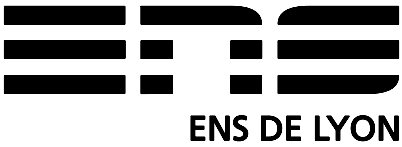


Codes correcteurs quantiques

Départements des sciences de la matière et d'informatique
Physique, information et calcul

Frédéric Combes, Alexis Morvan

8 mai 2012

Table des matières

Introduction

Les implémentations actuelles d'algorithmes quantiques sont sujettes à des erreurs, de l'ordre de 1 à 3%, bien plus nombreuses que pour le calcul classique, ce qui rend la création d'un ordinateur utilisant massivement le calcul quantique impossible si aucun moyen n'est mis en œuvre pour corriger les erreurs qui surviennent : c'est là l'utilité des codes correcteurs quantiques.

Un code fonctionne selon le schéma présenté par la figure 1.

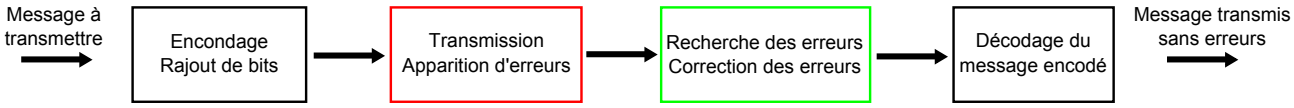


Figure 1 – Schéma de fonctionnement d'un code

Des codes correcteurs fonctionnant selon le même schéma sont utilisés en informatique classique pour pallier aussi aux erreurs qui surviennent. On pourrait être tenté de faire un parallèle entre ces deux codes, cependant, il se heurte aux problèmes de la mécanique quantique : le non-clonage, les problèmes liés à la mesure, et surtout les erreurs.

Un exemple permet de souligner la façon dont interviennent ces problèmes : on souhaite transmettre le message 10011 via un canal de transmission. Une méthode issue de l'informatique classique consiste à encoder le message en dupliquant les bits : on encode 10011 par 111 000 000 111 111. Ce message est alors transmis via le canal, et des erreurs surviennent. La personne qui réceptionne le message lit (mesure) 110 000 010 111 011. Il est clair que le message envoyé le plus vraisemblable est 111 000 000 111 111, c'est celui qui satisfait au minimum d'erreurs. Il ne reste alors plus qu'à décoder.

Lors de l'application cet algorithme aux qubits quantiques, trois points posent problème :
- la copie des qubits n'est pas autorisé en mécanique quantique à cause du théorème de non clonage,
- la mesure les bquits pour en obtenir une connaissance exacte n'est pas non plus possible pour la même raison,
- un seul type d'erreur est corrigé : le bit-flip, qui est le seul type d'erreur possible en informatique classique, mais qui est loin d'être le seul type en mécanique quantique : les déphasages et rotations sont aussi des erreurs possibles.

Nous verrons dans la suite qu'il est possible de passer outre ces restrictions, d'abord au travers d'un exemple simple qui permettra de contourner le théorème de non-clonage et le problème de la mesure, puis, en modélisant les erreurs dans un canal quantique, qu'il suffit de se contenter de corriger seulement deux types d'erreurs. Ensuite nous montrerons comment construire les codes correcteurs quantiques avec les outils hérités des codes correcteurs classiques.

1 Un exemple : le code du bit-flip à trois qubits

Une erreur similaire à l'erreur classique, le *bit-flip*, serait une erreur qui enverrait $|0\rangle$ sur $|1\rangle$ et $|1\rangle$ sur $|0\rangle$ avec une probabilité de p comme le montre le graphe 2.

Par linéarité, cette erreur n'est autre que l'application de l'opérateur $\hat{X}$ de Pauli avec une probabilité de p.
Le *théorème de non clonage* ne permet pas de réutiliser le codage par *redondance* (présenté dans l'introduction) qui permet de corriger ce type d'erreur en informatique classique. Cela reviendrait à encoder $|\psi\rangle^{\otimes 3}$ par exemple. Mais ceci est interdit par les lois de la mécanique quantique. Il faut donc trouver un moyen de contourner ce problème. La porte *controlled-not* (ou CNOT) décrite ci-dessous permet

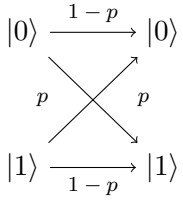


Figure 2 – Canal symétrique produisant un bit-flip avec une probabilité p

justement de contourner ce problème (figure 3).

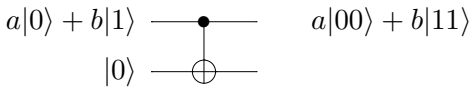


Figure 3 – Porte C-NOT

La porte CNOT, permet d'encoder un état à un qubit $a|0\rangle + b|1\rangle$ par un l'état à deux qubits $a|00\rangle + b|11\rangle$. Et ce sans pour autant violer le théorème de non-clonage car on ne duplique pas l'état $|\psi\rangle$. L'utilisation de deux portes CNOT avec deux autres qubits permet de réaliser un encodage effectif avec la *porte de Toffoli*, présentée par la figure 4.

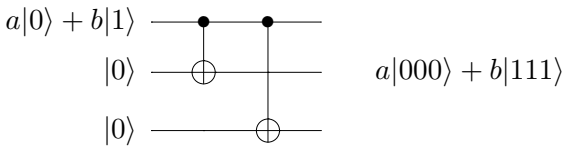


Figure 4 – Porte de Toffoli ou CCNOT

Ainsi, la porte de Toffoli permet de coder :

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

par le trois-qubit :

$$|\psi\rangle = a|000\rangle + b|111\rangle$$

Si maintenant chacun des trois qubits passe dans le canal bit-flip (celui-ci agit indépendamment sur chaque qubits), chaque qubit a alors une probabilité p d'effectuer un bit-flip. Faisons l'hypothèse supplémentaire que le bit-flip ne se produit que sur un qubit au plus. La correction de cette erreur se déroule alors en deux étapes : *le diagnostic du syndrome* et *la correction*.

1.1 La détection de l'erreur : diagnostic du syndrome

Il faut réaliser une mesure qui dira quelle erreur a eu lieu tout en ne n'affectant pas la superposition d'état (car c'est cette superposition qui contient l'information). Pour cela, considérons les quatre projecteurs suivants :

$$\begin{aligned}
\hat{P}_0 &= |000\rangle\langle 000| + |111\rangle\langle 111| \\
\hat{P}_1 &= |100\rangle\langle 100| + |011\rangle\langle 011| \\
\hat{P}_2 &= |010\rangle\langle 010| + |101\rangle\langle 101| \\
\hat{P}_3 &= |001\rangle\langle 001| + |110\rangle\langle 110|
\end{aligned}$$

Supposons pas exemple qu'à la sortie du canal, le 3-qubits soit dans l'état $|\phi\rangle = a|100\rangle + b|011\rangle$. La mesure du syndrome est l'application des projecteurs :

$$\langle\phi|P_i|\phi\rangle = \delta_{i1}$$

Ainsi, lors de la mesure des syndromes, on trouve exactement 1 pour $i = 1$ et 0 pour tout les autres, ce qui nous indique que l'erreur se situe sur le premier qubit. Pour cette mesure des syndromes, c'est la mesure qui donne un résultat non nul qui indique où se situe l'erreur.
Le plus important est que cette mesure de syndromes n'affecte pas la superposition d'état : les syndromes $\hat{P}_i$ projettent l'état $|\phi\rangle$ sur un sous-espace auquel il appartient. Donc la mesure des syndromes ne perturbe pas l'état.

