La preuve est vérifiable de façon probabiliste
PH	L'union des classes dans la hiérarchie polynomiale
P^{NP}	Résolubles en temps polynomial avec un oracle 3 pour un problème dans NP
PP	Probabilistically Polynomial : la réponse est correcte avec une probabilité légèrement supérieure à $\frac{1}{2}$
PSPACE	Résolubles avec une mémoire polynomiale mais un temps illimité
PSPACE-complete	Les problèmes les plus difficiles dans PSPACE
RL	Résolubles en espace logarithmique avec un algorithme aléatoire (réponse négative probablement correcte, réponse affirmative correcte avec certitude)
RP	Résolubles en temps polynomial avec un algorithme aléatoire
RLP	Résolubles en espace logarithmique et temps polynomial avec un algorithme aléatoire
UP	Fonctions en temps polynomial non-déterministes non ambiguës
ZPP	Pouvant être résolus par un algorithme aléatoire (réponse toujours correcte et temps de calcul moyen polynomiale)

Nous allons limiter notre analyse sur certaines classes citées ci-haut en tenant compte de leur importance dans notre travail notamment:

L'ensemble P: cet ensemble renferme les problèmes pour lesquels il existe un algorithme de complexité polynomiale pour trouver la solution. La complexité du temps de calcul pour un problème X exprime un rapport entre la taille de ses données et le temps utilisé pour le résoudre. Pour cette raison on indique souvent les problèmes appartenant à cette classe comme étant "efficients" ou "tractable ": bien qu'il soit naturel et raisonnable de considérer comme intraitable dans la pratique un problème qui demande un temps de l'ordre de n^{200} , il existe effectivement très peu de problèmes qui requièrent un temps d'un ordre de grandeur aussi élevé. P est réputé contenir de nombreux problèmes na-

turels, comme le calcul du plus grand commun diviseur. On pourrait alors généraliser P avec l'inclusion de tout autre problème qui peut être résolu en temps polynomiale par une machine de Turing non-déterministe. La classe qui en résulte prend le nom de NP et elle implique une relation triviale : $P \subseteq NP$ [Ferro et al., 2005].

P-complet: Un problème décisionnel appartient à ce classement s'il fait partie de la classe P et si tout problème dans P peut s'y réduire en temps poly-logarithmique en utilisant un ordinateur avec un nombre polynomial de processeurs. En pratique, si X est un problème dans P-complet alors il appartient à P et pour tout problème Y dans P, il existe des constantes c et k telles que Y puisse être réduit en X en $O((log(n))^c)$ avec un ordinateur qui possède $O(n^k)$ processeurs en parallèle [Ferro et al., 2005].

NP: "Non-deterministic Polynomial time" [Ferro et al., 2005]: il s'agit de l'ensemble des problèmes décisionnels pouvant être résolus en temps polynomial avec une machine de Turing non-déterministe. Ils peuvent également être validés par un algorithme polynomial.

L'ensemble NP-complet : Le classement NP-complet contient les problèmes les plus difficiles de NP. Ces problèmes semblent vraiment ne pas faire partie de P (bien que à l'heure actuelle on ne possède pas une réponse définitive, voir la question irrésolue P = N P). Si nous sommes capables de trouver un moyen de résoudre efficacement (algorithme polynomial) un problème de NP-complet, nous pouvons alors utiliser cet algorithme pour résoudre tous les problèmes de NP.

En effet, comme pour P et P-complet, un problème X appartient à NP-complet s'il est dans NP et si tout autre problème dans NP peut s'y réduire. On en déduit qu'une méthode "assez facile" pour prouver qu'un nouveau problème appartient à NP-complet est de montrer d'abord qu'il est dans NP, ensuite le réduire en un problème connu dans NP-complet. Par conséquent il est très utile de connaître certains des problèmes NP-complets: le problème de factorisation en entiers premiers, le problème de sac à dos, le problème du voyageur de commerce, le problème du plus court chemin ... sont des exemples de problèmes NP-complets [Ferro et al., 2005].

Problème de la factorisation: ce problème consiste à écrire un nombre composite sous la forme d'un produit de deux facteurs premiers.

Le problème du plus court chemin: Soit un graphe G = (X, V), où X est l'ensemble des sommets et V est l'ensemble des arcs. On appelle le problème du plus court chemin le problème suivant : étant donné un graphe G, nous associons à chaque arc u un nombre $l(u) \geq 0$ que nous appellerons la longueur de l'arc u. Trouver un chemin élémentaire μ, allant d'un sommet a à un sommet b, qui soit aussi petite que possible [Toubakh, 2020].

Le problème du voyageur de commerce: le problème du voyageur de commerce est l'un des problèmes classiques d'optimisation combinatoire. Il s'agit d'un voyageur en commerce qui souhaite visiter n villes données en passant par chaque ville exactement une fois (établir une tournée). Il commence par une ville quelconque et termine en retournant

à la ville de départ. Le problème consiste à trouver la tournée de longueur minimale passant par toutes les villes et revenant au point de départ. La notion de distance peut être remplacée par d'autres notions comme le temps qu'il met ou l'argent qu'il dépense, dans tous les cas, on parle de coût [Toubakh, 2020].

Le problème de sac à dos: le problème de sac à dos modélise des situations analogues au remplissage d'un sac à dos. Etant donné un sac de capacité b et n objets, où chaque objet j possède un poids et une valeur w_j, le problème qui se pose est: comment remplir le sac de sorte que le poids total des objets choisi n'excède pas la capacité du sac b tout en maximisant la valeur totale [Toubakh, 2020].

Le problème d'appariement minimal: le problème d'appariement minimal (Minimum Matching Problem (MMP)) est le suivant : il s'agit d'apparier les points deux à deux (il en faut donc un nombre pair) de façon que la longueur totale de l'appariement soit minimale. L'ensemble des configurations est constitué d'appariements c'est-à-dire de configurations où chaque point est relié à un et un seul autre point (une configuration acceptable est donc composée de N/2 liens). La fonction de coût est la somme des longueurs des liens de l'appariement[Toubakh, 2020].

L'arbre couvrant minimal [Toubakh, 2020]: le problème de l'arbre couvrant minimal (Minimum Spanning Tree (MST)) consiste à trouver l'arbre passant par tous les points (d'où le nom) dont la somme des longueurs des liens est minimale. L'ensemble des configurations est l'ensemble des arbres couvrants, c'est-à-dire l'ensemble des configurations constituées de N-1 liens et tel qu'il existe toujours un chemin entre deux points quelconques.

En général, lorsqu'un problème est connu pour être NP-complet, on peut raisonnablement penser qu'il est inutile d'en chercher une solution sous forme d'un algorithme de complexité polynomiale.

Les problèmes NP-complet sont tels que si un seul de ces problèmes est résoluble au moyen d'un algorithme de complexité polynomiale, tous les problème NP-complet le seront.

En conclusion, nous venons de voir qu'il existe des problèmes dont le nombre d'opérations croît très rapidement avec la taille des entrées. Parmi les problèmes NP-complets, nous allons prendre le problème de factorisation d'un entier composite en deux nombres premiers. Avant tout cela, introduisons d'abord un chapitre qui nous donne quelques idées sur l'informatique quantique.

Chapitre 3

Les principes fondamentaux de la théorie de l'information quantique

3.1 Introduction

Dans ce chapitre, nous allons donner quelques éléments essentiels dans la théorie de l'information quantique et plus particulièrement ceux qui nous aident à comprendre la méthode de recherche quantique de Grover.

3.2 Définition de quelques concepts clés

L'équation de Schrödinger: C'est l'équation de base de la mécanique quantique. Elle résulte de l'application de l'équation générale de propagation des ondes planes au mouvement de l'onde associée à une particule. Elle est donnée par la formule suivante [Akama, 2015]:

$$i\hbar\frac{\partial}{\partial t}\psi(\vec{r},t) = \hat{H}\psi(\vec{r},t) \tag{3.1}$$

où

- $\psi(\vec{r},t)$ représente une fonction d'onde de deux variables position et temps
- i représente l'imaginaire complexe;
- $\hbar$ la constante de Planck réduite;
- $\hat{H}$ l'opérateur hamiltonien (associé à l'observable énergie totale du système).

L'équation de Schrödinger est une équation différentielle du premier ordre par rapport au temps. Ce qui signifie que la donnée d'un état initial $\psi(\vec{r},t_0)$ suffit à déterminer $\psi(\vec{r},t)$

à tout instant ultérieur t. Ceci n'est valable que si l'évolution n'est pas interrompue par une mesure du système.

Espace de Hilbert [Akama, 2015] : Un espace de Hilbert $\mathscr{H}$ est un espace vectoriel sur les nombres complexes $\mathbb{C}$ muni d'un produit scalaire et complet pour la norme associée à ce produit scalaire. Un élément quelconque, ou vecteur, de l'espace $\mathscr{H}$ est appelé vecteur-ket. Par exemple $|\psi\rangle$

Si $\mathscr{H}$ est un espace de Hilbert de dimension n, on le note $\mathscr{H}^{\otimes n}$.

Soit $\mathscr{H}^{\otimes n}$ un espace de Hilbert, et soit :

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ . \\ . \\ \alpha_n \end{pmatrix} \tag{3.2}$$

un ket quelconque de $\mathscr{H}$. Le conjugué hermitique de $|\psi\rangle$ est le bra $\langle\psi|$:

$$\langle\psi| = |\psi\rangle^{\dagger} = \begin{pmatrix} \alpha_1^{\star} & \alpha_2^{\star} & .... & \alpha_n^{\star} \end{pmatrix} \tag{3.3}$$

où † représente l'opération de transposer et de conjuguer un vecteur ou une matrice et $\star$ représente l'opération de conjugaison complexe.

Le produit scalaire: A tout couple de kets $|\varphi\rangle$ et $|\psi\rangle$ pris dans cet ordre, on associe un nombre complexe, qui est leur produit scalaire et qu'on note : $\langle\varphi|\psi\rangle$.

Voici certaines propriétés de ce produit scalaire [Dan C. Marinescu, 2005]:

$$\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^{\star}; \tag{3.4}$$

$$\langle\varphi|\lambda_1\psi_1 + \lambda_2\psi_2\rangle = \lambda_1\langle\varphi|\psi_1\rangle + \lambda_2\langle\varphi|\psi_2\rangle(\textit{linéarité}). \tag{3.5}$$

On peut définir une norme sur $\mathscr{H}$, dite norme hermitienne : la norme de $|\psi\rangle$ est égale

$$\| \psi \| = \sqrt{\langle\psi|\psi\rangle} \tag{3.6}$$

$$\langle\psi|\psi\rangle \geq 0 \ (\textit{positive}) \tag{3.7}$$

$$\langle\psi|\psi\rangle = 0 \Leftrightarrow |\psi\rangle = 0 \tag{3.8}$$

Pour tout $|\psi\rangle$, $|\varphi\rangle \in \mathscr{H}$, $|\psi\rangle$ et $|\varphi\rangle$ sont orthogonaux si et seulement si $\langle\psi|\varphi\rangle = 0$.

Produit tensoriel: Soient deux espaces de Hilbert $\mathscr{H}^{1\otimes n}$ et $\mathscr{H}^{2\otimes m}$. On appelle

produit tensoriel de $\mathscr{H}^{1\otimes n}$ et $\mathscr{H}^{2\otimes m}$ l'espace $\mathscr{H}$ de dimension nm.

$$\mathscr{H}^{\otimes nm} = \mathscr{H}_1^{\otimes n} \otimes \mathscr{H}_2^{\otimes m} \tag{3.9}$$

A tout couple de kets, $|\psi_1\rangle \in \mathscr{H}_1^{\otimes n}$ *et* $|\psi_2\rangle \in \mathscr{H}_2^{\otimes m}$ on associe $|\psi\rangle^{n*m} = |\psi\rangle_1 \otimes |\psi\rangle_2$

ou tout simplement

$$|\psi\rangle^{nm} = |\psi\rangle_1 |\psi\rangle_2 \tag{3.10}$$

et que l'on appelle produit tensoriel de $|\psi_1\rangle$ et $|\psi_2\rangle$.
Le produit tensoriel a les propriétés usuelles de la multiplication, mais il est non commutatif.

Le bit classique: le bit classique est l'unité de mesure élémentaire de l'information. Un bit peut être défini comme une variable x dont la valeur appartient à un ensemble à deux éléments. Un bit aura donc comme valeur 0 ou 1. Le bit caractérise donc une propriété d'un système classique ne pouvant être que dans deux états différents; ouvert ou fermé, actif ou inactif, noir ou blanc, etc [Akama, 2015].

