
Applications des ordinateurs quantiques à la dynamique de systèmes quantiques ouverts

Auteur : Frédéric, Bastien

Promoteur(s) : Damanet, François

Faculté : Faculté des Sciences

Diplôme : Master en sciences physiques, à finalité approfondie

Année académique : 2024-2025

URI/URL : <http://hdl.handle.net/2268.2/22286>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.



FACULTÉ DES SCIENCES
DÉPARTEMENT DE PHYSIQUE

Applications des ordinateurs quantiques à la dynamique de systèmes quantiques ouverts

Auteur : FRÉDÉRIC Bastien
Promoteur : DAMANET François

TRAVAIL DE FIN D'ÉTUDES PRÉSENTÉ DANS LE CADRE DE L'OBTENTION DU GRADE DE
Master en Sciences Physiques

ANNÉE ACADÉMIQUE 2024-2025

Remerciements

En premier lieu, je remercie mon promoteur de mémoire, Pr. F. Damanet, pour son encadrement, son temps, ses conseils précieux et sa disponibilité tout au long de ce travail. Votre soutien et votre expertise ont été essentiels à la réussite de ce projet.

Je tiens à adresser mes sincères remerciements au Pr. P. Schlagheck, au Pr. N. Vandewalle et au Pr. Ph. Ghosez pour leur participation à mon comité de lecture. Je leur souhaite une excellente lecture.

Je souhaite également remercier l'ensemble des enseignants et intervenants qui m'ont transmis des connaissances et des compétences impressionnantes tout au long de ma formation.

Enfin, je remercie ma famille et mes amis pour leur patience, leur encouragement et leur présence tout au long de cette aventure. Votre soutien m'a donné la motivation nécessaire pour mener ce projet à terme.

Table des matières

1	Introduction	4
I	Information et ordinateurs quantiques	6
2	L'évolution du domaine de l'information quantique	6
3	Les états d'un système quantique	7
3.1	La sphère de Bloch	8
4	Les circuits quantiques	9
4.1	Les portes quantiques	9
4.2	L'équivalence des circuits quantiques	11
5	Les qubits supraconducteurs	12
6	IBM et ses contributions aux ordinateurs quantiques	13
7	Le bruit quantique	15
7.1	Les sources de bruit quantique	15
7.2	Les moyens de corrections des erreurs	16
II	Dynamique des systèmes quantiques ouverts	17
8	Postulats de la mécanique quantique	18
9	Opérateur densité	20
10	Reformulation des postulats avec l'opérateur densité	22
11	L'équation maîtresse	23
11.1	Approximations	23
11.2	Équation de Lindblad	24
11.3	Isomorphisme de Choi-Jamiolkowski	25
11.4	Représentation de Kraus	26
11.4.1	Opérateurs de Kraus	26
11.4.2	Cas discrétisé	27
11.4.3	Cas continu	27
11.5	Exemples	29
11.5.1	Canal bit-flip	30
11.5.2	Canal d'amortissement d'amplitude modifié	30
11.5.3	Canal à deux qubits	32
III	Dynamique des systèmes quantiques ouverts sur un ordinateur quantique	34
12	Algorithme basé sur le théorème de Nagy	34
12.1	Explication de l'algorithme [1]	35
12.2	Analyse de complexité	36

13 Utilisation de l'algorithme	37
13.1 Obtention des circuits quantiques	37
13.1.1 Protocole I	37
13.1.2 Protocole II	39
13.2 Utilisation des circuits	41
14 Préparation des états	42
14.1 État initial des qubits	42
14.2 Préparation des états superposés	42
14.3 Préparation des états intriqués	42
14.4 Contrôle des états avec des portes de rotation	43
15 Applications	43
15.1 Canal bit-flip	43
15.2 Canal d'amortissement d'amplitude modifié	46
15.3 Canal à deux qubits	47
16 Conclusion	50
A Annexe	55

1 Introduction

L'évolution temporelle des systèmes quantiques est un sujet traité depuis les années 1920 avec l'équation de Schrödinger. Cependant, cette équation ne décrit, dans sa forme la plus simple, que la dynamique de systèmes quantiques fermés, c'est-à-dire sans interaction avec le monde extérieur. La dynamique quantique ouverte est un domaine clé de la physique quantique qui examine l'évolution temporelle d'un système en interaction avec son environnement, ce qui est toujours le cas en pratique. De nombreux phénomènes physiques importants tels que le transport d'excitons [2], la conductivité moléculaire [3], la catalyse chimique [4] et les transitions de phase quantiques [5, 6] nécessitent l'étude de l'évolution temporelle d'un état électronique. Comme l'environnement est généralement trop complexe pour être traité de manière exacte, la dynamique quantique ouverte repose souvent sur des approximations qui consistent à moyenniser l'effet de l'environnement sur le système. Néanmoins, malgré ces approximations, la dynamique des systèmes quantiques ouverts reste, de manière générale, difficile à déterminer.

Avec l'avènement de l'information quantique, qui promet de révolutionner plusieurs domaines scientifiques comme la chimie [7, 8], les matériaux [9, 10, 11] ou l'intelligence artificielle [12], on peut légitimement espérer que la dynamique des systèmes quantiques ouverts pourrait être déterminée efficacement avec des ordinateurs quantiques, notamment avec l'algorithme proposé par Z. Hu et al. [1]. Par exemple, au niveau des interactions moléculaires à l'échelle quantique, les scientifiques pourraient accélérer le processus de découverte de médicaments et ainsi trouver des traitements pour des maladies actuellement incurables [13, 14]. De plus, les ordinateurs quantiques pourraient contribuer à trouver des alternatives plus efficaces et respectueuses de l'environnement aux réactions chimiques traditionnelles, qui sont souvent très énergivores [15].

Cependant, les algorithmes quantiques reposent sur l'utilisation de transformations unitaires. Or, l'évolution temporelle de l'état d'un système quantique ouvert est souvent non-unitaire, traduisant le caractère irréversible de l'évolution du système, ce qui pose souvent certains challenges. Nous verrons en effet que cette évolution peut être modélisée grâce à des opérateurs de Kraus, qui sont des opérateurs non-unitaires. Les opérateurs de Kraus sont des outils mathématiques qui permettent de représenter l'évolution d'un système quantique ouvert en tenant compte l'interaction avec son environnement.

Le but de ce travail est de reprendre un algorithme déjà présent dans la littérature [1] afin de l'expliquer en détails et de l'appliquer sur des exemples de systèmes quantiques ouverts. Cet algorithme permet de surpasser la non-unitarité des opérateurs de Kraus. Nous utiliserons donc des ordinateurs quantiques en temps réel sur le matériel d'IBM Quantum [16].

La méthode de l'algorithme consiste en effet à utiliser le théorème de Nagy [17] afin de transformer les opérateurs de Kraus du système en matrice de dilatation. Les matrices de dilatation fournissent une représentation équivalente des opérateurs de Kraus. C'est-à-dire qu'elles permettent de décrire les transformations du système sous une forme unitaire élargie, facilitant leur implémentation sur des ordinateurs quantiques. L'évolution temporelle de ces matrices permettent donc d'obtenir l'évolution des systèmes étudiés. Les matrices de dilatation pourront ensuite être interprétées en termes de portes quantiques traditionnelles et donc de circuits quantiques.

Ce mémoire est structuré en trois parties, chacune abordant des aspects clés des systèmes quantiques ouverts et de leur implémentation sur des ordinateurs quantiques. La première partie présente le fonctionnement des ordinateurs quantiques, en expliquant les principes fondamentaux qui les sous-tendent. Ensuite, la deuxième partie se concentre sur les systèmes quantiques ouverts, en détaillant leur dynamique. Puis, la troisième partie traite de l'utilisation de l'algorithme [1] pour déterminer la dynamique de systèmes quantiques ouverts.

Pour terminer, nous appliquerons l'algorithme dans des exemples concrets sur du véritable hardware quantique fourni par la division quantique d'IBM, disponible gratuitement sur le cloud. Ceci nous permettra de comparer les performances réelles des ordinateurs quantiques actuels aux résultats théoriques correspondants.

Avant de commencer, j'aimerais préciser que nous omettrons volontairement les $\hat{}$ sur les opérateurs afin de simplifier la notation. Je vous souhaite une bonne lecture.

Première partie

Information et ordinateurs quantiques

Dans cette partie, nous explorerons les concepts fondamentaux des ordinateurs quantiques et de leur fonctionnement. Nous commencerons par examiner l'évolution du domaine de l'information quantique, en retraçant les progrès récents et les tendances actuelles. Ensuite, nous introduirons des concepts essentiels tels que les états d'un système quantique, la sphère de Bloch et les portes quantiques, qui sont les éléments de base pour manipuler les qubits.

Nous aborderons également les circuits quantiques et leur équivalence, une notion importante pour la simplification et l'optimisation des opérations quantiques. Nous approfondirons ensuite l'étude des qubits supraconducteurs, une technologie clé dans la construction des ordinateurs quantiques actuels, avec un focus particulier sur les contributions d'IBM dans ce domaine.

Enfin, nous discuterons du bruit quantique, un challenge prédominant en informatique quantique, ainsi que de certaines méthodes qui permettent de limiter les erreurs dues au bruit.

Introduction

Dans un avenir proche, les ordinateurs quantiques sont susceptibles de révolutionner de nombreux domaines grâce à leur capacité unique à manipuler des qubits, contrairement aux ordinateurs classiques qui utilisent des bits binaires traditionnels (0 et 1). Un qubit est un système quantique à deux niveaux pouvant exister dans une superposition d'états. Cependant, la décohérence, qui est le processus par lequel un qubit perd son état quantique en raison de son interaction avec l'environnement, est un problème majeur en informatique quantique [18]. En maîtrisant et en limitant la décohérence, les ordinateurs quantiques pourraient devenir encore plus fiables et plus performants. L'attente envers les ordinateurs quantiques est, qu'un jour, ils atteignent la suprématie quantique. C'est-à-dire qu'ils puissent résoudre des problèmes pour lesquels un ordinateur classique ne pourrait pas y parvenir. Cela nécessiterait un nombre assez grand de qubits qui doivent être suffisamment protégés de la décohérence [19].

2 L'évolution du domaine de l'information quantique

Les ordinateurs quantiques permettent de développer des algorithmes quantiques capables de modéliser des systèmes quantiques ouverts. En 1982, Feynman a été le premier, à ma connaissance, à proposer l'utilisation d'algorithmes quantiques pour simuler des systèmes quantiques [20].

Depuis les débuts de l'information quantique, les enjeux se trouvaient au niveau des sources de bruit. Dès lors, en 1995, des études approfondies sur ces sources ont permis de réduire les erreurs dues à la décohérence [21]. En 2005, une étude a montré que le taux d'erreur pouvait descendre en dessous de 1% par porte [22].

Au cours des vingt dernières années, le domaine des circuits quantiques a connu une évolution marquée par des avancées significatives. À la fin des années 90, la recherche se concentrait principalement sur les aspects fondamentaux de la conception des qubits et des interactions quantiques [23]. Les premiers qubits supraconducteurs, qui sont un type particulier de qubits, ont été réalisés au terme des années 1990 [24, 25].

À partir des années 2010, l'accent s'est déplacé vers l'ingénierie de systèmes quantiques supraconducteurs à grande échelle. En effet, le nombre de types de qubits différents a considérablement augmenté, avec des améliorations notables dans les performances des qubits, atteignant des temps de cohérence de l'ordre de plusieurs dizaines de secondes [26].

Depuis 2012, les techniques de contrôle des qubits ont largement évolué, permettant des opérations quantiques avec une fidélité supérieure à 99,99% [27]. C'est-à-dire la probabilité que le résultat

obtenu, après une opération, soit conforme à l'état attendu avec un taux d'erreur inférieur à 0,01%. Très peu de temps après, l'intégration de technologies complémentaires telles que la cryogénie [28] a permis de concevoir des systèmes quantiques plus complexes et évolutifs.

Ensuite, en 2019, l'émergence de nouvelles applications en calcul quantique, telles que la démonstration de portes quantiques universelles [29], a marqué une étape importante dans le développement de l'informatique quantique. En effet, des expériences ont montré des progrès vers l'objectif de la suprématie quantique grâce à l'utilisation de transmons [30] qui est un qubit supraconducteur particulier. Ce dernier a été conçu pour surmonter les limitations des qubits conventionnels et a permis d'offrir des performances améliorées en termes de cohérence et de contrôle. Finalement, la suprématie quantique a été atteinte en 2019 à l'aide d'un processeur supraconducteur [19].

Très récemment, en 2024, la société Quantinuum a lancé le processeur H2, un système piégeant des ions qui a établi des records de performance et a démontré des capacités dépassant ce qui peut être simulé par des ordinateurs classiques [31].

Également en 2024, une avancée notable dans l'informatique quantique a été réalisée grâce à une collaboration entre Microsoft et Quantinuum. Ils ont développé un système capable de créer 4 qubits logiques, qui sont des représentations corrigées des qubits physiques, à partir de 30 qubits physiques avec un taux de fidélité de 99,8 % pour les opérations entre qubits [32].

3 Les états d'un système quantique

L'unité de base de l'ordinateur quantique est le qubit, l'équivalent quantique du bit classique. Alors qu'un bit classique peut être soit 0 soit 1, un qubit peut être dans une superposition de ces états (Fig. 1), représenté par une combinaison linéaire $\alpha|0\rangle + \beta|1\rangle$, où α et β sont des coefficients complexes. Ces coefficients déterminent les probabilités de mesurer respectivement les états $|0\rangle$ et $|1\rangle$, avec $|\alpha|^2 + |\beta|^2 = 1$. De là, un état quantique est mathématiquement représenté par un vecteur $|\psi\rangle$ appartenant à un espace de Hilbert \mathcal{H} , où $|\psi\rangle$ décrit la configuration complète du système quantique. La norme d'un tel vecteur d'état $|\psi\rangle$ est définie via le produit scalaire $\| |\psi\rangle \| = \sqrt{\langle \psi | \psi \rangle}$, et doit être égale à 1 pour assurer la conservation des probabilités.

Il est important de distinguer deux types d'états quantiques : les états purs et les états mixtes. Un état pur, comme ceux décrits jusqu'à présent, est entièrement caractérisé par un vecteur dans l'espace de Hilbert, par exemple $|\psi\rangle$. En revanche, un état mixte représente une situation où le système quantique est dans une combinaison statistique de plusieurs états purs et il peut être décrit par une matrice densité ρ . Cette distinction sera détaillée dans la section dédiée à l'opérateur densité.

Nous utiliserons la *base computationnelle* pour représenter les états d'un ou de deux qubits qui sont respectivement $\{|0\rangle, |1\rangle\}$ et $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Notons que nous adopterons la notation $|ij\rangle = |i\rangle \otimes |j\rangle$ pour exprimer plusieurs qubits, où \otimes désigne le produit vectoriel. Les états $|0\rangle$ et $|1\rangle$ sont souvent représentés dans un formalisme matriciel comme suit

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (1)$$

Dans cette représentation, les états $|0\rangle$ et $|1\rangle$ sont des vecteurs formant une base orthogonale de l'espace d'Hilbert, c'est-à-dire que $\langle 0|1\rangle = 0$. Notons que le vecteur dual du vecteur d'état $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ est le vecteur $\langle \psi| = (\alpha^* \quad \beta^*)$. Le produit scalaire de ces vecteurs est donc :

$$\langle \psi | \phi \rangle = (\psi_1^* \quad \psi_2^*) \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \psi_1^* \phi_1 + \psi_2^* \phi_2. \quad (2)$$

Les opérations sur les qubits sont effectuées à l'aide de portes quantiques, qui sont des transformations unitaires agissant sur l'état des qubits. En effet, au vu de l'équation de Schrödinger

$|\psi(t)\rangle = e^{-iHt/\hbar} |\psi(0)\rangle$, la porte représentant la transformation $e^{-iHt/\hbar}$ est nécessairement unitaire. En fait, les portes quantiques sont l'analogie des portes logiques classiques en informatique traditionnelle. Elles permettent d'effectuer des opérations logiques sur les qubits. Une section est dédiée à l'explication de ces portes quantiques.

Une porte quantique peut préparer un qubit dans une superposition d'états $|0\rangle$ et $|1\rangle$. Cette superposition d'états "collapse" dans un état classique déterministe lorsqu'on effectue une mesure, ce qui peut introduire des erreurs. Nous verrons, par la suite, les postulats de la mécanique quantique qui traitent de ce sujet.

L'espace de Hilbert \mathcal{H} associé à un système quantique composé de n qubits a une dimension 2^n car chaque qubit peut être dans une superposition des deux états $|0\rangle$ et $|1\rangle$. Ainsi, la base de cet espace est constituée des 2^n états de la forme $|i_1 i_2 \dots i_n\rangle$, où $i_k \in \{0, 1\}$ pour chaque qubit k . Par conséquent, à mesure que le nombre de qubits augmente, la dimension de l'espace de Hilbert croît exponentiellement. Cette propriété permet de représenter une quantité importante d'états qui peuvent être cependant difficiles à contrôler.

Dans les systèmes à plusieurs qubits, l'intrication quantique est un phénomène fondamental, où l'état global du système ne peut pas être décomposé en un produit d'états individuels des qubits. Deux qubits intriqués, par exemple, peuvent être dans un état où les résultats de mesure de l'un dépendent instantanément des résultats de l'autre, quelle que soit la distance qui les sépare. Cette propriété, qui contraste avec la physique classique, permet des phénomènes tels que la téléportation quantique [33] et joue un rôle clé dans les algorithmes quantiques et les protocoles de communication sécurisée. Nous verrons par la suite comment préparer de tels états dans un circuit quantique.

3.1 La sphère de Bloch

Afin de pouvoir représenter un état quantique à deux niveaux, nous utiliserons la sphère de Bloch. Celle-ci utilise des coordonnées polaires θ et ϕ , qui positionnent l'état du qubit sur la surface d'une sphère. Sur cette sphère, les états de base $|0\rangle$ et $|1\rangle$ correspondent au pôle nord et au pôle sud, tandis que toute superposition se situe ailleurs sur la surface de la sphère. Dès lors, les coefficients α et β de l'état quantique $|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle$ peuvent être directement reliés à ces angles via les relations $\alpha = \cos(\theta/2)$ et $\beta = e^{i\phi} \sin(\theta/2)$ avec $0 \leq \theta < \pi$ et $0 \leq \phi < 2\pi$. Les 3 axes x , y et z , dont l'origine en est le centre de la sphère, sont définis par :

$$\begin{cases} x = \sin \theta \times \cos \phi \\ y = \sin \theta \times \sin \phi \\ z = \cos \theta. \end{cases}$$

Ces axes correspondent aux observables X , Y et Z associées aux matrices de Pauli X , Y et Z que nous définirons dans la section suivante. Notons que la mesure sur Z donne les probabilités associées à l'obtention des états $|0\rangle$ et $|1\rangle$. À partir de cette sphère, on peut visualiser l'évolution temporelle d'un état en faisant tourner son vecteur sur celle-ci. Enfin, la sphère de Bloch permet également de modéliser la décohérence. En effet, lorsque l'état se rapproche du centre de la sphère, cela indique une perte de pureté due aux interactions avec l'environnement.

Cette sphère, de rayon unitaire, offre une représentation graphique intuitive de l'état quantique du qubit, permettant ainsi de mieux comprendre ses propriétés. On peut, par exemple, représenter un vecteur d'état $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ (Fig. 1), où nous avons que $|\alpha|^2 + |\beta|^2 = 1$.

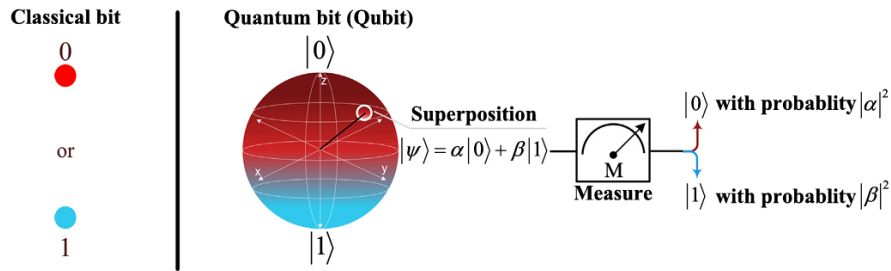


FIGURE 1 – À gauche, le bit classique, où les deux seuls états possibles sont 0 et 1. À droite, l'état quantique général $\alpha|0\rangle + \beta|1\rangle$ est représenté sur la sphère de Bloch. Celui-ci est une superposition des états $|0\rangle$ et $|1\rangle$. Cette superposition peut être mesurée avec la probabilité $|\alpha|^2$ d'obtenir l'état $|0\rangle$ et la probabilité $|\beta|^2$ d'obtenir l'état $|1\rangle$. Schéma issu de [34].

4 Les circuits quantiques

Lors de ce mémoire, nous serons amenés à réaliser des circuits quantiques qui sont une succession de portes quantiques unitaires. Cette section vise à expliquer en quoi consiste une porte quantique et comment des circuits quantiques peuvent être équivalents.

4.1 Les portes quantiques

Dans la théorie de l'informatique quantique, les portes quantiques sont des opérations fondamentales qui permettent de manipuler les états quantiques de qubits. Ces portes sont représentées mathématiquement par des matrices unitaires qui agissent sur l'espace de Hilbert des qubits et leur application consiste en des rotations d'états quantiques dans cet espace. Par exemple, des portes comme la porte Hadamard H ou la porte Pauli-X effectuent des transformations spécifiques sur un seul qubit, tandis que d'autres, comme la porte $CNOT$, permettent de manipuler l'intrication entre deux qubits. Ces portes sont définies juste après.

Les portes quantiques constituent les éléments de base des circuits quantiques et peuvent être appliquées sur des ordinateurs quantiques. De là, on peut créer des algorithmes quantiques qui sont construits en assemblant des séquences de portes élémentaires qui exécutent des calculs sur les qubits. Pour mener à bien ce mémoire, nous nous appuierons sur les portes disponibles mises à disposition par IBM et nous chercherons à concevoir des circuits quantiques capables d'implémenter les dynamiques de systèmes quantiques ouverts, en utilisant ces portes comme blocs de construction.

L'objectif principal de ce mémoire est de déterminer les circuits quantiques les plus adaptés pour représenter l'évolution d'un système quantique ouvert à partir de son équation maîtresse. Cette approche requiert non seulement une compréhension approfondie de chaque type de porte mais aussi une maîtrise de leur agencement pour obtenir des circuits cohérents avec les prédictions théoriques.

Dans la suite de cette section, nous allons détailler les différentes portes quantiques actuellement disponibles dans les ordinateurs quantiques d'IBM [16], afin d'appréhender leurs fonctionnalités et de les utiliser, plus tard, lors de la simulation de systèmes quantiques ouverts.

1. **La porte de Hadamard** est un opérateur à un qubit qui transforme une base d'états classiques $\{|0\rangle, |1\rangle\}$ en une base diagonale d'états superposés $\{|+\rangle, |-\rangle\}$ [35]. Elle est de taille 2×2 et est définie comme :

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (3)$$

L'action de cet opérateur dans la base computationnelle est :

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \quad (4)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle. \quad (5)$$

La porte de Hadamard est généralement représentée par un H .

2. **Les portes de rotation** sont également fréquemment utilisées. Elles appliquent une rotation d'un angle défini autour d'un des axes de la sphère de Bloch. Elles sont notées RK avec $K \in \{X, Y, Z\}$. Elles agissent sur un seul qubit à la fois et sont définies comme

$$RX(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix},$$

$$RY(\phi) = \begin{pmatrix} \cos(\frac{\phi}{2}) & -\sin(\frac{\phi}{2}) \\ \sin(\frac{\phi}{2}) & \cos(\frac{\phi}{2}) \end{pmatrix},$$

$$RZ(\lambda) = \begin{pmatrix} e^{-i\lambda/2} & 0 \\ 0 & e^{i\lambda/2} \end{pmatrix}.$$

3. **Les portes de Pauli** correspondent aux opérateurs de Pauli et sont souvent utilisées pour effectuer des rotations de spin sur un qubit unique. Elles peuvent être vues comme une rotation d'un angle π sur la sphère de Bloch.

