



Code quantique et tolérance aux fautes

IQ01 - Automne 2021

GROUPE 2 :

Téva DUCOTÉ (GI01)

François GUICHARD (GI01)

Matthieu GUIRAMAND (GI05)

Thomas LEYMONERIE (GI03)

Pierre POULIQUEN (GI05)

RAPPORT INTERMÉDIAIRE

Table des matières

1	Introduction	2
2	Définitions	2
3	Implantation, circuit correcteur	3
3.1	The three-qubit bit-flip code	3
3.1.1	Encodage	3
3.1.2	Détection d'erreur	4
3.1.3	Correction	4
3.1.4	Défaut	4
3.1.5	Plus loin : répétition code	4
3.2	Phase/Sign flip code	5
3.2.1	Présentation	5
3.2.2	Encodage	5
3.2.3	Implémentation	6
3.3	Repetition Code	6
3.4	Shor code	6
4	Tolérance aux fautes chez IBM	7
5	conclusion	9

1 Introduction

Toute branche de l'informatique possède son lot d'erreur. Afin d'avoir les meilleures performances et les résultats les plus précis possibles, il est de l'intérêt de l'ingénieur informaticien de réduire ses erreurs. Il en est de même pour l'informatique quantique. En effet, l'informatique quantique comme l'informatique classique se réalise grâce à des modèles physiques. Ces modèles n'étant pas parfait, dû notamment à des champs magnétiques, l'usure des composants physiques, ils se créaient des erreurs. Bien sûr, il en va de même pour l'informatique et nous allons voir les moyens pour atténuer l'impact de ces erreurs.

2 Définitions

Dans cette section, nous allons définir ce qu'est un code quantique puis ce qu'est la tolérance aux fautes.

Code quantique : La correction d'erreur quantique (QEC) est utilisée en informatique quantique pour protéger les informations quantiques des erreurs dues à la décohérence et à d'autres bruits quantiques. La correction d'erreur quantique est essentielle si l'on veut obtenir un calcul quantique tolérant aux pannes qui peut traiter non seulement le bruit sur les informations quantiques stockées, mais également les portes quantiques défectueuses, la préparation quantique défectueuse et les mesures défectueuses. Le code quantique s'applique donc à réduire les erreurs quantique dû à la décohérence.[2]

Tolérance aux fautes : La tolérance aux fautes est la propriété qui permet à un système de continuer à fonctionner correctement en cas de défaillance d'un ou plusieurs défauts au sein de certains de ses composants. Si sa qualité de fonctionnement diminue du tout, la diminution est proportionnelle à la gravité de la défaillance, par rapport à un système conçu naïvement, dans lequel même une petite défaillance peut provoquer une panne totale. La tolérance aux pannes est particulièrement recherchée dans les systèmes à haute disponibilité ou critiques. La capacité de maintenir la fonctionnalité lorsque des parties d'un système tombent en panne est appelée dégradation progressive. [5]

La décohérence quantique : Après la manipulation des phénomènes quantiques, il est nécessaire de les transposer aux règles de la physique classique. Cependant cette transition n'est pour le moment expliquer que par une théorie celle de la décohérence quantique. L'un des problème majeur de la physique quantique est le phénomène de superposition, qui est complètement inconnue au niveau macroscopique. Mais aussi la théorie quantique tient compte du phénomène de non-observabilité, puisque toutes observations provoque un effondrement de la fonction d'onde et perturbe l'état. La décohérence quantique donne lieu alors au postulat n°5 : la réduction du paquet d'onde. Cependant, elle rentre en contradiction avec le postulat n°6 : l'équation de Schrödinger. Étant la théorie la plus satisfaisante à ce jour, la communauté scientifique tente à prouver que la Réduction des paquets d'ondes n'est qu'une conséquence de l'équation de Schrödinger, et non une contradicton. [1]

3 Implantation, circuit correcteur

3.1 The three-qubit bit-flip code

Bit flip Error : Une erreur classique est l'inversion d'un bit dans le code, soit un bit passe de 0 à 1 ou vice versa. [6]

Cette méthode utilise le principe d'intrication afin de détecter des potentielles erreurs.

