Centre de recherche Inria-Rocquencourt

Borne inférieure sur la capacité d'un canal quantique

Evaluation de l'entropie résiduelle sur des codes correcteurs quantiques

Anne Marin

Rapport de stage réalisé en 2009
sous la direction de Jean-Pierre Tillich

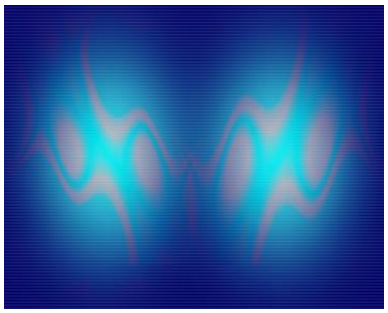
Mon stage s'est déroulé au sein de l' I.N.R.I.A. : l'Institut National de Recherche en Informatique et Automatique.
L'équipe SECRET dans laquelle je travaillais est la réunion des équipes de recherche en 'code' (comprendre codes correcteurs) et 'cryptographie'. Situé dans une lointaine contrée en banlieue de Versailles, le site de l'INRIA-Rocquencourt est un cadre atypique pour faire de la recherche. Protégé de part et d'autre par les fôrets de Fausses-Reposes, de Louvecienne et de l'Arboretum de Chèvreloup, on discute tout naturellement des derniers scoops de crypto, de quelles fonctions de hachage à la mode casser, des dernières attaques sur SHA 0,1,2,3..., autour d'un café, après l'expédidif repas du midi. Pour les nouveaux, c'est l'occasion de tout apprendre en programmation, du xor à la programmation parallèle intensive, la distance parcourue est à la hauteur de la disponibilité des membres de l'équipe. Une ambiance conviviale et un esprit d'équipe qui font briller un peu plus le monde de la recherche.

Table des matières

Problématique : La recherche en physique actuelle permet d'expérimenter différents type d'« éléments » pouvant être utilisés pour transporter, controler, coder de l'information quantique. D'un point de vue purement théorique, on peut, au préalable, étudier les propriétés physiques liées à l'information quantique, domaine d'étude assez jeune où beaucoup de choses restent encore à découvrir, où beaucoup de questions ouvertes restent encore à poser. Dans le cadre de mon stage, je me suis intéressée en particulier à une question primordiale en théorie de l'information, à savoir : La capacité d'un canal. Tout comme dans le cadre classique pour lequel elle a été défini au départ, la capacité d'un canal quantique veut représenter le nombre maximal d'information effective transmissible sur ce type de canal (bruité). C'est à dire, je veux savoir combien au maximum et dans l'idéal je peux récuperer d'unités d'information émise à partir d'un message recu erroné. Pour cela je m'autorise à coder l'information initiale, en y ajoutant de la redondance. Le taux d'information que je récupère dépends alors du bruit sur le canal, de la distribution et de la nature de ces erreurs (loi de probabilité sur le canal, ex : binaire symétrique), mais aussi de la capacité de correction du code et de la répartition/distribution des corrections possibles sur le message (Ensemble des valeurs possibles prises par l'entrée X, soit le code). D'après le théorème de Shannon, sur tout canal bruité, il existe un code de rendement idéal et maximal ; ce dernier définit la capacité du canal. Tout autre code possède un rendement inférieur.
Les définitions issues de Shannon ont été adaptée au cadre quantique. Ceci fera l'objet de la troisième partie de ce mémoire. Avant cela, la première partie sera consacrée à l'inévitable passage par la mécanique quantique, qui nous fournira les notions nécessaires à la présentation, en partie deux, des codes (correcteurs d'erreurs) quantiques. Nous étudierons une famille particulière de ces codes, les codes stabilisateurs, afin de tester la capacité, encore inconnue, du model de canal quantique de dépolaristation.

1 Introduction à l'Information Quantique : Que faut-il retenir de la mécanique quantique ?

Il existe actuellement plusieurs expérimentations physiques concluantes pour créer des **qubits** de manière plus ou moins satisfaisante. A titre d'exemples, on peut citer la polarisation d'un photon, un ion piégé, un atome à deux niveaux. Dans tous les cas, on nomme qubit *l'unité de mesure d'un signal quantique.* Lorsque les premières tentatives d'isolation de particules élémentaires ou plus généralement d'objets quantiques *individuels* aboutissent dans les années 1980, le qubit obtient par la même une réalité qui va justifier la naissance de l'information quantique dans la décennie suivante. Toutefois, aucun concept nouveau n'est alors necessaire. Il s'agit principalement d'adapter les notions de théorie de l'information à des objets issus de la mécanique quantique (ce qui n'est pas pour autant trivial).



Représentation artistique des fameux bits quantiques ou qubits.
Document U.Melbourne/Marc Coe.

1.1 Le Qubit

Définition 1. *Un qubit est un vecteur unitaire d'un espace de Hilbert de dimension 2 sur* $\mathbb{C}$.

Notation :
Soit $(\mathcal{H}, \|.\|_2)$ un espace de Hilbert de dimension 2 sur $\mathbb{C}$, muni de la norme $\|.\|_2$. On note

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathcal{H}$$
$$|1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \in \mathcal{H}$$

$(|0\rangle, |1\rangle)$ forme une base orthonormée de $(\mathcal{H}, \|.\|_2)$.

Remarques .

1. *Typiquement, on peut prendre* $\mathcal{H} = \mathbb{C}^2$.
2. *La notation de Dirac, habituelle chez les physiciens, utilisant les bra-kets* $\langle|, |\rangle$ *pour désigner un état quantique, s'est généralisée en science de l'information quantique. Dans l'idée de rester cohérent avec le sujet, nous suivrons donc ces notations.*

3. D'un point de vue « informatique », un qubit est une superposition linéaire (combinaison linéaire normalisée) des deux états possibles 0 et 1 d'un bit classique, en ce sens, on a établi une correspondance naturelle entre $\{0,1\}$ et les deux vecteurs de la base canonique de $\mathcal{H}$, $(|0\rangle, |1\rangle)$.

Autrement dit :

$$|\phi\rangle \text{ qubit } \overset{\text{par déf}}{\iff} \exists a,b \in \mathbb{C}, \begin{cases} |\phi\rangle = a|0\rangle + b|1\rangle \\ |a|^2 + |b|^2 = 1 \end{cases}$$

On dit que a et b sont les **amplitudes de probabilité**.

1.1.1 Produit scalaire hermitien sur $\mathcal{H}$

On utilise le produit scalaire hermitien usuel sur $\mathbb{C}^2$ (vu comme un espace de Hilbert de dimension 2 sur $\mathbb{C}$). Toutefois, on utilise les notations de Dirac :

Définitions-Notations . *Soit $|\phi\rangle \in \mathcal{H}$. $\langle\phi| := |\overline{\phi}\rangle^t$.*

On utilisera indifférement les notations $\langle\ ,\ \rangle$, $.\ ,$ ou de Dirac pour parler du produit scalaire hermitien :

Définition 2. $\forall |\phi\rangle, |\psi\rangle \in \mathcal{H}$,

$$\begin{aligned} \langle|\phi\rangle, |\psi\rangle\rangle &:= (\bar{a}\langle 0| + \bar{b}\langle 1|)(c|0\rangle + d|1\rangle) \\ &= \bar{a}c + \bar{b}d \\ &= \langle\phi|.|\psi\rangle \end{aligned}$$

La notation de Dirac contracte les notations des vecteurs et du produit scalaire hermitien en une seule écriture :
$\langle\phi|\psi\rangle := \langle|\phi\rangle, |\psi\rangle\rangle$.

Dans toute la suite, $\mathcal{H}$ désigne un espace de Hilbert de dimension 2 sur $\mathbb{C}$, muni du produit scalaire hermitien $\langle\ ,\ \rangle$.

1.1.2 Produit tensoriel

Rappel :
Soient E, F deux espaces vectoriels normés de dimension respective n et m et de base respective $(e_1, \ldots, e_n)$ et $(e'_1, \ldots, e'_m)$.
E et F induisent un espace vectoriel normé $E \otimes F$, de dimension $n \times m$, de base notée $(e_i \otimes e'_j)_{i,j}$, muni du produit tensoriel défini par

$$\forall \quad \lambda = \sum_{i=1}^{n} \lambda_i e_i \in E, \quad \mu = \sum_{i=1}^{m} \mu_i e'_i \in F, \boxed{\lambda \otimes \mu = \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_i \mu_j e_i \otimes e'_j}$$

Conséquence et Notations :
$\mathcal{H}^{\otimes n} := \underbrace{\mathcal{H} \otimes ... \otimes \mathcal{H}}_{\text{n termes}}$ est un espace de Hilbert de dimension 2^n (sur $\mathbb{C}$).
En s'autorisant à écrire $|i\rangle \otimes |j\rangle = |ij\rangle = |i \otimes j\rangle, \forall i \in F_2{}^p, j \in F_2{}^q$ tels que $p \times q = n$, on notera $(|i\rangle)_{i \in F_2{}^n}$ la base canonique de $\mathcal{H}^{\otimes n}$.

Exemple :
Soit $|\phi\rangle = a|0\rangle + b|1\rangle, |\psi\rangle = a'|0\rangle + b'|1\rangle \in \mathcal{H},\ a, a', b, b' \in \mathbb{C}$, tels que
$|a|^2 + |b|^2 = 1,\ |a'|^2 + |b'|^2 = 1,$

$$\begin{aligned} |\phi\rangle \otimes |\psi\rangle &= aa'|0\rangle \otimes |0\rangle + ab'|0\rangle \otimes |1\rangle + ba'|1\rangle \otimes |0\rangle + bb'|1\rangle \otimes |1\rangle \\ &= aa'|00\rangle + ab'|01\rangle + ba'|10\rangle + bb'|11\rangle \end{aligned}.$$

Définition 3. *On appelle mot de longueur n, ou registre à n qubits un vecteur unitaire de $\mathcal{H}^{\otimes n}$.*

On peut donc, à l'aide du produit tensoriel, créer des registres de n qubits qui vérifient bien que $\forall(|\phi\rangle, |\psi\rangle) \in \mathcal{H}^{\otimes n} \times \mathcal{H}^{\otimes m}, |\phi\rangle \otimes |\psi\rangle$ est un vecteur unitaire de $\mathcal{H}^{\otimes(n \times m)}$. Par contre, il existe une infinité d'élements dans $\mathcal{H}^{\otimes n}$ qui soit de norme 1 mais que l'on ne peut pas écrire sous la forme du produit tensoriel de (au moins) deux registres.
Lorsque qu'une telle transformation n'est pas possible, on dit que l'état du registre est *enchevétré* ou *intriqué*. **L'intrication quantique** est un phénomène totalement propre à la physique quantique et source de surprenantes conséquences, voire un peu magiques, comme l'idée de téléportation quantique... Mais ceci est une autre histoire.
Il est toutefois intéressant de considérer que ce phénomène joue un rôle dans le comportement particulier de nos qubits, et il est toujours intéressant de le garder dans un coin de sa mémoire.

1.2 Mesure et Projection

Un registre quantique est par nature une superposition d'état. Pour manipuler cet état, la première chose à faire est de pouvoir coder l'information. En mécanique quantique, on ne peut pas coder de la manière qu'en classique, où toutes les opérations possibles dans un corps fini sont programmables (opérations booleénnes élémentaires), et par suite, selon la thèse de Church-Turing, toutes fonctions naturellement descriptibles par un algorithme. [1] Il nous faut changer nos habitudes, puisqu'au lieu de manipuler des chaines de variables discrètes, typiquement de longueur n, nous vivons désormais sur des sphères unité, typiquement de dimension 2^n, sur lesquelles il faudra rester. C'est pourquoi l'évolution d'un registre quantique est décrit par une transformation unitaire, autrement dit, par une matrice $U \in M_{2^n}(\mathbb{C})$, unitaire ($\overline{U}^t.U = I$), telle que

$$|\psi\rangle \mapsto U.|\psi\rangle \quad \forall |\psi\rangle \in \mathcal{H}^{\otimes n}.$$

[1] Algorithme : Pour une tache donnée, décomposition en une suite finie d'opérations (ou étapes) élementaires permettant de la réaliser.

De façon encore plus prononcée, la manière de connaitre l'état dans lequel le registre se trouve diffère radicalement du monde classique. Par exemple, supposons que nous traitions du qubit $a|0\rangle + b|1\rangle$. Cela signifie que le qubit a une probabilité $|a|^2$ d'être dans l'état $|0\rangle$, et une probabilité $|b|^2$ d'être dans l'état $|1\rangle$. Mais la seule manière de connaitre cet état c'est de le mesurer. Cette mesure nous apporte alors l'information de la connaissance de cet état mais elle a aussi une autre implication. En effet, une fois mesuré, le qubit est dans un état déterminé (celui que l'on vient de mesurer).
En d'autre terme, il n'est pas équivalent d'exister dans un certain état et de connaitre l'état dans lequel on est. Il s'agit là d'un des principes fondamental de la mécanique quantique : la mesure peut modifier l'état du qubit.
Par exemple, si je mesure mon qubit $a|0\rangle+b|1\rangle$ selon la base $(|0\rangle, |1\rangle)$, et que j'obtiens le résultat $|0\rangle$, je suis sure à partir de maintenant que mon qubit est dans l'état $|0\rangle$, toute mesure ulterieure me confirmera cet état avec certitude (avec probabilité 1), et l'état du qubit est ainsi devenu $|0\rangle$.

Cette mesure correspond exactement à la *projection* d'un état pur quantique sur un sous espace orthogonal :

Définition 4. *La **mesure** d'un état pur quantique est définie par une décomposition de l'espace ambiant en somme directe de sous-espace orthogonaux*

$$\mathcal{H}^{\otimes n} = \overset{\perp}{\bigoplus_i} E_i$$

Action :
Notons Π_i la projection sur E_i et soit $|\psi\rangle \in \mathcal{H}^{\otimes n}$. Alors

1. *$|\psi\rangle$ est projeté sur E_i avec une probabilité $\|\Pi(|\psi\rangle\|^2$.*
2. *Le nouvel état après la mesure est $|\tilde{\psi}\rangle = \dfrac{\Pi_i(|\psi\rangle)}{\|\Pi_i(|\psi\rangle)\|}$.*
3. *Le résultat de la mesure est " l'état a été projeté dans le sous espace E_i".*

Remarque . *On remarque que la multiplication par un scalaire de norme 1 ne modifie pas la mesure d'un état quantique, au sens où l'état quantique sera projeté sur le même sous espace, avec la même probabilité. Et on se permettra, si besoin, de négliger le coefficient λ.*

1.3 Introduction au calcul quantique et circuit quantique

Comme nous l'avons vu précedement, l'idée du calcul quantique est d'adapter les bits d'information à la nature quantique. Plus précisement, supposons que l'on veuille traiter un vecteur $(a_1, \ldots, a_n)$ de n bits. Nous pourrons à la place coder un registre de n qubits $|a_1 \ldots a_n\rangle$. Jusque là, pas beaucoup d'innovation. Mais supposons de plus que pour un problème donné, on ait besoin de tester T valeurs possibles, notées $x_1, \ldots, x_T$, d'un vecteur binaire $(a_1, \ldots, a_n)$, dans le but de trouver une valeur vérifiant une certaine propriété. Pour cela, nous effectuerons au plus T tests. En passant à une version quantique du problème, nous pourrions coder un seul registre

de n qubits, de la forme $|\psi\rangle = \sum_{1\leq i\leq T} \alpha_i |x_i\rangle$, où $\sum_{1\leq i\leq T} |\alpha_i|^2 = 1$, sur lequel nous testerions une propriété. Et pour terminer, la projection de $|\psi\rangle$ selon une base complétée à partir de $(x_1, \ldots, x_T)$ nous donnerait le résultat du problème initial.
Nous savons qu'il est possible de créer un registre quantique, et qu'il est possible de coder ce registre grâce à n'importe quelle transformation unitaire.
Remarque (conséquence immédiate mais fondamentale) : la transformation étant unitaire, elle est inversible. Ceci est un principe fondamental en calcul quantique : Toute opération doit être inversible.