Bit quantique ou qubit: Un qubit est un vecteur unitaire d'un espace de Hilbert de dimension 2 sur $\mathbb{C}$. L'expression du qubit est donnée par:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \text{ [Akama, 2015]} \tag{3.11}$$

où α et β sont des coefficients complexes, ils représentent les amplitudes de probabilité d'obtenir l'état 1 et l'état 0 respectivement lors d'une mesure de l'état ψ . Ces deux états constituent une base orthogonale de l'espace de Hilbert du système. Ces coefficients satisfont la condition de normalisation suivante :

$$|\alpha|^2 + |\beta|^2 = 1 \tag{3.12}$$

$|\alpha|^2$: représente la probabilité d'avoir le bit 0, $|\beta|^2$: représente la probabilité d'avoir le bit 1. $|0\rangle$ et $|1\rangle$: représentent deux états orthogonaux dans le système quantique.

3.3 Postulats de la mécanique quantique

3.3.1 Introduction

Avec la description du monde microscopique, les lois et les principes de la mécanique classique changent. Il faut donc établir les bases ou les fondements de la mécanique quantique (les postulats de la mécanique quantique).

3.3.2 Enoncés des postulats [Cohen-Tannoudji et al., 2020]

1. Premier postulat: le premier postulat explique le principe de superposition et s'énonce comme suit: A un instant t fixé, l'état d'un système est complètement défini par la connaissance de son vecteur d'état $|\psi(t)\rangle$.

Un état quantique peut être représenté, à chaque instant t, par un vecteur dans un espace de Hilbert $\mathscr{H}$.

Il est habituellement noté sous la forme d'un ket $|\psi(t)\rangle$. Si on veut être précis, ce sont des vecteurs normalisés à une phase près $\Big(|\psi\rangle \sim |\psi\rangle e^{i\alpha}\Big)$.

La superposition des états en est la conséquence directe:

le spin, par exemple, est calculé (en notation bra-ket) par:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\Big(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle\Big) \tag{3.13}$$

où $|\uparrow\downarrow\rangle$ représente l'état du spin up et du spin down et $|\downarrow\uparrow\rangle$ représente l'état du spin down et du spin up; qui implique la superposition des états de spin communs pour deux particules de spin 1/2 (fermions). Dans les espaces hilbertiens, toute combinaison linéaire fait partie de l'espace des états : si $|\psi_1\rangle$ et $|\psi_2\rangle$ décrivent deux états du système et si λ est un complexe, alors la combinaison linéaire $\lambda|\psi_1\rangle + |\psi_2\rangle$ décrit un état du système.

2. Deuxième postulat: principe de correspondance ou description quantique d'une grandeur physique

À toute propriété observable, appelé observable A, (position, quantité de mouvement, spin, impulsion/énergie...), correspond un opérateur hermitien linéaire, $\hat{A}$, agissant sur les vecteurs d'un espace de Hilbert $\mathscr{H}$ [Marchildon, 2000].

L'opérateur de position dans l'espace des positions est

$$\hat{R} = \mathbf{r} \tag{3.14}$$

l'opérateur de quantité de mouvement

$$\hat{P} = \mathbf{p} \tag{3.15}$$

l'opérateur d'énergie potentielle classique:

$$V(\vec{R}) = V(\mathbf{r}); \tag{3.16}$$

Il s'énonce comme suit:

Toute grandeur physique $\mathscr{A}$ est décrite par une observable A agissant dans l'espace des états.

3. Troisième postulat: il exprime le principe de quantification et s'énonce comme suit: la mesure d'une grandeur physique $\mathscr{A}$ ne peut donner comme résultat qu'une des valeurs propres de l'observable A correspondante. Le troisième postulat explique la quantification que l'on observe pour certaines grandeurs. Les valeurs propres sont les valeurs pouvant résulter d'une mesure idéale de cette observable. Les vecteurs propres étant l'état quantique du système immédiatement après la mesure et résultant de cette mesure (cf. postulat V).

$$\hat{A}|u_n\rangle = a_n|u_n\rangle \tag{3.17}$$

où $\hat{A}$ est l'observable, $|u_n\rangle$, le vecteur propre et a_n la valeur propre correspondante.

Les états propres de tout observable $\hat{A}$ sont complets et forment une base de Hilbert (orthonormée) dans l'espace de Hilbert : tout vecteur $|\psi(t)\rangle$ peut se décomposer de manière unique sur la base des vecteurs propres $\{|u_i\rangle\}$, tel que:

$$|\psi\rangle = c_1|u_1\rangle + c_2|u_2\rangle + ... + c_n|u_n\rangle \tag{3.18}$$

4. Quatrième postulat: si l'on mesure la grandeur physique $\mathscr{A}$ sur un système dans l'état normé $|\psi\rangle$, la probabilité $P(a_n)$ d'obtenir comme résultat de la mesure la valeur propre a_n est:

$$P(a_n) = |\langle u_i|\psi\rangle|^2 = |c_n|^2 \tag{3.19}$$

où $|c_n|$ sont les coefficients complexes si a_n est non dégénérée;

$$P(a_n) = \sum_{i=1}^{g_n} |\langle u_n|\psi\rangle|^2 = \sum_{i=1}^{g_n} |c_n^i|^2 \tag{3.20}$$

si a_n est dégénérée g_n fois et $|u_n^i\rangle = |u_n^1\rangle, |u_n^2\rangle, ..., |u_n^{g_n}\rangle$ $(i = 1, g_n)$

La mécanique quantique est donc indéterministe. Il n'y a aucun moyen de prévoir à l'avance le résultat d'une mesure. On ne peut savoir que la probabilité d'obtenir tel ou tel résultat.

Dans le cas d'un spectre continu et non-dégénéré, la probabilité $dP(a)$ pour obtenir un résultat entre a et $a + da$ est :

$$dP(a) = |\langle \nu_a|\psi\rangle|^2 da, \tag{3.21}$$

où $|\nu_a\rangle$ est le vecteur propre associé à la valeur propre a de l'observable A.

5. Cinquième postulat: il explique la réduction du paquet d'onde. La réduction du paquet d'onde est un concept de la mécanique quantique selon lequel, après une mesure, un système physique voit son état entièrement réduit à celui qui a été mesuré (cf. principe d'incertitude). Il s'énonce comme suit: Si le résultat de mesure d'une grandeur physique $\mathscr{A}$ sur un système physique dans l'état $|\psi\rangle$ donne comme résultat a_n , alors l'état de ce système immédiatement après la mesure est la projection normée $\frac{P_n|\psi\rangle}{\sqrt{\langle\psi|P_n|\psi\rangle}}$ de ψ sur le

sous espace propre associé à a_n.

6. Sixième postulat: l'évolution dans le temps du vecteur d'état $|\psi(t)\rangle$ est gouvernée par l'équation de Schrödinger qui s'écrit dans le cas général:

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle \tag{3.22}$$

où H(t)est l'hamiltonien du système. C'est l'observable associée à l'énergie totale de ce système.

3.4 Génération des portes logiques

Les portes logiques sont des circuits qui nous permettent de réaliser des oracles: tout comme dans le cas classique, l'oracle quantique est constitué par des portes logiques contrôlées [Jaffali, 2020]. La fonction de l'oracle est donc de faire des opérations logiques.

Parmi les porte logiques, nous ne donnons que quelques unes qui vont être utilisées dans la réalisation de l'algorithme de Grover. Les figures ci-dessous (Figure 3.1 à 3.9) sont obtenues à l'aide d'une librairie qiskit [Wille et al., 2019b] et d'un langage python. Nous allons chaque fois donner le circuit, sa représentation matricielle, ainsi que son fonctionnement sur les vecteurs de base $|0\rangle$ et $|1\rangle$ pour chaque porte logique.

3.4.1 Génération de la porte d'Hadamard

La porte d'Hadamard, souvent notée H, est définie par la transformation suivante de la base canonique :

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{3.23}$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{3.24}$$

Cette porte H peut être représentée sous forme matricielle comme suit :

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.25}$$

Elle est souvent utilisée pour générer des états complètement parallélisés (dont l'écriture fait intervenir tous les états de base du système avec la même amplitude de probabilité).

3.4.2 Génération de l'opérateur de négation X

Elle modifie les vecteurs de la base comme suit: $X|0\rangle = |1\rangle$ et $X|1\rangle = |0\rangle$.

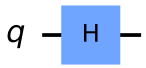

Figure 3.1: porte d' Hadamard

Figure 3.2: porte X

Sa représentation matricielle associée est donc la suivante $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

3.4.3 Génération de l'opérateur Y

Figure 3.3: porte Y

Sa représentation matricielle est la suivante:

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{3.26}$$

3.4.4 Génération de l'opérateur Z

Figure 3.4: porte Z

Sa représentation matricielle est la suivante:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.27}$$

Les trois opérateurs X, Y et Z sont appelés les opérateurs (ou les matrices) de Pauli.

Les quatre matrices X, Y, Z et H sont en effet étroitement liées. On peut ainsi rapidement établir les relations suivantes :

$X^2 = Y^2 = Z^2 = H^2 = \mathbb{I}_2$ où $\mathbb{I}_2$ est la matrice identité de dimension 2.

$XY = iZ,\ YZ = iX\ et\ ZX = iY$

$HXH = Z,\ HYH = -Y\ et\ HZH = X$

3.5 Opérateurs sur un système à plusieurs qubits

3.5.1 Portes à deux qubits

Tout comme un système à un qubit, un système à 2-qubits évolue selon une transformation unitaire. Un opérateur unitaire agissant sur un système à 2-qubits sera alors représenté par une matrice carrée d'ordre 4. De façon générale, toute opération unitaire sur un système à n-qubits peut être représentée par une matrice carrée d'ordre $N = 2^n$. On pourra ainsi manipuler des états à plusieurs qubits, et réaliser des opérations plus complexes sur ces derniers.

Plus précisément, et afin d'expliciter le calcul du produit tensoriel, si les deux matrices U_1 et U_2 s'expriment explicitement comme

$$U_1 = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}, U_2 = \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} \tag{3.28}$$

alors la matrice $U = U_1 \otimes U_2$ est égale à [Jaffali, 2020]:

$$U = \begin{pmatrix} a_1 \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} & b_1 \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} \\ c_1 \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} & d_1 \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_1a_2 & a_1b_2 & b_1a_2 & b_1b_2 \\ a_1c_2 & a_1d_2 & b_1c_2 & b_1d_2 \\ c_1a_2 & c_1b_2 & d_1a_2 & d_1b_2 \\ c_1c_2 & c_1d_2 & d_1c_2 & d_1d_2 \end{pmatrix} \tag{3.29}$$

Il apparaît donc intéressant d'étudier les principales portes logiques réversibles agissant sur des états à plusieurs qubits.

3.5.1.1 La porte $c-NOT$ ou Controlled-NOT

La porte $c-NOT$ est utilisée pour appliquer la porte NOT (ou porte X) en fonction de la valeur du premier qubit. Elle fonctionne de la manière suivante [Jaffali, 2020]:

$$\begin{array}{cc} \textit{état d'entrée} & \textit{état de sortie} \\ |00\rangle & |00\rangle \\ |01\rangle & |01\rangle \\ |10\rangle & |11\rangle \\ |11\rangle & |10\rangle \end{array} \tag{3.30}$$

La porte $c-NOT$ agit en effet sur un système à deux qubits. Elle repose sur le principe de porte contrôlée, c'est à dire que le premier bit sert de contrôle (bit de contrôle) et le second bit (bit cible) subit ou non la transformation associée (ici la porte X), en fonction de l'état du bit de contrôle. Comme nous savons comment l'opérateur $c-NOT$ transforme les vecteurs de la base de calcul $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, on peut alors le représenter par la matrice suivante:

$$c-NOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.31}$$

Ainsi, la valeur du bit cible est inchangée si le bit de contrôle vaut 0 et la valeur du bit cible est changée si le bit de contrôle vaut 1. En fait, le bit cible vaut à la sortie la somme modulo 2 des deux bits d'entrée, tandis que le bit de contrôle reste inchangé. On note alors, $c-NOT$: $(x, y) \rightarrow (x, x \oplus y)$.