3.1.1 Encodage

Considérons que la probabilité qu'une erreur de type bit flip apparaisse soit de p , alors l'arbre binaire quantique symétrique suivant :

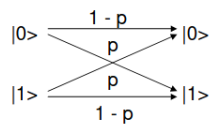


FIGURE 1 – cou

Afin de palier à ce problème, l'utilisation de 2 autres qubits permet de détecter ce type d'erreur. Nous définissons tout d'abord l'encodage suivant :

Soit $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$

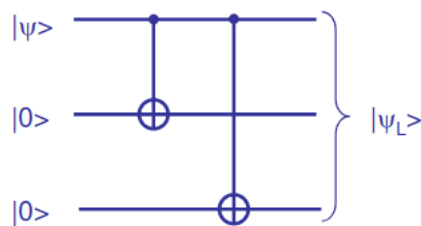


FIGURE 2 – Caption

$$\begin{aligned} |0\rangle &\rightarrow |0_L\rangle = |000\rangle \\ |1\rangle &\rightarrow |1_L\rangle = |111\rangle \end{aligned}$$

0 erreur	1 erreur	2 erreurs	3 erreurs
p^3	$p^2(1-p)$	$p(1-p)^2$	$(1-p)^3$

Nous pouvons donc en déduire : $|\psi\rangle = c_0|0_L\rangle + c_1|1_L\rangle = c_0|000\rangle + c_1|111\rangle$

3.1.2 Détection d'erreur

Il existe alors 4 types d'erreurs pouvant intervenir sur chacun des trois Qubit :

$P_1 = |000\rangle\langle 000| + |111\rangle\langle 111|$ erreur sur le Qubit n°1

$P_2 = |100\rangle\langle 100| + |011\rangle\langle 011|$ erreur sur le Qubit n°1

$P_3 = |010\rangle\langle 010| + |101\rangle\langle 101|$ erreur sur le Qubit n°2

$P_4 = |001\rangle\langle 001| + |110\rangle\langle 110|$ erreur sur le Qubit n°3

Lorsqu'il n'y a pas d'erreur de bit flip, $\langle \psi | P_1 | \psi \rangle = 1$ et $\langle \psi | P_{2,3,4} | \psi \rangle = 0$, dans le cas contraire, $\langle \psi | P_{2,3,4} | \psi \rangle = 1$. On peut en déduire que la mesure de $P_1 P_2 P_3$ et P_4 nous permet de déduire si une erreur de bit flip s'est produite ou non. Les opérateurs de projection ne change pas l'état du Qubit logique. Ils ne font simplement que décrire quel Qubit a été "flipped". La mesure de ses opérateurs ne donne aucune indication sur les valeurs de α ou de β . Mais seulement quel Qubit doit être corrigé.

Supposons que le Qubit n°1 a subi un flip erreur dans ce cas l'état $|\psi\rangle$ est : $|\psi\rangle = c_0 |100\rangle + c_1 |011\rangle$ Alors $\langle \psi | P_1 | \psi \rangle = 1$, ce qui vérifie qu'il y a bien eu un bit flip sur le Qubit n°1.

3.1.3 Correction

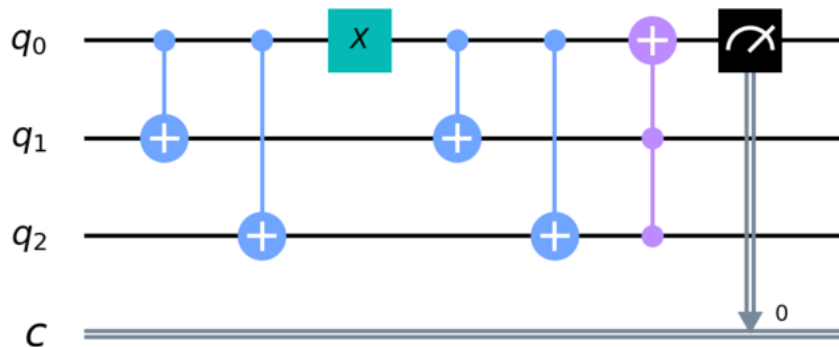


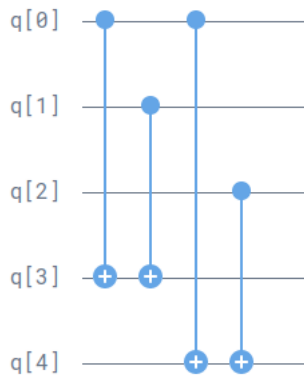
FIGURE 3 – Circuit bit flip code

Suivant l'erreur du bit flip, il faut placer un opérateur NOT sur le Qubit cible

3.1.4 Défaut

3.1.5 Plus loin : répétition code

La détection d'erreur est faite par l'emploi de deux autres Qubits auxiliaires. Deux portes CNOT sont couplés avec le premier Qubit auxiliaire, permettant de détecter une éventuel erreur dans le second Qubit. Le même méthode est utilisé pour détecter si une erreur est survenu dans le troisième Qubit.



