# Table des matières

**1 Introduction à la théorie algébrique des codes correcteurs d'erreurs 1**
I Théorie élémentaire des codes correcteurs . . . 3
I.1 Formalisation mathématique . . . 3
I.2 Encodage . . . 4
I.3 Distance de Hamming . . . 6
I.4 Décodage par maximum de vraisemblance . . . 8
I.5 Décodage par syndrome . . . 9
II Constructions élémentaires de codes et bornes . . . 11
II.1 Constructions génériques à partir de codes donnés . . . 11
II.2 Constructions de familles de codes classiques . . . 13
II.3 Bornes . . . 17
III Quelques rappels sur les corps finis . . . 20
IV Codes cycliques et codes BCH . . . 22
IV.1 Théorie élémentaire des codes cycliques . . . 22
IV.2 Borne BCH . . . 25
IV.3 Codes BCH . . . 26
IV.4 Codes de Reed-Solomon cycliques . . . 27
V Décodage des codes BCH . . . 27
V.1 Algorithme de Peterson . . . 28
V.2 Décodage par l'algorithme d'Euclide étendu . . . 31
VI Codes de Reed-Solomon, codes alternants et codes de Goppa . . . 32
VI.1 Codes de Reed-Solomon et algorithme de Welch-Berlekamp . . . 32
VI.2 Codes alternants . . . 34
VI.3 Codes de Goppa binaires . . . 35
VII Décodage en liste des codes de Reed-Solomon . . . 39
VII.1 Introduction au décodage en liste . . . 39
VII.2 Algorithme de Sudan . . . 40
VIII Exercices . . . 42

**Partie I Solutions des tests 44**

**Partie II Solutions des tests 45**

**Partie III Solutions des exercices 47**

**Partie IV Solutions des exercices 48**

# Chapitre 1

# Introduction à la théorie algébrique des codes correcteurs d'erreurs

La théorie des codes correcteurs d'erreurs a été introduite dans les années 1940 par Richard Hamming qui a proposé la notion de codes correcteurs d'erreurs, puis par Claude Shannon qui a formalisé mathématiquement la théorie de l'information.

La théorie des codes a pour but, lors d'un envoi d'informations sur un canal sous forme digitale (une suite de bits), de protéger cette information contre des dégradations extérieures liées au canal. Plus précisément, si lors d'une transmission des bits sont modifiés, on souhaite être capable de les retrouver et de les corriger, cette dernière opération étant appelée décodage. Les applications sont très nombreuses puisque les codes correcteurs sont présents dans toute utilisation nécessitant la sécurisation d'informations contre une dégradation extérieure ; on peut citer par exemple les communications satellites, les modems, les CD ou DVD (rayures), la télévision haute définition, etc. Mais les codes correcteurs sont aussi liés à d'autres parties des mathématiques comme la combinatoire, la théorie des groupes ou la théorie des réseaux. Ainsi il s'agit d'un champs disciplinaire ayant beaucoup de connexions avec des domaines très variés, allant du plus théorique au plus pratique.

Concrètement, on procède par le schéma général (simplifié) suivant :


Une information (un message $M$) est considérée comme une suite de $k$ bits $\{0,1\}$ à laquelle on va ajouter une redondance de $r$ bits (encodage) pour former un mot du code. L'idée est que l'ajout de redondance permet de pallier des erreur (des bits à $1$ qui se transforment en $0$ ou inversement) lors de l'envoi du mot du code sur le canal.

Un exemple simple de code détecteur d'erreurs est le *code de parité* : à une séquence de bits, on ajoute un bit de redondance de sorte que la somme soit paire. De cette façon, si un bit est modifié, on est capable de le voir puisque la somme des bits n'est plus paire. On dit qu'il s'agit d'un code $1$-détecteur d'erreurs car on est capable de détecter simplement $1$ erreur au plus :


Dans ce cas, on est capable de deviner qu'il y a une erreur dans le message reçu, mais on ne peut pas savoir à quel endroit. Il est possible, comme on va le voir, de faire mieux et de corriger les erreurs.

Il existe divers types de canaux sur lesquels on peut envoyer une information. En fonction du contexte et du canal, on pourra modéliser le type d'erreurs de manière différente. Le canal le plus simple est le canal binaire symétrique, où l'on considère une probabilité d'erreur $p$ fixe de transformation de $0$ en $1$ ou de $1$ en $0$, et donc $1-p$ la probabilité qu'un bit ne soit pas modifié. On considère toujours une probabilité d'erreur $p < 0,5$.

En pratique, $p$ sera petit et le taux d'erreur que l'on voudra corriger dépendra des applications considérées. Par exemple, pour la transmission de la voix humaine, on tolérera un taux d'erreur

de $10^{-4}$ (une erreur tous les 10 000 bits). Pour beaucoup d'applications, on demande un taux d'erreurs de l'ordre de $10^{-6}$, mais cela peut descendre à $10^{-9}$ ou moins pour des applications très précises : par exemple, si l'on envoie un texte chiffré, on veut que la probabilité d'erreur soit très faible pour être capable de le déchiffrer. Inversement, dans des contextes liés à des attaques cryptographiques, on peut chercher à décoder des taux d'erreurs proches de $0,5$.

**Remarque.** En pratique, le canal binaire est rarement utilisé car il est peu réaliste. On utilise souvent une modélisation dite canal à bruit blanc additif gaussien (BBAG) (avec le même type de codes), dont l'exposition dépasse ce cours. Il existe aussi d'autres types de canaux correspondant à diverses modélisations comme le canal à effacement, qui modélise le fait que, sur Internet, il arrive que des paquets se perdent et donc que l'on obtienne des mots non plus avec des erreurs, mais avec des effacements (les parties manquantes), ou encore le canal à évanouissement, où les erreurs sur une coordonnée sont liées aux coordonnées adjacentes.

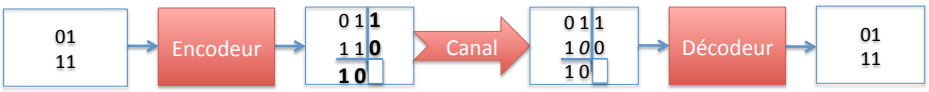
**Exemple 1.1.** Supposons une probabilité d'erreur de $0,1$ sur un canal binaire symétrique ; la probabilité d'envoyer 4 bits sans erreur est alors $0,9^4 \approx 0,65$, ce qui montre que la probabilité de ne pas avoir d'erreur sur un message décroît rapidement.

On peut remarquer que le problème de la correction d'erreurs peut être très facilement résolu par la répétition : ainsi chaque bit d'un message va être répété $k$ fois et l'on décodera un bit en prenant simplement la valeur qui apparaît le plus souvent (décodage à la majorité) :


De cette manière, quitte à augmenter le nombre de répétitions, on peut arriver à gérer totalement l'erreur. Mais cette solution est désavantageuse en termes de coût de communication (nombre de bits à envoyer pour un message donné). Soient $m$ le nombre de bits du message et $r$ la taille de la redondance en bits. La longueur du mot de code envoyé est alors $n = k + r$.

La théorie des codes consiste alors à trouver des codes tels que le nombre d'erreurs qui peut être corrigé est le plus grand possible avec à la fois un *taux de transmission* $k/n$ le plus élevé possible.

Voyons comment généraliser le principe du code de parité de façon à obtenir un code correcteur. On suppose que l'on a 4 bits à envoyer auxquels on va ajouter 4 bits de parité (deux en horizontal et deux en vertical) :




S'il y a une erreur (et une seule) dans la transmission, on peut retrouver sa position en calculant les bits de parité horizontaux et verticaux du message transmis. Ici, on mesure une erreur sur le second bit de parité horizontal et sur le second bit de parité vertical ; on en déduit que c'est (probablement) le quatrième bit du message originel qui a été transmis avec une erreur, et on le corrige donc.
On obtient ainsi un code de longueur 8 qui est capable de corriger une erreur. Pour ce code, on a $k = 4$, $r = 4$ et un taux de transmission de $\frac{k}{k+r} = 1/2$.

**Test 1.1.**

Montrer, en généralisant le point de vue du code de parité, que pour le code précédent, on est capable de corriger une erreur sur n'importe lequel des 8 bits.

Le taux de transmission du code précédent est $\frac{k}{k+r} = 1/2$. En fait, il est possible d'améliorer ce taux à $4/7$ tout en conservant la capacité à corriger une erreur.

*Premier code de Hamming.* Le code suivant a été introduit par R. Hamming en 1940. Il permet de corriger une erreur pour 4 bits d'information, mais a seulement une redondance de 3 au lieu de 4.

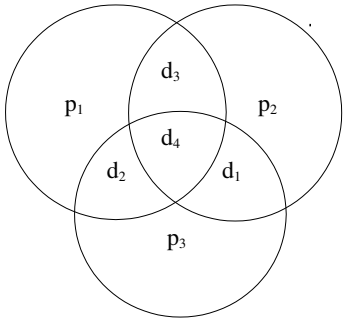



**Fig. 1.1. Premier code de Hamming.**

Supposons que l'information soit représentée par les bits $d_1, d_2, d_3, d_4$ ; on ajoute les bits de parité $p_1, p_2, p_3$ tels que les sommes $p_1 + d_2 + d_3 + d_4$, $d_1 + p_2 + d_3 + d_4$ et $d_1 + d_2 + p_3 + d_4$ soient paires.

**Test 1.2.**

Montrer que ce code peut corriger une erreur.

# I. Théorie élémentaire des codes correcteurs

## I.1. Formalisation mathématique

L'outil de base de la théorie des codes est la notion de corps finis. On pourra consulter le chapitre 4 du tome *Mathématiques L2*, Pearson Education, 2007, pour une introduction aux corps finis et le tome *Algèbre L3* chez le même éditeur, pour des approfondissements ; nous proposons un résumé dans la section III page 20.
Dans ce qui suit, on notera $\mathbb{F}_q$ un corps fini à $q$ éléments, où $q = p^r$ est une puissance d'un nombre premier $p$.

**Définition 1.2. (Code correcteur)** *Un code correcteur $\mathcal{C}$ sur $\mathbb{F}_q$ de longueur $n$ est un sous-ensemble de $\mathbb{F}_q^n$. Les éléments de $\mathcal{C}$ sont appelés des mots.*

**Définition 1.3. (Code linéaire)** *Un code linéaire $\mathcal{C}$ sur $\mathbb{F}_q$ de longueur $n$ et de dimension $k$ est un sous-espace vectoriel de $\mathbb{F}_q^n$ de dimension $k$. Un tel code est noté $[n, k]_q$.*

Dans ce qui suit, le terme *espace ambiant* désignera $\mathbb{F}_q^n$. La longueur du code est la dimension de l'espace ambiant.

**Définition 1.4. (Matrice génératrice)** *Une matrice génératrice d'un code linéaire $\mathcal{C}$ est une matrice dont les vecteurs lignes forment une base de l'espace vectoriel $\mathcal{C}$.*

Bien entendu, un code admet autant de matrices génératrices que de bases.

