

CALCUL QUANTIQUE ET NEUROINFORMATIQUE

Harshana Runjeet

3 mars 2024

Table des matières

1	Introduction à la neuroscience quantique(Stage)	3
1.1	Diffusion IRM des fibres tractographiques dans le cerveau	3
1.2	Résumé des articles lus et extraits de codes/ algorithmes	4
2	Pont entre neuroinformatique et quantique (Stage)	4
2.1	Les algorithmes variationnels et QAOA	4
2.2	Essai 1 avec contraintes du papier shortest path QAOA	5
2.3	Essai 2 nombre pair de connexions dans un code automatisé	10
2.4	Résultats du code avec essai 2	12
3	Plus amples détails, Simulation de systèmes quantiques (Stage)	15
4	Information quantique(Cours)	16
4.1	Histoire et vue globale	16
4.2	Jeu GHZ	17
4.3	Théorème de non-clonage	17
5	Protocoles et communication quantique (Cours)	18
5.1	Distribution quantique de clés	18
5.2	Encodage dense	19
5.3	Téléportation quantique	19
6	Circuits quantiques (Cours)	20
6.1	Circuits classiques et complexité	20
6.2	Calculs réversibles	21
6.3	Circuits quantiques et portes logiques	21
6.4	Ensemble universel	23
7	Algorithmes quantiques(Cours)	23
7.1	Circuits quantiques élémentaires	24
7.2	Algorithme de Deutsch	24
7.3	Algorithme de Deutsch-Jozsa	25
7.4	Problème de Bernstein-Vazirani	25
7.5	Transformée de Fourier quantique	25
7.6	Estimation de phase	25
7.7	Algorithme de factorisation de Shor	26
7.8	Algorithme de Grover	27

TABLE DES MATIÈRES

8 Codes de correction d'erreurs(Cours)	28
8.1 Condition de Knill-Laflamme	28
8.2 Code de Shor	28
8.3 Discrétisation des erreurs	28
A Annexe A :Reconstruction d'image quantique (Kiani et al.)	29
B Annexe B : Formalisme mathématique en information quantique	31
Annexes	29

1 Introduction à la neuroscience quantique(Stage)

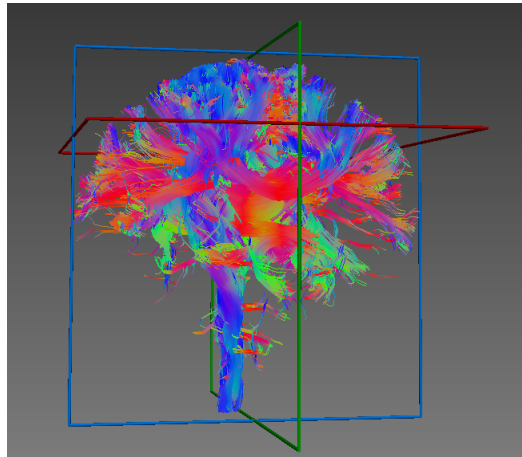
Une des plus grandes applications de l'informatique quantique consiste la compréhension du cerveau humain avec ses 86 milliards de neurones et 242 000 milliards de synapses. Dans le cerveau, l'activité d'une région peut affecter une autre région distante sans modifier l'activité des régions entre les deux. L'organisation du cerveau peut se comprendre en différentes strates allant du système nerveux de grandeur d'environ 1m, sous-système de 10cm, neurone de $100\mu m$, aux canaux d'ioniques de 1pm. Il y a plus de 85 milliards de cellules gliales, des cellules du système nerveux qui soutiennent et protègent les neurones, assurant des fonctions de soutien structurel et métabolique. On sait que n-qubits peuvent stocker 2^n bits, ce qui fait que 79 qubits suffisent à se rendre à l'ordre de grandeur du nombre d'Avogadro (nombre de cellules dans une mole) Plusieurs groupes de recherche travaillent sur des algorithmes quantiques dans le domaine de l'imagerie. C'est le cas dans l'article **Kiani**. Voici quelques notes de l'article en annexe [A](#).

Le cerveau comporte tellement de neurones qu'une analyse de toutes ses connexions demandent de grandes puissances de calcul. La théorie des graphes a déjà été envisagé auparavant, mais si on prend chaque voxel d'une imagerie médicale IRM comme étant un noeud d'un graphe, la taille est si énorme qu'il serait impossible de tracer les connexions du cerveau avec un ordinateur classique, qui est incapable de parallélisme, concept expliqué dans la section Information quantique. Une des méthodes de traçage de l'information dans le cerveau actuelle consiste en la tractographie. La tractographie vise à tracer une carte de diffusion de l'eau dans la matière blanche du cerveau. Actuellement, l'analyse et les algorithmes se fait localement à chaque voxel ce qui n'est pas aussi efficace qu'une analyse globale. Il a été empiriquement démontré que l'analyse globale donne une vue d'ensemble plus grande que l'analyse locale et une méthode d'analyse globale intéressante serait par la théorie des graphes. C'est pourquoi il y a intérêt à pousser la voie vers un algorithme quantique qui pourrait permettre le traçage globale à travers une méthode de graphe, ce que les ordinateurs classiques ne peuvent faire.

1.1 Diffusion IRM des fibres tractographiques dans le cerveau

jeurissen

Avec MI-Brain de Imeka :



L'imagerie par résonance magnétique sert à différencier les tissus dans le cerveau, ainsi que la séquence d'imagerie écho planaire. Lorsqu'un champ magnétique est envoyé dans le cerveau, le spin des protons aura tendance à s'aligner avec le champ magnétique ainsi qu'à adopter la même fréquence de précession. On peut appliquer un champ perpendiculaire et ensuite l'enlever, le temps de relaxation permet de dire si l'on se situe dans la matière blanche, ou encore le liquide cébrospinal. D'autre

part, il est possible d'appliquer un champ B0 et observer l'état du système. On applique ensuite un champ -B0 qui devrait annuler l'effet. Les endroits où l'on voit un signal affaibli indique qu'il y a eu diffusion des molécules (un déphasage). Le principe de tenseur de diffusion quant à lumière permet de déterminer les directions vers lesquelles les molécules d'eau sont les plus enclins à diffuser.

Il y a trois vues principales en imagerie : la vue axiale (dessus), la vue sagittale (de côté) ainsi que la vue coronale (de face/arrière). Ces trois vues sont un plan propre de l'espace 3D. Dans le liquide cérébrospinal, la diffusion est isotrope ; on devrait avoir une couleur plus pâle. Dans la matière blanche, le mouvement est plutôt anisotrope.

1.2 Résumé des articles lus et extraits de codes/ algorithmes

Voir ce carnet jupyter pour un résumé des articles HARDI, human connectome et Q-Ball imaging ainsi que mes essais de codes de type tracking et ma réflexion au fur et à mesure : [lecture_chapitre.ipynb](#)

2 Pont entre neuroinformatique et quantique (Stage)

Suite à la lecture sur la tractographie et la compréhension des algorithmes classiques, ainsi qu'à la compréhension des bases en information quantique (section Information quantique et Protocoles et communication quantique dans les sections de types Cours qui suivent), il est possible à présent de définir une vision du stage. La librairie `scilpy` permet un tracking avancé avec des paramètres avancés tels que z . À cette librairie pourrait être implémenté plusieurs outils utiles aussi tels que x et y . On a vu qu'il est possible de percevoir le cerveau dans l'article Qball imaging, comme un réseau de graphe avec chaque voxel représentant un noeud, et les ODF en un voxel vont indiquer le poids qui relie les différents voxels entre eux. C'est d'ailleurs ce que fait la librairie `tractography` dont je prendrai le temps d'élaborer sur les différents scripts dans une prochaine sous-section. De plus, voici un lien vers les codes de circuits faits avec la librairie `Qiskit`, qui couvrent les notions de base d'information quantique, les techniques de multiplication de matrice, ou encore de produit tensoriel. Il s'agit d'une mise en pratique des leçons disponibles sur le site d'IBM dans la rubrique apprentissage : [IBM_quantique.ipynb](#).

Les premières étapes consistent à écrire un code qui génère un graphe avec un nombre de noeuds et d'arêtes données (jusqu'à 20) qui prend en entrée un fichier `fodf.nii.gz` ainsi qu'un seuil qui détermine si on garde ou non une arête (dans les 8 directions voisines 2D) qui prend le nom de `fodfthreshold` ainsi que le nombre de qubits (soit m). Il sera ensuite question de déterminer de quelle manière interpréter le système ; voulons-nous maximiser le chemin et prendre les poids les plus forts pour un chemin donné, ou encore minimiser le système où dans ce cas il faudrait s'assurer de mettre le chemin le plus probable avec une plus faible valeur. Ensuite, viendrait l'écriture d'un script qui prend en entrée un graphe donné dans le but créer un hamiltonien basé sur la matrice d'adjacence.

2.1 Les algorithmes variationnels et QAOA

Notes sur l'article de QAOA :

QAOA vise à trouver l'énergie fondamentale soit la plus petite énergie par une fonction d'onde paramétrée, on cherche la valeur du paramètre qui va minimiser la fonction. Pour un état $|\psi\rangle$ donné, une énergie fondamentale E_0 , on aurait :

$$E_0 \leq \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$$

On prend une entrée et on génère un ket paramétré en appliquant un opérateur paramétré, avec ce ket paramétré, on peut calculer la moyenne de l'hamiltonien à multiple reprises et trouver le paramètre

qui va minimiser cette valeur moyenne.

$$\lambda = \min_{\theta} \langle \psi_0 | U^\dagger(\theta) H U(\theta) | \psi_0 \rangle$$

$$\lambda_{min} = E_0 \approx \langle \psi(\theta^*) | H | \psi(\theta^*) \rangle$$

Fonction objective et m contraintes :

On veut optimiser une fonction coût : $C(x)$ où x fait partie du domaine d . En quantique, on peut représenter le tout comme un opérateur H_C agissant sur un ket $|z\rangle$, où z est une chaîne de bits appartenant à l'ensemble $\{0, 1\}^{\otimes n}$.

$$H_C |z\rangle = \sum_{k=1}^{m'} C_k |z\rangle = C |z\rangle$$

Ici, C_k est associée à une contrainte, que, lorsque respectée, donne la valeur de 1 à C , 0 sinon. On définit une fonction coût pouvant être mappé à un Hamiltonien. Il faut considérer les contraintes du point de départ, de fin ainsi que les points intermédiaires de manière à avoir un chemin plus court. En ajoutant une valeur plus ou moins importante au M , on augmente la valeur de la fonction qu'on cherche à maximiser. On veut que C soit le plus grand possible de manière à garder le chemin avec les poids les plus grands (plus grande probabilité de diffusion). Ensuite, si une contrainte n'est pas respectée, M sera non nul ce qui viendra à diminuer la valeur de F qu'on cherche à maximiser en valeur absolue ($M > 0$). En d'autres mots, pour minimiser, il faut que la fonction coût soit positive ainsi que les contraintes. Pour maximiser, il faut que la fonction coût obligatoire soit négative (grand en valeur absolue mais petit avec le signe), et mettre une valeur positive aux contraintes, qui, si non respecté, s'additionne positivement à la fonction coût en augmentant la valeur, donc l'algorithme cherchant à minimiser la fonction objective, aura moins intérêt à briser les contraintes, pour avoir un nombre négatif le plus grand négativement possible.