3.2 Phase/Sign flip code

3.2.1 Présentation

Dans un ordinateur classique, les bits inversés sont les seules types d'erreurs. Cependant, en informatique quantique, il peut se produire une autre erreur : l'erreur d'inversion de phase.[7]

Ce type d'erreur affecte la phase du Qbit, c'est donc l'application d'une porte Z.

On obtient alors : $Z|0\rangle = |0\rangle$ et $Z|1\rangle = -|1\rangle$.

On se rend alors compte que $Z|+\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle$, $Z|-\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle$.

Et que, $|+\rangle = H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ et $|-\rangle = H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$. Alors, l'inversion de phase peut se ramener à une inversion de bit avec les équations suivantes :

$$ZH|0\rangle = Z|+\rangle = |-\rangle$$

$$ZH|1\rangle = Z|-\rangle = |+\rangle$$

Donc, en rajoutant des portes Hadamard à l'entrée et en sortie de l'Oracle de l'erreur, nous pouvons traiter l'erreur de la même manière que le Bit Flip Code avec deux autres Qbits auxiliaires et deux portes CNOT couplés avec le premier Qubit et les Qubits auxiliaires.

3.2.2 Encodage

On encode ce l'état du Qubit afin d'obtenir : $|\phi\rangle = a|0\rangle + b|1\rangle \rightarrow a|+++ \rangle + b|--- \rangle = a|0_L\rangle + b|1_L\rangle$ [8]

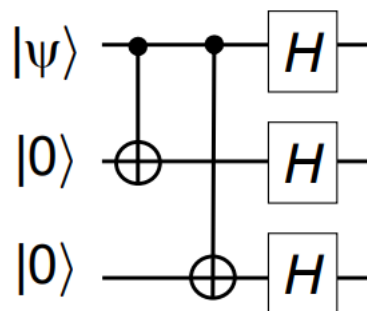


FIGURE 4 – Circuit de l'encodage du Qubit

3.2.3 Implémentation

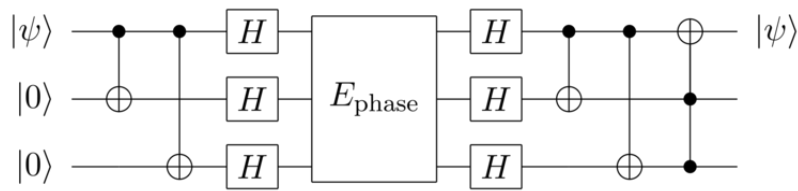
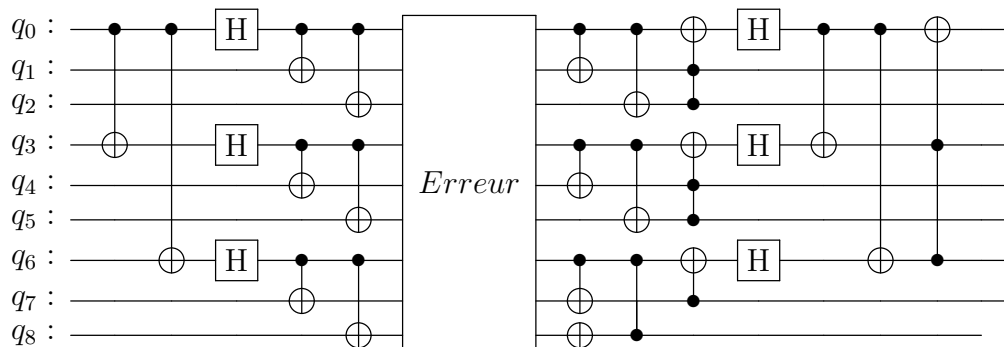


FIGURE 5 – Diagram Phase Flip Code

3.3 Repetition Code

3.4 Shor code

Nous avons vu précédemment des codes permettant de corriger des erreurs de bit flip ou phase flip, nous allons maintenant voir un code qui peut corriger l'occurrence des deux erreurs (ensemble ou non). Ce code, nommé Shor code, permet de corriger un qubit en utilisant 8 qubits de travail. En voici son diagramme :



Une démonstration mathématique du fonctionnement du code sera présentée. Voir le Jupyter Notebook pour une application détaillée du shor code.