- (a) La porte **Pauli-X** effectue une rotation de π autour de l'axe x dans la sphère de Bloch, ce qui correspond à un spin flip. Elle est également simplement appelée porte X ou porte NOT . Cette porte est l'équivalent quantique de la porte NOT en mécanique classique et est définie comme

$$X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (6)$$

- (b) La porte **Pauli-Y** effectue une rotation de π autour de l'axe Y sur la sphère de Bloch, ce qui correspond à une transformation du qubit $|0\rangle$ en $i|1\rangle$ et du qubit $|1\rangle$ en $i|0\rangle$. Elle est définie comme

$$Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \quad (7)$$

- (c) La porte **Pauli-Z** effectue une rotation de π autour de l'axe Z sur la sphère de Bloch, ce qui correspond à une transformation uniquement du qubit $|1\rangle$ en $-|1\rangle$, laissant le qubit $|0\rangle$ inchangé. Elle est également appelée porte phase-flip car elle correspond à un changement de la phase de π . Elle est définie comme

$$Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (8)$$

Elle peut être exprimée en fonction des portes Pauli-X et Pauli-Y par la relation $Z = -iXY$. Cela montre que Z résulte de la combinaison d'un bit-flip (X) et d'une phase-flip (Y) avec un facteur imaginaire.

4. **La porte U_3 d'IBM** correspond à une rotation sur un qubit en utilisant les trois angles θ , ϕ , et λ . En particulier, elle peut être utilisée pour implémenter toutes les autres portes de rotation. La matrice U_3 est définie par :

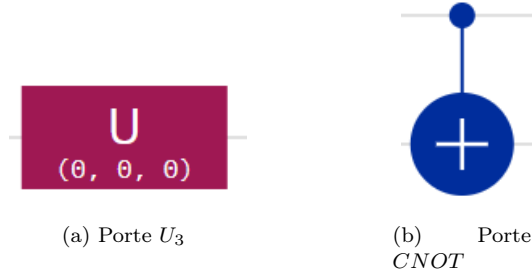


FIGURE 2 – Représentation des portes quantiques sur IBM Quantum Learning (Composer, [16]) : (a) La porte U_3 est configurée avec les trois angles égaux à 0. (b) La porte $CNOT$ est configurée avec le qubit de contrôle et le qubit cible qui sont respectivement les qubits supérieur et inférieur.

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda} \sin\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)} \cos\left(\frac{\theta}{2}\right) \end{pmatrix}.$$

Cette porte est très utilisée dans les algorithmes quantiques en raison de sa flexibilité et de sa capacité à générer des rotations complexes sur un seul qubit. Elle est représentée sur la figure (2).

5. **La porte $CNOT$** , ou porte NOT contrôlée, est une porte à deux qubits. Elle effectue un flip sur le second qubit (appelé qubit cible) seulement si le premier qubit (appelé qubit de contrôle) est dans l'état $|1\rangle$. Autrement dit, si le qubit de contrôle est dans l'état $|1\rangle$, alors l'état du qubit cible passera de l'état $|0\rangle$ à l'état $|1\rangle$ mais si le qubit de contrôle est dans l'état $|0\rangle$, alors rien ne se passe. La matrice représentant cette porte dans la base computationnelle est :

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

La porte $CNOT$ est souvent représentée par une ligne reliant un point sur le qubit de contrôle à un cercle avec une croix sur le qubit cible (Fig. 2). Elle est essentielle dans la création de l'intrication dans de nombreux algorithmes quantiques.

4.2 L'équivalence des circuits quantiques

L'équivalence entre deux circuits signifie que l'effet global de l'application d'un circuit est identique à l'effet global de l'application de l'autre circuit. Cette notion permet de simplifier les circuits sans en altérer les résultats finaux. De là, la synthèse des circuits quantiques est intéressante pour optimiser les ressources, notamment le nombre de qubits et de portes nécessaires [36].

Cette propriété d'équivalence est rendue possible grâce à des théorèmes de décomposition tels que la *décomposition ZYZ* [37], qui permettent de représenter une matrice unitaire en termes de rotations autour des axes. Plus précisément, une porte à un qubit peut être vue comme la succession d'une porte de multiplication scalaire, d'une porte $RZ(\alpha)$, d'une porte $RY(\beta)$ et finalement, d'une porte $RZ(\gamma)$ (Fig. 3). Une porte de multiplication scalaire correspond à une opération triviale qui laisse l'état quantique inchangé, sauf pour une éventuelle phase globale.

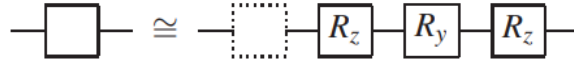


FIGURE 3 – Représentation de la décomposition ZYZ d’une porte quelconque à un qubit. La porte schématisée en pointillés représente la porte de multiplication scalaire. Schéma issu de [38].

Un déphasage $e^{i\alpha}$ sur un qubit cible, contrôlé par un qubit de contrôle, agit comme suit : si le qubit de contrôle est $|0\rangle$, le qubit cible reste inchangé ; si le qubit de contrôle est $|1\rangle$, le déphasage est appliqué au qubit cible. Cette porte à deux qubits peut être représentée par une porte à un qubit équivalente [39] (Fig. 4). Cette équivalence réduit la complexité des circuits quantiques en minimisant le nombre de qubits nécessaires, ce qui est crucial dans les dispositifs quantiques actuels et optimise les algorithmes, notamment ceux impliquant des interactions entre plusieurs qubits dans les systèmes quantiques ouverts.

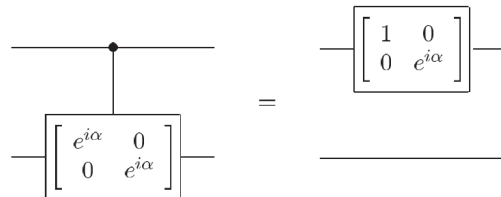


FIGURE 4 – Gauche : Déphasage contrôlé par le qubit supérieur, où l’effet est appliqué uniquement sur le qubit cible inférieur. Droite : Porte qui agit exclusivement sur le qubit supérieur. Schéma issu de [39].

Une autre application importante de cette équivalence est dans la réduction du temps d’exécution des circuits quantiques. En diminuant le nombre de portes à deux qubits, on limite les interactions longues et potentiellement sources de décohérence. Cela permet de rendre l’exécution des algorithmes plus rapide.

5 Les qubits supraconducteurs

Le fonctionnement des qubits supraconducteurs repose sur l’état de base $|0\rangle$ et l’état excité $|1\rangle$. Ils correspondent à des courants circulants dans des boucles supraconductives dans des directions opposées. Le qubit supraconducteur le plus couramment utilisé est le qubit de phase, qui tire parti de la phase quantique des courants circulants. En modulant la phase de manière contrôlée, les qubits de phase peuvent être placés dans des superpositions cohérentes d’états $|0\rangle$ et $|1\rangle$.

Les qubits supraconducteurs sont généralement réalisés sous forme de circuits électriques microscopiques, intégrés dans des dispositifs appelés cavités ou résonateurs (Fig. 5). Ces circuits sont fabriqués à partir de matériaux supraconducteurs tels que l’aluminium ou le niobium, refroidis à des températures proches du zéro absolu pour éliminer toute résistance électrique.

Le **transmon** est un type de qubit supraconducteur qui est une variante du qubit de charge (Fig. 5, a). Il utilise une jonction Josephson avec une capacité plus grande, ce qui le rend moins sensible aux fluctuations de charge et améliore sa stabilité. Le transmon exploite la phase du courant circulant dans une boucle supraconductrice et est particulièrement apprécié pour sa robustesse contre les bruits environnementaux. Cette conception permet d’atteindre une meilleure fidélité et une plus grande cohérence, ce qui en fait un choix populaire pour les systèmes quantiques modernes.

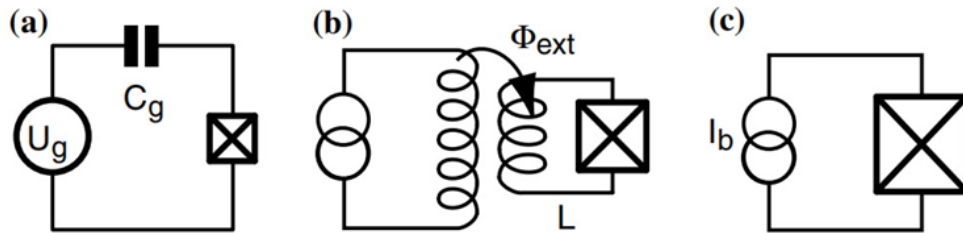


FIGURE 5 – Les trois qubits supraconducteurs de base. (a) Transmon (qubit de charge) (b) RF SQUID (qubit de flux) et (c) jonction polarisée en courant (qubit de phase). Schéma issu de [40].

Les composants clés d'un qubit supraconducteur comprennent une boucle supraconductrice, une jonction Josephson et un dispositif de lecture et de contrôle. La boucle supraconductrice agit comme le support du qubit, tandis que la jonction Josephson permet de contrôler la phase quantique et d'induire des transitions entre les états $|0\rangle$ et $|1\rangle$. Le dispositif de lecture et de contrôle détecte l'état du qubit et permet d'effectuer des opérations quantiques.

Les qubits supraconducteurs présentent plusieurs avantages, notamment leur haute fidélité, leur vitesse de fonctionnement et leur facilité d'intégration avec les technologies de fabrication de semi-conducteurs existantes. De plus, leur capacité à former des réseaux de qubits interconnectés ouvre la voie à des applications telles que la cryptographie quantique. Un avantage important des qubits supraconducteurs est leur capacité à être intégrés dans des circuits physiques sur une puce. Cette intégration permet de réaliser des systèmes quantiques à grande échelle, où plusieurs qubits peuvent être manipulés et contrôlés simultanément. Cette approche ouvre la voie à la réalisation de calculs quantiques à grande échelle.

Cependant, les qubits supraconducteurs sont sensibles aux fluctuations environnementales, telles que les variations de température et de champ magnétique, ce qui peut entraîner une décohérence et des erreurs quantiques. De plus, la réalisation de qubits supraconducteurs à grande échelle nécessite des avancées dans la conception et la fabrication de circuits quantiques, ainsi que des techniques de correction d'erreurs quantiques poussées.

6 IBM et ses contributions aux ordinateurs quantiques

IBM, l'acronyme de International Business Machines, est une entreprise qui fabrique et commercialise du matériel informatique et des logiciels. Cette entreprise propose également des services d'infrastructure, d'hébergement Internet et de conseil dans un large éventail de domaines liés à l'informatique. Parmi ces domaines, nous pouvons retrouver les dockers, les IA, le deeplearning, la blockchain mais surtout, l'informatique quantique. Une branche d'IBM est IBM Quantum dont l'objectif est de propager les ordinateurs quantiques et leur utilisation dans le monde. Ils se concentrent sur la recherche et le développement de technologies d'informatique quantique, la conception et la construction de systèmes d'ordinateurs quantiques. Ils proposent des services cloud qui permettent aux utilisateurs d'accéder à des ordinateurs quantiques via Internet (Fig. 6). Ils proposent également des tutoriels, cours en ligne et événements dans le but d'aider les gens intéressés à en savoir plus sur l'information quantique.

IBM est à l'avant-garde de la recherche et du développement en informatique quantique et l'une des mesures de sa contribution dans ce domaine est le nombre de qubits qu'elle utilise dans ses systèmes d'ordinateurs quantiques. Nous examinerons l'évolution du nombre de qubits utilisés par IBM dans ses systèmes, ainsi que son importance dans le développement de l'informatique quantique.

IBM Quantum a fait des progrès significatifs dans l'augmentation du nombre de qubits dans ses systèmes au fil du temps. Depuis les premières implémentations expérimentales avec quelques

qubits, IBM a constamment augmenté la capacité de ses ordinateurs quantiques. En effet, pendant des années, IBM a suivi une feuille de route pour l'informatique quantique qui doublait presque le nombre de qubits chaque année. L'entreprise a dévoilé le premier ordinateur quantique doté de plus de 1 000 qubits, soit l'équivalent des bits numériques d'un ordinateur ordinaire. La puce, appelée Condor, comporte 1 121 qubits supraconducteurs disposés en nid d'abeille [41]. Elle fait suite à d'autres machines d'IBM établissant des records, notamment une puce de 127 qubits appelée Eagle, lancée en 2021 et une autre de 433 qubits appelée Osprey. Cependant, l'entreprise affirme qu'elle va maintenant changer de vitesse et se concentrer sur l'amélioration de la résistance aux erreurs de ses machines plutôt que sur leur agrandissement.



FIGURE 6 – Ordinateur quantique mis à disposition par IBM et accessible par tous [42].

IBM a créé Qiskit qui est un logiciel dédié à l'informatique quantique et qui est adopté à l'échelle mondiale. Lancé en 2017, Qiskit, en tant que kit de développement logiciel, est un outil open-source qui a permis à plus de 550 000 utilisateurs de concevoir et d'exécuter des circuits quantiques sur les systèmes matériels quantiques d'IBM, ce qui a donné lieu à l'exécution de plus de 3 000 milliards de circuits quantiques à ce jour [43]. Au-delà de son rôle initial de kit de développement logiciel, Qiskit s'est transformé en une plateforme complète pour le calcul quantique. Ses services incluent notamment le *Qiskit Runtime Service*, qui optimise l'exécution de circuits quantiques en s'adaptant aux spécificités matérielles et le *Qiskit Transpiler Service* qui réduit le nombre de portes quantiques pour maximiser l'efficacité des circuits. Grâce à ces optimisations, notamment alimentées par l'intelligence artificielle, Qiskit répond aux exigences de charge de travail complexe requises par la recherche quantique contemporaine [43]. Les capacités de Qiskit évoluent en parallèle de l'infrastructure matérielle d'IBM, permettant une réduction significative des besoins en mémoire et des opérations nécessaires [44].

7 Le bruit quantique

Le bruit quantique est un phénomène inhérent aux systèmes quantiques qui a des implications significatives pour le développement et la performance des ordinateurs quantiques. Contrairement aux systèmes classiques, où le bruit peut souvent être modélisé et atténué, le bruit quantique présente des défis uniques en raison de la nature probabiliste et non déterministe des états quantiques. Cette section aborde les différentes sources de bruit quantique et leurs moyens de correction.

7.1 Les sources de bruit quantique

1. **La décohérence**, processus qui a déjà été vu à plusieurs reprises, est une source de bruit quantique importante. Elle est souvent mesurée par le temps de cohérence, qui indique combien de temps un qubit peut maintenir son état quantique avant d'être perturbé.
2. **Les mesures d'un état** peuvent induire des erreurs en raison de l'imprécision des instruments de mesure ou des fluctuations dans le processus de mesure lui-même.
3. **Le bruit de scintillation**, également appelé bruit $1/f$ ou bruit de flicker, est un type de bruit qui varie inversement avec la fréquence. Il est particulièrement prédominant dans les systèmes supraconducteurs et peut affecter la performance des qubits en introduisant des fluctuations à basse fréquence.
4. **Le bruit de charge** est un type de bruit qui affecte particulièrement les qubits de charge. La densité spectrale du bruit de charge peut être décrite par une relation qui montre des fluctuations importantes à basse fréquence, ce qui limite l'utilité des qubits de charge. Certains types de qubits, comme les transmons, ont été développés pour réduire la sensibilité longitudinale aux fluctuations de charge, ce qui améliore leur performance.

Considérons un système à 1 qubit avec les états $\{|0\rangle, |1\rangle\}$ couplé avec son environnement dont les états non-normalisés sont $\{|e_0\rangle, |e_1\rangle, |e_2\rangle, \dots\}$. Imaginons commencer dans l'état $|\Psi(0)\rangle = |e_0\rangle |x\rangle$ où $|x\rangle$ est soit l'état $|0\rangle$ soit l'état $|1\rangle$. Alors, les interactions système-environnement vont produire des intrications entre les états :

$$\begin{cases} |\Psi(0)\rangle = |e_0\rangle |0\rangle \rightarrow |\Psi(t)\rangle = |e_1\rangle |0\rangle + |e_2\rangle |1\rangle, \\ |\Psi(0)\rangle = |e_0\rangle |1\rangle \rightarrow |\Psi(t)\rangle = |e_3\rangle |0\rangle + |e_4\rangle |1\rangle. \end{cases} \quad (9)$$

Dès lors, nous avons que :

$$|e_0\rangle |x\rangle \rightarrow [|a\rangle \mathbb{1} + |b\rangle X + |c\rangle Y + |d\rangle Z] |x\rangle, \quad (10)$$

avec

$$|a\rangle = \frac{|e_1\rangle + |e_4\rangle}{2}, \quad |b\rangle = \frac{|e_2\rangle + |e_3\rangle}{2}, \quad (11)$$

$$|c\rangle = -i \frac{|e_2\rangle - |e_3\rangle}{2}, \quad |d\rangle = \frac{|e_1\rangle - |e_4\rangle}{2}. \quad (12)$$

Par linéarité, l'équation (10) doit être valide pour tout état $\psi = \alpha |0\rangle + \beta |1\rangle$. Nous pouvons donc généraliser les erreurs sur un état global comme :

$$|\Psi(0)\rangle = |e_0\rangle |\psi\rangle \rightarrow |\psi(t)\rangle = [|a\rangle \mathbb{1} + |b\rangle X + |c\rangle Y + |d\rangle Z] |\psi\rangle, \quad (13)$$

où X est une porte NOT , Y est une porte qui provoque un renversement du spin et une erreur sur la phase et Z est une porte qui provoque une phase [45]. Nous pouvons constater que l'interaction du système et de son environnement provoque des erreurs qui agissent de la même manière que des portes quantiques.

7.2 Les moyens de corrections des erreurs

La correction d'erreurs quantiques revête une importance capitale dans le domaine de l'informatique quantique, visant à garantir la fiabilité des calculs malgré les perturbations causées par le bruit quantique. Les qubits sont sensibles aux influences externes, ce qui peut entraîner des erreurs dans les calculs. Lorsqu'une erreur survient, un processus de décodage est utilisé pour identifier le syndrome d'erreur, fournissant des informations sur la nature de l'erreur. En fonction du syndrome, une correction adaptée est appliquée pour restaurer l'état quantique initial. Dès lors, on utilise des codes correcteurs d'erreurs qui sont des algorithmes conçus pour détecter et rectifier les erreurs quantiques. Parmi les exemples les plus courants, on trouve le code de Shor, le code de Steane et le code de surface [45]. Le code de surface est l'un des codes correcteurs d'erreurs les plus prometteurs pour les systèmes quantiques utilisant des qubits supraconducteurs. Il repose sur des grilles bidimensionnelles de qubits et des mesures de stabilisation pour détecter puis corriger les erreurs.

Il existe également les codes qui sont des approches avancées de correction d'erreurs, empilant plusieurs niveaux de codes correcteurs pour améliorer la fiabilité de la correction. Ces techniques visent à atteindre des seuils d'erreur plus bas. Ensuite, les mesures de stabilisation consistent en des opérations de mesure régulières effectuées sur les qubits pour détecter les erreurs sans perturber le calcul quantique en cours. En réalisant des mesures périodiques, les mesures de stabilisation assurent une surveillance continue du système quantique. Cela permet de détecter les erreurs dès qu'elles se produisent afin d'apporter une correction rapide et efficace.

Une autre façon de réduire les sources de bruit est l'utilisation de matériaux de meilleure qualité et de technologies avancées pour la fabrication de qubits. Par exemple, les qubits supraconducteurs bénéficient de progrès dans la réduction du bruit thermique et des fluctuations électromagnétiques.

Résumé

Dans cette partie, nous avons défini les états d'un système quantique et nous avons utilisé la sphère de Bloch pour pouvoir les représenter. Ces états seront manipulés par des portes quantiques qui ont également été définies et étudiées. Nous utiliserons cette notion de portes quantiques lors de la partie (III). Nous avons également expliqué la notion d'ordinateur quantique notamment via le bruit quantique, les qubits supraconducteurs et les ordinateurs quantiques que IBM met à notre disposition. Nous les utiliserons lors de la partie (III) afin de vérifier l'efficacité de l'algorithme étudié.

Dans la partie suivante, nous introduisons les systèmes quantiques ouverts et les équations gouvernant leur dynamique.

Deuxième partie

Dynamique des systèmes quantiques ouverts

Cette partie vise à fournir une base solide pour comprendre comment les systèmes quantiques qui interagissent avec leur environnement peuvent être décrits. Ces interactions sont essentielles à considérer car elles expliquent la décohérence et le bruit qui sont des phénomènes inévitables dans les dispositifs quantiques réels.

Nous commencerons par rappeler les postulats fondamentaux de la mécanique quantique et par introduire l'opérateur densité. Celui-ci permet de décrire les systèmes quantiques ouverts mixtes. Nous reformulerons ensuite les postulats en utilisant cet opérateur.

La deuxième moitié de la partie sera consacrée à l'étude des équations qui gouvernent l'évolution des systèmes quantiques ouverts. En particulier, nous détaillerons l'équation de Lindblad, l'isomorphisme de Choi-Jamiolkowski, et les opérateurs de Kraus, qui sont des outils mathématiques puissants pour modéliser ces systèmes. Enfin, nous terminerons par l'exemple du canal bit-flip, du canal d'amortissement d'amplitude modifié et d'un canal à deux qubits. Ces exemples illustrent comment ces concepts sont utilisés afin de modéliser des systèmes quantiques ouverts.

Introduction

Les systèmes quantiques ouverts, contrairement aux systèmes isolés, échangent de l'information et/ou de l'énergie et des particules avec un environnement externe (Fig. 7). Cela provoque des effets de décohérence et de dissipation dans le système, comme nous l'avons déjà mentionné. En modélisant un système ouvert par un opérateur densité ρ_S , nous capturons son état statistique y compris ses mélanges probabilistes dus aux interactions avec l'environnement qui est représenté par un opérateur densité ρ_E . La dynamique de ces systèmes ouverts est décrite par l'évolution non unitaire de ρ_S généralement modélisée à l'aide d'équations maîtresses dites équations de Lindblad. Ce type d'équation intègre l'influence de l'environnement de manière formelle.

Nous définissons la trace partielle $Tr_E(\rho_{SE})$ comme une opération qui consiste à sommer les éléments diagonaux de ρ_{SE} dans la base orthonormée $|\psi_E^{(k)}\rangle$ de l'espace de Hilbert de l'environnement E , afin d'obtenir l'état réduit du système S , noté $\rho_S = Tr_E(\rho_{SE})$. En d'autres mots, la trace partielle correspond à $Tr_E(\rho_{SE}) = \sum_k \langle k | \rho_{SE} | k \rangle$, où $\{|k\rangle\}$ forme une base orthonormée de l'environnement E . Elle est fondamentale pour isoler l'état du système S en éliminant les degrés de liberté associés à l'environnement. Cela permet de décrire l'état effectif sans en suivre chaque interaction avec son environnement. La base orthonormée $|\psi_E^{(k)}\rangle$ dans l'espace de Hilbert de l'environnement facilite ce processus en décomposant ρ_{SE} . En effet, cela nous permet de projeter l'état combiné SE sur celui de S . Ceci rend possible une modélisation réaliste de l'évolution de S , prenant en compte l'influence externe sans nécessiter la description complète de l'environnement.

Considérons un atome excité (S) dans un champ électromagnétique ambiant (E), exposé à des faisceaux lasers, qui servent de champs classiques externes. Dans ce système ouvert, l'atome excité peut échanger de l'énergie avec l'environnement (le champ ambiant), ce qui entraîne des phénomènes de décohérence : l'atome perd graduellement sa cohérence quantique en rayonnant de l'énergie sous forme de photons vers l'environnement. Le champ ambiant, modélisé par l'opérateur densité ρ_E , représente les degrés de liberté externes qui interagissent avec l'atome. La présence des faisceaux lasers en revanche, permet d'introduire des interactions contrôlées ce qui provoque des transitions précises entre les niveaux d'énergie de l'atome. La trace partielle $Tr_E(\rho_{SE})$ sur les états de l'environnement permet de décrire l'état réduit de l'atome ρ_S , en tenant compte à la fois des interactions inévitables avec le champ ambiant et des influences contrôlées des faisceaux lasers.

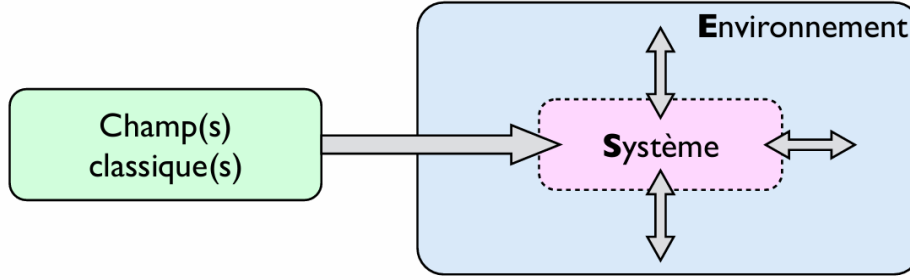


FIGURE 7 – Représentation d’un système quantique ouvert contrôlé par des champs classiques et couplé avec son environnement. Schéma issu de [46].

