Université Paris 8
A.A. 2021–2022

Introduction à la théorie des codes

Martino Borello

2 septembre 2021

Table des matières

Introduction

Si quelqu'un nous dit « Il faut canger ! », nous nous rendons immédiatement compte qu'il y a une **erreur** dans la phrase, car « canger » ne veut rien dire. Si nous essayons d'imaginer ce qu'on nous a dit, nous pensons naturellement à « Il faut manger ! », « Il faut changer ! » ou encore à « Il faut ranger ! ». Aucun d'entre nous ne pense à quelque chose comme « Il veut penser ! », car c'est trop **loin** de ce qu'on a entendu. Notre cerveau recherche donc naturellement une phrase sensée qui soit la plus **proche** possible de la phrase que nous avons entendu, pour **corriger** l'erreur.

Malheureusement, comme il y a tant de phrases proches de celle que nous avons entendue, nous restons dans l'incertitude et, pour l'éliminer, nous devons demander des **informations supplémentaires**. Une possibilité très naturelle est de demander à notre interlocuteur de **répéter** ce qu'il a dit. Nous pouvons également lui demander « Quoi ? » et si on reçoit comme réponse « La pomme. », on peut facilement imaginer que la première partie de la phrase aurait du être « Il faut manger ! ». En tout état de cause, la correction de l'erreur a un coût, c'est-à-dire qu'elle nécessite l'ajout d'informations.

Erreur, détection, proximité entre les mots, correction, ajout d'informations, répétition : ces concepts qui font en quelque sorte partie de notre vie quotidienne sont à la base de la théorie des codes, comme nous le verrons dans la suite.

Quelle est la cause de l'erreur ? Ce que nous appellerons le **bruit**. Concrètement, on peut imaginer le bruit comme une interférence dans notre appel téléphonique, comme une rayure sur notre CD, comme une tache sur notre livre... ou vraiment, comme un bruit qui nous empêche d'entendre notre ami qui discute bien avec nous ! Pour **Richard Hamming** (1915-1998), l'erreur était un trou bâclé dans les feuilles qu'il insérait dans l'ordinateur modèle V pendant les week-ends, dans les Laboratoires Bells aux États-Unis. Chaque fois que l'ordinateur découvrait une erreur dans l'entrée, il abandonnait simplement le travail.

« Deux week-ends de suite, je suis arrivé et j'ai trouvé qu'une erreur était survenue et rien n'avait été fait. J'était vraiment en colère et agacé parce que je voulais ces réponses et deux week-ends avaient été perdus. Et alors j'ai dit « Bon sang, si la machine peut détecter une erreur, pourquoi ne peut-elle pas localiser la position de l'erreur et la corriger ? ». »

(R. Hamming)

L'objet principal de ce cours seront les **codes correcteurs d'erreurs**, c'est-à-dire la solution que R. Hamming a proposé dans les années 40 pour résoudre son problème avec l'ordinateur. Ils sont encore utilisés aujourd'hui dans toutes sortes de communications (téléphone, réseaux, clés USB, satellites, sondes spatiales...).

Un autre pionnier dans la domaine (lui aussi aux Laboratoires Bells) a été **Claude Shannon** (1916-2001), qui a jeté les bases mathématiques de la théorie des codes et de l'information avec son article *A Mathematical Theory of Communication*. Célèbre est maintenant son schéma suivant, présenté dans cet article, que nous examinerons en détail dans le cours.

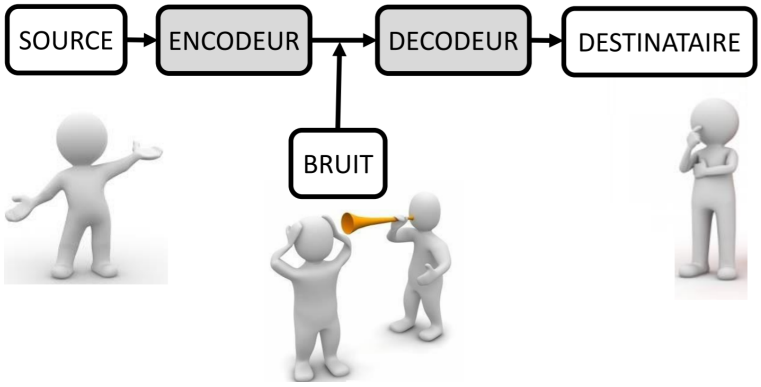

FIGURE 1 –

Enfin, il convient également de rappeler **Marcel Golay** (1902-1989), un collègue des deux chercheurs mentionnés ci-dessus, qui a contribué à l'élaboration de la théorie des codes (dont certains portent son nom).

De nombreux autres mathématiciens, ingénieurs et informaticiens continuent à développer ce domaine de recherche encore aujourd'hui. Ce cours se veut une introduction à ses concepts fondamentaux.

Chapitre 1

Premières notions sur les codes

Tout d'abord, on veut mentionner deux type de codage qui existent, c'est-à-dire le **codage convolutif**, qui traite l'un après l'autre les symboles d'un message de n'importe quelle longueur, et le **codage par bloc**, qui traite « en bloc » des messages avec un nombre fixe de symboles. Dans ce cours, on va considérer seulement le deuxième type de codage, qui est plus simple à développer. Notons que, dans ce cas, un message de n'importe quelle longueur est coupé en plusieurs blocs de la même longueur et chaque bloc est traité séparément.

Soit A un ensemble, dit **alphabet**. Dans le schéma en Figure 1, on a la situation suivante (qui sera notre configuration de base dorénavant) :

- la **source** émet une séquence de k éléments $u := (u_1, \ldots, u_k) \in A^k$, dit **message** ;
- l'**encodeur** transforme (on verra comment) cette séquence en une séquence de n éléments $x := (x_1, \ldots, x_n) \in A^n$, avec $n \geq k$, qui s'appelle **mot de code** de **longueur** n ;
- le **bruit** transforme ce mot de code en un autre $y := (y_1, \ldots, y_n) \in A^n$, de la même longueur, qui est le mot avec erreur ;
- le but (pas toujours possible) du **décodeur** est de reconstituer x à partir de y, c'est-à-dire corriger l'erreur. Avec x on peut facilement reconstituer le message u.

Définition 1.1. *L'ensemble des mots de code, c'est-à-dire l'ensemble de séquences qui sortent de l'encodeur, s'appelle justement* ***code****. Un code de* ***longueur*** *n sur A est un sous-ensemble de A^n.*

On a donc qu'un code est un objet, tandis que le **codage** est la « transformation » operée par l'encodeur. Ces termes sont souvent confus.

Définition 1.2. *Le nombre k/n est dit* ***taux de transmission*** *et l'entier $r = n - k$ est dit* ***redondance****.*

Les deux sont une manière de quantifier l'information qu'on doit ajouter pour permettre la correction de l'erreur. Idéalement, on voudrait un taux de transmission grande ou une redondance petite, mais cela - on verra - n'est pas forcement compatible avec la capacité de corriger l'erreur.

On peut toujours supposer que la transformation opérée par l'encodeur consiste à **ajouter** (on verra comment) des éléments à la séquence u émise par la source (pas forcement après). Donc on peut retrouver u à l'intérieur de x. Dans ce cas, on appelle **bits d'information** ceux de u et **bits de contrôle** ceux qu'on a ajouté.

Définition 1.3. *Un codage est dit* ***systématique*** (*ou* ***lisible***) *quand les bits de contrôle sont regroupés à la fin du mot de code, c'est à dire quand* $x = (u|\ldots)$.

1.1 Exemples historiques

1.1.1 Le code de parité

Soit k un entier positif et $A = \{0, 1\}$. La source émet $(u_1, \ldots, u_k)$ et l'encodeur le transforme en

$$x := (u_1, \ldots, u_k, u_1 + \cdots + u_k \bmod 2),$$

c'est-à-dire il ajoute un bit de contrôle, dit aussi de parité. De cette manière, tout mot de code a un nombre pair de 1. Ce codage est systématique.

Le décodeur reçoit le mot y, qui est égal à x avec des éléments éventuellement changés par le bruit. Si un nombre impair d'éléments a changé, le décodeur peut détecter l'erreur, car il constate qu'il y a un nombre impair de 1. Autrement, il ne peut rien dire. Ce code ne peut pas corriger l'erreur, car il n'est pas capable de le localiser.

Exemple 1.1. *La source envoie* $u = (1, 0, 1, 0, 1)$ *(ici* $k = 5$*), l'encodeur le transforme en* $x = (1, 0, 1, 0, 1, 1)$. *Le bruit change le quatrième élément, de manière que* $y = (1, 0, 1, 1, 1, 1)$. *Le décodeur reçoit* y *et il constate qu'il y a cinq* 1 *: il y a eu une erreur ! Mais en connaissant seulement* y*, il est impossible de localiser l'erreur produit pas le bruit, car il peut être survenu n'importe où. Notons que si le bruit change deux éléments, par exemple le quatrième et le cinquième, de manière que* $y = (1, 0, 1, 1, 0, 1)$*, le décodeur ne détecte pas l'erreur.*

Exercice 1.1. *Écrire la liste des éléments du code de parité pour* $k = 4$.

Notons que le taux de transmission de ce code est $(k+1)/k$ et sa redondance est égale à 1.

1.1.2 Le code de répétition

Soit n un entier positif et $A = \{0, 1\}$. La source émet $u \in A$ et l'encodeur le transforme en

$$x := (\underbrace{u, u, \ldots, u}_{n \text{ fois}}),$$

c'est-à-dire il répète n fois l'élément émis. Ce codage est systématique.

Le décodeur reçoit le mot y, qui est égal à x avec des éléments éventuellement changés par le bruit. Il utilise un décodage « majoritaire », c'est-à-dire il donne au destinataire la valeur qui apparaît le plus de fois (si le nombre de 1 est égale à celui des 0, le décodeur ne peut rien faire). Bien évidemment, il n'est pas certain que l'information reçue par le destinataire soit celle émise par la source. Cela dépend du nombre d'erreurs : si ce nombre est inférieur à $n/2$, alors le décodeur donne la bonne valeur, autrement non.

Exemple 1.2. *La source envoie $u = 1$, l'encodeur le transforme en $x = (1,1,1,1,1,1)$ (ici $n = 6$). Le bruit change le quatrième élément, de manière que $y = (1,1,1,0,1,1)$. Le décodeur reçoit y et il constate qu'il y a cinq 1 et un seul 0, donc il donne 1 au destinataire. Notons que si le bruit change trois éléments le décodeur ne peut rien faire, tandis que si le bruit change plus de trois éléments, le décodeur donne 0 au destinataire, qui n'est pas l'information émise par la source.*

Exercice 1.2. *Écrire la liste des éléments du code de parité pour $n = 5$.*

Notons que le taux de transmission de ce code est $1/n$ et sa redondance est égale à $n - 1$.

1.1.3 Le premier code de Hamming : le code « carré »

Soit $A = \{0,1\}$. La source émet $u := (u_1, u_2, u_3, u_4)$ et l'encodeur le transforme en

$$x := (u_1, u_2, u_1 + u_2, u_3, u_4, u_3 + u_4, u_1 + u_3, u_2 + u_4, u_1 + u_2 + u_3 + u_4).$$

où les sommes sont considérées modulo 2. On peut le visualiser plus facilement avec un carré :

u_1	u_2	$u_1 + u_2 \bmod 2$
u_3	u_4	$u_3 + u_4 \bmod 2$
$u_1 + u_3 \bmod 2$	$u_2 + u_4 \bmod 2$	$u_1 + u_2 + u_3 + u_4 \bmod 2$

qu'on doit déplier pour obtenir x. Notons qu'on l'encodeur ajoute donc un bit de parité à toute ligne et toute colonne du carré formé par u. Ce codage (tel qu'on l'a décrit) n'est pas systématique.

Ce code est capable de corriger une erreur : le décodeur reçoit le mot y, qui est égal à x avec un seul élément changé par le bruit. En regardant la parité des 1 sur toute ligne et sur toute colonne, le décodeur trouve la seule ligne et la seule colonne où il y a un nombre impair de 1. L'élément à changer est donc celui qui est à la ligne et colonne trouvées.