4 Tolérance aux fautes chez IBM

IBM pour International Business Machines Corporation est une entreprise multinationale américaine présente dans de nombreux domaines de l'informatique dont l'informatique quantique. En effet en 2014, la société a annoncé investir 3 milliards de dollars pour repousser les limites technologiques notamment grâce à la quantique. Au fil des quatre dernières années IBM a conçu plus d'une trentaine d'ordinateur quantique. Ces ordinateurs utilisent des atomes froids piégés et refroidis par laser (à une température frisant les -273°C , le zéro absolu) [4].

Name	Qubits	QV	CLOPS	Status	Total pending jobs	Processor type	Features
ibmq_alexington	127	-	-	Offline	0	Exige 15	-
ibmq_bronxlyn	65	32	1.5K	Online	750	Hummingbird 12	-
ibmq_losgatos	27	128	2K	Online - Queue paused	1	Palcoor 15.11	-
ibmq_montreal	27	128	2K	Online	10	Palcoor 16	-
ibmq_mumbai	27	128	1.5K	Online	15	Palcoor 15.1	-
ibmq_peeko	27	64	2.4K	Online	239	Palcoor 15.11	-
ibmq_peeko	27	64	2.3K	Online	874	Palcoor 15.11	-
ibmq_peeko	27	32	1.5K	Online	12	Palcoor 16	-
ibmq_peeko	27	32	1.5K	Online - Queue paused	12	Palcoor 16	-
ibmq_peeko	27	-	-	Offline	0	Palcoor 15	-
ibmq_peeko	16	32	2.4K	Online	85	Palcoor 14P	-
ibmq_peeko	7	32	2.3K	Online	28	Palcoor 15.11H	-
ibmq_peeko	7	32	2.7K	Online	10	Palcoor 15.11H	-
ibmq_peeko	7	32	2.4K	Online	0	Palcoor 15.11H	-
ibmq_peeko	7	32	2.3K	Online	41	Palcoor 14H	-
ibmq_peeko	7	16	2.4K	Online	79	Palcoor 15.11H	-
ibmq_peeko	5	32	2.3K	Online	50	Palcoor 15.11	-
ibmq_peeko	5	32	2.3K	Online	29	Palcoor 16L	-
ibmq_peeko	5	32	-	Online - Queue paused	1348	Palcoor 16L	-
ibmq_peeko	5	16	2.3K	Online	25	Palcoor 14T	-
ibmq_peeko	5	16	2.3K	Online	21	Palcoor 14T	-
ibmq_peeko	5	8	2.7K	Online	20	Palcoor 14T	-
ibmq_peeko	1	1	-	Online - Queue paused	79	Canary 11.3	-

FIGURE 6 – Liste des ordinateurs quantique d'IBM non exhaustive [3]

IBM se démarque notamment grâce à sa plateforme en ligne de simulation d'algorithme quantique IBM Quantum Service avec notamment : IBM Quantum Composer ou encore son "laboratoire" en ligne IBM Quantum Lab