**Remarque.**

1. En théorie des codes, les vecteurs générateurs de la base sont notés en ligne (et non en colonne).
2. Pour les *codes binaires* (c'est-à-dire définis sur $\mathbb{F}_2$), on note simplement $[n, k]$, plutôt que $[n, k]_2$, un code de longueur $n$ et de dimension $k$.

## I.2. Encodage

Soit $\mathcal{C}$ un code $[n, k]_q$. Un message $x$ est considéré comme un élément de $\mathbb{F}_q^k$. L'encodage est donné par une application linéaire de $\mathbb{F}_q^k$ dans $\mathbb{F}_q^n$. L'image de $x$ par cette application est un mot du code $\mathcal{C}$, l'encodage de $x$.
Étant donné une matrice génératrice $G$ du code $\mathcal{C}$, l'encodage de $x$ est donné par le produit matrice-vecteur $x.G$ (où $x$ est noté en ligne et, par construction, les lignes de $G$ forment une base de $\mathcal{C}$).
L'encodage de $x$ dépend du choix de la matrice génératrice utilisée. Nous verrons plus loin comment l'on peut toujours faire en sorte que les $k$ premières colonnes de la matrice génératrice $G$ forment la matrice identité $k \times k$.

**Exemple 1.5.** Le code de parité binaire de longueur $n$ est l'ensemble des mots de $\mathbb{F}_2^n$ de somme paire, donc comportant un nombre pair de 1. On vérifie aisément que c'est bien un espace vectoriel.

En longueur 3, ce code a pour dimension 2. En effet, on connaît deux mots, linéairement indépendants, ayant un nombre pair de 1 : 110 et 011. Ils engendrent un code contenu dans le code de parité ; mais comme ce dernier ne peut être de dimension 3 (il n'est pas égal à $\mathbb{F}_2^3$), nos deux mots forment une base du code de parité. Cela dit, le code de parité peut avoir d'autres matrices génératrices comme

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{ou} \quad \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

**Test 1.3.**

Montrer que, généralement, le code de parité binaire de longueur $n$ a pour dimension $n-1$ et est un code $[n, n-1]$.

**Exemple 1.6.** Soit $\mathcal{C}_1$ le code de matrice génératrice

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{pmatrix}.$$

Le code $\mathcal{C}_1$ est composé de $2^3$ mots : $\mathcal{C}_1 = \{\sum_{i=1}^{3} a_i \mathbf{c}_i \quad | \quad a_i \in \mathbb{F}_2\}$.

**Test 1.4.**

Montrer que le code de Hamming défini plus haut est un code linéaire $[7,4]$. Donner une matrice génératrice de ce code.

**Définition 1.7. (Produit sur $\mathbb{F}_q^n$)** *Par analogie avec le produit scalaire euclidien usuel, on définit un produit sur $\mathbb{F}_q^n$. Pour $x, y \in \mathbb{F}_q^n$, on a*

$$x.y = \sum_{i=1}^{n} x_i y_i.$$

**Définition 1.8. (Code dual)** *Soit $\mathcal{C}$ un code $[n,k]_q$. On définit le dual $\mathcal{C}^\perp$ de $\mathcal{C}$ par*

$$\mathcal{C}^\perp = \{y \in \mathbb{F}_q^n | x.y = 0, \forall x \in \mathcal{C}\}.$$

Le code $\mathcal{C}^\perp$ est un code $[n, n-k]_q$.

**Définition 1.9. (Code autodual)** *Un code $\mathcal{C}$ est dit auto-orthogonal quand $\mathcal{C} \subset \mathcal{C}^\perp$ ; il est dit autodual quand $C = C^\perp$.*

**Définition 1.10. (Matrice de parité)** *Soit $\mathcal{C}$ un code $[n,k]_q$ de matrice génératrice $G$. On dit qu'une matrice $H$ est une matrice de parité (ou matrice de contrôle de parité, ou matrice duale) de $\mathcal{C}$ si $H$ est une matrice génératrice du dual $\mathcal{C}^\perp$.*

Comme le code dual est un espace vectoriel, il peut bien sûr avoir plusieurs matrices génératrices. Pour trouver le code dual, il suffit de résoudre le système $G.x^t = 0$ pour $x = (x_1, \cdots, x_n)$. L'ensemble des $x$ solutions forme le code dual.

**Test 1.5.**

Trouver une matrice génératrice pour le dual du code de matrice génératrice

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

**Test 1.6.**

Montrer que $x \in C \Leftrightarrow H.x^t = 0$.

Pour encoder un message, il peut être très intéressant d'avoir une forme particulière pour la matrice génératrice.

**Définition 1.11. (Matrice génératrice sous forme systématique)** *On dit qu'une matrice génératrice $G$ d'un code est sous forme systématique quand $G = [I_k, A]$, où $I_k$ désigne la matrice identité $k \times k$ et $A$ est une matrice $k \times n-k$.*

Par abus de langage, on dit qu'un code est mis sous forme systématique s'il est donné par une matrice génératrice qui est sous forme systématique.

**Proposition 1.12.** *Quitte à permuter des colonnes, tout code $\mathcal{C}$ peut être mis sous forme systématique.*

**Preuve.** Une matrice génératrice d'un code $[n,k]$ étant une base du code, en tant qu'espace vectoriel, il existe nécessairement une sous-matrice $k \times k$ inversible dans cette matrice génératrice. Puisque des combinaisons linéaires des lignes de la matrice génératrice ne modifient pas le code, il existe donc une matrice génératrice du code comportant une sous-matrice identité $k \times k$. On peut alors se ramener à une forme systématique par permutation des colonnes. ∎

**Proposition 1.13.** *Si $\mathcal{C}$ est un code $[n,k]_q$ de matrice génératrice $G = [I_k, A]$, alors sa matrice de parité est $H = [-A^t, I_{n-k}]$.*

**Preuve.** On vérifie que $G.{}^tH = 0$ et que $H$ a la dimension voulue. ■

## I.3. Distance de Hamming

**Définition 1.14. (Poids de Hamming)** *Le poids de Hamming d'un mot $x$ de $\mathbb{F}_q^n$ est le nombre de coordonnées non nulles de $x$. On le note $w_H(x)$ (ou simplement $w(x)$).*

Pour $x, y \in \mathbb{F}_2^n$, on définit le produit terme à terme par

$$x * y \overset{\text{déf}}{=} (x_1 y_1, \cdots, x_n y_n).$$

Par exemple, si $x = (101001)$ et $y = (110101)$, alors $x * y = (100001)$.

**Lemme 1.15.** *Pour $x, y \in \mathbb{F}_2^n$, le poids de Hamming de la somme $x + y$ est donné par*

$$w(x + y) = w(x) + w(y) - 2w(x * y),$$

*où $x * y$ désigne le produit terme à terme.*

**Test 1.7.**

Démontrer ce lemme. En déduire que, pour des mots binaires $x$ et $y$ de poids multiples de 4 et tels que le produit de $x$ et $y$ est nul, le poids de $x + y$ est aussi un multiple de 4.

**Définition 1.16. (Polynôme énumérateur de poids)** *Pour un code $\mathcal{C}$ de longueur $n$, on appelle polynôme énumérateur de poids le polynôme $W_{\mathcal{C}}(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i$, où $A_i$ désigne le nombre de mots de poids $i$ du code.*

**Définition 1.17. (Distance de Hamming)** *La distance de Hamming entre deux mots $x$ et $y$ est le nombre de lettres qui diffèrent entre $x$ et $y$, c'est-à-dire*

$$d_H(x, y) = w_H(x - y).$$

**Théorème 1.18.** *La distance de Hamming $d_H$ définit une distance sur $\mathbb{F}_q^n$.*

**Preuve.** Vérifions les trois axiomes des distances. Pour $x, y \in \mathbb{F}_q^n$, il est clair que $d_H(x, y) = d_H(y, x)$. Si $d_H(x, y) = 0$, alors toutes les coordonnées sont identiques et donc $x = y$. Soient $x, y, z \in \mathbb{F}_q^n$ ; montrons que $d_H(x, z) \leq d_H(x, y) + d_H(y, z)$. La distance entre deux mots correspond au nombre de coordonnées distinctes. Considérons la $i$-ième coordonnée ( $1 \leq i \leq n$) ; si $x_i$ et $z_i$ sont égaux, la contribution est nulle sur la $i$-ième coordonnée et la contribution de $d_H(x, y)$ et $d_H(x, y)$ sur cette coordonnée est nécessairement plus grande. Maintenant si $x_i \neq z_i$, alors $x_i \neq y_i$ ou $y_i \neq z_i$ (voire les deux) et donc la contribution de $d_H(x, y) + d_H(y, z)$ sur la $i$-ième coordonnée est au moins 1, ce qui prouve le résultat. ■

**Définition 1.19. (Distance minimale)** *La distance minimale* $d$ *d'un code* $\mathcal{C}$ *est définie par*

$$d(C) = \min_{x,y \in C, x \neq y} d_H(x,y).$$

**Lemme 1.20.** *Pour un code linéaire* $\mathcal{C}$*, la distance minimale* $d(C)$ *est égale au plus petit poids d'un mot (non nul) du code :*

$$d(C) = \min_{x \in \mathcal{C}, x \neq 0} w_H(x).$$

**Test 1.8.**

Démontrer ce lemme.

**Test 1.9.**

Calculer la distance minimale du code de Hamming $\mathcal{H}_7$ de matrice génératrice

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

**Définition 1.21.** *On appelle code* $[n,k,d]_q$ *un code* $\mathcal{C}$ *de longueur* $n$*, de dimension* $k$ *et de distance minimale* $d$ *sur* $\mathbb{F}_q$.

La notion de distance de Hamming est la notion fondamentale associée aux codes correcteurs ; c'est elle qui dote l'espace ambiant d'une structure d'espace métrique.

Au-delà de la simple égalité de deux codes, on peut se poser la question de transformations qui conservent cette notion de distance entre deux mots de l'espace. Remarquant que la permutation de coordonnées est une telle transformation, on peut introduire une notion d'équivalence liée aux permutations sur les $n$ coordonnées d'un code. Cette notion d'équivalence est bien sûr moins fine que l'égalité entre deux codes, mais elle est pertinente dans la mesure où elle conserve globalement la distance entre deux mots de l'espace, et donc donne les mêmes capacités de décodage pour un code donné.

**Définition 1.22. (Codes équivalents)** *Deux codes binaires* $\mathcal{C}_1$ *et* $\mathcal{C}_2$ *sont dits équivalents quand il existe une permutation* $P$ *des coordonnées telle que tout mot de* $\mathcal{C}_1$ *permuté par* $P$ *est un mot de* $\mathcal{C}_2$.