Exemple 1.3. *La source envoie $u = (1,0,1,1)$, l'encodeur utilise le carré pour ajouter les bits de parité*

1	0	1
1	1	0
0	1	1

et il déplie le carré pour obtenir $x = (1, 0, 1, 1, 1, 0, 0, 1, 1)$. *Le bruit change le quatrième élément, de manière que* $y = (1, 0, 1, 0, 1, 0, 0, 1, 1)$. *Le décodeur reçoit* y *et il utilise encore une fois le carré, en pliant d'abord* y

1	0	1
0	1	0
0	1	1

et en remarquant que dans la deuxième ligne et dans la première colonne il y a un nombre impair de 1, *de manière que l'erreur est à la quatrième position de* y.

Exercice 1.3. *Écrire la liste des éléments du « code carré ».*

Notons que le taux de transmission de ce code est 4/9 et sa redondance est égale à 5.

Exercice 1.4. *Peut ce code corriger plus d'une erreur ?*

Exercice 1.5. *En analogie au « code carré », construire un code avec* $k = 9$ *et* $n = 16$. *Combien d'erreurs peut-il corriger ? Peut-on généraliser cette construction ?*

1.2 La métrique de Hamming

Dans l'ensemble A^n on peut définir une métrique.

Définition 1.4. *Soient* $a, b \in A^n$. *La* ***distance de Hamming*** *de* a *à* b *est*

$$d(a, b) := \#\{i \mid a_i \neq b_i\}.$$

Remarque 1.1. *Attention : la distance de Hamming n'est pas le nombre des symboles différents entre deux mots, mais le nombre des positionnes des lettres où les deux mots diffèrent. Exemple : « range » et « nager » ont globalement les mêmes symboles, mais leur distance de Hamming est* 4, *car ils diffèrent à la première, troisième, quatrième et cinquième position.*

Exercice 1.6. *Montrer que la distance de Hamming est une distance (c'est-à-dire elle est positive, symétriques et elle satisfait l'inégalité triangulaire).*

Définition 1.5. *Soit* C *un code. La* ***distance minimale*** *de* C *est*

$$d(C) := \min\{d(a, b) \mid a, b \in C\,, a \neq b\}.$$

Exercice 1.7. *Quelle est la distance minimale du code composé par les mots de longueur* 5 *de la langue française ? Et des trois codes introduits ci-dessus (parité, répétition et « carré ») ?*

La distance minimale d'un code est liée à sa capacité de détection et correction des erreurs. Pour le comprendre, il nous faut d'abord un lemme technique.

Lemme 1.1. *Soit $B(x,t) := \{x' \in A^n \mid d(x,x') \leq t\}$ la boule de rayon t centrée en x. Alors,*

$$B(x,t) \cap B(y,t) = \emptyset \text{ pour tous } x, y \in C,\ x \neq y \iff t \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor.$$

Démonstration. Notons d'abord que

$$B(x,t) \cap B(y,t) = \emptyset \iff d(x,y) > 2t.$$

En effet,

$\Leftarrow$) si $\exists z \in B(x,t) \cap B(y,t)$, alors $d(x,y) \leq d(x,z) + d(z,y) \leq 2t$;

$\Rightarrow$) si $d(x,y) = s \leq 2t$, soit $I := \{i \mid x_i \neq y_i\}$ et $J \subseteq I$ de cardinalité $\lfloor s/2 \rfloor$. Soit $z \in A^n$ tel que $z_i = x_i$ si $i \in J$, et $z_i = y_i$ autrement. Alors $d(y,z) = \#J = \lfloor s/2 \rfloor \leq t$ et $d(x,z) = \#(I \setminus J) = s - \lfloor s/2 \rfloor \leq t$. Donc $z \in B(x,t) \cap B(y,t)$.

Ainsi, si par l'absurde $t > \left\lfloor \frac{d(C)-1}{2} \right\rfloor$, soient $x, y \in C$ tels que $d(x,y) = d(C)$. Or, $d(C) \leq 2t$, de manière que $B(x,t) \cap B(y,t) \neq \emptyset$.
Vice versa, si par l'absurde il existe $x, y \in C$, $x \neq y$, tel que $\{z\} \subseteq B(x,t) \cap B(y,t)$, alors $d(C) \leq d(x,y) \leq d(x,z) + d(z,y) \leq 2t$, ce qui donne une contradiction. □

Soit maintenant x le mot émis de l'encodeur et y le mot modifié par le bruit.

On dit qu'un code C est **t-détecteur** si $B(x,t) \cap C = \{x\}$ pour tout $x \in C$, de manière que le décodeur est capable de détecter jusqu'à t erreurs. En effet, le décodeur normalement est capable de tester (on verra comment) si un vecteur appartient ou pas à C. S'il existe x, x' dans C, avec $x \neq x'$ tel que $x' \in B(x,t) \cap C$, alors $s = d(x,x') \leq t$. Or, si s erreur sur x modifient x en x', le décodeur ne sera pas capable de détecter l'erreur, car x' appartient lui aussi au code C.

On dit qu'un code C est **t-correcteur** si $B(x,t) \cap B(x',t) = \emptyset$ pour tout $x, x' \in C$, de manière que le décodeur est capable de corriger jusqu'à t erreurs. En effet, le décodeur normalement est capable de transformer y dans le mot le plus proche du code, s'il existe (on verra comment). S'il existe x, x' dans C, avec $x \neq x'$ tel que $B(x,t) \cap B(x',t) \neq \emptyset$, alors il existe sûrement un élément y dans l'intersection dont la distance de x' est inférieur ou égale à celle de x. Le décodeur transformera y dans le mot x', en se trompant. Cela ne peut pas arriver si les boules sont disjointes.

Proposition 1.1. *Un code C de distance minimale $d = d(C)$ est $(d-1)$-détecteur et $\lfloor \frac{d-1}{2} \rfloor$-correcteur d'erreurs.*

Démonstration. Clairement un code de distance minimale d ne peut pas détecter d erreur, car si $x, x' \in C$ ont $d(x,x') = d$, alors $\{x, x'\} \subseteq B(x,d)$.
La deuxième affirmation suit directement de la définition et du Lemme 1.1. □

La capacité de correction d'un code dépend donc de la possibilité d'avoir des boules du même rayon (chacune centrée en un mot du code) disjointes. On peut

facilement imaginer que cela contraste avec la possibilité d'avoir beaucoup de boules : si on pense à une boîte qui contient des oranges, plus les oranges sont grosses, moins elles sont dans la boîte. Cela se traduit avec l'inégalité suivante, qu'on énonce seulement pour $A = \{0, 1\}$ (sa généralisation à n'importe quel alphabet est juste un peu plus technique et elle est laissé par exercice).

Théorème 1.1 (Inégalité de Hamming)**.** *Soit C un code dans A^n, $k = \log_2(C)$, $d = d(C)$ et $e = \left\lfloor \frac{d-1}{2} \right\rfloor$. On a*

$$1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{e} \leq 2^{n-k}.$$

Démonstration. Puisque $A = \{0, 1\}$, la distance d'un mot de 0 est simplement le nombre de 1. Donc

$$\#B(0, e) = 1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{e}.$$

Clairement, la cardinalité de toute autre boule de rayon e est la même, par invariance par translation. Le Lemme 1.1 nous assure que les boules centrées dans les mots du code sont disjointes. Ainsi

$$\# \left(\bigcup_{x \in C} B(x, e) \right) = \#C \cdot \left(1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{e} \right) \leq \#A^n,$$

d'où l'inégalité de Hamming. □

Définition 1.6. *Un code C est dit **parfait** si on a l'égalité dans l'inégalité de Hamming.*

Exercice 1.8. *Est-ce que les codes introduits ci-dessus (parité, répétition et « carré ») sont parfaits ?*

Remarque 1.2. *Il y a peu de codes parfaits. On en verra une famille dans ce cours (les codes de Hamming).*

Chapitre 2

Codes linéaires

Comme on a remarqué dans les exercices du chapitre précédent, pour calculer la distance minimal d'un code $C \subseteq A^n$ de cardinalité M il faut faire, a priori, $\binom{M}{2}$ calculs de distances (un complexité $O(nM^2)$). On peut les réduire en prenant un alphabet A tel que $(A; +)$ soit un groupe abélien, et en supposant C un sous-groupe de A^n. En effet, sous ces hypothèses, on a que

$$d(a, b) = d(a - b, 0)$$

pour tout $a, b \in C$ et $a - b \in C$, de manière que

$$d(C) = \min\{d(a, b) \mid a, b \in C\,, a \neq b\} = \min\{d(c, 0) \mid c \in C,\, c \neq 0\}, \qquad (2.1)$$

qui réduit la complexité à $O(nM)$ (on amélioré le calcul, mais ça reste compliqué si la cardinalité est grande).

Un autre problème qu'on a c'est le **stockage** : si on a un « bon code » et veut l'utiliser, on a besoin de connaître ses mots, qui, a priori, doivent être stockés dans un tableau qui serait d'une taille nM, déraisonnable et inadaptée à tout usage pratique. Pour résoudre ce problème, on peut par exemple considérer un alphabet A qui n'est pas seulement un groupe abélien, mais tel que $(A; +, \cdot)$ soit un corps fini, et supposer que C soit un sous-espace vectoriel de A^n. En effet, pour décrire un espace vectoriel il suffit de donner une base, et celle-ci a cardinalité $\log_{\#A}(M)$ (pourquoi ? Le montrer par exercice).

Pour ces deux raisons, la théorie des codes a été du début liée à la théorie des corps finis et à son algorithmique.

2.1 Définitions et propriétés de base

Soit $K = \mathbb{F}_q$ un corps fini de cardinalité q, où q est une puissance d'un nombre premier (pour les généralités sur les corps fini voir par exemple [1, Chapitre 21]), et n un entier positif.

Le K-espace vectoriel K^n est muni de la métrique de Hamming.

Définition 2.1. *Un **code linéaire** C est un K-sous-espace de K^n. Ses **paramètres** sont sa **longueur** n, sa **dimension** $k = \log_q(\#C)$ et sa **distance minimale** d. Si on connaît la distance minimale d, on dit que le code C est un $[n, k, d]_q$ code (on dit simplement $[n, k, d]$ code, si $q = 2$, i.e. si le code est **binaire**), autrement on dit que le code C est un $[n, k]_q$ code (ou simplement $[n, k]$ code si le code est binaire).*

Définition 2.2. *Pour $c \in C$, on appelle **poids** de c le nombre de ses symboles non nuls, i.e.*

$$\mathrm{wt}(c) = \#\{i \mid c_i \neq 0\}.$$

*L'ensemble $\{i \mid c_i \neq 0\}$ est appelé **support** de c et indiqué $\mathrm{Supp}(c)$, de manière que $\mathrm{wt}(c) = \#\mathrm{Supp}(c)$.*

Notons que $\mathrm{wt}(c) = d(c, 0)$. On a donc le résultat suivant.

Proposition 2.1. *La distance minimale d'un code linéaire est égale au plus petit poids non nul de ce code.*

Démonstration. Cela suit du fait que $\mathrm{wt}(c) = d(c, 0)$ et de (2.1). □

Exercice 2.1. *Montrer que*

- *Le code de parité est un $[k + 1, k, 2]$ code.*
- *Le code de répétition est un $[n, 1, n]$ code.*
- *Le code « carré » est un $[9, 4, 4]$ code.*

On a déjà dit que l'un des avantages de considérer un code linéaire c'est qu'on peut le décrire avec une base. En théorie des codes, cette base est présentée toujours en forme de matrice.