En plus de la représentation matricielle, on peut introduire la représentation sous forme de circuit pour les opérateurs à plusieurs qubits. Les fils horizontaux matérialisent un qubit ou une particule, et les fils verticaux correspondent à une interaction entre les qubits, comme le fait qu'un qubit puisse contrôler l'action sur un second qubit. Le symbole de somme entouré représente la somme binaire modulo 2 pour la porte $c-NOT$.

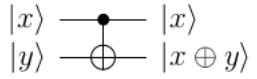

Figure 3.5: Représentation sous la forme d'un circuit quantique de l'opérateur à 2-qubits $c-NOT$ [Jaffali, 2020].

Enfin, l'application de l'opérateur $c-NOT$ à un état à 2-qubits quelconque $|\psi\rangle$ donne :

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \rightarrow^{c-NOT} \alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle + \delta|10\rangle. \tag{3.32}$$

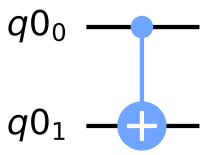

Figure 3.6: porte $c - NOT$

3.5.1.2 La porte SWAP (de l'anglais «SWAP= échange »)

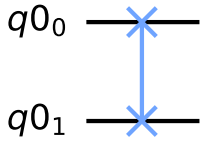

Figure 3.7: La porte SWAP

Comme son nom l'indique, la porte SWAP échange la place des deux qubits passés en paramètre : **SWAP** : $(x, y) \rightarrow (y, x)$. La porte SWAP se compose d'une succession de 3 portes $C - NOT$, avec alternance du bit de contrôle:

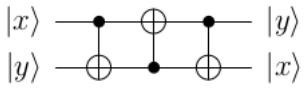

Figure 3.8: Représentation sous la forme d'un circuit quantique de la porte SWAP [Jaffali, 2020]

La matrice associée à l'action de la porte SWAP sous un registre à 2-qubits est la suivante:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.33}$$

L'application de l'opérateur SWAP à un état quelconque $|\psi\rangle$ donne:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \rightarrow^{SWAP} \alpha|00\rangle + \beta|10\rangle + \gamma|01\rangle + \delta|11\rangle. \tag{3.34}$$

3.5.2 Porte à trois qubits

L'opérateur TOF: mis en place par Tommaso Toffoli en 1980, cet opérateur TOF peut être considéré comme un $c-c-NOT$ (controled-controled-NOT). Cette porte apporte une grande aide dans la résolution du problème de réversibilité des portes logiques classiques.

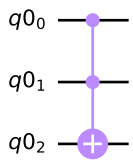

Figure 3.9: La porte de Toffoli

La représentation matricielle de la porte de Toffoli est la suivante [Jaffali, 2020]:

$$Toff = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.35}$$

Cette porte prend 3 qubits en entrée: les deux premiers servant pour le contrôle et le troisième étant le qubit cible. Cet opérateur réalise ainsi l'application unitaire suivante, TOF : $(x, y, z) \rightarrow (x, y, z \otimes xy)$. Le circuit correspondant à la porte de Toffoli est donc le suivant [Jaffali, 2020]:

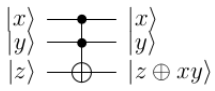

Figure 3.10: Représentation sous la forme d'un circuit quantique de la porte Toffoli

La porte de Toffoli transforme les états de base d'un système à trois qubits comme

état d'entrée	*état de sortie*	
$\|000\rangle$	$\|000\rangle$	
$\|001\rangle$	$\|001\rangle$	
$\|010\rangle$	$\|010\rangle$	
suit: $\|011\rangle$	$\|011\rangle$	[Jaffali, 2020]
$\|100\rangle$	$\|100\rangle$	
$\|101\rangle$	$\|101\rangle$	
$\|110\rangle$	$\|111\rangle$	
$\|111\rangle$	$\|110\rangle$	

La porte de Toffoli permettra, en considérant certains qubits comme auxiliaires, de modéliser toutes les portes logiques classiques qui ne sont pas facilement modélisables à l'aide des portes quantiques usuelles. Enfin, cette porte, ainsi que la porte X et SWAP, feront partie d'un ensemble universel d'opérateurs logiques permettant la génération de toutes les portes unitaires à n-qubits.

3.6 Mesure d'un système quantique

Un des aspects fondamentaux particulier de la physique quantique est que l'on ne peut pas mesurer les coefficients α et β de l'état quantique $|\psi\rangle = \alpha|0\rangle+\beta|0\rangle$.

Partons d'un état quantique de norme 1 : $\psi = \alpha|0\rangle + \beta|1\rangle$ avec $|\alpha|^2 + |\beta|^2 = 1$.

La mesure de l'état quantique $|\psi\rangle$ va renvoyer l'un des bits classiques 0 ou 1

- 0 avec une probabilité de $|\alpha|^2$
- 1 avec une probabilité de $|\beta|^2$

Exemple

Soit $|\psi\rangle = \frac{1-i}{\sqrt{3}}|0\rangle + \frac{1+2i}{\sqrt{15}}|1\rangle$

$|\alpha|^2 = |\frac{1-i}{\sqrt{3}}|^2 = \frac{2}{3}$

et

$|\beta|^2 = |\frac{1+2i}{\sqrt{15}}|^2 = \frac{1}{3}$

Ainsi $|\alpha|^2 + |\beta|^2 = 1$. Si on mesure $|\psi\rangle$ alors on obtient 0 avec une probabilité $\frac{2}{3}$ et 1 avec une probabilité $\frac{1}{3}$

On retient qu'à partir de l'état $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, avec α, $\beta \in \mathbb{C}$

- on ne peut pas mesurer les coefficients α et β

- La mesure de $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ avec $|\alpha|^2 + |\beta|^2 = 1$ va renvoyer l'un des bits classiques 0 ou 1; 0 avec une probabilité de $|\alpha|^2$ et 1 avec une probabilité de $|\beta|^2$

- la mesure transforme le qubit $|\psi\rangle$ en $|0\rangle$ ou en $|1\rangle$, les coefficients α et β ont disparu après mesure.

De manière plus générale, le principe de mesure en mécanique quantique est aussi associé à la notion d'observable, permettant de déterminer de manière probabiliste la valeur de la grandeur mesurée.

Il est admis par la communauté scientifique que l'on ne peut généralement pas établir de manière déterministe le résultat d'une mesure pour un état quantique, mais seulement déterminer les résultats possibles et les probabilités associées à chaque résultat. Toutes ces informations sont encodées dans le concept d'observable)

• **Définition**: **une observable**, notée A est un opérateur hermitien linéaire qui est associé à une propriété d'un système physique et agit sur des vecteurs de l'espace de Hilbert $\mathscr{H}$ (deuxième postulat). L'ensemble de ses vecteurs propres constitue une base orthonormée.

• Un opérateur A est **hermitien** si et seulement si il vérifie la relation $A = (A^*)^t$.

3.7 Ordinateur quantique

3.7.1 Introduction

Un ordinateur quantique tel que défini par André Berthiaume [Berthiaume, 1995] est similaire à une machine de Turing (ordinateur classique) tant dans leurs propriétés que dans leurs fonctionnements. Tous les deux possèdent un ruban semi- infini de cases contenant un caractère provenant d'un alphabet fini et une tête de lecture/écriture elle même possédant un ensemble infini d'états. Le tout se contrôle par une fonction de transition qui, en fonction de l'état de la tête et du caractère sous la tête, change l'état de la tête, écrit un caractère sur le ruban et déplace la tête à gauche ou à droite. La fonction de transition a la forme suivante: δ: $(b, q) \rightarrow (b', q', d)$ c'est-à- dire qu'en fonction du caractère b sous la tête de lecture et de l'état interne q, la machine écrit le caractère b', change son état interne pour q' et se déplace d'une case en direction d.

La différence principale provient du type d'information contenue dans chacune des cases du ruban. Supposons que l'alphabet de ruban soit l'ensemble $\{0, 1\}$. Alors que les cases de la machine de Turing peuvent contenir uniquement soit 0, soit 1, les cases d'un ordinateur quantique peuvent contenir un mélange de 0 et de 1 appelé superposition quantique. C'est cette superposition quantique qui est à la base de la puissance des ordinateurs quantiques face à leurs équivalents classiques.

3.7.2 Fonctionnement de l'ordinateur quantique

3.7.2.1 Support de l'information quantique

Contrairement à l'ordinateur classique qui utilise des bits classiques, c'est-à-dire que la valeur du bit classique est 0 ou 1, un ordinateur quantique utiliserait des bits quantiques, les qubits qui se distinguent des bits classiques par des symboles à gauche et à droite de 0 et de 1. Ces derniers peuvent avoir en même temps les valeurs 0 et 1 (comme mentionné ci-haut), donc une superposition de deux états. La puissance du calcul quantique trouve son origine dans la faculté d'un système quantique de se trouver dans une superposition d'un nombre gigantesque d'états : un nombre exponentiel en fonction du nombre des qubits manipulés. Nous sommes conscients que les ordinateurs classiques ont atteint un niveau de performance et de compacité considérable et continuent également de progresser. Cependant, il existe des problèmes "difficiles" que ces ordinateurs classiques ne peuvent pas résoudre efficacement. Calculer un produit, par exemple, est un problème facile. Le temps de calcul ne varie que comme le carré du nombre n de bits utilisés. Si on sait calculer 2×2, on saura calculer tout produit de nombres utiles. Le temps d'exécution d'un calcul facile varie, en fonction du nombre de bits, comme un polynôme. La factorisation est, en revanche, difficile. Le temps de calcul du meilleur algorithme classique connu augmente très rapidement avec le nombre de bits [Brune and Raimond, 2005].

3.7.2.2 Suprématie de l'ordinateur quantique

De même que le qubit, le registre d'entrée à n qubits d'un ordinateur quantique peut être préparé dans une superposition quantique des 2^n états possibles, correspondant aux 2^n nombres codés sur n bits. Une machine traitant de façon cohérente un registre d'entrée quantique pourrait donc être finalement dans une superposition de tous les résultats possibles du calcul effectué par la machine; c'est ce qu'on appelle **le parallélisme quantique**, qui donne ainsi une puissance à l'ordinateur quantique vis à vis de tout calculateur classique. On peut, par exemple, obtenir une superposition de toutes les valeurs d'une fonction f avec un nombre d'opérations du même ordre que pour obtenir une seule de ces valeurs sur une machine classique. Malheureusement, mesurer un qubit fournit, de façon aléatoire, une des valeurs possibles représentées dans la superposition. On obtient finalement, en « lisant » le registre de sortie, un ensemble de bits qui représente une valeur de la fonction f pour un argument aléatoire.

On peut, en revanche, rendre la mesure de qubits efficace en utilisant une autre propriété quantique essentielle : l'existence, pour un système composite, d'états non-séparables appelés "états intriqués". Ces états ne se mettent pas sous la forme d'un état produit d'états individuels de chaque qubit.

Etats intriqués [Engo, 2020] : pour fixer les idées, nous donnons un exemple d'état intriqué issus du produit tensoriel de deux états.

Soient deux états

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle \; et \; |\psi_2\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{3.36}$$

On voit que $|\psi_1\rangle$ peut-être écrit sous forme de produit tensoriel des états constituants $|\psi_1\rangle$ et $|\psi_2\rangle$:

$$|\psi_1\rangle = |0\rangle(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \tag{3.37}$$

Par contre $|\psi_2\rangle$ ne peut pas être factorisé. Ainsi, on dit que l'état $|\psi_2\rangle$ est intriqué parce qu'il ne peut pas être écrit sous forme de produit tensoriel des états constituants $|\psi_1\rangle$ et $|\psi_2\rangle$.

3.7.3 Structure d'un ordinateur quantique

Nous nous sommes d'abord attachés à déterminer l'architecture d'un ordinateur quantique. Tout calcul quantique peut se ramener à une série de manipulations d'un ou deux qubits dans des « portes logiques quantiques ». Tout comme en logique classique, il suffit de réaliser un petit nombre de portes élémentaires pour pouvoir ensuite réaliser un calcul quantique quelconque en construisant un réseau complexe de portes.