Comme la permutation des coordonnées d'un mot ne change pas son poids de Hamming, et que le polynôme énumérateur d'un code ne dépend que du poids des mots, on en déduit que deux codes équivalents ont le même polynôme énumérateur des poids. En revanche, la réciproque est fausse : deux codes peuvent avoir le même énumérateur des poids, mais être non équivalents.

Tout code est équivalent à un code qui peut être mis sous forme systématique. La notion d'équivalence peut être généralisée en ajoutant le fait de pouvoir multiplier les colonnes d'un code par un élément de $\mathbb{F}_q$ non nul.

**Définition 1.23.** *On appelle permutations monomiales l'ensemble des transformations obtenues en permutant les colonnes du code puis en multipliant les colonnes par des éléments non nuls du corps.*

Deux codes $\mathcal{C}_1$ et $\mathcal{C}_2$ sont dits monomialement équivalents quand il existe une permutation monomiale permettant de transformer $\mathcal{C}_1$ en $\mathcal{C}_2$.

**Définition 1.24.** *Le groupe d'automorphismes monomiaux d'un code* $\mathcal{C}$ *est l'ensemble des permutations monomiales laissant le code invariant.*

Dans le cas d'un code binaire, on parle simplement de groupe d'automorphismes et l'ensemble des permutations monomiales se limite aux permutations.

**Test 1.10.**

Montrer que les codes de matrices génératrices $\mathcal{C}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ et $\mathcal{C}_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$ sont distincts mais équivalents. Montrer que le groupe d'automorphismes de $\mathcal{C}_1$ est le groupe de permutations engendré par les trois permutations $(1,2)$, $(3,4)$ et $(1,3)(2,4)$.

## I.4. Décodage par maximum de vraisemblance

On se pose maintenant la question du décodage. Soit $x$ un mot d'un code $[n,k,d]_q$ ; ce mot est envoyé sur un canal. Lors de la transmission, des erreurs peuvent apparaître et on va recevoir $y = x + e$, où $e$ représente le vecteur d'erreurs reçu. Tous les mots du code sont à une certaine distance de $y$ (au plus $n$ bien sûr). La question se pose de savoir, à partir de $y$ (qui est un mot de l'espace ambiant), à quel mot du code transmis il peut correspondre. Si l'on suppose que l'erreur est a priori faible, il est naturel de postuler que le mot duquel on est parti est le mot du code $\mathcal{C}$ le plus proche de $y$.

**Définition 1.25. (Décodage au maximum de vraisemblance)** *Étant donné un code $\mathcal{C}$ de longueur $n$ et un mot $y \in \mathbb{F}_q^n$, on appelle décodage au maximum de vraisemblance de $y$ le fait d'associer à $y$ un mot du code $\mathcal{C}$ le plus proche, pour la distance de Hamming, de $y$.*

$$\mathtt{Decode}(y,\mathcal{C}) = x \quad \text{tel que} \quad d_H(x,y) = \min_{c\in\mathcal{C}} d_H(c,y).$$

Dans le cas où plusieurs mots du code sont à la même distance, on en retourne un quelconque dans cet ensemble.

**Théorème 1.26.** *Soit $\mathcal{C}$ un code $[n,k,d]_q$ ; il permet de corriger, par décodage à maximum de vraisemblance, au plus $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs de manière univoque.*

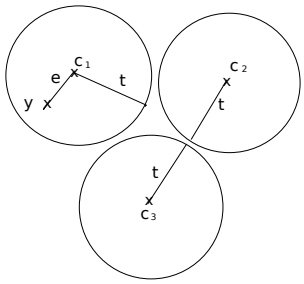



**Fig. 1.2. Décodage jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs.**

**Preuve.** On considère les boules de rayon $t = \lfloor \frac{d-1}{2} \rfloor$ centrées sur les mots du code $\mathcal{C}$ (les anglo-saxons parlent de *sphères*). Ce $t$ est le plus grand rayon tel que ces boules ne s'intersectent pas. Ainsi, à tout mot de l'espace inclus dans une des boules, on peut associer de manière univoque un mot du code : le centre de la boule. Ces mots de l'espace correspondent par définition exactement à tous les mots du code avec au plus $t$ erreurs. ■

**Test 1.11.** Montrer que ce résultat est aussi vrai pour des codes non linéaires.

## I.5. Décodage par syndrome

On va maintenant présenter une méthode générique de décodage des codes linéaires. Cette méthode, appelée *décodage par syndrome*, est optimale en ce sens qu'elle décode tout mot au maximum de vraisemblance : à tout mot de l'espace, on associe un mot de code le plus proche au sens de la distance de Hamming. Néammoins, elle a une complexité exponentielle et donc ne peut s'utiliser en pratique au-delà de codes de petites dimensions.

**Définition 1.27. (Syndrome)** *Soit $\mathcal{C}$ un code $[n, k, d]_q$ de matrice de parité $H$. Le syndrome d'un mot $x$ de $\mathbb{F}_q^n$ est le produit $H.x^t \in \mathbb{F}_q^{n-k}$.*

Notons que si $x \in \mathcal{C}$, alors son syndrome est nul (voir la définition 1.10 de la matrice de parité).

**Définition 1.28.** *Soient $\mathcal{C}$ un code $[n, k, d]_q$ et $a \in \mathbb{F}_q^n$. On définit le translaté de $a$ par $\mathcal{C}$ comme étant l'ensemble des mots $\{a + c | c \in \mathcal{C}\}$.*

Rappelons que l'on définit la relation $\sim_{\mathcal{C}}$ d'équivalence modulo un espace vectoriel $\mathcal{C}$ par

$$x \sim_{\mathcal{C}} y \iff x - y \in \mathcal{C}.$$

Le translaté de $a$ par $\mathcal{C}$ est sa classe d'équivalence modulo $\mathcal{C}$, ce qui induit les propriétés suivantes.

**Proposition 1.29.**

1. *Deux mots sont dans le même translaté si et seulement si ils ont le même syndrome.*
2. *Tous les translatés d'un code $\mathcal{C}$ ont le même nombre d'éléments $|\mathcal{C}|$.*
3. *Deux translatés sont soit disjoints soit égaux.*

**Preuve.** On a $x \sim_{\mathcal{C}} y$ si et seulement si $x - y \in \mathcal{C}$ ; mais cela est équivalent à avoir $H.(x+y)^t = 0$, c'est-à-dire que $x$ et $y$ ont même syndrome. Les deux propriétés suivantes viennent du fait que les translatés modulo $\mathcal{C}$ sont des classes d'équivalence pour $\sim_{\mathcal{C}}$. ∎

Les translatés de $\mathcal{C}$ forment une partition de $\mathbb{F}_q^n$. L'ensemble des translatés est l'ensemble quotient $\mathbb{F}_q^n / \sim_{\mathcal{C}}$, parfois noté aussi $\mathbb{F}_q^n / \mathcal{C}$.

**Définition 1.30. (Représentant principal)** *À tout translaté, on peut associer un mot de plus petit poids contenu dans le translaté. Ce mot est appelé représentant principal. Lorsqu'il existe plusieurs choix pour ce mot, on en prend un quelconque parmi les mots de même poids possibles.*

On peut alors créer une table de syndromes qui associe à chaque syndrome possible de $\mathbb{F}_q^{n-k}$ un représentant principal pour le translaté correspondant.

**Exemple 1.31.** Par exemple, si l'on prend le code $\mathcal{C}$ de matrice génératrice

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

et de matrice de parité

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

on peut construire la table de décodage par syndrome

| syndrome | représentant principal | mots du translaté - associés au représentant principal |
|---|---|---|
| $(000)^t$ | 00000 | 10111,01001,11110 |
| $(111)^t$ | 10000 | 00111,11001,01110 |
| $(001)^t$ | 01000 | 11111,00001,10110 |
| $(100)^t$ | 00100 | 10011,01101,11010 |
| $(110)^t$ | 11000 | 01111,10001,00110 |
| $(011)^t$ | 10100 | 00011,11101,01010 |
| $(101)^t$ | 01100 | 11011,00101,10010 |
| $(010)^t$ | 00010 | 10101,01011,11100 |

**Méthode** **Décodage par syndrome**

Pour un code de matrice de parité $H$, le décodage par syndrome consiste à décoder tout mot reçu $y$ en $y - cl(y)$, où $cl(y)$ désigne le représentant principal associé au syndrome $Hy^t$ de $y$.

**Théorème 1.32.** *Le décodage par syndrome est un décodage à maximum de vraisemblance.*

**Preuve.** Un tel décodage décode $y$ en un mot de $\mathcal{C}$ le plus proche de $y$. En effet, soit $c = y - cl(y) \in \mathcal{C}$ obtenu par décodage par syndrome de $y$ et soit $c_1 \in \mathcal{C}$, alors $d_H(y,c) \leq d_H(y,c_1)$ car $d_H(y,c) = w_H(y-c) = w_H(cl(y)) \leq w_H(y-c_1) = d_H(y,c_1)$ puisque $y - c$ et $y - c_1$ appartiennent au même translaté. ■

**Lemme 1.33.** *Le code $\mathcal{C}$ peut corriger jusqu'à $t$ erreurs si et seulement si tous les mots de poids au plus $t$ sont des représentants principaux. En particulier, si $\mathcal{C}$ est un code $[n,k,d]_q$, le décodage par syndrome permet de décoder au maximum de vraisemblance de manière univoque jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs.*

**Preuve.** Dans ce cas, le décodage par syndrome permet de retrouver de manière univoque tout mot du code auquel on ajoute une erreur de poids au plus $t$. Le cas où plusieurs représentants principaux (de même poids donc) sont possibles correspond au cas où l'on ne peut pas décoder de manière univoque et donc où l'on ne décode pas. On a vu précédemment que si $t \leq \lfloor \frac{d-1}{2} \rfloor$, il existait un mot de code unique le plus proche à distance $t$ et donc un unique représentant principal pour ce translaté. ■

**Exemple 1.34.** Reprenons le code de l'exemple précédent pour lequel on a construit une table des syndromes. Si l'on reçoit le mot $y = 00111$, on calcule le syndrome $s = (111)^t$ ; on va voir dans la table le représentant principal $(10000)$ et l'on décode $y$ en $y + (10000) = 10111$. Dans ce cas, on décode de manière univoque car il n'y a qu'un seul représentant principal possible. Si, par exemple, on avait obtenu comme syndrome $(011)^t$, il y aurait eu deux représentants principaux possibles et l'on aurait pu mal décoder. On remarque que pour ce code, la distance minimale est 2 et donc que le code ne décode pas nécessairement de manière univoque une erreur. Le théorème précédent montre en fait que dans le cas où le nombre d'erreurs à corriger est inférieur ou égal à $\lfloor \frac{d-1}{2} \rfloor$, il n'y a qu'un seul représentant principal pour les translatés correspondant aux erreurs de poids inférieur ou égal à $\lfloor \frac{d-1}{2} \rfloor$. Pour le cas où le nombre d'erreurs est strictement plus grand que $\lfloor \frac{d-1}{2} \rfloor$, il peut y avoir un ou plusieurs représentants principaux.