Définition 2.3. *Une **matrice génératrice** G d'un $[n, k, d]_q$ code C est une matrice $k \times n$ sur K dont les lignes sont les vecteurs d'une base de C.*

Remarque 2.1. *Avec un abus de langage, on parle souvent de « la » matrice génératrice d'un code, en sachant que c'est l'une des nombreuses matrices génératrices.*

Exemple 2.1. *Le code $C \subseteq \mathbb{F}_4^4$ avec matrice génératrice*

$$G := \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & \alpha & 1 \end{bmatrix}$$

(où $\alpha \in \mathbb{F}_4$ tel que $\alpha^2 = \alpha + 1$), est le code de longueur 4 et dimension 2 engendré par les vecteurs $(1, 0, 1, 1)$ et $(0, 1, \alpha, 1)$, i.e.

$$C = \{u_1(1, 0, 1, 1) + u_2(0, 1, \alpha, 1) \mid u_1, u_2 \in \mathbb{F}_4\} = \left\{(u_1, u_2)G \mid (u_1, u_2) \in \mathbb{F}_4^2\right\}.$$

Il a donc $4^2 = 16$ mots.

Exercice 2.2. *Donner la liste des éléments du code C de l'Exemple 2.1 et montrer que C est un $[4, 2, 3]_4$ code.*

Exercice 2.3. *Donner une matrice génératrice pour le code de parité, le code de répétition et le code « carré ».*

Si les premières k colonnes d'une matrice génératrice G d'un $[n, k, d]_q$ code C ont rang k, i.e. si la sous-matrice A de G formée par ces colonnes est inversible, alors

$$G' = A^{-1}G = [I_k \mid B],$$

où I_k est la matrice identité de dimension k et B est une matrice $k \times (n-k)$ sur K. Une matrice génératrice de cette forme (identité à gauche) est dite matrice génératrice en **forme systématique**.

Remarque 2.2. *Étant donnée une matrice génératrice G d'un $[n, k, d]_q$ code, l'encodeur multiplie le vecteur $u = (u_1, \ldots, u_k)$ émis par la source par cette matrice G, en obtenant un mot du code C. Si on change de matrice, on change de codage, même si le code ne change pas. Notons que le codage est systématique ssi la matrice génératrice considérée est en forme systématique.*

Exercice 2.4. *Soit C un $[4, 2]_4$ code avec matrice génératrice*

$$G := \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & \alpha^2 & 0 \end{bmatrix}.$$

- *Existe-t-elle une matrice génératrice en forme systématique pour C ? Si oui, la trouver.*
- *Combien de mot a-t-il ? En donner la liste.*
- *Quelle est la distance minimale de C ?*
- *Si $c = (\alpha^2, \alpha, 0, 1)$ sort de l'encodeur (qui a codé avec la matrice G), qui est le vecteur u émis par la source ?*

2.2 Matrice de parité et code dual

Une autre façon de définir un code linéaire est de donner une application linéaire dont il est le noyau (ou de manière équivalente un système linéaire homogène dont il est la solution). La matrice qui représente cette application est appelée matrice de parité ou matrice de contrôle.

Définition 2.4. *Une **matrice de parité** (ou de contrôle) H d'un $[n, k, d]_q$ code C est une matrice $(n-k) \times n$ sur K de rang $n-k$ telle que*

$$C = \{v \in K^n \mid vH^T = (0, \ldots, 0)\}$$

Chaque ligne de la matrice H représente une **équation de contrôle** de code, i.e. une équation linéaire que les coordonnées d'un mot du code doivent satisfaire, et le code est l'ensemble des solutions de ces équations de contrôle (qui sont linéairement indépendantes).

Exemple 2.2. *Le code de parité a matrice de parité donnée par* $H = [1, \ldots, 1]$. *En effet, un vecteur* $v = (v_1, \ldots, v_n) \in \mathbb{F}_2^n$ *appartient au code de parité ssi*

$$vH^T = v_1 + \cdots + v_n = 0$$

dans $\mathbb{F}_2$, *i.e. modulo* 2.

Exercice 2.5. *Montrer que tout mot* $c = (c_1, c_2, c_3, c_4)$ *du code* C *de l'Exemple 2.1 satisfait les équations*

$$\begin{cases} c_1 + \alpha c_2 + c_3 = 0 \\ c_1 + c_2 + c_4 = 0 \end{cases}$$

En déduire qu'un matrice de parité pour C *est*

$$H = \begin{bmatrix} 1 & \alpha & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

Étant donné un code linéaire C avec matrice génératrice G, comment trouver une matrice de parité ? Si on réfléchit un instant, ce problème équivaut à trouver une base du code dont une matrice de parité est égale à G, ce qui équivaut à résoudre un système de k équations linéaires en n inconnus. Cela est beaucoup plus simple lorsque la matrice est en forme systématique.

Proposition 2.2. *Soit* C *un* $[n, k]_q$ *code avec matrice génératrice* $G = [I_k|A]$. *Alors une matrice de parité de* C *est* $H = [A^T| - I_{n-k}]$.

Démonstration. Notons d'abord que $H = [A^T| - I_{n-k}]$ est de rang $n - k$. Donc il suffit de montrer que chaque ligne de H est une équation de contrôle. Puisque tout mot est une combinaison linéaire des lignes $g_1, \ldots, g_k$ de G, il suffit de montrer que $g_i H^T = 0$ pour tout $i \in \{1, \ldots, k\}$. Cela revient à montrer que $GH^T = 0$. Or,

$$GH^T = [I_k|A] \left[\frac{A}{-I_{n-k}}\right] = I_k A - A I_{n-k} = A - A = 0,$$

où la deuxième égalité est une multiplication de matrices par blocs (exercice). □

Exemple 2.3. *Une matrice de parité pour le* $[5, 3]_3$ *code engendré par*

$$G = \begin{bmatrix} 1 & 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{bmatrix}$$

est égale à

$$H = \begin{bmatrix} 2 & 0 & 1 & 2 & 0 \\ 1 & 1 & 2 & 0 & 2 \end{bmatrix}.$$

Donc pour trouver une matrice de parité il vaut mieux trouver d'abord une matrice génératrice en forme systématique, si possible.

Exercice 2.6. *Trouver une matrice de parité pour le $[5,2]_3$ code engendré par*

$$G = \begin{bmatrix} 2 & 0 & 1 & 2 & 0 \\ 1 & 1 & 2 & 0 & 2 \end{bmatrix}.$$

Définition 2.5. *Soit C un code linéaire dans K^n. Le (**code**) **dual** de C est*

$$C^\perp = \{v \in K^n \mid \langle v, c\rangle = 0, \forall c \in C\}$$

où $\langle v, c\rangle = \sum_{i=1}^n v_i c_i$ (produit scalaire standard).
*Si $C \subseteq C^\perp$ le code C est dit **auto-orthogonal**. Si $C = C^\perp$ le code C est dit **auto-dual**.*

Exercice 2.7. *Montrer que si C est un $[n,k]_q$ code avec matrice génératrice G et matrice de parité H, alors $C^\perp$ est un $[n, n-k]_q$ code avec matrice génératrice H et matrice de parité G. En particulier, $(C^\perp)^\perp = C$.*
Montrer qu'un code auto-dual a forcement dimension $n/2$, de manière qu'il peux exister seulement si n est pair.

Exemple 2.4. *Le dual du code parité est le code de répétition, et vice versa.*

Exercice 2.8. *Donner une matrice génératrice du dual du code « carré ».*

Si on veut utiliser la Proposition 2.2 mais le $[n,k]_q$ code n'est pas en forme systématique, on peut chercher k colonnes libres, permuter les colonnes avec une permutation $\sigma \in S_n$ telle que les k colonnes deviennent les premières k colonnes, multiplier à gauche par la matrice inverse de la matrice carrée formée par ces k colonnes, appliquer la Proposition 2.2 et finalement permuter les colonnes avec la permutation σ^{-1}.

Remarque 2.3. *On souligne ici l'interprétation des lignes et des colonnes des matrices génératrices et de parité d'un $[n,k]_q$ code :*

- ***Les lignes** $g_1, \ldots, g_k$ **d'une matrice génératrice** : elles engendrent par combinaison linéaire tous les mots du code, i.e. pour tout $c \in C$, il existe $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_q$ tels que*

$$c = \lambda_1 g_1 + \cdots + \lambda_k g_k.$$

- ***Les lignes** $h_1, \ldots, h_{n-k}$ **d'une matrice de parité** : un vecteur $v \in \mathbb{F}_q^n$ appartient à C ssi il est orthogonal à toute ligne, i.e. ssi $\langle v, h_i\rangle = 0$ pour tout $i \in \{1, \ldots, n-k\}$.*
- ***Les colonnes** $g^{(1)}, \ldots, g^{(n)}$ **d'une matrice génératrice** : pour tout mot $c \in C$, il existe $u \in \mathbb{F}_q^k$ tel que*

$$c = (\langle u, g^{(1)}\rangle, \ldots, \langle u, g^{(n)}\rangle).$$

- ***Les colonnes** $h^{(1)}, \ldots, h^{(n)}$ **d'une matrice de parité** : un vecteur $v = (v_1, \ldots, v_n) \in \mathbb{F}_q^n$ appartient à C ssi*

$$v_1 h^{(1)} + \cdots + v_n h^{(n)} = 0.$$

La dernière partie de la remarque nous indique un lien entre la matrice de parité et la distance minimale d'un code.

Proposition 2.3. *Un code linéaire a distance minimale d ssi toute famille de $d-1$ colonnes de sa matrice de parité est libre et il existe d colonnes linéairement dépendantes.*

Démonstration. Soit $\{h^{(i)}\}_{i\in I}$ un ensemble de $w = \#I \geq 1$ colonnes de la matrice de parité telles qu'il existe une combinaison linéaire (avec des coefficients qui ne sont pas tous égaux à zéro)

$$\sum_{i\in I} \lambda_i h^{(i)} = 0.$$

Le vecteur v tel que $v_j = \lambda_j$ si $j \in I$ et $v_j = 0$ autrement, est un mot non nul du code par le dernier point de la Remarque 2.3, et il a poids au plus w. Donc à tout ensemble de w colonnes liées on peut associer un mot du code de poids au plus w. Vice versa, si c est un mot de poids w, l'ensemble $\{h^{(i)}\}_{i\in \mathrm{Supp(c)}}$ est un ensemble de w colonnes liées, toujours par le dernier point de la Remarque 2.3. Cela nous montre que l'ensemble de mots non nuls de poids au plus w est vide ssi il n'existe pas de famille de w colonnes liées. On conclut graçe à la Proposition 2.1. □

Remarque 2.4. *Étant donné un code C et une matrice de parité H de C, on a en particulier que*

- *$d(C) = 1$ ssi une colonne de H est nulle ;*
- *$d(C) = 2$ ssi il n'y a pas de colonnes nulles et il y a au moins deux colonnes liées (i.e. l'une un multiple de l'autre) dans H ;*
- *$d(C) \geq 3$ ssi il n'y a pas de colonnes nulles ni de couple de colonnes liées dans H.*

Notons que donc pour établir si $d(C) \geq 3$ il suffit de regarder les colonnes d'une matrice de parité sans avoir besoin de donner la liste de tous les mots.

Exercice 2.9. *Quelle est la distance minimale du dual du code « carré » ?*

Exercice 2.10. *Soit C le $[7, 3]_7$ code engendré par*

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 2 & 4 & 6 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Quelle est la distance minimale de C ? Et de $C^\perp$?