Par ailleurs, les systèmes quantiques ouverts jouent un rôle capital dans le calcul quantique car l’influence de l’environnement sur le système de qubits peut altérer la fidélité des calculs. Les portes quantiques, dans ce contexte, permettent de simuler ou de compenser les effets de l’interaction avec l’environnement. Nous ne nous intéresserons pas à la manière de modéliser le bruit quantique mais nous le prendrons en compte dans la discussion des résultats qui aura lieu à la partie (III).

Dans le cadre de ce travail, nous modélisons des systèmes quantiques ouverts par des portes quantiques afin de pouvoir les implémenter sur des ordinateurs quantiques. Nous pouvons modéliser de tels systèmes par un opérateur densité ρ_S qui concerne le système isolé et par un opérateur densité ρ_E qui concerne l’environnement (Fig. 8).

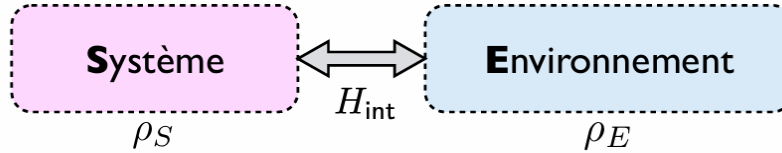


FIGURE 8 – Modélisation d’un système isolé couplé avec son environnement par un Hamiltonien d’interaction. Schéma issu de [46].

Lorsque l’on prend le système entier SE composé du système S et de l’environnement E , le système peut être considéré comme fermé. Il est possible de décrire l’état du système S par l’opérateur densité ρ_S en utilisant la trace partielle qui est définie comme :

$$\rho_S = \text{Tr}_E(\rho_{SE}) = \sum_k \langle \psi_E^{(k)} | \rho_{SE} | \psi_E^{(k)} \rangle, \quad (14)$$

où les vecteurs $|\psi_E^{(k)}\rangle$ forment une base orthonormée de l’espace de Hilbert de l’environnement.

Avant de passer concrètement à la dynamique des systèmes quantiques ouverts, nous allons expliquer les postulats de la mécanique quantique afin d’être sur de pouvoir bien comprendre la suite.

8 Postulats de la mécanique quantique

Les postulats de la mécanique quantique constituent une base théorique sur laquelle repose la description des systèmes quantiques. Dans cette section, nous présenterons ces principes fondamentaux, qui établissent les règles régissant les états, les observables et l’évolution des systèmes quantiques. Cette base nous servira à explorer des concepts plus avancés par la suite.

Postulat I

À tout système quantique est associé un espace de Hilbert complexe et son état est entièrement décrit par des vecteurs d'états, appelés vecteurs ket, $|\psi(t)\rangle$, normalisés dans cet espace. Ce postulat indique que l'état d'un système quantique est décrit par des vecteurs d'états définis dans un espace vectoriel complexe appelé espace des états du système.

Postulat II

L'évolution temporelle d'un état quantique $|\psi(t)\rangle$ est gouvernée par l'équation de Schrödinger qui décrit comment un vecteur d'état du système évolue au cours du temps sous l'action de l'opérateur hamiltonien H du système :

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H |\psi(t)\rangle, \quad (15)$$

où \hbar est la constante de Planck réduite. L'opérateur hamiltonien d'un système représente l'énergie totale de ce système.

Postulat III

Toute opération de mesure sur un système physique, qui implique l'interaction du système initial avec un système externe tel qu'un appareil de mesure, dans le but d'obtenir des informations sur l'état du système, produit nécessairement un ensemble déterminé de résultats. Cette opération est caractérisée par un ensemble d'opérateurs M_k , appelés opérateurs de mesure, qui agissent dans l'espace des états du système, où l'indice k fait référence aux différents résultats de mesure possibles. Les opérateurs de mesure satisfont à la relation

$$\sum_k M_k^\dagger M_k = \mathbb{1}, \quad (16)$$

qui est appelée relation de complétude. Si $|\psi\rangle$ est l'état du système immédiatement avant la mesure, la probabilité d'obtenir le résultat k est

$$p_k = \langle \psi | M_k^\dagger M_k | \psi \rangle. \quad (17)$$

Lorsque le résultat obtenu est p_k , l'état du système juste après la mesure est

$$|\psi_k\rangle = \frac{M_k |\psi\rangle}{\sqrt{\langle \psi | M_k^\dagger M_k | \psi \rangle}}. \quad (18)$$

Postulat IV

À chaque observable quantique O est associé un opérateur linéaire hermitien défini dans l'espace de Hilbert du système. Lorsqu'une mesure de l'observable est effectuée, les résultats possibles correspondent exclusivement aux valeurs propres a_n de cet opérateur, qui satisfait la relation :

$$O |\alpha_n\rangle = a_n |\alpha_n\rangle, \quad (19)$$

où $|\alpha_n\rangle$ sont les vecteurs propres associés aux valeurs propres a_n . Ces vecteurs propres forment une base orthonormée de l'espace de Hilbert, garantissant que tout état quantique peut être décomposé sur cette base.

9 Opérateur densité

En mécanique quantique, l'état d'un système est généralement représenté par un vecteur d'état $|\psi\rangle$, ce qui est adéquat pour décrire des états purs. Cependant, dans des situations plus complexes, comme pour des systèmes ouverts qui interagissent avec leur environnement, ou pour des systèmes dans des états mixtes, cette description peut être lourde. Pour faciliter les calculs, il peut être intéressant de décrire de tels systèmes en utilisant l'opérateur densité, noté ρ . Pour un tel opérateur, il existe un ensemble statistique $\{p_i, |\psi_i\rangle\}$ où les états $|\psi_i\rangle \in \mathcal{H}$ sont associés à la probabilité p_i de telle sorte à ce que :

$$\rho(t) = \sum_{i=1}^n p_i |\psi_i(t)\rangle \langle \psi_i(t)|, \quad (20)$$

avec $\sum_i p_i = 1$, où $p_i \geq 0$ est la probabilité de trouver le système dans l'état pur $|\psi_i\rangle$. Les différents vecteurs d'états $|\psi_i\rangle$ ne sont pas forcément orthogonaux entre eux.

La matrice ρ associée à l'opérateur ρ permet de décrire le système qui est dit dans un état mixte. Dans le cadre de ce travail, nous travaillerons avec des systèmes ouverts, c'est-à-dire des systèmes interagissant avec leur environnement. De tels systèmes sont décrits par plusieurs vecteurs d'états $|\psi_i\rangle$ et il est impossible de les décrire entièrement avec un unique vecteur d'état $|\psi\rangle$. Autrement dit, ces systèmes ouverts sont dans des états mixtes. D'après la définition (20), il est clair qu'un état pur n'est qu'un cas particulier de l'état mixte pour lequel le mélange statistique ne contient qu'un seul état. En effet, comme l'indique le postulat I, l'état d'un système quantique est décrit par les vecteurs d'états de ce système. Lorsqu'il suffit d'un seul vecteur d'état $|\psi\rangle$ pour décrire un système, on dit qu'il est dans l'état pur. Dans ce cas, la probabilité que le système soit dans l'état $|\psi\rangle$ est de 1. Autrement dit, lorsqu'un système se trouve avec certitude à l'instant t dans l'état $|\psi\rangle$ alors l'état $|\psi\rangle$ est appelé état pur. Le système physique est alors dit fermé.

Pour un système dans un état pur, l'opérateur de densité est défini comme le projecteur sur l'état du système considéré

$$\rho(t) = |\psi(t)\rangle \langle \psi(t)|. \quad (21)$$

Dès lors, puisque nous avons les équations (20) et (21), nous pouvons écrire l'opérateur densité comme :

$$\rho = \sum_i p_i \rho_i, \quad (22)$$

où p_i est la probabilité du système d'être dans l'état pur $\rho_i = |\psi_i\rangle \langle \psi_i|$. Nous pouvons également regarder l'opérateur densité dans la base des états du système $\{|i\rangle\}$ pour $i \in \{1, \dots, N\}$. Dans ce cas, la représentation matricielle de l'opérateur densité est

$$\rho = \sum_{i,j=1}^N \rho_{ij} |i\rangle \langle j|, \quad (23)$$

avec ρ_{ij} , l'élément de matrice : $\langle i | \rho | j \rangle$. On peut écrire la matrice ρ comme

$$\rho = \begin{pmatrix} \rho_{11} & \cdots & \rho_{1N} \\ \vdots & \ddots & \vdots \\ \rho_{N1} & \cdots & \rho_{NN} \end{pmatrix}. \quad (24)$$

Les éléments diagonaux, $\langle i | \rho | i \rangle$, de cette matrice sont les éléments formant les populations du système étudié dans les différents états $|i\rangle$.

Voici les propriétés de cet opérateur :

1. L'opérateur densité est un opérateur hermitien :

$$\rho^\dagger = \rho. \quad (25)$$

2. L'opérateur densité est défini positif :

$$\langle \phi | \rho | \phi \rangle = \langle \phi | \psi \rangle \langle \psi | \phi \rangle = |\langle \phi | \psi \rangle|^2 \geq 0, \quad \forall |\phi\rangle \in \mathcal{H}. \quad (26)$$

3. L'opérateur densité est de trace unitaire :

$$\text{tr}(\rho) = 1. \quad (27)$$

4. L'opérateur densité, lorsqu'il agit sur un état pur, est idempotent :

$$\rho^2 = \rho. \quad (28)$$

Dès lors, nous avons que $\text{Tr}(\rho^2) = 1$. En effet,

$$\text{Tr}(\rho^2) = \sum_i \langle i | \rho^2 | i \rangle = \sum_i \langle i | \psi \rangle \langle \psi | \psi \rangle \langle \psi | i \rangle = \sum_i |\langle i | \psi \rangle|^2 = 1. \quad (29)$$

5. Grâce à l'opérateur densité, nous pouvons trouver la valeur moyenne d'une observable O par la relation

$$\langle \psi | O | \psi \rangle = \text{tr}[O\rho]. \quad (30)$$

En effet,

$$\begin{aligned} \langle \psi | O | \psi \rangle &= \sum_{i,j} \langle \psi | i \rangle \langle i | O | j \rangle \langle j | \psi \rangle \\ &= \sum_{i,j} \langle i | O | j \rangle \langle j | \psi \rangle \langle \psi | i \rangle \\ &= \sum_{i,j} \langle i | O | j \rangle \langle j | \rho | i \rangle \\ &= \sum_i \langle i | O \rho | i \rangle \\ &= \text{Tr}(O\rho). \end{aligned}$$

Cet opérateur densité joue un rôle important dans le calcul des valeurs moyennes des observables, dans l'analyse de la décohérence quantique et plus généralement dans la théorie de l'information quantique.

10 Reformulation des postulats avec l'opérateur densité

Nous pouvons reformuler les postulats de la mécanique quantique en termes de l'opérateur de densité qui offre une description plus générale et élégante des systèmes quantiques. Cette reformulation permet d'étendre les principes fondamentaux aux systèmes quantiques mixtes, où des états purs et des mélanges d'états coexistent. Dans cette section, nous présenterons cette approche alternative.

Postulat I

L'état d'un système physique isolé est représenté, à un temps t fixé, par un opérateur de densité $\rho(t)$. Cet opérateur appartient à un espace de Hilbert défini sur le domaine des complexes, appelé espace d'état.

Postulat II

L'évolution temporelle de l'opérateur de densité $\rho(t)$ d'un système physique est décrite par l'équation de Liouville-Von Neumann

$$\dot{\rho}(t) = \frac{-i}{\hbar} [H, \rho(t)], \quad (31)$$

où \hbar est la constante de Planck réduite et H représente l'opérateur hamiltonien du système considéré.

Nous pouvons réécrire l'équation (31) de ce postulat, en prenant la trace partielle sur l'environnement E , comme

$$\dot{\rho}_S(t) = \frac{-i}{\hbar} \text{tr}_E([H_{SE}, \rho_{SE}(t)]). \quad (32)$$

où ρ_S est la matrice densité du système, tr_E est la trace partielle sur l'environnement et SE dénote l'ensemble système + environnement. Cette opération permet de décrire l'évolution du système principal tout en négligeant l'évolution de l'environnement. Dès lors, l'utilisation de la trace partielle permet de modéliser la perte d'information sur l'environnement tout en préservant l'information pertinente pour le système étudié, tout en rendant l'équation de mouvement plus maniable dans le cadre de la dynamique ouverte. Nous utiliserons ce postulat reformulé dans la section 11.4 de la partie suivante.

Postulat III

Si l'état du système est décrit par l'opérateur densité ρ juste avant une mesure, la probabilité que le résultat de la mesure de la quantité physique soit l'une de ses valeurs propres k est :

$$p_k = \text{Tr}(\rho M_k^\dagger M_k). \quad (33)$$

Lorsque le résultat obtenu est p_k , l'état du système juste après la mesure est

$$\rho' = \frac{M_k^\dagger \rho M_k}{\text{Tr}(\rho M_k^\dagger M_k)}. \quad (34)$$

Postulat IV

À chaque observable quantique O est associé un opérateur linéaire hermitien défini dans l'espace de Hilbert du système. Si l'état du système est décrit par l'opérateur densité ρ , la valeur moyenne de l'observable O peut être calculée comme :

$$\langle O \rangle = Tr(\rho O), \quad (35)$$

où Tr représente la trace dans l'espace de Hilbert. Lors d'une mesure, les résultats possibles correspondent aux valeurs propres a_n de l'opérateur O , qui satisfont :

$$O |\alpha_n\rangle = a_n |\alpha_n\rangle, \quad (36)$$

avec $|\alpha_n\rangle$ les vecteurs propres associés. La probabilité d'observer une valeur propre a_n est donnée par :

$$p_n = Tr(\rho |\alpha_n\rangle \langle \alpha_n|), \quad (37)$$

où $|\alpha_n\rangle \langle \alpha_n|$ est le projecteur sur l'état propre $|\alpha_n\rangle$.

Finalement, ce postulat reste inchangé par rapport au même postulat dans le cadre de Schrödinger.

11 L'équation maîtresse

Comme nous venons de le voir, l'opérateur densité $\rho(t)$ est un outil mathématique très utile pour décrire les états mixtes. Nous allons à présent voir l'évolution de ces états au cours du temps par une équation maîtresse et par la représentation de Kraus.

11.1 Approximations

On peut examiner les échelles de temps caractéristiques des systèmes quantiques ouverts pour comprendre l'influence de l'environnement sur leur dynamique. Lorsque le temps de corrélation de l'environnement τ_E , qui représente la durée pendant laquelle l'environnement conserve une "mémoire" de son interaction avec le système, est bien plus court que le temps de relaxation du système τ_R , qui caractérise la vitesse à laquelle le système retourne à l'équilibre sous l'effet de son interaction avec l'environnement, on peut appliquer l'approximation de Born-Markov [47]. Celle-ci suppose que l'information transférée vers l'environnement est rapidement dissipée, ce qui signifie qu'elle ne peut pas revenir dans le système. Ainsi, le système ne "garde pas de mémoire" de ses interactions passées avec l'environnement, et l'évolution future dépend uniquement de l'état présent du système. Cela simplifie les calculs en transformant l'équation maîtresse, qui décrit cette évolution, en une équation différentielle dépendant uniquement du temps actuel [48].

L'approximation séculaire, quant à elle, s'applique lorsque le temps d'évolution typique du système τ_S , qui représente l'échelle de temps sur laquelle l'état du système change de manière significative dû à son hamiltonien propre, est bien plus long que τ_E . Cette approximation consiste à négliger les termes oscillants à haute fréquence, qui n'ont qu'un effet négligeable sur l'évolution lente des populations d'états du système. En ignorant ces oscillations rapides, on se concentre sur l'évolution plus lente et à long terme des états du système, rendant ainsi l'équation maîtresse plus simple à résoudre.

11.2 Équation de Lindblad

Pour parler de systèmes quantiques ouverts, il est commode d'utiliser l'opérateur densité décrit précédemment par une équation d'évolution appelée équation de Lindblad [49, 50] :

$$\dot{\rho}(t) = -i[H, \rho(t)] + \sum_{j=1}^N \mathcal{D}[L_j]\rho(t), \quad (38)$$

où H est le Hamiltonien décrivant l'évolution unitaire du système et où $\mathcal{D}[L_j]$ est le superopérateur de dissipation :

$$\mathcal{D}[L_j]\rho(t) = \gamma_j[L_j\rho(t)L_j^\dagger - \frac{1}{2}\{L_j^\dagger L_j, \rho(t)\}], \quad (39)$$

où les L_j sont les opérateurs de saut quantique et sont associés à un facteur de dissipation γ_k .

Puisque l'équation de Lindblad (39) est linéaire en ρ , il est possible de lui associer un superopérateur \mathcal{L} appelé le Liouvillien et défini comme

$$\dot{\rho}(t) = \mathcal{L}\rho(t), \quad (40)$$

où \mathcal{L} est le Liouvillien composé d'une partie unitaire reprenant l'Hamiltonien du système ainsi que d'un superopérateur de dissipation \mathcal{D} . Le superopérateur \mathcal{L} préserve la trace et génère une application complètement positive $e^{\mathcal{L}t}$ qui décrit l'évolution temporelle du système. Ce superopérateur peut dépendre du temps et lorsque c'est le cas, il possède au moins un état stationnaire qui est un opérateur densité ρ_{SS} , où SS dénote "Steady State", satisfaisant :

$$\mathcal{L}\rho_{SS} = 0. \quad (41)$$

Cette équation signifie que la matrice de densité à l'état stationnaire est une matrice propre du superopérateur \mathcal{L} correspondant à la valeur propre zéro, comme nous le verrons plus loin. Une matrice densité ρ appartient à l'espace des opérateurs $\mathcal{H} \otimes \mathcal{H}$ au contraire du superopérateur \mathcal{L} qui appartient à l'espace de Liouville $L = (\mathcal{H} \otimes \mathcal{H})^* \otimes (\mathcal{H} \otimes \mathcal{H})$. Le produit scalaire dans l'espace de Liouville est défini comme $A.B = Tr(A^\dagger B)$ avec $A, B \in L$.

L'équation de Lindblad peut être comprise dans le cadre d'une approche par *dynamical mapping*, où l'état du système est obtenu en retraçant les degrés de liberté d'un bain environnemental. L'opérateur densité total, décrivant le système et son environnement, suit une dynamique unitaire donnée par le Hamiltonien global. Cependant, en effectuant une trace partielle sur les degrés de liberté du bain, on obtient une dynamique réduite pour l'état du système $\rho(t)$, souvent non unitaire en raison des interactions avec l'environnement.

Cette procédure aboutit à une description effective où l'évolution de $\rho(t)$ est générée par le Liouvillien \mathcal{L} (Fig. 9). La dynamique résultante est irréversible, car les informations perdues dans le bain ne peuvent être récupérées. Cela contraste avec l'évolution unitaire du système+bain, qui reste réversible. L'application dynamique forme un semi-groupe d'opérateurs, traduisant cette irréversibilité fondamentale.

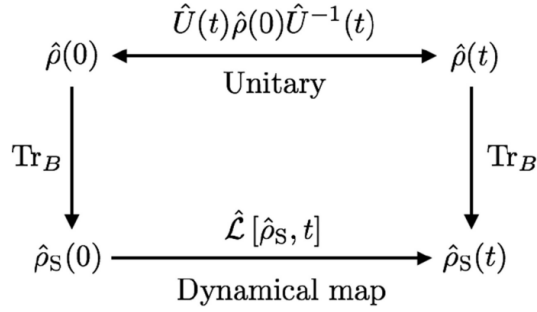


FIGURE 9 – La dynamique unitaire complète du système+bain est décrite par l’équation de Liouville–von Neumann (31), évoluant $\rho(0)$ sous l’action de $H = U(t)\rho(0)U^{-1}(t)$ (flèches horizontales supérieure). La trace partielle (flèches verticales) permet de réduire $\rho(0)$ à $\rho_S(0)$, qui suit une dynamique irréversible sous l’action du Liouvillien \mathcal{L} (flèche horizontale). Schéma issu de [51].

11.3 Isomorphisme de Choi-Jamiolkowski

Il est possible d’établir une correspondance entre les opérateurs quantiques ρ et les états quantiques $|\rho\rangle$ par l’isomorphisme de Choi-Jamiolkowski [52, 53]. En effet, il est possible d’établir une relation entre la matrice densité qui est de taille $N \times N$ et un vecteur de taille $N^2 \times 1$:

$$\rho = \begin{pmatrix} \rho_{11} & \dots & \rho_{1N} \\ \vdots & \ddots & \vdots \\ \rho_{N1} & \dots & \rho_{NN} \end{pmatrix} \iff |\rho\rangle = \begin{pmatrix} \rho_{11} \\ \vdots \\ \rho_{NN} \end{pmatrix}. \quad (42)$$

Cet isomorphisme se base sur l’idée que toute opération quantique, qui peut être vue comme un superopérateur agissant sur des opérateurs d’un espace de Hilbert, peut être représentée par un état quantique dans un espace de Hilbert élargi.

Plus précisément, si nous avons un superopérateur \mathcal{L} qui agit sur un opérateur ρ de l’espace de Hilbert \mathcal{H} , l’isomorphisme de Choi-Jamiolkowski nous dit que nous pouvons associer à \mathcal{L} une matrice densité dans l’espace $\mathcal{H} \otimes \mathcal{H}$. Cette matrice densité est appelée état de Choi.

L’isomorphisme de Choi-Jamiolkowski permet donc d’obtenir une représentation matricielle L du superopérateur \mathcal{L} . De cette manière, nous avons donc simplement

$$|\dot{\rho}\rangle = L|\rho\rangle \quad (43)$$

et la solution de l’équation maîtresse est

$$|\rho(t)\rangle = e^{Lt}|\rho(0)\rangle. \quad (44)$$

La solution de l’équation maîtresse est donc facilement obtenue s’il est possible de calculer l’exponentielle de la matrice représentant le Liouvillien. En pratique, on passe par le calcul des vecteurs propres et valeurs propres du Liouvillien. Comme il n’est en général pas hermitien, il possède des vecteurs propres droits $|\rho_j\rangle$ et gauches $\langle L_j^\dagger|$ satisfaisant

$$L|\rho_j\rangle = \lambda_j|\rho_j\rangle, \quad \langle L_j^\dagger|L = \lambda_j\langle L_j^\dagger|, \quad (45)$$

où λ_j sont les valeurs propres de Liouvillien. Sous forme d’opérateurs, cela donne

$$\mathcal{L}(\rho_j) = \lambda_j\rho_j, \quad L_j^\dagger\mathcal{L} = \lambda_jL_j^\dagger. \quad (46)$$

De l'équation (44), on a donc que l'opérateur densité $\rho(t)$ se décompose dans la base des états propres droits ρ_j et gauches L_j du Liouvillien [54] comme :

$$\rho(t) = \sum_j e^{\lambda_j t} c_j \rho_j, \quad (47)$$

où $c_j = \text{tr}[L_j^\dagger \rho(0)]$.

La valeur propre du Liouvillien correspondant à l'état stationnaire est $\lambda_j = 0$, puisque cela donne $\mathcal{L}(\rho_j) = 0$, ce qui est la définition de l'état stationnaire donnée plus haut dans l'équation (41). De la solution générale (44), nous voyons d'ailleurs que la composante de l'opérateur densité sur l'état propre ρ_j associé à la valeur propre 0 ne change pas au cours du temps, puisque lorsque $\lambda_j = 0$ nous avons $e^{\lambda_j t} = 1$.

Pour déterminer complètement la dynamique du système, la connaissance de la matrice de densité à l'état stationnaire ρ_S ne suffit pas, comme en témoigne l'équation (44) : Il est nécessaire de connaître tout le spectre du superopérateur Liouvillien \mathcal{L} . Il faut faire attention car puisque \mathcal{L} n'est pas hermitien, ses vecteurs propres ne sont généralement pas orthogonaux : $\text{Tr}[\rho_i^\dagger \rho_j] \neq 0$. Si le Liouvillien est diagonalisable, nous pouvons utiliser les états propres \mathcal{L} comme base de l'espace de Liouville. Sous cette hypothèse, on peut trouver une décomposition unique pour tout opérateur A dans la base des vecteurs propres ρ_i de \mathcal{L} :

$$A = \sum_i c_i \rho_i, \quad (48)$$

où c_i , les coefficients de projection.