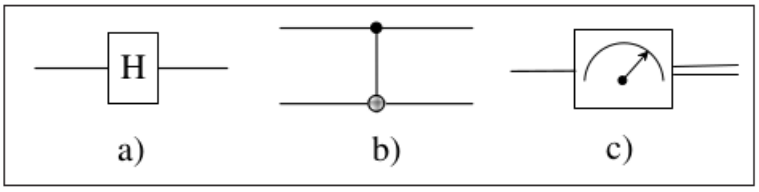

Figure 3.11: Représentation symbolique usuelle d'opérations quantiques élémentaires : (a) porte de Hadamard, (b) porte CNot (c) mesure de l'état logique [Brune and Raimond, 2005]

3.7.4 Etat des lieux de la recherche sur l'ordinateur quantique

En 1981, le physicien américain Paul Benioff tente de démontrer la possibilité théorique des ordinateurs quantiques en décrivant le premier modèle de mécanique quantique adaptée à un ordinateur. Il y détaillait une machine de Turing fonctionnant selon les principes de la mécanique quantique [Benioff, 1981].

Il faut cependant attendre 1982 et les théories du physicien américain Richard Feynman pour que la véritable histoire des ordinateurs quantiques commence. Cette année-là, Richard Feynman a théorisé ce que l'on appelle aujourd'hui **les simulateurs quantiques** [Yefsah and Sayrin, 2022].

Pour démontrer sa théorie, Richard Feynman utilise le principe de superposition d'états des particules élémentaires comme hypothèse. Le physicien imagine des dispositifs

conçus pour fournir des informations sur des problèmes physiques spécifiques. A travers ses recherches, Feynman a tenté de montrer qu'un ordinateur classique connaîtrait un ralentissement exponentiel dans la simulation des phénomènes quantiques, contrairement à son hypothétique simulateur quantique universel.

Enfin, en 1985, le physicien britannique David Deutsch [Deutsch, 1985b] réussit à démontrer la validité de la théorie de Feynman. Il décrit le premier ordinateur quantique universel et présente ses caractéristiques, dont fait partie le parallélisme quantique causé par la superposition des états, pour le calcul en parallèle et non plus en séquence, comme c'est le cas dans les ordinateurs traditionnels.

3.7.5 Apparition des premiers prototypes

Le premier prototype d'ordinateur quantique a été présenté en 1998 par International Business Machines(IBM) dans son centre de recherche d'Almaden à San José en Californie [Steffen et al., 2011]. Il disposait de 2 qubits. En 2001, le centre de recherche d'Almaden présente le premier processeur quantique à 7 qubits, composé d'une seule molécule à 7 spins nucléaires. Le 13 février 2007, l'entreprise D-Wave Systems montre publiquement **Orion**, le premier ordinateur quantique adiabatique de 16 qubits, c'est-à-dire isolé et résistant à la chaleur [Bouchareb and Laboudi, 2012].

En février 2016, IBM met à disposition publiquement le processeur **IBM Quantum Experience** [Singh and Singh, 2016], le premier ordinateur quantique connecté au cloud. Il dispose d'un processeur de 5 qubits.

En mai 2017, IBM annonce avoir construit un processeur 17 qubits plus élaboré qui pourrait servir de base aux systèmes commerciaux. La même année, Google sort un processeur de 20 qubits et annonce vouloir atteindre 49 qubits d'ici la fin de l'année [Feynman and IOP, 2016].

En février 2018, Google annonce une nouvelle étape avec un processeur informatique quantique de 72 qubits appelé **Bristlecone**. En 2018 également, Intel confirme le développement d'une puce de test supraconductrice de 49 qubits, appelée **Tangle Lake** [Telescope, 2018]. En 2019, IBM présente son premier ordinateur quantique commercial doté de 20 qubits, le **IBM Q System One** [Koppenhöfer et al., 2020].

Comme on peut le constater, les prototypes n'ont pas manqué depuis deux décennies. La recherche bat son plein, mais de nombreux défis restent encore à maîtriser. L'augmentation du nombre de qubits, et les contraintes inhérentes, représente un des nombreux obstacles que les ordinateurs quantiques devront surmonter.

L'informatique quantique est désormais une réalité, mais certains chercheurs et analystes mettent en garde contre des attentes démesurées. Les grandes annonces des constructeurs et le battage médiatique donnent l'illusion d'une arrivée prochaine des ordina-

teurs quantiques, mais les ordinateurs quantiques universels prendront encore beaucoup de temps avant d'arriver.

3.7.6 Contraintes

Le problème de la réalisation pratique de l'ordinateur quantique est beaucoup plus ardu. Les systèmes physiques utilisés pour représenter les qubits doivent satisfaire des conditions sévères. Idéalement, ils doivent être manipulables individuellement, mais en grand nombre. Ils doivent aussi pouvoir être initialisés dans un état quantique précis. On doit pouvoir mesurer leur état final pour « lire » le résultat du calcul. Ils doivent interagir fortement entre eux, pour réaliser la dynamique conditionnelle des portes logiques quantiques. Finalement, ils doivent être formidablement bien isolés du milieu extérieur. Très rapidement, la superposition quantique devient une simple alternative probabiliste classique (0 ou 1 plutôt que 0 et 1). Cette « décohérence », même si elle reste faible à l'échelle d'un qubit individuel, devient très rapidement grande lorsqu'elle s'attaque à la superposition d'un grand nombre de qubits. Elle ramène l'ordinateur quantique dans le monde, plus habituel, du calcul classique. Il est donc essentiel que la durée de vie des qubits soit beaucoup plus longue que le temps nécessaire pour réaliser une porte.

Nous suggérons aux lecteurs de consulter la référence [Couteau, 2005] pour les détailles sur l'état d'avancement de l'ordinateur quantique ainsi que les critères de DiVincenzo et 'Carte routière' de l'Ordinateur Quantique

En conclusion, nous n'avons pas donné toute la théorie de l'information quantique. Nous nous sommes intéressés aux principaux outils qui nous aideront à commenter l'algorithme de Grover. La formulation de l'algorithme de Fermat sous forme d'un algorithme de recherche d'un carré pour factoriser un entier n donné, s'identifie à l'algorithme de Grover comme nous allons le voir dans le chapitre 4.

Chapitre 4

Approche quantique de la méthode de Fermat

4.1 Introduction

Pierre de Fermat, né à Beaumont-de-Lomagne en 1601 et décédé à Castres en 1665, est l'un des grands génies des mathématiques vivant au "Siècle des génies"[Unguru, 1976].

Il fut l'un des plus grands mathématiciens français du $XVII^e$ siècle, contemporain de René Descartes et de Blaise Pascal. Sa méthode consiste à factoriser un entier composite en deux nombres premiers p et q. Avec cette méthode, nous cherchons premièrement la partie entière de la racine carrée du nombre à factoriser, puis nous cherchons à écrire ce nombre sous la forme d'une différence de deux carrés. Dans ce chapitre, nous allons également voir que la factorisation avec cette méthode a une complexité qui augmente rapidement avec la taille du nombre à factoriser, ce qui fait que ce problème soit classé parmi les problèmes NP-complets. Néanmoins, il existe un algorithme quantique de Grover qui a une complexité polynomiale quantique, qui soit son équivalent quantique comme nous allons le voir dans ce chapitre.

4.2 Algorithme de Fermat

Soit un nombre n à factoriser en deux nombres premiers a et b.

Lemme: L'ensemble des couples $(a, b) \in \mathbb{N}^2$ tels que $n = ab$ avec $a \geq b$, et celui des couples $(r, s) \in \mathbb{N}^2$ tels que

$$n = r^2 - s^2 \tag{4.1}$$

sont en relation.

Démonstration : Soient A l'ensemble des couples $(a, b) \in \mathbb{N}^2$ tels que $n = ab$ avec

$a \geq b$, et B l'ensemble des couples (r,s) $\in \mathbb{N}^2$ tels que

$$n = r^2 - s^2 \tag{4.2}$$

Les applications f :$A \to B$ et g : $B \to A$ définies par:

$$f((a,b)) = \left(\frac{a+b}{2}, \frac{a-b}{2}\right) \tag{4.3}$$

et

$$g((r,s)) = (r+s, r-s) \tag{4.4}$$

sont réciproques l'une de l'autre. En effet, il suffit de remarquer que pour tout (a,b) $\in \mathbb{N}^2$, on a l'égalité

$$ab = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2 \tag{4.5}$$

et que si $n = r^2 - s^2$ où (r,s) $\in \mathbb{N}^2$, alors

$$n = (r+s)(r-s) \tag{4.6}$$

et l'on a $r + s \geq r - s$.

Dans le cas où n est le produit de deux entiers proches l'un de l'autre, il est facile d'écrire n comme une différence de deux carrés, et donc de factoriser n. C'est l'idée de Fermat.

Soit l'entier n à factoriser et soit $[\sqrt{n}]$ la partie entière de $\sqrt{n}$. Supposons que l'on ait $n = ab$ où a et b sont proches l'un de l'autre, avec $a \geq b$.

Posons

$$r = \frac{a+b}{2} \tag{4.7}$$

et

$$s = \frac{a-b}{2} \tag{4.8}$$

On a

$$n = r^2 - s^2 \tag{4.9}$$

L'entier s est petit et r est donc plus grand que $[\sqrt{n}]$ tout en lui étant proche. Par suite, il existe un petit entier naturel u tel que $([\sqrt{n}]+u)^2 - n$ soit un carré. Afin de déterminer un tel entier u, on examine successivement les entiers $[\sqrt{n}]+1, [\sqrt{n}]+2, ..$ et on teste pour chacun d'eux si son carré moins n est un carré. Si l'on y parvient, on obtient n comme une différence de deux carrés, ce qui fournit une factorisation de n

Exemple

Soit n= 2813 et $S = [\sqrt{2813}] = 53$, la partie entière de la racine carré de n. Alors nous calculons $(53+u)^2 - 2813$, avec u qui prend les valeurs suivantes: $u = 1, 2, 3, \cdots$ et

nous examinons pour chaque valeur de u, l'expression $(53+u)^2-2813$ jusqu'à ce qu'on trouve un carré parfait ce qui implique une racine parfaite.

Tableau 4.1: Recherche d'une racine carrée parfaite

S	u	$S+u$	$(S+u)^2$	$(S+u)^2-n$	$\sqrt{(S+u)^2-n}$
53	1	54	2916	103	10,1489
53	2	55	3025	212	14,5602
53	3	56	3136	323	17,9722
53	4	57	3249	436	20,8806
53	5	58	3364	551	23,4734
53	6	59	3481	668	25,8457
53	7	60	3600	787	28,0535
53	8	61	3721	908	30,133
53	9	62	3844	1031	32,1092
53	10	63	3969	1156	34

Ainsi donc 2813 s'écrit sous forme d'une différence de deux carrés: $2813=(63)^2-(34)^2$ et nous avons 2813 qui se factorise comme suit: $2813=(63-34)(63+34)=29\times 97$.

Nous voyons ici que le nombre u est égal à 10 parce que nous avons calculé $S+1$, $S+2$, ..., $S+10$ pour trouver le carré parfait "1156" et enfin déduire de sa racine parfaite "34". Le nombre u est donc appelé le nombre d'itérations.

4.2.1 Complexité de l'algorithme de Fermat

4.2.1.1 Expression analytique de la complexité

Nous partons de

$$[\sqrt{n}]=[\sqrt{pq}] \tag{4.10}$$

La méthode consiste a chercher

$$R=\frac{p+q}{2} \tag{4.11}$$

sous la forme de $A+u$ avec

$$A=[\sqrt{pq}] \tag{4.12}$$

$$[\sqrt{pq}]+u=\frac{p+q}{2}=R \tag{4.13}$$

$$u=\frac{p+q}{2}-[\sqrt{pq}] \tag{4.14}$$

Posons $t=q-p$ avec $q>p$, on obtient

$$u=\frac{p+p+t}{2}-\sqrt{p(p+t)} \tag{4.15}$$

$$u=p+\frac{t}{2}-p\sqrt{1+\frac{t}{p}} \tag{4.16}$$

En sachant que

$$\sqrt{1+\frac{t}{p}} = 1 + \frac{t}{2p} - \frac{t^2}{8p^2} + \cdots \tag{4.17}$$

l'équation (4.16) devient:

$$u = p + \frac{t}{2} - p - \frac{t}{2} + \frac{t^2}{8p} + \cdots \tag{4.18}$$

$$u \simeq \frac{t^2}{8p} \tag{4.19}$$

(où u est le nombre d'itérations ou opérations élémentaires). Pour démontrer que l'algorithme de Fermat a une complexité qui augmente rapidement avec la taille du nombre à factoriser, on fixe p et on varie q(c'est-à-dire que q va également prendre les valeurs croissantes des nombres premiers).

Si la différence entre p et q est très grande, nous avons plusieurs itérations. Si c'est le contraire, nous avons peu d'itérations.