**Test 1.12.**

Reprendre la matrice précédente H. Décoder par syndrome le mot (11010). Montrer que pour un tel mot, le décodage est univoque. Montrer que le mot (10010) a deux chances sur trois d'être mal décodé avec cette table de syndromes.

Le décodage par syndrome est donc un algorithme qui permet de décoder au maximum de vraisemblance et de manière univoque jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs. Cette méthode donne une solution au problème du décodage, mais elle nécessite un nombre exponentiel d'opérations (de l'ordre de $\mathcal{O}(2^n)$) : il faut écrire la table des syndromes, ce qui est lourd dès que les paramètres grandissent.

La problématique de la théorie des codes correcteurs peut finalement être résumée à ce qui suit.

Synthèse **Problématique des codes correcteurs d'erreurs**

1. Construire de *bons* codes : des codes tels qu'à la fois le taux $k/n$ et $d$ soient les plus grands possibles. Avoir $k/n$ grand assure une redondance relativement faible ; avoir $d$ grand assure un pouvoir de correction important.
   Mais ces deux conditions sont antinomiques, car un taux $k/n$ très grand correspond intuitivement à un distance petite (ainsi pour $k = n$, on obtient $d = 1$ qui ne corrige rien). Inversement, si l'on veut un code avec un $d$ très grand, on peut prendre le code à répétition [111111...1] qui effectivement corrige très bien, mais correspond à une redondance trop importante.
2. Être capable de les décoder efficacement : idéalement en temps polynomial en $n$ (la longueur du code) et pas simplement par décodage par syndrome qui est exponentiel en $n$.

Le reste de ce chapitre permettra de donner des exemples de tels codes.

# II. Constructions élémentaires de codes et bornes

## II.1. Constructions génériques à partir de codes donnés

### II.1.1. Codes poinçonnés

Soit $\mathcal{C}$ un code $[n, k, d]_q$ donné par une matrice génératrice G. Le *code poinçonné* de $\mathcal{C}$ pour la $i$-ième colonne est obtenu à partir de $\mathcal{C}$ en enlevant la colonne $\{i\}$ à la matrice G.

Supposons que $d \geq 2$. Le code poinçonné sur la $i$-ième colonne est un code $[n-1, k-1, d']_q$, où la distance minimale $d'$ vérifie $d' = d - 1$ s'il existe un mot de $\mathcal{C}$ de poids $d$ avec une coordonnée non nulle sur la $i$-ième colonne, et $d' = d$ sinon.

### II.1.2. Extension de codes

Il existe plusieurs façons d'étendre un code en ajoutant une colonne aléatoire, mais une manière de procéder particulièrement intéressante consiste à ajouter à une matrice génératrice une colonne de parité, de telle sorte que chaque nouvelle ligne ainsi obtenue ait une somme nulle.

**Définition 1.35.** *Soit $\mathcal{C}$ un code $[n, k, d]_q$. Le code étendu $\hat{\mathcal{C}}$ est défini par*

$$\hat{\mathcal{C}} \stackrel{def}{=} \{(x_1, \cdots, x_n, x_{n+1}) \quad | \quad (x_1, \cdots, x_n) \in \mathcal{C} \text{ et } \sum_{i=1}^{n+1} x_i = 0\}.$$

Dans le cas binaire, si $d$ est impair, le code étendu a pour distance minimale $d+1$.

**Test 1.13.**

Montrer que le code de Hamming $\mathcal{H}_7$ $[7,4,3]$ donne comme code étendu un code $\mathcal{H}_8$ $[8,4,4]$. Donner l'énumérateur des poids du code étendu.

### II.1.3. Somme directe

On définit la *somme directe* de deux codes $\mathcal{C}_1[n_1,k_1,d_1]_q$ et $\mathcal{C}_2[n_2,k_2,d_2]_q$ par

$$\mathcal{C}_1 \oplus \mathcal{C}_2 \stackrel{déf}{=} \{(c_1|c_2) \quad | \quad c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\},$$

où $(c_1|c_2)$ désigne la concaténation des deux mots $c_1$ et $c_2$. Ce nouveau code est de longueur $n_1+n_2$. Il est engendré (comme espace vectoriel) par les mots de la forme $(b|0), b \in \mathcal{C}_1$ et $(0|\ b), b \in \mathcal{C}_2$, et il est donc de dimension $k_1+k_2$.
Une matrice génératrice de la somme directe des codes est la somme directe de deux matrices génératrices $G_i$ des $\mathcal{C}_i$, c'est-à-dire une matrice $(n_1+n_2)\times(n_1+n_2)$ diagonale par blocs avec les $G_i$ sur la diagonale.

La somme directe de deux codes satisfait la propriété intéressante suivante.

**Lemme 1.36.** *Le polynôme énumérateur des poids d'une somme directe de codes $\mathcal{C}_1 \oplus \mathcal{C}_2$ vérifie*

$$W_{\mathcal{C}_1\oplus\mathcal{C}_2}(x,y) = W_{\mathcal{C}_1}(x,y) * W_{\mathcal{C}_2}(x,y).$$

**Test 1.14.**

Démontrer ce lemme.

### II.1.4. Construction de Plotkin (ou construction $(u|u+v)$)

Étant donné deux codes $\mathcal{C}_1$ et $\mathcal{C}_2$ de même longueur, la *construction de Plotkin* est le code donné par

$$C = \{(u|u+v)|u \in \mathcal{C}_1, v \in \mathcal{C}_2\}.$$

En raisonnant comme pour la somme directe, on voit qu'il admet pour matrice génératrice

$$\begin{bmatrix} G_1 & G_1 \\ 0 & G_2 \end{bmatrix}.$$

Cette construction est particulièrement intéressante pour le cas d'un code binaire. On peut montrer par exemple que si $\mathcal{C}_1$ est le code de parité (vu dans la sous-section I.2) de longueur 4 (encore noté $\mathbf{1}_4{}^{\perp}$) et $C_2$ le mot tout à un de longueur 4, alors le code obtenu par cette construction est équivalent au premier code de Hamming étendu binaire $[8,4,4]$ vu dans l'introduction. Plus généralement, on a le résultat suivant pour les codes binaires.

**Proposition 1.37.** *Soient $\mathcal{C}_1[n,k_1,d_1]$ et $\mathcal{C}_2[n,k_2,d_2]$ deux codes binaires de même longueur. La construction de Plotkin produit un code $\mathcal{C}[2n, k_1+k_2, \min(2d_1,d_2)]$.*

**Preuve.** La longueur du code et la dimension découlent directement de la matrice génératrice du code construit.

Soit $d$ la distance minimale du code. Comme il existe des mots de poids $2d_1$ et $d_2$, on a $d \leq \min(2d_1, d_2)$. Soit $c \in \mathcal{C}$ un mot non nul s'écrivant $c = (c_1|c_1 + c_2)$ pour $c_i \in \mathcal{C}_i$. Le poids de Hamming de $c$ vérifie

$$w_H(c) = w_H(c_1) + w_H(c_1 + c_2) = w_H(c_1) + w_H(c_1) + w_H(c_2) - 2w_H(c_1 * c_2),$$

d'après le lemme 1.15, soit

$$w_H(c) = 2(w_H(c_1) - w_H(c_1 * c_2)) + w_H(c_2) \geq w_H(c_2)$$

puisque $w_H(c_1) - w_H(c_1 * c_2) \geq 0$. Donc, pour tout $c \in \mathcal{C}$ non nul, $w_H(c) \geq w_H(c_2) \geq \min(2d_1, d_2)$ et donc $d \geq \min(2d_1, d_2)$, ce qui montre le résultat. ■

## II.2. Constructions de familles de codes classiques

### II.2.1. Codes de Hamming

On peut généraliser la construction simple de Hamming que nous avions vue, en petite longueur, page 3. On commence par prouver le lemme suivant.

**Lemme 1.38.** *Un code linéaire $\mathcal{C}$ a une distance minimale $d$ si et seulement si sa matrice de parité $H$ a $d$ colonnes dépendantes et aucun ensemble d'au plus $d-1$ colonnes indépendantes, et aucun sous-ensemble d'au plus $d-1$ colonnes qui soient linéairement dépendantes.*

**Preuve.** Si $\mathcal{C}$ a une distance minimale $d$, alors il existe un mot $c$ de poids $d$ tel que $Hc^t = 0$, que l'on peut aussi voir comme une relation de dépendance entre $d$ colonnes de $H$ puisque $c$ est de poids $d$.
Réciproquement, s'il n'existe aucun ensemble d'au plus $d-1$ colonnes indépendantes, il n'existe pas de vecteur $x$ de poids au plus $d-1$ tel que $Hx^t = 0$ et donc, de fait, pas de mot de poids inférieur ou égal à $d-1$ dans le code. ■

Le code de Hamming se définit à partir de sa matrice de parité.

**Définition 1.39. (Code de Hamming)** *Pour un entier $r$ positif non nul, on construit une matrice $\mathcal{H}_r$ à $2^r - 1$ lignes et $r$ colonnes, dont les lignes sont les éléments non nuls de $\mathbb{F}_2^r$.*
*On appelle code de Hamming binaire d'ordre $r$ le code admettant $\mathcal{H}_r$ pour matrice de parité.*

**Lemme 1.40.** *Un code de Hamming binaire d'ordre $r$ a pour paramètres $[2^r - 1, 2^r - r - 1, 3]$.*

**Preuve.** La longueur et la dimension du code dérivent des paramètres de $\mathcal{H}_r$. Maintenant, par construction, on ne peut trouver de relation de dépendance entre deux colonnes de $\mathcal{H}_r$ (comme nous travaillons sur $\mathbb{F}_2$, cela signifierait l'égalité de deux colonnes), et donc d'après le lemme précédent, le code a pour distance minimale au moins 3. Comme on peut construire facilement une relation de dépendance entre deux colonnes, par exemple $(10..0) + (01..0) = (110..0)$, le code a exactement 3 pour distance minimale. ■

La famille des codes de Hamming constitue le premier exemple de famille infinie pouvant corriger une erreur puisque $d = 3$ implique $[(3-1)/2] = 1$ d'après le théorème 1.26.

**Test 1.15.**

Essayons de généraliser la construction de Hamming sur $\mathbb{F}_3$. On considère le corps $\mathbb{F}_3$ et $m = 2$. Plutôt que de prendre comme colonnes du code tous les mots non nuls de $\mathbb{F}_3^2$, on prend tous les mots non nuls, mais à une constante multiplicative près. Par exemple, pour $(10)$ et $(20)$, on ne prendra qu'un seul représentant. Montrer que l'on obtient alors un code $[4, 2, 3]_3$.