1.3.1 Portes logiques

Rappelons tout d'abord que toute opération booléenne peut par exemple être décomposée selon les deux opérateurs NAND (NON-ET) et NOR (NON-OU). C'est pourquoi on les nomme **portes universelles**. Nous aimerions de nouveau adapter les choses au cadre quantique, typiquement afin d'autoriser le codage des vecteurs de base (de $\mathcal{H}^{\otimes n}$) par des opérations booléennes. Néanmoins, la plupart des portes logiques usuelles sont irréversibles, la question est donc de trouver des opérations "équivalentes" mais réversibles.

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité de l'opérateur irréversible NAND.

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Table de vérité de l'opérateur irréversible NOR.

Le calcul reversible est certe une contrainte du calcul quantique, mais finalement, au prix de rajouter des **bits de contrôles**, toutes les opérations élémentaires classiques peuvent être adaptées. L'idée est de coder des registres quantiques, en ne prenant en compte que les vecteurs de type $|i\rangle, i \in {\mathbb{F}_2}^n$, (typiquement les vecteurs de base) et en considérant que i représente une chaine de bits. Nous cherchons donc des transformations unitaires impliquant des opérations classiques élémentaires sur les « bits de i ».

porte CNOT
Pour obtenir des opérateurs inversibles, le calcul quantique nécessite des opérations de contrôle. Le premier exemple en a hérité le nom, c'est l'opérateur CNOT (pour Control NOT) :
La porte CNOT est définit par l'action sur $i \in {\mathbb{F}_2}^2$ de l'application : $(x, y) \mapsto (x, x \oplus y),\ x, y \in \mathbb{F}_2$.
Si $y = 0$, l'opérateur CNOT agit comme la fonction COPY classique sur x et rend (x, x). Si $y = 1$, il agit comme la fonction classique NOT sur x et rend $(x, 1 \oplus x)$. On

dit que x est le bit de **contrôle** et y le bit **cible**.

Sous forme matricielle, $CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

Sur le même principe, pour reproduire des fonctions booléennes ayant 2 entrées, il faut donc prendre deux bits de contrôles.

porte de Toffoli :
La porte de Toffoli permet d'effectuer toutes les opérations booléennes élémentaires ayant deux entrées.
Elle est définie par l'action sur $i \in {\mathbb{F}_2}^3$ de l'application : $(x, y, z) \mapsto (x, y, z \oplus yx)$, $x, y, z \in \mathbb{F}_2$, où x et y sont les bits de contrôle.
Par exemple, si $z = 1$, la porte de Toffoli agit comme la fonction NAND classique sur les bits de contrôle.
On peut montrer qu'elle permet de reproduire tous les circuits logiques classiques. On dit que la porte de Toffoli est une porte logique universelle pour toutes les opérations (sous forme) réversibles de la logique booléenne.

Transformation unitaire sur un qubit :
Voici les plus célèbres opérateurs unitaires élémentaires :

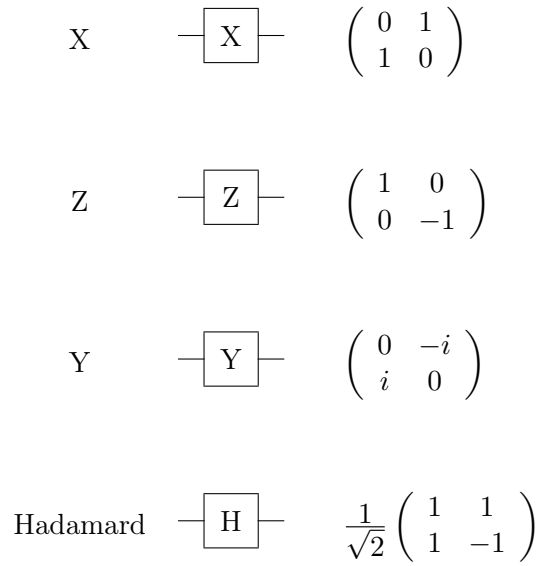


FIG. 1 – Noms, symboles et écriture matricielle des principales transformations unitaires sur un qubit.

Remarque . *Les opérateurs X, Y, Z vont nous être indispensable pour la suite. On les nomme opérateurs de Pauli. Le X correspond à l'opération NOT (la seule*

opération logique booléenne directement réversible). Elle vérifie donc
$X|0\rangle = |1\rangle$
$X|1\rangle = |0\rangle$
Z et de Y sont également des opérations quantiques élémentaires :
$Z|0\rangle = |0\rangle$
$Z|1\rangle = -|1\rangle$
$Y|0\rangle = i|1\rangle$
$Y|1\rangle = -i|0\rangle$

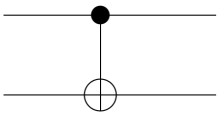
Théorème 1. *Toute transformation unitaire peut se décomposer en produit (tensoriel) de transformations unitaires sur un qubit et de portes cNOT.*

Démonstration 1. *Conséquence immédiate du "théorème de réduction des endomorphismes unitaires".*

1.3.2 Circuit quantique

Circuit quantique représentant la porte CNOT :

Exemples :



Exemple :

Etat initial	porte CNOT	Etat final
$\|\varphi\rangle = \|10\rangle$	$\|1\rangle$ — $\|1\rangle$, $\|0\rangle$ — $\|1\rangle$	$\|\varphi\rangle = \|11\rangle$

• :	qubit de contrôle
$\oplus$:	somme modulo 2
trait horizontal :	évolution
trait vertical :	produit tensoriel (l'ordre n'a pas d'importance)
carré :	portes logiques (CNOT, H, X, Y, Z, CU (Controle (d'une transformation) Unitaire))

2 Codes correcteurs d'erreurs quantiques

Les codes correcteurs d'erreurs quantiques, ou **QECC**, pour Quantum Error-Correcting Codes, existent dans le même but qu'en classique : pour protéger l'information. Toutefois, contrairement au cas classique, la correction d'erreur d'un système quantique est nécessaire au plus bas degré de l'echelle. Typiquement, il ne serait pas nécessaire de protéger, à l'aide de codes correcteurs classiques, les bits circulants à l'intérieur d'un ordinateur, si on considère que la transmission y est « instantanée ». Les codes correcteurs classiques prennent tout leur sens lors de transmission par réseaux (type téléphonie, télévision, internet) ou sur des supports endommageables (type CD). Au niveau quantique, la probabilité d'erreur existe dès la création du qubit, et c'est pourquoi, même pour une transmission aussi brève que celle existant à l'intérieur d'un ordinateur, on envisage d'utiliser les QECC pour lutter contre le phénomène de *décohérence*, qui menace continuellement tout système quantique.
Les QECC pourraient également servir de **répétititeurs** lors de transmissions de signal quantique sur de longue distance, comme pour la transmission quantique de clefs secrètes utilisée en cryptographie quantique.

2.1 Définition

Les erreurs quantiques sur le canal de dépolarisation

Une erreur est un opérateur unitaire agissant sur un registre quantique.
Soit un qubit $|\varphi\rangle = a|0\rangle + b|1\rangle$.
Un qubit passant par un **canal de dépolarisation** peut subir trois types d'erreurs :

- Erreur de type « inversion de bit » ou *bit-flip* : $|\varphi\rangle$ se transforme en $a|1\rangle + b|0\rangle$. C'est en fait l'image de $|\varphi\rangle$ par l'opérateur de Pauli X. (défini précédemment).
- Erreur de type « inversion de phase » , ou *phase-flip* : $|\varphi\rangle$ se transforme en $a|0\rangle - b|1\rangle$. C'est en fait l'image de $|\varphi\rangle$ par l'opérateur de Pauli Z.
- Les deux erreurs se produisent en même temps : $|\varphi\rangle$ se transforme en $a|1\rangle - b|0\rangle$. A une « phase » près, c'est l'image de $|\varphi\rangle$ par l'opérateur de Pauli Y.
  Remarque : Y a été défini par $Y = iXZ$ pour être être hermitienne. L'image d'un qubit par Y ne correspond donc pas exactement à celle de XZ. Toutefois, on a vu que la mesure quantique ne fait pas la différence entre $|\varphi\rangle$ et $\lambda|\varphi\rangle$, si $|\lambda|^2 = 1$. Au sens de la mesure, ces états sont donc équivalents, et on peut donc considérer que Y correspond bien à la réalisation conjointe des deux erreurs X et Z.

Propriété 1. *$\mathcal{G}_1 = \{I, X, Y, Z\} \times \{\pm I, \pm iI\}$ forme un groupe multiplicatif vérifiant*
- *$\forall\ P \in \mathcal{G}_1$, P est unitaire.*
- *$\forall\ P,\ Q \in \mathcal{G}_1$, P et Q commutent ou anticommutent.*

*On le nomme le **groupe de Pauli** (d'ordre 16).*

Propriété 2. *$\mathcal{G}_1$ induit un groupe d'ordre 4^{n+1} :*
$\mathcal{G}_n = {\mathcal{G}_1}^{\otimes n} = \{I, X, Y, Z\}^{\otimes n} \times \{\pm I, \pm iI\}$
(avec les mêmes propriétés que $\mathcal{G}_1$).

Le canal de dépolarisation est l'adaptation quantique d'un canal symétrique, dans lequel on rappelle que chaque composante du mot de code est soit reçue parfaitement

soit est échangée pour un des autres Card(F)-1 symboles, uniformément, où F est l'alphabet fini de définition du code.
En identifiant l'échange de symbôles à une erreur possible, sur un canal symétrique, on a donc une répartition uniforme de Card(F)-1 erreurs possibles. On définit le canal de dépolarisation comme un canal sans mémoire suivant une loi de probabilité uniforme P, vérifiant, pour une certaine probabilité $p \in [0,1]$:
$\begin{cases} P(E = \mathcal{E}) = p/3 & \forall\ \mathcal{E} \in \{X, Y, Z\} \\ P(E = I) = 1 - p \end{cases}$
où E est la variable aléatoire discrète associée à l'erreur sur un qubit.

Encodage

Soient $k < n$ et notons $\mathcal{S}_{\mathcal{H}^{\otimes k}}(0,1)$ la sphère unité de $\mathcal{H}^{\otimes k}$, c'est à dire l'ensemble des registres quantiques de longueur k.

Définition 5.

Le codage est un opérateur unitaire U de $\mathcal{H}^{\otimes n}$ dans $\mathcal{H}^{\otimes n}$

Le code est l'image par U de $\mathcal{S}_{\mathcal{H}^{\otimes k}}(0,1) \otimes |\psi_0\rangle$, où $|\psi_0\rangle$ est un registre cible de longueur $n-k$.
Dans la pratique, on utilise $|0_{n-k}\rangle = \underbrace{|0 \dots 0\rangle}_{(n\text{-}k)\ termes}$ comme registre cible.

Correction d'erreur

La correction d'erreur se déroule selon deux étapes génériques :

1. La mesure : La mesure est définie par un décomposition de l'espace ambiant en somme de sous espace orthogonaux *adaptés au code*, c'est-à-dire $\mathcal{H}^{\otimes n} = \overset{\perp}{\oplus} \mathscr{E}_i C$, et à chaque E_i, on associe une erreur $\mathscr{E}_i$ et on pose $\mathscr{E}_0 = I$. En projetant un registre selon cette base, d'une part, on met en évidence l'erreur $\mathscr{E}_i$ correspondant à E_i. D'autre part, le code étant un des sous espace de décomposition de $\mathcal{H}^{\otimes n}$, on ne modifie pas l'état du registre après la mesure (la projection d'un vecteur de C sur $\overset{\perp}{\oplus} \mathscr{E}_i C$ est le vecteur lui même).
2. La correction : On applique l'opérateur unitaire ${\mathscr{E}_i}^{-1} = \overline{\mathscr{E}}^t$ pour retrouver l'état initiale.

2.2 Exemple : Le code de Shor à 9 qubits

Le code de Shor à 9 qubits est une adaptation quantique du code à répétition classique à 3 bits. Il est défini sur le canal de dépolarisation et protège $k = 1$ qubit. Il est construit avec la composition de deux sous codes : les codes **bit-flip** et **phase-flip** à 3 qubits.

2.2.1 Le code bit flip à trois qubits

Ce code est conçu pour le modèle de canal bit-flip. Le canal bit-flip est un canal sans mémoire déterminé par une probabilité $p \in [0,1]$ qu'un qubit du registre subisse

une erreur X : $\begin{cases} P(E=X)=p \\ P(E=I)=1-p \end{cases}$

Codage : on utilise l'opération dite de « copie », qui vérifie :

$$\begin{aligned} |0\rangle \otimes |00\rangle &\mapsto |000\rangle \\ |1\rangle \otimes |00\rangle &\mapsto |111\rangle \end{aligned} \tag{1}$$

On peut donner explicitement un opérateur unitaire vérifiant (1), en posant par exemple

$$\begin{aligned} U_{C2NOT} &:= \begin{bmatrix} I\otimes I & & \\ & & X\otimes X \end{bmatrix}. \\ &= \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \end{bmatrix}. \end{aligned}$$

Alors on a $C = U_{C2NOT}(|\psi\rangle \otimes |00\rangle) \mid |\psi\rangle \in \mathcal{H}, \| |\psi\rangle \| = 1\}$
$= \{a|000\rangle + b|111\rangle \mid a,b \in \mathbb{C},\ |a|^2+|b|^2=1\}$
Ce code permet de protéger tout registre de trois qubits d'un bit-flip.

Décodage et correction :

1. Mesure : Pour déceler l'erreur produite, on projette le vecteur recu $|\tilde{\psi}\rangle$ selon la décomposition suivante :

$$\mathcal{H}^{\otimes 3} = Vect(|000\rangle,|111\rangle) \oplus Vect(|100\rangle,|011\rangle) \oplus Vect(|010\rangle,|101\rangle) \oplus Vect(|001\rangle,|110\rangle)$$

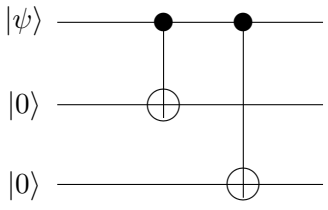


FIG. 2 – Circuit d'encodage du code bit-flip à 3 qubits

Notons :

$$\begin{array}{l} E_0 = Vect(|000\rangle, |111\rangle) \\ E_1 = Vect(|100\rangle, |011\rangle) \\ E_2 = Vect(|010\rangle, |101\rangle) \\ E_3 = Vect(|001\rangle, |110\rangle) \\ \text{et } \Pi_0, \Pi_1, \Pi_2, \Pi_3 \text{ les projections respectives associées.} \end{array} \tag{2}$$

On suppose que $|\tilde{\psi}\rangle$ subit au plus une erreur de bit-flip. Alors il existe une unique projection non nulle de $|\tilde{\psi}\rangle$ sur $\oplus E_i$ correspondant à au plus une unique erreur.