2.2 Essai 1 avec contraintes du papier shortest path QAOA

Il s'agit de l'Hamiltonien (temporaire) donné par :

$$H = -H_C + \alpha \left(\sum_{j=1}^n x_{sj} - 1 \right) + \alpha \left(\sum_{i=1}^n x_{id} - 1 \right) + \alpha \sum_{k \in V, k \neq s, d} \left(\sum_{i=1}^n x_{ik} - \sum_{j=1}^n x_{kj} \right)$$

On sait que C est le coût du plus court chemin,

$$C = \sum_{i,j} w_{ij} x_{ij}$$

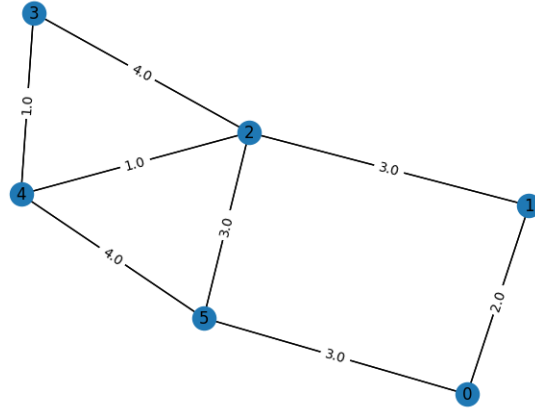
Dans notre cas, nos poids sont donnés par :

$$w_{ij} = w_{ji} = P_{mat}(i) \cdot P_{mat}(j) \cdot [P_{Diff}(i, r_{ij}) + P_{Diff}(j, r_{ji})]$$

Dans `reconst.py`, on peut voir : $w = sf[start_x, start_y, dir_{id}] + sf[conn_x, conn_y, dir_{id}]$ Dans un premier temps, on va poser la matrice d'adjacence suivante :

$$\text{matrice d'adjacence avec poids} = 10 \times \begin{bmatrix} 0.0 & 0.2 & 0.0 & 0.0 & 0.0 & 0.3 \\ 0.2 & 0.0 & 0.3 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.4 & 0.1 & 0.3 \\ 0.0 & 0.0 & 0.4 & 0.0 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.1 & 0.0 & 0.4 \\ 0.3 & 0.0 & 0.3 & 0.0 & 0.4 & 0.0 \end{bmatrix}$$

Qui s'exprime par le graphe suivant :



On peut ensuite définir des qubits associés aux arêtes, dans l'ordre qu'ils apparaissent dans la matrice triangulaire supérieure excluant la diagonale (en commençant par la première ligne de la gauche vers la droite).

Qubit	Edge	Poids
q_0	0 – 1	2
q_1	0 – 5	3
q_2	1 – 2	3
q_3	2 – 3	4
q_4	2 – 4	1
q_5	2 – 5	3
q_6	3 – 4	1
q_7	4 – 5	4

Avec la formule donnée par la fonction objective, on a pour le premier terme (Coût obligatoire) :

$$H_C = 2q_0 + 3q_1 + 3q_2 + 4q_3 + 1q_4 + 3q_5 + 1q_6 + 4q_7$$

$$H_C = 2\left(\frac{I_0 - Z_0}{2}\right) + 3\left(\frac{I_1 - Z_1}{2}\right) + 3\left(\frac{I_2 - Z_2}{2}\right) + 4\left(\frac{I_3 - Z_3}{2}\right) + 1\left(\frac{I_4 - Z_4}{2}\right) + 3\left(\frac{I_5 - Z_5}{2}\right) + 1\left(\frac{I_6 - Z_6}{2}\right) + 4\left(\frac{I_7 - Z_7}{2}\right)$$

$$H_C = \frac{21}{2}I^{\otimes 8} - 1Z_0 - 1.5Z_1 - 1.5Z_2 - 2Z_3 - 0.5Z_4 - 1.5Z_5 - 0.5Z_6 - 2Z_7$$

Voici la représentation en chaîne de Pauli avec Qiskit :

```

1  h_c = SparsePauliOp.from_list([
2      ("IIIIIIII", all_weights_sum/2),
3      ("IIIIIIIZ", -mat_adj[0,1] /2),
4      ("IIIIIIZI", -mat_adj[0,5] /2),
5      ("IIIIIZII", -mat_adj[1,2] /2),
6      ("IIIIZIII", -mat_adj[2,3] /2),
7      ("IIIZIIII", -mat_adj[2,4] /2),
8      ("IIZIIIII", -mat_adj[2,5] /2),
9      ("IZIIIIII", -mat_adj[3,4] /2),
10     ("ZIIIIIII", -mat_adj[4,5] /2)])
    
```

Voici la première contrainte qui s'assure que l'on a un seul edge de départ :

$$H_S = q_0 + q_1$$

$$H_S = \frac{I_0 - Z_0}{2} + \frac{I_1 - Z_1}{2} - I$$

Voici la représentation en chaîne de Pauli avec Qiskit :

```
1 h_s = SparsePauliOp.from_list([("IIIIIIIZI", -0.5),
2                               ("IIIIIIIZ", -0.5)])
```

$$H_F = q_1 + q_5 + q_7$$

$$H_F = \frac{I_1 - Z_1}{2} + \frac{I_5 - Z_5}{2} + \frac{I_7 - Z_7}{2} - I$$

Voici la représentation en chaîne de Pauli avec Qiskit :

```
1 h_f = SparsePauliOp.from_list([("IIIIIIIZI", -0.5),
2                               ("IIZIIIII", -0.5),
3                               ("ZIIIIIII", -0.5),
4                               ("IIIIIIIII", 0.5)])
```

Pour les points intermédiaires, nous avons :

$$\sum_{k \in V, k \neq s, d}^n \left(\sum_{i=1}^n x_{ik} - \sum_{j=1}^n x_{kj} \right)$$

Une manière méthodique de procéder pour poser une direction au graphe est de prendre les lignes de la matrice d'adjacence comme des noeuds où les liens sont sortant et les colonnes comme les noeuds entrant uniquement en considérant la matrice triangle supérieure pour le chemin de l'aller. Pour le chemin du retour, on peut faire la même chose avec la matrice triangulaire inférieure. Les k sont tous les noeuds intermédiaires, soit appartenant à l'ensemble 1,2,3,4.

Ainsi, pour $k = 1$, on a :

$$\left(\sum_{i=1}^n x_{i1} - \sum_{j=1}^n x_{1j} \right)$$

$$H_1 = q_0 - q_2$$

Ainsi, pour $k = 2$, on a :

$$\left(\sum_{i=1}^n x_{i2} - \sum_{j=1}^n x_{2j} \right)$$

$$H_2 = q_2 - q_3 - q_4 - q_5$$

Ainsi, pour $k = 3$, on a :

$$\left(\sum_{i=1}^n x_{i3} - \sum_{j=1}^n x_{3j} \right)$$

$$H_3 = q_3 - q_6$$

Ainsi, pour $k = 4$, on a :

$$\left(\sum_{i=1}^n x_{i4} - \sum_{j=1}^n x_{4j} \right)$$

$$H_4 = q_6 + q_4 - q_7$$

Voici la représentation en chaîne de Pauli avec Qiskit :

```

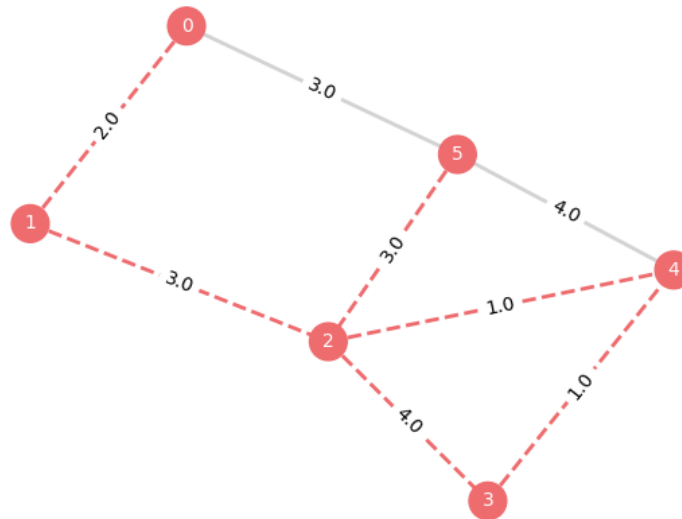
1  h_1 = SparsePauliOp.from_list([("IIIIIIIZ", -0.5),
2                                  ("IIIIIZII", +0.5)])
3
4  h_2 = SparsePauliOp.from_list([("IIIIIIII", -1.0),
5                                  ("IIIIIZII", -0.5),
6                                  ("IIIIZIII", +0.5),
7                                  ("IIIZIIII", +0.5),
8                                  ("IIZIIIII", +0.5)])
9
10 h_3 = SparsePauliOp.from_list([("IIIIZIII", -0.5),
11                                 ("IZIIIIII", +0.5)])
12
13 h_4 = SparsePauliOp.from_list([("IIIIIIII", +0.5),
14                                 ("IIIZIIII", -0.5),
15                                 ("IZIIIIII", -0.5),
16                                 ("ZIIIIIII", +0.5)])
    
```

On a donc l'expression finale avec $\alpha = 23$:

```

1  h = - h_c + alpha * (h_s**2 + h_f**2 + h_1**2 + h_2**2 + h_3**2 +
    h_4**2)
    
```

Avec le nombre de répétitions à 3, nous obtenons le résultat suivant :



Ce résultat comporte plusieurs problèmes :