### II.2.2. Codes de Golay

Le code de Golay binaire a été introduit par M. J. E. Golay en 1949 sous forme de séquence pour les radars ; il peut être produit par de multiples constructions. Nous présentons ici une méthode simple, due à V. Pless, qui permet aussi de prouver la distance minimale du code *à la main*. On montrera ultérieurement que ce code est très bon.

Soit $A$ une matrice de taille $11 \times 11$ construite de la façon suivante : on indice les colonnes de $0$ à $10$ pour que les coordonnées $x_i (0 \leq i \leq 10)$ de la première ligne valent $1$ si $i$ est un carré modulo $11$ et $0$.

On déduit les $10$ lignes suivantes par permutations circulaires vers la gauche. On considère alors la matrice $12 \times 12$ bordée $\tilde{A}$ construite à partir de $A$, en ajoutant une colonne à gauche de $1$ et une ligne en haut de $1$, sauf dans le coin supérieur gauche, où l'on met $0$. On obtient comme matrice

$$\tilde{A} = \begin{bmatrix} 011111111111 \\ 111011100010 \\ 110111000101 \\ 101110001011 \\ 111100010110 \\ 111000101101 \\ 110001011011 \\ 100010110111 \\ 100101101110 \\ 101011011100 \\ 110110111000 \\ 101101110001 \end{bmatrix}.$$

**Définition 1.41. (Code de Golay $\mathcal{G}_{24}$)** *Soit $\tilde{A}$ la matrice définie ci-dessus. Le code $\mathcal{G}_{24}$ de matrice génératrice $G \overset{\text{déf}}{=} [I_{24}|\tilde{A}]$ est appelé code de Golay binaire étendu $\mathcal{G}_{24}$.*

**Théorème 1.42.** *Le code de Golay binaire étendu $\mathcal{G}_{24}$ a pour paramètres $[24, 12, 8]$.*
*On obtient le code de Golay $[23, 12, 7]$ en poinçonnant une colonne du code précédent.*

**Preuve.** La preuve se déroule en plusieurs étapes. Tout d'abord, on vérifie facilement que le code $\mathcal{G}_{24}$ est autodual. En effet, on commence par remarquer que le produit de la première ligne de $G$ avec une ligne quelconque de $G$ est nul, puis que le produit de la seconde ligne de $G$ avec toutes les autres est nul et donc que, par cyclicité, le produit de deux lignes quelconques de $G$ est nul. La remarque précédente montre que le code $\mathcal{G}_{24}$ est inclus dans son dual. Comme la dimension de $\mathcal{G}_{24}$ est $12$ et que la dimension du dual est $n - k = 24 - 12 = 12$, le code est bien égal à son dual, donc il est autodual.
Maintenant, comme le poids de chaque ligne de $G$ est un multiple de 4 et que le code est autodual, on en déduit par le test 1.7 que tous les mots de $\mathcal{G}_{24}$ ont un poids multiple de 4. Étant donné qu'il existe un mot de poids $8$ (par exemple la deuxième ligne de $G$), la distance minimale du code est soit 4 soit $8$.
La matrice $\tilde{A}$ est symétrique donc, par la proposition 1.13, le code dual de $\mathcal{G}_{24}$ admet $G' \overset{\text{déf}}{=} [\tilde{A}|I_{24}]$ pour matrice génératrice. Ainsi, comme $\mathcal{G}_{24}$ est autodual, il peut s'écrire avec les deux matrices génératrices $G$ ou $G'$. On va maintenant montrer que le code ne peut contenir de mot

de poids 4 et donc qu'il a bien 8 pour distance minimale.
Soit $c$ un mot du code de poids 4. On peut écrire $c = (a, b)$, où $a$ et $b$ sont deux vecteurs de longueur 12. Les possibilités pour $w_H(a)$ et $w_H(b)$ sont donc $(0,4), (1,3), (2,2), (3,1)$ ou $(4,0)$. Par symétrie entre $G$ et $G'$ (et donc entre $a$ et $b$), on a uniquement à considérer les trois premières possibilités. Le cas $w_H(a) = 0$ n'est pas possible car on obtiendrait le vecteur nul ; le cas $w_H(a) = 1$ correspond à la matrice $G$ et donc on vérifie qu'il n'y a pas de mot de poids 4. Maintenant pour le cas $w_H(a) = 2$, obtenu par somme de deux lignes de $G$, on peut vérifier en sommant la première ligne de $G$ à toutes les autres que le poids du mot obtenu est au moins 8. On vérifie aussi que la somme de la seconde ligne avec toutes les autres a pour poids 8. La cyclicité de $A$ assure alors que, plus généralement, la somme de deux lignes de $G$ a pour poids 8 et donc que le cas $w_H(a) = 2$ n'est pas possible, ce qui assure que $d = 8$ pour ce code.

Si l'on poinçonne $\mathcal{G}_{24}$ sur une colonne quelconque, les matrices génératrices $G$ et $G'$ montrent que nécessairement on va amputer une coordonnée non nulle d'un mot de poids 8 et que l'on va obtenir un code $[23, 12, 7]$. ■

Le code $[23, 12, 7]$ construit ci-dessus est appelé code de Golay binaire.

**Test 1.16.**

Soit $a = (a_0, \cdots, a_{n-1})$. On construit la matrice $A = (a_{ij})_{i,j=0\ldots n-1}$ de taille $n \times n$ où chaque ligne est obtenue à partir de la précédente par permutation circulaire sur la *gauche*. Montrer que $A$ est symétrique.

### II.2.3. L'hexacode

Soit le corps à 4 éléments $\mathbb{F}_4 = \{0, 1, \omega, \omega^2\}$ tel que $\omega^3 = 1$ et $1 + \omega + \omega^2 = 0$.
On appelle hexacode le code $\mathcal{H}_6$ de matrice génératrice

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & \omega & \omega^2 \\ 0 & 0 & 1 & 1 & \omega^2 & \omega \end{bmatrix}.$$

**Test 1.17.**

Montrer que $\mathcal{H}_6$ est un code $[6, 3, 4]_4$.

### II.2.4. Codes de Reed-Muller

Les codes de Reed-Muller ont été introduits à l'origine par D. E. Muller en 1954, puis Irving S. Reed a donné une méthode de décodage la même année. Ces codes, de longueurs une puissance de deux, ont été la première famille de codes pour lesquels on a pu décoder un nombre infini d'erreurs. Ils ont été utilisé pour la sonde Mariner entre 1969 et 1973 pour transmettre des photos de la planète Mars. Là encore, plusieurs constructions sont possibles. Nous donnons ici une construction matricielle simple basée sur la construction de Plotkin vue ci-dessus.

Soit $m$ un entier positif. Les codes de Reed-Muller sont des codes binaires de longueur $2^m$, indicés par un paramètre $0 \leq r \leq m$. On note $\mathcal{R}(r, m)$ le code de Reed-Muller de longueur $2^m$ et d'indice $r$.

**Définition 1.43. (Codes de Reed-Muller)** *Les codes de Reed-Muller* $\mathcal{R}(r,m)$ *sont définis par récurrence comme*

$$\mathcal{R}(r,m) = \{(u|u+v)|u \in \mathcal{R}(r,m-1), v \in \mathcal{R}(r-1,m-1)\},$$

*où* $\mathcal{R}(0,m) = (11\cdots1)$*, le code associé au mot tout à* 1 *de longueur* $2^m$*, et* $\mathcal{R}(m,m) = \mathbb{F}_2^{2^m}$*, de matrice génératrice la matrice identité de dimension* $2^m$*.*

Soit $G(r,m)$ une matrice génératrice de $\mathcal{R}(r,m)$. On a alors

$$G(r,m) = \begin{bmatrix} G(r,m-1) & G(r,m-1) \\ 0 & G(r-1,m-1) \end{bmatrix}.$$

**Exemple 1.44.** Le code $\mathcal{R}(1,2)$ construit à partir des codes $\mathcal{R}(0,1)$ et $\mathcal{R}(1,1)$ de matrices génératrices respectives [11] et $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ a pour matrice génératrice

$$\left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 \end{array}\right]$$

et est un code $[4,3,2]$.

**Test 1.18.**

Montrer que le code $\mathcal{R}(1,3)$ est un code $[8,4,4]$.

**Théorème 1.45.** *Soient* $m$ *et* $r$ *deux entiers positifs, avec* $0 \leq r \leq m$*.*

1. *Pour* $0 \leq i \leq j \leq m$*, on a* $\mathcal{R}(i,m) \subset \mathcal{R}(j,m)$*.*
2. *La dimension de* $\mathcal{R}(r,m)$ *est* $\sum_{i=0}^{r} \binom{m}{i}$ *(où* $\binom{m}{i}$ *désigne le coefficient binomial* $C_m^i$*).*
3. *La distance minimale de* $\mathcal{R}(r,m)$ *est* $2^{m-r}$*.*

**Preuve.** Le premier point se montre par une récurrence immédiate à partir de la construction. Le point 2) s'obtient aussi par récurrence à partir de la construction et de la propriété sur les coefficients binomiaux du triangle de Pascal : $\binom{n+1}{k+1} = \binom{n}{k} + \binom{n}{k+1}$.
Pour le point 3), on utilise le résultat sur la distance liée à la construction de Plotkin (proposition 1.37). La propriété est vraie pour $m = 1$. Supposons qu'elle soit vraie jusqu'à $m-1$ pour tout $r$. Soit le code $\mathcal{R}(r,m)$ ; ce code est obtenu par la construction de Plotkin à partir de deux codes $\mathcal{R}(r,m-1)$ et $\mathcal{R}(r-1,m-1)$ de distance minimale respective $2^{m-1-r}$ et $2^{m-1-(r-1)} = 2^{m-r}$ par hypothèse de récurrence. Nous avons vu à la proposition 1.37 que la distance minimale du nouveau code était $\min(2.2^{m-1-r}, 2^{m-r}) = 2^{m-r}$, ce qui montre le résultat. ■

**Test 1.19.**

Écrire une matrice génératrice du code $\mathcal{R}(1,4)$. Montrer que c'est un code $[16,5,8]$.

**Remarque.** Pour la sonde Mariner, c'est le code $\mathcal{R}(1,5)$ $[32,6,16]$ qui corrige 7 erreurs qui a été utilisé.

## II.3. Bornes

Il existe plusieurs type de bornes pour les codes ; nous présentons ici les bornes principales.