2. Correction : Avec les mêmes notations, si j=0, il n'y a pas d'erreur et on ne fait rien, sinon, on applique $X^{-1} = X$ au $j$ième qubit pour retrouver l'état quantique initialement encodé.

Par ailleurs,

$$\begin{aligned} \sum_{i=0}^{3} \Pi_i(|\tilde{\psi}\rangle) &= \Pi_j(|\tilde{\psi}\rangle),\ j \in \{0 \dots 3\} \\ &= |\tilde{\psi}\rangle \end{aligned}$$

Donc la projection laisse l'état quantique inchangé.

Ce code est donc capable de *corriger* une erreur de type bit-flip. Par ailleurs, il est aussi capable d'en *détecter* deux. En effet, si exactement deux erreurs de bit-flip surviennent sur deux qubits distincts, l'état sera projeté dans un autre sous espace que E_0, mais on va confondre ces deux erreurs avec une erreur agissant sur le troisième qubit ce qui nous mènerait, après correction, à un état quantique pur mais différent du qubit que l'on protège.

2.2.2 Le code phase-flip à trois qubits

Ce code est conçu pour le modèle de canal phase-flip. Le canal phase-flip est un canal sans mémoire déterminé par une probabilité $p \in [0, 1]$ qu'un qubit subisse une erreur Z :

$$\begin{cases} P(E = Z) = p \\ P(E = I) = 1 - p \end{cases}$$

Codage : Pour repérer une erreur de type Z, on code les qubits grâce à la matrice de Hadamard qui vérifie :

$$\begin{array}{l} H|0\rangle = \dfrac{(|0\rangle + |1\rangle)}{\sqrt{2}}, \text{ noté } |+\rangle \\ H|1\rangle = \dfrac{(|0\rangle - |1\rangle)}{\sqrt{2}}, \text{ noté } |-\rangle \end{array} \tag{3}$$

Le codage phase-flip est défini par l'application :

$$|0\rangle \otimes |00\rangle \mapsto \frac{(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)}{2\sqrt{2}} = |+\rangle|+\rangle|+\rangle$$
$$|1\rangle \otimes |00\rangle \mapsto \frac{(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{2\sqrt{2}} = |-\rangle|-\rangle|-\rangle$$

Autrement dit, $C = \{a|+++\rangle + b|---\rangle \mid a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1\}$
Ce codage correspond par exemple à la matrice unitaire U_{phase} définie par :
$U_{phase} := (H \otimes H \otimes H) \otimes U_{C2NOT}$

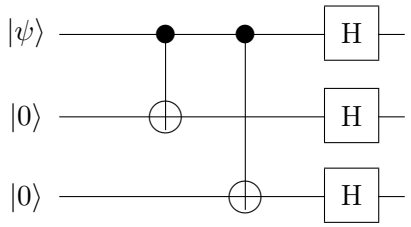


FIG. 3 – Circuit d'encodage du code phase-flip à trois qubits

Décodage et correction :

1. Mesure : L'inversion de phase agit comme une inversion de bit sur la base $(|+++\rangle, |---\rangle)$, ainsi on projette le vecteur encodé sur $\mathcal{H}^{\otimes 3}$ selon la décomposition :

$$\mathcal{H}^{\otimes 3} = Vect(|+++\rangle, |---\rangle) \oplus Vect(|-++\rangle, |+--\rangle) \oplus$$
$$Vect(|+-+\rangle, |-+-\rangle) \oplus Vect(|++-\rangle, |--+\rangle)$$

2. Correction : Le vecteur ayant subi au plus une erreur Z, il existe une unique projection non nulle dans cette décomposition, écrit dans l'ordre cela nous donne le numéro du qubit à corriger, ce que l'on fait en lui appliquant $Z^{-1} = Z$, excepté pour 0 qui correspond à pas d'erreur, auquel cas on ne fait rien.

De manière analogue au code bit-flip, on montre que le code phase-flip à trois qubits est capable de corriger une erreur de phase-flip et d'en détecter deux.
Exemple :
Supposons que le registre $|\phi\rangle = |+++\rangle = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes 3}$ soit transformé en $|\tilde{\phi}\rangle = \frac{1}{2\sqrt{2}}(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)(|0\rangle + |1\rangle) = |--+\rangle$.
Alors $|\tilde{\phi}\rangle$ est projeté sur $Vect(|++-\rangle, |--+\rangle)$. On détecte donc effectivement une erreur. Toutefois pour ce code, la mesure de cet espace correspond à l'erreur Z agissant sur le troisième qubit. Pour corriger, on appliquerait donc : $(I \otimes I \otimes Z)^{-1}|\tilde{\phi}\rangle = (I \otimes I \otimes Z)|\tilde{\phi}\rangle = ((\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)^{\otimes 3} = |---\rangle \neq |\phi\rangle$.

Le code de Shor :

En composant les codes bit-flip et phase flip, le code de Shor permet de protéger un qubit de toute erreur (de Pauli) possible en utilisant un registre de 9 qubits.

Codage : On note, pour $\epsilon \in \{0,1\}$, $|\epsilon_t\rangle := \underbrace{|\epsilon\rangle \otimes ... \otimes |\epsilon\rangle}_{\text{t termes}}$

On définit l'image du codage par l'image de la base :

$$\begin{aligned} |0\rangle \otimes |0_8\rangle &\mapsto \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \text{ noté } |0_L\rangle \\ |1\rangle \otimes |0_8\rangle &\mapsto \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \text{ noté } |1_L\rangle \end{aligned} \tag{4}$$

Pour expliciter une matrice d'encodage, on cherche une transformation unitaire $\mathcal{V}$ vérifiant (4). On peut définir cette matrice de la manière suivante : pour $|\psi\rangle$ un vecteur unitaire de $\mathcal{H}$,

$$\begin{aligned} \mathcal{V}(|\psi\rangle \otimes |0_8) &= W(U_{phase}(|\psi\rangle \otimes |00\rangle) \otimes |0_6\rangle) \\ &= W((H \otimes H \otimes H)(U_{C2NOT}(|\psi\rangle \otimes |00\rangle)) \otimes |0_6\rangle) \end{aligned}$$

où W est une matrice de $M_{2^9}(\mathbb{C})$, que l'on a défini par bloc :

$$W = U_{C2NOT} \otimes U_{C2NOT} \otimes U_{C2NOT}$$

Finalement : $\mathcal{V}((a|0\rangle + b|1\rangle) \otimes |0_8) = a\big(U_{C2NOT}(H|0\rangle \otimes |00\rangle)\big)^{\otimes 3} + b\big(U_{C2NOT}(H|1\rangle \otimes |00\rangle)\big)^{\otimes 3}$

et $C = \{\mathcal{V}(|\psi\rangle \otimes |0_8) \mid |\psi\rangle \in \mathcal{H}, \||\psi\rangle\| = 1\}$
$= \{a|0_L\rangle + b|1_L\rangle \,|\, |a|^2 + |b|^2 = 1\}$

Représentation sous forme de circuit :
figure 3

Décodage

1. Mesure :

On utilise la décomposition de $\mathcal{H}^{\otimes 9}$ en 2^8 sous espaces orthogonaux de dimension 2 sur $\mathbb{C}$:

$$\mathcal{H}^9 = \bigoplus_{0 \le i \le 255}^{\perp} E_i$$

où E_0 est l'espace du code lui-même.

$$E_0 = Vect\Big(\frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}, \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}\Big)$$

Pour des facilités de lisibilité on peut adopter les notations suivantes :
$E_i = E_{\alpha_1\alpha_2,\alpha_3\alpha_4,\alpha_5\alpha_6,\sigma_1\sigma_2}$ où $\forall j \in \{1,...,6\}, \alpha_j \in \{0,1\}$ et $\sigma_1, \sigma_2 \in \{+,-\}$
Par exemple :

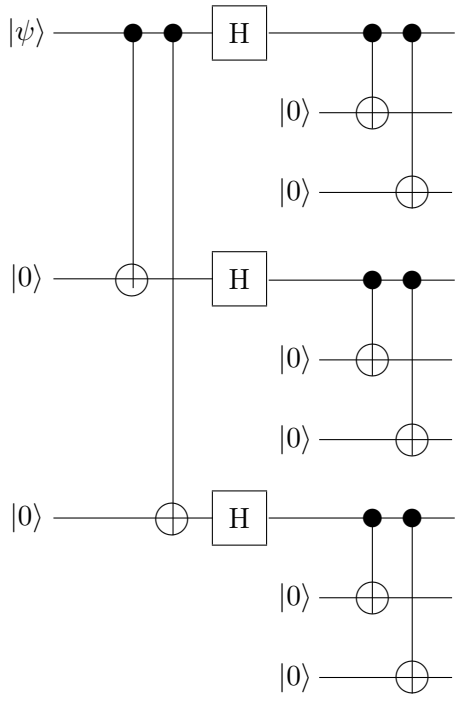


FIG. 4 – Circuit d'encodage du code de Shor

1.

$$E_{10,00,00,++} = Vect\Big(\frac{(|100\rangle + |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}},$$

$$\frac{(|100\rangle - |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}\Big)$$

Et à $E_{10,00,00,++}$, on associe l'erreur $XIIIIIIII$.

2.

$$E_{00,00,00,-+} = Vect\Big(\frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}},$$

$$\frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}\Big)$$

Et à $E_{00,00,00,-+}$, on associe l'erreur $IIIIIIZII$.

On établit ainsi une correspondance pour tout i entre E_i et une erreur de Pauli :

- $\alpha_1\alpha_2$ détermine une erreur de bit-flip sur un des trois premiers qubits (qubits 1 à 3).
- $\alpha_3\alpha_4$ détermine une erreur de bit-flip sur un des trois qubits suivants (qubits 4 à 6).
- $\alpha_3\alpha_4$ détermine une erreur de bit-flip sur un des trois qubits suivants (qubits 7 à 9).
- $\sigma_1\sigma_2$ détermine une erreur de phase-flip sur n'importe quel qubit.
  En effet, par exemple l'action d'un seul phase-flip sur n'importe lequel des trois

qubits aura le même effet sur le registre. En d'autres termes, on ne distingue pas l'action de $ZIIIIIIII$ de celle de $IZIIIIIII$ ou de $IIZIIIIII$, et un registre ayant subit uniquement l'une de ces trois erreur sera projeté dans le même sous espace. Il en est de même pour les trois qubits suivants, et pour les trois derniers qubits. La donnée de $\sigma_1\sigma_2$ suffit donc à determiner une erreur de phase-flip sur n'importe quel qubit.

Ce code permet ainsi de corriger systématiquement une erreur arbitraire (un phase-flip ou un bit-flip ou les deux). En réalité, on peut corriger souvent plus de une erreur. C'est le cas par exemple lorsque que deux erreurs de bit-flip arrivent, l'une sur un des trois premiers qubits (qubit 1, 2 ou 3), et l'autre sur un des trois qubits suivants (4, 5 ou 6). Et il en est de même pour toute autre configuration de ce type.

2. Correction :
La correction se fait sans surprise en appliquant X ou Z sur le qubit concerné, ou encore Y en cas de deux erreurs, phase-flip et bit-flip, sur le même qubit.

Remarque :
Notons :
$M_i = I \otimes ... \otimes I \otimes M \otimes I \otimes ... \otimes I \in \mathcal{G}_n$.
(ième terme)

Cette notation pourra être reprise par la suite.

Certains erreurs laissent le code inchangé. Par exemple deux erreurs de phase-flip dans une même close laisse l'état quantique inchangé : $Z_1Z_2|\psi\rangle = Z_2Z_3|\psi\rangle = Z_4Z_5|\psi\rangle = Z_5Z_6|\psi\rangle = Z_7Z_8|\psi\rangle = Z_8Z_9|\psi\rangle = |\psi\rangle$
De même si l'on fait agir 3 bit-flip sur les trois qubits d'une close dans 2 des trois closes, le code restera là aussi inchangé. En fait, on peut montrer (ce qui se vérifie immédiatement) que :
$\forall\, M \in \mathcal{S} := \langle Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9, X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9\rangle$,
et $\forall\, |\varphi\rangle \in C$, $M|\varphi\rangle = |\varphi\rangle$.
On dit que le code de Shor est un **code stabilisateur** associé au groupe $\mathcal{S}$.

2.3 Les codes stabilisateurs

Les codes stabilisateurs sont une sous famille très riche des codes correcteurs quantiques. Ils possèdent cette qualité remarquable d'être entièrement décrit par un groupe discret : le **groupe stabilisateur** du code. C'est un sous groupe du **goupe de Pauli** :

Définition 6. *Soit S un sous groupe commutatif de $\mathcal{G}_n$ tel que $-1_{\mathcal{G}_n} \notin S$.*
*C est un **code stabilisateur** associé à l'ensemble S si*
$C = \{|\psi\rangle \in \mathcal{H}^{\otimes n} \quad | \quad M|\psi\rangle = |\psi\rangle \quad \forall M \in S\}$.
*Les éléments de S sont les **stabilisateurs** du code.*

Lorsqu'on parlera de groupe stabilisateur S (associé à ou associant un code C), on supposera désormais que S vérifie les conditions : S est un sous groupe de $\mathcal{G}_n$ commutatif, tel que $-1_{\mathcal{G}_n} \notin S$.

Les codes stabilisateurs sont aux codes quantiques ce que les codes linéaires sont aux codes classiques. Ils partagent en effet les mêmes notions de matrice de parité, syndrôme et paramètres : longueur, dimension et distance minimale.

Proposition 2. *Soit C un code stabilisateur associé à un groupe stabilisateur S généré par $n-k$ stabilisateurs mutuellement indépendants, où $k < n$.*
Alors $\boxed{\dim(C) = 2^k}$.

Pour démontrer cette proposition, nous allons d'abord énoncer quelques propriétés du groupe de Pauli.