4.2.1.2 Implémentation de l'algorithme de Fermat

La factorisation avec la méthode de Fermat est la suivante:

$([\sqrt{n}] + u)^2 - n = un\ carré.$

posons $A = [\sqrt{n}]$ et $C = A^2 - n$

tant que C n'est pas un carré:

calculer $C \leftarrow (A+1)^2 - n = A^2 - n + 2A + 1$

$C \leftarrow C + 2A + 1$

incrémenter $A \leftarrow A + 1$,

Calculer une racine carrée s de C et retourner $(A+s)(A-s)$

A l'aide d'un code en langage C, nous avons utilisé cette implémentation et nous avons factorisé le nombre 148194198791 = 385013 x 384907.

4.2.2 Dépendance du nombre d'itérations en fonction de la différence entre p et q ou de la taille n

Soit n le nombre à factoriser; fixons $p = 2$ et évaluons le nombre d'itérations u pour des valeurs de q croissantes.

- n=10 (composé de deux chiffres); $u \simeq \frac{p+q}{2} - \sqrt{n} = 3.5 - 3.162 = 0.337$

- n=26 (composé de 2 chiffres; $q = 13$); $u \simeq \frac{p+q}{2} - \sqrt{n} = 7.5 - 5.099 = 2.401$ (différence entre p et q est de l'ordre de dizaines)

- n=194(composé de 3 chiffres ; q=97); $u \simeq \frac{p+q}{2} - \sqrt{n} = 49.5 - 13,9283882772 = 35.5716$(différence entre p et q est de l'ordre de cent)

- n=1994(composé de 4 chiffres ; $q = 997$); $u \simeq \frac{p+q}{2} - \sqrt{n} = 499.5 - 44,6542271235322 = 454.8458$ (différence entre p et q est de l'ordre de mille)

- n=19946 (composé de 5 chiffres ; $q = 9997$); $u \simeq \frac{p+q}{2} - \sqrt{n}$
  =4999.5-141,230308361909=4858.2697

- n=199982(composé de 6 chiffres ; $q = 99991$); $u \simeq \frac{p+q}{2} - \sqrt{n}$=49996.5-447.193470435 =49549.3065 etc.

Les figures ci-dessous sont obtenues à l'aide d'un logiciel gnuplot et d'un code en langage python. Nous avons fixé p à 2 et nous avons fait varier q. Comme nous le voyons, le nombre d'opérations telles que données par la relations (4.19) est une fonction de la différence entre p et q. Les figures ci-dessous sont obtenues en prenant $p = 2$ et q prenant les valeurs dans l'intervalle de 2 à 1000000 et $n = pq$ et un ajustement avec une fonction d'expression $f(x) = a.x^b$

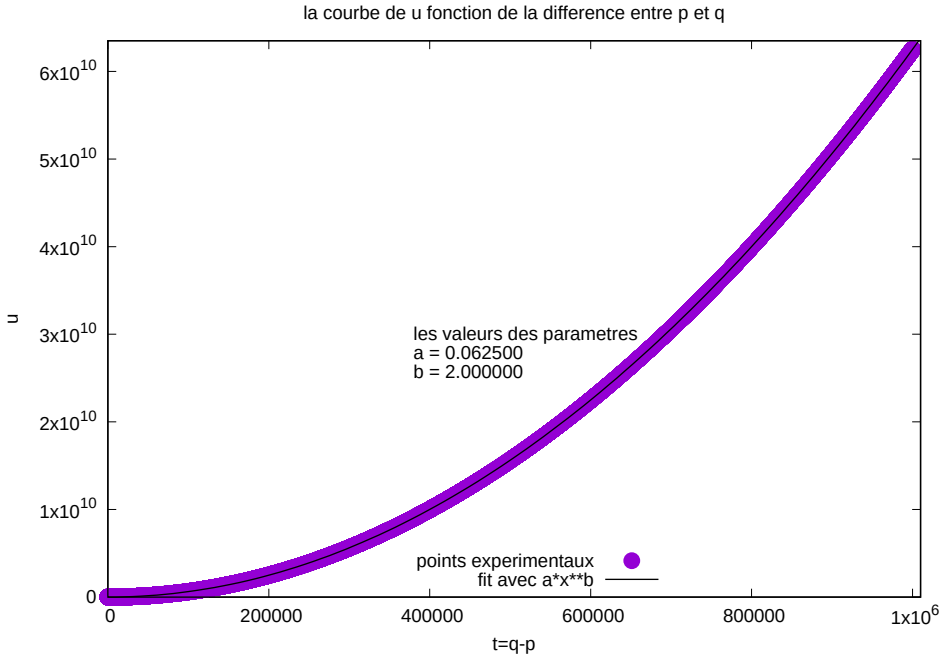

Figure 4.1: variation du nombre d'itérations en fonction de la différence entre p et q

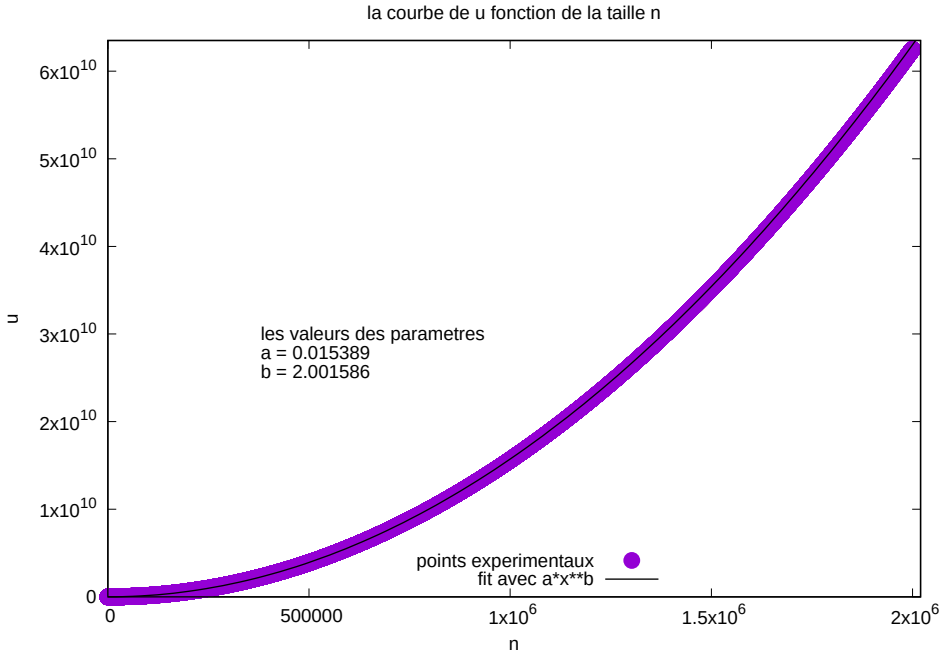

Figure 4.2: variation du nombre d'itérations en fonction de la taille n

Les figures ci-dessus sont obtenues en prenant $p = 2$ et q prenant les valeurs dans l'intervalle de 2 à 1000000. Nous voyons que pour ces deux figures, le paramètre "b" en exposant de la fonction d'ajustement $a.x^b$, qui est un des paramètres d'ajustement est égal à 2; ce qui implique que le nombre d'itérations a une allure d'une fonction quadratique linéaire.

4.3 Résolution quantique du problème de factorisation de Fermat

4.3.1 Introduction

La méthode de Grover va nous permettre de passer de l'approche classique à l'approche quantique. Pour la factorisation de Fermat, nous montrons que ce problème revient à chercher un élément dans une liste des données.

En effet, grâce à l'algorithme de Grover, qui est un algorithme de recherche, nous allons chercher un carré parmi les valeurs $([\sqrt{n}] + u)^2 - n$ (où u=1,2,...) qui se trouvent dans l'état général du système décrit par le vecteur d'état $|\psi\rangle$ (equation 3.11).

4.3.2 Présentation de l'algorithme de Grover

L'algorithme de Grover est un algorithme de recherche dans une liste. Il résout ce problème en procédant comme suit:

soit la fonction f: $\mathbb{Z} \to \mathbb{Z}/2\mathbb{Z}$ Où $\mathbb{Z}/2\mathbb{Z} = \{0; 1\}$, alors il faut chercher un argument x tel que f (x) = 1.

- La fonction f doit être associée à un circuit logique quantique pour former un **oracle**.
- Le nombre total d'éléments parmi lesquels il faut chercher la solution est noté $N = 2^n$.
- Notons également que l'algorithme de Grover ne fournit pas une solution sûre à 100%, mais une réponse qui a de grandes chances d'être la bonne.

Voici en résumé le circuit de Grover complet et nous allons expliquer étape par étape l'algorithme de Grover.

Les figures sont obtenues en considérant un oracle de type controled z (c.z) avec en entrée deux qubits.

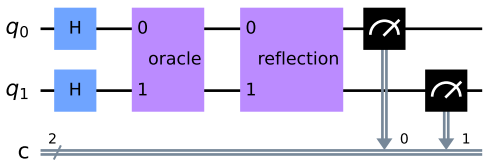

Figure 4.3: Circuit complet de l'algorithme de Grover

Les qubits au départ doivent être initialisés à l'état zéro, noté q_0. Il y a également un état spécial q_1 permettant de mesurer l'indice de la solution. L'ensemble forme un **registre**

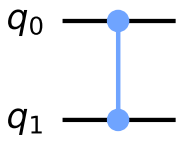

Figure 4.4: Initialisation des qubits

Maintenant, il faut préparer le registre en une superposition uniforme en appliquant la porte d'Hadamard à chaque qubit du registre. Ceci s'exprime mathématiquement par:

$|registre\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle$

Après suit l'étape d'application de l'oracle au registre préparé; ceci a pour but de changer l'amplitude de l'élément "solution" en la multipliant par -1.

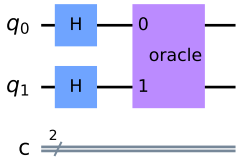

Figure 4.5: Phase Hadamard et Phase oracle

On applique encore une fois la porte d'Hadamard à chaque qubit dans le registre puis un décalage de phase conditionnel -1 à chaque état de base de calcul, sauf l'état $|0\rangle$; cela peut être représenté par l'opération unitaire O_0 car O_0 représente le décalage de phase conditionnel sur $|0\rangle$ uniquement.

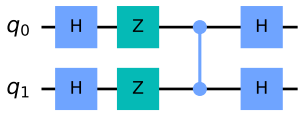

Figure 4.6: Phase oracle, application de la porte H, décalage de phase conditionnel

Enfin on mesure le registre pour obtenir l'index de l'élément "solution".

Plus généralement, lorsque nous avons en entrée plusieurs qubits nous aurons un circuit de Grover de la forme suivante, comme l'a montré Nielsen et Chuang, dans la figure ci-dessous (figure 4.7):

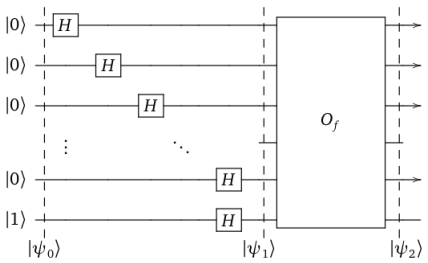

Figure 4.7: Circuit de Grover avec plusieurs qubits [Bodin, 2021]

- Initialisation. le qubit en entrée est un (n + 1)-qubit :

- $|\psi_0\rangle = |0...0\rangle.|1\rangle = |\underline{0}\rangle.|1\rangle$: nous rappelons que pour $0 \leq k \leq N-1$, $\underline{k}$ est l'écriture binaire de k sur n bits. Ainsi, $|\underline{k}\rangle$ pour $k = 0, \cdots, 2^n - 1$, désigne les n-qubits de la base canonique: $|\underline{0}\rangle = |0.0...0\rangle$, $|\underline{1}\rangle = |0.0...1\rangle$, $\cdots$,$|\underline{2^n - 1}\rangle = |1.1...1\rangle$

- Transformation d'Hadamard. On s'intéresse d'abord seulement aux n premières lignes.

Après la transformation de Hadamard (une porte de Hadamard sur chacune des n premières lignes) alors le n-qubit est:$|\underline{0}\rangle+|\underline{1}\rangle+...+|\underline{k_0}\rangle+...+|\underline{2^n-1}\rangle$ (à un facteur multiplicatif près). Ainsi tous les qubits k se retrouvent simultanément en entrée de l'oracle.