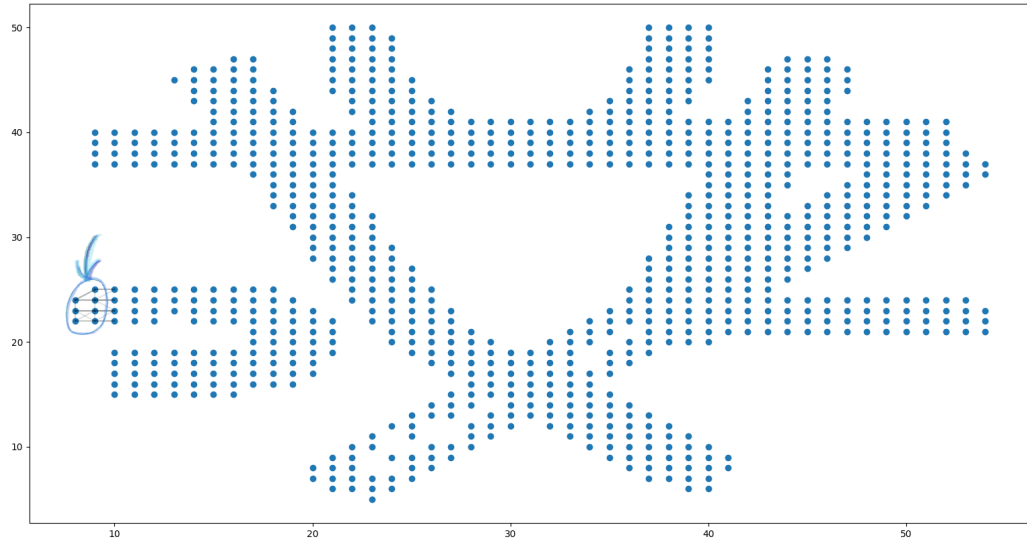
1. Le chemin 4 vers 2 existe alors qu'il ne devrait pas en raison de la directionnalité qu'on a donné à notre graphe.
2. Le résultat est très sensible avec le moindre changement de nombre de répétitions et de valeur de α .
3. Il devrait y avoir la même probabilité d'obtenir ce chemin que celui étant $\{q_0, q_2, q_3, q_4, q_5\}$ qui nous est pas donné par le tableau de distribution suivant :

2.3 Essai 2 nombre pair de connexions dans un code automatisé

Puisque nous n'avons pas un graphe directionnel, le troisième terme des contraintes en ce qui concerne les points intermédiaires est à retravailler. Pour cela, nous allons poser une contrainte qui assure que chaque noeud a un nombre pair de connexions. Cette fois-ci, nous pouvons prendre une région du fibercub soit les 7 premiers noeuds de la matrice d'adjacence suivante :

$$\begin{bmatrix} 0. & 0.03623491 & 0. & 0.13946645 & 0.02477716 & 0. & 0. \\ 0.03623491 & 0. & 0.06762991 & 0.00076388 & 0.27605707 & 0. & 0. \\ 0. & 0.06762991 & 0. & 0. & 0. & 0.24649557 & 0.22642747 \\ 0.13946645 & 0.00076388 & 0. & 0. & 0.00281244 & 0. & 0. \\ 0.02477716 & 0.27605707 & 0. & 0.00281244 & 0. & 0.00310163 & 0. \\ 0. & 0. & 0.24649557 & 0. & 0.00310163 & 0. & 0.00471262 \\ 0. & 0. & 0.22642747 & 0. & 0. & 0.00471262 & 0. \end{bmatrix}$$

Cette matrice donne lieu à un sous-graphe du fibercup :



L'Hamiltonien que nous allons considérer cette fois-ci est :

$$f(x) = \sum_i w_i x_i + \alpha \left(\sum_{i \in D} x_i - 1 \right)^2 + \alpha \left(\sum_{i \in F} x_i - 1 \right)^2 + \alpha \sum_k \left(\left(\prod_{i \in K} Z_i - 1 \right)^2 \right)$$

Pour ce deuxième essai, j'ai procédé par une généralisation du code à partir d'une matrice aléatoire quelconque. La matrice aléatoire est générée par le code suivant :

https://github.com/harshana20011/quactography/blob/harshana_test/scripts/automatisation_harsh/generate_random_matrices.py

On enregistre ensuite une la matrice dans un fichier csv qu'on pourra lire pour extraire diverses informations : C'est ce que fait le code suivant pour attribuer à chaque arête du graphe un indice (qui

sera l'indice du qubit), détermine pour chaque qubit le noeuds qui sont associés au edge, les connexions possibles avec les autres noeuds voisins :

https://github.com/harshana20011/quactography/blob/harshana_test/scripts/automatisation_harsh/connexions_qubits.py

Ensuite, on peut écrire l'Hamiltonien en chaînes de Pauli. Pour ce faire, on procède de manière similaire à l'essai 1 en associant les x_i à $\frac{I-Z_i}{2}$. Une fois les qubits et les indices ainsi que toutes les informations sur le graphe extraites, en plus de la détermination du noeud de départ et de fin par l'utilisateur, il est possible d'écrire l'Hamiltonien sous forme de chaînes de Pauli. Pour se faire, on procède de manière similaire à l'essai 1 en associant les x_i à $\frac{I-Z_i}{2}$. Pour les termes intermédiaires, avec les lignes de la matrice d'adjacence excluant la diagonale et en ne considérant que la matrice triangulaire supérieure, on a accès aux connexions liées à chaque noeud intermédiaire, on peut donc utiliser ces connaissances pour faire des opérations sur des chaînes de caractères et bâtir nos termes en chaînes de Pauli. Voici le terme le plus complexe, soit le terme intermédiaire : Disons qu'on a un nombre de qubits étant à 4 : On initialise à "I"*4 = "IIII". Disons qu'on a 2 noeuds intermédiaires étant où le premier a comme connexions q0 et q2, tandis que le deuxième a comme connexions q3 et q1, On devrait retourner ceci :

$$h1_{int} = \text{SparsePauliOp.fromlist}([("IIII", -1), ("IZIZ", 1)])$$

$$h2_{int} = \text{SparsePauliOp.fromlist}([("IIII", -1), ("ZIZI", 1)])$$

Ce qui respecte l'équation du terme intermédiaire suivant : $\sum_K (\prod_{i \in I_K} Z_i - I)$ Le terme intermédiaire prendra donc l'allure suivante : $h_{int} = h1_{int}^2 + h2_{int}^2$

https://github.com/harshana20011/quactography/blob/harshana_test/scripts/automatisation_harsh/hamiltonian.py

Une fois qu'on a l'expression du Hamiltonien en chaînes de Pauli, il ne reste plus qu'à définir un alpha (plusieurs valeurs pour tester au besoin) et à sommer les différents termes dans le script principal et appliquer le circuit et l'algorithme de QAOA :

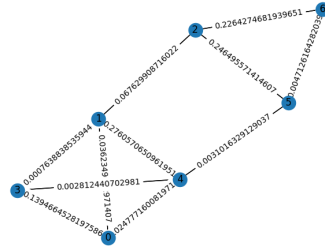
https://github.com/harshana20011/quactography/blob/harshana_test/scripts/automatisation_harsh/qaoa_path.py

Avec cela, nous pouvons tracer un graphique du coût minimal en fonction de la valeur du paramètre α afin de voir la entre les deux (linéaire ou pas etc.) et éventuellement voir s'il existe un paramètre de α optimal. Il resterait à visualiser les chemins optimaux associés à chaque coût et alpha afin de vraiment voir le choix idéal, automatiser la visualisation pour n'importe quel graphe, et tester de nouvelles méthodes avec COBYLA et autres optimiseurs en plus de simplement regarder de meilleures contraintes pour l'Hamiltonien. Voici quelques ajouts ayant été faits avec un main pour automatiser le tout avec multiprocessing pour les différentes valeurs de alpha : Voici le code pour la visualisation des chemins obtenus et afficher le graphes des coûts minimaux en fonction de la valeur de alpha :

https://github.com/harshana20011/quactography/blob/harshana_test/scripts/automatisation_harsh/visualize_paths_opt.py
https://github.com/harshana20011/quactography/blob/harshana_test/scripts/automatisation_harsh/alphas_min_graph.py
https://github.com/harshana20011/quactography/blob/harshana_test/scripts/automatisation_harsh/main.py

2.4 Résultats du code avec essai 2

Voici une image du graphe résultants :



Voici les informations concernant le graphe :

```
Connexions possibles depuis 0 au(x) noeud(s) : [1, 3, 4]
Connexions possibles depuis 1 au(x) noeud(s) : [0, 2, 3, 4]
Connexions possibles depuis 2 au(x) noeud(s) : [1, 5, 6]
Connexions possibles depuis 3 au(x) noeud(s) : [0, 1, 4]
Connexions possibles depuis 4 au(x) noeud(s) : [0, 1, 3, 5]
Connexions possibles depuis 5 au(x) noeud(s) : [2, 4, 6]
Connexions possibles depuis 6 au(x) noeud(s) : [2, 5]
Toutes connexions possibles (doublées) : [[1, 3, 4], [0, 2, 3, 4], [1, 5, 6], [0, 1, 4], [0, 1, 3, 5], [2, 4, 6], [2, 5]]

Arêtes depuis 0 au(x) noeud(s) : [1, 3, 4]
Arêtes depuis 1 au(x) noeud(s) : [2, 3, 4]
Arêtes depuis 2 au(x) noeud(s) : [5, 6]
Arêtes depuis 3 au(x) noeud(s) : [4]
Arêtes depuis 4 au(x) noeud(s) : [5]
Arêtes depuis 5 au(x) noeud(s) : [6]
Toutes connexions sans doublement: [[1, 3, 4], [2, 3, 4], [5, 6], [4], [5], [6]]

Indice : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Départ : [0, 0, 0, 1, 1, 1, 1, 2, 2, 3, 4, 5]
Destin : [1, 3, 4, 2, 3, 4, 5, 6, 4, 5, 6]
```

Voici l'écriture de chaque terme de l'Hamiltonien ainsi que l'expression finale de forme QUBO :

```
Coût obligatoire = SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[ 5.14239554e-01+0.j, -1.81174536e-02+0.j, -6.97332264e-02+0.j,
    -1.23885800e-02+0.j, -3.38149544e-02+0.j, -3.81941927e-04+0.j,
    -1.38028533e-01+0.j, -1.23247786e-01+0.j, -1.13213734e-01+0.j,
    -1.40622035e-03+0.j, -1.55081646e-03+0.j, -2.35630821e-03+0.j])

Qubit à sommer sur les x_i de départ: q([0, 1, 2]) - I

Contrainte de départ = SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[ 0.5+0.j, -0.5+0.j, -0.5+0.j, -0.5+0.j])

Qubit à sommer sur les x_i de fin: q([6, 9, 10]) - I

Contrainte de fin = SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[ 0.5+0.j, -0.5+0.j, -0.5+0.j, -0.5+0.j])
Liste de noeuds intermédiaires: [1, 2, 3, 4, 6]

Qubit à multiplier sur les x_i intermédiaires: q([0, 3, 4, 5])
Qubit à multiplier sur les x_i intermédiaires: q([3, 6, 7])
Qubit à multiplier sur les x_i intermédiaires: q([1, 4, 8])
Qubit à multiplier sur les x_i intermédiaires: q([2, 5, 8, 9])
Qubit à multiplier sur les x_i intermédiaires: q([7, 10])

Contrainte intermédiaire = [SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[-1+0.j, 1+0.j]), SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[-1+0.j, 1+0.j]), SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[-1+0.j, 1+0.j]), SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[-1+0.j, 1+0.j]), SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
    coeffs=[-1+0.j, 1+0.j])]
somme sur les termes intermédiaires au carré: SparsePauliOp(['IIIIIIIII', 'IIIIIIIIIZ', 'IIIIIIIZI', 'IIIIIIIZI'],
```