Propriété 3. *Pour tout $A \in \mathcal{G}_n \neq I$, il existe B et $C \in \mathcal{G}_n$ tels que B commute avec A et C anticommute avec A.*

Démonstration : Soit $A \in \mathcal{G} \neq I$
1. L'identité commute avec A.
2. $A \neq I$ donc il existe $P \in \{X, Y, Z\}$ tel que $A_i = \varepsilon P$, ($\varepsilon \in \{\pm 1, \pm i\}$). Posons $B = B_i = Q$ où $Q \in \{X, Y, Z\} \setminus P$. A et B anticommutent. $\diamond$

Propriété 4. *Pour toute famille $(A_1, \ldots, A_m) \in {\mathcal{G}_n}^m$ d'éléments mutuellements indépendants, commutatifs, et tel que $-1_{\mathcal{G}_n} \notin \langle A_1, \ldots, A_n \rangle$, il existe $B \in \mathcal{G}_n$ et $j \in \{1, \ldots, m\}$, tels que B anticommute avec A_j et B commute avec A_i $\forall i \neq j$.*

Démonstration succincte :
Soit $A_j \in \{A_1, \ldots, A_m\}$. Par l'indépendance des A_i, on déduit qu'il existe t tel que le $t$ième de A_j $(A_j)_t \neq I$. On pose $B = P \in \{X, Y, Z\} \backslash (A_j)_t$. Ensuite, si il existe $i \neq j$ tel que B et A_i anticommute, il existe k tel que $(B)_k$ et $(A_i)_k$ anticommute et tel que $k \neq t$ car A_j et A_i commutent.
On pose $(B)_k = (A_j)_t$.
En répétant le procédé si nécéssaire, de par les propiétés de $\langle A_i \rangle$, on obtient un élément B voulu.$\diamond$

Propriété 5. *Soit C le code associé au groupe stabilisateur $\mathcal{S} := \langle \mathscr{S}_1 \rangle$, $\mathscr{S}_1 \in \mathcal{G}_n$, $\mathscr{S}_1 \neq I$. Alors $\mathcal{H}^{\otimes n} = C \overset{\perp}{\oplus} NC$, où $N \in \mathcal{G}_n$ anticommute avec $\mathscr{S}_1$.*

Démonstration :
$-1 \notin \mathcal{S}$ donc $i \notin \mathcal{S}$ (résonnons par l'absurde, si i appartenait à $\mathcal{S}$, alors -1 appartiendrait à $\mathcal{S}$). Donc les éléments de $\mathcal{S}$ ne sont que des produits tensoriels de I,X,Y,Z, et donc tous unitaires et hermitiens.
$\mathscr{S}_1$ est hermitienne et unitaire donc $\mathscr{S}_1$ est diagonalisable dans une certaine base dans laquelle ses valeurs propres sont toutes -1 et 1. Donc
$\forall\ |\varphi\rangle \in \mathcal{H}^{\otimes n}$, $|\varphi\rangle = |\varphi_0\rangle + |\varphi_1\rangle$ où $\begin{cases} \mathscr{S}_1|\varphi_0\rangle = |\varphi_0\rangle \\ \mathscr{S}_1|\varphi_1\rangle = -|\varphi_1\rangle \end{cases}$
Donc $\mathcal{H}^{\otimes n} = C + F$ où $F := \{|\varphi\rangle \mid \mathscr{S}_1|\varphi\rangle = -|\varphi\}$

De plus, $\mathscr{S}_1 \neq I$, donc d'après la propriété 3, il existe $N \in \mathcal{G}_n$ qui anticommute avec $\mathscr{S}_1$.
Montrons que $F = NC$:
Soit $|\varphi\rangle \in F$. Posons $|\psi\rangle = N^{-1}|\varphi\rangle$. Alors $\mathcal{S}_1|\psi\rangle = \mathcal{S}_1 N^{-1}|\varphi\rangle = -N^{-1}\mathcal{S}_1|\varphi\rangle$ car N^{-1} et $\mathcal{S}_1$ anticommutent.
d'où $\mathcal{S}_1|\psi\rangle = +N^{-1}|\varphi\rangle = |\psi\rangle$.
et $|\varphi\rangle = N|\psi\rangle$, $|\psi\rangle \in C$ donc $F \subset NC$.
Réciproquement si $|\varphi\rangle \in C$, alors $\mathcal{S}_1 N|\varphi\rangle = -N|\varphi\rangle$ donc $N|\varphi\rangle \in F$ et $NC \subset F$.
Donc $F = NC$ et $\mathcal{H}^{\otimes n} = C + NC$.
De plus,

$$|\varphi\rangle \in C \cap NC \Leftrightarrow (\mathscr{S}_1|\varphi\rangle = |\varphi\rangle et \mathscr{S}_1|\varphi\rangle = -|\varphi\rangle) \Leftrightarrow |\varphi\rangle = 0$$

Enfin

$$\begin{aligned} \forall\ |\psi\rangle, |\varphi\rangle \in C, \langle|\varphi\rangle, N|\psi\rangle\rangle \quad &:= \langle\varphi|N|\psi\rangle \\ &= \langle\varphi|\overline{\mathscr{S}_1}^{\,t} N|\psi\rangle \\ &= \langle\varphi|\mathscr{S}_1 N|\psi\rangle (\mathscr{S}_1 \text{ hermitienne }) \\ &= -\langle\varphi|N\mathscr{S}_1|\psi\rangle \\ &= -\langle\varphi|N|\psi\rangle \end{aligned}$$

Donc $\langle|\varphi\rangle, N|\psi\rangle\rangle = 0$.
Finalement : $\mathcal{H}^{\otimes n} = C \overset{\perp}{\oplus} NC$.
Par ailleurs, N est unitaire (donc inversible), donc $\dim(C) = \dim(NC)$
soit $\dim(C) = \dim(\mathcal{H}^{\otimes n})/2 = 2^n/2 = 2^{n-1}$. $\diamond$

Nous sommes maintenant prêt à démontrer la proposition 2, que l'on peut réenoncer de la manière suivante :

Proposition 3. $\mathcal{H}^{\otimes n} = \overset{\perp}{\bigoplus_{s\in\{0,1\}^{n-k}}} \mathcal{E}_s C$ *et* $\dim(C) = 2^k$.

Démonstration :
Notons $r = n - k$ le nombre de générateur de S et montrons par récurrence sur r que :

$\forall r \in \{0, \ldots, n\}, \mathcal{H}^{\otimes n} = \overset{\perp}{\bigoplus_{s\in\{0,1\}^r}} \mathcal{E}_s C$, et C vérifie $\dim(C) = 2^{n-r}$.

- $\underline{r = 0}$: Alors $S = \{I\}$ et $C = \mathcal{H}^{\otimes n}$.
- $\underline{r = 1}$: C'est la propriété 5.
- Supposons que pour un certain $r \in \{1, \ldots, n-1\}$, on ait
  $S = \langle\mathscr{S}_1, \ldots, \mathscr{S}_r\rangle$, $\mathcal{H}^{\otimes n} = \overset{\perp}{\bigoplus_{s\in\{0,1\}^r}} \mathcal{E}_s C$ et $\dim(C) = 2^{n-r}$.
  Soit $\mathscr{S}_{r+1} \in \mathcal{G}_n \setminus (\mathcal{S} \times \{\pm I, \pm iI\})$ tel que $\mathscr{S}_{r+1}$ commute avec tous les éléments de S.
  Et soit $\mathcal{S}' := Vect(\mathscr{S}_1, \ldots, \mathscr{S}_r, \mathscr{S}_{r+1})$.
  $\mathscr{S}_{r+1} \neq I$ et est indépendant de $\mathscr{S}_j$ $\forall j \in \{1, \ldots, r\}$ donc il existe $N \in \mathcal{G}_n$ tel que N anticommute avec $\mathscr{S}_{r+1}$ et commute avec tous les autres. (propriété 4).

De plus, C est stable par $\mathscr{S}_{r+1}$. En effet,
$\forall\ |\psi\rangle \in C,\ \forall\ j \in \{1,\dots,r\},\quad \mathscr{S}_j\mathscr{S}_{r+1}|\psi\rangle = \mathscr{S}_{r+1}\mathscr{S}_j|\psi\rangle = \mathscr{S}_{r+1}|\psi\rangle.$
Donc $\mathscr{S}_{r+1}C = C$ (De même $\mathscr{S}_{r+1}\mathcal{E}_sC = \mathcal{E}_sC$)
On se place alors dans C et on note D le code stabilisateur associé à $\langle\mathscr{S}_{r+1}\rangle$ dans C.
On a, d'après la propriété 5, $C = D \overset{\perp}{\oplus} ND$ et $\dim(D) = \dim(C)/2 = 2^{n-(r+1)}$
De plus, dans $\mathcal{H}^{\otimes n}$, D est exactement stabilisé par $\mathscr{S}_j$, $\forall j \in \{1,\dots,r\}$ et par $\mathscr{S}_{r+1}$, donc D est le code stabilisateur associé à S'.
On a montré que $\mathcal{H}^{\otimes n} = \overset{\perp}{\bigoplus_{s\in\{0,1\}^r}} \mathcal{E}_s(D \overset{\perp}{\oplus} ND) = \overset{\perp}{\bigoplus_{s\in\{0,1\}^r}} (\mathcal{E}_sD \overset{\perp}{\oplus} \mathcal{E}_sND)$
Par ailleurs, on a toujours : $\mathscr{S}_{r+1}$ et $\mathcal{E}_s$ soit commutent, soit anticommutent.
Ainsi, si $\mathscr{S}_{r+1}$ et $\mathcal{E}_s$ commutent, alors $\mathscr{S}_{r+1}$ et $N\mathcal{E}_s$ anticommutent. On pose alors $\mathcal{E}_{(s,0)} := \mathcal{E}_s$ et $\mathcal{E}_{(s,1)} := N\mathcal{E}_s$.
Sinon, si $\mathscr{S}_{r+1}$ et $\mathcal{E}_s$ anticommutent, alors $\mathscr{S}_{r+1}$ et $N\mathcal{E}_s$ commutent. On pose alors $\mathcal{E}_{(s,0)} := N\mathcal{E}_s$ et $\mathcal{E}_{(s,1)} := \mathcal{E}_s$.
(On peut montrer que en fait, $\mathcal{E}_s$ et $\mathscr{S}_{r+1}$ commutent.)
Finalement, on obtient $\mathcal{H}^{\otimes n} = \overset{\perp}{\bigoplus_{s\in\{0,1\}^{r+1}}} \mathcal{E}_sD$,et $\dim(D) = 2^{n-(r+1)}$
(Remarque : $\mathcal{E}_0D = D$, prendre par exemple $\mathcal{E}_0 = I$.)
Conclusion : la propriété est vraie au rang 0 et est héréditaire donc elle est vraie pour tout $r \in \{1\dots,n\}$. $\diamond$

L'orthogonalité des sous espaces de $\mathcal{H}^{\otimes n}$ est donc entièrement déterminée par les relations de commutativité entre les éléments de $\mathcal{G}_n$. (Il est équivalent de décomposer $\mathcal{H}^{\otimes n}$ en somme directe de sous espaces orthogonaux que de décomposer $\mathcal{G}_n$ en sous groupes commutatifs et anti commutatifs à partir de S). Nous allons donc définir une loi $\star$ qui caractérise la relation d'être ou non commutatif entre deux erreurs :

Définition 7. *Soient $A, B \in \mathcal{G}_1$.*
- *On définit une loi $\star$ sur $\mathcal{G}_1$ par :*
  $A \star B := \begin{cases} 0 \text{ si } A \text{ et } B \text{ commutent.} \\ 1 \text{ sinon.} \end{cases}$
- *Soient $A, B \in \mathcal{G}_n$. Notons $A = \bigotimes_{1\le i\le n} A_i$, $B = \bigotimes_{1\le i\le n} B_i$.*
  $A \star B := \sum_{1\le i\le n} A_i \star B_i \bmod 2$

De plus la loi $\star$ dans $\mathcal{G}_n = \{I, X, Y, Z\}^{\otimes n} \times \{\pm I, \pm iI\}$ ne dépend pas des coefficients $\{\pm 1, \pm i\}$. C'est pourquoi on préfère se placer dans le groupe quotient :

Définition 8. *On nomme **groupe effectif**, et on note G_1, le groupe quotient*

$$G_1 = \frac{\mathcal{G}_1}{Z(\mathcal{G}_1)} = \frac{\mathcal{G}_1}{\{\pm I, \pm iI\}}$$

*($Z(\mathcal{G}_1)$ est le **centre** de $\mathcal{G}_1$)*

Proposition 4. *G_1 est isomorphe à $((Z/2Z \times Z/2Z), +)$*

Démonstration : G_1 est un groupe d'ordre 4, commmutatif, dont tous les éléments sont d'ordre 2. $\diamond$
On note additivement la loi de groupe sur G_1.

Corollaire 5. $G_n \simeq (Z/2Z \times Z/2Z)^n$

Dans G_1, on identifiera :
$$\begin{cases} \text{la classe de } I : \hat{I} & \text{à } (0,0) \\ \text{la classe de } X : \hat{X} & \text{à } (1,0) \\ \text{la classe de } Z : \hat{Z} & \text{à } (0,1) \\ \text{ et la classe de } Y : \hat{Y} & \text{à } (1,1) \end{cases}$$

Propriété 6. *Soient* $A, B \in \mathcal{G}_1$
$$A \star B = \begin{cases} 0 \text{ si } \hat{A} = \hat{B} \text{ ou } \hat{A} = \hat{I} \text{ ou } \hat{B} = \hat{I} \\ 1 \text{ sinon} \end{cases}$$

Il pourra aussi arriver que pour simplifier les choses, on utilise la loi $\star$ dans G_n, puisque l'application définie par $\hat{\mathcal{E}} \star \hat{\mathcal{F}} = \mathcal{E} \star \mathcal{F}$ est bien définie. Toutefois il faudra alors prendre des précautions de langage car elle ne représente plus une relation de commutativité dans G_n.

Syndrôme

Définition 9. *Soit* $S = \langle \mathscr{S}_i, 1 \leq i \leq n-k \rangle$*, un groupe stabilisateur généré par* $n-k$ *stabilisateurs* $\mathscr{S}_i$ *mutuellements indépendants.*
On définit le ***syndrôme*** σ *d'une erreur* $\mathcal{E} \in \mathcal{G}_n$ *par :*
$\sigma(\mathcal{E}) = (\mathcal{E} \star \mathscr{S}_i)_{1 \leq i \leq n-k}$.
Notation : *lorsqu'il n'y a pas d'ambiguité, on notera* $\mathcal{E}_s$ *un élément de* $\mathcal{G}_n$ *vérifiant* $\sigma(\mathcal{E}_s) = s$.

Dans la suite, sauf mention du contraire, on travaille dans le groupe effectif. On notera I, X, Z, Y les représentants des classes $\hat{I}, \hat{X}, \hat{Z}, \hat{Y} \in G_1$.

Matrice de parité

Pour présenter la matrice de parité d'un code stabilisateur, il convient d'introduire tout d'abord deux applications linéaires, notées P_X et P_Z, définies par :

$$P_X : \begin{array}{c} G_1 \to G_1 \\ \begin{cases} X \mapsto X \\ Z \mapsto I \end{cases} \end{array} \qquad P_Z : \begin{array}{c} G_1 \to G_1 \\ \begin{cases} X \mapsto I \\ Z \mapsto Z \end{cases} \end{array}$$

P_X et P_Z étant linéaires, on en déduit que $P_X(Y) = X$ et $P_Z(Y) = Z$.
De plus, pour $U \in \{X, Z\}$, on définit P_U sur G_n par :

$$\begin{array}{rccc} P_U : & G_n & \longrightarrow & G_n \\ & (E_0 \dots E_{n-1}) & \longmapsto & (P_U(E_0)) \dots P_U(E_{n-1})) \end{array}$$

Notons par ailleurs que la condition $-1_{\mathcal{G}_n} \notin S$ implique que le groupe stabilisateur est un sous groupe du groupe effectif.