D'autres types de syndromes peuvent être utilisés comme les mesures de $\hat{Z}_1\hat{Z}_2$ et de $\hat{Z}_1\hat{Z}_3$, où $\hat{Z}_i$ est l'opérateur de Pauli selon l'axe x sur le i-ème qubit. Mesurer $\hat{Z}_1\hat{Z}_2$ revient à tester l'égalité entre les qubits 1 et 2.

1.2 Correction

Le diagnostic du syndrome donne sur quel qubit l'erreur à eu lieu. Il suffit alors d'appliquer la procédure suivante :
- syndrome 0 : ne rien faire
- syndrome i avec $i \in \{1, 2, 3\}$: basculer le i-ème qubit avec $\hat{X}_i$

Cette méthode ne fonctionne que si le bit-flip affecte au plus un qubit.

1.3 Un autre type d'erreur : le phase-flip

Le bit-flip n'est pas le seul type d'erreur possible. D'autres erreurs peuvent apparaître, comme un changement de phase relative : par exemple, un cana de transmission peut changer la phase relative entre $|0\rangle$ et $|1\rangle$ de π avec une probabilité p :

$$a|0\rangle + b|1\rangle \longmapsto a|0\rangle - b|1\rangle$$

Afin de traiter cette erreur, il est intéressant de changer de base et de se placer dans la base de Hadamard $(|+\rangle, |-\rangle)$: $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ et $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$.

L'action de l'erreur peut ainsi se visualiser à l'aide du graphe de la figure 5.

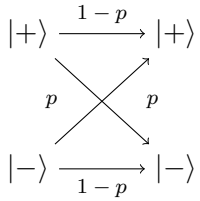


Figure 5 – Canal produisant un *phase-flip* avec probabilité p

Il s'agit en fait du cas précédent, mais dans une autre base. Toute la procédure développée précédemment peut s'appliquer dans cette nouvelle base. En fait, il est possible de corriger n'importe quelle erreur de type changement de phase relative :

$$a|0\rangle + b|1\rangle \longmapsto a|0\rangle + e^{i\theta}b|1\rangle$$

grâce à un changement de base (qui revient à réaliser une rotation autour de la *sphère de Bloch*).

2 Formalisme de la théorie des codes correcteurs quantiques

Dans cette partie, nous essayerons d'introduire de manière physique les outils adaptés au traitement des codes correcteurs quantiques. Nous rappellerons la notion d'opérateur densité puis nous parlerons des opérations quantiques.

2.1 Opérateur de densité

Pour un *état pur* $|\psi\rangle$, l'opérateur densité est décrit par l'opérateur :

$$\rho = |\psi\rangle\langle\psi|$$

ainsi, pour les qubits, si $|\psi\rangle$ s'écrit $a|0\rangle + b|1\rangle$, l'opérateur densité associé s'écrit :

$$\rho_{|\psi\rangle} = \begin{pmatrix} |a|^2 & ba^* \\ ab^* & |b|^2 \end{pmatrix}$$

Mais là où l'opérateur densité est vraiment intéressant, c'est lorsqu'il s'agit de décrire un ensemble statistique d'états quantiques. Soit $\{p_i, |\psi_i\rangle\}$ un ensemble d'états quantiques avec leur probabilité associée pour un système donné. L'opérateur densité est alors défini par :

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

L'opérateur de densité permet alors de décrire de façon précise un mélange statistique d'états quantique.

2.2 Les opérations quantiques

Afin de pouvoir étudier les erreurs , il est nécessaire de décrire quelles sont les opérations réalisable sur une distribution d'états quantique, i.e. sur un opérateur de densité. Cette étude se justifie par le fait que les erreurs ainsi que les corrections ne sont que des opérations sur l'opérateur densité d'état.

Tout d'abord, étudions le cas d'une transformation sur un état quantique, présentée par la figure 6.

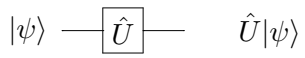


Figure 6 – Opération quantique sur les vecteurs d'état : il s'agit d'une simple opération unitaire

Que se passe-t-il pour l'opérateur de densité ? Pour commencer par un cas simple, considérons le cas d'un état pur : $\rho = |\psi\rangle\langle\psi|$. D'après ce qui précède, après l'opération quantique, l'état s'écrit $U|\psi\rangle$. Donc l'opérateur densité après évolution s'écrit :

$$\hat{\Phi}(\rho) = \hat{U}|\psi\rangle\langle\psi|\hat{U}^\dagger$$

Comme les opérateurs de densité sont des barycentres d'opérateurs de densité d'état pur, cette propriété ce généralise immédiatement à tout état mixte :

$$\hat{\Phi}(\rho) = \hat{U}\rho\hat{U}^\dagger$$

Le schéma de cette transformation est présenté par la figure 7.

En théorie, seules les opérations unitaires sont autorisées par la mécanique quantique et cette description devrait suffire. Cependant, l'environnement joue très souvent un rôle de décohérence et influe sur le système (c'est notamment le cas des erreurs qui sont en fait un effet de l'environnement). Ce n'est plus le système étudié qui subit une transformation unitaire comme décrit ci-dessus, mais le système

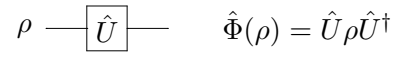


Figure 7 – Opération quantique unitaire sur les opérateurs densités

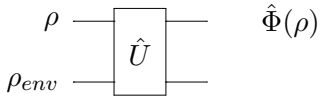


Figure 8 – Opération quantique incluant une interaction avec l'environnement

global incluant l'environnement, comme le montre la figure 8.

Afin de décrire tout de même ce genre d'opération, nous allons introduire la *réprésentation des opérateurs quantique comme somme d'opérateur* (en anglais, *operator-sum representation*). Une opération quantique va opérer sur la matrice de densité et la transformer en une autre matrice de densité. Il est à noter que l'espace des états avant et après l'opération peut être différent.

Si nous prenons l'exemple du bit-flip, nous voyons que l'erreur envoie ρ sur ρ avec une probabilité $1-p$ et envoie ρ sur $\hat{X}\rho\hat{X}$ avec une probabilité p (figure 9).

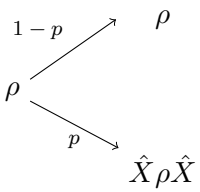


Figure 9 – Action du bit-flip sur l'opérateur de densité

Ainsi, l'opérateur densité résultant peut être pris comme barycentre de ces deux résultats affectés de leur probabilités respectives :

$$\hat{\Phi}(\rho) = (1-p)\rho + p\hat{X}\rho\hat{X}$$

En généralisant ce résultat, il est possible de décrire une opération quantique par :

$$\hat{\Phi}(\rho) = \sum_k p(k)\rho_k = \sum_k \hat{E}_k \rho \hat{E}_k^\dagger$$

où les ρ_k représente les opérateurs densité possibles à la sortie, $p(k)$ les probabilités associées à ces opérateurs densité. Dans la deuxième écriture, les $\hat{E}_k$ sont les *opérations élémentaires* de $\hat{\Phi}$. Ces opérateurs ne sont pas nécessairement unitaires ; dans l'exemple traité ci-dessus, les opérateurs élémentaires sont $\hat{E}_0 = \sqrt{1-p}Id$ et $\hat{E}_1 = \sqrt{p}\hat{X}$.