Voici les résultats pour $[0.5 \cdot poids_{total}, poids_{total}, 2 \cdot poids_{total}, \dots, 4 \cdot poids_{total}]$

```

message: Optimization terminated successfully.
success: True
status: 1
fun: 0.5409342712792186
x: [ 1.225e+00  6.990e-01]
nfev: 44
maxcv: 0.0
Minimum cost: 0.5409342712792186
message: Optimization terminated successfully.
success: True
status: 1
fun: 3.0172053258709584
x: [ 1.244e+00 -1.209e+00]
nfev: 38
maxcv: 0.0
Minimum cost: 3.0172053258709584
message: Optimization terminated successfully.
success: True
status: 1
fun: 6.391698755005148
x: [ 1.232e+00 -5.971e-01]
nfev: 41
maxcv: 0.0
Minimum cost: 6.391698755005148
message: Optimization terminated successfully.
success: True
status: 1
fun: 6.954489470387642
x: [ 1.239e+00 -1.918e+00]
nfev: 55
maxcv: 0.0
Minimum cost: 6.954489470387642
message: Optimization terminated successfully.
success: True
status: 1
fun: 9.116272128303226
x: [ 1.236e+00 -1.439e+00]
nfev: 81
maxcv: 0.0
Minimum cost: 9.116272128303226

```

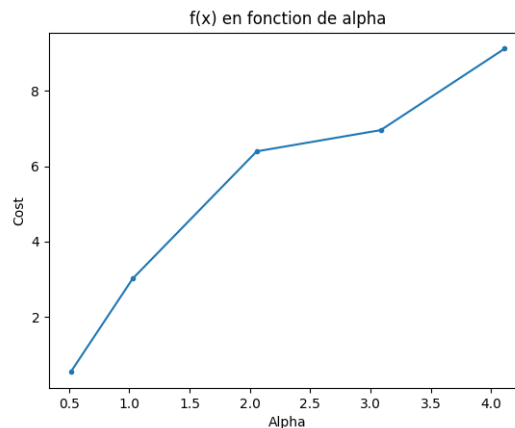
Voici le fichier dans lequel on enregistre les résultats des valeurs de alpha, le coût minimal de l'optimisation avec COBYLA ainsi que le chemin associé :

```

1 0.5142395537586716,0.5409342712792186,10101010010
2 1.0284791075173432,3.0172053258709584,11111111011
3 2.0569582150346863,6.391698755005148,11110111111
4 3.0854373225520293,6.954489470387642,11011110110
5 4.113916430069373,9.116272128303226,01110110010

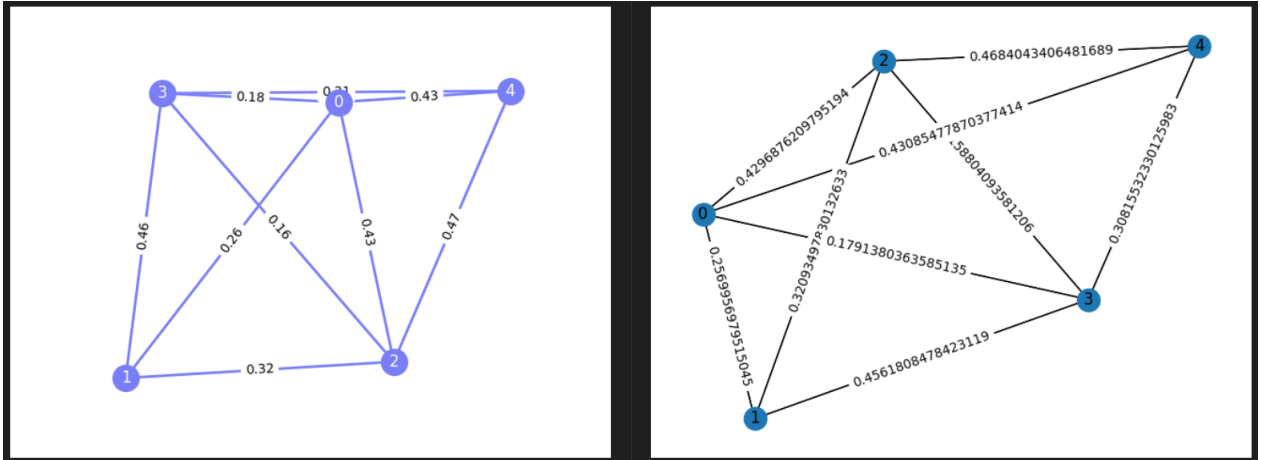
```

Voici le coût minimal en fonction de la valeur de alpha :



À améliorer : Il faut ajouter la visualisation des chemins optimaux, il faut vérifier l'écriture du hamiltonien et s'assurer que les termes intermédiaires contiennent les edges liés au départ et à la fin dans le produit de matrice de Pauli Z, il faut enregistrer les images de chemins ainsi que de fonction de alpha dans un fichier output avec la valeur de alpha, le chemin et le coût en affichant le chemin, apprendre à manipuler Networkx, ajouter multiprocessing. Correction : Les modifications ont été apportées dans les liens github présents dans la section Essai 2 nombre pair de connexions dans un code automatisé.

Vérification de la visualisation du graphe avec deux méthodes différentes soit rustworkx ou networkx :



3 Plus amples détails, Simulation de systèmes quantiques (Stage)

4 Information quantique(Cours)

Les QPU (quantum processor units) peuvent résoudre les problèmes plus rapidement que des cartes normales en exploitant les phénomènes de superposition et d'intrication. Éventuellement, avec les QPU, on vise à avoir une analyse personnelle du connectome de chaque individu ainsi qu'une analyse de prévention contre les pathologies neurologiques.

Une citation de Feynman : "Living things are made of atoms according to the laws of physics, and the laws of physics present no barrier to reducing the size of computers until bits are the size of atoms and quantum behavior holds sway."

Les qubits existent dans un état de superposition de toute valeurs possibles dans l'espace jusqu'à réduction du paquet d'onde. Les états peuvent être visualisés avec la sphère de Bloch. Boson sampling method utilisé en chimie quantique et application en théorie des graphes.

4.1 Histoire et vue globale

matrice définie positive (opérateur) : si $\langle \psi | A | \psi \rangle > 0 \forall |\psi\rangle \neq 0$. Une des questions intéressantes qu'a emmené l'information quantique est celle de déterminer si le clonage existe ; en théorie, si c'est le cas, alors il serait possible d'émettre un signal allant plus vite que la lumière ce qui serait en contradiction avec la théorie de la relativité d'Einstein. En 1980, la théorie du non-clonage a été découverte. Une autre avenue intéressante a été , dès les années 1970, de savoir comment avoir un contrôle sur un système quantique unique. Les ordinateurs classiques peuvent simuler un ordinateur quantique, mais les simulations ne sont pas efficaces, de telle sorte que les ordinateurs quantiques s'avèrent quant à eux, beaucoup plus rapides. L'efficacité d'un algorithme dépend de la complexité de calcul. (Polynomial plus efficace que Exponentiel).

Machine de Turing : On dit que tout algorithme peut être efficacement traité par la machine de Turing (probabiliste). En 1948, Shannon a travaillé sur d'importantes questions en sciences de l'information et de ce questionnement, en découle les résultats suivants : Noiseless channel coding theorem qui quantifie les ressources physiques nécessaires pour stocker la sortie d'une source d'information ainsi que le Noisy channel coding theorem, qui quantifie le nombre d'information pouvant être transmis dans une ligne avec bruit. Dans le but d'avoir une transmission avec le moins de bruits possibles, Shannon énonce l'idée de codes de corrections pour palier à cet effet de bruit.

Cryptographie quantique : Algorithme de Shor **ShorAlgo**L'algorithme de Shor est un des rares algorithmes avec un avantage sur les algorithmes classiques. L'algorithme est excellent pour deux choses : factorisé les nombres exponentiellement plus rapide que tous les autres algorithmes connus. Cela affecte les systèmes de crypto comme la RSA encryption.

L'intérêt de l'État y est puisque le développement d'un système de Cryptographie quantique permettrait la protection des informations confidentielles et bancaires. C'est pourquoi l'argent est dans le quantique ! La factorisation de nombres par l'algorithme de Shor fonctionne ainsi :

- On choisit un nombre entier aléatoire plus petit que le nombre à factoriser. Pour des nombres quantiquement sécuritaires, il faudrait un ordinateur quantique pour déterminer le plus grand commun diviseur. Le rôle de l'ordinateur quantique est de déterminer la période du nombre à factoriser, tel que l'algorithme détermine si un nouveau nombre aléatoire doit être testé pour trouver la solution.

La puissance des ordinateurs quantiques réside dans leur capacité à calculer en parallèle une superposition d'état si celui-ci a été dans un état intriqué. Il faut donc trouver la bonne séquence de portes logiques dans l'algorithme afin de pouvoir exploiter toute cette puissance de calcul et d'éviter

de perdre de l'information, de réduire au maximum le bruit avec des codes de correction d'erreurs etc. Les calculs en parallèle sont faits dans la méthode globale de tractographie où on se base sur les données à pleins d'endroits différents en tentant de relier chaque morceaux du cerveau dans sa généralité. Cette méthode n'est pas encore au point et a le risque de trop coller aux données et pas du tout aux contraintes biologiques connues par exemple.

Pour des détails sur la notation et le formalisme mathématique, voir annexe B.

4.2 Jeu GHZ

Dans tous les exemples qui suivent soit le jeu GHZ, l'encodage dense et la téléportation quantique, on commence par avoir un état intriqué afin de pouvoir faire profit du paradoxe EPR pour transmettre de l'information de manière particulière. Lorsque deux particules sont intriquées, l'information de l'un indique l'état de l'autre, peu importe la distance qui sépare ces deux qubits ou plus, intriqués. Ceci semble à premier abord violer la théorie de la relativité d'Einstein qui dit que rien ne puisse voyager plus vite que la lumière. Les particules intriquées sont créées au même lieu, lorsqu'on mesure leur spin par exemple, plus tard, on remarque que si une particule possède une orientation de spin, alors à coup sûr, l'autre particule possède un spin d'orientation opposée, en autant que l'angle des deux détecteurs soient le même. Le paradoxe EPR avait initialement pour but de montrer que l'interprétation de Copenhague et la superposition d'état est réduite uniquement lors d'une mesure. Pour montrer une absurdité, ils ont développé le concept d'intrication quantique. Ils supposent que deux particules sont intriquées, et que l'on effectue une mesure sur une des deux, étant donné la réduction du paquet d'onde qui décrit les deux particules, l'autre devrait être immédiatement affecté et tout ceci montrerait que l'information a voyagé plus vite que la lumière qui est en contradiction avec la théorie de la relativité. Pour palier à cela, EPR suggèrent la théorie des variables cachées, de telle sorte que l'état était déterminé d'avance. Toutefois en 1964, Bell énonça un moyen de prouver si la théorie des variables cachées est valide ou pas grâce à son équation d'inégalité de Bell :

$$P(Z, X) - P(Z, Q) - P(Q, X) \leq 1$$

Comme les probabilités respectent une relation sinusoidale plutôt que linéaire, alors la loi des variables cachées s'avère fausse, et l'inégalité de Bell également !