2.3 Borne de Singleton et codes MDS

Soit C un $[n, k, d]_q$ code linéaire et H sa matrice de parité. Clairement, elle a au plus $n-k$ colonnes libres, de manière que $n-k+1$ colonnes sont forcement liées. Donc forcement, par la Proposition 2.3,

$$d(C) \leq n - k + 1.$$

Cette inégalité est appelée **borne de Singleton** et elle est vraie même si le code n'est pas linéaire :

Théorème 2.1 (borne de Singleton)**.** *Soit $C \subseteq A^n$ et $k := \log_{\#A}(\#C)$. Alors*

$$d(C) \leq n - k + 1.$$

Démonstration. On peut considérer la projection $\pi : A^n \to A^{n-d(C)+1}$ sur les premières $n - d(C) + 1$ coordonnées. Soient $c, c' \in C$, $c \neq c'$. Puisque $d(c, c') \geq d(C)$, on a $\pi(c) \neq \pi(c')$, de manière que $\pi_{|_C}$ est injective. Donc

$$\#C \leq (\#A)^{n-d+1}.$$

□

Définition 2.6. *Un $[n, k, d]_q$ code est appelé **MDS*** (Maximum Distance Separable) *si $d = n - k + 1$, i.e. si ses paramètres satisfont l'égalité dans la borne de Singleton. Un code est dit **MDS trivial** lorsque $k \in \{1, n - 1, n\}$.*

Exemple 2.5. *Les codes de répétition et de parité sont des exemples de codes MDS triviaux binaires. Le code de l'Exemple 2.1 est un code MDS (non trivial).*

Exercice 2.11 $(*)$**.** *Montrer que, si $A = \mathbb{F}_2$, le code de répétition, le code de parité et l'espace $\mathbb{F}_2^n$ sont les seuls codes MDS possibles, de manière qu'il n'existe pas de codes MDS non triviaux binaires.*

Remarque 2.5 (Propriétés des codes MDS)**.** *Les codes MDS sont, par définition, des codes avec la distance minimale la plus large possible étant données longueur et dimension. Ils sont donc des codes « optimaux » pour la correction des erreurs. Ils ont d'autres propriétés intéressantes qu'on énonce ici laissant leur preuve par exercice.*

1. *Un $[n, k]_q$ code est MDS ssi chaque ensemble de $n - k$ colonnes de sa matrice de parité est de rang $n - k$.*
2. *Si C est MDS, alors $C^\perp$ l'est aussi.*
3. *Un $[n, k]_q$ code est MDS ssi chaque ensemble de k colonnes de sa matrice de génératrice est de rang k.*

Exercice 2.12. *Montrer les propriétés 1., 2. et 3. de la Remarque 2.5.*

2.4 Distribution des poids

Une propriété combinatoire importante d'un code est sa distribution de poids, c'est-à-dire le nombre de mots de chaque poids. Plus précisément, la **distribution des poids** d'un $[n, k]_q$ code est la liste

$$A_0(C), A_1(C), \ldots, A_n(C)$$

avec

$$A_i(C) = \#\{c \in C \mid \mathrm{wt}(c) = i\}.$$

Souvent on écrit seulement les éléments non nuls de cette liste. La distribution de poids d'un code ne donne pas seulement la capacité de correction des erreurs du code, mais permet également de calculer la probabilité de détection et de correction des erreurs.

Remarque 2.6. *Tous les $A_i(C)$ sont des entiers non négatifs. En plus, $A_0(C) = 1$ et $A_0(C) + \ldots + A_n(C) = \#C$ pour tous les codes.*

Exercice 2.13. *Montrer que si C est un $[n, k]$ code binaire et $(1, 1, \ldots, 1) \in C$, alors $A_{n-i}(C) = A_i(C)$.*

Il est de pratique courante de représenter la distribution des poids par un polynôme.

Définition 2.7. *Le **polynôme** (**énumérateur**) **des poids** d'un code C est le polynôme homogène*

$$w_C(x, y) = \sum_{c \in C} x^{n-\mathrm{wt}(c)} y^{\mathrm{wt}(c)} = \sum_{i=0}^{n} A_i(C) x^{n-i} y^i.$$

Exemple 2.6. *On peut facilement calculer les polynômes des poids de nos exemples historiques (exercice).*

- *Le polynôme des poids du $[n, 1, n]$ code de répétition C est*

$$w_C(x, y) = x^n + y^n.$$

- *Le polynôme des poids du $[n, n-1, 2]$ code de parité C est*

$$w_C(x, y) = \frac{(x+y)^n + (x-y)^n}{2}.$$

- *Le polynôme des poids du $[9, 4, 4]$ code « carré » C est*

$$w_C(x, y) = x^9 + 9x^5y^4 + 6x^3y^6.$$

La distribution des poids d'un code et de son dual sont très liés, comme le résultat suivant montre.

Théorème 2.2 (identité de MacWilliams)**.** *Soit C un code sur $\mathbb{F}_q$ et $C^\perp$ son dual. Alors*

$$w_{C^\perp}(x, y) = \frac{1}{\#C} \cdot w_C(x + (q-1)y, x - y).$$

Démonstration. La démonstration de ce théorème dépasse le cadre de ce cours introductif. Les personnes intéressées peuvent consulter le livre [2]. Elle utilise la transformée de Fourier discrète. □

Exercice 2.14. *Montrer (sans calculer la liste de ses éléments) que le polynôme des poids du dual du code « carré » $C^\perp$ est*

$$w_{C^\perp}(x,y) = x^9 + 6x^6y^3 + 9x^5y^4 + 9x^4y^5 + 6x^3y^6 + y^9.$$

Exercice 2.15 $(*)$**.** *Montrer que le polynôme des poids d'un $[4,2,3]_q$ code C (notons que c'est un code MDS) est*

$$w_C(x,y) = x^4 + 4(q-1)xy^3 + (q-1)(q-3)y^4.$$

2.5 Équivalence de codes

‘ Soit S_n le groupe symétrique de degré n, i.e. le groupe des permutations de l'ensemble $\{1,\dots,n\}$. Pour tout $\sigma \in S_n$ et $x = (x_1,\dots,x_n) \in \mathbb{F}_q^n$, on définit

$$x^\sigma = (x_{\sigma^{-1}(1)},\dots,x_{\sigma^{-1}(n)}).$$

Exemple 2.7. $(x_1, x_2 \dots, x_n)^{(1\,2\,\dots\,n)} = (x_n, x_1, \dots, x_{n-1})$.

On peut représenter cette « action » de S_n sur $\mathbb{F}_q$ (pour plus de détails sur les actions des groupes sur les ensembles voir [1, Chapitre 18]) à travers des matrices appelées **matrices de permutation** :

$$x^\sigma = xM_\sigma$$

où M_σ est une matrice dont toutes les coefficients sont nuls à l'exception des coefficients indexés par $(i,\sigma(i))$ qui valent 1.

Exemple 2.8. $M_{(1\,2\,3\,4)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$. *En effet*

$$(x_1,x_2,x_3,x_4)^{(1\,2\,3\,4)} = (x_4,x_1,x_2,x_3) = (x_1,x_2,x_3,x_4)M_{(1\,2\,3\,4)}.$$

Une matrice de permutation est donc une matrice avec un seul élément non nul (et égal à 1) sur toute ligne et sur toute colonne. Elle est toujours inversible. Elle représente la transformation linéaire de $\mathbb{F}_q^n$ associée à σ.

Remarque 2.7. *Une propriété très importante des ces transformations linéaires est le fait qu'elle ne changent pas le poids du vecteur, i.e., pour tout $\sigma \in S_n$ et $x \in F_q^n$,*

$$\mathrm{wt}(x^\sigma) = \mathrm{wt}(xM_\sigma) = \mathrm{wt}(x).$$

*Elle sont donc des **isométries** pour la métrique de Hamming. Elle ne sont pas la forme plus générale d'isométrie qu'on peut considérer : par exemple les **matrices monomiales**, i.e. avec un seul élément non nul (mais pas forcement égal à 1) sur toute ligne et sur toute colonne, représentent des isométries aussi. Dans la cadre de ce cours, pour simplifier la notation, nous ne considérerons que les matrices de permutation.*

L'« action » de S_n sur $\mathbb{F}_q^n$ induit une « action » sur les codes dans $\mathbb{F}_q^n$, donnée par

$$C^\sigma = \{c^\sigma \mid c \in C\},$$

pour $C \subseteq \mathbb{F}_q^n$ et $\sigma \in S_n$.

Définition 2.8. *Soit C un code dans $\mathbb{F}_q^n$. Un élément $\sigma \in S_n$ est un **automorphisme** (de permutation) de C si $C^\sigma = C$.*

Exercice 2.16. *Montrer que l'ensemble $\mathrm{Aut}(C)$ des automorphismes d'un code C est un sous-groupe de S_n.*

$\mathrm{Aut}(C)$ est appelé **groupe des automorphismes** (de permutation) de C.

Exercice 2.17. *Soit C un code linéaire. Montrer que $\mathrm{Aut}(C) = \mathrm{Aut}(C^\perp)$.*

Définition 2.9. *Soient C_1 et C_2 deux codes dans $\mathbb{F}_q^n$. Ils sont **équivalents** (par permutation) s'il existe $\sigma \in S_n$ telle que*

$$C_2 = C_1^\sigma.$$

Notation : $C_1 \sim C_2$.

Exercice 2.18. *Montrer que $\sim$ est une relation d'équivalence.*

Par la Remarque 2.7, codes équivalents (par permutation) ont les mêmes paramètres et la même distribution des poids. Attention : le vice versa n'est pas vrai, comme montré dans l'exemple suivant.

Exemple 2.9. *Les deux $[4, 2, 3]_7$ codes C_1 et C_2 avec matrices génératrices respectivement*

$$G_1 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

et

$$G_2 = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 3 & 4 \end{bmatrix}$$

ont le même polynôme des poids, i.e.

$$w_{C_1}(x, y) = w_{C_2}(x, y) = x^4 + 24xy^3 + 24y^4,$$

mais ils ne sont pas équivalents (on peut par exemple vérifier - avec des longs calculs - que C_2 ne contient aucun mot de poids 3 avec tous les éléments non nuls égaux, tandis que C_1 si).

2.6 Décodage par syndrome

On déjà vu que dans notre configuration de base l'encodeur sort un mot de code $x \in C \subseteq K^n$ et que le bruit le transforme en un autre élément $y \in K^n$. Dans le cas des codes linéaires, on peut exprimer l'erreur comme

$$e = y - x \in K^n.$$

Soit H une matrice de parité pour C. Le décodeur peut facilement détecter s'il y a eu une erreur en calculant yH^T. En effet, $yH^T = 0$ ssi $y \in C$. Donc, si $\mathrm{wt}(e) \leq d(C) - 1$, le décodeur dira (justement) qu'il y a eu une erreur ssi $yH^T \neq 0$.

Définition 2.10. *On appelle yH^T le **syndrome** de y.*

Notons que $y = x + e$ et

$$yH^T = (x + e)H^T = xH^T + eH^T = 0 + eH^T = eH^T,$$

de manière que le syndrome de y (que le décodeur peut facilement calculer) est le même que celui de e (que le décodeur ne connaît pas).

Supposons que le décodeur ait fait un pré-calcul des tous les syndromes des erreurs possibles de poids inférieur ou égale à $\left\lfloor \frac{d(C)-1}{2} \right\rfloor$ en les mettant dans un tableau du type

Mots de poids $\leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$	syndromes
e	eH^T

Exercice 2.19. *Montrer que tous les syndromes dans le tableau sont différents.*

Si $\mathrm{wt}(e) \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$, le décodeur reçoit y, il calcule son syndrome yH^T et il le compare avec les syndromes présents dans le tableau. Il obtient donc e, et ensuite $x = y - e$.

Remarque 2.8. *Ce procédé est particulièrement simple si $d(C)$ est égale à 3 ou 4 (de manière que le code peut corriger une erreur), car les syndromes sont simplement des multiples des colonnes de la matrice de parité transposées. Dans le cas binaire, ils sont exactement les colonnes de la matrice de parité (transposées). Par contre, ce procédé devient inutilisable en pratique dès qu'on considère de longueur grandes et une capacité de correction plus importante : la taille du tableau devient trop grande. Toutefois, il est un modèle pour d'autres types de décodages.*

Exercice 2.20. *Utiliser le décodage par syndrome pour corriger $y = (\alpha, \alpha^2, 1, 0)$, pour le $[4, 2, 3]_4$ code de l'Exemple 2.1.*

2.7 Extensions des codes

Il existe plusieurs façons d'obtenir de nouveaux codes à partir de familles de codes déjà construites. L'une d'entre elles est l'extension (pour d'autres façons, voir [3, Section 1.5]).