Il existe une approche rigoureuse, à la fois mathématique et physique, de cette décomposition en opérateur somme, qui peut être trouvée dans [1]. Nous retiendrons seulement le résultat suivant :

Théorème 2.1 *Une erreur $\hat{\Phi}$ peut être décrite par un opérateur somme :*

$$\hat{\Phi}(\rho) = \sum_k \hat{E}_k \rho \hat{E}_k^\dagger$$

où $\{\hat{E}_k\}$ est un ensemble d'opérateurs allant de l'espace des états d'entrée dans l'espace des états de sortie.

Grâce à ce formalisme, nous sommes en mesure de décrire n'importe quelle erreur.

2.3 Les conditions de correction d'erreur quantique

Les états quantiques sont encodés dans un *code correcteur quantique* formellement défini par un sous espace $\mathcal{C}$ d'un espace de Hilbert plus grand. Le projeteur sur le code $\mathcal{C}$ sera noté $\hat{P}$. Dans l'exemple du bit-flip, $\mathcal{C}$ est le sous espace de l'espace des états de 3-qubits engendré par $|000\rangle$ et $|111\rangle$. Le projecteur s'écrit alors $\hat{P} = |000\rangle\langle 000| + |111\rangle\langle 111|$. L'erreur et la correction sont deux opérations quantiques que nous noterons $\hat{\Phi}$ et $\hat{R}$. Pour que la correction soit effective, il faut que :

$$\left(\hat{R} \circ \hat{\Phi}\right)(\rho) = \rho$$

Le théorème suivant donne les *conditions de corrigibilité d'une erreur quantique* :

Théorème 2.2 *Soit $\mathcal{C}$ un code quantique et $\hat{P}$ le projecteur sur $\mathcal{C}$. Supposons que $\hat{\Phi}$ soit une opération quantique dont les opérations élémentaires sont les $\{\hat{E}_i\}$. Une condition nécessaire et suffisante pour l'existence d'une opération de correction $\hat{R}$ de $\hat{\Phi}$ sur $\mathcal{C}$ est que :*

$$\hat{P}\hat{E}_i^\dagger \hat{E}_j \hat{P} = \alpha_{ij}\hat{P} \tag{1}$$

où α est une matrice hermitienne.

Pour une démonstration, voir [1].

Ainsi, pour voir si une erreur est corrigible, il suffit de décomposer cette erreur en opérations élémentaires et de vérifier la relation (1).

2.4 Discrétisation des erreurs

Le théorème précédent permet de voir si une erreur en particulier est corrigible. Mais la connaissance de l'erreur ne va pas toujours de soi. Cependant, en informatique quantique le théorème suivant facilite les choses :

Théorème 2.3 *Soit $\mathcal{C}$ un code quantique et $\hat{R}$ une correction des erreurs $\{\hat{E}_i\}$ de $\hat{\Phi}$, soit $\hat{F}$ un opérateur quantique avec pour opérations élémentaires les $\{\hat{F}_j\}$ tel que chaque $\hat{F}_j$ soit combinaison linéaire des $\{\hat{E}_i\}$.*
Alors le code correcteur $\hat{R}$ corrige aussi les effets de $\hat{F}$ sur le code $\mathcal{C}$.

Une démonstration peut être trouvée dans [1]

Ce théorème donne un outil formidable pour la correction des erreurs : il suffit de corriger un nombre fini d'erreurs bien choisi pour corriger toutes les combinaisons linéaires de ces « erreurs élémentaires ».

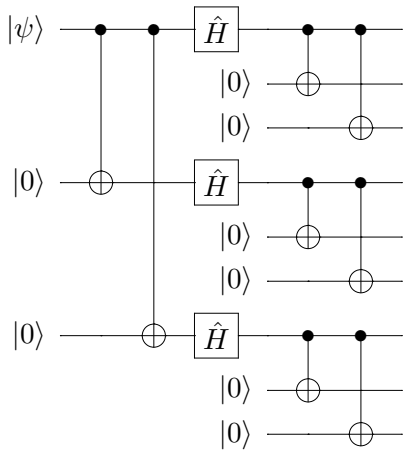
Pour un qubit, l'espaces des opérations sur ce qubit est engendré par les matrices de Pauli et l'identité : Id, $\hat{X}$, $\hat{Y}$ et $\hat{Z} = \hat{X}\hat{Y}$. Pour un n-qubit, l'espaces des opérations sur ce n-qubit est engendré par les produits tensoriels des opérateurs de Pauli et l'identité : $Id \otimes Id$, $\hat{X} \otimes Id$, $Id \otimes \hat{X}$, $\hat{Y} \otimes Id$,... Il suffit alors de corriger les erreurs induites par cette base pour pouvoir corriger n'importe quelle erreur.

2.5 Code de Shor

Ce code n'est rien d'autre qu'une combinaison astucieuse des codes du bit-flip et du phase-flip vu dans la première section : le qubit est encodé par un état 9-qubit :

$$\begin{aligned} |0\rangle &\mapsto \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{(2)}} \\ |1\rangle &\mapsto \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{(2)}} \end{aligned}$$

ce qui est réalisé par le circuit suivant :



Grâce à la théorie développée auparavant, de simples calculs montrent que ce code permet, avec la correction adéquate, de corriger n'importe quelle erreur, pourvu qu'elle n'affecte qu'un seul qubit : en effet, ce code permet d'éviter les erreurs associées aux opérateurs Id, $\hat{X}$, $\hat{Y}$ et $\hat{Z}$ et donc par combinaisons linéaires, il permet d'éviter toutes les erreurs affectant un qubit.

3 Construction des codes correcteurs quantiques

Dans les parties précédentes, nous avons montré que la réalisation d'un code correcteur quantique est possible, mais cependant, nous n'en avons pas construit. C'est ce que nous allons faire dans cette partie, en se dotant d'abord des outils liés au codage classique puis en les transposant à la mécanique quantique de la manière la plus simple possible. Nous donnerons ensuite un exemple de code correcteur, les codes CSS.

3.1 Notions de codage classique

Le formalisme de la mécanique quantique est l'algèbre linéaire, il semble donc pertinent de se doter de codes linéaires pour pouvoir les appliquer sans problèmes à la mécanique quantique.

Les messages à transmettre sont des suites de k éléments de $\{0,1\}$ qui peuvent être vus comme des vecteurs de $(\mathbb{Z}/2\mathbb{Z})^k = \mathbb{F}_2^k$. Pour pouvoir corriger les erreurs, il faut que le message soit redondant. Il est donc nécessaire qu'un message codé soit plus long : un message codé est une suite de N éléments de $\{0,1\}$, soit un vecteur de $\mathbb{F}_2^N$.