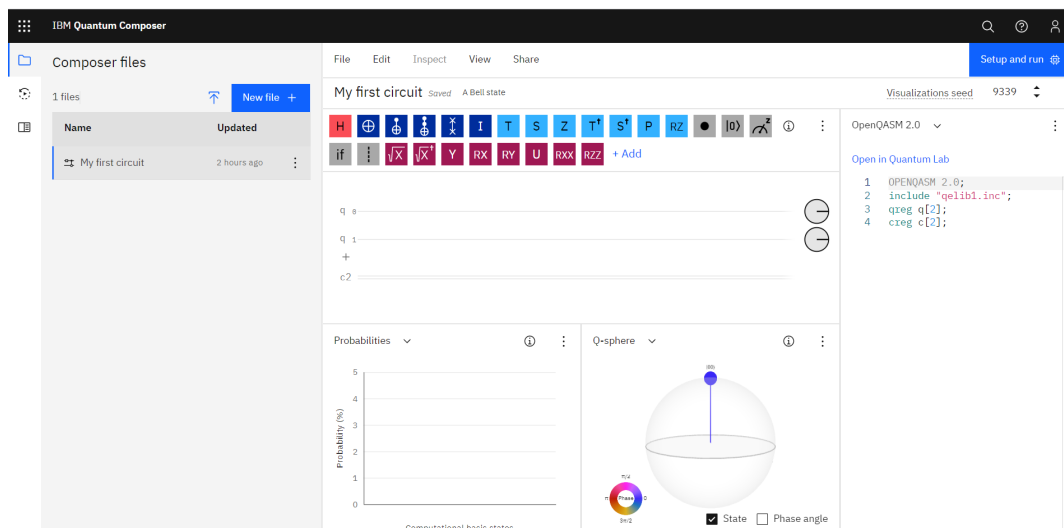


FIGURE 7 – IBM Quantum Composer

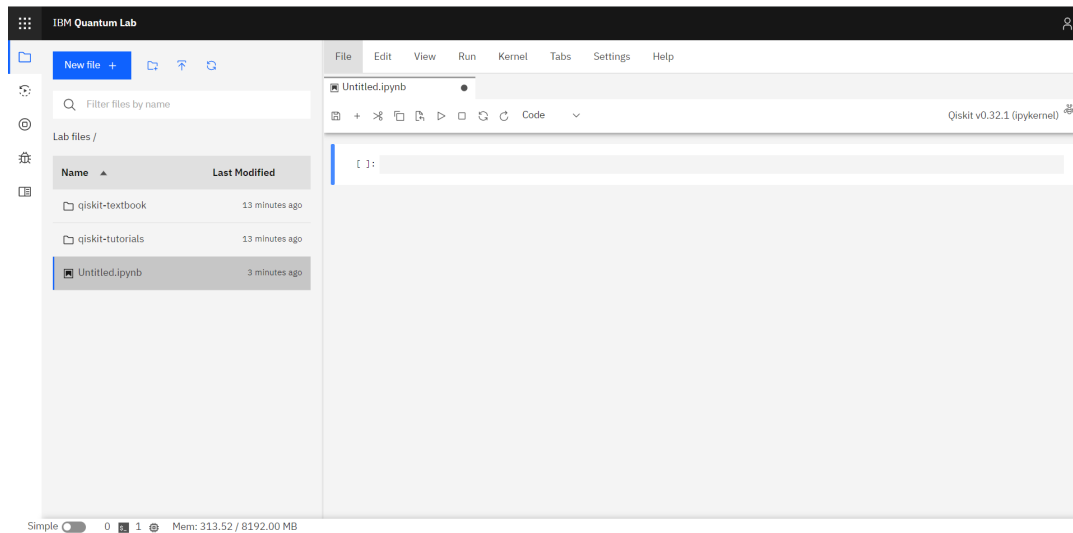


FIGURE 8 – IBM Quantum Lab

Cette plateforme permet à n'importe qui de réaliser n'importe quel circuit quantique très facilement soit graphiquement avec IBM Quantum Composer, qui génère d'ailleurs un code python avec qiskit, ou en codant avec le laboratoire compilateur en ligne IBM Quantum Lab, sous python avec encore une fois la librairie Qiskit.

L'intérêt est de rendre accessible à tous l'informatique quantique et ainsi faire passer IBM comme référence en terme d'informatique quantique mais aussi de permettre à tous de pouvoir utiliser un ordinateur quantique car en effet chaque code produit via leurs outils ou la librairie qiskit peut être compilé sous un réel ordinateur quantique. Ainsi une des problématiques qu'il en résulte est de créer l'ordinateur quantique le plus fiable possible. En effet, comme dit précédemment, un ordinateur quantique essaye de s'approcher au maximum du zéro absolu pour fonctionner et cela représente un réel défi physique. À ce jour, IBM a atteint les 127 qubits mais ce n'est pas sans sacrifice car le processeur Eagle R1 (à Washington) possède un taux d'erreur sur les CNOTs plus important que d'autres processeurs.