Définition 2.11. *Soit C un $[n, k, d]_q$ code. Son **extension** est le code*

$$\widehat{C} := \{(c_1, \ldots, c_{n+1}) \in \mathbb{F}_q^{n+1} \mid (c_1, \ldots, c_n) \in C \text{ et } c_1 + \ldots + c_{n+1} = 0\},$$

qui est appelé aussi C étendu.

Théorème 2.3. *$\widehat{C}$ est un $[n+1,k]_q$ code et sa distance minimale est d, s'il existe un mot c dans C de poids d tel que $c_1+\ldots+c_n=0$, et $d+1$ autrement.*

Démonstration. On a que $\widehat{C}$ est un code linéaire. En effet, $\widehat{C}$ est stable par combinaison linéaire : soient

$$c=(c_1,\ldots,c_{n+1}) \quad \text{et} \quad d=(d_1,\ldots,d_{n+1})$$

deux mots dans $\widehat{C}$. Alors

$$\lambda c+\mu d=(\lambda c_1+\mu d_1,\ldots,\lambda c_{n+1}+\mu d_{n+1})$$

appartient à $\widehat{C}$. En effet

$$(\lambda c_1+\mu d_1,\ldots,\lambda c_n+\mu d_n)\in C,$$

car C est un code linéaire, et

$$\lambda c_1+\mu d_1+\ldots+\lambda c_{n+1}+\mu d_{n+1}=\lambda(c_1+\ldots+c_{n+1})+\mu(d_1+\ldots+d_{n+1})=0$$

La longueur de $\widehat{C}$ est $n+1$ par définition. À tout mot de C correspond un et un seul mot de $\widehat{C}$, de manière que $\#\widehat{C}=\#C$. Donc $\dim\widehat{C}=k$. L'affirmation sur la distance minimale suit directement de la définition de $\widehat{C}$. □

Soit H une matrice de parité pour C. Une matrice de parité pour $\widehat{C}$ est

$$\left(\begin{array}{ccc|c} & H & & \begin{matrix}0\\ \vdots\\ 0\end{matrix} \\ \hline 1 & \ldots & 1 & 1 \end{array}\right)$$

car aux équations de contrôle de C on ajoute juste $c_1+\ldots+c_{n+1}=0$, qui nous donne la dernière ligne.

Exercice 2.21. *Déterminer les paramètres de l'extension du $[5,3]_3$ code de l'Exemple 2.3 et de son dual.*

Chapitre 3

Familles de codes linéaires remarquables

Nous avons vu dans le premier chapitre quelques familles de codes dont les paramètres n'étaient pas « optimaux ». Dans ce chapitre, nous présentons quelques familles de codes classiques, très étudiées et à la base d'autres constructions. Pour simplifier la notation on va considérer seulement des codes binaires, mais les mêmes familles existent aussi sur d'autres corps finis.

3.1 Codes de Hamming et codes simplexes

On définit les codes de Hamming à partir de leur matrice de parité et à permutation près.

Définition 3.1. *Soit m un entier positif. Le* ***code de Hamming*** $\mathcal{H}_m$ *est le* $[2^m - 1\ ,\ 2^m - m - 1]$ *code binaire dont une matrice de parité H_m est une matrice $m \times (2^m - 1)$ dont les colonnes sont les éléments non nuls de l'espace $\mathbb{F}_2^m$ (dans un ordre quelconque).*

Si on change l'ordre des colonnes, le code reste un code de Hamming : en vérité $\mathcal{H}_m$ définit une classe de codes équivalents. Il convient souvent de choisir l'ordre des colonnes de manière d'avoir l'identité à la fin (ou au début), pour pouvoir appliquer la Proposition 2.2 et trouver une matrice génératrice.

Exemple 3.1. *On peut choisir comme matrice de parité de $\mathcal{H}_3$ la matrice*

$$H_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

de manière que

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

est une matrice génératrice de $\mathcal{H}_3$ par la Proposition 2.2.

Exercice 3.1. *Montrer que la distance minimale de $\mathcal{H}_m$ est 3 pour tout $m \geq 3$.*

Par l'Exercice 3.1, tout code de Hamming est donc 1-correcteur. Ils sont parfait : en effet

$$1 + \binom{n}{1} = 1 + n = 2^m = 2^{(2^m-1)-(2^m-m-1)} = 2^{n-k}.$$

Exercice 3.2. *Considérons le code $\mathcal{H}_3$ avec la matrice génératrice donnée dans l'Exemple 3.1. Décoder $y = (0, 1, 1, 1, 1, 1, 0)$.*

Définition 3.2. *Les duaux des codes de Hamming $\mathcal{S}_m = \mathcal{H}_m^\perp$ sont appelés **codes simplexes**. Ces sont donc des $[2^m - 1, m]$ codes binaires, dont une matrice génératrice est H_m.*

On va déterminer la distribution des poids des codes simplexes et donc des codes de Hamming (graçe au Théorème 2.2).

Théorème 3.1. *Soit $n = 2^m - 1$. Les polynômes des poids des codes simplexes et des codes de Hamming sont*

$$w_{\mathcal{S}_m}(x, y) = x^n + nx^{\frac{n-1}{2}} y^{\frac{n+1}{2}}$$

et

$$w_{\mathcal{H}_m}(x, y) = \frac{1}{n+1}\left[(x+y)^n + n(x+y)^{\frac{n-1}{2}}(x-y)^{\frac{n+1}{2}}\right].$$

En particulier, les codes simplexes $\mathcal{S}_m$ sont des $[2^m - 1, m, 2^{m-1}]$ codes.

Démonstration. Par définition, $\mathcal{S}_m = \{xH_m \mid x \in \mathbb{F}_2^m\}$. Appellons $h^{(1)}, \ldots, h^{(n)}$ les colonnes de H_m. Si $x \neq 0$, alors

$$\mathrm{wt}(xH_m) = n - \#\{j \mid \langle x, h^{(j)} \rangle = 0\}.$$

Or, l'ensemble des vecteurs orthogonaux à x est un sous-espace vectoriel de $\mathbb{F}_2^m$ de dimension $m - 1$, qui contient donc $2^{m-1} - 1$ vecteurs non nuls. Ainsi,

$$\mathrm{wt}(xH_m) = n - (2^{m-1} - 1) = 2^{m-1} = 2^{\frac{n+1}{2}}.$$

Cela implique donc que tout mot non nul a poids 2^{m-1}, d'où la forme du polynôme des poids de $\mathcal{S}_m$. La forme du polynôme des poids de $\mathcal{H}_m$ descends directement en appliquant le Théorème 2.2 au polynôme $w_{\mathcal{S}_m}(x, y)$. □

Exercice 3.3. *Vérifier que*

$$w_{\mathcal{S}_3}(x,y) = x^7 + 7x^3y^4$$

et que

$$w_{\mathcal{H}_3}(x,y) = \frac{1}{8}\left[(x+y)^8 + 8(x+y)^3(x-y)^4\right] = x^7 + 7x^4y^3 + 7x^3y^4 + y^7.$$

Remarque 3.1. *Notons que le Théorème 3.1 nous permets de calculer facilement la distribution des poids de codes grandes sans devoir calculer la liste de ses mots. Par exemple, $\mathcal{H}_5$ est un $[31, 26, 3]$ code avec $2^{26} = 67108864$ mots avec polynôme des poids*

$$w_{\mathcal{H}_5}(x,y) = \frac{1}{32}\left[(x+y)^{31} + 31(x+y)^{15}(x-y)^{16}\right] = x^{31} + 155x^{28}y^3 + \dots$$

Donc, sans en faire la liste, on sait qu'il y a 155 mots de poids 3 en $\mathcal{H}_5$.

3.2 Codes de Reed-Muller

On peut introduire les codes de Reed-Muller à partir des codes de Hamming, à travers la construction vue ci-dessus.

Soit H_m une matrice de parité d'un code de Hamming $\mathcal{H}_m$. Considérons le code C_m engendré par

$$\left(\begin{array}{ccc|c} & & & 0 \\ & H_m & & \vdots \\ & & & 0 \\ \hline 1 & \dots & 1 & 1 \end{array}\right)$$

(i.e. le dual $\widehat{\mathcal{H}_m}^{\perp}$ du code de Hamming).

Exercice 3.4. *Montrer que $w_{C_m}(x,y) = x^{2^m} + (2^{m+1}-2)x^{2^{m-1}}y^{2^{m-1}} + y^{2^m}$.*

Exemple 3.2. *Pour $m = 3$ on obtient le $[8, 4, 4]$ code avec matrice génératrice*

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

avec polynôme des poids $x^8 + 14x^4y^4 + y^8$. On peut montrer que c'est un code auto-dual (exercice) et il est donc égal à $\widehat{\mathcal{H}_3}$.

Regardons la matrice G de l'Exemple 3.2. La première ligne est la première coordonnée de tous les vecteurs de $\mathbb{F}_2^3$ (dans l'ordre qu'on avait choisi pour H_3, plus le vecteur nul à la fin). Si on pense aux vecteurs de $\mathbb{F}_2^3$ comme des vecteurs $[x_1, x_2, x_3]$, la première ligne est donc égale au polynôme x_1 évalué sur tous les

vecteurs de $\mathbb{F}_2^3$. De la même façon, la deuxième ligne est égale au polynôme x_2 évalué sur tous les vecteurs de $\mathbb{F}_2^3$ (dans le même ordre), et la troisième ligne est égale au polynôme x_3 évalué sur tous les vecteurs de $\mathbb{F}_2^3$ (dans le même ordre). Dans la quatrième ligne, tout élément vaut 1 et on peut l'interpréter comme l'évaluation du polynôme constant 1 sur tous les vecteurs de $\mathbb{F}_2^3$. Puisqu'un mot du code $\widehat{\mathcal{H}_3}^\perp = \widehat{\mathcal{H}_3}$ est une combinaison linéaire des lignes de G (avec des coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \in \mathbb{F}_2$), suivant l'interprétation ci-dessus, on peut le voir comme l'évaluation de

$$p(x) = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \lambda_4$$

sur tous les vecteurs de $\mathbb{F}_2^3$. Cela peut être exprimé avec la notation suivante :

$$(p(v))_{v \in \mathbb{F}_2^3},$$

où il est sous-entendu que l'ordre est librement choisi mais fixe. Notons que $p(x)$ est un polynôme générique à trois indéterminées de degré au plus 1 sur $\mathbb{F}_2$ (pour simplifier la notation, dans ce cours nous incluons également le polynôme nul parmi les polynômes de degré au plus $r \in \mathbb{N}$). On a donc montré que

$$\widehat{\mathcal{H}_3}^\perp = \{(p(v))_{v \in \mathbb{F}_2^3} \mid p(x) \in \mathbb{F}_2[x_1, x_2, x_3] \text{ de degré au plus 1}\}.$$

Il est facile de voir que cela peut être généralisé, i.e. que

$$\widehat{\mathcal{H}_m}^\perp = \{(p(v))_{v \in \mathbb{F}_2^m} \mid p(x) \in \mathbb{F}_2[x_1, \ldots, x_m] \text{ de degré au plus 1}\}.$$

Ces codes sont des cas particuliers de codes de Reed-Muller.

Définition 3.3. *Soient m, r deux entiers avec $0 \leq r \leq m$. On appelle **code de Reed-Muller** d'ordre r en m le code*

$$\mathcal{RM}(r, m) = \{(p(v))_{v \in \mathbb{F}_2^m} \mid p(x) \in \mathbb{F}_2[x_1, \ldots, x_m] \text{ de degré au plus } r\}.$$

Comme pour les codes de Hamming, si on change l'ordre des colonnes, le code reste un code de Reed Muller : en vérité, $\mathcal{RM}(r, m)$ définit une classe de codes équivalents.

Exemple 3.3. *Si $r \leq 1$, on retrouve des codes qu'on connaît :*
- *$\mathcal{RM}(0, m)$ est un code de répétition.*
- *$\mathcal{RM}(1, m) = \widehat{\mathcal{H}_m}^\perp$.*

Clairement, tout code $\mathcal{RM}(r, m)$ a longueur 2^m. On veut comprendre quelle est la dimension de ces codes.