### II.3.1. Borne de Hamming

**Théorème 1.46. (Borne de Hamming)** *Soient $n$ et $d$ deux entiers positifs et $t = \lfloor \frac{d-1}{2} \rfloor$. Soit $A_q(n,d)$ (respectivement $B_q(n,d)$) le nombre maximum de mots que peut contenir un code (respectivement un code linéaire) de longueur $n$ et de distance minimale $d$, alors*

$$B_q(n,d) \leq A_q(n,d) \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i} (q-1)^i}.$$

**Preuve.** Soit un code $\mathcal{C}$ de longueur $n$ et de distance $d$ avec $A_q(n,d)$ mots. Les boules $B(c,t)$ centrées sur les mots $c$ du code et de rayon $t$ sont disjointes (car sinon il existerait deux mots à distance strictement inférieure à $d$) et donc la réunion de toutes les boules disjointes $B(c,t)$ est un ensemble d'éléments de l'espace contenu dans $\mathbb{F}_q^n$. Les éléments d'une boule sont composés des mots à distance $0$ du centre, ainsi que des mots à distance $1$, jusqu'à la distance $t$. Le nombre de mots à distance $0$ est $1$, le nombre de mots à distance $1$ est $n(q-1) = \binom{n}{1}(q-1)$. Plus généralement, le nombre de mots à distance $i$ vaut $\binom{n}{i}(q-1)^i$, le nombre de choix pour l'emplacement de l'erreur fois le nombre de possibilités pour l'erreur en une coordonnée donnée. On obtient donc $A_q(n,d)$ boules disjointes avec le même nombre d'éléments, ce qui donne $A_q(n,d)\sum_{i=0}^{t} \binom{n}{i}(q-1)^i \leq q^n$. De plus, on a trivialement $B_q(n,d) \leq A_q(n,d)$, ce qui donne le résultat. ∎

**Définition 1.47. (Code parfait)** *Un code qui vérifie l'égalité dans le théorème précédent est appelé un code parfait.*

Le fait qu'un code soit parfait correspond au fait que, pour un tel code, *tous* les mots de l'espace sont décodables de manière univoque, c'est-à-dire que tous les mots de l'espace sont contenus dans la réunion des boules de rayon $t = \lfloor \frac{d-1}{2} \rfloor$ centrées sur les mots du code (ce qui n'est pas du tout le cas en général).

**Exemple 1.48.** Le code de Hamming binaire $[7,4,3]$ est parfait car $2^4 = \frac{2^7}{1+\binom{7}{1}}$. On en déduit que $A_2(7,3) = 16$ car il y a une borne sur le nombre de mots maximum et un code qui l'atteint.

**Test 1.20.**

Montrer que tous les codes de Hamming sont parfaits.

**Test 1.21.**

Un code de distance minimale paire $2r$ peut-il être parfait ?

Plus généralement, on appelle *rayon d'empilement* $r_p(\mathcal{C})$ d'un code $\mathcal{C}$ le plus grand rayon tel que toutes les boules de rayon $r_p(\mathcal{C})$ centrées sur des mots de codes sont disjointes. Pour un code linéaire $[n,k,d]_q$, on a $r_p = \lfloor \frac{d-1}{2} \rfloor$.
On peut alors aussi définir naturellement le problème dual.

**Définition 1.49. (Rayon de recouvrement)** *Le rayon de recouvrement $r_c(\mathcal{C})$ d'un code est le plus petit rayon $r$ tel que la réunion des boules $B(c,r)$ (pour $c$ parcourant tous les mots du code) recouvre tout l'espace.*

**Test 1.22.** Montrer que $r_e(\mathcal{C}) \leq r_c(\mathcal{C})$. Quand sont-ils égaux ?

### II.3.2. Borne de Singleton

**Théorème 1.50. (Borne de Singleton)** *Soit $\mathcal{C}$ un code $[n, k, d]_q$. La distance minimale $d$ du code $\mathcal{C}$ vérifie la majoration $d \leq n - k + 1$.*

**Preuve.** À permutation près, le code $\mathcal{C}$ peut s'écrire sous la forme systématique $G = [I_k|A]$, où $A$ est une matrice $k \times n - k$. En notant $a_1$ la première ligne de $A$, le poids de la première ligne de $G$ est donc $1 + w_H(a_1) \leq 1 + (n - k)$. Par définition, on a $d \leq w_H(c)$ pour tout mot du code, d'où le résultat. ■

**Définition 1.51. (Codes MDS)** *Les codes satisfaisant l'égalité dans le théorème précédent sont dits codes à distance maximum séparable (MDS).*

On peut citer comme exemple non trivial, l'hexacode $[6, 3, 4]_4$ sur $\mathbb{F}_4$ vu dans la section précédente. Un autre exemple de codes MDS est la famille des codes de Reed-Solomon que l'on verra plus tard.

### II.3.3. Borne de Gilbert-Varshamov

Les bornes vues précédemment sont des bornes supérieures. Nous montrons ici la borne de Gilbert-Varshamov introduite en 1952, dans des contextes différents et séparément par Gilbert et Varshamov.

On aura besoin du lemme suivant. Rappelons que l'on note $A_q(n, d)$ le nombre maximal possible de mots pour un code sur $\mathbb{F}_q$ de longueur $n$ et de distance minimale $d$.

**Lemme 1.52.** *Soit $\mathcal{C}$ un code linéaire de longueur $n$, de distance minimale $d$ et tel que $\mathcal{C}$ contient $A_q(n, d)$ mots. Le rayon de recouvrement de $\mathcal{C}$ est alors au plus $d - 1$.*

**Preuve.** Supposons que $r_c(\mathcal{C}) \geq d$, alors il existe un mot $x$ de l'espace $\mathbb{F}_q - \mathcal{C}$ dont la distance à chaque mots de $\mathcal{C}$ est au moins $d$ (sinon on aurait $r_c(\mathcal{C}) < d$). On peut alors construire le code $\mathcal{C}' = \mathcal{C} \cup x\}$. Ce code a une distance minimale au moins de $d$ en raison de la condition sur le rayon de recouvrement et contient $|\mathcal{C}| = A_q(n, d) + 1$ mots. Or, comme $A_q(n, d)$ est maximal, ce n'est pas possible, donc on a $r_c(\mathcal{C}) \leq d - 1$. ■

Nous pouvons maintenant établir la borne de Gilbert-Varshamov.

**Théorème 1.53. (Borne de Gilbert-Varshamov)** *Soit $\mathcal{C}$ un code de longueur $n$ et de distance minimale $d$. On a alors*

$$\frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i} \leq A_q(n, d).$$

**Preuve.** Soit $\mathcal{C}$ un code de longueur $n$ et de distance minimale $d$ contenant $A_q(n, d)$ mots. Par définition du rayon de recouvrement, la réunion des boules de rayon $r_c$, centrées sur les

mots du code, recouvre tout l'espace $\mathbb{F}_q^n$. D'après le lemme précédent, le rayon de recouvrement de $\mathcal{C}$ est au plus $d-1$ et donc la réunion des $B(c, d-1)$ recouvre l'espace $\mathbb{F}_q^n$, soit

$$A_q(n,d) \geq \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i}.$$

Ce théorème n'est pas constructif. En pratique, les meilleurs codes connus dans le cas binaire sont des codes réalisent, à peu de chose près, l'égalité pour cette borne. En particulier, on peut prouver, mais cela dépasse l'objet de ce cours, que des codes aléatoires binaires (des codes définis par une matrice génératrice dont tous les coefficients sont pris soit à 0 soit à 1 avec la même probabilité 1/2) satisfont cette borne inférieure, asymptotiquement, presque tout le temps.

**Test 1.23.**

Montrer que l'on peut, par une preuve très similaire au cas $A_q(n,d)$, prouver que $B_2(n,d)$ vérifie la même inégalité que $A_2(n,d)$ dans la borne de Gilbert-Varshamov.

**Remarque.** En utilisant la formule de Stirling et le fait que dans la somme $\sum_{i=0}^{d-1} \binom{n}{i}$ le terme dominant correspond à $\binom{n}{d-1}$, on peut obtenir une bonne approximation asymptotique de la somme $\sum_{i=0}^{d-1} \binom{n}{i}$ en

$$\binom{n}{d} \approx n^{H_2(\frac{d}{n})},$$

pour $H_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$. Cette approximation permet d'avoir une idée des paramètres d'un code binaire linéaire $[n,k,d]$ satisfaisant la borne de Gilbert-Varshamov. Pour un tel code $\mathcal{C}$, on a

$$\{\text{Nombre de mots d'une boule de rayon } d-1\} \times 2^k \approx 2^n,$$

ce qui donne, en utilisant l'approximation pour la somme de coefficients binomiaux,

$$n^{H_2(\frac{d}{n})}.2^k \approx 2^n$$

et donc

$$d \approx nH_2^{-1}(\frac{k}{n}).$$

Cela montre donc que, pour un taux $k/n$ fixé, les codes satisfaisant la borne de Gilbert-Varshamov (dont on sait qu'il en existe, même si la borne n'est pas constructive) ont une distance qui augmente linéairement avec $n$. Par exemple, pour un code de taux $1/2$, la borne prouve l'existence de codes binaires $[n, n/2, \delta n]$ avec $\delta = H_2^{-1}(1/2) \approx 0,11$.
Plus généralement, on dit qu'une famille de codes dont les paramètres sont de la forme $[n, Rn, \delta n]$, pour $n$ tendant vers l'infini, est asymptotiquement bonne. Cela signifie que, pour un taux $k/n$ fixé, la distance croîtlinéairement par rapport à la longueur, ce qui est le mieux que l'on puisse obtenir.

## III. Quelques rappels sur les corps finis

Dans cette section, nous rappelons des notions sur les corps finis nécessaires pour considérer les codes cycliques que l'on verra à la section suivante. On pourra consulter le chapitre 4 du tome *Mathématiques L2* pour une introduction aux corps finis et le tome *Algèbre L3*, chez le même éditeur, pour des approfondissements et les preuves de ce que nous présentons ci-dessous.

Il existe deux types de corps finis : les corps avec $p$ éléments où $p$ est premier (ces corps sont construits en considérant le quotient $\mathbb{Z}/p\mathbb{Z}$), et les corps sur des extensions qui sont construits comme quotients $\mathbb{F}_q[x]/(f(x))$ de $\mathbb{F}_q[x]$ par (l'idéal engendré par) un polynôme $f(x)$ irréductible de degré $r$ de $\mathbb{F}_q[x]$. Ici, $\mathbb{F}_q$ désigne un corps fini donné, qui peut être de la forme $\mathbb{F}_p$ (pour $p$ premier) ou bien construit lui-même en tant qu'extension de $\mathbb{F}_p$. Dans tous les cas, le cardinal $q$ du corps est une puissance d'un nombre premier $p$.
Par exemple, si l'on prend $\mathbb{F}_2$ le corps à deux éléments $\{0,1\}$ et $x^2+x+1$ irréductible de degré 2 sur $\mathbb{F}_2[x]$, le corps $\mathbb{F}_4 \stackrel{\text{déf}}{=} \{0,1,w,w^2\}$ peut s'obtenir comme $\mathbb{F}_2[x]/(x^2+x+1)$ ; on peut identifier $x$ à $w$, où la somme et la multiplication se font modulo $x^2+x+1$. Tout élément de $\mathbb{F}_4$ peut s'écrire comme un polynôme $\alpha+\beta w$ de degré 1 avec $\alpha$ et $\beta$ dans $\mathbb{F}_2$. Ainsi, modulo $x^2+x+1$, on obtient $w^2 = 1+w$ et $w.(1+w) = w+w^2 = 1$.