Coder est donc une application linéaire G de $\mathbb{F}_2^k \to \mathbb{F}_2^N$. Cependant en mécanique quantique, les transformations doivent être unitaires, en particulier, ce sont des endomorphismes, ainsi, il est donc plus pertinent de définir l'encodage G par :

$$\left\{ \begin{array}{ccc} \mathbb{F}_2^k \times \mathbb{F}_2^{N-k} & \to & \mathbb{F}_2^N \\ (w_1, ..., w_k, a_1, ..., a_{N-k}) & \mapsto & (w'_1, ..., w'_N) = G(w_1, ..., w_k, 0, ..., 0) \end{array} \right.$$

qui est inversible sur $\mathbb{F}_2^k \times \{0...0\}$, il sera donc facile de le transposer à la mécanique quantique. Un résultat de la théorie des codes classique nous dit que le générateur G peut toujours s'écrire sous la forme suivante, sans perdre de pouvoir correcteur :

$$\left\{ \begin{array}{ccc} \mathbb{F}_2^k \times \mathbb{F}_2^{N-k} & \rightarrow & \mathbb{F}_2^N \\ (w_1, ..., w_k, a_1, ..., a_{N-k}) & \mapsto & (w_1, ..., w_k, w'_0, ..., w'_{N-k}) \end{array} \right.$$

qui peut parfois être pratique pour appliquer les résultats à la mécanique quantique.

Un code C est l'image de $\mathbb{F}_2^N$ par G, c'est un sous espace vectoriel de $\mathbb{F}_2^N$. L'écart d entre les mots va correspondre au nombre d'erreurs qu'il est possible de détecter (qui vaut $d-1$), comme le montre la figure 10. Si $d-1$ erreurs sont détectables, alors il est possible d'en corriger $t = \lfloor \frac{d-1}{2} \rfloor$, comme le montre la figure 11. Une erreur e est dite corrigible si elle affecte moins de t bits, et dans la suite, E désigne l'ensemble des erreurs corrigibles.

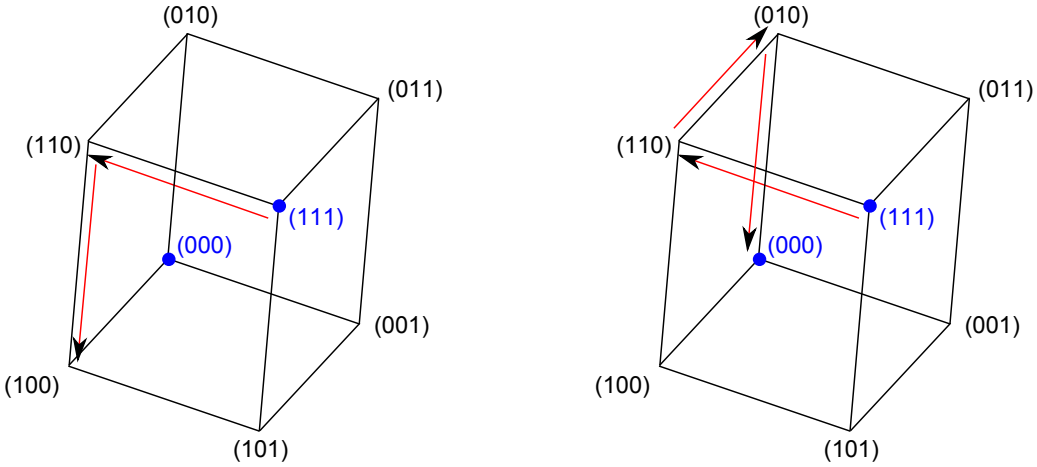


Figure 10 – Illustration de la notion de distance : le cube est une représentation de $\mathbb{F}_2^3$. Un message (un vecteur de un bit, (0) ou (1)) est codé par un vecteur de trois bits. Les vecteurs du code sont en bleu, et une erreur correspond à une flèche rouge. A gauche, deux erreurs surviennent, le vecteur résultant n'est pas un vecteur du code : cette erreur peut être détecter. A droite, trois erreurs surviennent, le vecteur résultant est un vecteur du code : cette erreur ne peut pas être détecter. Ainsi, s'il y a $d-1=2$ erreurs ou moins qui surviennent, elles seront détectées à coup sûr.

Maintenant qu'on s'est donné un code permettant de corriger un certain nombre d'erreurs, il est nécessaire de trouver un moyen de les détecter, c'est ce à quoi sert l'application H, nommée matrice de contrôle.

Si un message w est correct (ou erroné de manière non détectable), alors c'est un vecteur de C, et si le mot est erroné (mais corrigible), alors il sera de la forme $w+e$ ou $w \in C$ et $e \in E$. Toute base $h_1, ..., h_{N-k}$ de $C^* \equiv C^\perp$ va vérifier les propriétés suivantes :

$$w \in C \Leftrightarrow \forall i, h_i(w) = 0$$

et

$$\forall e \neq e' \in E, \forall w \in C, \exists i : h_i(w+e) = h_i(e) \neq h_i(e')$$

En effet, si e et e' sont des erreurs corrigibles, c'est qu'elles affectent moins de t bits chacune. $e+e'$ en affecte donc au maximum $2t$, or $d-1 \geq 2t$: $e+e'$ est une erreur détectable, donc $e+e' \notin C$, soit $\exists i : h_i(e+e') = h_i(e) + h_i(e') \neq 0$.

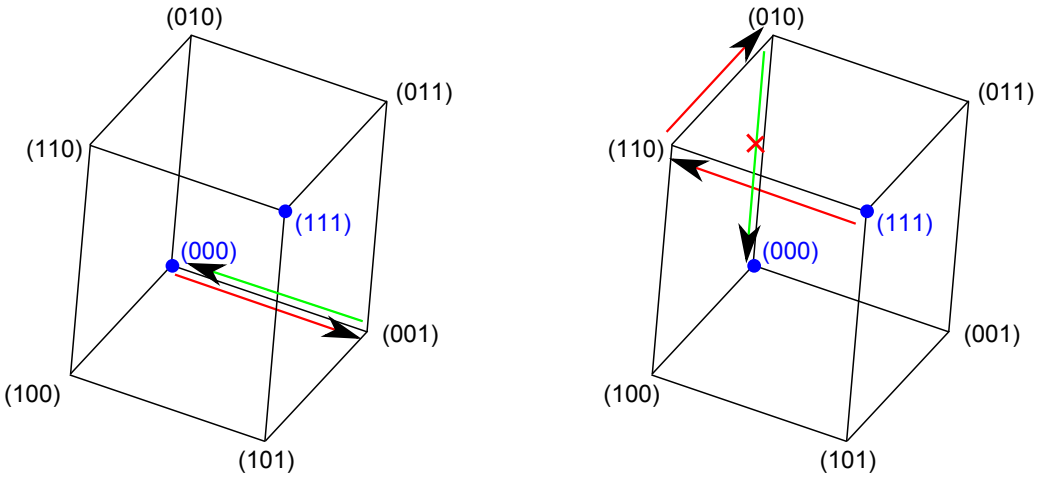


Figure 11 – Illustration de la notion de capacité de correction du même code. Les mots du code sont en bleu, et une erreur correspond à une flèche rouge. La correction au maximum de vraisemblance (c'est a dire qu'un vecteur erroné est corrigé par le vecteur du code qui est le plus proche) est notée par une flèche en vert. A gauche, une erreur survient, et la correction est forcément correcte. A droite, deux erreurs surviennent, et la correction est mauvaise : le vecteur corrigé est (000), le vecteur originel est (111). Ce code a une capacité de correction de $t = 1$.