Définition 10. *On définit la* ***matrice de parité*** *d'un code linéaire par la matrice dont les lignes sont les $n-k$ stabilisateurs de S, vus comme des éléments de $\mathbb{F}_2{}^n \times \mathbb{F}_2{}^n$*

Formulation explicite :
$S = \langle S_1, \ldots, S_{n-k} \rangle$, où $\forall i$, les S_i appartiennent à G_n et sont mutuellement indépendants et commutatifs.

$$H = \left[\begin{array}{c|c} P_X(S_1) & P_Z(S_1) \\ \vdots & \vdots \\ P_X(S_{n-k}) & P_Z(S_{n-k}) \end{array}\right]$$

On écrira indifferement les lignes de H comme des éléments de $\{I, X, Z, Y\}^n$ ou de $\mathbb{F}_2{}^{2n}$.
Exemple :
Si $\mathcal{S} = \langle YYI, IYY \rangle$, on pourra écrire

$$H = \left[\begin{array}{cc|cc} X & X & I & Z & Z & I \\ I & X & X & I & Z & Z \end{array}\right] = \left[\begin{array}{ccc} Y & Y & I \\ I & Y & Y \end{array}\right]$$

On pourra également noter

$$\mathcal{S} \overset{\text{Not}}{=} \left|\begin{array}{cc|cc} X & X & I & Z & Z & I \\ I & X & X & I & Z & Z \end{array}\right| = \left|\begin{array}{ccc} Y & Y & I \\ I & Y & Y \end{array}\right|$$

On peut vérifier alors que $\forall \mathcal{E} \in \mathcal{G}_n$, $HE = \sigma(\mathcal{E})$, où $E = \hat{\mathcal{E}} \in G_n$.
On redéfinit ainsi la loi $\star$ dans $\mathbb{F}_2{}^{2n}$ par

$E * F = (P_X(E)|P_Z(E).(P_Z(F)|P_X(F))$ où . est le produit scalaire usuel sur $\mathbb{F}_2{}^{2n}$.

Ainsi si on prend S comme un groupe commutatif d'ordre $n-k$, tous les éléments de $\mathcal{G}_n$ qui commutent avec tous les éléments de S forment le **normalisateur** de S (car $\mathcal{S}$ est commutatif) que l'on note $N(S)$. Si on prends S comme un espace vectoriel de dimension $n-k$ sur F_2, on peut (légitimement) définir $N(S)$ comme l'orthogonal de S pour la loi $\star$. Si il n'y a pas de confusion possible, on notera $S^\perp$ l'orthogonal de S pour la loi $\star$.

Propriété 7.

1. $\dim(S^\perp) = n + k$
2. $\dim(\frac{S^\perp}{S}) = 2k$

Démonstration :

$$S^\perp = \{E \in G_n \mid E * S_i = 0,\ \forall i, 1 \leq i \leq n\} = \{x \in F_2^{2n} \mid Hx = 0\} = Ker(H)$$

or $\dim(Ker(H)) + rg(H)) = \dim(\mathbb{Z}/2\mathbb{Z}^{2n})$
donc $\dim(Ker(H)) = 2n - (n-k) = n + k$ et $\dim(S^\perp) = n + k$
De plus, comme les éléments de S sont mutuellement commutatif, $S \subset S^\perp$.
Et $\dim(\frac{S^\perp}{S}) = n + k - (n-k) = 2k$. $\diamond$

Proposition 6. *$S^\perp \setminus S$ constitue un ensemble d'erreur non-détectable.*

Démonstration :
C'est évident. Les erreurs de $S^\perp \setminus S$ commutent avec tous les éléments du stabilisateur, et sont donc de syndrôme nul. Mais elles n'appartiennent pas au stabilisateur, et ne laissent donc pas le registre codé inchangé.
On dit qu'elles constituent un ensemble d'erreur **effectives**, contrairement aux erreurs qui constituent le stabilisateur, qui n'ont aucun effet sur le registre. (on peut dire qu'elle constituent un ensemble d'erreurs **inoffensives** ou **neutres**).

Matrice « génératrice »
Pour parler des matrices d'encodage d'un code, nous nous référons à [5], et nous allons introduire une nouvelle transformation.

Définition 11. *Une **transformation de Clifford** est une matrice unitaire qui laisse le groupe de Pauli globalement invariant par conjuguaison.*

Soit $\mathscr{V}$ une matrice de Clifford de taille $2^n \times 2^n$.
Posons de plus $\forall i \in \{1, \dots, n\}$, $\overline{Z}_i = \mathscr{V} Z_i \overline{\mathscr{V}}^t$.
Alors

Proposition 7. *Le code défini par $C = \{\mathscr{V}(|\varphi\rangle \otimes |0_{n-k}\rangle) \mid |\varphi\rangle \in \mathcal{H}^{\otimes k}, \||\varphi\rangle\| = 1\}$ est un code stabilisateur, associé à $\langle \overline{Z}_{k+i} \rangle_{1 \leq i \leq n-k}$.*

Démonstration :
Les transformations de Clifford conservent la loi $\star$ sur $\mathcal{G}_n$,
donc $\langle \overline{Z}_{k+i} \rangle_{1 \leq i \leq n-k}$ est un groupe commutatif.
De plus $\mathscr{V}$ est inversible, donc, en tant qu'espace vectoriel sur $\mathbb{F}_2$,
d'une part $\dim(\langle \overline{Z}_{k+i} \rangle_{1 \leq i \leq n-k}) = \dim(\langle Z_{k+i} \rangle_{1 \leq i \leq n-k}) = n - k$
d'autre part $\{\overline{Z}_{k+i}, 1 \leq i \leq n-k\}$ est une famille libre de $\mathbb{F}_2^{2n}$.
Donc $\langle \overline{Z}_{k+i} \rangle_{1 \leq i \leq n-k}$ est un groupe stabilisateur.
De plus,

$$\forall |\tilde{\varphi}\rangle \in C,\ \overline{Z}_{k+i}|\tilde{\varphi}\rangle = \mathscr{V} Z_i \overline{\mathscr{V}}^t |\tilde{\varphi}\rangle = \mathscr{V} Z_{k+i} \overline{\mathscr{V}}^t \mathscr{V}(|\varphi\rangle \otimes |0_{n-k}\rangle) = \mathscr{V}(|\varphi\rangle \otimes Z_i |0_{n-k}\rangle)$$
$$= \mathscr{V}(|\varphi\rangle \otimes |0_{n-k}\rangle) = |\tilde{\varphi}\rangle$$

Donc $\langle \overline{Z}_{k+i} \rangle_{1 \leq i \leq n-k}$ est un groupe stabilisateur associé à C. (Et C est un code stabilisateur.)
$\diamond$

En gardant les mêmes notations, posons également
$\forall i \in \{1, \dots, n\}$, $\overline{X}_i = \mathscr{V} X_i \overline{\mathscr{V}}^t$.
Alors :

Proposition 8. *Les classes d'équivalence de $\{\overline{X}_i, \overline{Z}_j,\ i, j \in \{1, \dots, k\}\}$ forment une base de $\frac{S^\perp}{S}$ On les nomme **opérateurs logiques** du code.*

Démonstration (Idée) :
Les transformations de Clifford conservent la loi $\star$ sur $\mathcal{G}_n$, donc

– d'une part $\forall i \in \{1,\dots,k\}$, $\forall j \in \{k+1,\dots,n\}$, $\overline{X}_i \star \overline{Z}_j = 0$ et $\overline{Z}_i \star \overline{Z}_j = 0$ donc

$$\langle \overline{X}_i, \overline{Z}_j \rangle_{i,j\in\{1,\dots,k\}} \subset S^{\perp} \tag{5}$$

– d'autre part $\forall i \in \{1,\dots,k\}$,

$$\overline{X}_i \star \overline{Z}_j = \delta_{ij} \tag{6}$$

D'après (5), on peut définir notre relation d'équivalence sur $\langle \overline{X}_i, \overline{Z}_j \rangle_{i,j\in\{1,\dots,k\}}$. Maintenant, supposons qu'il existe P et $Q \in \langle \overline{X}_i, \overline{Z}_j \rangle_{i,j\in\{1,\dots,k\}} \backslash I_{G_n}$ tel que P et Q soit dans la même classe d'équivalence. Alors $P - Q \in \mathcal{S}$.
Mais d'après (6), il existe $R \in \langle \overline{X}_i, \overline{Z}_j \rangle_{i,j\in\{1,\dots,k\}}$ tel que $R \star P = 1$ et $R \star Q = 0$, donc $R \star (P - Q) = 1$, ce qui est en contradiction avec $P - Q \in \mathcal{S}$.
Donc chaque élément de $\langle \overline{X}_i, \overline{Z}_j \rangle_{i,j\in\{1,\dots,k\}}$ représente une classe d'équivalence distincte.
On termine en disant que $\langle \overline{X}_i, \overline{Z}_j \rangle_{i,j\in\{1,\dots,k\}}$ est (en tant qu'espace vectoriel sur $\mathbb{F}_2$), de dimension $2k$, (c'est l'image de $\langle X_i, Z_j \rangle_{i,j\in\{1,\dots,k\}}$ par une transformation inversible),
donc $\dim(\langle \overline{X}_i, \overline{Z}_j \rangle_{i,j\in\{1,\dots,k\}}) = 2k = \dim(\frac{S^{\perp}}{S})$
d'où la proposition. $\diamond$

Syndrôme et $\overline{X}_i$:
Remarque (#) :

$\sigma(\overline{X}_i) = e_i$ où $e_i = (0,\dots,\underset{\substack{\uparrow \\ i\text{ème terme}}}{1},\dots,0)$

En effet, pour $i, j \in \{1,\dots,n-k\}$, $X_{k+i} \star Z_{k+j} = \delta_{ij}$, donc $\overline{X}_{k+i} \star \overline{Z}_{k+j} = \delta_{ij}$.

Proposition 9. *Les classes d'équivalence de* $\{\overline{X}_{k+i},\ i \in \{1,\dots,n-k\}\}$ *forment une base de* $\dfrac{G_n}{S^{\perp}}$.

Démonstration (Idée) :
Supposons P et $Q \in \langle \overline{X}_{k+i} \rangle_{i\in\{1,\dots,n-k\}}$ distincts et dans la même classe d'équivalence.
Alors on aurait $P - Q \in S^{\perp}$, donc $\sigma(P - Q) = 0$ et d'après la remarque (#), cela entraine $P - Q = I$ soit $P = Q$: Contradiction
Donc chaque élément de $\langle \overline{X}_{k+i} \rangle_{i\in\{1,\dots,n-k\}}$ représente une classe d' équivalence distincte. Enfin on vérifie qu'en tant qu'espace vectoriel sur $\mathbb{F}_2$, $\dim(\langle \overline{X}_{k+i} \rangle_{i\in\{1,\dots,n-k\}}) = n - k = \dim(\frac{G_n}{S^{\perp}})$.
ce qui termine la démonstration
$\diamond$

On peut ainsi exhiber une base orthonormale de $n-k$ vecteurs $(\overline{X}_{k+1},\dots,\overline{X}_n)$ de $\frac{G_n}{S^{\perp}}$ adaptée au syndrôme.
Dans la suite du rapport, sauf mention contraire, on $\overline{X}_i, \overline{Z}_i$ est défini comme dans cette partie.

Définition 12.

1. *On définit le* ***poids de Pauli*** ω *d'une erreur* $\mathcal{E} = \underset{i\in\{1,\dots,n\}}{\otimes} \mathcal{E}_i \in G_n$, *par* $\omega(\mathcal{E}) = Card(\mathcal{E}_i \mid \mathcal{E}_i \neq I, 1 \leq i \leq n)$
2. *On définit la* ***distance minimale****, notée d, d'un code stabilisateur par le poids minimum d'une erreur* $\in S^{\perp} \setminus S$.
3. *Le code stabilisateur est de paramètre [n,k,d].*

Remarque :

1. Comme pour les codes linéaires, la distance minimale est le plus petit poids d'une erreur *effective* de syndrôme nul.
2. Les paramètres d'un code stabilisateur caractérisent sa longueur, sa dimension, et sa distance minimale. Toutefois rappelons que pour un codage stabilisateur U :

$$U : \begin{array}{ccc} \mathcal{H}^{\otimes k} \otimes \mathcal{H}^{\otimes(n-k)} & \rightarrow & \mathcal{H}^{\otimes n} \\ |\varphi\rangle \otimes |\psi_0\rangle & \mapsto & U(|\varphi\rangle \otimes |\psi_0\rangle), \quad |\psi_0\rangle \text{ registre cible.} \end{array}$$

   - on code effectivement une chaine de n qubits (mais sur une superposition de 2^n états).
   - la dimension du code au sens propre est 2^k.

Code dégénéré.
On dit qu'un code est dégénéré s'il existe (au moins) deux erreurs distinctes qui se corrigent de la même façon.
Par exemple, le code de Shor à 9 qubits est dégénéré. Nous avons vu en effet (cf section 4.2) qu'il existe des erreurs corrigeables, distinctes, et qui se corrigent pourtant de la même façon, par exemple $ZIIIIIIII$, $IZIIIIIII$ et $IIZIIIIII$ sont trois erreurs qui peuvent se corriger avec $ZIIIIIIII$.

3 La capacité d'un canal de dépolarisation

Au sens courant, la capacité d'un support quelconque désigne la contenance maximale de ce support. On peut alors penser la capacité d'un canal comme le volume maximal d'*information* que peut supporter ce canal. La **théorie de l'information** repose fondamentalement sur le besoin de définir et de mesurer cette information, afin de répondre dans un langage mathématique à deux questions simples mais essentielles : quelle *compression* minimale des données, et quelle *redondance* optimale (la plus petite possible pour le plus grand rendement) de code pouvons-nous espérer ? De nos jours, on utilise principalement le bit comme unité de mesure, puisqu'il est devenu l'unité de transport universel de tout codage d'information moderne (numérique). Un ensemble d'information constitue un message. Sans codage préalable, mesurer la longueur d'un message suffirait pour mesurer l'information qu'il contient. Mais la redondance qu'implique le codage, comme la compression possible qui existe, nous oblige à aller chercher une notion plus subtile pour rendre compte du taux d'information exact que contient le message. Celle ci est issue de la thermodynamique. Il s'agit de l'entropie, introduite par Rudolf Clausius en 1865, fonction d'état qui sert dans la mesure des échanges thermiques. Les concepts finalement plus généraux d'échange d'informations se sont appropriés cette notion par l'intermédiaire de Claude Shannon, qui, en particulier, en a dérivé quelques définitions pour parvenir à une théorie cohérente de mesure de l'information.

3.1 Définitions

Définition 13. *L'**entropie** $H(X)$ d'une variable aléatoire discrète X à valeur dans $\mathscr{X}$ et suivant une loi de probabilité $p_X(.)$ est définie par*

$$H(X) = -\sum_{x\in\mathscr{X}} p(X=x)\log p(X=x)$$

L'entropie de X représente intuitivement le nombre (de bits) d'information sur X. (C'est un calcul d'espérance sur la longueur nécessaire pour décrire n'importe quelle valeur de X).
Remarque : Bien que mesurer en bit, l'entropie pourra prendre toutes valeurs réelles positives. En tant qu'unité de mesure, le bit n'est plus vu comme un simple entier pouvant prendre deux valeurs 0 ou 1 mais comme la longueur de ce bit, et par conséquent comme un nombre réel que l'on peut diviser (exemple de mesure : $0,3$ bit d'information, $0,000001$ bit d'information ...).