**Théorème 1.54.** *Tout corps fini $\mathbb{F}_q$ admet un élément primitif $\beta$, c'est à dire un élément $\beta$ tel que*

$$\mathbb{F}_q = \{0, 1 = \beta^0, \beta, \beta^2, \cdots, \beta^{q-2}\}.$$

Une notion importante pour un élément $\alpha$ de $\mathbb{F}_q$ est la notion de polynôme minimal $m_\alpha(x)$, c'est-à-dire le polynôme unitaire de plus petit degré dont $\alpha$ est racine. On a le théorème suivant.

**Théorème 1.55.** *Soient $\mathbb{F}_{q^r}$ un corps fini et $\alpha$ un élément de ce corps de polynôme minimal $m_\alpha(x) \in \mathbb{F}_q[x]$ sur $\mathbb{F}_q$, alors :*

1. *$m_\alpha(x)$ est irréductible dans $\mathbb{F}_q[x]$ ;*
2. *si $\alpha$ est une racine d'un polynôme $g(x)$ de $\mathbb{F}_q[x]$, alors $m_\alpha(x)$ divise $g(x)$ ;*
3. *$m_\alpha(x)$ divise $x^{q^r} - x$ ;*
4. *le degré de $m_\alpha(x)$ est au plus $r$.*

Une autre notion utilisée pour la suite est la notion de *classe cyclotomique*. On remarque tout d'abord que si $\alpha$ est une racine d'un polynôme $f(x)$ sur $\mathbb{F}_{q^r}$, puisque $(x+y)^q = x^q + y^q$, alors $f(\alpha^q) = f(\alpha)^q = 0$. Plus généralement, si $\alpha$ est un zéro d'un polynôme $f(x)$ à coefficients dans $\mathbb{F}_q$, alors $\{\alpha^q, \alpha^{q^2}, \ldots\}$ sont aussi des zéros de $f(x)$.

**Définition 1.56. (Classe cyclotomique)** *Soient $q$ une puissance d'un premier $p$ et $a \in \{0, \cdots, q^r - 1\}$ un entier.*
*On appelle $q$-classe cyclotomique de $a$ modulo $q^r - 1$ l'ensemble*

$$C_a \stackrel{\text{déf}}{=} \{a, aq, aq^2, \cdots, aq^{m-1}\},$$

*où $m$ est la plus petite valeur non nulle telle que $a \equiv aq^m \pmod{q^r - 1}$.*

Soit $\beta$ un élément primitif du corps $\mathbb{F}_{q^r}$. Une classe cyclotomique $C_a$ rassemble tous les éléments $\beta^j$, pour $j \in C_a$, ayant le même polynôme minimal sur $\mathbb{F}_q$.

**Lemme 1.57.** *Pour un corps $\mathbb{F}_{q^r}$, le cardinal d'une classe cyclotomique $C_a$ divise $r$.*

**Preuve.** À toute classe cyclotomique $C_a$ de $n$ éléments, on peut associer un polynôme minimal $m_\alpha(x)$ associé à un élément $\alpha$ de $\mathbb{F}_{q^r}$ en prenant $m_\alpha(x) = \prod_{j \in C_a}(x - \alpha^j)$ ; par construction de la classe cyclotomique, on a $m_\alpha(x) \in \mathbb{F}_q[x]$. On peut alors construire un sous-corps de $\mathbb{F}_{q^r}$ en prenant $\mathbb{F}_q[x]/(m_\alpha(x))$. Comme $m_\alpha(x)$ a pour degré $n$, ce corps est isomorphe à $\mathbb{F}_{q^n}$ et est inclus dans $\mathbb{F}_{q^r}$. Or $\mathbb{F}_{q^n}$ sous-corps de $\mathbb{F}_{q^r}$ implique que $n$ divise $r$. ■

**Exemple 1.58.** Sur $\mathbb{F}_{2^4}$, les 2-classes cyclotomiques modulo 15 sont $C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8\}$, $C_3 = \{3, 6, 12, 9\}$, $C_5 = \{5, 10\}$ et $C_7 = \{7, 14, 13, 11\}$.
On vérifie que le cardinal de chaque classe cyclotomique (1, 2 ou 4) divise 4.

**Rappel — Racine $n$-ième de l'unité**

Soient $\mathbb{F}_{q^r}$ un corps et $\alpha$ un élément de ce corps. On dit que $\alpha$ est une racine $n$-ième de l'unité si la plus petite valeur de $m$ telle que $\alpha^m = 1$ est $n$.

Voyons comment construire, pour tout corps fini $\mathbb{F}_q$, une extension de ce corps qui contient une racine $n$-ième.

**Définition 1.59.** *Soient $\mathbb{F}_q$ un corps et $1 \leq n \leq q-1$. On appelle ordre de $n$ sur $q$, noté $\mathrm{ord}_q(n)$, la plus petite valeur de $m$ telle que $n$ divise $q^m - 1$.*

La définition précédente assure que pour $n$ fixé et un corps $\mathbb{F}_q$ fixé, on peut trouver une extension $\mathbb{F}_{q^{\mathrm{ord}_q(n)}}$ qui admet une racine $n$-ième. Précisément, si l'on note $\beta$ une racine primitive de $\mathbb{F}_{q^{\mathrm{ord}_q(n)}}$, $\alpha = \beta^{\frac{q^{\mathrm{ord}_q(n)}-1}{n}}$ est une racine $n$-ième dans cette extension.

Nous avons construit la notion de $q$-classe cyclotomique modulo $q^r - 1$ ; voyons comment généraliser cette construction modulo des $n$ quelconques. Une $q$-classe cyclotomique $C_a$ modulo $n$ est l'ensemble $C_a = \{a, aq, aq^2, \cdots, aq^{m-1}\}$, où $m$ est la plus petite valeur non nulle telle que $aq^m \equiv a \pmod{n}$.

On a alors le théorème suivant qui sera largement utilisé dans le chapitre suivant.

**Théorème 1.60.** *Soient $\alpha$ une racine $n$-ième de l'unité avec $\alpha \in \mathbb{F}_{q^{\mathrm{ord}_q(n)}}$ et $m_{\alpha^i}(x)$ le polynôme minimal associé à $\alpha^i$, alors*

$$x^n - 1 = \Pi_{i \in I} m_{\alpha^i}(x),$$

*où $I$ est un ensemble de représentants des $q$-classes cyclotomiques modulo $n$.*

**Exemple 1.61.** Soient $n = 13$ et $q = 3$, alors les 3-classes cyclotomiques modulo 13 sont $C_0 = \{0\}$, $C_1 = \{1, 3, 9\}$, $C_2 = \{2, 6, 5\}$, $C_4 = \{4, 12, 10\}$ et $C_7 = \{7, 8, 11\}$. Cette décomposition implique que $x^{13} - 1$ se décompose dans $\mathbb{F}_q[x]$ comme le produit de $x - 1$ et de quatre polynômes de degrés 3.

**Définition 1.62.** *Soient $\alpha$ une racine $n$-ième de l'unité dans une extension $\mathbb{F}_{q^{\mathrm{ord}_q(n)}}$ de $\mathbb{F}_q$ et $g(x) \in \mathbb{F}_q[x]$ un polynôme qui divise $x^n - 1$. L'ensemble $Z$ des zéros de $g(x)$ est l'ensemble des $i$ tels que $\alpha^i$ est racine de $g(x)$.*

L'ensemble des zéros correspond à $\{\alpha^i\}$ pour $i$ dans la réunion des $q$-classes cyclotomiques associées à la décomposition de $g(x)$ dans le théorème précédent.

**Test 1.24.**

Calculer les 2-classes cyclotomiques modulo 15.

**Test 1.25.**

Calculer $\mathrm{ord}_2(9)$. Calculer les 2-classes cyclotomiques modulo 9. Factoriser $x^9 - 1$ sur $\mathbb{F}_2$.

**Test 1.26.**

Construire le corps $\mathbb{F}_8$ comme une extension de $\mathbb{F}_2$ de degré 3.

# IV. Codes cycliques et codes BCH

## IV.1. Théorie élémentaire des codes cycliques

De la même façon qu'ajouter de la structure aux codes non linéaires permet d'obtenir des codes linéaires, plus faciles à manipuler de par leur matrice d'encodage, on peut se poser la question d'ajouter de la structure aux codes linéaires. C'est l'idée des codes linéaires cycliques qui sont encore plus structurés que les codes linéaires.

**Définition 1.63. (Code cyclique)** *Un code $\mathcal{C}$ est dit cyclique si, pour tout mot $c = (c_0, c_1, \cdots, c_{n-1})$ de $\mathcal{C}$, le mot $c' = (c_{n-1}, c_0, c_1, \cdots, c_{n-2})$ appartient aussi au code $\mathcal{C}$.*

Bien qu'il soit possible qu'un code soit cyclique et non linéaire, on ne considère dans la suite que le cas des codes cycliques linéaires.

Si l'on prend un mot quelconque et que l'on regarde le code linéaire engendré par ses permutations circulaires, on obtient en général tout l'espace ou le code des mots de poids pair, dans le cas d'un mot binaire de poids pair ; mais, dans certains cas, on obtient des codes de dimensions bien plus petites. Par exemple, le code linéaire engendré par les permutations circulaires du mot $(1,1,0,1,0,0,0)$ est un code linéaire cyclique de dimension strictement inférieure à 7.

**Test 1.27.**

Construire le code engendré par les permutations circulaires de $(1,1,0,1,0,0,0)$. Quels sont les paramètres du code ainsi construits ?

On peut mieux appréhender les codes cycliques en se plaçant d'un point de vue polynomial. On associe au mot $a = (a_0, a_1, \cdots, a_{n-1})$ de longueur $n$ sur $\mathbb{F}_q$ le polynôme $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$ ; dans la suite, on identifiera un mot $a$ à sa représentation polynomiale $a(x)$. Dans ce contexte, on peut exprimer le fait qu'un code soit cyclique par le fait que, représenté polynomialement, il soit stable par multiplication par $x$ dans l'anneau quotient $\mathbb{F}_q[x]/(x^n - 1)$ ; en effet, si $c = (c_0, c_1, \cdots, c_{n-1}) = c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$, alors

$$\tilde{c} \stackrel{\text{déf}}{=} (c_{n-1}, c_0, c_1, \cdots, c_{n-2}) = c_{n-1} + c_0 x + c_1 x^2 + \cdots + c_{n-2}x^{n-1} = xc(x) - c_{n-1}(x^n - 1).$$

En d'autres termes, les codes cycliques linéaires sont des idéaux de $\mathbb{F}_q[x]/(x^n - 1)$. On a alors le théorème suivant.