11.4 Représentation de Kraus

Une méthode alternative à l'équation maîtresse pour décrire la dynamique de systèmes quantiques ouverts est l'utilisation de la représentation de Kraus. Comme nous le verrons dans un instant, les opérateurs de Kraus permettent de modéliser l'évolution d'un état quantique en tenant compte des interactions avec l'environnement. L'algorithme, que nous présenterons dans la prochaine partie, utilise cette représentation. C'est pourquoi nous l'utiliserons tout au long de ce mémoire.

11.4.1 Opérateurs de Kraus

Les opérateurs de Kraus sont utilisés pour représenter les transformations de superopérateurs qui décrivent l'évolution d'un état quantique mixte dans un système ouvert. Si l'on considère un état quantique mixte ρ , son évolution sous l'effet de bruit peut être exprimée par une opération linéaire complètement positive :

$$\boxed{\rho(t) = \sum_k \rho_k(t) = \sum_k M_k(t) \rho(0) M_k^\dagger(t)}, \quad (49)$$

où $\rho(t)$ est la matrice densité du système au temps t et les M_k sont les opérateurs de Kraus qui satisfont à la relation de fermeture :

$$\sum_k M_k^\dagger M_k = \mathbb{1}. \quad (50)$$

Cette condition garantit que la trace de l'état quantique est préservée, c'est-à-dire que la probabilité totale reste égale à un. Chaque opérateur de Kraus, M_k , représente une transformation possible que l'état quantique peut subir en raison de son interaction avec l'environnement. En d'autres termes, l'évolution du système quantique dans un environnement ouvert peut être vue comme une combinaison statistique de plusieurs transformations possibles, chacune ayant une probabilité associée d'occurrence.

Un exemple typique de processus de bruit est le *canal de déphasage* ou *phase damping*, qui décrit une perte de cohérence sans échange d'énergie avec l'environnement. Ce type de bruit est caractérisé par deux opérateurs de Kraus, définis par [39] :

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda(t)} \end{pmatrix}, \quad M_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda(t)} \end{pmatrix}, \quad (51)$$

où $\lambda(t)$ peut être interprétée comme la probabilité qu'un photon provenant du système ait été dispersé sans perte d'énergie. Dans ce canal, M_0 préserve la cohérence tandis que M_1 induit une perte de cohérence sans échanger d'énergie avec l'environnement. L'effet de ce canal sur un état de qubit initialement dans une superposition $\rho = |\psi\rangle\langle\psi|$ avec $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ se traduit par une perte progressive de cohérence (c'est-à-dire, la réduction des termes hors-diagonaux de ρ), tout en conservant les populations des états $|0\rangle$ et $|1\rangle$.

11.4.2 Cas discrétisé

Il est possible de passer de l'équation de Lindblad (38) aux opérateurs de Kraus en discrétisant l'équation par un pas de temps Δt avec $\frac{d\rho}{dt} \approx \frac{\rho(t+\Delta t) - \rho(t)}{\Delta t}$ [55]. Remarquons que lorsque $\Delta t \rightarrow 0$, nous retrouvons l'équation de Lindblad. Nous obtenons ainsi

$$\rho(t + \Delta t) = \mathbb{1} + (G + G^\dagger)\Delta t + \Delta t \sum_{j \neq 0} L_j^\dagger L_j + \mathcal{O}(\Delta t^2) \quad (52)$$

$$= (\mathbb{1} + G^\dagger \Delta t)(\mathbb{1} + G \Delta t) + \Delta t \sum_{j \neq 0} L_j^\dagger L_j, \quad (53)$$

où nous avons introduit $G = -iH - \frac{1}{2} \sum_{j \neq 0} L_j^\dagger L_j$. Cet élément est nécessaire pour que les opérateurs de Kraus puissent satisfaire à la condition de normalisation (50). De cette condition, on peut séparer la contribution du premier opérateur de Kraus A_0 des autres opérateurs A_k en définissant :

$$\begin{cases} A_{k \neq 0}(\Delta t) &= L_k \sqrt{\Delta t} \\ A_0(\Delta t) &= \mathbb{1} + G \Delta t. \end{cases} \quad (54)$$

Ce sont les opérateurs de Kraus A_k pour des temps discrétisés Δt . Cette discrétisation facilite beaucoup les calculs mais n'est valable que dans un domaine très réduit où $\Delta t \ll 1$. Nous allons voir une procédure (Fig. 10) permettant de trouver les opérateurs de Kraus dans le cas continu à partir d'une équation de Lindblad [56].

11.4.3 Cas continu

Le but est de relier notre équation maîtresse $\dot{\rho}(t) = \mathcal{L}\rho(t)$ (40) à la représentation de Kraus (49) :

$$\rho(t) = \phi_t[\rho(0)] = \sum_k M_k(t) \rho(0) M_k^\dagger(t), \quad (55)$$

qui est complètement positive et où les M_k sont les opérateurs de Kraus (50) dans le cas continu. La première étape consiste à trouver les éléments de matrices du Liouvillien L_{kl} dans la base $\{G_i\}$. Puisque le produit scalaire dans l'espace de Liouville est défini par la trace : $A.B = \text{tr}(A^\dagger B)$, nous avons que

$$L_{lm} = \text{tr}[G_l \mathcal{L}(G_m)], \quad (56)$$

où $\{G_a\}$ est une base orthonormée quelconque dans l'espace de Hilbert, à savoir, $G_a = G_a^\dagger$ et $\text{tr}[G_a G_b] = \delta_{ab}$. Pour toute application ϕ , on peut exprimer ρ et $\phi(\rho)$ comme une somme dans une base :

$$\phi(\rho) = \sum_l \text{tr} \left[\phi \left(\sum_m \text{tr}[G_m \rho] G_m \right) \right] G_l = \sum_{l,m} \text{tr}[G_l \phi(G_m)] \text{tr}[G_m \rho] G_l. \quad (57)$$

On peut mettre cette équation sous la forme d'une équation matricielle :

$$\phi(\rho) = (F r)^T G, \quad (58)$$

où $F_{lm} := \text{tr}[G_l \phi(G_m)]$ et où $r_m := \text{tr}[G_m \rho]$. De là, on trouve la matrice F qui résout l'équation $\hat{F} = LF$. Notons que lorsque l'équation maîtresse est indépendante du temps,

$$F(t) = e^{Lt}, \quad (59)$$

[56]. On peut définir la matrice S comme :

$$S_{ab} := \langle \alpha_1 | \phi(|\alpha_2\rangle \langle \beta_2|) | \beta_1 \rangle, \quad (60)$$

où $\{|\alpha\rangle\}$ est une base orthonormée de l'espace de Hilbert avec $\alpha = 1, 2, \dots, N$ et où la transformation linéaire ϕ est une matrice $N^2 \times N^2$. a et b désignent les indices des paires (α_1, α_2) et (β_1, β_2) , respectivement. Cette matrice S correspond à la *représentation de Choi* de l'opérateur ϕ . On peut établir le lien entre S et F [56] par la relation :

$$S_{ab} = \sum_{r,s} F_{sr} \langle \beta_2 | G_r | \alpha_2 \rangle \langle \alpha_1 | G_s | \beta_1 \rangle. \quad (61)$$

En utilisant la base $\tau_a \equiv |\alpha_1\rangle \langle \alpha_2|$, qui forme une base orthonormée non hermitienne dans l'espace de Hilbert, on peut exprimer l'action de ϕ sur la matrice densité ρ sous la forme suivante :

$$\phi(\rho) = \sum_{a,b} S_{ab} \tau_a \rho \tau_b^\dagger. \quad (62)$$

Il est clair que si ϕ agit sur des opérateurs hermitiens, alors la matrice S est également hermitienne, ce qui implique que $S_{ab} = S_{ba}^*$. Choi a démontré que ϕ est complètement positive si et seulement si S est une matrice positive, c'est-à-dire $S \geq 0$ [52]. La matrice S peut toujours être décomposée en une somme de produits extérieurs, ce qui donne :

$$S = \sum_k V(k) V^\dagger(k), \quad (63)$$

où les vecteurs $V(k)$ sont les vecteurs propres normalisés de S . Par exemple, si les valeurs propres et les vecteurs propres normalisés de S sont donnés par l'équation $SX(k) = \lambda_k X(k)$, on peut définir :

$$V(k) = \sqrt{|\lambda_k|} X(k),$$

ce qui mène à la décomposition orthogonale suivante :

$$S = \sum_k \lambda_k X(k) X^\dagger(k).$$

Notons que toutes les décompositions de S sont liées par des transformations unitaires, ce qui signifie que, étant donnée une décomposition particulière, toutes les autres décompositions peuvent être obtenues par des transformations unitaires. Chaque décomposition de S correspond alors à une

décomposition de Kraus de l'opérateur ϕ . En particulier, à partir des équations (62) et (63), on obtient :

$$\phi(\rho) = \sum_{k,a,b} V(k)_a \tau_a \rho \tau_b^\dagger V_b^*(k) = \sum_k M_k(t) \rho M_k^\dagger(t),$$

où les opérateurs de Kraus M_i sont définis par :

$$M_k := \sum_a V(k)_a \tau_a = V(k) \cdot \tau. \quad (64)$$

Résumons cette procédure générale en quelques mots. Vous trouverez une illustration de ces mots sur la figure (10). L'équation maîtresse nous a donné la matrice L , qui permet de déterminer la matrice F , caractérisant l'application complètement positive ϕ . Pour obtenir la décomposition de Kraus, il suffit de déterminer la matrice S à partir de F , puis de la diagonaliser. Ainsi, on peut trouver la décomposition de S en opérateurs de Kraus.

Il est intéressant de noter qu'il n'est pas toujours possible de trouver une équation de Lindblad avec les opérateurs de Kraus d'un système. Cependant, il est toujours possible de trouver des approximations qui permettent de définir une équation de Lindblad correspondante [56].

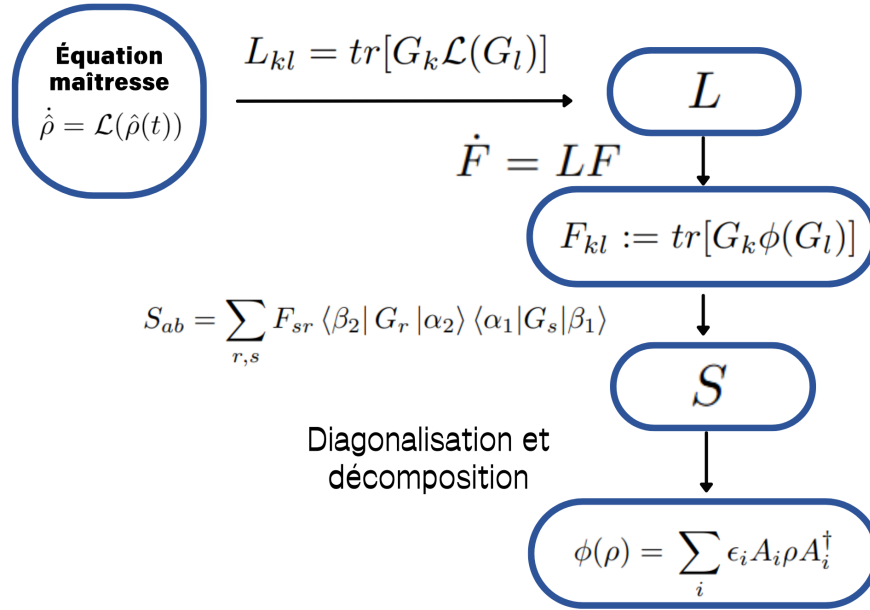


FIGURE 10 – Représentation schématique de la procédure générale : On démarre d'une équation maîtresse pour laquelle on trouve le Liouvillien. Ensuite, on peut trouver une matrice F par la décomposition de ϕ . De là, on peut trouver S qui est la représentation de Choi de ϕ . La diagonalisation de S permet de trouver les opérateurs de Kraus A_i .

11.5 Exemples

Maintenant que nous avons vu la théorie sous-jacente à la dynamique des systèmes quantiques ouverts, nous allons décrire trois exemples de tels systèmes. Pour chaque système, nous expliquerons en quoi il consiste, nous verrons l'équation maîtresse associée, lorsqu'elle est pertinente, ainsi que les opérateurs de Kraus. Cela nous permettra d'établir une base théorique pour leur modélisation sur un ordinateur quantique qui sera décrite dans la prochaine partie.

11.5.1 Canal bit-flip

Le canal bit-flip est un exemple classique de bruit quantique affectant un qubit. Il modélise une situation où l'état d'un qubit subit, avec une probabilité $1 - p$, une inversion de son état logique, c'est-à-dire $|0\rangle \leftrightarrow |1\rangle$. Il est courant de représenter le bit-flip par la représentation de Kraus. On retrouve assez facilement dans la littérature les opérateurs de Kraus M_k (64) pour le bit-flip qui sont définis comme suit [39] :

$$M_0 = \sqrt{p}\mathbb{1} = \sqrt{p} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad M_1 = \sqrt{1-p}\sigma_x = \sqrt{1-p} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (65)$$

avec $\mathbb{1}$ la matrice identité et σ_x la porte de Pauli-X. On peut cependant retomber assez facilement sur l'équation de Lindblad. En effet, puisqu'on sait que l'opérateur M_1 est de la forme $M_1 = f(t)L_1$, alors on constate directement que $L_1 = \sigma_x$. Ensuite, si le seul phénomène étudié est le bit-flip, on peut supposer que $H = 0$ puisque le Hamiltonien H décrit l'évolution unitaire du système, qui est absente dans un processus purement dissipatif comme le bit-flip. Dès lors, si on reprend l'équation de Lindblad (38), avec l'opérateur de saut $L_1 = \sigma_x$, alors on trouve :

$$\dot{\rho}(t) = \gamma \left(\sigma_x \rho(t) \sigma_x^\dagger - \frac{1}{2} \{ \sigma_x^\dagger \sigma_x, \rho(t) \} \right) \quad (66)$$

$$= \gamma (\sigma_x \rho(t) \sigma_x - \rho(t)), \quad (67)$$

avec σ_x , la matrice de Pauli X. On remarquera que $\sigma_x^\dagger \sigma_x = \mathbb{1}$, où $\mathbb{1}$ est la matrice identité de dimension 2 et $\sigma_x^\dagger = \sigma_x$. γ est ici le taux de bit-flip et est relié à p par $p = e^{-\gamma t}$.

Physiquement, le canal bit-flip peut être interprété comme l'interaction d'un qubit avec un environnement qui provoque une erreur de type bit-flip avec probabilité $1 - p$. Par exemple, cela peut représenter le couplage d'un qubit à un bain thermique où les fluctuations thermiques induisent des inversions d'état. Donc, le canal bit-flip constitue un modèle de bruit fondamental souvent utilisé pour concevoir et tester des algorithmes de correction d'erreurs quantiques, car il représente un type simple mais réaliste d'erreur pouvant survenir dans des systèmes quantiques physiques.

11.5.2 Canal d'amortissement d'amplitude modifié

Le canal d'amortissement d'amplitude est un modèle qui décrit la perte d'énergie d'un qubit, par exemple via la relaxation d'un état excité vers l'état fondamental. Un processus qui effectue une telle relaxation est, par exemple, l'émission spontanée. *L'émission spontanée* est un phénomène fondamental en physique quantique, caractérisé par la désexcitation d'un atome ou d'un système quantique à deux niveaux, passant d'un état d'énergie excité à un état d'énergie plus faible (généralement l'état fondamental) tout en émettant un photon.

Lorsqu'un système est dans son état excité, il ne peut y rester indéfiniment à cause de ses interactions avec le vide quantique environnant, qui agit comme un bain de modes électromagnétiques. Cette interaction conduit à une émission spontanée, décrite par un taux de désintégration γ , représentant la probabilité par unité de temps de l'émission d'un photon et du retour à l'état fondamental.

Ce phénomène peut être modélisé par une équation maîtresse :

$$\dot{\rho}(t) = \gamma (\sigma_+ \rho(t) \sigma_- - \frac{1}{2} \{ \sigma_- \sigma_+, \rho(t) \}), \quad (68)$$

avec γ le taux d'émission spontanée, σ_+ et σ_- les opérateurs de Pauli correspondant respectivement à la transition du niveau excité vers le niveau inférieur et inversement. C'est-à-dire $\sigma_+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$

et $\sigma_- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$. Cette équation est de la même forme que l'équation (38) où $H = 0$ et l'opérateur de saut L_k est $\sqrt{\gamma} \sigma_+$. Dès lors, on a que :

$$G = -iH - \frac{1}{2} \sum_{k \neq 0} L_k^\dagger L_k \quad (69)$$

$$= -\frac{1}{2} \sqrt{\gamma} \sigma_- \sqrt{\gamma} \sigma_+ \quad (70)$$

$$= -\frac{\gamma}{2} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad (71)$$

$$= -\frac{\gamma}{2} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \quad (72)$$

On est dans le cas où le superopérateur est indépendant du temps. On peut montrer que, dans ce cas, le premier opérateur de Kraus M_0 peut prendre la forme $M_0 = e^{Gt}$ et que le deuxième opérateur de Kraus M_1 peut prendre la forme $M_1 = f(t)L$, où L est l'opérateur de saut. Dès lors, M_0 vaut

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{e^{-\gamma t}} \end{pmatrix}. \quad (73)$$

Ensuite, en connaissant la forme de l'équation maîtresse, on sait qu'il n'y a que 2 opérateurs de Kraus.

Le deuxième opérateur de Kraus doit être de la forme $M_1 = f(t)L_1 = f(t)\sqrt{\gamma} \sigma_+ = \begin{pmatrix} 0 & \sqrt{\gamma} f(t) \\ 0 & 0 \end{pmatrix}$.

Donc dans ce cas, on peut directement utiliser la relation de fermeture (50)

$$\sum_k M_k^\dagger M_k = M_0^\dagger M_0 + M_1^\dagger M_1 = \begin{pmatrix} 1 & 0 \\ 0 & e^{-\gamma t} + f^2(t) \end{pmatrix} = \mathbb{1}, \quad (74)$$

ainsi,

$$f^2 + e^{-\gamma t} = 1 \iff f^2 = 1 - e^{-\gamma t} \iff f = \sqrt{1 - e^{-\gamma t}}. \quad (75)$$

Finalement, les deux opérateurs de Kraus associés à l'équation maîtresse (78) sont :

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{e^{-\gamma t}} \end{pmatrix}, \quad M_1 = \begin{pmatrix} 0 & \sqrt{1 - e^{-\gamma t}} \\ 0 & 0 \end{pmatrix}. \quad (76)$$

Vous remarquerez que nous avons déjà cité ces opérateurs dans (51), où $\lambda(t) = e^{-\gamma t}$.

Pour modéliser un qubit, donc un système quantique à deux niveaux, il est d'usage [57] d'utiliser l'Hamiltonien :

$$H = \frac{\omega_0}{2} \sigma_z = \frac{1}{2} \begin{pmatrix} \omega_0 & 0 \\ 0 & -\omega_0 \end{pmatrix}, \quad (77)$$

où $\hbar\omega_0$ représente la différence d'énergie entre les deux niveaux du qubit et σ_z est l'opérateur de Pauli Z. Cet Hamiltonien décrit l'évolution temporelle du système en l'absence de toute interaction avec l'environnement. Il introduit une phase relative entre les états excité et fondamental, donnant lieu à une rotation de la matrice densité autour de l'axe correspondant à l'opérateur de Pauli σ_z . Ainsi, le terme $-i[H, \rho(t)]$ dans l'équation maîtresse représente l'évolution unitaire du qubit due à cette différence d'énergie entre les états. Cette évolution est ajoutée à l'effet dissipatif modélisé par l'émission spontanée, représentée par le second terme de l'équation maîtresse. Dès lors, l'équation maîtresse devient :

$$\dot{\rho}(t) = -i[H, \rho(t)] + \gamma \left(\sigma_+ \rho(t) \sigma_- - \frac{1}{2} \{ \sigma_- \sigma_+, \rho(t) \} \right). \quad (78)$$

Dans ce cas, nous avons que $G = -iH - \frac{1}{2} \sum_{k \neq 0} L_k^\dagger L_k = -iH - \frac{1}{2} \gamma \sigma_- \sigma_+$. C'est-à-dire

$$G = \begin{pmatrix} -i\omega_0 & 0 \\ 0 & i\omega_0 - 1/2\gamma \end{pmatrix}. \quad (79)$$

Avec le même argument que précédemment, le premier opérateur de Kraus est

$$M_0 = e^{Gt} = \begin{pmatrix} e^{-i\omega_0 t} & 0 \\ 0 & e^{(i\omega_0 - \frac{1}{2}\gamma)t} \end{pmatrix}. \quad (80)$$

En utilisant les définitions (54), on peut définir le deuxième opérateur de Kraus associé à l'équation maîtresse (78) comme $M_1 = f(t)L_k = f(t)\sqrt{\gamma}\sigma_+ = \begin{pmatrix} 0 & f(t) \\ 0 & 0 \end{pmatrix}$. Ensuite, en utilisant la condition de normalisation (50), c'est-à-dire :

$$\sum_k M_k^\dagger M_k = M_0^\dagger M_0 + M_1^\dagger M_1 = \begin{pmatrix} 1 & 0 \\ 0 & e^{-\gamma t} + f^2 \end{pmatrix} = \mathbb{1}. \quad (81)$$

On retrouve la même condition sur $f(t)$ que précédemment car la contribution de l'hamiltonien dans M_0 est uniquement complexe, sans partie réelle. Dès lors, les composantes de l'hamiltonien s'annulent lors du produit $M_0^\dagger M_0$. On trouve finalement que M_1 reste inchangé par rapport à l'exemple sans hamiltonien. Finalement, les deux opérateurs de Kraus associés à l'équation maîtresse (78) sont :

$$M_0 = e^{Gt} = \begin{pmatrix} e^{-i\omega_0 t} & 0 \\ 0 & e^{(i\omega_0 - \frac{1}{2}\gamma)t} \end{pmatrix}, \quad M_1 = \begin{pmatrix} 0 & \sqrt{1 - e^{-\gamma t}} \\ 0 & 0 \end{pmatrix}. \quad (82)$$

11.5.3 Canal à deux qubits

Dans cet exemple, nous utiliserons un modèle présenté dans la référence [58]. Nous considérons un système composé de deux qubits, dont l'Hamiltonien est $H = \sigma_x \otimes \frac{1}{2}(\mathbb{1} - \sigma_z) + \mathbb{1} \otimes \frac{1}{2}(\mathbb{1} + \sigma_z)$. Cet Hamiltonien décrit en fait l'effet d'une porte *CNOT*, où le deuxième qubit joue le rôle du qubit de contrôle et le premier du qubit cible.

Dans ce modèle en particulier, nous considérons le premier qubit comme le système ouvert et le second qubit comme son environnement. Nous modélisons ainsi ici une dynamique hautement non-Markovienne pour le premier qubit, puisque son environnement possède une mémoire infinie (ici, $\tau_E = \infty$).

Nous choisissons l'état initial du système et de son environnement $\rho_{SE}(0)$, c'est-à-dire l'état initial du premier qubit couplé avec l'état initial du second qubit $\rho_S(0) \otimes \rho_E(0)$ comme celui de la référence [58], à savoir :

$$\rho_{SE}(0) = \rho_S(0) \otimes \rho_E(0) = \begin{pmatrix} \frac{1-r_0}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1+r_0}{2} \end{pmatrix}, \quad (83)$$

où r_0 est un paramètre réel compris entre 0 et 1.