Jeu classique A,B,C et on demande que vaut X, que vaut Y. On gagne si XXX = -1 ou XYY, YXY, YYX = +1. En classique, on ne peut pas gagner si l'information ne circule pas ; il faut que les joueurs répondent 1 toujours pour gagner 3/4 du temps, sinon en résolvant le système d'équation d'optimisation , on obtient une absurdité montrant qu'il est impossible d'avoir une réponse gagnante avec une stratégie planifiée.

Par contre, dans un jeu quantique, où l'on pose la même question, mais cette fois-ci on prépare un état à 3 qubits intriqués soit l'état GHZ, et que si un joueur se fait demander X, il mesure X(matrice de Pauli) sur l'état et il obtient une réponse, s'il a Y, il mesure Y (matrice de Pauli) sur l'état. En faisant les calculs, on se rend compte qu'avec cette technique, il obtiendra toujours -1 s'il se fait demander XXX et 1 pour les autres combinaisons, puisque en développant au long, on s'aperçoit qu'il y a un nombre impair de signes moins pour XXX et un nombre paire de signes moins pour les autres.

4.3 Théorème de non-clonage

$$\begin{aligned} U_{copy} |\psi\rangle |0\rangle &= |\psi\rangle |\psi\rangle \\ U_{copy} |0\rangle |0\rangle &= |0\rangle |0\rangle \\ U_{copy} |1\rangle |0\rangle &= |1\rangle |1\rangle \end{aligned}$$

On applique U sur un état arbitraire et on trouve une absurdité :

5 Protocoles et communication quantique (Cours)

Les protocoles de cette section prennent en compte des systèmes à deux qubits de manière à être facilement réalisable physiquement avec la technologie du jour.

5.1 Distribution quantique de clés

La Cryptographie quantique se base sur les lois physiques pour distribuer des clés quantiques. Pour cela, deux chaînes de communication sont utilisées entre Alice et Bob ; celle qui est publique et celle qui est privée. Celle qui est publique est classique alors que celle qui est privée est plutôt quantique. Cette communication quantique se base en fait sur l'état de polarisation de photons. Pour que Alice et Bob puisse savoir ce que la clé contient, ils doivent faire une mesure. De cette manière, si Ève l'eavesdropper écoutait A et B, elle viendrait effectuer une mesure et perturberait le système de telle sorte qu'elle se fera remarquer si elle reste assez longtemps (trop pour que ce ne soit pas considéré comme du bruit).

Le premier protocole de Cryptographie quantique est BB84, celui de Bennet et Brassard. Ce protocole se repose sur trois principes :

1. Les états quantiques ne peuvent être copiés, ce qui empêche Ève de pouvoir interférer, écouter, puis copier ce qu'Alice avait envoyé et l'envoyer à Bob ni vu ni connu.
2. Effectuer une mesure sur le système viendra perturber ce-dernier.
3. Les mesures sont des processus irréversibles (car l'entropie augmente).

Pour commencer, dans ce protocole, Alice produit une chaîne de $2n$ qubits compris dans les choix $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ Alice envoie cette séquence aléatoire sur un channel quantique à Bob et Bob mesure chaque qubit aléatoirement dans la base 0,1 ou $+/ -$. Il y a donc Alice qui choisit une base aléatoirement, et Bob également. Par les lois de la probabilité, il devrait y avoir n qubit dans la représentation 0,1 et n autres dans la représentation $+/ -$. Alice et Bob regardent à quel qubit ils ont choisit la même base et ils gardent uniquement celles qui ont été mesurées dans la même base, car leur mesure seront corrélées. Les qubits gardés s'appellent sifted key.

Ensuite, Alice et Bob peuvent vérifier s'il y a des erreurs et à partir d'un certain taux, ils peuvent utiliser une autre clé pour se parler. Voici un exemple qui montre qu'Ève ne peut pas deviner le message d'Alice avec une porte CNOT :

On suppose qu'Alice a un état $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

Ève prépare un état $|0\rangle$ et crée l'état $|0\rangle = |+\rangle \otimes |0\rangle = \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle}{\sqrt{2}}$

Si Ève applique un CNOT à l'état en utilisant $|0\rangle$ comme cible

$$\text{CNOT}(|+\rangle \otimes |0\rangle) = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

De cette sorte qu'une mesure de A sera la même qu'Ève, dans la base $|0\rangle, |1\rangle$.

Si maintenant, Alice fait une mesure dans la base $|+\rangle, |-\rangle$

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \left[\frac{(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)}{2} + \frac{(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{2} \right]$$

* ne pas oublier $\frac{1}{\sqrt{2}}$ pour la conversion

$$= \frac{1}{\sqrt{2}} \left[\frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} + \frac{|00\rangle - |01\rangle - |10\rangle + |11\rangle}{2} \right]$$

$$= \frac{1}{\sqrt{2}} \left[\frac{|00\rangle + |11\rangle}{2} \right]$$

Les résultats sont encore corrélés.

Si Alice avait plutôt un état $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$, alors Ève aurait le résultat opposé à Alice

→ Comme Ève ne sait pas si Ève avait initialement l'état $|+\rangle$ ou $|-\rangle$, il n'y a aucun moyen qu'elle puisse deviner l'état d'Ève avec une CNOT

Il existe un deuxième protocole en Cryptographie quantique qui est celle d'Ekert en 1991 où Alice et Bob choisissent un état de Bell pour commencer. Ils prennent chacun un qubit de la paire intriquée et choisissent une base aléatoire dans laquelle faire leur mesures. Lorsqu'ils ont choisi la même base, ils gardent le qubit et font leur sifted key avec ceux issus de la même base lors de la mesure. Comme leur résultats sont corrélés à chaque fois, ils peuvent aisément déterminer si Ève écoute ou non et il existe des codes de correction pour les erreurs qui seraient présentes.

5.2 Encodage dense

Ce protocole vise à partager deux bits classique à travers la transmission d'un seul qubit issue de la paire intriquée.

#1 :

Alice et Bob prépare un état à deux qubits intriqués, soit en partant chacun de $|0\rangle$, en appliquant la porte Hadamard au qubit d'Alice et une porte CNOT où le contrôle est à Alice et la cible est Bob. Après cela, ils ont l'état de Bell $|\phi^+\rangle$. Ensuite, A et B se séparent.

#2 :

Si Alice veut envoyer le message 00, alors elle ne fait rien sur son qubit. Si elle veut envoyer le message 01, elle applique $(X \otimes I)$, si elle veut envoyer

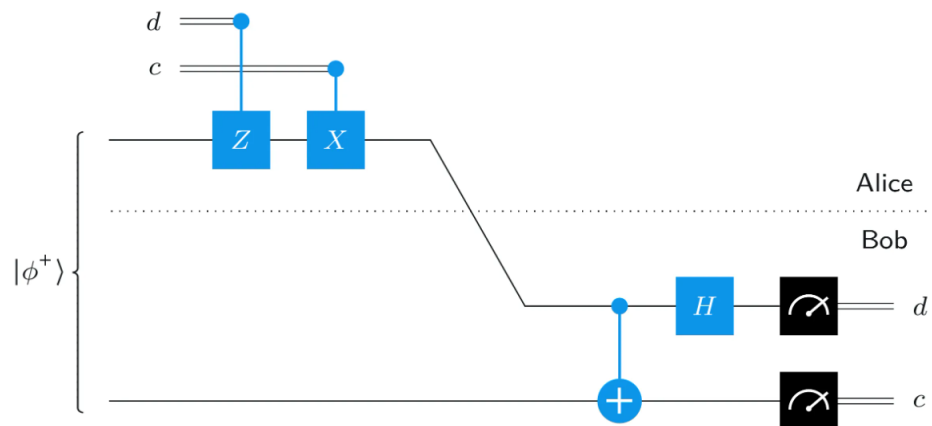
10, elle applique $(Z \otimes I)$ et si elle souhaite envoyer 11, alors elle applique $(iY \otimes I)$ ou bien XZ. Ensuite, elle envoie son qubit à Bob.

#3 :

Bob reçoit le qubit d'Alice et applique une porte CNOT avec comme contrôle le qubit d'Alice et la cible son qubit issu de la paire intriquée. Ensuite, il applique une porte Hadamard au qubit d'Alice.

#4 :

Il mesure le qubit envoyé par Alice (après avoir subi H) et obtient le premier bit classique, et il mesure son qubit à lui pour obtenir le deuxième bit classique.



5.3 Téléportation quantique

Ce protocole vise à transmettre un état quantique quelconque à Bob, soit $\chi = \alpha|0\rangle + \beta|1\rangle$, on suggère que l'état est normalisé, et que la somme des modules des coefficients au carré donne 1.

#1 :

Alice et Bob prépare un état à deux qubits intriqués, soit en partant chacun de $|0\rangle$, en appliquant la porte Hadamard au qubit d'Alice et une porte CNOT où le contrôle est à Alice et la cible est Bob. Après cela, ils ont l'état de Bell $|\phi^+\rangle$. Ensuite, A et B se séparent.

#2 :

Alice applique une porte CNOT avec l'état $|\chi\rangle$ en contrôle et sa paire intriquée comme cible. (La porte CNOT fait en sorte que lorsque le contrôle vaut 1, la cible est flipée.) Ici, l'état du système vaut $|\chi\rangle \otimes |\phi^+\rangle$

On en prend la porte CNOT avec le contrôle étant le premier qubit et le second étant le qubit d'Alice en cible. On ne fait rien au qubit de Bob, en troisième place.

#3 :

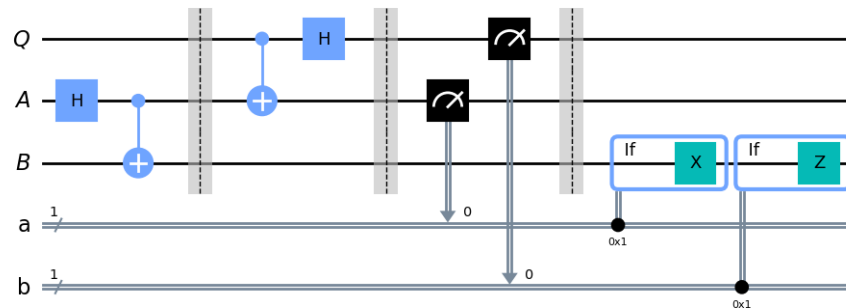
Alice applique une porte Hadamard sur le nouvel état du système, en se rappelant que :

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

#4 :

Alice fait une mesure sur sa paire intriquée ainsi que l'état qu'elle souhaite partager à Bob. Pour ceci, il faut d'abord réécrire l'état de l'étape précédente dans la base standard à 2 qubits $\{00, 01, 10, 11\}$. Ceci fait naturellement apparaître 4 choix potentiels de sa mesure. Si elle envoie 00 à Bob, alors elle a mesuré $I|\chi\rangle$, si elle envoie 01, alors Bob devra appliqué la porte X sur son qubit pour retrouver l'état initial, si elle envoie 10, il devra appliqué la porte Z, et si elle envoie 11, il devra appliqué la porte X, suivi de Z. En circuit, on peut donc montrer ces opérations par des portes Z et X conditionnels, avec le contrôle étant le bit classique 0 ou 1 envoyé par Alice.