Définition 3.4. *On appelle **fonction booléenne** sur $\mathbb{F}_2^m$ toute application de $\mathbb{F}_2^m$ sur $\mathbb{F}_2$.*

Notons que toute application de $\mathbb{F}_2^m$ sur $\mathbb{F}_2$ est polynomiale (on le voit par exemple en utilisant l'interpolation lagrangienne), mais différents polynômes peuvent donner la même application (par exemple x_1^2 et x_1 donnent les mêmes valeurs si évalués sur $\mathbb{F}_2$). Pour obtenir une représentation unique, il faut considérer les éléments de l'anneau quotient

$$\mathbb{F}_2[x_1, \ldots, x_m]/(x_1^2 + x_1, \ldots, x_m^2 + x_m),$$

i.e. des polynômes où chaque indéterminée apparaît au degré au plus 1. Cette représentation s'appelle **forme algébrique normale**. Une base pour l'espace des fonctions booléennes est donnée par l'ensemble des monômes de la forme

$$\prod_{i \in I} x_i.$$

Notons que le degré de chaque monôme est donné par $\#I$. Le **degré** d'une fonction booléenne est le degré maximal des monômes de sa forme algébrique normale.

Exemple 3.4. *La forme algébrique normale de la fonction*

$$(x_1, x_2, x_3) \mapsto x_1^2 + x_1x_2 + x_2^3x_3 + x_1x_2^2x_3^3$$

est

$$(x_1, x_2, x_3) \mapsto x_1 + x_1x_2 + x_2x_3 + x_1x_2x_3,$$

d'où on voit que son degré est 3*, i.e. le degré de* $x_1x_2x_3$*.*

Avec ces notions, on peut voir facilement que

$$\mathcal{RM}(r, m) = \{(f(v))_{v \in \mathbb{F}_2^m} \mid f : \mathbb{F}_2^m \to \mathbb{F}_2 \text{ de degré au plus } r\}.$$

Cela nous permet de trouver facilement la dimension de $\mathcal{RM}(r, m)$. Voyons un exemple.

Exemple 3.5. *On a que*

$$\mathcal{RM}(2, 3) = \{(f(v))_{v \in \mathbb{F}_2^3} \mid f : \mathbb{F}_2^3 \to \mathbb{F}_2 \textit{ de degré au plus } 2\}.$$

Une fonction booléenne de degré au plus 2 *est une combinaison linéaire des monômes*

$$1, x_1, x_2, x_3, x_1x_2, x_2x_3, x_1x_3,$$

de manière que, en gardant le même ordre des vecteurs de $\mathbb{F}_2^3$ *de l'Exemple 3.2, une matrice génératrice pour* $\mathcal{RM}(2, 3)$ *est donnée par*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_1x_2 \\ x_2x_3 \\ x_1x_3 \end{matrix}.$$

Donc $\mathcal{RM}(2,3)$ est un $[8,7]$ code. On peut vérifier que c'est le code de parité de longueur 8.

Exercice 3.5. *Montrer que*

$$\dim \mathcal{RM}(r,m) = \sum_{i=0}^{r} \binom{m}{i}.$$

Notons qu'en particulier $\mathcal{RM}(m,m) = \mathbb{F}_2^m$.

Proposition 3.1. *Le dual d'un code de Reed-Muller est un code de Reed-Muller. En particulier*

$$\mathcal{RM}(r,m)^\perp = \mathcal{RM}(m-r-1,m).$$

Démonstration. Notons d'abord que

$$\dim \mathcal{RM}(r,m) + \dim \mathcal{RM}(m-r-1,m) = \sum_{i=0}^{r} \binom{m}{i} + \sum_{i=0}^{m-r-1} \binom{m}{i}$$

$$= \sum_{i=0}^{r} \binom{m}{i} + \sum_{i=0}^{m-r-1} \binom{m}{m-i} = \sum_{i=0}^{r} \binom{m}{i} + \sum_{j=r+1}^{m} \binom{m}{j} = 2^m$$

qui est la longueur, de manière que la dimension de $\mathcal{RM}(m-r-1,m)$ est celle du dual de $\mathcal{RM}(r,m)$. Il suffit donc de montrer que $\mathcal{RM}(m-r-1,m)$ est contenu dans $\mathcal{RM}(r,m)^\perp$: soit $w = (f(v))_{v\in\mathbb{F}_2^m}$ un mot de $\mathcal{RM}(m-r-1,m)$, i.e. supposons que f soit de degré au plus $m-r-1$. On doit montrer que x est orthogonal à tout mot de $\mathcal{RM}(r,m)$, i.e. à tout $c = (g(v))_{v\in\mathbb{F}_2^m}$ avec g de degré au plus r. Or

$$\langle w, c\rangle = \sum_{v\in\mathbb{F}_2^m} f(v)g(v) = \sum_{v\in\mathbb{F}_2^m} fg(v),$$

où fg est de degré au plus $m-1$, i.e. fg est une somme de monômes de degré inférieur à n. Soit $\prod_{i\in I} x_i$ l'un de ces monômes. Puisque $\#I < m$, il existe $j \in \{1,\dots,m\} - I$. On a

$$\sum_{v\in\mathbb{F}_2^m} \prod_{i\in I} x_i = \sum_{v_1\in\mathbb{F}_2} \cdots \sum_{v_n\in\mathbb{F}_2} \prod_{i\in I} x_i = \sum_{v_j\in\mathbb{F}_2} \left(\begin{array}{c} \text{quelque chose} \\ \text{qui ne dépend pas de } j \end{array} \right) = 0.$$

Donc $\langle w, c\rangle = 0$. □

Exercice 3.6. *Calculer le polynôme des poids de $\mathcal{RM}(2,4)$. En déduire que c'est un $[16,11,4]$ code.*

Proposition 3.2 (voir Chapter 13 of [2])**.** *La distance minimale d'un code $\mathcal{RM}(r,m)$ est 2^{m-r}.*

3.3 Codes de Golay binaires

On va introduire les codes de Golay binaires avec un approche simple d'Alan Turing. Soient C_1 et C_2 des codes avec matrices génératrices respectivement

$$G_1 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \text{ et } G_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Exercice 3.7. *Vérifier que C_1 et C_2 sont des codes $\mathcal{RM}(1,3)$ de paramètres $[8,4,4]$ et que $C_1 \cap C_2$ est le code de répétition de paramètres $[8,1,8]$.*

Définition 3.5. *Le **code de Golay binaire étendu** $\mathcal{G}_{24}$ est le code de longueur 24 défini par*

$$\mathcal{G}_{24} = \{(a+x \mid b+x \mid a+b+x) \in \mathbb{F}_2^{24} \mid a,b \in C_1,\ x \in C_2\}.$$

Exercice 3.8. *Montrer que $\mathcal{G}_{24}$ est un code linéaire, de dimension 12 et qu'il est auto-dual.*

Théorème 3.2 ([2])**.** *Le code $\mathcal{G}_{24}$ est un $[24,12,8]$ code auto-dual et c'est l'unique code avec ces paramètres, à équivalence près.*

Exercice 3.9 $(*)$**.** *En utilisant le Théorème 2.2, l'Exercice 2.13 et le fait que dans un code auto-dual binaire tout mot a poids pair (le monter en amont), montrer que*

$$w_{\mathcal{G}_{24}}(x,y) = x^{24} + 759x^{16}x^8 + 2576x^{12}y^{12} + 759x^8x^{16} + y^{24}.$$

(il peut être utile d'utiliser un logiciel de calcul formel).

En enlevant la même coordonnée à tout mot d'un code de Golay étendu $\mathcal{G}_{24}$, on obtient un $[23,12,7]$, qui est appelé **code de Golay** $\mathcal{G}_{23}$. On peut montrer que si on change la coordonnée enlevée les codes qu'on obtient sont tous équivalents.

Exercice 3.10. *Montrer que $\mathcal{G}_{23}$ est un code parfait.*

Chapitre 4

Exercices avec correction

Exercice 4.1. *Soit C le $[9, 4, 3]$ code dont les mots sont de la forme $(c_1, c_2, c_1 + c_2, c_3, c_4, c_3 + c_4, c_1 + c_3, c_2 + c_4, c_1 + c_2 + c_3 + c_4)$.*

a) *Est-ce que le mot $x := (1, 0, 1, 1, 1, 0, 0, 0, 1)$ appartient à C ? Justifier.*

b) *Quel est le mot y du code C le plus proche à x ? Justifier.*

c) *Est-ce que $z := (0, 0, 1, 0, 0, 1, 0, 0, 1)$ est dans $C^\perp$? Justifier.*

Solution :

a) Non. En effet $c_2 + c_4 = 1 \neq 0$.

b) C'est $y = (1, 0, 1, 1, 1, 0, 0, 1, 1)$ (on peut le trouver en remarquant que C est le code « carré » et utilisant la méthode de décodage vue pour ce code). En effet, puisque $d(x, y) = 1$ et la distance minimal de C est 3, il n'existe pas un autre mot y' du code C avec distance plus petite ou égale à 1 de x, car sinon $d(y, y') \leq d(y, x) + d(x, y') \leq 2$.

c) Oui. En effet,

$$\langle c, z\rangle = (c_1 + c_2) + (c_3 + c_3) + (c_1 + c_2 + c_3 + c_4) =$$

$$= 0c_1 + 0c_2 + 0c_3 + 0c_4 = 0$$

pour tout $c \in C$.

Exercice 4.2. *Montrer qu'un $[n, n-1, 2]$ code est forcement le code de parité de longueur n.*

Solution : Soit C un $[n, n-1, 2]$ code. Alors $C^\perp$ a longueur n, dimension $n - (n-1) = 1$. De plus, sa matrice génératrice est forcement

$$H := [1 \ \cdots \ 1],$$

car sinon, par la Proposition 2.3, C n'aurait pas distance minimale 2, en ayant des colonnes nulles. Ainsi, par la Proposition 2.2, on a qu'une matrice génératrice

de C est

$$G := \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \end{bmatrix},$$

ce qui implique que tout mot $(c_1, c_2, \ldots, c_n)$ est tel que

$$c_n = \sum_{i=1}^{n-1} c_i.$$

On a appelé cela code de parité.