La dynamique du premier qubit, bien que non-Markovienne, peut être représentée par des opérateurs de Kraus, venant de la référence [59], définis comme :

$$M_0(\tau) = \frac{1}{\sqrt{2r_\tau(1+r_0)}} \begin{pmatrix} -\sqrt{(1+r_0)(r_\tau + \sin^2(\tau) - r_0 \cos^2(\tau))} & i\sqrt{(1-r_\tau)(r_\tau - \sin^2(\tau) + r_0 \cos^2(\tau))} \\ -i\sqrt{(1+r_0)(r_\tau - \sin^2(\tau) + r_0 \cos^2(\tau))} & \sqrt{(1-r_\tau)(r_\tau + \sin^2(\tau) - r_0 \cos^2(\tau))} \end{pmatrix}, \quad (84)$$

$$M_1(\tau) = \frac{\sqrt{r_\tau + r_0}}{\sqrt{2r_\tau(1+r_0)}} \begin{pmatrix} 0 & \sqrt{r_\tau + \sin^2(\tau) - r_0 \cos^2(\tau)} \\ 0 & i\sqrt{r_\tau - \sin^2(\tau) + r_0 \cos^2(\tau)} \end{pmatrix}, \quad (85)$$

où τ est le temps sans dimension et $r_\tau = \sqrt{\sin^2(\tau) + r_0^2 \cos^2(\tau)}$.

Résumé

Dans cette partie, nous avons présenté les postulats de la mécanique quantique. Ensuite, nous avons défini l'opérateur densité (20) qui décrit l'état d'un système quantique ouvert. Celui-ci nous a permis d'établir une reformulation des postulats dans un contexte plus approprié pour les systèmes quantiques mixtes. Puis, nous avons défini l'équation maîtresse (38) qui décrit l'évolution de l'opérateur densité au cours du temps. Et puisque notre algorithme utilise les opérateurs de Kraus, nous avons établi une procédure générale pour passer de l'équation maîtresse à la représentation de Kraus. Finalement, nous avons décrit trois exemples de systèmes quantiques ouverts pour lesquels nous avons partagé la représentation de Kraus.

Maintenant que nous avons présenté trois systèmes quantiques ouverts, nous allons étudier comment les modéliser sur un ordinateur quantique réel.

Troisième partie

Dynamique des systèmes quantiques ouverts sur un ordinateur quantique

Puisque que nous avons toute la connaissance théorique nécessaire au sujet des ordinateurs quantiques et des systèmes quantiques ouverts, nous pouvons expliquer comment la mettre en pratique. Cette partie vise à expliquer un algorithme [1] qui permet d’implémenter la dynamique de systèmes quantiques ouverts (II). Nous verrons également comment modéliser les matrices de l’algorithme en circuits quantiques. Pour ce faire, nous passerons d’abord par une préparation des états. Ensuite, nous verront quels circuits permettent d’évoluer ces états dans le temps. Nous finirons enfin par implémenter l’algorithme sur les trois exemples (11.5.1, 11.5.2, 11.5.3) qui ont été présentés dans la partie précédente.

Introduction

L’algorithme présenté dans cette partie génère des matrices unitaires. Cependant, ces matrices ne correspondent pas directement aux portes unitaires à un ou deux qubits que nous avons introduites précédemment. Pour implémenter un circuit sur un ordinateur quantique, ce qui est l’objectif de ce mémoire, il est nécessaire d’utiliser des portes unitaires, mais il faut aussi qu’elles consistent en des *portes à un ou deux qubits* seulement, pour des raisons pratiques. C’est pourquoi, nous ferons la conversion des matrices unitaires obtenues par l’algorithme en matrices représentant des portes à un ou deux qubits.

Afin de trouver ces portes, nous ferons une première conversion des matrices unitaires en matrices à deux niveaux grâce à un protocole donné par [39]. Une matrice à deux niveaux est une matrice unitaire qui agit uniquement sur un sous-espace de dimension deux, correspondant à deux états du système. Ces matrices sont particulièrement utiles pour simplifier l’analyse des systèmes car elles permettent de décomposer des matrices unitaires plus grandes en blocs plus petits. Ces blocs plus petits correspondent aux transitions entre paires d’états. Ces matrices à deux niveaux pourront ensuite nous donner, via un protocole donné par [36], les matrices représentant nos portes quantiques applicables telles que les portes X , H et $CNOT$. D’autres façons d’obtenir ces portes existent comme directement via Qiskit, que nous verrons par la suite.

Après avoir trouvé les circuits représentant les matrices unitaires générées par l’algorithme, il faut préparer les états sur lesquels nous appliquerons ces circuits. Nous expliquerons en détails les différentes façons de préparer des états.

Finalement, nous utiliserons l’algorithme sur un canal bit-flip (116), sur un canal d’amortissement d’amplitude modifié (68) et sur un canal à deux qubits (84), expliqués à la partie précédente pour illustrer l’algorithme. Nous étudierons l’évolution des populations à l’état fondamental et à l’état excité au cours du temps. Nous effectuerons également un changement de base afin de visualiser cette évolution de deux manières différentes.

12 Algorithme basé sur le théorème de Nagy

Introduction

Dans le cadre de l’implémentation des systèmes quantiques ouverts, il est possible de passer de la forme de l’équation maîtresse (38) à la description par opérateurs de Kraus (49) en utilisant la matrice densité (20). Cette approche repose sur une décomposition de la matrice densité en une somme d’opérateurs agissant sur l’état du système (49). C’est cette transition qui sera exploitée dans l’algorithme que nous allons étudier pour simuler efficacement l’évolution des systèmes quantiques ouverts.

Les opérateurs de Kraus, qui rappelons-le, ne sont pas des opérateurs unitaires, peuvent être convertis en matrices unitaires grâce à une dilatation minimale, comme le garantit le théorème de Nagy [60]. La dilatation minimale consiste à trouver la plus petite matrice unitaire qui inclut les opérateurs de Kraus en tant que sous-partie, tout en préservant l’évolution complète et cohérente du système. Cela signifie que l’on agrandit l’espace de Hilbert du système pour inclure un environnement auxiliaire. Cette procédure garantit que, même si le système principal subit des processus non unitaires, l’ensemble système-environnement conserve une

évolution unitaire. Cela permet l'évolution de l'état initial par le biais de portes quantiques unitaires, tout en utilisant beaucoup moins de ressources que celles requises par d'autres dilatations plus conventionnelles comme la dilatation de Stinespring [61]. Ces matrices de dilatation peuvent alors être appliquées sur l'état initial du système. En appliquant ces matrices unitaires, on peut suivre l'évolution temporelle du système en prenant en compte les interactions avec l'environnement.

Nous ferons également une analyse de complexité brève de l'algorithme afin de comprendre et de comparer les ressources nécessaires à son utilisation. Maintenant que nous savons globalement ce que fait l'algorithme, passons à son explication.

12.1 Explication de l'algorithme [1]

Nous ferons évoluer l'opérateur densité $\rho(t)$ suivant la matrice initiale ρ de la forme de l'équation (20), qui représente la composition physique initiale du système. L'état initial du système est important car l'évolution des populations va en dépendre. D'abord, nous préparons les états $|\phi_i\rangle$ liés à notre matrice densité initiale par $\rho(0) = \sum_i p_i |\phi_i\rangle \langle \phi_i|$ (22) où p_i est la probabilité de trouver l'état $|\phi_i\rangle$ avec la condition $\sum_i p_i = 1$. Ces états nous permettront ensuite d'obtenir les vecteurs dans l'espace dilaté v_i . Cette préparation consiste à appliquer des portes H , X et $CNOT$ afin que les états initiaux soient les états voulus. Ensuite, nous utiliserons les opérateurs de Kraus tel que :

$$|\phi_{ik}(t)\rangle = M_k v_i, \quad (86)$$

afin de propager les vecteurs initiaux v_i dans le temps.

Le théorème de dilatation de Nagy [60, 17] est un résultat fondamental en théorie des opérateurs qui permet d'étendre tout opérateur de contraction défini sur un espace de Hilbert à un opérateur unitaire sur un espace de Hilbert plus large. Une contraction A est un opérateur linéaire tel que sa norme est inférieure ou égale à 1. Cela correspond à des transformations qui, en quelque sorte, "compressent" les états du système. Les opérateurs de Kraus sont des contractions. Le théorème garantit qu'il existe un espace de Hilbert plus grand, noté \mathcal{H}_A , et un opérateur unitaire U_A agissant sur cet espace, tels que l'évolution de A sur \mathcal{H}_A peut être exprimée en termes de U_A . En d'autres mots, toute contraction A d'un espace de Hilbert \mathcal{H}_A a un opérateur unitaire correspondant U_A d'un espace de Hilbert plus large \mathcal{H}_U tel que :

$$A^n = P_{\mathcal{H}_A} U_A^n P_{\mathcal{H}_A}, \quad n \in N, \quad (87)$$

où $P_{\mathcal{H}_A}$ est l'opérateur de projection dans l'espace \mathcal{H}_A .

Donc, via l'équation (87), il est possible d'utiliser une matrice unitaire U_A pour reproduire l'effet d'une contraction A qui sont appliqués au maximum N fois. Autrement dit, il est possible d'implémenter des matrices non-unitaires par des portes quantiques unitaires. Nous utiliserons des matrices de dilatation minimales unitaires de la forme (pour $N = 1$) :

$$U_A = \begin{pmatrix} A & D_A^\dagger \\ D_A & -A^\dagger \end{pmatrix}, \quad (88)$$

avec $D_A = \sqrt{\mathbb{1} - A^\dagger A}$, l'opérateur de défauts agissant sur l'opérateur A et $\mathbb{1}$ étant la matrice identité. On peut voir D_A comme un opérateur qui dévie l'action de A . En effet, si A représente une certaine transformation sur un état quantique, D_A représente une transformation orthogonale à celle de A dans l'espace de Hilbert, assurant ainsi que la norme (ou la probabilité totale) est préservée. De plus, nous constatons que si les opérateurs A sont de dimensions $n \times n$, alors la matrice de dilatation U_A pour $N = 1$ sera de taille $2n \times 2n$. Les matrices de dilatation minimales unitaires, pour $N = 2$, ont la forme

$$U_A = \begin{pmatrix} A & 0 & D_A^\dagger \\ D_A & 0 & -A^\dagger \\ 0 & \mathbb{1} & 0 \end{pmatrix}. \quad (89)$$

De cette définition, on peut constater que le nombre N dans l'équation (87) est un paramètre qui définit à la fois la forme et l'applicabilité de la dilatation minimale unitaire. Par la suite, nous travaillerons avec $N = 1$ fixé. Les opérateurs A sont nos opérateurs de Kraus M_k . Après avoir préparé l'état (86), nous pouvons appliquer nos portes unitaires U_{M_k} :

$$|\phi_{ik}(t)\rangle = M_k v_i \rightarrow U_{M_k}(v_i^T, 0, \dots, 0)^T. \quad (90)$$

On peut calculer notre matrice densité $\rho(t)$ via l'équation (20). Cependant, on peut voir que certaines quantités physiques importantes, comme les populations des différents états, peuvent être extraite simplement en appliquant une projection sur l'état $|\phi_{ik}(t)\rangle$ sans pour autant connaître l'état en question. En d'autres mots, nous n'avons pas besoin des coefficients de phases associés à chaque état mais seulement de leurs valeurs absolues. Donc pour obtenir les populations évoluant au cours du temps, on peut prendre la diagonale du vecteur $|\phi_{ik}(t)\rangle \langle \phi_{ik}(t)|$:

$$\text{diag}(|\phi_{ik}(t)\rangle \langle \phi_{ik}(t)|) = (|c_{ik1}|^2, \dots, |c_{ikn}|^2)^T. \quad (91)$$

Finalement, en additionnant les résultats par k et i , on obtient les éléments diagonaux de $\rho(t)$:

$$\text{diag}(\rho(t)) = \sum_{ik} p_i \text{diag}(|\phi_{ik}(t)\rangle \langle \phi_{ik}(t)|), \quad (92)$$

qui représentent les populations dans la base étudiée, en l'occurrence, la base computationnelle.

On pourrait s'intéresser aux populations dans une autre base telle que la base composée des états $|+\rangle$ et $|-\rangle$ (4). Pour ce faire, on utilise la matrice de transformation unitaire T appliquée à chacun de nos états $|\phi_{ik}(t)\rangle$ avant d'effectuer la mesure :

$$T |\phi_{ik}(t)\rangle = TM_k v_i \rightarrow \begin{pmatrix} T & 0 \\ 0 & \mathbb{1} \end{pmatrix} U_{M_k}(v_i^T, 0, \dots, 0)^T. \quad (93)$$

Pour passer de la base computationnelle à la base $\{|+\rangle, |-\rangle\}$, nous utiliserons la matrice de transformation $T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. On peut ensuite effectuer la mesure :

$$\text{diag}(T\rho(t)T^\dagger) = \sum_{ik} p_i \text{diag}\left(T |\phi_{ik}(t)\rangle \langle \phi_{ik}(t)| T^\dagger\right). \quad (94)$$

Les éléments qui ne sont pas sur la diagonale ne peuvent pas être obtenus directement, sans tomographie quantique qui est une méthode pour reconstruire l'état quantique complet d'un système en mesurant différentes observables.

12.2 Analyse de complexité

Dans cette partie, nous analyserons la complexité de l'algorithme [1]. Par complexité, je fais référence au nombre de portes quantiques nécessaires pour réaliser une opération. On va évaluer combien de portes sont requises pour simuler les opérateurs de Kraus M_k pour un système physique donné. Cela dépendra de la dimension du système, c'est-à-dire du nombre d'éléments indépendants nécessaires pour décrire complètement l'état du système. Nous regarderons également l'influence du nombre de qubits sur la quantité de portes nécessaires pour établir l'algorithme qui vient d'être présenté.

Lorsqu'on considère l'équation des opérateurs U_{M_k} , chaque terme contient $\frac{n^2-n}{2}$ éléments non nuls provenant de M_k , n^2 éléments provenant de D_{M_k} et $\frac{n^2-n}{2}$ éléments provenant de $-M_k^\dagger$. Ainsi, le nombre total d'éléments non nuls pour chaque opérateur de Kraus M_k , pour un état préparé donné, est de $2n^2 - n$ [39].

De même, si on considère l'expression de l'évolution de l'état $|\psi_{ik}(t)\rangle = M_k v_i$, le calcul nécessite également $2n^2 - n$ opérations pour chaque opérateur M_k et pour chaque état préparé v_i , incluant toutes les multiplications et additions nécessaires au produit matriciel.

Le nombre maximum d'opérateurs de Kraus pour une seule opération quantique est n^2 , comme indiqué par la décomposition $\rho = \sum_k M_k \rho M_k^\dagger$ [39]. Si l'état mixte ρ est représenté comme une combinaison d'environ n états purs $\rho = \sum_i p_i |\phi_i\rangle \langle \phi_i|$, alors le nombre total d'opérations à effectuer est donné par le produit des opérateurs de Kraus et des états purs. Par conséquent, le nombre total de portes nécessaires est approximativement $2n^5 - n^4$ pour des portes de taille $2n \times 2n$.

La dimension de notre matrice de dilatation U est donnée par $n \cdot m$ où m est le nombre d'opérateurs de Kraus utilisés dans la somme $\sum_k M_k \rho M_k^\dagger$. En général, m est égal à n^2 . Nous devons donc utiliser des matrices U de taille $n^3 \times n^3$.

Dans notre méthode, chaque opérateur de Kraus M_k est supposé être une matrice générale. Le nombre de portes unitaires à deux niveaux nécessaires pour réaliser U_{M_k} est alors polynomial en la dimension du

système n , avec une complexité de $\mathcal{O}(n^2)$. Sur une plateforme conventionnelle basée sur des qubits, où q qubits représentent un système de dimension $n = 2^q$, le nombre de portes unitaires à deux niveaux devient exponentiel en fonction du nombre de qubits, soit $\mathcal{O}(4^q)$. Cette croissance exponentielle est nécessaire si M_k est une matrice générale sans propriétés particulières car l'approximation d'opérateurs unitaires arbitraires est généralement exponentielle.

Cependant, dans les cas où M_k est une matrice creuse, c'est-à-dire qu'elle ne représente que des transitions entre un sous-ensemble de l états quantiques (avec $l \ll n$), l'algorithme peut devenir polynomial en q . Dans ces situations, la dimension effective du système est réduite de n à l pour chaque M_k , et le nombre de portes unitaires à deux niveaux devient $\mathcal{O}(l^2)$, sans dépendance directe sur n ou q . De tels scénarios se produisent fréquemment dans des contextes réalistes, comme dans le cas du canal d'amortissement d'amplitude.

De plus, tous les opérateurs M_k doivent être appliqués à tous les états composés initiaux $|\phi_{ki}\rangle$, mais cette étape peut être effectuée en parallèle, ce qui permet de réduire la profondeur totale du circuit.

Pour une évolution simple, un qubit ancillaire supplémentaire est nécessaire pour doubler la dimension du système, tandis que pour l'évaluation d'observables, deux qubits ancillaires suffisent pour tripler la dimension. Ces qubits ancillaires ne dépendent pas directement de la taille du système ou de la complexité de l'opération quantique, ce qui rend leur utilisation peu coûteuse.

Finalement, la méthode présente une complexité de $\mathcal{O}(n^5)$ dans le cas général, c'est-à-dire lorsqu'aucune hypothèse n'est émise. Cela-dit, elle peut être significativement optimisée lorsque les matrices M_k sont creuses, réduisant ainsi la dépendance au nombre de qubits et rendant l'approche plus adaptée aux simulations de systèmes quantiques ouverts complexes.

13 Utilisation de l'algorithme

Pour trouver le circuit quantique approprié à notre algorithme, il faut trouver comment retranscrire les matrices de dilatation U_{M_k} (88) en terme de circuits. Ces matrices peuvent être décomposées en portes unitaires. Pour ce faire, on utilise un protocole [39] constitué de matrices de rotation. Nous obtenons grâce à ce protocole, des matrices unitaires à deux niveaux. Ensuite, nous pouvons utiliser le protocole donné dans l'Appendix A de l'article [36] pour extraire de ces matrices à deux niveaux, les différents circuits représentant les U_{M_k} . Nous verrons par la suite comment ce processus peut être évité en pratique.

13.1 Obtention des circuits quantiques

Rappelons qu'une *matrice à deux niveaux* est une matrice qui agit uniquement sur un sous-espace de dimension deux au sein d'un espace de dimension plus élevée. Ces matrices sont couramment utilisées dans le contexte des qubits, où chaque qubit est représenté par un espace de Hilbert de dimension deux. Une matrice à deux niveaux peut être construite pour affecter uniquement les deux états de base $|0\rangle$ et $|1\rangle$ d'un qubit, laissant inchangés les autres états d'un système plus grand. Afin de trouver ces matrices, nous utiliserons un protocole donné par [39], comme expliqué précédemment.

13.1.1 Protocole I

La première étape de ce protocole consiste à appliquer une première matrice de rotation U_1 sur la matrice U , dont on veut extraire les matrices à deux niveaux. Le but est d'obtenir un zéro sur l'élément (2,1) de la matrice résultante $U_1.U$. On effectuera ensuite une deuxième rotation U_2 sur cette matrice résultante afin d'obtenir un zéro sur l'élément (3,1) en conservant le zéro obtenu précédemment. On continue de la sorte jusqu'à obtenir des zéros sur toute la première colonne à l'exception de l'élément positionné en (1,1). On s'arrangera pour que cet élément soit égal à un. À la fin de cette étape, nous aurons appliqué $l - 1$ matrices de rotation où la matrice U est de dimension $l \times l$. Nous appellerons la matrice résultante de l'application de toutes les matrices de rotation U_R .

$$U_{l-1}.U_{l-2} \dots U_2.U_1.U = U_R = \begin{pmatrix} 1 & U_R(1,2) & \cdots & U_R(1,l) \\ 0 & U_R(2,2) & \cdots & U_R(2,l) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & U_R(l,2) & \cdots & U_R(l,l) \end{pmatrix} \quad (95)$$

Pour ce faire, nous utiliserons des matrices de Givens $G(i, j, \theta)$ [62] :

$$G(i, j, \theta) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos \theta & \cdots & \sin \theta & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin \theta & \cdots & \cos \theta & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (96)$$

avec $G(k, k, \theta) = 1$ pour $k \neq i, j$, $G(k, k, \theta) = \cos \theta$ pour $k = i, j$ et $G(i, j, \theta) = -G(j, i, \theta) = -\sin \theta$. Notons que le premier cosinus apparaît à la $i^{\text{ème}}$ ligne et que le second apparaît à la $j^{\text{ème}}$ ligne qui ne sont pas forcément juxtaposées. Dans le cas d'une matrice de dilation U_{M_k} (4×4) pour un qubit, un exemple de matrice de Givens est donné par :

$$U_1(\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (97)$$

avec

$$\cos \theta = \frac{U_1(1, 1)}{\sqrt{U_1(1, 1)^2 + U_1(2, 1)^2}}, \quad \sin \theta = \frac{U_1(2, 1)}{\sqrt{U_1(1, 1)^2 + U_1(2, 1)^2}}.$$

Puisque les U_i avec $i \in \{1, \dots, l-1\}$ et U sont unitaires, il s'en suit que $U_{l-1}.U_{l-2} \dots .U_2.U_1.U = U_R$ est unitaire. Pour toute matrice unitaire, chaque ligne et chaque colonne doivent avoir une norme égale à 1. Les éléments de la première ligne doivent donc être nuls puisque l'élément en (1,1) vaut 1. De cette propriété, on peut finalement trouver la dernière matrice U_l par la relation :

$$U_l = (U_{l-1}.U_{l-2} \dots .U_2.U_1.U)^\dagger. \quad (98)$$

Notons que la matrice U_l n'est pas une matrice de Givens. Nous avons que $U_l.U_{l-1}.U_{l-2} \dots .U_2.U_1.U = \mathbb{1}$, autrement-dit, $U = (U_l.U_{l-1}.U_{l-2} \dots .U_2.U_1)^\dagger$. A ce stade, les matrices U_i ne sont pas encore toutes des matrices à deux niveaux. En effet, une fois les quatre matrices obtenues, nous devons vérifier qu'elles soient bien à deux niveaux. Pour le vérifier, il suffit de regarder si la matrice agit sur deux états ou pas. Si la matrice agit sur plus que deux états, ce n'est pas une matrice à deux niveaux.

Formellement, pour convertir une matrice U_i de dimension 4×4 en une matrice V_i de dimension 4×4 avec une structure à deux niveaux, il convient de construire une sous-matrice 3×3 en excluant la première ligne et la première colonne de la matrice U_i . Dès lors, on peut réitérer le protocole pour trouver les sous-matrices 3×3 adéquates.

De manière moins formelle, il se peut qu'il suffise d'échanger deux colonnes afin que la matrice devienne une matrice à deux niveaux. Permuter les colonnes d'une matrice de transformation sur deux qubits revient à appliquer une permutation sur la base de calcul des états qubits. Cela correspond à réarranger l'ordre des sorties possibles après l'application de la transformation, ce qui modifie la manière dont les états de sortie sont interprétés, mais sans changer les valeurs dans chaque colonne.

Par exemple, pour la matrice U_4 ci-dessous, il suffit de permuter les lignes une et quatre afin d'obtenir une matrice à deux niveaux. En effectuant cette transformation, on utilise une matrice à deux niveaux (V_1) appliquée sur une matrice qui n'est pas à deux niveaux (U_4) pour obtenir une matrice à deux niveaux (V_1). En d'autres mots, grâce à ce qui a été expliqué précédemment, nous pouvons représenter une matrice unitaire par différentes matrices à deux niveaux.

$$V_2 = V_1.U_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{1-e^{-\gamma t}} & \sqrt{e^{-\gamma t}} \\ 0 & 0 & -\sqrt{e^{-\gamma t}} & -\sqrt{1-e^{-\gamma t}} \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (99)$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\sqrt{e^{-\gamma t}} & -\sqrt{1-e^{-\gamma t}} \\ 0 & 0 & -\sqrt{1-e^{-\gamma t}} & \sqrt{e^{-\gamma t}} \end{pmatrix}. \quad (100)$$

Une fois cela effectué, nous avons trouvé toutes les matrices à deux niveaux nécessaires pour représenter la matrice U_{M_k} . Il ne reste plus qu'à trouver le circuit correspondant.

13.1.2 Protocole II

Pour trouver le circuit correspondant aux matrices à deux niveaux, nous utiliserons un théorème énoncé dans la référence [36] suivi d'un protocole permettant d'identifier les opérateurs. Le théorème est le suivant :

Pour tout opérateur unitaire U_{AB} , il est possible de l'écrire comme

$$U_{AB} = U_A \otimes U_B U_d V_A \otimes V_B \quad (101)$$

avec U_A, U_B, V_A, V_B des opérateurs locaux et U_d , un opérateur unitaire diagonal non local.