Voici les calculs complets, en intégrant tous les qubits :

$$|\psi_1\rangle = H^{\otimes n+1}|\psi_0\rangle \tag{4.20}$$

$$|\psi_1\rangle = H^{\otimes n}|0\rangle.H|1\rangle \tag{4.21}$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}|\underline{k}\rangle.\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{4.22}$$

Oracle: l'oracle fait apparaître le signe **"-"** devant le terme correspondant à $|\underline{k_0}\rangle$. Nous obtenons à la sortie de l'oracle

$$|\underline{0}\rangle + |\underline{1}\rangle + ... - |\underline{k_0}\rangle + ... + |\underline{2^n-1}\rangle \tag{4.23}$$

En détaillant et en sachant que $(-)^{f(k)} = 1$ dans la mesure où $k = k_0$, nous obtenons

$$|\psi_2\rangle = O_f|\psi_1\rangle \tag{4.24}$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}(-)^{f(k)}|\underline{k}\rangle.\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{4.25}$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}(|\underline{0}\rangle + |\underline{1}\rangle + ... - |\underline{k_0}\rangle + ... + |\underline{2^n-1}\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{4.26}$$

Notons qu'en une seule évaluation de l'oracle, on arrive à distinguer le terme de rang k_0 des autres termes grâce au phénomène de parallélisme quantique.

4.3.3 Transformation géométrique

On peut bien comprendre la transformation de Grover en utilisant une interprétation géométrique.

Premièrement, l'élément recherché noté $|k_0\rangle$ est représenté sur l'axe des y. L'axe des x représente la somme de tous les autres éléments , notés $|\chi\rangle$ à l'exception de l'élément recherché. Un vecteur noté $|\psi_H\rangle$ somme de tous les états y compris $|k_0\rangle$, fait un angle $\frac{\theta}{2}$ avec l'axe des x. Or la porte G est définie comme suit: $G = S_{\psi_H} \circ O_f$. C'est cette porte G, appelée la transformation de Grover qui nous permet de retrouver l'élément recherché.

Schématiquement, la transformation par G d'un état quelconque $|\phi\rangle$ est la transformation suivante comme le montre également Nielsen et Chuang [Nielsen and Chuang, 2010]:

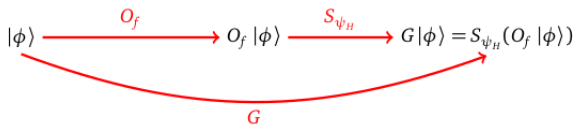

Figure 4.8: Transformation d'un état quelconque ϕ par la porte de Grover

Analysons maintenant comment la transformation G agit sur l'ensemble des qubits.

$$|\psi_H\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=o}^{2^n-1} |\underline{k}\rangle \; avec \; 2^n = N \tag{4.27}$$

Dans cette somme nous mettons à part le qubit correspondant à l'état $|k_0\rangle$, qui est l'état que l'on doit déterminer:

$$|\psi_H\rangle = \sqrt{\frac{N-1}{N}}|\chi\rangle + \frac{1}{\sqrt{N}}|\underline{k_0}\rangle \tag{4.28}$$

où $|\chi\rangle = \frac{1}{\sqrt{N-1}} \sum_{k \neq k_0} |\underline{k}\rangle$

Ecrivons maintenant $|\psi_H\rangle$ à l'aide d'une écriture trigonométrique:

$$|\psi_H\rangle = cos\left(\frac{\theta}{2}\right)|\chi\rangle + sin\left(\frac{\theta}{2}\right)|\underline{k_0}\rangle \tag{4.29}$$

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

Proposition

La transformation de Grover est une rotation d'angle θ (centrée à l'origine). Autrement dit, pour tout qubit $|\phi\rangle$, $G|\phi\rangle$ est obtenu à partir de $|\phi\rangle$, par une rotation d'angle θ. La

composition de deux symétries axiales est une rotation, l'angle θ de cette rotation étant le double de l'angle entre les axes.

Idée de l'algorithme

Le but de l'algorithme de Grover est de déterminer l'état $|k_0\rangle$. Cet état est repérable après l'application de l'oracle O_f.

Partons des equations (4.27) et (4.28) : $|\psi_H\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle = \sqrt{\frac{N-1}{N}}|\chi\rangle + \frac{1}{\sqrt{N}}|\underline{k_0}\rangle$ (nous rappelons que l'écriture d'un état souligné dans un ket désigne que cet état est écrit dans une base donnée), alors

$$O_f|\psi_H\rangle = \sqrt{\frac{N-1}{N}}|\chi\rangle - \frac{1}{\sqrt{N}}|\underline{k_0}\rangle \tag{4.30}$$

L'oracle marque un état "solution" en inversant la phase, c'est à dire en multipliant par -1 le coefficient de l'état qui est l'état solution.

Si par exemple nous avons un registre(état général) $|\psi\rangle = \alpha_0|0...0\rangle + \alpha_1|0...01\rangle + ... + \alpha_{N-1}|1...10\rangle + \alpha_N|1...1\rangle$ telle que la solution soit l'état $|1...1\rangle$ alors l'application de l'oracle sur le registre $|\psi\rangle$ va modifier le coefficient α_N de l'état solution qui deviendra $-\alpha_N$, et nous obtenons donc:

$O_f|\psi\rangle = \alpha_0|0...0\rangle + \alpha_1|0...01\rangle + ... + \alpha_{N-1}|1...10\rangle - \alpha_N|1...1\rangle$

Ce marquage est effectué par l'intrication entre le qubit $|out\rangle$ initialisé avec les portes X puis H et $|\psi\rangle$ tel que $|out\rangle$ est inversé si et seulement si l'état de l'espace de recherche correspond au critère de l'oracle. En d'autre terme, $|out\rangle$ sera dans l'état $|1\rangle$ si le registre $|\psi\rangle$ est dans un état solution. Cela va avoir pour effet d'inverser la phase de l'état solution dans $|\psi\rangle$ et de perturber la moyenne des α_i (en pointillé sur l'histogramme) par ce changement de phase.

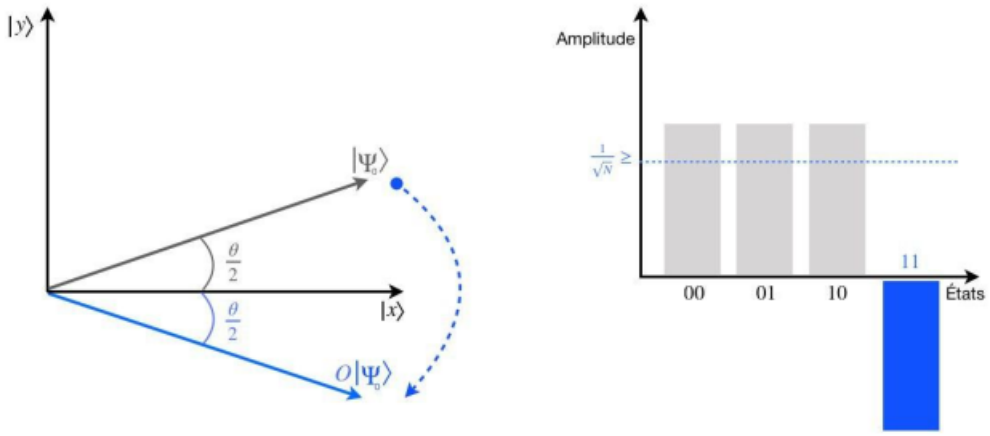

Figure 4.9: Représentation de l'application de l'oracle [Rodriguez, 2021]

Amplification d'amplitude: l'opération d'amplification d'amplitude va permettre d'amplifier la phase des états cibles. Comme les états solutions sont marqués par l'oracle, cette opération va cibler les états solutions et augmenter leurs amplitudes tout en réduisant celles des autres états.

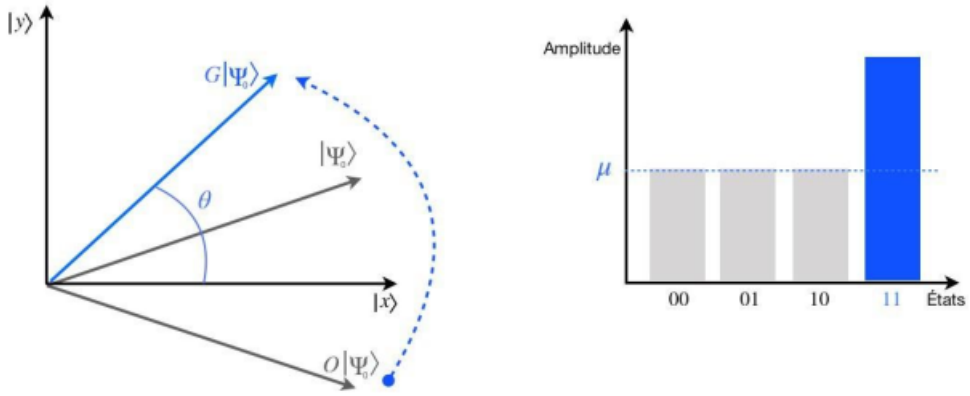

Figure 4.10: Amplification d'amplitude [Rodriguez, 2021]

Sur cette figure, nous observons l'état $|\psi\rangle$ inversé par l'oracle puis amplifié par l'opération d'amplification d'amplitude.

$$A = 2|\psi\rangle\langle\psi| \tag{4.31}$$

Cette opération rééquilibre les valeurs des états. Une itération fait gagner un angle θ à l'unique état solution de l'oracle. Nous allons réitérer l'opération

$$G = OA|\psi\rangle. \tag{4.32}$$

Après "l" étapes, l'état $|\psi\rangle$ devient:

$$|\psi_l\rangle = G^l|\psi\rangle \tag{4.33}$$

et cet état va se rapprocher de l'état solution $|y\rangle$

Détermination du nombre d'itérations

Nous avons vu que $|\psi_H\rangle = cos(\frac{\theta}{2})|\chi\rangle + sin(\frac{\theta}{2})|\underline{k_0}\rangle$

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

Par identification,

$$cos\left(\frac{\theta}{2}\right)|\chi\rangle + sin\left(\frac{\theta}{2}\right)|\underline{k_0}\rangle = \sqrt{\frac{N-1}{N}}|\chi\rangle + \frac{1}{\sqrt{N}}|\underline{k_0}\rangle \tag{4.34}$$

$$cos\left(\frac{\theta}{2}\right)|\chi\rangle = \sqrt{\frac{N-1}{N}}|\chi\rangle \tag{4.35}$$

$$cos\left(\frac{\theta}{2}\right) = \sqrt{\frac{N-1}{N}} \tag{4.36}$$

$$sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}} \tag{4.37}$$

Dans la pratique, $\frac{\theta}{2}$ est très petit, ce qui fait que $|\psi_H\rangle$ est presque confondu avec l'axe $|\chi\rangle$. Par conséquent,

$$sin\left(\frac{\theta}{2}\right) \simeq \frac{\theta}{2} = \frac{1}{\sqrt{N}}(approximation\ de\ petits\ angles) \tag{4.38}$$

$$\theta = \frac{2}{\sqrt{N}} \tag{4.39}$$

Nous déterminons le nombre d'itérations en appliquant la transformation de Grover pour arriver à former un angle de $\frac{\pi}{2}$ avec l'axe $|\chi\rangle$.

Nous avons vu que $G = S\psi_H \circ O_f$ =rotation d'angle θ, ce qui conduit à la formation d'un qubit $G|\psi_H\rangle$ qui forme un angle de $\frac{\theta}{2} + \theta$ avec l'axe $|\chi\rangle$ Nous souhaitons former un angle de $\frac{\pi}{2}$ avec l'axe $|\chi\rangle$, pour cela il faut appliquer $G^l = l\theta$. Le qubit $G^l|\psi_H\rangle$ forme un angle de $\frac{\theta}{2} + l\theta$ avec l'axe $|\chi\rangle$. Il faut donc que $\frac{\theta}{2} + l\theta$ soit environ égal à $\frac{\pi}{2}$

$$\frac{\theta}{2} + l\theta = \frac{\pi}{2} \Rightarrow l = \frac{\pi}{2\theta} - \frac{1}{2} \simeq \frac{\pi}{2\theta} \tag{4.40}$$

Avec $\theta = \frac{2}{\sqrt{N}}$, on obtient $l = \frac{\pi\sqrt{N}}{4}$

Nous voyons que la complexité de l'algorithme de Grover est $O\sqrt{N}$.