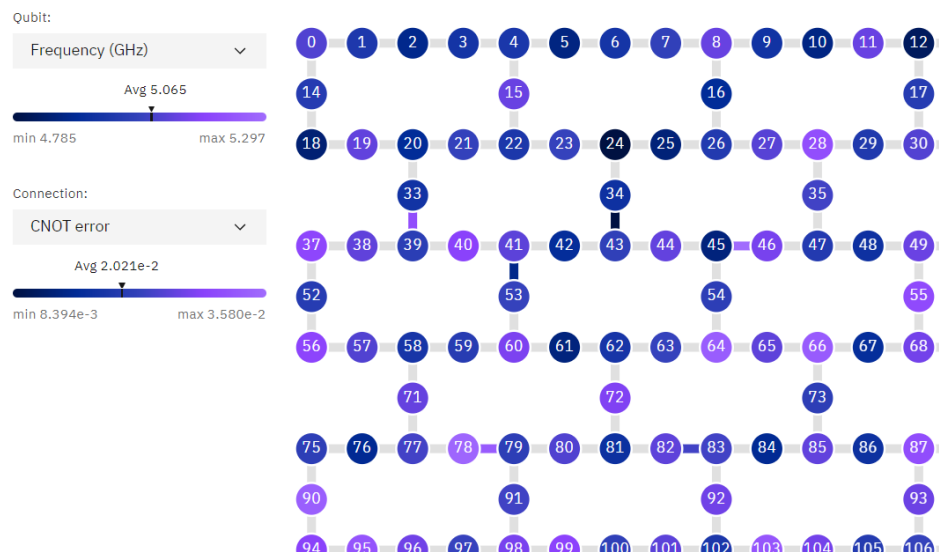


FIGURE 9 – Taux d'erreur CNOT gate processeur Eagle R1

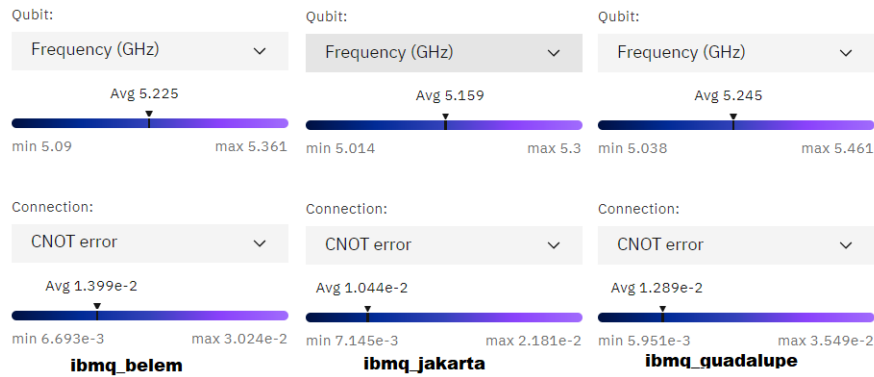


FIGURE 10 – Taux d'erreur CNOT gate sur d'autres processeurs

Sur les différents processeurs pris au hasard ¹, on tombe sur les moyennes suivantes :

1. Moyenne des minimums : $6.596e-3 < 8.394e-3$ soit 27% d'écart relatif,
2. Moyenne des maximums : $2.918e-2 < 3.580e-2$ soit 22% d'écart relatif,
3. Moyenne des moyennes : $1.244e-2 < 2.021e-2$ soit 62% d'écart relatif.

Ainsi, le processeur avec 127 qubits possède globalement plus de probabilité d'erreur que sur les portes CNOT. Finalement on peut se poser la question : comment fait IBM pour répondre à cette problématique. Pour cela on se basera sur l'article suivant : **Fault-Tolerant Logical Gates in the IBM Quantum Experience** <https://arxiv.org/abs/1806.02359> [à compléter]

5 conclusion

1. À noter qu'un échantillon de 3 éléments n'est pas représentatif d'une moyenne réelle puisque qu'IBM propose plus de 25 processeurs. L'idée est surtout de prendre quelques processeur avec 5-7 qubits et de comparer avec le nouveau processeur.

Références

- [1] Les codes correcteurs quantiques. *Collège de France*, 2004.
- [2] Schaeetz Chiaverini, Leibfried. Realization of quantum error correction. *Nature* 432, 2004.
- [3] IBM. Ibmquantum.
- [4] Leïla Marchand. Ordinateur quantique : cinq questions pour (enfin) tout comprendre. *LesEchos*, 2021.
- [5] Oscar González / H Shrikumar / John A. Stankovic / Krithi Ramamritham. Adaptive fault tolerance and graceful degradation under dynamic hard real-timescheduling. *University of Massachusetts Amherst*, 1997.
- [6] Quantum Computing UK. Quantum error correction : Bit flip code in qiskit. *Quantum Computing UK*, 2020.
- [7] Quantum Computing UK. Quantum error correction : Phase flip code in qiskit. *Quantum Computing UK*, 2020.
- [8] Fumiko Yamaguchi. Quantum error correction. *Stanford University*, 2005.