Un **opérateur local** agit uniquement sur un sous-système (comme A ou B) sans affecter l'autre. Par exemple, U_A et V_A agissent uniquement sur A , tandis que U_B et V_B agissent uniquement sur B . Un **opérateur non local**, comme U_d , agit conjointement sur les deux sous-systèmes A et B , introduisant une corrélation ou une interaction entre eux.

Il est important de noter que toute mesure de l'intrication n'est pas affectée par des opérateurs unitaires locaux. Par conséquent, l'intrication engendrée par U_{AB} reste inchangée lorsqu'elle est remplacée par $U_d V_A \otimes V_B$. Cela signifie que la quantité maximale d'intrication pouvant être obtenue en appliquant un opérateur unitaire général U_{AB} est identique à celle générée par U_d . Nous allons maintenant expliquer le protocole donné par [36] afin d'établir la relation 101.

Nous travaillerons dans la base magique pour faciliter les calculs. La matrice de changement de base est :

$$M = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{pmatrix}$$

qui peut être utilisée par la relation $U_b = M^\dagger U M$.

La première étape consiste à trouver les valeurs propres $e^{2i\epsilon_k}$ et vecteurs propres $|\Psi_k\rangle$ de l'opérateur unitaire symétrique ($U^T U$).

Avant de passer à **la deuxième étape**, expliquons ce qu'est un vecteur maximalement intriqué. C'est un état quantique qui présente le plus haut degré possible d'intrication entre les deux sous-systèmes. C'est-à-dire lorsque les mesures effectuées sur l'un des sous-systèmes sont parfaitement corrélées avec les mesures effectuées sur l'autre sous-système, de manière à ce que la connaissance de l'état de l'un des sous-systèmes détermine complètement l'état de l'autre, même si ces deux sous-systèmes sont spatialement séparés. Dès lors, **la deuxième étape** suit le théorème :

Pour toute base maximalement intriquée $|\Psi_k\rangle$, il existe des phases ξ_k et des opérateurs unitaires locaux V_A et V_B tels que :

$$V_A \otimes V_B e^{i\xi_k} |\Psi_k\rangle = |\Phi_k\rangle \quad (102)$$

Comme nous en avons déjà parlé dans la section 11.3, un état est maximalelement intriqué s'il présente le plus haut degré d'intrication possible entre ses sous-systèmes. L'état réduit de chaque sous-système est totalement mixte, c'est-à-dire qu'il contient un maximum d'incertitude. On obtient un état réduit en prenant la trace partielle du système. Les états de Bell sont un bon exemple de base maximalelement intriquée.

Le but de cette étape consiste d'abord de trouver des états $|e\rangle$ et $|f\rangle$ ainsi qu'une phase $e^{i\delta}$ pour ensuite définir les matrices V_A et V_B . Une fois cela fait, nous pourrons utiliser la relation (102) pour lier nos vecteurs propres $|\Psi_k\rangle$ aux vecteurs $|\Phi_k\rangle$. Comme on l'a vu, la première étape nous permet de trouver nos vecteurs propres $|\Psi_k\rangle$. Ensuite, il est toujours possible de les écrire en fonction de vecteurs réels dans la base magique que nous définirons comme $\{|\bar{\Psi}_k\rangle\}$. Cela peut être réalisé par la relation :

$$|\Psi_k\rangle = e^{\gamma_k} |\bar{\Psi}_k\rangle \quad (103)$$

où e^{γ_k} est une phase complexe et $|\bar{\Psi}_k\rangle$ est un état réel dans la base magique. Une fois que les $|\bar{\Psi}_k\rangle$ sont trouvés, nous considérons deux états différents $|\bar{\Psi}_i\rangle$ et $|\bar{\Psi}_j\rangle$. Ensuite, grâce à ces états, nous définissons les états $|e, f\rangle = |e\rangle \otimes |f\rangle$ et $|e^\perp, f^\perp\rangle = |e^\perp\rangle \otimes |f^\perp\rangle$ tels que

$$\begin{cases} |e, f\rangle &= \frac{1}{\sqrt{2}}(|\bar{\Psi}_i\rangle - i|\bar{\Psi}_j\rangle) \\ |e^\perp, f^\perp\rangle &= \frac{1}{\sqrt{2}}(|\bar{\Psi}_i\rangle + i|\bar{\Psi}_j\rangle), \end{cases} \quad (104)$$

où $|e\rangle, |e^\perp\rangle \in \mathcal{H}_A$ et où $|f\rangle, |f^\perp\rangle \in \mathcal{H}_B$ avec \mathcal{H}_A et \mathcal{H}_B , respectivement les Hamiltoniens des sous espaces A et B. Nous avons construit les états $|e, f\rangle$ et $|e^\perp, f^\perp\rangle$ tels qu'ils doivent être orthogonaux par définition. Ceci implique immédiatement que ces vecteurs doivent donner la décomposition de Schmidt. Il est possible de retrouver cette décomposition en inversant le système (104). Donnons-la pour les deux premiers vecteurs $|\bar{\Psi}_1\rangle$ et $|\bar{\Psi}_2\rangle$:

$$\begin{cases} |\bar{\Psi}_1\rangle &= \frac{1}{\sqrt{2}}(|e, f\rangle + |e^\perp, f^\perp\rangle) \\ |\bar{\Psi}_2\rangle &= \frac{-i}{\sqrt{2}}(|e, f\rangle - |e^\perp, f^\perp\rangle) \end{cases} \quad (105)$$

De manière équivalente, on peut trouver les vecteurs $|\bar{\Psi}_3\rangle$ et $|\bar{\Psi}_4\rangle$:

$$\begin{cases} |\bar{\Psi}_3\rangle &= \frac{-i}{\sqrt{2}}(e^{i\delta}|e, f^\perp\rangle + e^{-i\delta}|e^\perp, f\rangle) \\ |\bar{\Psi}_4\rangle &= \frac{1}{\sqrt{2}}(e^{i\delta}|e, f^\perp\rangle - e^{-i\delta}|e^\perp, f\rangle) \end{cases} \quad (106)$$

On peut trouver les états $|e\rangle, |e^\perp\rangle, |f\rangle$ et $|f^\perp\rangle$ avec ces équations. Une fois cela effectué, on peut définir les matrices V_A et V_B :

$$V_A = |0\rangle\langle e| + |1\rangle\langle e^\perp| e^{i\delta}, \quad V_B = |0\rangle\langle f| + |1\rangle\langle f^\perp| e^{-i\delta} \quad (107)$$

Dès lors, avec les phases ξ_k appropriées, nous trouvons les équations (102).

La troisième étape du protocole consiste à trouver les ϵ_k tels que :

$$|\tilde{\Psi}_k\rangle = e^{-i\epsilon_k} U |\Psi_k\rangle. \quad (108)$$

Notons que l'ensemble d'états $|\tilde{\Psi}\rangle$ est maximalelement intriqué.

La quatrième étape du protocole consiste à trouver les valeurs propres $e^{i\lambda_k}$ de la matrice U_d de la relation (101) ainsi que les matrices unitaires U_A et U_B telles que

$$U_A^\dagger \otimes U_B^\dagger e^{i(\lambda_k + \xi_k + \epsilon_k)} |\tilde{\Psi}_k\rangle = |\Phi_k\rangle. \quad (109)$$

Une fois ces 4 étapes effectuées, nous avons trouvés les opérateurs unitaires U_A, U_B, V_A, V_B et U_d qui nous permettent de décrire la matrice U_{AB} trouvée grâce au protocole précédent.

13.2 Utilisation des circuits

Qiskit met à disposition tout un ensemble de fonctions facilitant l'utilisation des circuits quantiques. Dans les fonctions qu'offre Qiskit, une fonction très utile permet de générer automatiquement un circuit quantique à partir d'une matrice unitaire. Il suffit d'utiliser la méthode `.unitary`. Cette méthode prend en entrée une matrice unitaire et retourne un circuit équivalent réalisant la transformation définie par cette matrice. Cela permet d'éviter la spécification manuelle des portes quantiques, ce qui peut être complexe, en particulier pour des opérations unitaires de grande taille.

Dans ce mémoire, nous utiliserons cette fonction pour simplifier l'implémentation de certains circuits, tout en assurant que le circuit final est correctement configuré pour effectuer la transformation souhaitée. En cas de besoin de contrôle plus précis sur les portes quantiques utilisées, Qiskit propose aussi la possibilité de concevoir le circuit manuellement. On peut dès lors utiliser le protocole décrit dans la section précédente pour créer notre propre circuit. Ainsi, en fonction des exigences de chaque cas, le choix pourra être fait entre une implémentation automatique, via `.unitary`, ou manuelle, pour un contrôle total sur le protocole d'implémentation.

La simulation d'un circuit quantique dans Qiskit se déroule en plusieurs étapes essentielles, permettant de reproduire les effets d'un ordinateur quantique. Vous trouverez en annexe un exemple de code illustrant les explications qui vont suivre. Tout d'abord, il est nécessaire de concevoir le circuit en ajoutant les différentes portes quantiques. Cela peut être fait soit en construisant manuellement le circuit porte par porte, soit en utilisant la méthode `.unitary` de Qiskit pour générer directement un circuit correspondant à une matrice unitaire donnée.

Une fois le circuit créé, il est conseillé de le passer par une phase de transpilation. La transpilation consiste à optimiser le circuit afin de réduire le nombre de portes et d'adapter les opérations aux contraintes spécifiques du simulateur ou du matériel quantique. En utilisant la fonction `transpile` de Qiskit, on obtient un circuit optimisé pour exécuter la transformation voulue avec une meilleure efficacité. Cette optimisation peut être réalisée à plusieurs niveaux, allant des optimisations générales (par exemple, réduire le nombre de portes globales) à des optimisations spécifiques liées aux caractéristiques du matériel ou du simulateur cible. Selon le niveau choisi, allant de 0 à 3, Qiskit ajuste le circuit pour maximiser sa performance en fonction des ressources disponibles, ce qui permet de réduire la durée d'exécution et le bruit dans le cas d'un matériel quantique réel. Nous utiliserons l'optimisation de niveau trois suivant cette fonction.

Nous simulerons le circuit de deux manières différentes. La première manière consiste à utiliser un simulateur quantique intégré dans Qiskit et permettant de reproduire fidèlement les effets d'un ordinateur quantique. La deuxième manière consiste à utiliser un hardware réel, c'est-à-dire un ordinateur quantique qui est mis à disposition par IBM.

Enfin, une fois le circuit exécuté, les données obtenues sont récupérées et analysées. Les résultats de mesure permettent alors de visualiser les distributions de probabilité des états finaux des qubits. En traçant ces données sous forme de graphique, on peut interpréter les résultats de la simulation et valider les propriétés du circuit conçu. Ces étapes offrent ainsi une méthode complète et cohérente pour simuler un circuit quantique, permettant de vérifier théoriquement ses performances avant une éventuelle exécution sur un ordinateur quantique réel.

Il est possible d'utiliser un ordinateur quantique réel directement via Qiskit. En effet, l'utilisateur doit d'abord se connecter à IBM Quantum via un compte IBM. Il est possible de procéder à cette identification directement sur le code python. En effet, l'accès aux backends matériels d'IBM est réalisable en important une clé d'API dans le script Qiskit, ce qui permet à l'utilisateur de se connecter aux différents ordinateurs quantiques disponibles.

Pour exécuter le circuit sur un backend quantique réel, il suffit de remplacer le simulateur par un backend IBM Quantum (comme `ibm_brisbane`). Qiskit offre une variété de backends physiques, chacun ayant ses propres caractéristiques (nombre de qubits, taux de bruit, et temps de cohérence). Une fois le backend choisi, le circuit est soumis à la même phase de transpilation, mais avec des optimisations spécifiques au matériel sélectionné. Cette transpilation est nécessaire pour adapter le circuit aux contraintes du processeur quantique choisi, maximisant ainsi la probabilité d'obtenir des résultats fiables.

En procédant ainsi, nous pouvons comparer les résultats obtenus sur le simulateur et sur le matériel quantique réel, évaluant ainsi les impacts du bruit et des erreurs inhérentes aux ordinateurs quantiques actuels. Bien que nous sachions représenter une matrice unitaire sous forme de circuit, il est important de savoir préparer les états initiaux qui influencent la dynamique du système. Passons à l'explication de la préparation de ces états.

14 Préparation des états

Avant d'appliquer un circuit quantique, il est essentiel de préparer les états des qubits de manière adéquate car l'état initial détermine le succès de l'algorithme. Dans notre travail, nous utiliserons deux qubits, ce qui nous permet de générer des états de superposition et d'intrication en fonction des besoins du circuit.

14.1 État initial des qubits

Par défaut, les qubits sont initialisés dans l'état de base $|0\rangle$ avant l'application de toute opération quantique. Cet état $|0\rangle$ est un état propre de la porte Pauli-Z, ce qui signifie que la mesure de chaque qubit dans la base $\{|0\rangle, |1\rangle\}$ donnerait toujours $|0\rangle$ pour un qubit non modifié.

Lorsqu'on souhaite travailler avec un qubit dans l'état $|1\rangle$, il suffit d'appliquer une porte X sur le qubit en question pour qu'il passe de l'état $|0\rangle$ à l'état $|1\rangle$ puisque les états sont initialisés dans l'état de base $|0\rangle$. Dans les exemples que nous présenterons par la suite, nous préparerons l'état v_1 dans l'état $|1\rangle$ comme mentionné ci-dessus.

Pour des applications spécifiques, il est souvent nécessaire de préparer des états initiaux autres que $|0\rangle$, notamment des états de superposition ou des états intriqués. Cela se fait en appliquant une série de portes quantiques appropriées pour transformer les états de base $|0\rangle$ en des états plus complexes.

14.2 Préparation des états superposés

Pour obtenir une superposition sur un ou plusieurs qubits, la porte de Hadamard est généralement utilisée. Rappelons qu'un *état superposé* d'un qubit est une combinaison linéaire des états de base $|0\rangle$ et $|1\rangle$. Par exemple, pour créer une superposition sur un seul qubit, on applique la porte de Hadamard H à l'état $|0\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle. \quad (110)$$

Dans les exemples que nous présenterons par la suite, nous préparerons l'état v_2 dans l'état superposé $|+\rangle$. En appliquant une porte de Hadamard à chaque qubit de deux qubits initialisés à $|0\rangle$, on obtient un état de superposition à poids égaux de tous les états possibles :

$$H|0\rangle \otimes H|0\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle).$$

Cet état, souvent appelé un état de superposition maximale, est fréquemment utilisé dans les algorithmes quantiques pour explorer simultanément tous les états de la base computationnelle.

14.3 Préparation des états intriqués

Pour certains algorithmes, il est nécessaire de préparer des états intriqués. Rappelons qu'un état intriqué concerne au moins deux qubits dont les états sont corrélés de manière telle que l'état de l'un dépend de l'état de l'autre, peu importe la distance qui les sépare. Par exemple, dans un état intriqué $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, si l'un des qubits est mesuré en $|0\rangle$, l'autre sera automatiquement en $|0\rangle$ aussi.

L'*état de Bell* est un exemple d'état intriqué souvent utilisé dans les circuits à deux qubits :

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (111)$$

Il peut être généré en appliquant une porte Hadamard au premier qubit, suivie d'une porte CNOT avec le premier qubit comme contrôle et le second qubit comme cible (Fig. 11).

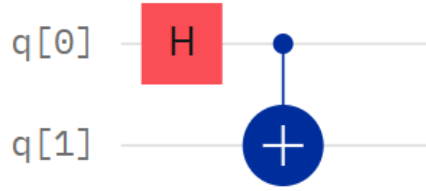


FIGURE 11 – Représentation des portes utilisées pour préparer l'état de Bell donné dans l'équation (111).

L'application de la porte Hadamard sur le premier qubit donne :

$$H |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \quad (112)$$

En prenant en compte le deuxième qubit, cela donne :

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle). \quad (113)$$

Ensuite, l'application de la porte CNOT est exprimée comme :

$$CNOT \left(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \right) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle. \quad (114)$$

Nous retrouvons bien l'état de Bell défini en (111).

14.4 Contrôle des états avec des portes de rotation

Enfin, il est parfois nécessaire d'introduire des rotations contrôlées pour ajuster précisément l'amplitude et la phase des qubits. Pour cela, on peut utiliser des portes de rotation. Par exemple, pour introduire une phase θ sur un qubit dans l'état $|+\rangle$ (4), on utilise une rotation $RZ(\theta)$:

$$RZ(\theta) |+\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\theta} |1\rangle \right)$$

En résumé, la préparation d'états est une étape primordiale pour l'implémentation de circuits quantiques. En utilisant des portes Hadamard H , $CNOT$ et de rotation, il est possible de générer des superpositions, des intrications et des états avec des phases contrôlées qui serviront de base pour l'exécution des algorithmes présentés dans la section précédente. Maintenant que nous savons comment préparer les états, il ne reste plus qu'à illustrer tout ce qui a été vu jusqu'à présent par des applications. Nous discuterons ensuite des résultats.

15 Applications

Tout au long de cette partie, nous avons présenté un algorithme et nous avons expliqué comment l'implémenter de manière concrète sur de l'hardware. Il est temps à présent d'illustrer ces explications par les trois exemples qui ont été expliqués à la partie II.

15.1 Canal bit-flip

Pour l'exemple du canal bit-flip (Eq. (116) et (65)), nous allons préparer l'état initial $|\phi(0)\rangle = |1\rangle$. Pour ce faire, on utilise une porte X . La matrice densité associée à cet état initial est $\rho_0 = |\phi(0)\rangle \langle \phi(0)| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. Nous travaillons dans l'espace dilaté, c'est pourquoi notre état initial devient :

$$|\phi(0)\rangle = |1\rangle \rightarrow v_1 = (0, 1, 0, 0)^T \quad (115)$$

(90), où v_1 est le vecteur qui représente l'état $|\phi(0)\rangle$ dans l'espace dilaté.

Rappelons que ce phénomène peut être modélisé par l'équation maîtresse (116) :

$$\dot{\rho}(t) = \gamma (\sigma_x \rho(t) \sigma_x - \rho(t)), \quad (116)$$

avec γ le taux d'émission spontanée et σ_x l'opérateur de Pauli-X.

Une fois que l'état initial est préparé, il faut le propager par les matrices unitaires dilatées (90). Pour les construire, rappelons les opérateurs de Kraus :

$$M_0 = \sqrt{p} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad M_1 = \sqrt{1-p} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (117)$$

Ensuite, nous pouvons trouver l'opérateur de défaut $D_{M_k} = \sqrt{\mathbb{1} - M_k^\dagger M_k}$ pour M_0 et M_1 . Dès lors, les matrices de dilatation, via l'équation (88) sont

$$U_{M_0} = \begin{pmatrix} a & 0 & b & 0 \\ 0 & a & 0 & b \\ b & 0 & -a & 0 \\ 0 & b & 0 & -a \end{pmatrix}, \quad (118)$$

$$U_{M_1} = \begin{pmatrix} 0 & b & a & 0 \\ b & 0 & 0 & a \\ a & 0 & 0 & -b \\ 0 & a & -b & 0 \end{pmatrix}, \quad (119)$$

où $a = \sqrt{e^{-\gamma t}}$ et $b = \sqrt{1 - e^{-\gamma t}}$. Nous obtenons ainsi deux matrices 4×4 dépendantes du temps $U_{M_1} v_1$ et $U_{M_2} v_2$. Il suffit dès lors d'utiliser la fonction `.unitary` de Qiskit pour en trouver le circuit associé. Un exemple qui illustre le circuit pour $U_{M_1} v_1$ du canal bit-flip, au temps $t = 500ps$, est illustré dans la figure 12. Remarquons que les portes de ce circuit diffèrent des portes énoncées au préalable dans le théorème (101). En effet, IBM ne peut pas exprimer toutes les portes possibles mais seulement une partie de celles-ci. Dès lors, IBM possède une liste de portes qui sont utilisées pour créer le circuit, ce qui est équivalent au théorème (101), sous une autre forme. De plus, IBM optimise les portes utilisées pour que l'implémentation du système quantique ouvert sur l'hardware fonctionne au mieux, ce qui modifie également la structure des portes.

De là, grâce à la mesure, nous obtenons les populations des états $|0\rangle$ et $|1\rangle$. Nous avons d'abord calculé théoriquement les populations du canal bit-flip pour t de 0 à 1000ps pour un pas de temps de 10ps. Puis nous avons simulé via AerSimulator proposé par Qiskit. Enfin, nous avons simulé le système sur l'ordinateur quantique d'IBM Brisbane. Nous avons utilisés 1024 shots pour le simulateur et pour l'hardware, c'est-à-dire 1024 réalisations du circuit. En effet, à cause du caractère aléatoire de la mesure en physique quantique, il est toujours nécessaire de moyenner les résultats sur plusieurs réalisations d'un circuit. De là, un nombre insuffisant de shots ne permettrait pas de reproduire la description de l'équation maîtresse. Nous avons représenté les résultats sur le graphique (13). Nous avons choisi $\gamma = 1.52 \cdot 10^9 s^{-1}$ comme dans la référence [1].

Ensuite, nous avons procédé à un changement de base grâce à l'équation (93). Nous avons ainsi obtenu les populations des états $|+\rangle$ et $|-\rangle$ (Fig. 13). Nous pouvons constater que dans cette base, il n'y a aucun changement des populations via le bit-flip.

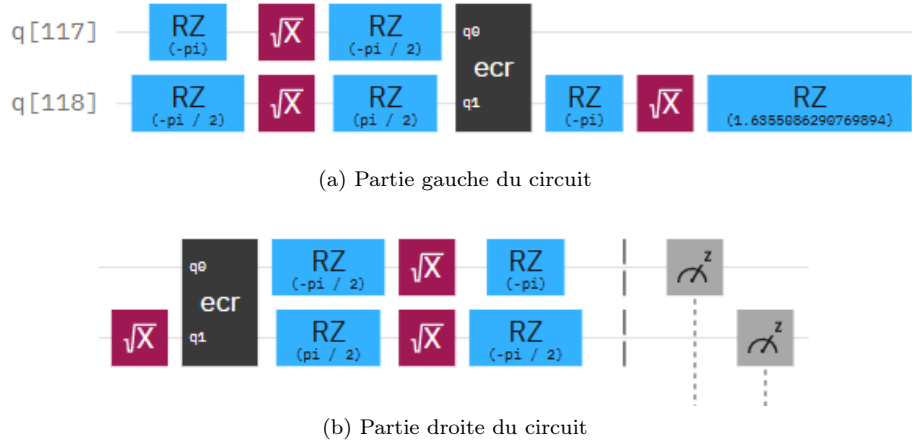


FIGURE 12 – Illustration du circuit pour le canal bit-flip en $t = 500ps$. La porte \sqrt{X} est la porte "square-root NOT gate", elle implémente la racine carrée de la porte X et reproduit ses effets lorsqu'elle est appliquée deux fois à la suite. La porte ecr est une porte de résonance croisée en écho. Elle sert à créer des états intriqués et à atténuer certains termes indésirables dus aux erreurs [63]. Les portes grisées sont les mesures effectuées sur les qubits.