Il est possible de définir l'application H, appelée la matrice contrôle de la manière suivante : $(e = (e1, ...eN)$ est l'erreur qui survient)

$$\begin{cases} \mathbb{F}_2^N \times \mathbb{F}_2^{N-k} & \to & \mathbb{F}_2^N \times \mathbb{F}_2^{N-k} \\ (w_1 + e_1, ..., w_N + e_N, a_1, ..., a_{N-k}) & \mapsto & (w_1 + e_1, ..., w_N + e_N, h_1(e), ..., h_{N-k}(e)) \\ & & = H(w_1 + e_1, ..., w_N + e_N, 0, ..., 0) \end{cases}$$

H est alors un endomorphisme inversible sur $\mathbb{F}_2^N \times \{0...0\}$; il est donc aisé de le transposer à la mécanique quantique.

Les deux propriétés précédentes nous disent que la quantité $h_1(e), ..., h_{N-k}(e)$, appelée syndrome, est distincte pour toute erreur de E. Connaître le syndrome, c'est donc connaître l'erreur, et pouvoir la corriger.

On peut donner en exemple d'application le code qui est à l'origine des deux figures 10 et 11. Dans cet exemple :

$$k = 1 \text{ et } N = 3$$

L'application d'encodage G est définie par :

$$G : (w, a_1, a_2) \mapsto (w, w, w)$$

Donc le code C est $C = \{(000), (111)\} = \text{Vect}\{G(0, a_1, a_2), G(1, a_1, a_2)\}$. La distance entre les vecteurs de C est de 3 donc les erreurs affectant moins de $d - 1 = 2$ bits peuvent être détectées et celles en affectant moins de $t = 1$ peuvent être corrigées.

Une base de l'espace dual est :

$$\{h_1 = (1, 1, 0), h_2 = (1, 0, 1)\}$$

ce qui permet de définir H par :

$$(w + e_1, w + e_2, w + e_3, a, a) \mapsto (w + e_1, w + e_2, w + e_3, e_1 + e_2, e_1 + e_3)$$

Il est alors possible de calculer le syndrome de chaque erreur pouvant survenir, c'est la table syndrome–erreur (table 1).

erreur	syndrome
abc	a+b,a+c
000	00
001	01
010	10
100	11
011	11
101	10
110	01
111	00

Table 1 – Table des erreurs. Le syndrome est unique tant qu'une seule erreur (ou moins) survient, au delà, il ne l'est plus : la capacité de correction de ce code est $t = 1$

Le protocole de transmission est alors simple :
- encoder message (appliquer G),
- transmettre le message codé,
- mesurer le syndrome du message reçu (appliquer H),
- appliquer la correction correspondant à l'erreur détectée grâce à la table syndrome–erreur,
- décoder le message (via l'inverse de $G|_C$).

3.2 Transposition au codage quantique

Avec les précautions qui ont été prises, la transposition au codage quantique est aisée et la définition de G et H s'adaptent immédiatement :

$$G : \left\{ \begin{array}{ccc} \mathcal{H}_2^{\otimes k} \otimes \mathcal{H}_2^{\otimes N-k} & \to & \mathcal{H}_2^{\otimes N} \\ |w_1, ..., w_k\rangle \otimes |a_1, ..., a_{N-k}\rangle & \mapsto & |w_1, ..., w_k, a_1 + w'_0 ..., a_{N-k} + w'_{N-k}\rangle \end{array} \right.$$

$$H : \left\{ \begin{array}{ccc} \mathcal{H}_2^{\otimes N} \otimes \mathcal{H}_2^{\otimes N-k} & \to & \mathcal{H}_2^{\otimes N} \otimes \mathcal{H}_2^{\otimes N-k} \\ |w_1 + e_1, ..., w_N + e_N\rangle & \mapsto & |w_1 + e_1, ..., w_N + e_N\rangle \\ \otimes |a_1, ..., a_{N-k}\rangle & & \otimes |a_1 + h_1(e), ..., a_{n-k} + h_{N-k}(e)\rangle \end{array} \right.$$

L'état quantique $|a_1, ..., a_{N-k}\rangle$ doit au préalable être préparé dans l'état $|0, ..., 0\rangle$.

G et H sont des involutions, donc unitaires, donc la mécanique quantique n'interdit pas la réalisation d'un circuit les codant ; en réalité, la réalisation d'un tel circuit ne fait appel qu'à l'opération $CNOT$.

L'exemple donné dans la section précédente se code donc très facilement : pour $x \in \{0, 1\}$, l'application G est :

$$G : |x\rangle|00\rangle \mapsto |x\rangle|0 + x, 0 + x\rangle$$

G est prolongée à $|\phi\rangle = a|0\rangle + b|1\rangle$ par linéarité : $G|\phi\rangle = a|000\rangle + b|111\rangle$. On retrouve le code proposée dans la section 1. L'écriture matricielle de G en tant qu'endomorphisme de $\mathbb{F}_2^N$ va permettre de réaliser le circuit quantique qui calcule G en tant qu'endomorphisme de $\mathcal{H}_2^{\otimes N}$. Cette écriture matricielle est :

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Le circuit correspondant est donné par la figure 12, le lien entre l'écriture matricielle est immédiat : si $i \neq j$ et $G_{ij} = 1$, alors il faut faire un $CNOT$ contrôlé par le j-ème qubit sur le i-ème : on reconnaît ${}^tG$ dans le circuit.

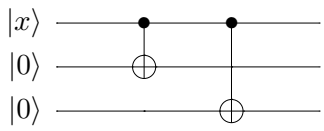


Figure 12 – Circuit d'encodage pour $G : |x\rangle|00\rangle \mapsto |x\rangle|0+x, 0+x\rangle$

Le calcul du syndrome par H se fait de la même manière qu'on a adapté G. L'écriture matricielle de H est :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

et le circuit qui code ceci, construit de la même manière que celui qui code G, est représenté sur la figure 13. La mesure du syndrome est alors effectuée sur les qubits non intriqués $|s_1\rangle$ et $|s_2\rangle$. La table syndrome-erreur (tableau 1, page 10), permet alors de corriger un bit-flip. Un lecteur attentif aura remarqué que la mesure de ce syndrome revient à tester l'égalité entre les qubits 1 et 2 et entre les qubits 1 et 3, ce qui avait été proposé dans la section 1.

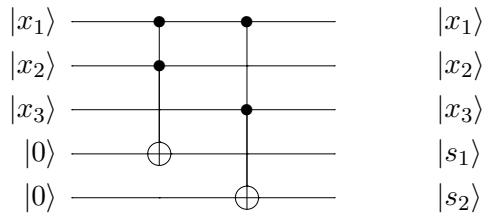


Figure 13 – Circuit d'encodage pour H

Il faut cependant remarquer que ce montage ne corrige qu'un seul bit-flip, le code est donc totalement vulnérable aux phase-flips ; pour corriger ces deux types d'erreur, il faut combiner deux codes correcteurs de manière adroite, dont un dans la base de Hadamard, afin qu'il corrige les erreurs de type phase-flip.