6 Circuits quantiques (Cours)

Les circuits ont pour but de manipuler de l'information. En combinant un ensemble de portes logiques universel, il est possible de recréer toutes les opérations connues. Un ensemble de portes formant un protocole se nomme un circuit digital. L'action des portes logiques peuvent être représentée dans une table de vérité.

6.1 Circuits classiques et complexité

Il existe la porte NOT qui inverse le bit (X), la porte AND, la porte XOR (\oplus) qui vaut 1 uniquement si le bit A et différent de B, la porte NAND (L'inverse de AND). La porte NAND à elle seule constitue

un ensemble universel de porte puisqu'un ordinateur peut entièrement être construit avec cette porte. La porte Nand est irréversible. Un exemple de porte réversible est la porte Fredkin. 1 bit est un bit de contrôle, et les deux autres seront swiched si et seulement si le bit de contrôle vaut 1. Il existe également la porte Toffoli qui fait une porte AND sur les deux bit initiaux, et applique une porte XOR sur le résultat du AND ainsi qu'un bit cible :

C1	C2	T	T'
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Complexité : On classe les problèmes en fonction de la taille $C_n = |C_n|$:

On a le temps polynomial (Problèmes faciles pour lesquels $|C_n| \leq n^\alpha$), les problèmes NP (Non-deterministic polynomial time) problème où vérifier la réponse est facile (problème du plus court chemin). Ces ordres sont indépendants des ensembles de portes universels ou encore des ordinateurs utilisés. On a NP-difficile : Au moins aussi dur que le problème le plus difficile dans NP, et NP-complet tel que une solution à un des problèmes complets donne la solution à tous les autres de la catégorie (tous problèmes NP).

6.2 Calculs réversibles

Si une porte fait passer 2 qubits en 1 on a une porte irréversible. Toutefois, il est toujours possible d'exprimer un calcul irréversible comme un calcul réversible :

$$g(x, 0^m) \rightarrow g(x, f(x))$$

Le travail pour effacer un qubit est le suivant : $W \geq K_B T \ln 2$

6.3 Circuits quantiques et portes logiques

Les circuits quantiques se divisent en trois parties : l'état initial, l'ensemble de portes et la mesure. On peut choisir les portes dans un ensemble universel de portes. La taille du circuit donnera sa complexité. En quantique, les portes logiques sont en fait des opérateurs unitaires. Les opérateurs unitaires possèdent des propriétés intéressantes comme le fait que $U^\dagger = U$ et $\exp(iHt)$ est aussi unitaire en autant que H le soit aussi. En fonction du nombre de qubit, la matrice prendra les dimensions 2^n , soit pour un qubit une matrice 2 par 2. En quantique, pour inverser les qubit 0 à 1 et vice-versa, on passe par l'opérateur de Pauli X, qui agit comme porte NOT.

La porte NOT peut être déduite par l'égalité suivante :

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \langle 0|X|0 \rangle & \langle 0|X|1 \rangle \\ \langle 1|X|0 \rangle & \langle 1|X|1 \rangle \end{bmatrix}$$

Pour satisfaire cette égalité, $X = |0\rangle\langle 1| + |1\rangle\langle 0|$

On a l'action de l'opérateur Z qui change la phase de l'état sur le deuxième élément dans la base $\{|0\rangle, |1\rangle\}$

$$Z|j\rangle = (-1)^j|j\rangle$$

Une matrice de changement de phase par un angle quelconque est défini comme

$$Phase_{shift} = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\theta) \end{bmatrix}$$

On peut appliquer des rotations dans la sphère de Bloch de telle sorte à appliquer un changement de phase. D'abord, rappelons un état quelconque $|\psi\rangle = \cos\theta|0\rangle + e^{i\phi}\sin\theta|1\rangle$. On peut écrire la matrice P dans la base standard : $P = |0\rangle\langle 1| + e^{i\gamma}|1\rangle\langle 0|$ Dans cette représentation, on voit bien comment Z est en réalité un cas spécifique de P , où on pose $\phi = \pi$. Ainsi, avec un angle de par exemple $\pi/4$, on peut obtenir la porte T etc.

$$P|\psi\rangle = \cos\theta|0\rangle + \exp(i(\gamma + \phi))\sin\theta|1\rangle$$

Dans la même logique, la matrice de Hadamard s'écrit : $H = \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)$. De manière générale, la porte H a cet effet sur un état $|\psi\rangle$ quelconque :

In general, the Hadamard gate takes the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ into the state

$$H|\psi\rangle = \left(\frac{\alpha + \beta}{\sqrt{2}}\right)|0\rangle + \left(\frac{\alpha - \beta}{\sqrt{2}}\right)|1\rangle \quad (8.17)$$

This means that the probability of finding the qubit in the state $|0\rangle$ is changed from

$$|\alpha|^2 \text{ to } \left|\frac{\alpha + \beta}{\sqrt{2}}\right|^2 = \left(\frac{\alpha^* + \beta^*}{\sqrt{2}}\right)\left(\frac{\alpha + \beta}{\sqrt{2}}\right) = \frac{1}{2}(|\alpha|^2 + |\beta|^2 + \text{Re}(\alpha\beta^*))$$

and similarly for the probability of finding the system in the $|1\rangle$ state. We can regroup the terms in (8.17) to give another interpretation of the output of a Hadamard gate:

$$H|\psi\rangle = \alpha \frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \alpha|+\rangle + \beta|-\rangle \quad (8.18)$$

That is, the Hadamard gate has turned a state that, with respect to the standard or computational basis, had the probability $|\alpha|^2$ of finding the system in the state $|0\rangle$ and the probability $|\beta|^2$ of finding the system in the state $|1\rangle$ into a state that has the probability $|\alpha|^2$ of finding the system in the state $|+\rangle$ and the probability $|\beta|^2$ of finding the system in the state $|-\rangle$.

(image tirée du livre de David MacMahon)

Pour un opérateur à un qubit donné, tout opérateur de ce type peut s'exprimer sous la forme :

$$U = \exp(ia)R_z(b)R_y(c)R_z(d), \text{ a,b,c,d des nombres réels.}$$

La porte Y a pour effet : $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \rightarrow |\psi'\rangle = -i\beta|0\rangle + i\alpha|1\rangle$ Finalement, lorsqu'on effectue une mesure, la probabilité d'être dans un état suite à la mesure est égale au module carré du coefficient devant l'état, comme le dit la règle de Born.

Pour un système de deux qubits, il y a toutes les combinaisons de paire possible avec 00, 01, 10, 11 en bra comme en ket, ce qui donne une matrice 4 par 4. La porte $CNOT = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|$

Lorsqu'on fait la conception de circuits, on va souvent miser sur la décomposition d'opérateurs unitaires à un seul qubit. Il peut être montré que $X = HZH$.

Voici une série d'identités utiles :

$$XYX = -Y$$

$$HXH = Z$$

$$HZH = X$$

$$HYH = -Y$$

Il existe également des portes de contrôle qui, appliqué à un ket quelconque donne :

$$|c\rangle |t\rangle = |c\rangle U^c |t\rangle$$

Dans ce cas, U agit sur le deuxième qubit uniquement si le premier qubit est à l'état 1.

Cas particulier : en ce qui concerne la porte CZ, peu importe le qubit cible ou de contrôle, les deux doivent valoir 1 afin que l'opérateur Z puisse agir. Preuve :

$$CZ |00\rangle = |00\rangle$$

$$CZ |01\rangle = |01\rangle$$

$$CZ |10\rangle = |10\rangle$$

$$CZ |11\rangle = -|11\rangle$$

C'est pourquoi on peut dessiner cette porte de contrôle comme un point autant sur le qubit de contrôle que sur celui qui est ciblé.

6.4 Ensemble universel

Voici des ensembles de portes universels en informatique quantique :

$$\{H, T, CNOT\}$$

$$\{H, T, CNOT, R_y(\theta)\}$$

$$\{H, T, CNOT, R_y(\theta), R_z(\theta)\}$$

$$\{H, T, CNOT, R_y(\theta), R_z(\theta), R_x(\theta)\}$$

Un ensemble universel est suffisant pour la conception de toutes portes quantiques (opérateurs unitaires), existants. Les ensembles de Clifford (H, CNOT, S) ne sont pas des ensembles universels. Pour qu'un ensemble soit universel, il faut minimalement avoir une porte à 2 qubits qui permet les superpositions (ex. H), une porte CNOT ou bien Toffoli pour l'intrication, une porte complexe (ex. une matrice de rotation S, T).

7 Algorithmes quantiques(Cours)

En informatique quantique, un concept utile : le modèle de requête. On suggère une entrée connue encoder sous forme de chaîne de bits dont la totalité de l'information n'est pas donnée au calcul, il s'agit plutôt d'une fonction accessible sous forme de requêtes. Ce type d'entrée est souvent appelé une boîte noire ou un oracle. Suite à une requête, une partie seulement de la chaîne de bits de l'entrée est accessible. L'efficacité de ce type de modèle est basé sur le nombre de requêtes de l'algorithme. Il existe

des problèmes à recherche unique tel qu'il y ait une seule solution possible telle que la fonction $f(x)$ pour un x précis donne 1, et 0 pour tous les autres cas. À ce moment, on cherche cette seule valeur de x étant la solution au problème.

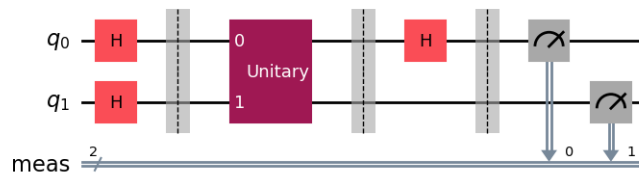
7.1 Circuits quantiques élémentaires

Pour réaliser un circuit dans la veine du modèle de requête, il faut d'abord des portes de requête qui calculent la fonction $f(x)$ prenant des entrées de différentes valeurs de x . Si on met des portes de requête qui font une requête, alors la complexité de l'algorithme et du circuit est déterminé simplement par le nombre de portes du circuit qui font une requête. Toutefois, implémenter des portes n'est pas suffisant, il faut des portes unitaires pour pouvoir les appliquer sur des états quantiques. Dans ces cas, la porte unitaire U_f est toujours une matrice de permutation. (Chaque colonne un élément valant 1 unique, et les autres à zéro)

7.2 Algorithme de Deutsch

L'algorithme de Deutsch a pour but de résoudre le problème de parité dans le cas le plus simple. Prenons des fonctions constante (met tout à un ou tout à zéro) et les fonctions balancées (ne rien changer ou permuter 0 à 1 et vice-versa), pour les différencier, un algorithme classique aurait besoin de tester $f(0)$ suivi de $f(1)$ soit deux requêtes pour déterminer si f est constante ou balancée, tandis qu'un algorithme quantique peut le déterminer en une seule requête.