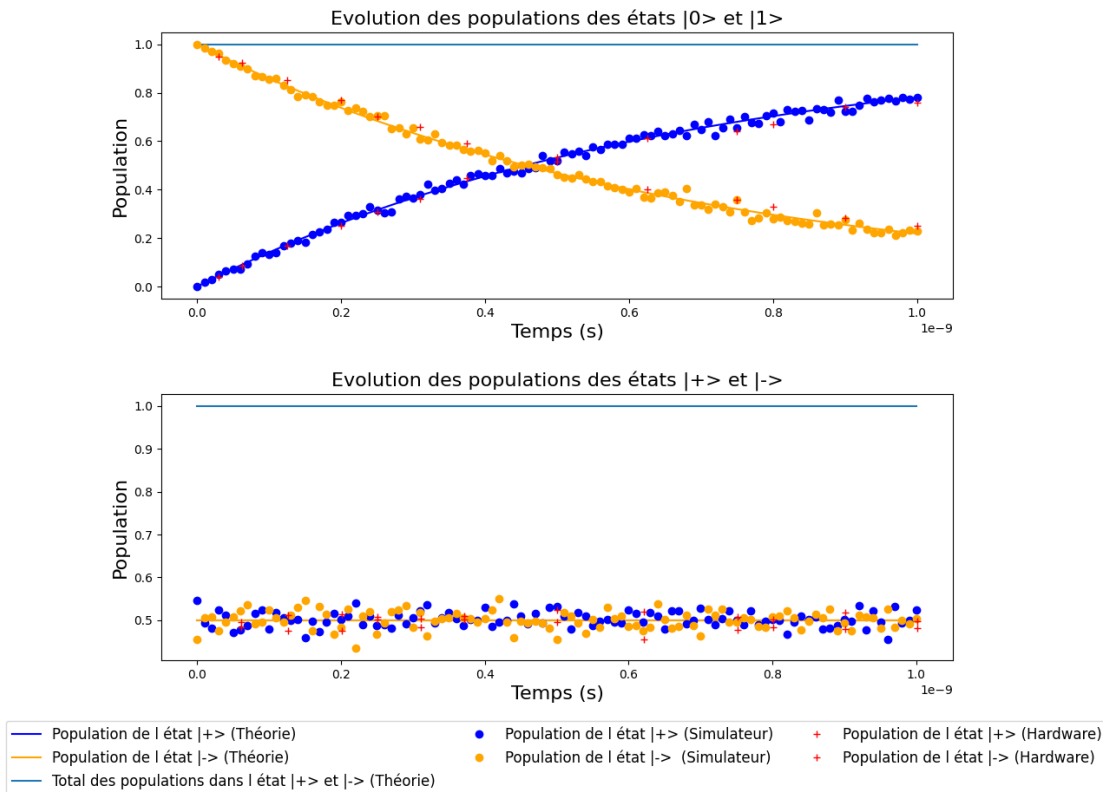


FIGURE 13 – Représentation de l'évolution temporelle des populations de l'état fondamental $|0\rangle$ et de l'état excité $|1\rangle$ (haut) et de l'état $|+\rangle$ et de l'état $|-\rangle$ (bas) pour le canal bit-flip. Les lignes continues sont obtenues par des calculs numériques classiques des vecteurs de sortie. Les points sont obtenus par le simulateur IBM Qiskit (AerSimulator). Les croix sont obtenus par le dispositif IBM Quantum Brisbane. Pour chaque point, nous avons utilisés 1024 shots pour le simulateur et pour l'hardware, c'est-à-dire 1024 réalisations du circuit.

Remarquons que les résultats sur l'hardware, représentés par les croix rouges sur le graphique, reproduisent très bien ce que prédit l'équation maîtresse. Cela dit, il faut préciser que ce n'est possible que grâce à un nombre de shots suffisamment élevé et grâce à un nombre de qubits suffisamment faible.

15.2 Canal d'amortissement d'amplitude modifié

Pour l'exemple du canal d'amortissement d'amplitude modifié (Eq. (68) et (82)), nous allons préparer les états initiaux $|\phi_1(0)\rangle = |1\rangle$ et $|\phi_2(0)\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Pour ce faire, on utilise une porte X pour préparer $|\phi_1(0)\rangle$ et une porte Hadamard H pour préparer $|\phi_2(0)\rangle$. Nous assumerons une composition physique telle que $\rho_0 = |\phi(0)\rangle\langle\phi(0)| = \frac{1}{2}(|1\rangle\langle 1| + |+\rangle\langle +|)$. La matrice densité associée à ces états initiaux est donc

$$\rho_0 = \frac{1}{4} \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}. \quad (120)$$

Les vecteurs v_1 et v_2 représentent respectivement les états $|1\rangle$ et $|+\rangle$ dans l'espace de dilatation, c'est-à-dire l'état de base du système pour lesquels nous appliquons les matrices de transformations unitaires U_{Mk} en fonction du temps. Nous travaillons dans l'espace dilaté, c'est pourquoi nos états initiaux deviennent (90) :

$$|\phi_1(0)\rangle = |1\rangle \rightarrow v_1 = (0, 1, 0, 0)^T, \quad (121)$$

$$|\phi_2(0)\rangle = |+\rangle \rightarrow v_2 = \frac{1}{\sqrt{2}}(1, 1, 0, 0)^T. \quad (122)$$

Rappelons que ce phénomène peut être modélisé par l'équation maîtresse (78) :

$$\dot{\rho}(t) = -i[H, \rho(t)] + \gamma \left(\sigma_+ \rho(t) \sigma_- - \frac{1}{2} \{ \sigma_- \sigma_+, \rho(t) \} \right), \quad (123)$$

avec γ le taux d'émission spontanée et σ_+ et σ_- les opérateurs de Pauli correspondant respectivement à la transition du niveau excité vers le niveau inférieur et inversement.

Une fois que les états initiaux sont préparés, il faut les propager par les matrices unitaires dilatées (90). Pour les construire, rappelons les opérateurs de Kraus (76) :

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{e^{-\gamma t}} \end{pmatrix}, M_1 = \begin{pmatrix} 0 & \sqrt{1 - e^{-\gamma t}} \\ 0 & 0 \end{pmatrix}. \quad (124)$$

Ensuite, nous pouvons trouver les opérateurs de défaut $D_{Mk} = \sqrt{I - M_k^\dagger M_k}$ pour M_0 et M_1 , c'est-à-dire :

$$D_{M0} = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{1 - e^{-\gamma t}} \end{pmatrix}, D_{M1} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{e^{-\gamma t}} \end{pmatrix}. \quad (125)$$

Dès lors, les matrices de dilatation, via l'équation (88) sont

$$U_{M0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{e^{-\gamma t}} & 0 & \sqrt{1 - e^{-\gamma t}} \\ 0 & 0 & -1 & 0 \\ 0 & \sqrt{1 - e^{-\gamma t}} & 0 & -\sqrt{e^{-\gamma t}} \end{pmatrix}, \quad (126)$$

$$U_{M1} = \begin{pmatrix} 0 & \sqrt{1 - e^{-\gamma t}} & \sqrt{e^{-\gamma t}} & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & \sqrt{e^{-\gamma t}} & -\sqrt{1 - e^{-\gamma t}} & 0 \end{pmatrix}. \quad (127)$$

Grâce à ces matrices de dilatation, nous pouvons simuler notre circuit. C'est ce que nous avons fait pour ce système, dans les mêmes conditions que pour le canal bit-flip. La seule différence est le nombre de circuit car puisqu'on applique séparément chaque matrice de transformation sur chaque vecteur, nous nous retrouvons avec 4 circuits distincts $U_{M0}v_1$, $U_{M0}v_2$, $U_{M1}v_1$ et $U_{M1}v_2$. Nous effectuons ensuite des mesures

sur les populations de chaque qubits à la fin de chaque circuit. Pour avoir la population totale, il suffit d'additionner les populations obtenues pour chaque circuit.

Les résultats sont illustrés sur la figure 14. Ensuite, nous avons procédé au même changement de base (93) que précédemment. Nous avons ainsi obtenu les populations des états $|+\rangle$ et $|-\rangle$ (Fig. 14).

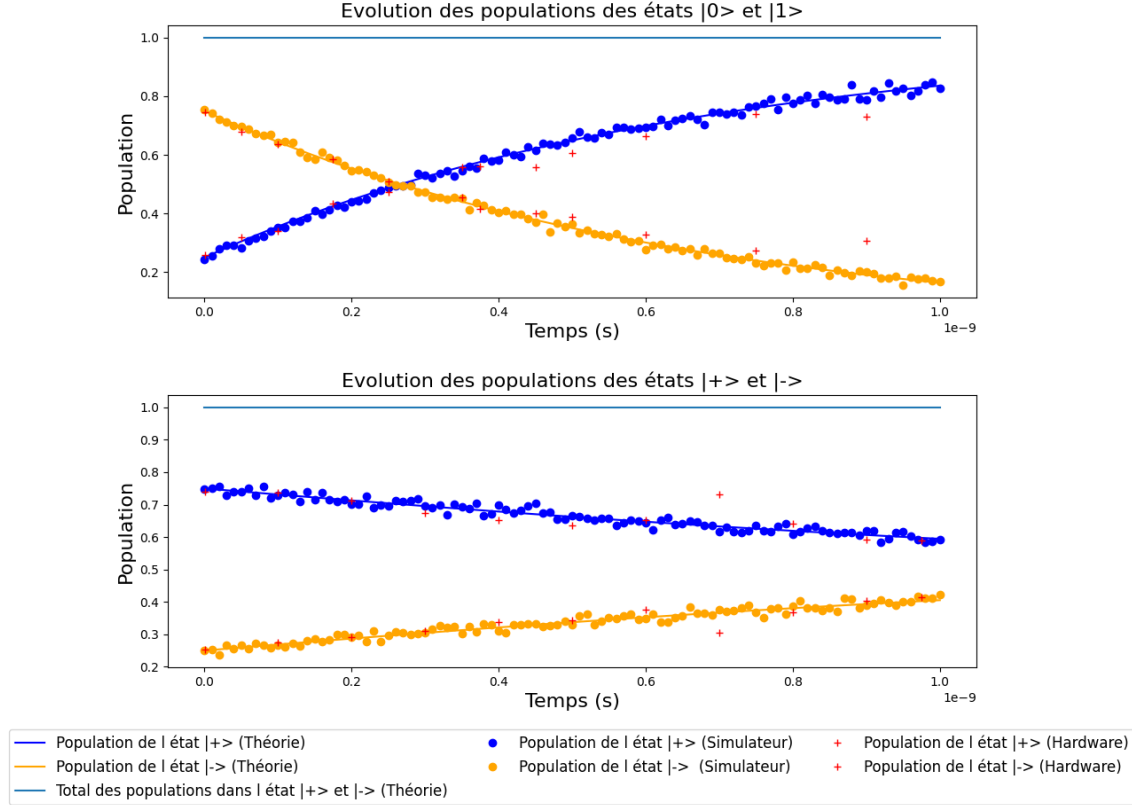


FIGURE 14 – Représentation de l'évolution temporelle des populations de l'état fondamental $|0\rangle$ et de l'état excité $|1\rangle$ (haut) et de l'état $|+\rangle$ et de l'état $|-\rangle$ (bas) pour le canal d'amortissement d'amplitude modifié. Les lignes continues sont obtenues par des calculs numériques classiques des vecteurs de sortie. Les points sont obtenus par le simulateur IBM Qiskit (AerSimulator). Les croix sont obtenus par le dispositif IBM Quantum Brisbane. Pour chaque point, nous avons utilisés 1024 shots pour le simulateur et pour l'hardware, c'est-à-dire 1024 réalisations du circuit.

15.3 Canal à deux qubits

Pour l'exemple du canal à deux qubits (Eq. (84)), nous allons préparer, de la même manière que pour le canal bit-flip, l'état initial $|\phi(0)\rangle = |1\rangle$. Pour ce faire, nous utilisons une porte X . Nous travaillons dans l'espace dilaté, c'est pourquoi notre état initial devient, par l'équation (90),

$$|\phi(0)\rangle = |1\rangle \rightarrow v_1 = (0, 1, 0, 0)^T, \quad (128)$$

où v_1 est le vecteur qui représente l'état $|\phi(0)\rangle$ dans l'espace dilaté.

Une fois que l'état initial est préparé, il faut le propager par les matrices unitaires dilatées (90). Nous pouvons les construire avec les opérateurs de Kraus (84). Ensuite, nous pouvons utiliser la fonction `.unitary` de Qiskit. Cependant, cette fonction nécessite des matrices purement unitaires. Or, les erreurs numériques des calculs rendent les matrices très légèrement non-unitaires. Pour surpasser ce problème, on peut forcer

l'unitarité des matrices en utilisant la décomposition en valeurs singulières. Cette méthode consiste à factoriser une matrice en une matrice unitaire de vecteurs propres à gauche, une matrice diagonale de valeurs singulières et une matrice unitaire de vecteurs propres à droite. Dans ce contexte, le terme valeur singulière exprime des scalaires non-négatifs représentant les amplitudes des transformations opérées par la matrice d'origine. Ceci permet de simuler le système via le simulateur AerSimulator. Nous avons donc présenté la simulation de l'évolution des états fondamental et excité au cours du temps sur la figure (15). Ce simulateur, étant purement logiciel, peut simuler n'importe quelle porte quantique unitaire.

Nous avons lancé le programme sur l'hardware IBM Kyiv et sur l'hardware IBM Brisbane. Ceux-ci ont renvoyé une erreur et n'ont pas pu simuler le canal à deux qubits. Cela reflète qu'IBM peut-être considéré comme une boîte noire pour laquelle on ne connaît pas tout. Effectivement, l'erreur renvoyée n'est pas expliquée. On pourrait imaginer que l'erreur proviendrait du fait qu'un ordinateur quantique a des limitations dues aux lois physiques, à la connectivité entre les qubits et aux technologies de fabrication. En effet, dans ce cas nous utilisons les ordinateurs quantiques d'IBM pour lesquels, seules quelques portes unitaires sont définies [16]. Dès lors, la matrice n'a pas pu être décomposée en portes listées par IBM et le circuit n'a donc pas pu être lancé sur l'hardware. Le simulateur AerSimulator a par contre réussi à décomposer la matrice unitaire en portes quantiques car il n'est pas limité par une liste préétablie.

De là, on pourrait voir une piste d'amélioration de l'algorithme en proposant une décomposition de la matrice unitaire en portes disponibles sur IBM afin d'avoir plus de chance de pouvoir l'implémenter sur de l'hardware.

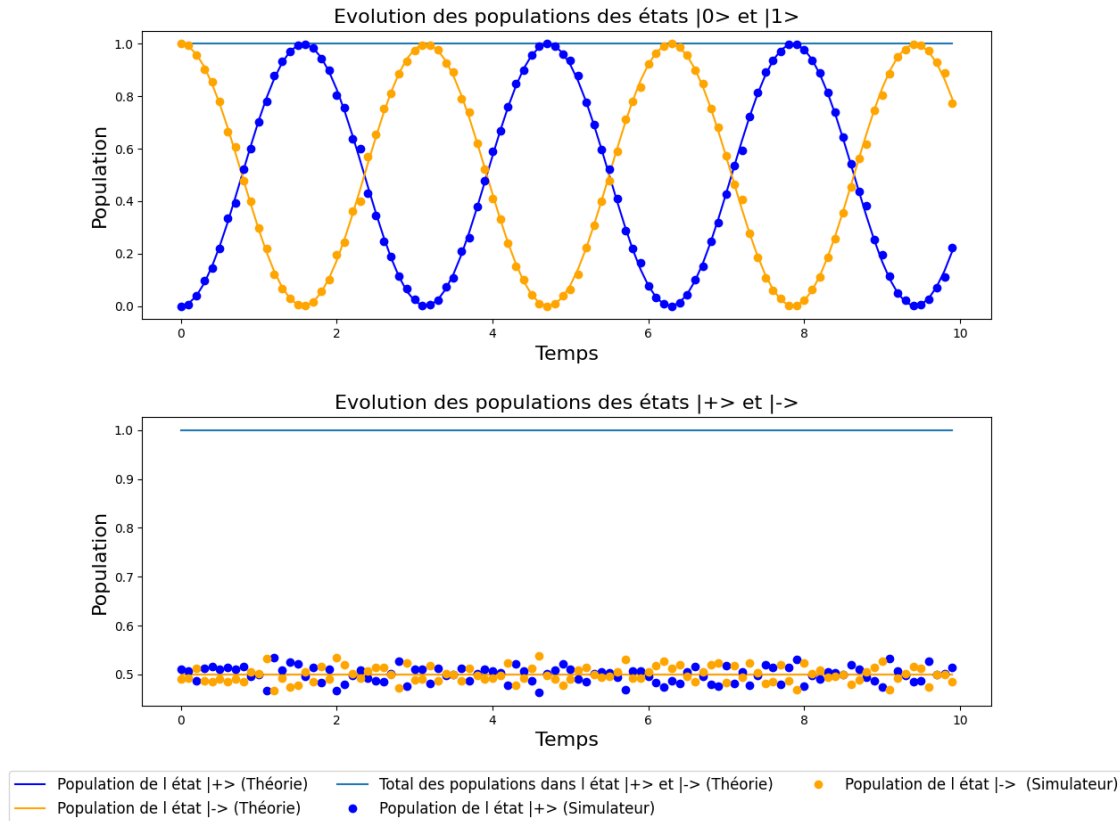


FIGURE 15 – Représentation de l'évolution temporelle des populations de l'état fondamental $|0\rangle$ et de l'état excité $|1\rangle$ (haut) et de l'état $|+\rangle$ et de l'état $|-\rangle$ (bas) pour le canal à deux qubits. Les lignes continues sont obtenues par des calculs numériques classiques des vecteurs de sortie. Les points sont obtenus par le simulateur IBM Qiskit (AerSimulator). Pour chaque point, nous avons utilisés 1024 shots pour le simulateur, c'est-à-dire 1024 réalisations du circuit.

Résumé

Dans cette partie, nous avons présenté et appliqué un algorithme présent dans la littérature [1]. Celui-ci nous a permis d'utiliser la représentation de Kraus (49) afin d'exprimer la dynamique de systèmes quantiques ouverts en matrices unitaires. Nous avons ensuite proposé deux protocoles afin de convertir les matrices unitaires, obtenues par l'algorithme, en portes quantiques à un ou deux qubits. Cela-dit, nous avons expliqué qu'il était possible, en pratique, de surpasser cette conversion très théorique par une fonction proposée par Qiskit. Après, nous avons vu comment préparer les états initiaux. Et finalement, nous avons illustré l'algorithme par trois exemples concrets où nous avons aperçu l'efficacité mais également les limites de l'algorithme.

16 Conclusion

Dans ce mémoire, nous avons exploré comment déterminer la dynamique de systèmes quantiques ouverts à l'aide d'un algorithme quantique. Ces systèmes quantiques ouverts, décrits par des dynamiques non-unitaires, posent des défis importants, particulièrement dans leur représentation et leur implémentation pratique sur des systèmes quantiques unitaires.

Dans la partie I, nous avons présenté les états d'un système quantique en utilisant la sphère de Bloch pour les représenter. Nous avons ensuite défini les portes quantiques utilisées pour manipuler ces états, et l'équivalence de certains circuits. Nous avons également abordé la notion d'ordinateur quantique, en particulier à travers les qubits supraconducteurs, le bruit quantique et les ordinateurs quantiques mis à disposition par IBM.

Dans la partie II, nous avons exposé les postulats de la mécanique quantique. Nous avons ensuite introduit l'opérateur densité (20), qui permet de décrire l'état d'un système quantique ouvert. Cet outil nous a conduit à reformuler les postulats dans un cadre mieux adapté aux systèmes quantiques mixtes. Par la suite, nous avons présenté l'équation maîtresse (38), qui régit l'évolution temporelle de l'opérateur densité. Nous avons également établi une méthode générale pour passer de l'équation maîtresse à la représentation de Kraus. Enfin, nous avons illustré cette approche à travers trois exemples de systèmes quantiques ouverts, en détaillant leur représentation de Kraus.

Dans la partie III, nous avons présenté un algorithme issu de la littérature qui nous a permis d'exploiter la représentation de Kraus pour décrire la dynamique des systèmes quantiques ouverts à l'aide de matrices unitaires. Nous avons ensuite expliqué la procédure de préparation des états initiaux de ces systèmes. Par la suite, deux protocoles ont été proposés pour convertir les matrices unitaires générées par l'algorithme en portes quantiques à un ou deux qubits. Cela dit, nous avons également souligné qu'en pratique, cette conversion très théorique peut être remplacée par une fonction intégrée dans Qiskit, offrant une solution plus efficace. Nous avons également vu les limites actuelles des ordinateurs quantiques par la contrainte des portes disponibles sur IBM ainsi que les inconvénients de Qiskit qui est en voie de développement et pour lequel, les fonctions changent régulièrement.

Un aspect clé abordé dans ce mémoire était la transition de l'équation maîtresse aux opérateurs de Kraus, une étape nécessaire pour décrire l'évolution des systèmes ouverts. Bien que théoriquement élégante, cette représentation devient rapidement inapplicable à mesure que le nombre de qubits ou d'opérateurs de Kraus augmente, en raison de la complexité croissante des calculs et des contraintes des plateformes actuelles, telles que la limitation à trois circuits simultanés sur IBM Quantum.

Dans ce travail, nous avons appliqué une méthode existante permettant de simuler des systèmes quantiques ouverts sur un hardware quantique unitaire. Cette approche, bien que reprise de la littérature, a été adaptée à notre problématique spécifique, illustrant ainsi ses limites et son potentiel pour des études futures.

Comme les représentations de Kraus sont en pratiques difficiles à trouver à partir d'une équation maîtresse, une alternative à l'algorithme que nous avons utilisé serait un algorithme permettant d'outrepasser la représentation de Kraus comme dans les articles [64, 65, 66, 67]. Dans ces articles, la dynamique peut-être déterminée directement à partir de l'équation maîtresse.

Références

- [1] Zixuan Hu, Rongxin Xia, and Sabre Kais. A quantum algorithm for evolving open quantum dynamics on quantum computing devices. *Scientific reports*, 10(1) :3301, 2020.
- [2] Wjatscheslaw Popp, Dominik Brey, Robert Binder, and Irene Burghardt. Quantum dynamics of exciton transport and dissociation in multichromophoric systems. *Annual review of physical chemistry*, 72(1) :591–616, 2021.
- [3] Xin-Qi Li, JunYan Luo, Yong-Gang Yang, Ping Cui, and YiJing Yan. Quantum master-equation approach to quantum transport through mesoscopic systems. *Physical Review B—Condensed Matter and Materials Physics*, 71(20) :205304, 2005.
- [4] Christopher K Prier, Danica A Rankic, and David WC MacMillan. Visible light photoredox catalysis with transition metal complexes : applications in organic synthesis. *Chemical reviews*, 113(7) :5322–5363, 2013.
- [5] Armando Relaño, JM Arias, Jorge Dukelsky, José Enrique García-Ramos, and Pedro Pérez-Fernández. Decoherence as a signature of an excited-state quantum phase transition. *Physical Review A—Atomic, Molecular, and Optical Physics*, 78(6) :060102, 2008.
- [6] Lea F Santos and Francisco Pérez-Bernal. Structure of eigenstates and quench dynamics at an excited-state quantum phase transition. *Physical Review A*, 92(5) :050101, 2015.
- [7] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19) :10856–10915, 2019.
- [8] Abhinav Anand, Philipp Schleich, Sumner Alperin-Lea, Phillip WK Jensen, Sukin Sim, Manuel Díaz-Tinoco, Jakob S Kottmann, Matthias Degroote, Artur F Izmaylov, and Alán Aspuru-Guzik. A quantum computing view on unitary coupled cluster theory. *Chemical Society Reviews*, 51(5) :1659–1684, 2022.
- [9] Minh Tuan Duong. *Développement de points quantiques de pérovskite pour la technologie de l'information quantique tout optique et la spintronique*. PhD thesis, Université Grenoble Alpes, 2023.
- [10] Anaïs Dréau. *Spins individuels dans le diamant pour l'information quantique*. PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2013.
- [11] Ioanna Kriekouki. *Approche industrielle aux boîtes quantiques dans des dispositifs de silicium sur isolant complètement déplété pour applications en information quantique*. PhD thesis, Université Grenoble Alpes, 2022.
- [12] Nahed Abdelgaber and Chris Nikolopoulos. Overview on quantum computing and its applications in artificial intelligence. In *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 198–199. IEEE, 2020.
- [13] Nick S Blunt, Joan Camps, Ophelia Crawford, Róbert Izsák, Sebastian Leontica, Arjun Mirani, Alexandra E Moylett, Sam A Scivier, Christoph Sunderhauf, Patrick Schopf, et al. Perspective on the current state-of-the-art of quantum computing for drug discovery applications. *Journal of Chemical Theory and Computation*, 18(12) :7001–7023, 2022.
- [14] Yudong Cao, Jhonathan Romero, and Alán Aspuru-Guzik. Potential of quantum computing for drug discovery. *IBM Journal of Research and Development*, 62(6) :6–1, 2018.
- [15] Peter Cooper, Philipp Ernst, Dieter Kiewell, and Dickon Pinner. Quantum computing just might save the planet. *McKinsey & Company*, 2022.
- [16] IBM. Ibm quantum, 2024. <https://www.ibm.com/quantum> : Consulté le 4 novembre 2024.
- [17] Béla Sz Nagy, Ciprian Foias, Hari Bercovici, and László Kérchy. *Harmonic analysis of operators on Hilbert space*. Springer Science & Business Media. Non consulté directement, mentionné dans : E. Levy and O. M. Shalit. [60], 2010.
- [18] Almut Beige, Daniel Braun, Ben Tregenna, and Peter L Knight. Quantum computing using dissipation to remain in a decoherence-free subspace. *Physical review letters*, 85(8) :1762, 2000.
- [19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779) :505–510, 2019.