Exercice 4.3. *Soit $C \subseteq \mathbb{F}_2^6$ un code avec matrice génératrice*

$$G := \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

a) *Donner une matrice génératrice, la dimension et la distance minimale de $C^\perp$. Justifier.*
b) *Montrer que la distance minimale de C est 3.*
c) *Est-ce que $C = C^\perp$? Justifier.*
d) *Est-ce que C est un code parfait ? Justifier.*

Solution :

a) Une matrice génératrice de $C^\perp$ est, par la Proposition 2.2,

$$H := \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

La dimension de $C^\perp$ est $3 = 6 - 3$ et sa distance minimale est 3, car dans G il n'y a pas de colonnes nulles ni deux colonnes idéntiques, de manière que la distance minimal est au moins 3, et il y a 3 colonnes linéairment dépendantes (par exemple les 3 dernières).

b) On observe que G est égale à H à une permutation de colonnes près. Donc C a la même distance minimale de $C^\perp$.

c) Non. Par exemple $c := (1, 0, 0, 0, 1, 1) \in C$ mais $\langle c, c \rangle = 0$, de manière que c n'est pas dans $C^\perp$.

d) Non. Dans ce cas, $e = 1$, $n = 6$, $k = 3$. On a donc

$$1 + 6 < 2^{6-3} = 8.$$

Exercice 4.4. *Soit C le code de longueur 4 sur $\mathbb{F}_3$ dont les mots sont de la forme $(c_1, c_2, c_1 + c_2, c_1 - c_2)$.*

a) *Écrire une matrice génératrice pour C.*
b) *Combien y a-t-il de mots dans C ? Justifier.*
c) *Quel sont les paramètres de C ? Justifier.*
d) *Est-ce que C est auto-dual ? Justifier.*
e) *Est-ce que C est MDS ? Justifier.*
f) *Est-ce que $v = (1, 1, 2, 1)$ appartient à C ? Si non, le corriger (c'est-à dire, trouver le mot du code le plus proche à v).*

Solution :

a) $(c_1, c_2, c_1 + c_2, c_1 - c_2) = c_1(1, 0, 1, 1) + c_2(0, 1, 1, 2)$, et les deux vecteurs sont linéairement indépendentes, de manière qu'une matrice génératrice pour C est
$$G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$
b) Tout mot est une combinaisons linéaire des deux vecteurs d'une base avec coefficients dans $\mathbb{F}_3$. Ainsi il y a $3^2 = 9$ mots.
c) La longueur est 4 et la dimension est 2. Une matrice de parité pour C est (par la Proposition 2.2)
$$H = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 1 & 2 & 0 & 2 \end{bmatrix},$$
qui n'a pas ni de colonnes nulles ni de colonnes qui sont l'une un multiple de l'autre. Donc, par la Proposition 2.3, la distance minimale est au moins 3. Puisque dans le code il y a de mots de poids 3 (par exemple les lignes de G), la distance minimale est 3.
d) Soient h_1 et h_2 les deux lignes de H. On a que $-h_1 - h_2$ est égale à la première ligne de G et $h_2 - h_1$ est égale à la deuxième ligne de G, de manière que le code engendré par H est C, qui est donc auto-dual.
e) Oui, parce que $3 = 4 - 2 + 1$.
f) Le vecteur $v = (1, 1, 2, 1)$ n'appartient pas à C. En effet, $v_4 = 1 \neq 0 = v_1 - v_2$. Le mot $c = (1, 1, 2, 0)$ a distance 1 de v et il est donc le plus proche (car la distance minimale est 3). Pour le trouver, on aurait pu utiliser la méthode de décodage par syndrome aussi.

Exercice 4.5. *Soit C un $[n, k, d]$ code avec matrice génératrice $G = [I_k|A]$, où I_k est la matrice identité de taille $k \times k$ sur $\mathbb{F}_2$. Supposons $k \geq 3$.*

a) *Montrer que $d \geq 3$ si et seulement si toute ligne de G a poids au moins 3 et toutes les lignes de A sont différentes.*
b) *Soit $A = U + I_k$, où U est la matrice de taille $k \times k$ sur $\mathbb{F}_2$ avec toute entrée égale à 1. Est-ce que le mot*
$$v = (1, \underbrace{0, \ldots, 0}_{2k-2 \text{ fois}}, 1)$$
appartient au code C ?

c) *Soit $A = U + I_k$ comme dans le point* b). *Le mot recu*

$$u = (\underbrace{0, \ldots, 0}_{k-1 \text{ fois}}, \underbrace{1, \ldots, 1}_{k+1 \text{ fois}})$$

n'appartient pas au code C. Le corriger (c'est-à dire, trouver le mot du code le plus proche à u).

Solution :

a) Puisque G est en forme systématique, $H = [A^T | I_{n-k}]$, par la Proposition 2.2. Par la Proposition 2.3, on a aussi que $d \geq 3$ si et seulement si toute colonne de H est non nulle et il n'y a pas de colonnes égales. Cette dernière condition équivaut au fait que les colonnes de A^T (qui sont les lignes de A) aient poids au moins 2 (pour être non nulles et différentes de celles de I_{n-k}) et qu'elles soient différentes, ce qui est équivalent au fait que toute ligne de G a poids au moins 3 et toutes les lignes de A sont différentes.

b) Toutes les lignes de $A = U + I_k$ sont différentes et toute ligne de G a poids $k = 1 + (k-1)$. Donc la distance minimale de C est 3 par a). Ainsi v, qui a poids 2, n'appartient pas à C.

c) Puisque $A^T = A$, on a que $H = [A | I_k]$. Ainsi Hu^T, qui est la somme des $k+1$ dernières colonnes de H, est égale à

$$Hu^T = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Cela est la dernière colonne de H, c'est-à-dire le syndrome de

$$e = (0, 0, \ldots, 0, 1),$$

de manière que la correction de u est

$$c = u - e = (\underbrace{0, \ldots, 0}_{k-1 \text{ fois}}, \underbrace{1, \ldots, 1}_{k \text{ fois}}, 0)$$

Exercice 4.6. *Soit C le code linéaire de longueur 5 sur $\mathbb{F}_5$ dont une matrice génératrice est*

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 & 3 \end{bmatrix}.$$

a) *Donner la définition de code linéaire et des ses paramètres. Quel sont les paramètres de C ? Justifier.*

b) *Combien y a-t-il de mots dans C ? Justifier.*

c) *Donner une matrice de parité pour C et calculer le syndrome de*

$$v = (0, 1, 2, 3, 4).$$

Est-ce que v appartient à C ? Si non, le corriger.

Solution :

a) Un code linéaire C de longueur n sur un corps fini K est un sous-espace de l'espace vectoriel K^n. Ses paramètres sont sa longueur n, sa dimension k et sa distance minimale, i.e.

$$d = \mathrm{d}(C) = \min_{c,c' \in C, c \neq c'} d(c, c').$$

Puisque une matrice génératrice est une matrice dont les lignes forment une base de C, on a que la longueur de C est 5 et sa dimension est 3. Pour calculer sa distance minimale, on calcule d'abord une matrice de parité : par la Proposition 2.2, on a que

$$H = \begin{bmatrix} 1 & 1 & 1 & 4 & 0 \\ 1 & 2 & 3 & 0 & 4 \end{bmatrix}$$

est une matrice de parité pour C. Cette matrice n'a pas de colonnes nulles, ni de couples de colonnes liés, donc la distance minimale est au moins 3 par la Proposition 2.3. Mais toute ligne de la matrice génératrice a poids 3, de manière que la distance minimale est exactement 3.

b) Dans C il y a $125 = 5^3$ mots.

c) Au point a) on a calculé une matrice de parité H. On peut donc calculer le syndrome de v avec H :

$$vH^T = (0, 4).$$

Il est diffént du vecteur nul. Donc v n'appartient pas à C. On peut remarquer facilement qu'il est égal à la dernière colonne de H. Ainsi

$$(0, 0, 0, 0, 1)H^T = vH^T.$$

Par conséquent, le mot corrigé est

$$v - (0, 0, 0, 0, 1) = (0, 1, 2, 3, 3).$$

Exercice 4.7. *Soit C un $[n, k, d]$ code sur $\mathbb{F}_2$ tel que $C \subseteq C^\perp$ (code auto-orthogonal).*

a) *Montrer tout mot de C a poids pair.*

b) *Montrer que $\mathbf{1} = (1, 1, \ldots, 1) \in C^\perp$.*

c) *Montrer que $\dim C \leq n/2$.*

d) *Montrer que si G est une matrice génératrice pour C, alors GG^T est égale à la matrice nulle $k \times k$.*

e) *En sachant que tout le mot non nul d'un code simplexe $\mathcal{S}_m$ a poids 2^{m-1}, montrer que, pour $m \geq 3$, $\mathcal{S}_m \subseteq \mathcal{H}_m$ ($\mathcal{H}_m$ est un code de Hamming).*

Solution :

a) Tout mot c doit être orthogonal à soi même. Or, sur $\mathbb{F}_2$ on a $1 \cdot 1 = 1$, de manière que $\mathrm{wt}(c) \equiv \langle c, c\rangle \bmod 2$. Ainsi, forcement le poids de c est pair.

b) Pour tout mot $c \in C$ on a

$$\langle c, \mathbf{1}\rangle \equiv \mathrm{wt}(c) \bmod 2$$

et donc $\langle c, \mathbf{1}\rangle = 0$ par le point a). Ainsi, par la définition du code dual, $\mathbf{1} \in C^\perp$.

c) $\dim C \leq C^\perp = n - \dim C$. Ainsi $2 \dim C \leq n$.

d) Le coefficient (i, j) de GG^T est égale au produit scalaire de la i-ème ligne de G fois la j-ème ligne de G. Puisque le code est auto-orthogonal, ce produit est nul.

e) Puisque $\mathcal{H}_m$ est le dual d'un code simplexe, on doit juste montrer que $\mathcal{S}_m$ est auto-orthogonal. Pour le faire, il suffit (par définition de code dual) de montrer que si x, y sont deux mots de $\mathcal{S}_m$, alors $\langle x, y\rangle = 0$. Sur $\mathbb{F}_2$ on a que $\langle x, y\rangle \equiv \#\{i | x_i = 1 \text{ et } y_i = 1\} \bmod 2$. En plus

$$\mathrm{wt}(x + y) = \mathrm{wt}(x) + \mathrm{wt}(y) - 2\#\{i | x_i = 1 \text{ et } y_i = 1\}.$$

Puisque $\mathrm{wt}(x+y), \mathrm{wt}(x)$ et $\mathrm{wt}(y)$ sont divisibles par 4, on a que $\#\{i | x_i = 1 \text{ et } y_i = 1\}$ est pair, de manière que $\langle x, y\rangle \equiv 0 \bmod 2$.

Exercice 4.8. *Soit $C \subseteq \mathbb{F}_2^5$ un code linéaire et soit*

$$w_C(x, y) := x^5 + 2x^4y + x^3y^2 + x^2y^3 + 2xy^4 + y^5 = (x^2 - xy + y^2)(x + y)^3$$

son polynôme énumérateur des poids.

a) *Quels sont les paramètres de C ? Justifier.*

b) *Déterminer le polynôme énumérateur des poids de $C^\perp$ (justifier). Quelle est donc la distance minimale de $C^\perp$?*

c) *Donner une matrice génératrice possible de C.*

Solution :

a) La longueur de C est 5, qui est le degré de $w_C(x, y)$.
On a $\#C = 1+2+1+1+2+1 = 8 = 2^{\dim C}$, de manière que $\dim C = 3$.
On a $A_1(C) = 2 \neq 0$, de manière que $\mathrm{d}(C) = 1$.
Donc C est un $[5, 3, 1]$ code

b) Par le Théorème 2.2 on a que

$$w_{C^\perp}(x, y) = \frac{1}{\#C} \cdot w_C(x + y, x - y)$$

$$= \frac{1}{8}((x + y)^2 + (x + y)(x - y) + (x - y)^2)(2x)^3 = x^5 + 3x^3y^2.$$

Donc $A_1(C^\perp) = 0$ et $A_2(C^\perp) = 3 \neq 0$, de manière que $\mathrm{d}(C^\perp) = 2$.

c) On a que $(1,1,1,1,1) \in C$, car $A_5(C) = 1$. De plus $A_1(C) = 2$ et on peut supposer que, à une permutation près, on a $(1,0,0,0,0), (0,1,0,0,0) \in C$. Ce trois vecteurs sont linéairement indépendants, de manière que

$$G := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

est une matrice génératrice possible de C.

Exercice 4.9. *Soit $n \geq 3$ un entier et soit C un $[n, k, n-k+1]$ code (i.e. C est un code MDS binaire).*

a) *Montrer que C admet une matrice génératrice G' en forme systématique.*
b) *Montrer que le poids de toute ligne de G' est $n-k+1$.*
c) *En déduire que $k \in \{1, n-1, n\}$ (i.e. C est MDS trivial).*

Solution :

a) Soit $G = [A|B]$ une matrice génératrice de C, où A est une matrice $k \times k$ et B une matrice $k \times (n-k)$. On sait, par la Remarque 2.5, que $\text{rang}(A) = k$, de manière que A est inversible. Soit $G' = A^{-1}G$. Multiplier à gauche pour une matrice inversible est équivalent à un changement de base de C, donc G' est une matrice génératrice de C. De plus, $G' = [I_k | A^{-1}B]$ est en forme systématique.
b) Toute ligne l de G' a au moins $k-1$ zéros, de manière que $\text{wt}(l) \leq n-(k-1)$. Or, $l \neq 0$ (en effet l est un vecteur d'une base de C) de manière que $\text{wt}(l) \geq \text{d}(C) = n-k+1$. Donc $\text{wt}(l) = n-k+1$.
c) Soit $k \geq 2$ et soient l_1 et l_2 les deux premières lignes de G'. On a donc

$$l_1 = (1, \underbrace{0, \ldots, 0}_{k-1 \text{ fois}}, \underbrace{1, \ldots, 1}_{n-k \text{ fois}}) \text{ et } l_1 = (0, 1, \underbrace{0, \ldots, 0}_{k-2 \text{ fois}}, \underbrace{1, \ldots, 1}_{n-k \text{ fois}}).$$

Ainsi $l_1 + l_2 = (1, 1, \underbrace{0, \ldots, 0}_{n-2 \text{ fois}}) \in C$.