Définition 14. *Si le couple (X,Y) suit une loi $p(x,y)$, alors l'**entropie conditionnelle** $H(X|Y)$ est définie par*

$$\begin{aligned} H(X|Y) &= \sum_{y\in\mathscr{Y}} p(y)H(X|Y=y) \\ &= -\sum_{y\in\mathscr{Y}} p(y)\sum_{x\in\mathscr{X}} p(x|y)\log(p(x|y)) \\ &= -\sum_{y\in\mathscr{Y}}\sum_{x\in\mathcal{X}} p(y,x)\log(p(x|y)) \\ &= -E_{p(y,x)}\log(p(X|Y)) \end{aligned}$$

L'entropie conditionnelle de X sachant Y est en quelque sorte la mesure de l'information sur X qui n'est pas contenue dans Y. On peut la voir comme la distance, à $P(X)$ fixé, séparant l'information de X de celle de Y.

Définition 15. *Etant données deux variables aléatoires X et Y suivant une loi conjointe $p_{(X,Y)}(.,.)$ et des lois marginales respectives $p_X(.)$ et $p_Y(.)$. L'**information mutuelle** $I(X,Y)$ est définie par*

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Définition 16. *Etant données deux variables aléatoires discrètes X et Y suivant une loi conjointe $p_{(X,Y)}(.,.)$ et des lois marginales respectives $p_X(.)$ et $p_Y(.)$, et étant donné un canal de transmission d'entrée X et de sortie Y, on définit la **capacité** de ce canal par*

$$Q = \max_{p_X(.)}(I(X,Y))$$

$max(I(X,Y))$ étant le maximum d'information commune à X et Y, dans un canal où X est la variable d'entrée, et Y la variable de sortie, $max(I(X,Y))$ est le maximum d'information transmise (dans l'idéal, si on décode au maximum (de vraissemblance)). De plus, notons qu'un canal est défini par les probabilités de transitions p_λ entre la variable d'entrée et la variable de sortie, c'est-à-dire par la donnée de $p(y|x)$. Donc sur un canal, $H(Y|X)$ ne dépend que des p_λ et $I(X,Y) = H(Y) - H(p_\lambda)$.

Ceci permet de calculer la capacité de nombreux canaux, comme celle du canal binaire symétrique, qui se déduit du fait que l'on peut supposer que Y suit également une loi uniforme. Malheureusement, de telles suppositions ne peuvent être faites dans le cas d'un canal quantique, puisqu'il faudrait considérer que l'on peut discrétiser des mots de code, ce qui ne se fait pas comme ça.

Toutefois, dans le cas particulier des codes stabilisateurs, nous pouvons énoncer un résultat proche des notions classiques que l'on connait pour calculer la capacité sur un canal discret.

Contexte :
Considérons l'ensemble des erreurs pouvant affecter un registre d'un code stabilisateur C, de paramètre $[n,k,d]$, associé au groupe stabilisateur $\mathcal{S}$, et considérons une mesure définie par $\mathcal{H}^{\otimes n} = \overset{\perp}{\oplus}\mathscr{E}_s C$
On note S la variable associée au syndrôme erreur σ, S est à valeur dans $\mathscr{S} = \mathbb{F}_2^{n-k}$.
Par ailleurs, pour une erreur P donnée sur un registre $|\varphi\rangle$, on remarque que
$(Q+P)|\varphi\rangle = P|\varphi\rangle$ quelque soit $Q \in \mathcal{S}$ (Q erreur neutre). Pour calculer l'entropie conditionnelle d'une erreur connaissant la valeur d'un syndrôme, on va considérer E, la variable aléatoire associée à une erreur $\mathscr{E}$, à valeur dans $\dfrac{G_n}{\mathcal{S}}$.
On peut alors définir

$$\begin{cases} p(E = \mathscr{E}) = \displaystyle\sum_{e \in \mathscr{E}} p(E_{\text{reel}} = e) \\ \sigma(\mathscr{E}) = \sigma(e_0) \text{ tel que } \mathscr{E} = e_0 + \mathcal{S} \end{cases}$$

Alors $p(S=s) = \sum_{e,\sigma(e)=s} p(E_{\text{reel}}=e) = \sum_{\mathscr{E},\sigma(\mathscr{E})=s} p(E=\mathscr{E})$,
et pour $\mathscr{E} \in \frac{G_n}{S}$ tel que $\sigma(\mathscr{E}) = s$,
$p(E=\mathscr{E}|S=s) = \sum_{Q\in\mathcal{S}} p(E_{reel} = e_0 + Q|S=s)$ tel que $\sigma(e_0) = s$

En se plaçant dans ce contexte tout à fait général pour les codes stabilisateurs, on énonce ainsi le théorème utilisé dans les articles traitant du sujet (cf [6], [7]) mais non démontré (ni vraiment énoncé) :

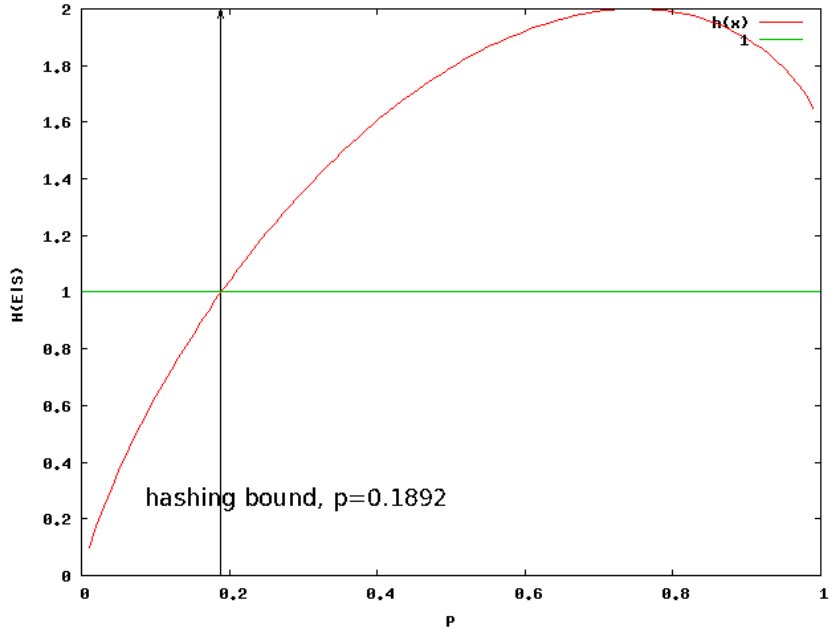
Théorème 2. *Pour un code stabilisateur C de paramètre-dimension k tel que $H_2(E|S) < k$, il existe une famille de codes stabilisateurs concaténés $(C_n)_{n\geq 1}$ utilisant C comme code interne, de rendement supérieur à une constante $R > 0$ donnée, qui soit telle que la probabilité d'erreur après décodage de C_n tend vers 0 lorsque n tend vers l'infini.*

Remarque .

1. *H_2 représente l'entropie prise en base 2.*
2. *On explique la construction de codes concaténés dans la sous section suivante.*
3. *Une erreur neutre correspondrait en classique à une erreur, par exemple deux bits distincts inversés, qui ne changerait pas le sens du mot. C'est un peu comme si les erreurs neutres quantiques représentaient des synonymes naturels pour les mots quantiques.*
4. *Ce théorème est vrai quelque soit le modèle de canal étudié, qu'il soit même quantique ou classique. D'ailleurs, dans le cas des codes linéaires classiques, on vérifie qu'on a exactement, avec les notations usuelles, $H(X|Y) = H(E|S)$. Et en considérant le second théorème de Shanon, le théorème 2 devient alors évident.*

Nous nous placerons maintenant sur le canal de dépolarisation.

hashing borne :
La « borne de hachage » (hashing bound) est une borne quantique sur la capacité, calculée pour des codes stabilisateurs aléatoires de longueur $n = 1$, conséquence directe du théorème 2.
$H(E|S) = \sum p(s) \sum p(\mathscr{E}|s) \log p(\mathscr{E}|s)$.
Sur le canal de dépolarisation, cela donne :
E est à valeur dans $\{I, X, Y, Z\}$, S dans $\{0, 1\}$,
$p(S=0) = 1-p$, $p(S=1) = p(E=X) + p(E=Y) + p(E=Z) = 3\frac{p}{3} = p$
donc $H(E|S) = p \log \frac{p}{3} + (1-p) \log p$.



On en déduit p_{max} tel que $H(E|S) < k$ de l'ordre de

$$\boxed{p_{max} \sim 0.1892}$$

La borne de hachage a été pensé au départ comme la limite maximale de la capacité sur le canal de dépolarisation, ce qui a été démentit par la suite.

A l'heure actuelle, les meilleures bornes connues sur la capacité $\mathcal{Q}$ d'un canal de dépolarisation $\mathscr{C}(p)$ donne

$$\begin{array}{lll} \mathcal{Q} = 0 & \text{si} & p > 0.25 \\ \mathcal{Q} > 0 & \text{si} & p \leq 0.1913 \end{array}$$

Pour trouver une meilleure borne inférieure, sujet de ce rapport, il suffit d'exhiber un code dont $k - H(E|S)$ est strictement positif, pour un meilleur seuil de correction. C'est exactement la démarche qui a été suivi ici.

3.2 Comment construire de (bons) codes correcteurs

En résumé, un bon code correcteur est un code réalisant le meilleur compromis redondance/seuil de correction. On veut à la fois minimiser la redondance pour augmenter le rendement, et augmenter la redondance pour améliorer le seuil de correction. La redondance se mesure avec l'entropie conditionnelle, on cherche donc, sur un canal $\mathscr{C}(p)$, à minimiser au maximum l'entropie conditionnelle telle que $H(E|S)$ soit inférieure à k pour p le plus grand possible.

Regardons tout d'abord comment construire des codes correcteurs quantiques. Il existe plusieurs méthodes génériques, notons que nous avons négligé les méthodes fondées sur les graphes, celles-ci pourraient être étudiées ultèrieurement.

3.2.1 Codes CSS

Les codes **CSS**, acronyme issu des noms de leurs auteurs Calderbank-Shor-Steane, constituent une sous-famille des codes stabilisateurs. Ils sont formés à partir de codes linéaires classiques et exploitent en particulier le concept de code dual. Pour construire un code CSS, on utilise deux codes linéaires C_1 $[n, k_1]$ et C_2 $[n, k_2]$ tels que $k_2 < k_1$ et $C_2 \subset C_1$.

Notons G_{C_2} (respectivement $G_{C_1^\perp}$) la matrice génératrice de C_2 (respectivement $C_1^\perp$), c'est à dire la matrice dont les lignes forment une base de C_2 (respectivement $C_1^\perp$).
Alors, et en utilisant les notations de la section 2.4, le groupe $\mathcal{S}$ défini par

$$\mathcal{S} = \begin{array}{|c|c|} \multicolumn{1}{c}{\overset{n}{\longleftrightarrow}} & \multicolumn{1}{c}{\overset{n}{\longleftrightarrow}} \\ \hline G_{C_2} & \\ \hline & G_{{C_1}^\perp} \\ \hline \end{array}$$

est un groupe stabilisateur d'ordre $n - k_1 + k_2$.

Démonstration :
Tout d'abord C_2 est de paramètre $[n, k_2]$ donc les lignes de G_{C_2} forment une famille libre de k_2 vecteurs.
Autorisons nous à noter un code linéaire : (paramètres, matrice de parité, matrice génératrice).
On rappelle que le dual d'un code linéaire $([n, k], H, G)$ est un code $([n, n-k], G, H)$.
Donc les lignes de $G_{{C_1}^\perp}$ forment une famille libre de $n - k_1$ vecteurs.
Ensuite, par construction, on identifie les lignes de G_{C_2} (respectivement $G_{C_1^\perp}$) soit à des vecteurs de $\{I, X\}^n$ (respectivement $\{I, Z\}^n$), soit à des vecteurs de $\{I, Z\}^n$ (respectivement $\{I, X\}^n$), donc les lignes de G_{C_2} (respectivement $G_{C_1^\perp}$) sont orthogonales entre elles.
Finalement, pour montrer la mutuelle orthogonalité des éléments de $\mathcal{S}$, il ne reste plus qu'à vérifier que ${C_1}^\perp \subset {C_2}^\perp$, et que $C_2 \subset ({C_1}^\perp)^\perp$
C'est une conséquence directe de $C_2 \subset C_1$. $\diamond$
Exemple :
Le plus simple des code CSS est construit avec le code linéaire de Hamming classique $[7, 4, 3]$, que l'on va noter C_h.
La matrice de parité H asssociée à C_h est :

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

Et la matrice gnératrice G est :

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

Posons $C_1 = C_h$ et $C_2 = C_h^{\perp}$.
On vérifie que $C_2 \subset C_1$, et comme le code de Hamming vérifie même $C^{\perp\perp} = C$, on a aussi $G_{C_2} = G_{{C_1}^{\perp}} = H$ D'où le groupe stabilisateur du code CSS de C_h sur $C_h^{\perp}$ est :

$$\mathcal{S} = \left| \begin{array}{c|c} XIXIXIX & \\ IXXIIXX & \\ IIIXXXX & \\ & ZIZIZIZ \\ & IZZIIZZ \\ & IIIZZZZ \end{array} \right| \overset{Not}{=} \left| \begin{array}{c} XIXIXIX \\ IXXIIXX \\ IIIXXXX \\ ZIZIZIZ \\ IZZIIZZ \\ IIIZZZZ \end{array} \right|$$

Le code ainsi formé, nommé **code de Steane** à 7 qubits est de paramètre $[7, 3, 1]$.

3.2.2 Codes concaténés

Soient C^I, C^E deux codes stabilisateurs de paramètres respectifs $[n_i, 1, d_i]$ et $[n_e, k_e, d_e]$.