4.3.4 Présentation du problème de Fermat sous forme de l'algorithme de Grover

Soit un entier n à factoriser. Définissons les différents vecteurs d'états $|\varphi_1\rangle$, $|\varphi_2\rangle$,.. $|\varphi_p\rangle$ dans lesquels se trouvent les valeurs (où p est le nombre des valeurs trouvées en calculant $([\sqrt{n}] + u)^2 - n)$. Pour cela on a des espaces vectoriels de Hilbert de ces vecteurs d'états de dimensions d_i. Désignons par C_i les valeurs trouvées avec i=1,2,..p

$$|\varphi_1\rangle^{\otimes d_1} = C_1 \tag{4.41}$$

$$|\varphi_2\rangle^{\otimes d_2} = C_2 \tag{4.42}$$

$$\vdots \tag{4.43}$$

$$|\varphi_p\rangle^{\otimes d_m} = C_p \tag{4.44}$$

En utilisant le produit tensoriel entre deux vecteurs, définissons:

$|\psi\rangle = |\psi\rangle^{\otimes d_1 d_2 ... d_m} = |\varphi_1\rangle^{\otimes d_1} \otimes |\varphi_2\rangle^{\otimes d_2} ... \otimes |\varphi_k\rangle^{\otimes d_m}$ dans lequel se trouvent toutes les valeurs des états $|\varphi_i\rangle; i = 1, p.$

Toutes les valeurs sont donc dans un état $|\psi\rangle$ grâce au produit tensoriel. Le vecteur d'état général est alors appelé "espace de recherche".

Le carré cherché est identifié grâce à la fonction O_f: $|\psi\rangle \rightarrow (-1)^{f(\psi)}|\psi\rangle$. La fonction O_f consiste à laisser fixes les **non** carrés et changer le signe de l'amplitude (de la phase) du **bon** carré [Kachigar, 2016].

4.3.5 Exemple pour factoriser 15 en utilisant l'approche quantique de Fermat

Nous cherchons l'élément dans une liste des valeurs $([\sqrt{n}] + u)^2 - n$ qui soit un carré. $\Rightarrow ([\sqrt{15}] + 1)^2 - 15 = 1$

"1"est donc un carré.

Nous choisissons l'espace de recherche dans lequel on trouve le carre "1". Après l'initialisation et l'application de la porte d'Hadamard, nous obtenons:

$|\psi\rangle = \frac{1}{\sqrt{2^2}}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$

Comme la somme des carrés des coefficients doit être égale à 1, les coefficients $\frac{1}{\sqrt{N}} = \frac{1}{\sqrt{2^2}} = \frac{1}{2}$

Commençons par écrire dans notre circuit l'initialisation des états $|\psi\rangle$ et $|out\rangle$:

L'oracle doit marquer la solution, ici $|01\rangle$(écriture de 1 en binaire) en inversant son amplitude $\frac{1}{\sqrt{2}}$. Le circuit va effectuer cette opération avec la porte **SWAP**. Les qubits de contrôles seront ceux du registre de l'espace de recherche $|\psi\rangle$ et le qubit cible sera $|out\rangle$.

Si les qubits de contrôle sont dans l'état $|0\rangle$ qui correspond à l'état $|01\rangle$ de $|\psi\rangle$ alors, il faut cherche une porte logique jouant sur le bit de contrôle, pour pouvoir inverser l'état $|out\rangle$. Pour cela il faut utiliser la porte **SWAP**.

La porte SWAP est en fait une triple application de la porte $c - NOT$, en alternant la place du bit de contrôle. On appellera $c - NOT_1$ la porte $c - NOT$ avec premier qubit en bit de contrôle, et $c - NOT_2$ la porte $c - NOT$ en prenant l'autre qubit en contrôle.

L'application de l'opérateur $c - NOT$ à un état à deux qubits quelconque $|\psi\rangle$ donne:

$C - NOT|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \sigma|11\rangle \rightarrow \alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle + \sigma|10\rangle.$

L'application de l'opérateur SWAP à un état quelconque $|\psi\rangle$ donne :

$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \sigma|11\rangle \rightarrow \alpha|00\rangle + \beta|10\rangle + \gamma|01\rangle + \sigma|11\rangle.$

Pour notre cas, tous les états ont même amplitude car on a appliqué à l'espace de recherche $|\psi\rangle$ la porte d'Hadamard. Nous obtenons après l'application de la porte SWAP un état $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle + \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|11\rangle$

Après l'application de l'oracle, nous ajoutons l'opération d'amplification définie par:

$A = 2|\psi\rangle\langle\psi| - I = |\psi\rangle(|\psi\rangle^{\star})^{T} - I$

Remarque: en appliquant la porte SWAP à $|\psi\rangle$, nous trouvons deux états $|10\rangle$ et $|01\rangle$ qui ont été inversés. Il faut alors voir l'état qui correspond à un carré. Cet état est $|01\rangle$.

Voici quelques résultats de simulation en essayant d'évaluer deux états avec un oracle associé à la fonction. L'oracle SWAP doit évaluer deux états: l'état $|01\rangle$ et $|10\rangle$ avec l'état voulu qui est $|01\rangle$ et qui correspond à 1 (écriture de 1 en binaire). Les autres états $|00\rangle$, $|11\rangle$ de l'espace de recherche $|\psi\rangle$ restent inchangés. Le code a été fait à l'aide du langage de programmation "python". Nous devons spécifier l'état recherché et le programme le mentionnera dans les résultats. Les étapes de l'initialisation, de l'application de la porte d'Hadamard et de l'amplification ont été également prises en compte.

Après avoir tourné le code nous obtenons les résultats suivants:

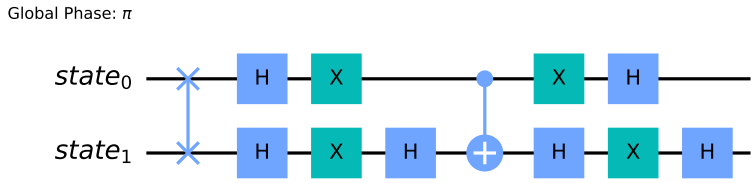

Figure 4.11: Etape de l'application de la porte d'Hadamard, renversement et amplification de la phase

Le programme mentionnera:

– évaluation par l'oracle a réussi

– état recherché est: $|01\rangle$

4.3.6 Exemple pour factoriser 21 en utilisant l'approche quantique de Fermat

A l'aide de la librairie qiskit avec un programme tourné en python, nous avons simulé l'algorithme de Fermat, en cherchant un carré premièrement à l'aide d'un programme non quantique (qui génère d'abord les valeurs $([\sqrt{n}]+u)^2-n)$, puis identifie parmi les valeurs trouvées, une valeur qui correspond à un carré. Notons que la condition d'arrêt pour la génération des nombres $([\sqrt{n}]+u)^2-n)$ est fixée à $\sqrt{n}$ pour ne pas produire plus de carrés inutiles.

Ici commence l'approche quantique: l'oracle évalue ensuite l'indice du carré trouvé et donne la valeur de cet indice. On remarquera également que l'indice trouvé n'est rien d'autre que le nombre d'itérations dans le cas classique.

Le nombre n est factorisé, en utilisant le carré identifié.

Voici les résultats de simulation:

- génération des nombres $([\sqrt{n}] + u)^2 - n)$ avec un programme non quantique: nous avons obtenu la liste suivante: [4, 15, 28, 43];
- les carrés trouvés parmi [4, 15, 28, 43]$\Rightarrow$ 4;
- la factorisation de 21 : $21 = 3 \times 7$;
- nombres d'itérations =1.

En pratique, l'approche quantique de Fermat peut se réaliser en deux temps:

- calculer les valeurs $([\sqrt{n}] + u)^2 - n$ en utilisant un programme non quantique;
- comparer les valeurs $([\sqrt{n}] + u)^2 - n = S^2$ en utilisant l'algorithme de Grover.

L'algorithme de Grover n'est pas un algorithme efficace de factorisation comme l'algorithme de Shor [Shor, 1994], mais a été choisi pour montrer qu'on peut passer d'une complexité exponentielle classique à une complexité polynomiale quantique.

4.4 Application

Chiffrement RSA

Le système RSA, du nom de ses auteurs Rivest, Shamir, Adleman, a été proposé en 1978. Il utilise une fonction à sens unique avec clés privées et secrètes permettant l'inversion de la fonction par celui connaissant ces clés privées. Il est basé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers. Nous allons partir d'un exemple pour découvrir comment coder et décoder un message en utilisant le système RSA.

Génération des clés

- Bob choisit deux nombres premiers p et q. Il doit garder ces nombres secrets.
- Bob calcule $n = p * q$ et $\varphi(n) = (p-1)(q-1)$
- Bob choisit un nombre entier e premier avec $\varphi(n)$

- Le nombre d est calculé selon la formule suivante: $e*d+\varphi(n)*x=1$ (formule pour calculer les coefficients de Bézout)

Distribution des clés

Le couple (n, e) constitue la clé publique de Bob. Il la rend disponible à Alice en lui envoyant ou en la mettant dans un annuaire. Le couple (n, d) constitue quand à lui sa clé privée.

Cette clé publique étant nécessaire pour le chiffrement, elle doit être disponible pour toute personne ayant un message à crypter pour Bob. Tout en étant un élément de la clé publique de Bob, e peut aussi figurer dans la clé publique de n'importe qui d'autre. Toutefois chacun doit attribuer une valeur différente à n, qui dépend du choix de p et q.

Chiffrement du message

Pour crypter le message Alice représente le message sous la forme d'un ou plusieurs entiers M compris entre 0 et n-1. Elle calcule $C \equiv M^e(modulo\ n)$ grâce à la clé publique (n, e) de Bob et envoie C à Bob.

Déchiffrement du message

Bob reçoit C. Il calcule le nombre d à l'aide des coefficients de Bézout et trouve ainsi d. La clé de déchiffrement, encore appelée clé privée est ainsi (d,n). Il déchiffre le message comme suit.

$M \equiv C^d(modulo\ n)$

Nous constatons que le chiffrement RSA repose sur le problème de factorisation d'un nombre, produit de deux nombres premiers. Une des méthodes pour factoriser un nombre composé donné n est la méthode de Fermat comme nous l'avons fait.

Prenons l'exemple de $n = 11413 = 101 \times 113$.

Nous choisissons $p = 101$ et $q = 113$. Nous calculons une fonction $\varphi(n) = (p-1)(q-1)$. Ceci donne $100 \times 112 = 11200$. Notre fonction $\varphi(n) = 11200$. Notons que cette fonction nous donne le nombre des nombres qui sont premiers avec n

Nous choisissons le nombre e dans l'intervalle]2;11200[pour constituer la clé publique. Pour chaque nombre se trouvant dans l'intervalle]2;11200[, nous pouvons chercher un nombre qui soit premier avec $\varphi(n)$. Prenons $e = 3533$ car le plus grand commun diviseur (PGCD) entre 3533 et 11200 est égale à 1.

Notre clé publique est donc (11413;3533). Pour trouver la clé privée, nous commençons à calculer les coefficients de Bézout en utilisant la formule d'Euclide où x et d sont des inconnues.

$$e \times d + x \times \varphi(n) = 1 \ \Rightarrow 3533 \times d + 11200 \times x = 1 \tag{4.45}$$

Le nombre d est donc égal 6597 et la clé privée est donc constituée par $(d; n)$, soit donc (6597;11413).

Utilisons la clé publique pour chiffrer un message M constituée par les chiffres qui suivent.$M = 9726$.

Le message chiffré est donc $C = M^e(modulo\ n)$.

On a $C = (9726)^{3533}(modulo\ 11413) = 5761$.

La valeur du message chiffré est 5761.

Pour déchiffrer le message c'est-à-dire retrouver M, nous utilisons la formule suivante:

$M = C^d(modulo\ n)$. On a $M = (5761)^{6597}(modulo\ 11413) = 9726$ et nous retrouvons bien évidemment $M = 9726$.

Conclusion La sécurité de l'information repose sur les clés. Le cas de la factorisation des entiers en deux nombres premiers dans le but de trouver les clés publique et privée en utilisant la méthode de Fermat a une complexité exponentielle. Cette méthode ne donnerait pas des résultats en un temps raisonnable pour la factorisation des entiers composés de plusieurs chiffres. Nous pourrions recourir à l'ordinateur quantique s'il était déjà mis en place et retrouver rapidement la clé privée pour déchiffrer le message et casser RSA.