Voici le circuit :



Supposons l'état initial $|\psi_0\rangle = |01\rangle$ et que l'action de U_f est

$$U_f |xy\rangle = |x, y \oplus f(x)\rangle$$

Gardons en tête cette équation très utile :

$$|0 \oplus a\rangle - |1 \oplus a\rangle = (-1)^a(|0\rangle - |1\rangle)$$

$$|\psi_1\rangle = H_0 \otimes H_1 |01\rangle = |+-\rangle \quad (1)$$

$$|\psi_2\rangle = U_f |\psi_1\rangle = U_f |+-\rangle \quad (2)$$

$$= U_f \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \quad (3)$$

$$= \frac{U_f}{2} \left((|0\rangle - |1\rangle) |0\rangle + (|0\rangle - |1\rangle) |1\rangle \right) \quad (4)$$

$$= \frac{1}{2} |0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle |0\rangle + |0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle |1\rangle \quad (5)$$

$$= \frac{1}{2} (-1)^{f(0)} (|0\rangle - |1\rangle) |0\rangle + (-1)^{f(1)} (|0\rangle - |1\rangle) |1\rangle \quad (6)$$

$$= \frac{1}{\sqrt{2}} ((-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle) |-\rangle \quad (7)$$

$$|\psi_3\rangle = H_0 \psi_2 \quad (8)$$

$$= \frac{1}{\sqrt{2}} ((-1)^{f(0)} |+\rangle + (-1)^{f(1)} |-\rangle) |-\rangle \quad (9)$$

$$(10)$$

En développant les ket $+$ et $-$ du premier registre, on obtient une probabilité totale de mesurer $|0\rangle$ au premier registre si la fonction est constante, et $|1\rangle$ si la fonction est équilibrée.

7.3 Algorithme de Deutsch-Jozsa

7.4 Problème de Bernstein-Vazirani

L'algorithme de Bernstein-Vazirani est un cas appliqué de l'algorithme de Deutsch-Jozsa, avec cette fois-ci $f(x)$ qui vaut $a \cdot x$.

7.5 Transformée de Fourier quantique

La transformée de Fourier quantique est un élément qui sert de sous-routine à des algorithmes plus avancés. C'est une des catégories d'algorithmes qui est plus rapide que les algorithmes classiques, la deuxième catégorie est celle de Grover qui concerne les problèmes non structurés, plus généraux mais avec un gain quadratique plutôt qu'exponentiel comme pour la Transformée de Fourier quantique.

Plus précisément, l'algorithme de la Transformée de Fourier quantique permet de trouver plus facilement la périodicité d'une fonction, ce qui est parfois très difficile à faire pour des fonctions complexes classiquement. Cet algorithme n'est qu'en fait qu'un cas appliqué de l'algorithme d'estimation de phase que nous allons voir dans la sous-section suivante.

7.6 Estimation de phase

Entrées : un premier registre à t qubits qui encodera la valeur propre associée au vecteur propre en entrée au deuxième registre (état propre des portes CU appliqué tel qu'un retour de phase est effectué)
Nombre de qubits en entrée : $t = n + \log(2 + 1/2\epsilon)$

On applique la QFT inverse ensuite et on obtient la phase de chaque qubits qui donne la valeur propre approximée de l'état propre d'entrée.

7.7 Algorithme de factorisation de Shor

Prenons N un nombre que nous souhaitons factoriser en produit de deux nombres premiers, On choisit x quelconque inférieur et co-premier à N et on cherche r tel que $x^r = 1(mod N)$, considérer comme un problème difficile en classique.

En entrée, nous avons une boîte $U_{x,N}$ qui transforme $|j\rangle |k\rangle$ en $|j\rangle |x^j k mod N\rangle$. En sortie, nous avons le plus petit entier r tel que $x^r = 1(mod N)$. Soit, on commence avec un état initial $|0\rangle |1\rangle$, auquel on crée une superposition avec la porte Hadamard à n qubits, et auquel on applique la boîte noire résultant en

$$\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |x^j mod N\rangle$$

De cela, nous pouvons faire l'approximation suivante :

$$\frac{1}{\sqrt{r 2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} \exp(2i\pi s j / r) |j\rangle |u_s\rangle$$

$$u_s = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-2i\pi s k}{r}\right) |x^k mod N\rangle$$

Ensuite, on applique la transformée de Fourier inverse pour avoir $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |s/r\rangle |u_s\rangle$ De cela, en mesurant le premier registre, on peut extraire s/r et en appliquant l'algorithme de fractions continues, on peut trouver r .

Box 5.3: The continued fractions algorithm

The idea of the continued fractions algorithm is to describe real numbers in terms of integers alone, using expressions of the form

$$[a_0, \dots, a_M] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_M}}}}, \quad (5.49)$$

where a_0, \dots, a_M are positive integers. (For applications to quantum computing it is convenient to allow $a_0 = 0$ as well.) We define the m th convergent ($0 \leq m \leq M$) to this continued fraction to be $[a_0, \dots, a_m]$. The *continued fractions algorithm* is a method for determining the continued fraction expansion of an arbitrary real number. It is easily understood by example. Suppose we are trying to decompose $31/13$ as a continued fraction. The first step of the continued fractions algorithm is to split $31/13$ into its integer and fractional part,

$$\frac{31}{13} = 2 + \frac{5}{13}. \quad (5.50)$$

Next we invert the fractional part, obtaining

$$\frac{31}{13} = 2 + \frac{1}{\frac{13}{5}}. \quad (5.51)$$

These steps – split then invert – are now applied to $13/5$, giving

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{5}} = 2 + \frac{1}{2 + \frac{1}{2}}. \quad (5.52)$$

Next we split and invert $5/3$:

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{3}}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{2}}}. \quad (5.53)$$

The decomposition into a continued fraction now terminates, since

$$\frac{3}{2} = 1 + \frac{1}{2} \quad (5.54)$$

may be written with a 1 in the numerator without any need to invert, giving a final continued fraction representation of $31/13$ as

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}}. \quad (5.55)$$

It's clear that the continued fractions algorithm terminates after a finite number of 'split and invert' steps for any rational number, since the numerators which appear (31, 5, 3, 2, 1 in the example) are strictly decreasing. How quickly does this termination occur? It turns out that if $\varphi = s/r$ is a rational number, and s and r are L bit integers, then the continued fraction expansion for φ can be computed using $O(L^2)$ operations – $O(L)$ 'split and invert' steps, each using $O(L^2)$ gates for elementary arithmetic.

Tiré du Nielsen and Chuang

La recherche d'ordre est très liée à la décomposition en facteur premier : Il est possible de trouver un facteur de N s'il existe une solution non triviale à l'équation $x^2 = 1(mod N)$, et qu'on puisse obtenir un nombre co-premier à N qui a tendance à avoir un ordre r pair de telle sorte que $x = y^{r/2}(mod N)$ ne soit pas une solution triviale à $x^2 = 1(mod N)$.

Une solution non triviale mène à ce que soit $gcd(x-1, N)$ ou $gcd(x+1, N)$ soit un facteur non-trivial de N .

7.8 Algorithme de Grover

8 Codes de correction d'erreurs(Cours)

8.1 Condition de Knill-Laflamme

8.2 Code de Shor

8.3 Discrétisation des erreurs

-Final

A Annexe A :Reconstruction d'image quantique (Kiani et al.)

- Les algorithmes de reconstruction d'images utilisent les relations entre une image cible (ou une fonction cible), ainsi que sa représentation dans l'espace des fréquences (Fourier). La pertinence des algorithmes quantiques est qu'ils sont exponentiellement plus efficaces que les algorithmes classiques. Le fait d'utiliser le quantique permet de reconstruire une image à partir d'une fonction d'onde plutôt que les données ce qui pourrait prendre moins de temps et demander de plus petites doses de radiation.

- En fonction du type de données collectées, il existe différents algorithmes. Il y a une première catégorie qui reconstruit les images recueillies dans l'espace des fréquences (k-espace) en MRI. On utilise donc la Transformée de Fourier inverse. La deuxième catégorie vise à reconstruire les images à partir de CT/PET (Computed Tomography et Positron emission Tomography) scan ; on reconstruit les images à partir d'un ensemble de projections (intégrales de ligne sur une fonction). On utilise la transformée de Radon inverse dans ce cas-ci.

Pour le CT, les fonctions caractérisent les coefficients d'atténuation alors que pour le PET, il s'agit de fonctions caractérisant les concentrations d'un traceur radioactif. La Transformée de Radon dans ce contexte consiste à retourner des intégrales de ligne de ces fonctions à des angles spécifiques. Un algorithme de reconstruction prend la Transformée de Radon d'une fonction F et retourne une fonction G qui ressemble à F . La méthode utilisée par Kiani et al. est celle donnée par "The Fourier Slice Theorem".

Voici comment on peut appliquer la transformée de Radon à une fonction :

$$\mathcal{R}F(x, y) = f(\rho, \theta) = \iint F(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy$$

Il s'agit ici d'une intégrale de ligne le long d'une ligne $x \cos \theta + y \sin \theta = \rho$. Le but de l'algorithme est de trouver une fonction $G(x, y)$ telle que $G(x, y) \approx F(x, y)$.

La transformée de Radon et les mathématiques de l'imagerie médicale **Beatty** On part de quelques hypothèses et définitions mentionnées dans la thèse de Beatty :

- Les rayons-X ont tous la même énergie
- Tous les rayons-X ont une largeur nulle
- Les rayons ne sont pas assujettis aux phénomènes de réfraction et de diffraction (bien que des algorithmes de correction existent pour remédier à ce problème.)

Définitions :

L'intensité en fonction de la propagation du nombre de photon et l'énergie est donnée par la relation :

$$I(x) = N(x) \cdot E$$

Le coefficient d'atténuation est donné par : $A(x)$

La relation entre l'intensité du rayon entrant avec le rayon sortant sont donnés par la loi de Beer-Lambert :

$$I(x) = \exp -A(x)x$$

On a donc une décroissance exponentielle de l'intensité en fonction de la distance. On a la relation suivante et il s'agit de ce que font les algorithmes de CT scan. :

$$\ln \frac{I_0}{I_1} = \int_{x_0}^{x_1} A(x) dx$$

Pour décrire une ligne, il n'est pas pratique de prendre un plan cartésien, car dans un tel plan, les lignes verticales posent un problème avec leur pente infinie, tandis que le plan polaire n'est pas

pratique pour des systèmes qui dépendent de lignes parallèles. On prend donc une paramétrisation point normal :

$$l_{t,\theta} = \{z \in \mathbb{R}^2 : \langle z, (\cos\theta, \sin\theta) \rangle = t\}$$

Avec l'ensemble de points contenant la ligne :

$$l_{t,\theta} = (x(s), y(s)) = \{(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta) : s \in \mathbb{R}\}$$

Ce qui est mesuré physiquement est en fait le logarithme du rapport des intensités initiale sur finale, ce qui consiste en la transformée de Radon. On veut donc trouver la transformée de Radon inverse afin de retrouver la fonction initiale.