Un manière simple de protéger contre un bit-flip ou un phase-flip (ou un bit-flip et un phase-flip sur le même qubit) et de coder notre qubit $|\phi\rangle = a|0\rangle + b|1\rangle$ en $|\phi_1\rangle = a|000\rangle + b|111\rangle$, puis de passer dans la base de Hadamard et de coder $|\phi_1\rangle$. Pour cela, notons :

- $|+\rangle = \hat{H}_2|0\rangle = \dfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$,
- $|-\rangle = \hat{H}_2|1\rangle = \dfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Dans la base de Hadamard, $\hat{H}_2|\phi_1\rangle = a|+++\rangle + b|---\rangle$. Chaque qubit est alors encodé par le même code C, mais dans la base de Hadamard, alors ce sont les erreurs de phase qui sont corrigées. Le codage est le suivant :

- $|+\rangle = \dfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ est encodé par $\dfrac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$,

– $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ est encodé par $\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$

L'état de départ $|\phi\rangle$ est alors encodé par $|\phi_2\rangle$:

$$|\phi_2\rangle = \frac{1}{2\sqrt{2}}(a(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \\ +b(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle))$$

Ce code permet de corriger une erreur de phase sur chaque groupe de 3 qubits. Après avoir corrigé les erreurs (mais au maximum une par groupe de 3 qubits) de phase éventuelles, il faut décoder $|\phi_2\rangle$ pour obtenir $|\phi_1'\rangle$, qui est $|\phi_1\rangle$ dans la base de Hadamard, éventuellement affecté par des erreurs de bit. Il faut faire alors le changement de base, puis mesurer le syndrome et corriger une éventuelle erreur de bit sur $|\phi_1'\rangle$ pour obtenir $|\phi_1\rangle$, qu'il suffit alors de décoder afin d'obtenir $|\phi\rangle$.

Ce code est de distance $3 \leq d \leq 4$ car une seule erreur (quelconque, bit-flip et/ou phase-flip) sur un seul qubit peut être corrigée à coup sûr, ie. $t = 1$, donc $d = 3$ ou 4. Un calcul donnerait $d = 3$. Le fait qu'il soit possible de corriger une erreur de phase par groupe de 3 qubits n'est pas une amélioration du code : si 3 erreurs surviennent, elles arrivant au hasard, elles n'ont aucune raison de se répartir entre chacun de trois groupes, elle pourraient toutes affecter le même groupe, rendant la correction impossible.

La figure 14 présente le circuit d'encodage de ce code corrigeant une erreur de type bit-flip et/ou phase-flip.

Ce code n'est rien d'autre que le code de Schor qui avait déjà été introduit comme code permettant de corriger une erreur quelconque dans la section 2. Il n'est cependant pas optimal, il existe des codes bien meilleurs. Il faut choisir plus attentivement les codes pour corriger erreurs de bits et de phase, utiliser les propriétés du passage dans la base de Hadamard et les propriétés des erreurs. C'est ce que font les codes CSS et les codes stabilizer.

3.3 Application : les codes CSS

Les codes CSS ont étés développés simultanément par Calderbank et Schor d'un coté et par Steane de l'autre.

3.3.1 Construction

Les codes CSS reposent sur la propriété suivante :

Soit $C \subset \mathbb{F}_2^N$ un code, et $a, b \in \mathbb{F}_2^N$ deux vecteurs arbitraires. Définissons :

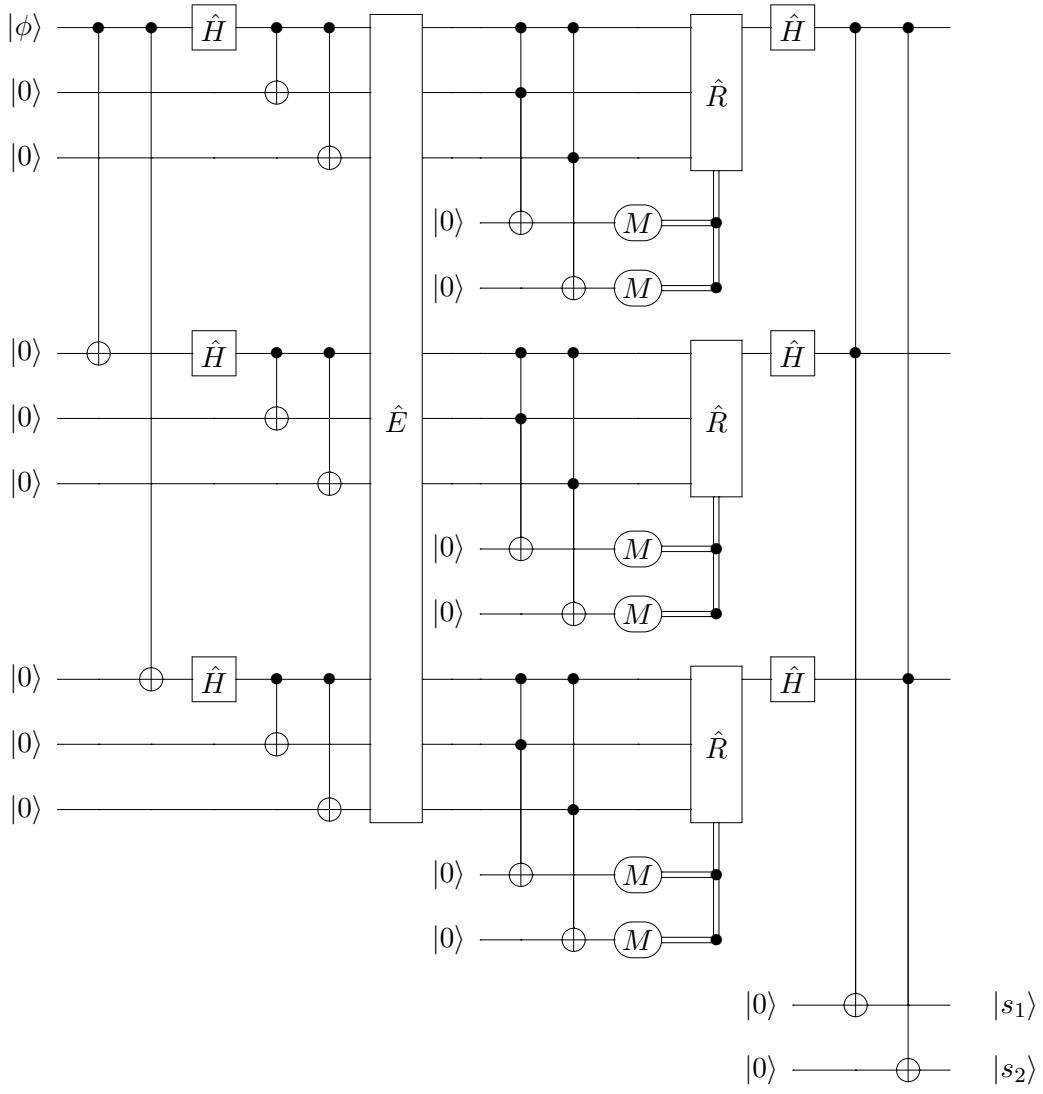
$$|\psi\rangle = \frac{1}{\sqrt{\#C}} \sum_{c \in C} (-1)^{a \cdot c} |c + b\rangle$$