Conclusion générale

Dans ce travail, nous avons comparé la complexité d'un même algorithme en utilisant deux approches: l'une quantique et l'autre classique. Nous sommes partis de la factorisation d'un entier en deux nombres premiers p et q avec la méthode de Fermat. Dans le cas classique, la complexité augmente rapidement avec la taille du nombre à factoriser comme nous l'avons démontré en donnant d'abord une expression analytique du nombre d'itérations, puis en faisant un code dans un langage en python et en utilisant un logiciel gnuplot pour tracer une courbe. Nous avons utilisé un ajustement avec une fonction d'expression $f(x) = a.x^b$ et nous avons trouvé une courbe du nombre d'itérations (en fonction de la différence entre p et q impliquant une augmentation de la taille du nombre n à factoriser). Nous avons conclu que la complexité de la factorisation avec la méthode de Fermat augmente rapidement avec la taille du nombre à factoriser.

En utilisant l'approche quantique de Fermat, l'algorithme de Grover dont la complexité dans la littérature est polynomiale, nous avons refait les calculs pour avoir une expression de la complexité de l'algorithme de Grover. Par la suite, nous avons formulé le problème de la factorisation avec la méthode de Fermat sous la forme d'un problème de recherche d'un carré parmi les valeurs $([\sqrt{n}] + u)^2 - n$. Cette liste des valeurs est considérée dans le cas quantique comme un ensemble de valeurs se trouvant dans un vecteur d'état général défini par le produit tensoriel de chacune des valeurs de l'expression $([\sqrt{n}] + u)^2 - n$, jusqu'à un nombre $l = [\sqrt{n}]$; où $u = 1, 2, 3, ...$ Cela à été simulé a l'aide d'un code en python et d'un logiciel qiskit permettant de tracer des circuits quantiques.

Si nous avions un ordinateur quantique, les phénomènes de superposition et du parallélisme quantiques auraient été mis en évidence. Comme cela, on passerait de la complexité qui augmente rapidement avec la taille du nombre à factoriser dans le cas classique à la complexité polynomiale quantique, ce qui affirme une accélération spectaculaire de la factorisation avec la méthode de Fermat. On passe ainsi de la complexité qui augmente rapidement avec la taille du nombre à factoriser à la complexité polynomiale pour un même algorithme.

Le système RSA est basé sur la difficulté de trouver les nombres premiers p et q qui factorisent un très grand nombre n (composé de plusieurs chiffres). Comme nous l'avons vu, ces deux nombres sont très importants notamment dans la constitution des clés publique et privée. Or avec l'approche quantique de la factorisation avec la méthode de Fermat,

trouver les facteurs p et q n'est plus un problème qui nécessite un temps irraisonnable (plusieurs années). Nous pouvons donc casser RSA en découvrant rapidement les nombres p et q, qui sont des paramètres importants dans la constitution des clés publique et privée.

Perspectives et recommandations

L'ordinateur quantique reste idéal, cela fait que notre travail reste théorique. Nous recommandons donc aux futurs chercheurs de se lancer rapidement dans les projets de création des calculateurs quantiques.

Notre travail a porté sur un des problèmes de la classe NP-complet et n'est donc pas exhaustif. Néanmoins, les futurs chercheurs pourront travailler sur un autre cas de problème NP-complet et passer du classique au quantique en utilisant un même algorithme. Nous suggérons aux futurs chercheurs de tester un autre problème NP-complet diffèrent de celui que nous avons testé.

Bibliographie

[Akama, 2015] Akama, S. (2015). *Elements of quantum computing*. Springer.

[Aumeunier, 1977] Aumeunier, R. (1977). Cryptographie asymétrique: Description d'un nouveau cryptogramme.

[Benioff, 1981] Benioff, P. (1981). Quantum mechanical hamiltonian models of discrete processes. *Journal of Mathematical Physics*, 22(3):495–507.

[Bernstein and Vazirani, 1993] Bernstein, E. and Vazirani, U. (1993). Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 11–20.

[Berthiaume, 1995] Berthiaume, A. (1995). *L'ordinateur quantique: complexité et stabilisation des calculs*. Université de Montréal.

[Berthiaume and Brassard, 1992] Berthiaume, A. and Brassard, G. (1992). The quantum challenge to structural complexity theory. In *Computational Complexity Conference*, pages 132–137. Citeseer.

[Berthiaume and Brassard, 1994a] Berthiaume, A. and Brassard, G. (1994a). Oracle quantum computing. *Journal of modern optics*, 41(12):2521–2535.

[Berthiaume and Brassard, 1994b] Berthiaume, A. and Brassard, G. (1994b). Oracle quantum computing. *Journal of modern optics*, 41(12):2521–2535.

[Bodin, 2021] Bodin, A. (2021). *Un peu de mathématiques pour l'informatique quantique*.

[Boneh and Durfee, 2000] Boneh, D. and Durfee, G. (2000). Cryptanalysis of rsa with private key d less than n/sup 0.292. *IEEE transactions on Information Theory*, 46(4):1339–1349.

[Bouchareb and Laboudi, 2012] Bouchareb, A. and Laboudi, Z. (2012). Implémentation des algorithmes.

[Boyer-Kassem, 2015] Boyer-Kassem, T. (2015). Qu'est-ce que la mécanique quantique?

[Brune and Raimond, 2005] Brune, M. and Raimond, J.-M. (2005). L'ordinateur quantique: un défi pour les expérimentateurs. *Images de la physique*, pages 111–117.

[Bruno et al., 2016] Bruno, M.-A., Nizzi, M.-C., Laureys, S., and Gosseries, O. (2016). Consciousness in the locked-in syndrome. In *The Neurology of Conciousness*, pages 187–202. Elsevier.

[Carmi, 1977] Carmi, G. (1977). *Des Templiers aux Massenies du Saint-Graal*. FeniXX.

[Cohen-Tannoudji et al., 2020] Cohen-Tannoudji, C., Diu, B., and Laloë, F. (2020). Mécanique quantique-tome 1: Nouvelle édition.

[Courtois, 2019] Courtois, N. T. (2019). Si seulement enigma tournait moins vite, ou cryptanalyse d'enigma avec un rotor faible. *Bulletin de l'ARCSI n*, 46:79.

[Couteau, 2005] Couteau, C. (2005). *Vers une source de photons uniques indiscernables a l'aide de boites quantiques semiconductrices II-VI.* PhD thesis, Université Paris Sud-Paris XI.

[Dan C. Marinescu, 2005] Dan C. Marinescu, G. M. M. (2005). *approaching quantum computing.*

[Deutsch, 1985a] Deutsch, D. (1985a). Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117.

[Deutsch, 1985b] Deutsch, D. (1985b). Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117.

[Deutsch and Jozsa, 1992] Deutsch, D. and Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558.

[Engo, 2020] Engo, N. (2020). Optique quantique.

[Ferro et al., 2005] Ferro, L., KHIN, S., and SALMAN, N. (2005). Résolution pratique de problèmes np-complets. Technical report, Technical report, 26 juin.

[Feynman and IOP, 2016] Feynman, R. and IOP, C. (2016). Information quantique.

[Fournaise, 2022] Fournaise, N. (2022). *Protocoles cryptographiques pour le respect de la vie privée.* PhD thesis, Limoges.

[Gimblett, 2009] Gimblett, R. H. (2009). *Le Service naval du Canada, 1910-2010: Cent ans d'histoire.* Dundurn.

[Giraud et al., 2019] Giraud, M., Lafourcade, P., and Ordinateur, G. I. S. (2019). Mission cryptographie. *Ressi*, 19:15.

[harizaka Chrystelle, 2019] harizaka Chrystelle, T. (2019). *Etudes du protocole BB84 en cryptographie quantique, mémoire en vue de l'obtention du Diplôme de Master*.

[Jaffali, 2020] Jaffali, H. (2020). *Étude de l'Intrication dans les Algorithmes Quantiques: Approche Géométrique et Outils Dérivés*. PhD thesis, Bourgogne Franche-Comté.

[Jauvart, 2017] Jauvart, D. (2017). *Sécurisation des algorithmes de couplages contre les attaques physiques*. PhD thesis, Université Paris-Saclay.

[Kachigar, 2016] Kachigar, G. (2016). *Étude et conception d'algorithmes quantiques pour le décodage de codes linéaires*. PhD thesis, Université de Rennes 1, France.

[Knuth, 1976] Knuth, D. E. (1976). Big omicron and big omega and big theta. *ACM Sigact News*, 8(2):18–24.

[Koppenhöfer et al., 2020] Koppenhöfer, M., Bruder, C., and Roulet, A. (2020). Quantum synchronization on the ibm q system. *Physical Review Research*, 2(2):023026.

[Kortchinsky, 2004] Kortchinsky, K. (2004). Cryptographie et reverse-engineering en environnement win32. In *Actes de la conférence SSTIC*, volume 2004, pages 129–144.

[Lamaute et al., 2016] Lamaute, N., Piccoli, A., Chen, L.-C., and Cotoranu, A. (2016). A substitution cipher for musical cryptography. *Proceedings of Student-Faculty Research Day, CSIS, Pace University*.

[Manullang et al., 2020] Manullang, A. S., Puspasari, R., and Verina, W. (2020). Penyandian database menggunakan metode base64 dan rot13. *Jurnal Mahasiswa Fakultas Teknik dan Ilmu Komputer*, 1(1):283–292.

[Marchildon, 2000] Marchildon, L. (2000). *Mécanique quantique*. De Boeck Supérieur.

[Nielsen and Chuang, 2010] Nielsen, M. A. and Chuang, I. L. (2010). Quantum computation and quantum information. *Quantum Computation and Quantum Information*.

[Perrin, 2018] Perrin, D. (2018). Fermat, mersenne, factorisation et nombres parfaits.

[Rodriguez, 2021] Rodriguez, J. (2021). Évaluation du potentiel des machines quantiques pour l'optimisation combinatoire. In *22 ème conférence ROADEF de la Société Française de Recherche Opérationnelle et Aide à la descision*.

[Rotella, 2018] Rotella, Y. (2018). *Mathématiques discrètes appliquées à la cryptographie symétrique*. PhD thesis, Sorbonne université.

[Shor, 1994] Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee.

[Shor, 2003] Shor, P. W. (2003). Why haven't more quantum algorithms been found? *Journal of the ACM (JACM)*, 50(1):87–90.

[Singh and Singh, 2016] Singh, J. and Singh, M. (2016). Evolution in quantum computing. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pages 267–270. IEEE.

[Soto and Bassham, 2000] Soto, J. and Bassham, L. (2000). Randomness testing of the advanced encryption standard finalist candidates. Technical report, Booz-Allen And Hamilton Inc Mclean Va.

[Standard et al., 1999] Standard, D. E. et al. (1999). Data encryption standard. *Federal Information Processing Standards Publication*, 112.

[Steffen et al., 2011] Steffen, M., DiVincenzo, D. P., Chow, J. M., Theis, T. N., and Ketchen, M. B. (2011). Quantum computing: An ibm perspective. *IBM Journal of Research and Development*, 55(5):13–1.

[Stern, 1998] Stern, J. (1998). *Science du secret (La)*. Odile Jacob.

[Telescope, 2018] Telescope, E. (2018). Europe reveals its astroparticle vision. *Physics World*, page 11.

[Toubakh, 2020] Toubakh, R. (2020). Réalisation d'une méthode méta heuristique pour la résolution d'un problème np-complet.

[Turing et al., 1995] Turing, A. M., Girard, J.-Y., Basch, J., and Blanchard, P. (1995). *La machine de Turing*. Editions du seuil.

[Unguru, 1976] Unguru, S. (1976). Fermat revivified, explained and regained. *Francia*, 4:774–789.

[Vaudenay, 1995] Vaudenay, S. (1995). *La sécurité des primitives cryptographiques*. PhD thesis, ANRT [diff.].

[Wille et al., 2019a] Wille, R., Van Meter, R., and Naveh, Y. (2019a). Ibm's qiskit tool chain: Working with and developing for real quantum computers. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1234–1240. IEEE.

[Wille et al., 2019b] Wille, R., Van Meter, R., and Naveh, Y. (2019b). Ibm's qiskit tool chain: Working with and developing for real quantum computers. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1234–1240. IEEE.

[Yefsah and Sayrin, 2022] Yefsah, T. and Sayrin, C. (2022). Simulation quantique avec des atomes froids. comment manipuler et sonder des systèmes quantiques à l'échelle de l'atome individuel. *Reflets de la physique*, (71):8–15.