- [20] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6) :467–488, Jun 1982.
- [21] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. *Physical review A*, 52(4) :R2493, 1995.
- [22] Emanuel Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029) :39–44, 2005.
- [23] DV Averin. Solid-state qubits under control. *Nature*, 398(6730) :748–749, 1999.
- [24] Lev B Ioffe, Vadim B Geshkenbein, Mikhail V Feigel’Man, Alban L Fauchere, and Gianni Blatter. Environmentally decoupled sds-wave josephson junctions for quantum computing. *Nature*, 398(6729) :679–681, 1999.
- [25] JE Mooij, TP Orlando, L Levitov, Lin Tian, Caspar H Van der Wal, and Seth Lloyd. Josephson persistent-current qubit. *Science*, 285(5430) :1036–1039, 1999.
- [26] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy Lloyd O’Brien. Quantum computers. *nature*, 464(7285) :45–53, 2010.
- [27] Kenton R Brown, Andrew C Wilson, Yves Colombe, C Ospelkaus, Adam M Meier, Emanuel Knill, Dietrich Leibfried, and David J Wineland. Single-qubit-gate error below 10^{-4} in a trapped ion. *Physical Review A—Atomic, Molecular, and Optical Physics*, 84(3) :030303, 2011.
- [28] JM Hornibrook, JI Colless, ID Conway Lamb, SJ Pauka, H Lu, AC Gossard, JD Watson, GC Gardner, S Fallahi, MJ Manfra, et al. Cryogenic control architecture for large-scale quantum computing. *Physical Review Applied*, 3(2) :024010, 2015.
- [29] Ling Hu, Yuwei Ma, Weizhou Cai, Xianghao Mu, Yuan Xu, Weiting Wang, Yukai Wu, Haiyan Wang, YP Song, C-L Zou, et al. Quantum error correction and universal gate set operation on a binomial bosonic logical qubit. *Nature Physics*, 15(5) :503–508, 2019.
- [30] Zhuo-Ping Hong, Bao-Jie Liu, Jia-Qi Cai, Xin-Ding Zhang, Yong Hu, ZD Wang, and Zheng-Yuan Xue. Implementing universal nonadiabatic holonomic quantum gates with transmons. *Physical Review A*, 97(2) :022332, 2018.
- [31] Quantinuum. Quantinuum launches industry-first trapped-ion 32-qubit quantum computer that challenges the world’s best supercomputers. <https://www.quantinuum.com/press-releases/quantinuum-launches-industry-first-trapped-ion-56-qubit-quantum-computer-that-challenges-the-worlds-best-2024>. Visité le 20/11/2024.
- [32] Quantinuum. A new breakthrough in logical quantum computing reveals the scale of our industry leadership, 2024. Visité le 20/11/2024.
- [33] Dik Bouwmeester, Jian-Wei Pan, Klaus Mattle, Manfred Eibl, Harald Weinfurter, and Anton Zeilinger. Experimental quantum teleportation. *Nature*, 390(6660) :575–579, 1997.
- [34] Saeed Golestan, MR Habibi, SY Mousazadeh Mousavi, Josep M Guerrero, and Juan C Vasquez. Quantum computation in power systems : An overview of recent advances. *Energy Reports*, 9 :584–596, 2023.
- [35] Agustin Borgna. *Vers une formalisation d’une chaîne de compilation pour un ordinateur quantique/ Theses. fr*. PhD thesis, Université de Lorraine, 2023.
- [36] Barbara Kraus and J Ignacio Cirac. Optimal creation of entanglement using a two-qubit gate. *Physical Review A*, 63(6) :062309, 2001.
- [37] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5) :3457, 1995.
- [38] Vivek V Shende, Stephen S Bullock, and Igor L Markov. Synthesis of quantum logic circuits. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, pages 272–275, 2005.
- [39] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [40] Yuhui Wang. Analysis on the mechanism of superconducting quantum computer. In *Journal of Physics : Conference Series*, volume 1634, page 012040. IOP Publishing, 2020.
- [41] IBM Quantum. Quantum roadmap 2033, 2023. Visité le 20/11/2024.

- [42] Université de Sherbrooke. La collaboration entre hydro-sherbrooke et l'université de sherbrooke donne un nouvel essor aux recherches sur le réseau électrique, 2020. <https://www.usherbrooke.ca/actualites/nouvelles/details/42889> : Consulté le 2 novembre 2024.
- [43] IBM Newsroom. Ibm expands qiskit, world's most performant quantum software, 2024. <https://newsroom.ibm.com/2024-05-15-IBM-Expands-Qiskit,-Worlds-Most-Performant-Quantum-Software> : Consulté le 12 novembre 2024.
- [44] IBM Newsroom. Built for the era of quantum utility and beyond, 2024. <https://www.ibm.com/quantum/blog/quantum-roadmap-2033> : Consulté le 12 novembre 2024.
- [45] F. Damanet. Quantum information and computation, phys0997-1. Notes de cours, Université de Liège, 2023-2024.
- [46] T. Bastin, J. Martin, and P. Schlagheck. Mécanique quantique avancée, phys3021-1. Notes de cours, Université de Liège, 2022-2023.
- [47] GM Moy, JJ Hope, and CM Savage. Born and markov approximations for atom lasers. *Physical Review A*, 59(1) :667, 1999.
- [48] F. Damanet. Dynamique quantique dissipative et application à la superradiance, 2011-2012.
- [49] Goran Lindblad. On the generators of quantum dynamical semigroups. *Communications in mathematical physics*, 48 :119–130, 1976.
- [50] Vittorio Gorini, Andrzej Kossakowski, and Ennackal Chandy George Sudarshan. Completely positive dynamical semigroups of n-level systems. *Journal of Mathematical Physics*, 17(5) :821–825, 1976.
- [51] Jonas Larson and Themistoklis Mavrogordatos. *The Jaynes–Cummings model and its descendants : modern research directions*. IoP Publishing, 2021.
- [52] Man-Duen Choi. Completely positive linear maps on complex matrices. *Linear algebra and its applications*, 10(3) :285–290, 1975.
- [53] Andrzej Jamiolkowski. Linear transformations which preserve trace and positive semidefiniteness of operators. *Reports on mathematical physics*, 3(4) :275–278, 1972.
- [54] Fabrizio Minganti, Alberto Biella, Nicola Bartolo, and Cristiano Ciuti. Spectral theory of liouvillians for dissipative phase transitions. *Phys. Rev. A*, 98 :042118, Oct 2018.
- [55] Gabriel T Landi. Quantum information and quantum noise. *en. graduate course in Quantum Information and Quantum noise*, 2018.
- [56] Erika Andersson, James D Cresser, and Michael JW Hall. Finding the kraus decomposition from a master equation and vice versa. *Journal of Modern Optics*, 54(12) :1695–1716, 2007.
- [57] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. *Applied physics reviews*, 6(2), 2019.
- [58] Hiroyuki Hayashi, Gen Kimura, and Yukihiko Ota. Kraus representation in the presence of initial correlations. *Physical Review A*, 67(6) :062109, 2003.
- [59] D. M. Tong, J. L. Chen, J. Y. Huang, L. C. Kwек, and C. H. Oh. Kraus representation for the density operator of a qubit. *Laser Physics*, 16(11) :1512–1516, Nov 2006.
- [60] Eliahu Levy and Orr Moshe Shalit. Dilation theory in finite dimensions : the possible, the impossible and the unknown. 2014.
- [61] W Forrest Stinespring. Positive functions on c^* -algebras. *Proceedings of the American Mathematical Society*, 6(2) :211–216, 1955.
- [62] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- [63] IBM Quantum. Ecrgate documentation - qiskit. <https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.ECRGate>, 2024. Lien visité le 24/12/24.
- [64] Anthony W Schlimgen, Kade Head-Marsden, LeeAnn M Sager, Prineha Narang, and David A Mazziotti. Quantum simulation of the lindblad equation using a unitary decomposition of operators. *Physical Review Research*, 4(2) :023216, 2022.
- [65] Nobuyuki Yoshioka, Yuya O. Nakagawa, Kosuke Mitarai, and Keisuke Fujii. Variational quantum algorithm for nonequilibrium steady states. *Phys. Rev. Res.*, 2 :043289, Nov 2020.

- [66] Nathan Ramusat and Vincenzo Savona. A quantum algorithm for the direct estimation of the steady state of open quantum systems. *Quantum*, 5 :399, 2021.
- [67] Hirsh Kamakari, Shi-Ning Sun, Mario Motta, and Austin J. Minnich. Digital quantum simulation of open quantum systems using quantum imaginary–time evolution. *PRX Quantum*, 3 :010320, Feb 2022.

A Annexe

Cette annexe est le code que j'ai utilisé pour simuler le 2^{ème} exemple de ce mémoire, à savoir le canal d'amortissement d'amplitude modifié (68), dans la section (III). Elle vous permettra d'implémenter le circuit facilement en python via Qiskit. Attention, les jobs ont été lancés au préalable sur IBM Quantum [16]. Attention, les mises à jour de Qiskit sont très fréquentes et il n'est pas rare qu'une partie d'une librairie soit remplacée par une autre. Il est donc impératif de se tenir au courant des nouveautés de Qiskit.

```
import matplotlib.pyplot as plt
import numpy as np
from qiskit import QuantumRegister, QuantumCircuit, transpile
from qiskit_aer import AerSimulator
import numpy as np
import matplotlib.pyplot as plt
from qiskit.quantum_info import Operator
from qiskit_ibm_runtime import QiskitRuntimeService

time_steps = np.arange(0, 1000 * 10**(-12), 10 * 10**(-12))
gamma = 1.52 * 10**9
time = list(time_steps)
Omega0 = 0.314 * 10**9

# Définissons nos opérateurs de Kraus
def M0(t):
    return np.array([[np.exp(-1j * Omega0 * t), 0], [0, np.exp(1j * Omega0 * t - 1/2 * gamma * t)]], dtype=complex)
def M1(t):
    return np.array([[0, np.sqrt(1 - np.exp(-gamma * t))], [0, 0]], dtype=complex)

# Définissons les DMk
def DMk(t, Mk):
    M_t = Mk(t)
    return np.sqrt(np.eye(2) - M_t.T.conj() @ M_t)

# Définissons nos matrices de dilatations
def U_Mk(t, Mk):
    M_t = Mk(t)
    DM_t = DMk(t, Mk)
    return np.block([[M_t, DM_t.T.conj()], [DM_t, -M_t.T.conj()]])

# Définition des états préparés
V1 = np.array([[0], [1], [0], [0]], dtype=complex) # Etat préparé comme |1>
V2 = (1/np.sqrt(2)) * np.array([[1], [1], [0], [0]], dtype=complex) # Etat préparé comme 1/Sqrt(2) (|0> + |1>)

# Calcul des UMk Vi pour un état donné
def CalculateUmkVi(U_Mk, v_i, t, Mk):
    results = []
    for i in t:
        U = U_Mk(i, Mk)
        temporaire = U @ v_i
        results.append(temporaire)
    return np.array(results)

# Application de M0 sur les états préparés
UMOV1 = CalculateUmkVi(U_Mk, V1, time, M0)
UMOV2 = CalculateUmkVi(U_Mk, V2, time, M0)

# Application de M1 sur les états préparés
UMiV1 = CalculateUmkVi(U_Mk, V1, time, M1)
UMiV2 = CalculateUmkVi(U_Mk, V2, time, M1)

# Calcul des populations
populations_g_combined = 1/2 * (np.abs(UMOV1[:, 0])**2 + np.abs(UMOV2[:, 0])**2 + np.abs(UMiV1[:, 0])**2 + np.abs(UMiV2[:, 0])**2)
populations_e_combined = 1/2 * (np.abs(UMOV1[:, 1])**2 + np.abs(UMOV2[:, 1])**2 + np.abs(UMiV1[:, 1])**2 + np.abs(UMiV2[:, 1])**2)
populations_tot = populations_g_combined + populations_e_combined

#SIMULATEUR:
population_ground_alltime = []
population_01_alltime = []

# Définition du simulateur utilisé
```

```

Simulator = AerSimulator()
qreg_q = QuantumRegister(2, 'q')

#Préparation de l'état V1 = |01>
circuitV1 = QuantumCircuit(qreg_q)
circuitV1.x(0)

#Préparation de l'état V2 = 1/Sqrt(2) (|00>+ |01>)
circuitV2 = QuantumCircuit(qreg_q)
circuitV2.h(0)

for t in time_steps:
    a = np.exp(- 1j * Omega0 * t)
    b = np.exp(- 1/2 * gamma * t + 1j * Omega0 * t)
    c = np.exp(- gamma * t)
    d = np.exp(+ 1j * Omega0 * t)
    e = np.exp(- 1/2 * gamma * t - 1j * Omega0 * t)

    UMO = np.array([
        [a, 0, 0, 0],
        [0, b, 0, np.sqrt(1 - c)],
        [0, 0, - d, 0],
        [0, np.sqrt(1 - c), 0, -e]
    ])

    UM1 = np.array( [
        [0, np.sqrt(1-c), np.sqrt(c), 0],
        [0, 0, 0, 1],
        [1, 0, 0, 0],
        [0, np.sqrt(c), -np.sqrt(1-c), 0]
    ])

    #On crée nos circuits
    circuitUM0 = QuantumCircuit(qreg_q)
    circuitUM0.unitary(UMO, [0,1])

    circuitUM1 = QuantumCircuit(qreg_q)
    circuitUM1.unitary(Operator(UM1), [0,1])

    #On peut appliquer les circuits représentant UMO et UM1 sur les états préparés avec nos circuits V1 et V2
    UMOV1 = circuitV1.compose(circuitUM0)
    UM1V1 = circuitV1.compose(circuitUM1)

    UMOV2 = circuitV2.compose(circuitUM0)
    UM1V2 = circuitV2.compose(circuitUM1)
    #Ajouter une mesure sur les états après (V1 + UMO)
    UMOV1.measure_all()
    UM1V1.measure_all()

    UMOV2.measure_all()
    UM1V2.measure_all()
    #On compile les circuits
    compiled_circuit1 = transpile(UMOV1, Simulator, optimization_level=3)
    compiled_circuit2 = transpile(UM1V1, Simulator, optimization_level=3)

    compiled_circuit3 = transpile(UMOV2, Simulator, optimization_level=3)
    compiled_circuit4 = transpile(UM1V2, Simulator, optimization_level=3)

    #On lance sur le simulateur
    job1 = Simulator.run(compiled_circuit1)
    job2 = Simulator.run(compiled_circuit2)
    job3 = Simulator.run(compiled_circuit3)
    job4 = Simulator.run(compiled_circuit4)

    #On récupère les données
    result1 = job1.result()
    result2 = job2.result()
    result3 = job3.result()
    result4 = job4.result()

    #On récupère les comptages
    count1 = result1.get_counts(compiled_circuit1)
    count2 = result2.get_counts(compiled_circuit2)
    count3 = result3.get_counts(compiled_circuit3)
    count4 = result4.get_counts(compiled_circuit4)

```

```

#réinitialiser les populations
population_ground = 0
population_excited = 0
#Calculer à tous les temps les populations
population_ground = 2 * (count1.get('00',0) + count2.get('00', 0) + count3.get('00', 0) + count4.get('00', 0))/4096

population_01 = 2 * (count1.get('01',0) + count2.get('01', 0) + count3.get('01', 0) + count4.get('01', 0))/4096
#Les ajouter dans mon tableau
population_ground_alltime.append(population_ground)
population_01_alltime.append(population_01)

# Sur l'hardware, après avoir lancé les circuits sur IBM Quantum

# Se connecter à IBM
MonToken = "PUT_YOUR_TOKEN_HERE"
service = QiskitRuntimeService(channel="ibm_quantum", token=MonToken)

#t = 1
job1 = service.job('cxmqrp2ky7rg0084fh0g' )
job2 = service.job('cxmqrptgckkg008syprg' )
job3 = service.job('cxmvq0w6t010008d7h70' )
job4 = service.job('cxmvq1mgckkg008sz1tg' )
#t = 100
job5 = service.job('cxm6ta9gckkg008sx2p0' )
job6 = service.job('cxm6t9hvw7kg008smy9g' )
job7 = service.job('cxm6q24vw7kg008smxx0' )
job8 = service.job('cxm6q1cky7rg0084dty0' )

jobs = [job1, job2, job3, job4, job5, job6, job7, job8] # Pour l'anneze, je ne montre que 8 jobs mais une trentaine de jobs ont été effectués

timeHardware = np.array([1,100]) #Je mets les temps en fonction des apparitions dans le tableau des populations

N_job = len(jobs)

counts = []
counts_normalisés = []
population_ground_hard = []
population_excited_hard = []

for i in range(N_job):
    counts.append(jobs[i].result().data()['counts'])
    counts_normalisés.append({state: count / 2048 for state, count in counts[i].items()}) #Normalisation

for i in range(N_job // 4):
    ground = counts_normalisés[4 * i].get('0x0', 0) + counts_normalisés[4 * i + 1].get('0x0', 0)
    + counts_normalisés[4 * i + 2].get('0x0', 0) + counts_normalisés[4 * i + 3].get('0x0', 0)
    excited = counts_normalisés[4 * i].get('0x1', 0) + counts_normalisés[4 * i + 1].get('0x1', 0)
    + counts_normalisés[4 * i + 2].get('0x1', 0) + counts_normalisés[4 * i + 3].get('0x1', 0)
    population_ground_hard.append(ground)
    population_excited_hard.append(excited)

# Le changement de base via la théorie
def construct_matrix(T, I): #Construire la matrice (T 0, 0 I)
    top = np.hstack((T, np.zeros_like(I)))
    bottom = np.hstack((np.zeros_like(T), I))
    return np.vstack((top, bottom))

# Définitions de la matrice de changement de base
T = (1 / np.sqrt(2)) * np.array([[1, 1], [1,-1]])
I = np.eye(2)
MatriceTI = construct_matrix(T, I)

def ApplyT(U_Mk, t, v_i, Mk): #Calcul de changement de base
    results = []
    for i in t:
        U = U_Mk(i, Mk)
        temporaire = MatriceTI @ (U @ v_i).flatten()
        results.append(temporaire)
    return np.array(results)

# On calcule TI @ U_Mk @ v_i
result_v1_M0 = ApplyT(U_Mk, time, V1, M0)
result_v1_M1 = ApplyT(U_Mk, time, V1, M1)

```

```

result_v2_M0 = ApplyT(U_Mk, time, V2, M0)
result_v2_M1 = ApplyT(U_Mk, time, V2, M1)

# Extraire les populations des états de base et excités après changement de base
populations_g_combined_ti = 1/2 * (np.abs(result_v1_M0[:, 0])**2 + np.abs(result_v1_M1[:, 0])**2
+ np.abs(result_v2_M0[:, 0])**2 + np.abs(result_v2_M1[:, 0])**2 )
populations_e_combined_ti = 1/2 * (np.abs(result_v1_M0[:, 1])**2 + np.abs(result_v1_M1[:, 1])**2
+ np.abs(result_v2_M0[:, 1])**2 + np.abs(result_v2_M1[:, 1])**2)
populations_tot_ti = populations_g_combined_ti + populations_e_combined_ti

# Simulateur pour le changement de base
population_ground_alltimeT = []
population_01_alltimeT = []

for t in time_steps:
    a = np.exp(- 1j * Omega0 * t)
    b = np.exp(- 1/2 * gamma * t + 1j * Omega0 * t)
    c = np.exp(- gamma * t)
    d = np.exp(+ 1j * Omega0 * t)
    e = np.exp(- 1/2 * gamma * t - 1j * Omega0 * t)

    UMO = np.array([
        [a, 0, 0, 0],
        [0, b, 0, np.sqrt(1 - c)],
        [0, 0, - d, 0],
        [0, np.sqrt(1 - c), 0, -e]
    ])

    UM1 = np.array( [
        [0, np.sqrt(1-c), np.sqrt(c), 0],
        [0, 0, 0, 1],
        [1, 0, 0, 0],
        [0, np.sqrt(c), -np.sqrt(1-c), 0]
    ])

    #On crée nos circuits
    circuitUM0 = QuantumCircuit(qreg_q)
    circuitUM0.unitary(UMO, circuitUM0.qubits)

    #Appliquer le changement de base
    circuitUM0.unitary(MatriceTI, circuitUM0.qubits)

    circuitUM1 = QuantumCircuit(qreg_q)
    circuitUM1.unitary(Operator(UM1), [0,1])

    #Appliquer le changement de base
    circuitUM1.unitary(MatriceTI, circuitUM1.qubits)

    #On peut appliquer les circuits représentants UMO et UM1 sur les états préparés avec les circuits V1 et V2
    UMOV1 = circuitV1.compose(circuitUM0)
    UM1V1 = circuitV1.compose(circuitUM1)
    UMOV2 = circuitV2.compose(circuitUM0)
    UM1V2 = circuitV2.compose(circuitUM1)

    #Ajouter une mesure sur le circuit
    UMOV1.measure_all()
    UM1V1.measure_all()
    UMOV2.measure_all()
    UM1V2.measure_all()

    #On compile les circuits
    compiled_circuit1 = transpile(UMOV1, Simulator, optimization_level=3)
    compiled_circuit2 = transpile(UM1V1, Simulator, optimization_level=3)
    compiled_circuit3 = transpile(UMOV2, Simulator, optimization_level=3)
    compiled_circuit4 = transpile(UM1V2, Simulator, optimization_level=3)

    #On lance sur le simulateur
    job1 = Simulator.run(compiled_circuit1)
    job2 = Simulator.run(compiled_circuit2)
    job3 = Simulator.run(compiled_circuit3)
    job4 = Simulator.run(compiled_circuit4)

    #On récupère les données
    result1 = job1.result()
    result2 = job2.result()

```

```

result3 = job3.result()
result4 = job4.result()

#On récupère les comptages
count1 = result1.get_counts(compiled_circuit1)
count2 = result2.get_counts(compiled_circuit2)
count3 = result3.get_counts(compiled_circuit3)
count4 = result4.get_counts(compiled_circuit4)

#réinitialiser les populations
population_groundT = 0
population_excitedT = 0

#Calculer à tous les temps les populations
population_groundT = 2 * (count1.get('00',0) + count2.get('00', 0) + count3.get('00',0) + count4.get('00', 0) )/4096
population_01T = 2 * (count1.get('01',0) + count2.get('01', 0) + count3.get('01',0) + count4.get('01', 0) )/4096

#Les ajouter dans le tableau
population_ground_alltimeT.append(population_groundT)
population_01_alltimeT.append(population_01T)

# Hardware pour le changement de base

# Se connecter à IBM
MonToken = "PUT_YOUR_TOKEN_HERE"
service = QiskitRuntimeService(channel="ibm_quantum", token=MonToken)

#t = 1
job1 = service.job('cnnxqny4a290008xfvz0' )
job2 = service.job('cnnxqne0v15000804d40' )
job3 = service.job('cnnxpvk082700083t68g' )
job4 = service.job('cnnxpv34a290008xfvvg' )

#t = 100
job5 = service.job('cxm6ta9gcckg008sx2p0' )
job6 = service.job('cxm6t9hvw7kg008smy9g' )
job7 = service.job('cxm6q24vw7kg008smxx0' )
job8 = service.job('cxm6q1cky7rg0084dty0' )

jobs = [job1, job2, job3, job4, job5, job6, job7, job8]
timeHardware = np.array([100]) #Je mets les temps en fonction des apparitions dans le tableau des populations

N_job = len(jobs)

counts = []
counts_normalisés = []
population_ground_hard = []
population_excited_hard = []

for i in range(N_job):
    counts.append(jobs[i].result().data()['counts'])
    counts_normalisés.append({state: count / 2048 for state, count in counts[i].items()}) #Normalisation

for i in range(N_job // 4):
    ground = counts_normalisés[4 * i].get('0x0', 0) + counts_normalisés[4 * i + 1].get('0x0', 0)
    + counts_normalisés[4 * i + 2].get('0x0', 0) + counts_normalisés[4 * i + 3].get('0x0', 0)
    excited = counts_normalisés[4 * i].get('0x1', 0) + counts_normalisés[4 * i + 1].get('0x1', 0)
    + counts_normalisés[4 * i + 2].get('0x1', 0) + counts_normalisés[4 * i + 3].get('0x1', 0)
    population_ground_hard.append(ground)
    population_excited_hard.append(excited)

```