- (i) $\mathcal{R}(\alpha f + \beta g) = \alpha \mathcal{R}f + \beta \mathcal{R}g$
- (ii) $\mathcal{R}f(t, \theta) = \mathcal{R}f(-t, -\theta)$
- (iii) $\mathcal{R}f(t, \theta) = \int_{-\infty}^{\infty} f(x(s), y(s)) ds = \int_{-\infty}^{\infty} f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta) ds$

De plus, on ne considère que les fonctions intégrables entièrement.

La transformée de Fourier

On passe en fréquence :

$$Ff(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$$

$$F^{-1}f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega) e^{i\omega x} d\omega$$

$$F^{-1}Ff(x) = f(x)$$

On peut faire la transformée de Fourier en deux dimensions, dans quel cas on obtient une double intégrale.

Il reste le théorème de la tranche centrale, ainsi que les méthodes de résolution du problème de la reconstruction d'image avec l'hamiltonien et les algorithmes variationnels quantiques. (Hamiltonien et méthodes d'interpolation en passant par le théorème de la tranche centrale, et autre formule de type Fourier.)

B Annexe B : Formalisme mathématique en information quantique

Parallélisme quantique : capacité à effectuer des calculs sur tous les états de la superposition en même temps.

Réduction du paquet d'onde : Lors d'une mesure, on a bel et bien un état défini et le but à chaque mesure, une fois que la fonction s'est effondrée, c'est que l'état reste le même, bien que difficile à réaliser expérimentalement.)

Structure	Avantages	Inconvénients
Qubits supraconducteurs	Technique la plus répandue versatile et rapide	-
Ions piégés	-	Temps de vie plus lent, difficile à piéger Et si on mesure et on a un état excité, si on attend trop longtemps, même si on ne perturbe pas le système, ce-dernier aura tendance à revenir à l'état fondamental
Qubits de spin	Up ou Down sert à encoder l'information	-
Qubits topologiques	matériaux exotiques	-
Fibre optique	longue vie, source illimitée (suffit d'un rayon de lumière/ photons)	Induit peu d'interaction (peu d'interférences et on perd l'avantage quantique)

États de Bell : Les états de Bell sont des états quantiques de deux qubits intriqués. Les états de Bell peuvent être généralisés pour des système à plus de deux qubits par ce qu'on appelle les états GHZ. Ces états sont utiles car ils sont utilisés dans les circuits de téléportation quantique et d'encodage dense.

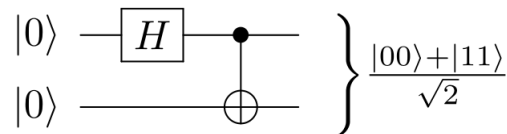
$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)(1)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B - |1\rangle_A \otimes |1\rangle_B)(2)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |1\rangle_B + |1\rangle_A \otimes |0\rangle_B)(3)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B)(4)$$

Pour fabriquer un état de Bell à partir d'un état initial de qubit à 0, il faut passer par ce circuit :



Pour fabriquer n'importe quel état de Bell à partir de kets initiaux quelconques :

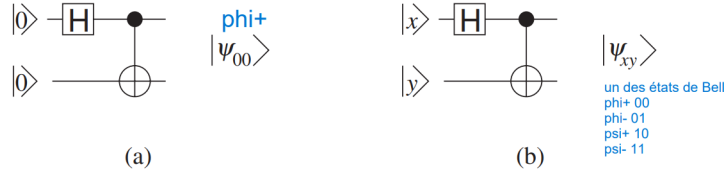


Fig 6.1 (a) A circuit that creates the entangled state $|\psi_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ from the unentangled computational-basis state $|00\rangle$. (b) A circuit that creates the four orthonormal entangled Bell states $|\psi_{xy}\rangle$ from the unentangled computational-basis state $|xy\rangle$.

(tiré du livre David Mermin Quantum Computer science an Introduction p.137)

On peut ainsi passer d'un état de Bell à l'autre appliquant les portes X ou/et Z.

La base et l'orthogonalité entre les états ϕ et ψ : Les états de Bell forment une base et le produit scalaire entre deux états donne le delta de Kronecker, soit :

$$\langle\Phi|\Phi\rangle = 1 \quad \langle\Psi|\Psi\rangle = 1 \quad \langle\Phi|\Psi\rangle = 0$$

La base standard pour un système de deux qubits : $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

De plus, les états de Bell sont des états propres de XX et de ZZ associés aux valeurs propres ± 1 .

Corrélation : Dépendamment de l'état choisi (un des quatre états de Bell), il y a des moyens que deux personnes distantes puissent mesurer exactement le même résultat si les deux prennent une paire de l'état intriqué. Si l'état initial est l'état $|\Phi^+\rangle$, alors les deux personnes vont mesurer exactement le même résultat. Si l'état de départ est plutôt $|\phi^-\rangle$, alors pour que la deuxième personne puisse mesurer avec certitude la même chose que la personne première, alors cette deuxième personne devra appliquer une porte Z (changement de phase global) avant de mesurer.

Voici donc un résumé des portes à appliquer pour tous les états de Bell :

États de Bell	Base dans laquelle B doit mesurer
$ \Phi^+\rangle$	$\mathbb{I} \cdot b_1$
$ \Phi^-\rangle$	$Z \cdot b_1$
$ \Psi^+\rangle$	$X \cdot b_1$
$ \Psi^-\rangle$	$XZ \cdot b_1$

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

$$|1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)$$

Matrice de Pauli σ_x :

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Matrice de Pauli σ_y :

$$\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Matrice de Pauli σ_z :

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

De plus, $\sigma_x|\pm\rangle = \pm|\pm\rangle$, et X représente aussi une porte NOT.

$$R_x(\theta) = \exp(-i\theta X/2)$$

$$R_y(\theta) = \exp(-i\theta Y/2)$$

$$R_z(\theta) = \exp(-i\theta Z/2)$$

En utilisant la définition en série de l'exponentielle, nous avons $e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$. Nous avons de plus, $e^{i\theta A} = I + i\theta A - \frac{(\theta A)^2}{2!} - \frac{i(\theta A)^3}{3!} + \dots$. Dans le cas particulier où $A^2 = I$, nous obtenons finalement $e^{i\theta A} = \cos\theta I + i\sin\theta A$.

The Rotation Operators

$$\begin{aligned} R_x(\theta) &\equiv e^{-i\theta X/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \\ R_y(\theta) &\equiv e^{-i\theta Y/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \\ R_z(\theta) &\equiv e^{-i\theta Z/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \end{aligned}$$

If $\hat{n} = (n_x, n_y, n_z)$ is a real unit vector in three dimensions, then it can be shown that the operator $R_{\hat{n}}(\theta)$ rotates the Bloch vector by an angle θ about the \hat{n} axis, where

$$R_{\hat{n}}(\theta) \equiv \exp(-i\theta \hat{n} \cdot \vec{\sigma}/2)$$

and $\vec{\sigma}$ denotes the three component vector (X, Y, Z) of Pauli matrices. Furthermore, it is not hard to show that $(\hat{n} \cdot \vec{\sigma})^2 = I$, and therefore we can use the special case operator exponential and write

$$\begin{aligned} R_{\hat{n}}(\theta) &= \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)\hat{n} \cdot \vec{\sigma} \\ &= \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)(n_x X + n_y Y + n_z Z) \end{aligned}$$

GLENDINNING

Opérateur Hadamard H :

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

$$H|0\rangle = |+\rangle$$

$$H|1\rangle = |-\rangle$$

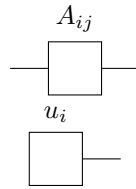
$$H|+\rangle = |0\rangle$$

$$H|-\rangle = |1\rangle$$

$$H = \frac{1}{\sqrt{2}}(X + Z) = \exp(-i\pi(\frac{X + Z}{2\sqrt{2}}))$$

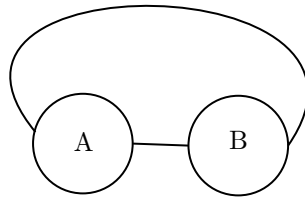
$$(HSH)^2 = X$$

Notation tensorielle : On peut voir que deux pattes représentent deux indices différents et lorsqu'une ligne revient sur elle même, cela signifie qu'on somme les éléments diagonaux de telle sorte qu'il s'agit exactement de la trace d'une matrice $\text{Tr}\{AB\} = \text{Tr}\{BA\}$ puisque l'on peut glisser le B en avant du A le long de l'identité, donc on peut ainsi avec la notation tensorielle en extirper des propriétés intéressantes juste à l'oeil.



Ici, nous avons $\sum A_{ij} B_{jk}$

$$\text{Tr}(AB) = \text{Tr}(BA) = \sum_{j=1}^n \sum_{i=1}^n (a_{ij} b_{ji})$$



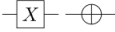








Voici quelques tables de portes logiques :

XOR :		
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Décomposition spectrale :

Si $A^\dagger A = AA^\dagger$, et que A possède une représentation diagonale dans une base donnée, alors A peut s'écrire sous la forme : $A = \sum a_i |u_i\rangle\langle u_i|$. C'est pourquoi puisque Z est diagonale dans la base standard, que $Z = |0\rangle\langle 0| + |1\rangle\langle 1|$

Voici un tableau résumé des portes logiques : **XANADU**

Gate	Matrix	Circuit element(s)	Basis state action
X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$		$X 0\rangle = 1\rangle$ $X 1\rangle = 0\rangle$
H	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$		$H 0\rangle = \frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$ $H 1\rangle = \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$
Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$		$Z 0\rangle = 0\rangle$ $Z 1\rangle = - 1\rangle$
S	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$		$S 0\rangle = 0\rangle$ $S 1\rangle = i 1\rangle$
T	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$		$T 0\rangle = 0\rangle$ $T 1\rangle = e^{i\pi/4} 1\rangle$
Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$		$Y 0\rangle = i 1\rangle$ $Y 1\rangle = -i 0\rangle$
RZ	$\begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$		$RZ(\theta) 0\rangle = e^{-i\frac{\theta}{2}} 0\rangle$ $RZ(\theta) 1\rangle = e^{i\frac{\theta}{2}} 1\rangle$
RX	$\begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$		$RX(\theta) 0\rangle = \cos\frac{\theta}{2} 0\rangle - i\sin\frac{\theta}{2} 1\rangle$ $RX(\theta) 1\rangle = -i\sin\frac{\theta}{2} 0\rangle + \cos\frac{\theta}{2} 1\rangle$
RY	$\begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$		$RY(\theta) 0\rangle = \cos\frac{\theta}{2} 0\rangle + \sin\frac{\theta}{2} 1\rangle$ $RY(\theta) 1\rangle = -\sin\frac{\theta}{2} 0\rangle + \cos\frac{\theta}{2} 1\rangle$