Donc

$$\text{d}(C) = n-k+1 \leq 2 \Leftrightarrow k \geq n-1.$$

Exercice 4.10. *Soit $C \subseteq \mathbb{F}_2^{16}$ un code avec matrice génératrice*

$$G := \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

a) *Quels sont les paramètres de C ? Justifier.*
b) *Quels sont les paramètres de $C^\perp$? Justifier.*

c) *Est-ce que le mot* $x := (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0)$ *appartient à* C *? Justifier.*
d) *Quel est le mot* y *du code* C *le plus proche à* x *? Justifier.*

Solution :
a) La longueur de C est 16 et sa dimension est 5 (nombre de colonnes et de lignes de G réspectivement). On observe que $G = \left[\begin{array}{c|c} H_4 & \mathbf{0} \\ \hline \mathbf{1} & 1 \end{array}\right]$, de manière que C est un code $\mathcal{RM}(1,4)$. Sa distance minimale est donc 8. Ainsi C est un $[16,5,8]$ code.
b) Par la Proposition 3.1, on a que $C^\perp$ est un code $\mathcal{RM}(2,4)$, de manière que $C^\perp$ est un $[16,11,4]$ code.
c) Soit l_5 la dernière ligne de G. On a que $\mathrm{d}(l_5,x) = 1 < 8 = \mathrm{d}(C)$. Donc x n'appartient pas à C, par définition de distance minimale.
d) On a déjà observé que l_5 a distance 1 de x. S'il existait un autre $y \in C$ avec $\mathrm{d}(y,x) \leq 1$, alors on aurait $\mathrm{d}(y,l_5) \leq 1+1 = 2$, ce qui donne une contradiction. Donc l_5 est le mot le plus proche à x.

Exercice 4.11. *Soit* C *le code sur* $\mathbb{F}_7$ *dont une matrice génératrice est*

$$G := \begin{bmatrix} 1 & 0 & 0 & 0 & 6 & 2 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 & 2 & 5 \\ 0 & 0 & 0 & 1 & 5 & 6 \end{bmatrix}.$$

a) *Quelle est la longueur de* C *? Et sa dimension ? Combien y a-t-il de mots dans* C *? Justifier les réponses.*
b) *Montrer que la distance minimale de* C *est égale à* 3*.*
c) *Montrer que la distance minimale de* $C^\perp$ *est égale à* 5*.*
d) *Est-ce que* $v = (0,6,1,0,0,4)$ *appartient à* C *? Si non, le corriger (c'est-à dire, trouver le mot du code le plus proche à* v*).*

Solution :
a) La longueur de C est 6, qui est le nombre de colonnes de G, sa dimension est 4, qui est le nombre de lignes de G. Dans C il y a $7^4 = 2401$ mots, car la dimension est 4 et le corps est $\mathbb{F}_7$.
b) Une matrice de parité pour C est (par la Proposition 2.2)

$$H := \begin{bmatrix} 6 & 2 & 2 & 5 & 6 & 0 \\ 2 & 2 & 5 & 6 & 0 & 6 \end{bmatrix}.$$

On remarque qu'il n'y a pas de colonnes nulles, donc la distance minimale est au moins 2. On doit montrer qu'il n'existe pas deux colonnes linéairement dépendantes. Appellons $h_1, \ldots, h_6$ les colonnes de H. Clairement les couples h_i, h_6, pour $i \in \{1,\ldots,5\}$, et h_i, h_5, pour $i \in \{1,\ldots,4\}$, sont linéairement indépendantes. Les couples h_i, h_2, pour $i \in \{1,3,4\}$ le sont aussi, car tout multiple de h_2 a les coefficients identiques. Finalement,

on vérifie que le déterminant de $[h_1, h_3]$, $[h_1, h_4$ et $[h_3, h_4]$ est différent de 0 dans $\mathbb{F}_7$. Donc la distance minimale est au moins 3 par la Proposition 2.3. On cherche un mot de poids 3. N'importe quelle ligne de G (par exemple) a poids 3, donc la distance minimale est 3.

c) Puisque $3 = 6 - 4 + 1$, le code C est MDS. Par la Remarque 2.5, $C^\perp$ l'est aussi. Donc sa distance minimale est égale à $6 - 2 + 1 = 5$.

d) Le syndrome de v est $s = Hv^T = (0, 6)^T \neq (0, 0)^T$, donc v n'appartient pas à C. On remarque que s est la dernière colonne de H, de manière que $s = H(0, 0, 0, 0, 0, 1)^T$. Ainsi, le mot du code le plus proche à v est $(0, 6, 1, 0, 0, 3) = (0, 6, 1, 0, 0, 4) - (0, 0, 0, 0, 0, 1)$.

Exercice 4.12. *Soit C un code linéaire sur $\mathbb{F}_2$ dont le polynôme énumerateur des poids est*

$$w_C(x, y) = x^8 + 14x^4y^4 + y^8.$$

a) *Combien y a-t-il de mots dans C ? Justifier.*

b) *Quel sont les paramètres de C ? Justifier.*

c) *Trouver une matrice génératrice pour C (justifier). En déduire que C est forcement équivalent à $\mathcal{RM}(1, 3)$.*

d) *Trouver le polynôme énumerateur des poids de $C^\perp$. Quelle est la distance minimale de $C^\perp$?*

e) *Est-ce que $v = (1, 1, 1, 0, 1, 1, 1, 1)$ appartient à C ? Si non, le corriger (c'est-à dire, trouver le mot du code le plus proche à v).*

Solution :

a) Le nombre de mots dans C est la somme des coefficients de $w_C(x, y)$, par définition de polynôme de poids. Donc C a 16 mots.

b) Par définition de $w_C(x, y)$, la longueur de C est le degré de $w_C(x, y)$, i.e. 8. Soit k la dimension de C. On sait que $2^k = 16$, de manière que $k = 4$. Finalement, on a que $A_1(C) = A_2(C) = A_3(C) = 0$ et $A_4(C) = 14 \neq 0$, de manière que la distance minimale de C est 4. Donc C est un $[8, 4, 4]$ code.

c) Soient $c_1, c_2 \in C$ de poids 4, $c_1 \neq c_2$. Le poids de $c_1 + c_2$ est forcement 4 ou 8. Dans le premier cas,

$$\#(\mathrm{supp}(c_1) \cap \mathrm{supp}(c_2)) = 2 \tag{4.1}$$

où $\mathrm{supp}(c) = \{i \mid c_i \neq 0\}$. Dans le deuxième cas, $c_1 + c_2 = u = (1, \ldots, 1)$, qui est le seul mot de poids 8 (dans ce cas, c_1, c_2, u sont linéairement dépendantes). Sans perte de généralité, on cherche une matrice génératrice du code dont la dernière ligne est u. Donc les autres lignes ont poids 4 et elles doient satisfaire (4.1). Ainsi, à près d'équivalence, on a

$$G := \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

On peut observer qu'une telle matrice est de la forme

$$\begin{bmatrix} H_3 & \underline{0}^T \\ \underline{1} & 1 \end{bmatrix}.$$

qui est une matrice génératrice d'un code $\mathcal{RM}(1,3)$.

d) On sait que $\mathcal{RM}(1,3)$ est auto-dual. Ainsi $w_{C^\perp}(x,y) = w_C(x,y)$ et la distance minimale de $C^\perp$ est 4.

e) Le mot v a poids 7, donc il n'appartient pas à C. La distance entre v et u (défini ci-dessus) est 1, donc u est le mot du code le plus proche à v, car la distance minimale est 4.

Exercice 4.13. *Soit C le code linéaire sur $\mathbb{F}_3$ dont une matrice génératrice est*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 & 2 & 1 \end{bmatrix}.$$

a) *Quel sont les paramètres de C ? Justifier.*

b) *Combien y a-t-il de mots dans C ? Justifier.*

c) *Trouver le polynôme des poids $w_C(x,y)$ de C. Justifier.*

d) *En sachant que le coefficient de x^4y^2 dans $w_C(x,y)$ est 4, trouver les mots de poids minimal (non nuls) de C. Justifier.*

Solution :

a) La longueur de C est 6 (nombre de colonnes) et sa dimension est 4 (nombre de lignes). La matrice de parité de C est, par la Proposition 2.2,

$$H = \begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 0 \\ 1 & 2 & 2 & 1 & 0 & 2 \end{bmatrix}.$$

On utilise le résultat de la Proposition 2.3 : il n'y a pas de colonnes nulles, donc la distance minimale est au moins 2. On remarque que la troisième colonne est 2 fois la première (et la quatrième colonne est 2 fois la deuxième). Donc il y a (au moins) deux colonnes liées, de manière que la distance minimale est 2.

b) Dans C il y a $3^4 = 81$ mots, car 4 est la dimension de l'espace vectoriel et 3 la cardinalité du corps.

c) Puisque $\#C = 81$ est trop grande, nous pouvons plus facilement déterminer le polynôme des poids de $C^\perp$ et après utiliser le Théorème 2.2. Puisque H (déterminée dans le point a)) est la matrice génératrice de $C^\perp$, on a que les mots de $C^\perp$ sont

$$c_1 = (0,0)G = (0,0,0,0,0,0)$$

$$c_2 = (0,1)G = (1,2,2,1,0,2)$$

$$c_3 = (0,2)G = (2,1,1,2,0,1)$$

$$c_4 = (1,0)G = (1,1,2,2,2,0)$$
$$c_5 = (1,1)G = (2,0,1,0,2,2)$$
$$c_6 = (1,2)G = (0,2,0,1,2,1)$$
$$c_7 = (2,0)G = (2,2,1,1,1,0)$$
$$c_8 = (2,1)G = (0,1,0,2,1,2)$$
$$c_9 = (2,2)G = (1,0,2,0,1,1)$$

On a $\mathrm{wt}(c_1) = 0$, $\mathrm{wt}(c_5) = \mathrm{wt}(c_6) = \mathrm{wt}(c_8) = \mathrm{wt}(c_9) = 4$ et $\mathrm{wt}(c_2) = \mathrm{wt}(c_3) = \mathrm{wt}(c_4) = \mathrm{wt}(c_7) = 5$, de manière que

$$w_{C^\perp}(x,y) = x^6 + 4x^2y^4 + 4xy^5.$$

Ainsi, par le Théorème 2.2 (on utilise le fait que $(C^\perp)^\perp = C$), on a que

$$w_C(x,y) = \frac{1}{\#C^\perp} w_{C^\perp}(x+2y, x-y) =$$

$$= \frac{1}{9}((x+2y)^6 + 4(x+2y)^2(x-y)^4 + 4(x+2y)(x-y)^5).$$

d) Le coefficient de x^4y^2 dans $w_C(x,y)$ est 4, de manière qu'il y a 4 mots de poids 2, qui est le poids minimal pour le point a). Déjà dans le point a) on a observé deux couples de colonnes liées, ce qui nous donne les 4 mots de poids minimal :

$$(1,0,1,0,0,0), (2,0,2,0,0,0)$$

$$(0,1,0,1,0,0), (0,2,0,2,0,0).$$

Bibliographie

[1] J. P. Escofier. *Toute l'algèbre de la Licence*. 2éme édition. Dunod, 2006.

[2] F. J. MacWilliams et N. J. A. Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[3] W. C. Huffman et V. Pless. *Fundamentals of error-correcting codes*. Cambridge University Press, 2010.