Alors :

$$\hat{H}_2^{\otimes N} |\psi\rangle = \frac{(-1)^{a \cdot b}}{\sqrt{\#C^\perp}} \sum_{c \in C^\perp} (-1)^{b \cdot c} |c + a\rangle$$

Cette propriété peut s'interpréter par : faire une erreur de phase (c'est ce que représente le $(-1)^{a \cdot c}$, par exemple $N = 5$, a = 00101, signifie « faire une erreur de phase sur les qubits 3 et 5 ») sur le code C, c'est faire une erreur de bit (c'est ce que représente le $|a + c\rangle$, par exemple, pour $N = 5$, $a = 00101$ signifie « faire une erreur de bit les qubits 3 et 5 ») dans $C^\perp$ dans la base de Hadamard. Et réciproquement, faire une erreur de bit (c'est $|c + b\rangle$) dans un code C, c'est faire une erreur de phase (c'est $(-1)^{b \cdot c}$) sur $C^\perp$ dans la base de Hadamard.

Figure 14 – Circuit permettant de réaliser le code qui protège un qubit contre une erreur arbitraire. H correspond au passage dans la base de Hadamard, E à l'apparition de l'erreur, M est une mesure du syndrome et R est la désintrication (ie. le décodage) et la correction de l'éventuelle erreur



Cette propriété est très intéressante : il suffit d'utiliser deux codes C_1 et C_2 tels que $C_1^\perp \subset C_2$ et $C_2^\perp \subset C_1$ pour que les vecteurs de la forme :

$$|\psi_i\rangle = \frac{1}{\sqrt{\#C_2^\perp}} \sum_{c \in C_2^\perp} |c + w_i\rangle$$

(où w_i est un vecteur de $C_1/C_2^\perp$ (espace vectoriel quotient)) soient protégés contre les erreurs de bits et les erreurs de phases dans la limite des pouvoirs de correction de C_1 et C_2. Une preuve de ce résultat fondamental est donnée dans [2].

Le code que nous allons considérer est le nouveau code C engendré par les $|\psi_i\rangle$:

$$C = \text{Vect}\left\{ |\psi_i\rangle = \frac{1}{\sqrt{\#C_2^\perp}} \sum_{c \in C_2^\perp} |c + w_i\rangle \middle| w_i \in C_1/C_2^\perp \right\}$$

La correction est alors simple et utilise les résultats de codes classiques : notons H_1 et H_2 les matrices de contrôle des codes C_1 et C_2. Notons également e_x l'erreur de bit et e_z l'erreur de phase étant apparue durant la transmission de l'état $|\phi\rangle \in C$. Ainsi :

$$|\phi\rangle = \sum_i \alpha_i |\psi_i\rangle = \sum_i \sum_{c \in C_2^\perp} \alpha_i |c + w_i\rangle = \sum_{c \in C_1} \beta_c |c\rangle$$

car $c + w_i$ est un vecteur de $C_1/C_2^\perp$, donc un vecteur de C_1. L'état reçu après apparition des erreurs est :

$$|\phi'\rangle = \sum_{c \in C_1} \beta_c (-1)^{c \cdot e_z} |c + e_x\rangle$$

L'application de $H_1 : |x\rangle|0\rangle \mapsto |x\rangle|H_1(x)\rangle$ va permettre de récupérer le syndrome $H_1(e_x)$:

$$\begin{array}{rcl} H_1|\phi'\rangle & = & \sum_{c \in C_1} \beta_c (-1)^{c \cdot e_z} |c + e_x\rangle |H_1(c + e_x)\rangle \\ & = & \sum_{c \in C_1} \beta_c (-1)^{c \cdot e_z} |c + e_x\rangle |H_1(e_x)\rangle \\ & = & \left(\sum_{c \in C_1} \beta_c (-1)^{c \cdot e_z} |c + e_x\rangle\right) |H_1(e_x)\rangle \end{array}$$

Il suffit alors de mesurer le syndrome avec H_1 pour connaitre l'erreur e_x. La mesure du syndrome avec H_2 après être passé dans la base de Hadamard, donne accès au syndrome de l'erreur de phase, ce qui permet d'appliquer les corrections nécessaires.

3.3.2 Exemple

Dans les corps finis, il est possible d'avoir $C^\perp \subset C$, ainsi, si $C_1^\perp \subset C_1$, choisir $C_1 = C_2$, permettra alors d'avoir le même algorithme pour décoder les bit-flip et les phase-flip. Il est ensuite intéressant de choisir $\dim C_1 = \dim C_1^\perp + 1$, ainsi l'espace vectoriel quotient $C_1/C_1^\perp$ sera de dimension deux, ce qui permettra d'encoder un qubit.

Un tel code C_1 existe pour $N = 7$ et $k = 4$, et est engendré par les vecteurs :

$$\begin{array}{l} x_1 = (1001011) \\ x_2 = (0101110) \\ x_3 = (0010111) \\ x_4 = (0001101) \end{array}$$

$C_1 = \text{Vect}\{x_1, x_2, x_3, x_4\}$. Son orthogonal est engendré par les vecteurs :

$$\begin{array}{l} x_1 = (1001011) \\ x_2 = (0101110) \\ x_3 = (0010111) \end{array}$$

$C_1^\perp = \text{Vect}\{x_1, x_2, x_3, \} \subset C_1$. Ce code permet de corriger une erreur.

Deux représentants de $C_1/C_1^\perp$ sont $w_0 = (0000000)$ et $w_1 = x_4 = (0001101)$, ce qui permet de construire les deux vecteurs $|\psi_0\rangle$ et $|\psi_1\rangle$ qui seront protégés contre une erreur de bit et/ou de phase :

$$\begin{aligned} |\psi_0\rangle &= \frac{1}{\sqrt{8}} \sum_{i_1,i_2,i_3 \in \{0,1\}} |w_0 + i_1 x_1 + i_2 x_2 + i_3 x_3\rangle \\ |\psi_1\rangle &= \frac{1}{\sqrt{8}} \sum_{i_1,i_2,i_3 \in \{0,1\}} |w_1 + i_1 x_1 + i_2 x_2 + i_3 x_3\rangle \end{aligned}$$

Pour réaliser le circuit correspondant à ce code, il faut remarquer que des portes $CNOT$ ne suffiront pas car les $|\psi_0\rangle$ et $|\psi_1\rangle$ sont des états quantiques intriqués, donc pour calculer le générateur G, il faudra faire appel à $\hat{H}_2^{\otimes 3}$ pour obtenir la somme des 8 états intriqués :

$$\hat{H}_2^{\otimes 3} \otimes Id^{\otimes 4} |000xxxx\rangle = (|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)|xxxx\rangle$$

Les trois qubits intriqués vont correspondre à la sommation sur (i_1, i_2, i_3) qui est réalisée à l'aide de $CNOT$: un portail $CNOT$ contrôlé par le qubit correspondant à i_j est placé sur le qubit l si le l-ième bit de x_j vaut 1. C'est le circuit présenté sur la figure 15.