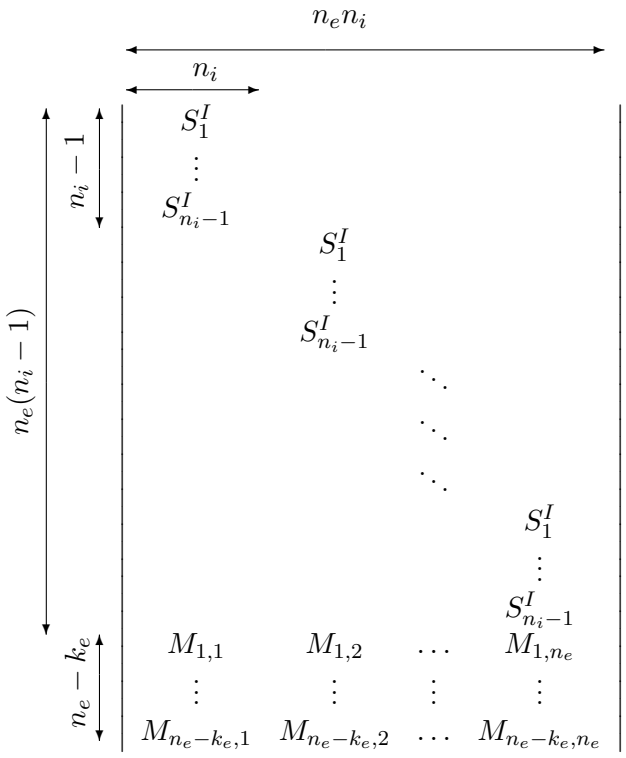
Notons : $\mathcal{S}^I = \left| \begin{array}{c} S_1^I \\ \vdots \\ S_{n_i-k_i}^I \end{array} \right|$ le stabilisateur du code interne,

$\overline{X}_1^I$ et $\overline{Z}_1^I$ ses opérateurs logiques, et $\overline{Y}_1^I = \overline{X}_1^I + \overline{Z}_1^I$,

$$\mathcal{S}^E = \left| \begin{array}{c} S_1^E \\ \vdots \\ S_{n_e-k_e}^E \end{array} \right| \text{ le stabilisateur du code externe,}$$

et pour $i = 1, ..., k_e$, $\begin{cases} \overline{X}_i^E \\ \overline{Z}_i^E \\ \overline{Y}_i^E = \overline{X}_i^E + \overline{Z}_i^E \end{cases}$ les opérateurs logiques du code externe.

Pour concaténer C^I en tant que **code interne** avec C^E comme **code externe**, on construit les stabilisateurs du code concaténé par bloc de la manière suivante :



où $\forall\, i \in \{1, \dots, n_e - k_e\},\, j \in \{1, \dots, n_e\}, M_{ij} = \begin{cases} \overline{X}_1^I & \text{si} \quad (S_i^E)_j = X \\ \overline{Y}_1^I & \text{si} \quad (S_i^E)_j = Y \\ \overline{Z}_1^I & \text{si} \quad (S_i^E)_j = Z \\ \mathbf{I} = \underbrace{I \dots I}_{n_i} & \text{si} \quad (S_i^E)_j = I \end{cases}$

en notant $(S_i^E)_j$ le $j$ième terme de (S_i^E).

On obtient par construction un ensemble de $n_e(n_i - k_i) + n_e - k_e$ générateurs dans $G_{n_e n_i}$ mutuellement indépendants et commutatifs (la vérification se fait facilement), qui définissent un code de paramètre $[n, k] = [n_i n_e, k_e]$.

Exemple :
Prenons le code bit-flip à 3 qubits comme code interne, et le code phase-flip à 3 qubits comme code externe.

$\mathcal{S}^I = \left| \begin{array}{c} ZZI \\ IZZ \end{array} \right|$

$\overline{X}_1^I = XXX$
$\overline{Z}_1^I = ZII$
$\overline{Y}_1^I = YXX$

$\mathcal{S}^E = \left| \begin{array}{c} XXI \\ IXX \end{array} \right|$

$\overline{X}_1^E = XII$
$\overline{Z}_1^E = ZZZ$
$\overline{Y}_1^E = YZZ$

Alors, le groupe stabilisateur du code concaténé est engendré par :

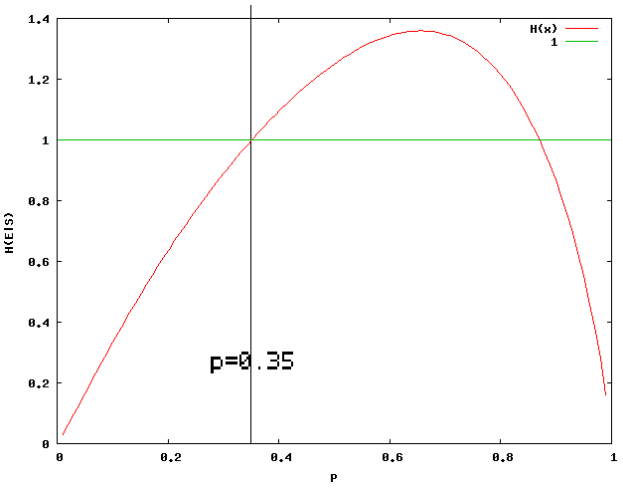
$$\mathcal{S} = \left|\begin{array}{ccc} ZZI & & \\ IZZ & & \\ & ZZI & \\ & IZZ & \\ & & ZZI \\ & & IZZ \\ \overline{X}_1^I & \overline{X}_1^I & \mathbf{I} \\ \mathbf{I} & \overline{X}_1^I & \overline{X}_1^I \end{array}\right| = \left|\begin{array}{ccc} ZZI & & \\ IZZ & & \\ & ZZI & \\ & IZZ & \\ & & ZZI \\ & & IZZ \\ XXX & XXX & III \\ III & XXX & XXX \end{array}\right|$$

On remarque que l'on a exactement formé le code de Shor à 9 qubits.

3.2.3 Sur quels paramètres peut-on jouer ?

Le modele de canal
Contexte : On travaille sur le canal de dépolarisation, défini par les probabilités de transitions $(p_X, p_Y, p_Z, p_I) = (\frac{p}{3}, \frac{p}{3}, \frac{p}{3}, 1-p)$. Tout en restant sur ce canal, on aimerait fabriquer des codes qui après concaténation des uns par les autres, modifient la distributions de ces probabilités pour le code concaténé de la manière suivante :

1. On aimerait conserver une configuration symétrique : $p_X = p_Z$.
2. Sur le canal de dépolarisation, on suppose que la réalisation de Y est indépendante des réalisations de X et Z. Pourtant Y est l'erreur qui correspond à l'action conjointe de X et Z sur un même qubit. On aimerait tester un modèle du type $(p_X, p_Y, p_Z, p_I) = (a, a^2, a, p_I)$. Ce modèle ci tout seul n'est pas concevable puisque alors $p_I = 1 - 2a - a^2$ et $[0, 1]$ n'est pas stable par la fonction $f(x) = -x^2 - 2x + 1$. Par contre, le modèle $(p_X, p_Y, p_Z, p_I) = (p(1-p), p^2, p(1-p), (1-p)^2)$ est tout à fait imaginable.
   Certaines études confirment cette intuition. Dans [7], G. Smith montre qu'on améliore sensiblement le seuil atteint par des « cat codes »(type de code comme le bit-flip ou le phase-flip, cf section 4.2.3) sur ce type de canal, comparativement au même code sur le canal de dépolarisation.
   Pour se donner une idée, regardons l'entropie conditionnelle d'un code aléatoire de longueur 1 sur ce modèle de canal :
   $H(E|S) = (2p^2 - p)\left(2p(1-p)\log p(1-p) + p^2 \log p^2\right) + (1-p)^4 \log (1-p)^2$



On sait toutefois que la capacité est inférieure à 0.25, ce modèle ne peut donc pas être perçu comme un modèle atteignable, mais il suggère un modèle bien meilleur que le canal de dépolarisation à priori. Notre but étant d'améliorer le seuil de correction sur le canal de dépolarisation, nous allons essayer de nous servir de la concaténation pour changer les choses et tendre, après concaténation, vers le second modèle.

Dégénérescence et distance minimale

Bien sûr, on voudrait que la distance minimale du code obtenu soit la plus grande possible pour corriger un maximum d'erreur. Par ailleurs, la distance minimale est liée à la dégénérescence du code, puisque un code est dégénéré si le poids minimale des éléments du stabilisateur est inférieur à la distance minimale. La dégénérescence est probablement une source riche dans la recherche de meilleurs codes, puisqu'un code dégénéré possède la propriété qu'une seule transformation permette de corriger différentes erreurs.

4 Le temps des résultats

4.1 Programmation entropique

Le calcul de l'entropie résiduelle sur un code $C : [n, k, d]$ a été mené à l'aide de la recherche exhaustive des coefficients des polynômes enumérateurs des poids de toutes les erreurs $\mathscr{E} \in \frac{G_n}{\mathcal{S}}$, soit de 2^{n+k} polynômes.

4.1.1 Polynômes enumérateurs de poids

Le **polynôme enumérateur des poids** d'un sous espace vectoriel E de $\mathbb{F}_2^n$ est le polynôme homogène défini par la formule suivante :

$$W_E(x, y) = \sum_{v \in E} x^{n-\omega(v)} y^{\omega(v)} = \sum_{i=0}^{n} \mathcal{N}_i \, x^{n-i} y^i$$

où $\omega(v)$ est le poids de v et $\mathcal{N}_i$ le nombre de vecteurs de poids i dans E.

Rappelons que l'on cherche à calculer :
$H(E|S) = \sum_{s \in \mathscr{S}} p(s) \sum_{\mathscr{E} \in \frac{G_n}{\mathcal{S}}} p(\mathscr{E}|s) \log p(\mathscr{E}|s)$
Par ailleurs, $p(\mathscr{E} \,|\, s) = \sum_{Q \in \mathcal{S}} p(E_{\text{réel}} = \mathscr{E}_0 + Q \,|S\, = s)$ où $\mathscr{E} = \mathscr{E}_0 + \mathcal{S}$ (cf section 3.1)
Or, par définition du canal de dépolarisation, pour une erreur e,
$p(e|s) = (1-p)^{n-\omega(e)} (\frac{p}{3})^{\omega(e)}$ où $\omega(e)$ est le poids de Pauli d'une erreur e
donc $p(\mathscr{E} \,|\, s\,) = \sum_{e \in \mathscr{E}} (1-p)^{n-\omega(e)} (\frac{p}{3})^{\omega(e)} = W_{\mathscr{E}}(\frac{p}{3}, p)$.
De plus, $\sum_{s \in \mathscr{S}} p(s) = \sum_{\mathscr{E}, \sigma(\mathscr{E}) = s} p(\mathscr{E} \,|\, s) = \sum_{\mathscr{E}, \sigma(\mathscr{E}) = s} W_{\mathscr{E}}(\frac{p}{3}, p)$.
On est donc amené à calculer :

$$\boxed{H(E|S) = \sum_{s \in \mathscr{S}} \Big(\sum_{\sigma(\mathscr{E}) = s} W_{\mathscr{E}}(\frac{p}{3}, p) \Big) \sum_{\sigma(\mathscr{E}) = s} \Big(W_{\mathscr{E}}(\frac{p}{3}, p) \log W_{\mathscr{E}}(\frac{p}{3}, p) \Big)} \tag{7}$$

4.1.2 Algorithme

Entrée : n, k, $n-k$ générateurs d'un groupe stabilisateur $\mathcal{S}$, $2k$ générateurs de $\frac{\mathcal{S}^\perp}{\mathcal{S}}$, $n-k$ générateurs de $\frac{G_n}{\mathcal{S}^\perp}$, $\{p_1, \ldots, p_l\}$.

Sortie : $H(E|S)$ pour $p = p_1, \ldots, p_l$

L'algorithme programmé est l'algorithme naïf de calcul de la formule (7).

La compléxité de cet algorithme est en $O(4^n)$.

Remarque . *Le programme principal a été optimisé pour que les calculs soient faits avec le moins d'opérations possibles (calcul de poids, utilisation d'un code de Gray*

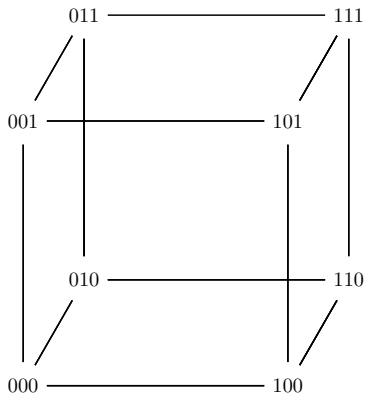


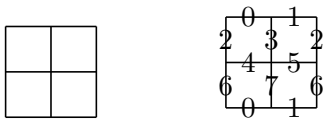
FIG. 5 – Illustration d'un code de Gray.
Chaques sommets adajacents ne diffèrent que d'un bit.

pour parcourir les tableaux) et pour utiliser au mieux la parallélisation des calculs, qui s'y prête bien.

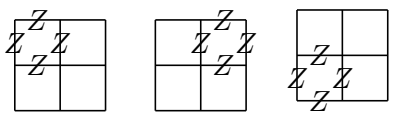
4.2 Les familles de codes testées

4.2.1 Codes toriques

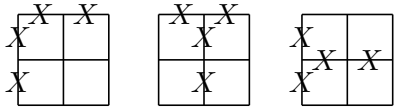
Les codes toriques sont une sous famille des codes stabilisateurs définis par un tore de $\mathbb{Z}^2$. Le quadrillage défini par le tore sur $\mathbb{Z}^2$, définit des relations de parité de poids 4. En identifiant X et Z à chaque arête, et en introduisant une numérotation des arêtes, on peut former des éléments de G_n en X et des éléments en Z, tous de poids 4, commutatifs, et indépendants. On obtient de cette manière, pour $h \in \mathbb{N}$, une famille de codes de paramètre longueur : $n = 2h^2$, de dimension fixe $k = 2$, et de distance minimale $d = h$.
Les codes de longueur $n = 8$ et $n = 18$ ont été testés.
Code torique à 8 **qubits** $[8, 2, 2]$ **:**



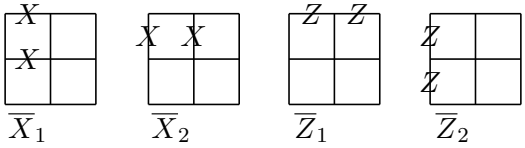
On définit 3 générateurs du stabilisateur, en Z, de poids 4, en choisissant 3 des 4 « carrés » possibles :



On définit 3 génerateurs en X de poids 4 en choisissant 3 des 4 « croix » possibles :

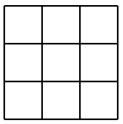


Et on peut prendre les opérateurs logiques de telle manière :



$$\text{Soit : } \mathcal{S} = \left| \begin{array}{c} ZIZZZIII \\ IZZZIZII \\ ZIIIZIZZI \\ XXXIIIXI \\ XXIXIIIX \\ IIXIXXXI \end{array} \right| \quad \begin{array}{l} \overline{X}_1 = XIIIXIII \\ \overline{X}_2 = IIXXIIII \\ \overline{Z}_1 = ZZIIIIII \\ \overline{X}_2 = IIZIIIZI \end{array}$$

Code torique à 18 **qubits** $[8, 2, 3]$ **:**



On construit exactement de la même manière le stabilisateur à partir de ce tore, ainsi on définit 8 génerateurs en Z de poids 4 en choisissant 8 des « 9 » carrés possibles, on définit 8 génerateurs en X de poids 4 en choisissant 8 des 9 « croix » possibles, et les opérateurs logiques convenables.

Remarque. D'une part, pour concaténer un code en interne, il faut que la dimension du code interne soit 1. On peut créer des codes de dimension 1 à partir des codes toriques réguliers. Pour cela, il suffit de compléter le stabilisateur du code d'origine par un vecteur de $\langle \overline{X}_i, \overline{Z}_j \rangle_{1 \leq i,j \leq k}$.
D'autre part, pour les codes de dimension 1, on a considéré des paramètres « avancés »$[n, k, d_I, d_X, d_Y, d_Z]$, où d_I est le poids minimal d'un élément du stabilisateur, et d_X, d_Y, d_Z le poids minimal respectif des éléments de chaque classe d'équivalence de $\dfrac{S^\perp}{S}$. La distance minimale est alors $d = min(d_X, d_Y, d_Z)$.