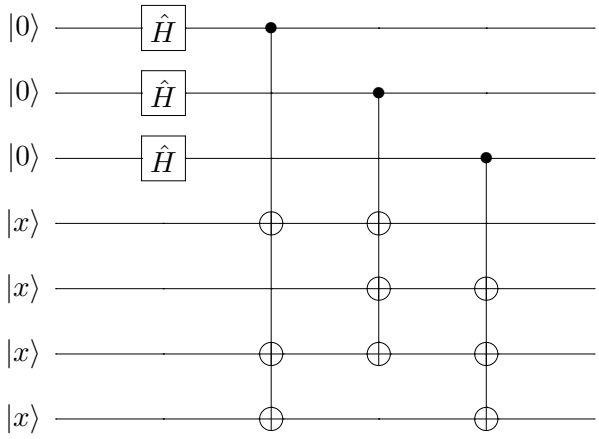


Figure 15 – Circuit d'encodage pour la sommation sur (i_1, i_2, i_3) qui réalise la sommation $\sum_{i_1,i_2,i_3 \in \{0,1\}} |x + i_1 x_1 + i_2 x_2 + i_3 x_3\rangle$

Avant de faire cette manipulation, il faut être dans l'état $a|w_0 = 0...0\rangle + b|w_1 = x_4\rangle$, ce qui est réalisé à l'aide de $CNOT$ contrôlés par $|\phi\rangle = a|0\rangle + b|1\rangle$. C'est le circuit présenté sur la figure 16.

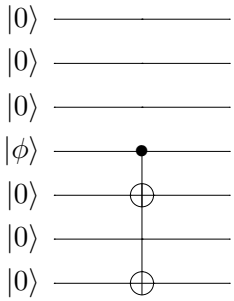


Figure 16 – Circuit d'encodage pour x_4, qui réalise la superposition $a|0\rangle + b|x_4\rangle$

Ces deux circuits s'agencent ensemble pour former le circuit d'encodage représenté par la figure 17.

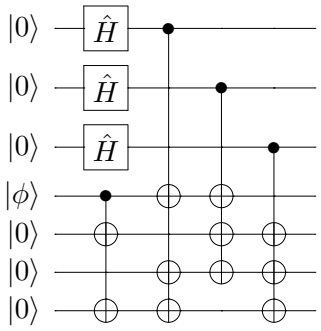


Figure 17 – Circuit d'encodage pour un code CSS

Le circuit de décodage ne présente pas de difficultés dans sa réalisation, et il s'obtient très simplement en fonction des générateurs de $C_1^{\perp}$ par la méthode qui a été présentée avant (utilisant la matrice), le circuit correspondant est représenté sur la figure 18. Les 3 qubits $|s_1 s_2 s_3\rangle$ correspondent au syndrome de l'erreur de phase et les trois qubits $|p_1 p_2 p_3\rangle$ correspondent au syndrome de l'erreur de phase. On remarquera qu'effectivement, le même circuit est utilisé pour calculer les deux syndromes, par choix de $C_1 = C_2$.

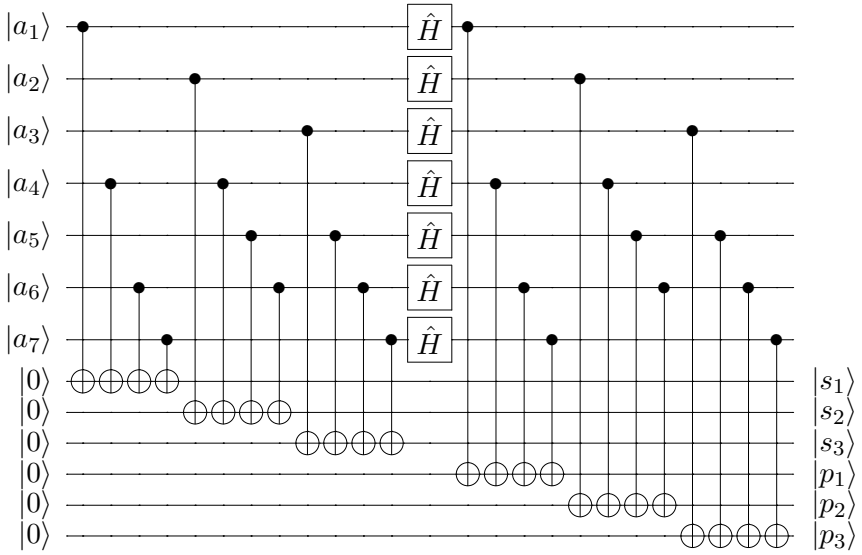


Figure 18 – Circuit de décodage pour un code CSS

Par rapport au code précédemment présenté, celui-ci est meilleur selon de divers critères : il utilise moins de qubits pour encoder (7 au lieu de 9), donc est plus robuste aux erreurs, il utilise moins de qubits pour calculer le syndrome (6 au lieu de 8) ce qui est avantageux, et permet la correction simultanée de l'erreur de bit et de l'erreur de phase. Il est cependant possible de construire des codes encore plus performants que celui-ci, c'est le cas des codes stabilizer, dont une version utilise uniquement 5 qubits pour en encoder 1.

Conclusion

Malgré les difficultés a priori, il est possible de faire des codes quantiques performants, qui sont essentiel pour le développement d'ordinateurs quantiques.

Nous avons vu, d'abord au travers d'exemple, que les difficultés posées par le théorème de non-clonage peuvent être contournées. Puis, après avoir formalisé ce qu'est une erreur, nous avons montré qu'il suffit en fait d'en corriger deux, les erreurs de bit et les erreurs de phase. Puis finalement, nous avons présenté une méthode pour construire des codes de plus en plus performants.

Le sujet des codes quantiques est cependant très vaste, et nous n'en avons traité qu'une petite partie. Néanmoins, grâce aux travaux qui ont été faits sur les codes classiques, beaucoup de résultats peuvent être dérivés facilement : les limites du codages, nommées bornes de Hamming quantiques, exprimant le mieux que peux faire une code en terme de rapport capacité de correction sur nombre de qubits rajoutés, des techniques plus avancés (utilisation de sur-corps de $\mathbb{F}_2$, polynômes, ...) permettant de faire des codes meilleurs – dont une, l'utilisation du sur-corps $\mathbb{F}_4$ de $\mathbb{F}_2$, se révèle étroitement liées aux erreurs quantiques : les éléments de $\mathbb{F}_4$ peuvent être, d'une certaine manière, vus comme les opérateurs $Id, \hat{X}, \hat{Y}, \hat{Z}$, ce qui rend tout code les utilisant très robustes vis-à-vis de ces erreurs.

Le sujet des codes quantiques est d'ailleurs, pour sa richesse, un domaine actif de recherche.

Références

[1] Michael A.Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum information*, Cambridge University press (2000).

[2] Ranee K. Brylinskiu, Goong Chen, *Mathematics of Quantum Computation*, Chapman & Hall/CRC (2002).

[3] Phillip Kaye,Raymond Laflamme and Michele Mosca, *An Introduction to Quantum Computing*, Oxford university press (2007).

[4] Jozef Gruska, *Quantum Computation*, McGraw-Hill (1999).