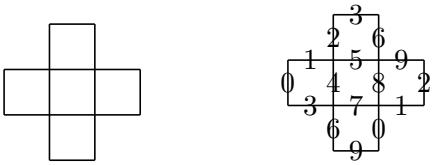
Code torique à 8 **qubits** $[8, 1, 3]$ **:**
A partir du code torique à 8 qubits, en posant $S'_7 = \overline{Y}_1 \overline{Y}_2$, on construit un code torique irrégulier de paramètres avancés :

n	k	d_I	d_X	d_Y	d_Z
8	1	4	4	4	3

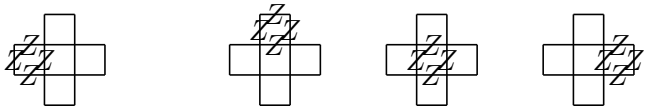
4.2.2 Codes « escaliers » ou code GZ

Ces codes sont une sous famille des codes stabilisateurs, de paramètre $t \in \mathbb{N}$, construit de la manière suivante : le quotient de $Z[i]$ par l'idéal engendré par $(t + (t+1)i)Z[i]$ définit un quadrillage dans Z^2 sur lequel on définit notre stabilisateur, exactement comme pour les codes toriques. On construit ainsi des codes de longueur $n = (2t+1)^2 + 1$, de dimension fixe $k = 2$, et de distance minimale $d = 2t + 1$.

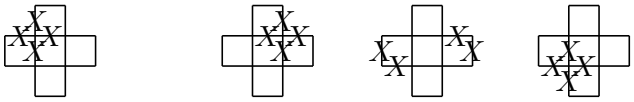
Code $[10, 2, 3]$ **:**



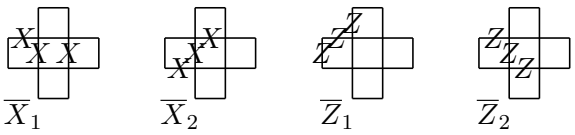
Générateurs du stabilisateur en Z :



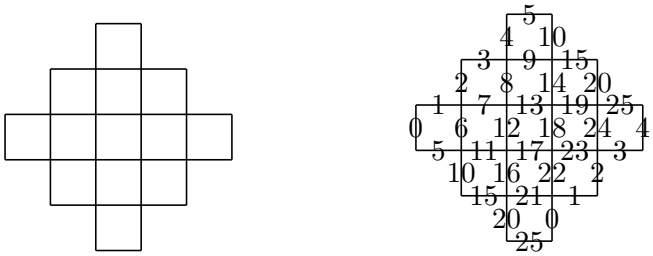
Générateurs du stabilisateur en X :



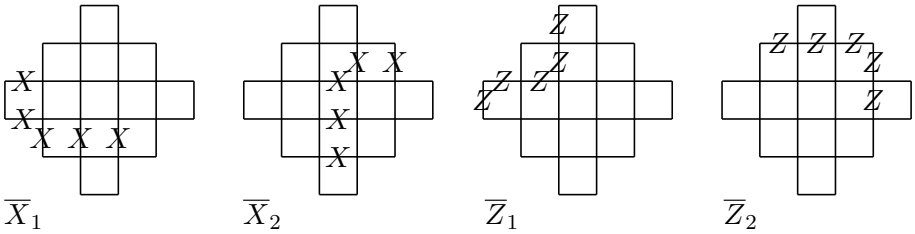
Opérateurs logiques :



Code $[26, 2, 5]$ **:**



Comme pour les codes toriques, on définit 12 générateurs du stabilisateur en Z de poids 4 en choisissant 12 des 13 carrés possibles, et 12 générateurs du stabilisateurs en Z de poids 4 en choisissant 12 des 13 croix possibles.
Pour les opérateurs logiques, on pourra prendre :



Tout comme pour les codes toriques, on peut également construire des codes irréguliers de dimension $k = 1$ à partir de ces codes.

Code $[10, 1, 3]$ **:**
En posant $S'_9 = \overline{X}_1$, on obtient un code de paqramètre :

n	k	d_I	d_X	d_Y	d_Z
10	1	3	3	3	4

Code $[26, 1, 5]$ **:**
En posant $S'_{25} = \overline{X}_1$, on obtient un code de paramètre :

n	k	d_I	d_X	d_Y	d_Z
26	1	4	5	5	8

4.2.3 Codes de Shor (cat codes)

Construits sur le modèle des codes bit-flip ou phase-flip à 3 qubits, on peut faire référence à ces codes comme aux codes de Shor ou plus souvent aux « cat codes » .

Code phase-flip ou Shor -Z $[3, 1, 1]_Z$ **:**

$$\mathcal{S} = \left| \begin{array}{c|c} ZZI & \overline{X}_1 = XXX \\ IZZ & \overline{Z}_1 = ZII \end{array} \right.$$

Code bit-flip ou Shor -X $[3,1,1]_X$ **:**

$$\mathcal{S} = \left| \begin{array}{c|c} XX\,I & \overline{X}_1 = Z\,Z\,Z \\ I\,XX & \overline{Z}_1 = X\,I\,I \end{array} \right.$$

Code de Shor -Y $[3,1,1]_Y$ **:**

$$\mathcal{S} = \left| \begin{array}{c|c} YY\,I & \overline{X}_1 = XXX \\ I\,YY & \overline{Z}_1 = Y\,I\,I \end{array} \right.$$

A une permuation près, ces trois codes ont bien sur les mêmes paramètres, de type :

n	k	d_I	d_X	d_Y	d_Z
3	1	2	3	1	3

Code de Shor $-Y$ $[5,1,1]_Y$ **:**

$$\mathcal{S} = \left| \begin{array}{c|c} YY\,I\,I\,I & \\ I\,YY\,I\,I & \overline{X}_1 = XXXXX \\ I\,I\,YY\,I & \overline{Z}_1 = Y\,I\,I\,I\,I \\ I\,I\,I\,YY & \end{array} \right.$$

Les codes $-X$ et $-Z$ sont tout à fait symétriques. Les paramètres avancés pour ces trois codes sont, à une permutation près :

n	k	d_I	d_X	d_Y	d_Z
5	1	2	5	1	5

4.2.4 Les marginaux

Le « code à 5 **qubits »**

$$\mathcal{S} = \left| \begin{array}{c|c} XZZXI & \\ IXZZX & \overline{X}_1 = XXXXX \\ XIXZZ & \overline{Z}_1 = Z\,Z\,Z\,Z\,Z \\ ZXIXZ & \end{array} \right.$$

Le code à 5 qubits, comme on l'appelle communément, est le plus petit code de distance 3. Il a la particularité de présenter une symétrie parfaite en X, Y et Z.

n	k	d_I	d_X	d_Y	d_Z
5	1	4	3	3	3

Les codes maison :

Dans l'idée de tendre vers un meilleur modèle de canal, on a essayé de trouver des codes dont les paramètres se rapprocheraient d'un modèle « $d_X = d_z$, $d_Y = d_X^2$ ». En longueur 3, ce n'est évidement pas possible, en longueur 4, c'est presque aussi évident. Nous avons essayé de trouver des codes « intéressants » en longueur 5. Il s'ensuit qu'il existe un certain nombre de codes dont les configurations sont de type :

Code à 5 qubit $-A$ $[5, 1, 2]$:

$$\mathcal{S} = \left| \begin{array}{l} XXXX\,I \\ ZZZZ\,I \\ I\,XYZ\,Z \\ I\,XX\,I\,X \end{array} \right| \quad \begin{array}{l} \overline{X}_1 = XXI\,I\,I \\ \overline{Z}_1 = I\,Z\,Z\,I\,I \end{array}$$

n	k	d_I	d_X	d_Y	d_Z
5	1	3	2	3	2

Code à 5 qubit $-B$ $[5, 1, 2]$

$$\mathcal{S} = \left| \begin{array}{l} X\,I\;\;I\,X\,I \\ I\;\;I\,Z\,I\,Z \\ I\;\;Z\,X\,I\,X \\ Z\,X\,Z\,Z\,I \end{array} \right| \quad \begin{array}{l} \overline{X}_1 = XZ\,I\,I\,I \\ \overline{Z}_1 = I\,XZ\,I\,I \end{array}$$

n	k	d_I	d_X	d_Y	d_Z
5	1	2	2	3	2

Mais il parait difficile de faire « mieux », soit dans ce sens, de trouver un code $[n, k, d_I, 3, 3, 5]$.

4.3 Codes concaténés

Globalement, en concaténant deux codes, nous aimerions que le code externe tende à améliorer les « défauts » du code interne. En effet, la concaténation conserve certaines propriétés du code interne. Par exemple, le poids minimal d'un élément du stabilisateur du code interne sera transmis par construction au code concaténé. La dégénérescence du code interne, en particulier, sera donc préservée. Or la dégénérescence est une bonne propriété pour augmenter potentiellement le seuil de correction d'un code. Seuls des codes dégénérés peuvent dépasser la borne de hachage. On peut donc essayer de prendre des codes dégénérés pour code interne. Le code externe quant à lui pourrait servir à changer la distribution des erreurs, pour tendre vers un meilleur modèle. (cf section 3.2.3)
Les tests effectués ont globalement suivi cette idée. Nous présentons ici une série de résultats obtenus. Le tableau présente le résultat du seuil de correction obtenu ($p_{\max}$) pour chaque code testé, ainsi que les paramètres $[n, k, d_I, d_X, d_Y, d_Z]$ qui donnent quelques informations sur l'effet qu'a eu la concaténation sur le schéma de répartition des erreurs. Suivent les tableaux présentants ces résultats.

Code interne / Code externe	$[7,1,1]_{Steane}$	$[8,2,2]$	$[10,2,3]$	$[10,1,3]$	$[18,2,3]$	$[26,2,5]$	$[26,1,5]$
	$[7,1,4,3,3,3]$ 0.187	$[8,2,2,3,4,5]$ 0.1874	0.188132	$[10,1,3,3,3,4]$ 0.18801	0.18807	0.1879	$26,1,4,5,5,8]$ 0.1874

Code externe \ Code interne	$[3,1,1]_X$	$[3,1,1]_Y$	$[5,1,1]_Y$	$[5,1,2]_C$	$[5,1,2]_D$
	$[3,1,2,1,3,3]$ 0.19013		$[5,1,2,1,5,5]$ 0.19035	$[5,1,2,2,3]$ 0.186 ± 0.001	$[5,1,3,2,2,3]$ 0.187
$[3,1,1]_Y$				$[15,1,2,3,6,6]$ 0.187	
$[5,1,1]_Y$					$[25,1,3,3,10,10]$ 0.188
$[5,1,1]_C$		$[15,1,2,2,,4,5]$ 0.189			
$[5,1,1]_D$		$[15,1,2,2,6,7]$ 0.189			
$[5,1,3]_{qb}$		$[15,1,2,5,5,5]$ 0.1900	$[25,1,2,5,7,7]$ 0.1905	$[25,1,2,6,6,7]$ 0.186 ± 0.001	$[25,1,3,6,6,7]$ 0.188
$[7,1,1]_A$		$[21,1,2,3,4,6]$ 0.1900			
$[8,1,3]$		$[24,1,2,4,7,8]$ 0.1901			

Code externe \ Code interne	$[5,1,1]_Z$	$[5,1,3]_{qb}$	$[7,1,1]_A$	$[8,1,3]$
		$[5,1,4,3,3,3]$ 0.18895	$[7,1,1,1,4,4]$ 0.189	$[8,1,4,4,4,3]$ 0.188
$[5,1,1]_X$	$[25,1,2,5,5,9]$ 0.1905			
$[5,1,1]_Y$				
$[5,1,1]_C$		$[25,1,4,6,6,9]$ 0.188		
$[5,1,1]_D$		$[25,1,4,6,6,9]$ 0.188		

4.4 Analyse des résultats

Conclusion :

- Les codes toriques présentent l'avantage d'avoir des générateurs de poids fixe et faible quelque soit la longueur. Les longueurs testées sont malheureusement trop faibles pour en profiter en testant des codes fortement dégénérés. Le premier code dégénéré apparaitrait pour $n \geq 5$, soit une longueur de 50 qubits.
- Cat codes : ce sont les codes qui présentent les meilleurs résultats, en tant que code interne, et au pire à égalité avec le code à 5 qubits (non dégénéré) en tant que code externe.
- Notons que le résulat de la concaténation d'un cat code à 3 qubits en interne avec un code torique irrégulier à 8 qubits en externe n'est pas si mauvais comparitivement à ce dont on pouvait s'attendre. C'est une piste intéressante pour la suite.

5 Conclusion

Le travail accompli lors de ce stage a permis de confirmer ou d'infirmer le comportement supposé de plusieurs familles de codes correcteurs quantiques. Il a surtout permis de mettre au point un programme de calcul systématique du seuil de correction de n'importe quel *petit* code stabilisateur. La limite raisonnable de ce programme est en effet de tester des codes de longueur pouvant aller jusqu'à 26 qubits (environ douze jours de calcul sur 32 coeurs, le temps est à diviser par 4 pour chaque longueur inférieure). Au programme principal s'ajoutent d'autres petits programmes utiles, permettant de générer ces familles de codes, de générer la « base-syndrôme » adaptée au stabilisateur, et surtout de concaténer les codes entre eux. Par ailleurs, il serait intéressant de chercher un algorithme sous-exponentiel qui permettrait toujours ce calcul systématique mais tenant compte du fait que le nombre de polynômes enumérateurs distincts est bien inférieur au nombre de syndrômes. Le temps de calcul serait sans doute nettement amélioré, sans atteindre toutefois les performances atteintes pour des tests particuliers, qui se basent sur la symétrie du code testé. C'est le cas du code donnant la meilleure borne actuelle, produit de la concaténation d'un cat-code à 5 qubits, avec un code à 51 qubits, cf [6].

D'un point de vue plus personnel, ce stage a été l'occasion de plonger au coeur d'un monde passionant de théorie de l'information quantique, riche de dépendre de plusieurs domaines, de la jeune théorie de l'information aux solides fondations mathématiques. Mais est-ce étonnant qu'encore une fois, le monde quantique soit une superposition de mondes différents ?

Références

[1] Preskill John, *Quantum Computation's Course*
Chap7 : Quantum error correction
Chap5 : Quantum Information Theory .

[2] Le Bellac, Michel, *Introduction à l'information quantique*

[3] Nielsen and Chuang, *Quantum Correction and Quantum Computation.*

[4] Thomas and Cover, *Elements of Information Theorie.*

[5] Tillich Jean-Pierre, Poulain David, Ollivier Harold, *Quantum Serial turbo-codes.*

[6] Fern Jesse, Whakley K. Brigitta, *New lower bounds on the non-zero capacity of Pauli Channels.* arXiv :0708.1597v3 [quant-ph]

[7] Smith G., Smolin J., *Degenerate quantum codes for Pauli channels.* Phys.Rev.Lett 98(3) :030501,2007.quant/ph0604107

[8] Knuth Donald E., *The art of computer programming* Pre-fascicle 2A. A draft of section 7.2.1.1 : generating all n-tuples. Algorithm L, p10.