**Théorème 1.64.** *Soit $\mathcal{C}$ un code cyclique linéaire de longueur $n$ et de dimension $k$ sur $\mathbb{F}_q$, où les mots de $\mathcal{C}$ sont considérés comme des polynômes, éléments de $\mathbb{F}_q[x] \stackrel{\text{déf}}{=} \mathbb{F}_q[X]/(X^n - 1)$. Il existe dans $\mathcal{C}$ un unique polynôme unitaire $g(x)$, non nul de plus petit degré. Il vérifie :*

1. *$g(x)$ divise tout mot $c(x)$ de $\mathcal{C}$ ;*
2. *$g(x)$ divise $X^n - 1$ dans $\mathbb{F}_q[X]$ ;*
3. *$\deg(g(x)) = n - k$.*

Le polynôme $g$ ainsi défini est appelé polynôme générateur du code $\mathcal{C}$.

**Preuve.** Il existe au moins un polynôme unitaire de plus petit degré de $\mathcal{C}$ puisque les degrés possibles sont entiers et bornés par 1. Supposons qu'il existe deux tels polynômes unitaires $g_1$ et $g_2$ ; alors on peut toujours construire un polynôme unitaire de la forme $a(g_1 - g_2)$ (pour $a \in \mathbb{F}_q$) qui appartient au code et qui est de degré strictement inférieur à $\deg(g_1)$. Mais puisque $\deg(g_1) = \deg(g_2)$ est le plus petit degré possible, on obtient une contradiction et donc $g(x)$ est unique. Pour 1), soit $c(x)$ un mot non nul du code, alors si l'on fait la division euclidienne de $c(x)$ par $g(x)$, on obtient $c(x) = g(x)q(x)+r(x)$ avec $\deg(r(x)) < \deg(g(x))$ ou $r(x) = 0$. Comme le code est cyclique et que $g(x) \in \mathcal{C}$, la multiplication de $g(x)$ par $x^j$ pour $j \leq n-1-\deg(g(x))$ est toujours un mot du code et donc $g(x)q(x) \in \mathcal{C}$ qui donne $r(x) \in \mathcal{C}$. Mais comme $g(x)$ est de degré minimal, $r(x)$ est nécessairement nul, ce qui prouve le résultat. Soit maintenant $c$ un mot du code tel que $c_{n-1} = 1$, alors comme $g(x)$ divise $c(x)$ et son décalé de une coordonnée $c'(x) = xc(x) - c_{n-1}(x^n - 1)$, $g(x)$ divise $x^n - 1$ ce qui prouve 2). Tout mot du code peut être vu comme un polynôme de degré au plus $n-1$, multiple d'un polynôme de degré $\deg(g(x))$ ; ces polynômes peuvent donc s'écrire sous la forme $g(x)h(x)$ pour $\deg(h) \leq n-1-\deg(g(x))$. Il y a donc exactement $q^{n-\deg(g(x))}$ mots possibles qui, par construction, sont tous distincts et donc la dimension du code est $k = n - \deg(g(x))$, ce qui donne 3) et achève la preuve de ce théorème. ∎

Le théorème précédent montre que le bon contexte pour étudier les codes cycliques est l'anneau $\mathbb{F}_q[x]/(x^n - 1)$, comme le montre la proposition suivante.

**Proposition 1.65.** *Les codes cycliques sont exactement les idéaux de l'anneau* $\mathbb{F}_q[x]/(x^n-1)$.

**Preuve.** Dans l'anneau $\mathbb{F}_q[x] = \mathbb{F}_q[x]/(x^n - 1)$, la permutation circulaire, à droite, d'un mot du code est égale à la multiplication par $x$. En effet, le théorème précédent montre qu'un code cyclique est un idéal de $\mathbb{F}_q[x]/(x^n - 1)$. Réciproquement, les idéaux de $\mathbb{F}_q[x]/(x^n - 1)$ sont engendrés par les diviseurs de $(x^n - 1)$ et chacun des diviseurs de $(x^n - 1)$ engendre un code cyclique. ∎

En d'autres termes, l'étude des codes cycliques se ramène à l'étude des diviseurs de $x^n - 1$ qui sont en bijection avec l'ensemble des codes cycliques. Pour éviter le cas des diviseurs multiples, on se place par la suite dans le cas où la caractéristique du corps et la longueur du code sont premiers entre eux.

**Exemple 1.66.** Prenons $n = 7$. Sur $\mathbb{F}_2$, les diviseurs irréductibles de $x^7 - 1$ sont $x + 1, x^3 + x^2 + 1, x^3 + x + 1$. Les $2^3$ diviseurs de $x^7 - 1$ peuvent se déduire de ces polynômes ; il y a donc 8 codes cycliques de longueurs 7 sur $\mathbb{F}_2$. En particulier, le code cyclique engendré par le polynôme générateur $x^3 + x + 1$ donne un code $[7, 4]$ qui a une distance minimale de 3 et se trouve être le code de Hamming $[7, 4, 3]$.

**Test 1.28.**

Construire tous les codes cycliques de longueur 7 sur $\mathbb{F}_2$ : donner leur polynôme générateur, leur dimension et leur distance minimale. Faire de même pour les codes cycliques de longueur 9.

On déduit du théorème précédent le corollaire suivant.

**Corollaire 1.67.** *Soit $\mathcal{C}$ un code cyclique sur $\mathbb{F}_q$ de longueur $n$ et de polynôme générateur $g(x) = \sum_{i=0}^{n-k} g_i x^i$ de degré $n-k$, alors, l'ensemble des mots de $\mathcal{C}$ est l'ensemble $\{g(x)a(x) | deg(a(x)) \leq k-1\}$. Le code $\mathcal{C}$ a pour dimension $k$ et a pour matrice génératrice la matrice*

$$\begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ & & \cdots & & \cdots & & \cdots & \\ 0 & 0 & 0 & \cdots & 0 & g_0 & \cdots & g_{n-k} \end{bmatrix}.$$

Une autre façon de caractériser les codes cycliques peut se faire à partir des zéros du polynôme générateur, que l'on peut aussi définir à partir de *l'ensemble de définition.*

**Définition 1.68. (Ensemble de définition)** *Pour $\alpha$ une racine $n$-ième de l'unité dans une extension $\mathbb{F}_{q^{ord_q(n)}}$ de $\mathbb{F}_q$ et $g(x)$ un polynôme générateur d'un code cyclique $\mathcal{C}$ (donc qui divise $x^n - 1$), l'ensemble de définition $T$ du code $\mathcal{C}$ est l'ensemble des $i$ tels que $\alpha^i$ est une racine de $g(x)$.*

L'ensemble $T$ correspond à la réunion des $q$-classes cyclotomiques associées à la décomposition de $g(x)$ dans le théorème 1.64.

Le code dual est lui aussi cyclique.

**Proposition 1.69.** *Si $\mathcal{C}$ est un code cyclique de longueur $n$ sur $\mathbb{F}_q$, alors le code dual $\mathcal{C}^\perp$ est aussi cyclique.*

PREUVE. Dire que $\mathcal{C}$ est cyclique équivaut à dire que le groupe d'automorphisme de $\mathcal{C}$ contient la permutation circulaire $(1, 2, 3, \cdots, n)$. En notant $H$ une matrice de parité de $\mathcal{C}$, on a, pour tout $c \in \mathcal{C}$, $H.c^t = 0$. Puisque le code est cyclique, pour toute permutation circulaire $P$, on a $H.(c.P)^t = 0 = H.P^t.c^t = H.P^{-1}.c^t$ et donc tout élément de $\mathcal{C}$ est orthogonal à $HP^{-1}$. Mais comme $P^{-1}$ est aussi une permutation circulaire, le code engendré par $HP^{-1}$ est égal à $\mathcal{C}^\perp$. Le code dual $\mathcal{C}^\perp$ étant stable par $P^{-1}$, il est bien cyclique. ■

Pour tout $g(x) = g_0 + g_1 x + \cdots + g_{n-k-1}x^{n-k-1} + x^{n-k}$ polynôme générateur d'un code cyclique $\mathcal{C}$ de longueur $n$, il existe $h(x) = h_0 + h_1 x + \cdots + h_{k-1}x^{k-1} + x^k$ tel que $g(x)h(x) = x^n - 1$. Si l'on se place dans $\mathbb{F}_q[x] = \mathbb{F}_q[x]/(x^n - 1)$, on obtient $g(x)h(x) = 0$ et donc, en écrivant le fait que chacun des coefficients pour un degré donné est nul dans $g(x)h(x)$ (ainsi que pour les produits de $g(x)$ avec les permutés circulaires de $h(x)$), on obtient le théorème suivant.

**Théorème 1.70.** *Si $\mathcal{C}[n, k]_q$ est un code cyclique de polynôme générateur $g(x)$, alors $\mathcal{C}^\perp$ est aussi un code cyclique de polynôme générateur $g^\perp(x) = x^{n-k}\frac{h(x^{-1})}{h_0}$, où $h(x) = \frac{x^n-1}{g(x)}$.*

PREUVE. Le résultat est obtenu directment à partir du calcul de $g(x)h(x) = 0$ dans $\mathbb{F}_q[x] = \mathbb{F}_q[x]/(X^n - 1)$ en notant que les coefficients de $g^\perp(x)$ sont exactement ceux de $h(x)$, mais en sens inverse. ■

**Remarque.** Lorsque l'on choisit une racine $n$-ième de l'unité pour un code cyclique de longueur $n$, il peut y avoir plusieurs choix possibles. En effet, à partir d'une telle racine $\alpha$, tous les $\alpha^i$ avec $pgcd(i, n) = 1$ sont aussi des racines $n$-ièmes. Le fait de prendre des racines $n$-ièmes différentes peut mener à des codes distincts, mais toujours équivalents. Deux codes équivalents ayant les

mêmes propriétés en termes de distance de Hamming, on ne précise pas en général la racine utilisée lorsque l'on regarde les propriétés générales du code. Bien sûr, quand on le construit concrètement, on doit le faire.

**Test 1.29.**

Justifier la remarque précédente.

**Test 1.30.**

Quel est le dual du code cyclique binaire de longueur 7 de polynôme générateur $x^3+x+1$ ? Quel est son ensemble de définition ? Quel est celui de son dual ?

## IV.2. Borne BCH

La borne BCH est par la suite utilisée pour la définition des codes BCH qui ont été introduits a peu près simultanément par Hocquenghem d'un côté et Bose et Chauduri de l'autre en 1959.

Nous commençons par rappeler la notion de matrice de Vandermonde, vue au chapitre 2 dans le cadre de l'interpolation.

**Rappel — Matrice de Vandermonde**

Soient $x_1, x_2, \cdots, x_n$ des éléments d'un anneau. Leur *matrice de Vandermonde* $V$ est définie par

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ & & \cdots & \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix}.$$

Son déterminant vaut $\det(V) = \Pi_{1\leq i<j\leq n}(x_i - x_j)$.

En particulier, cela signifie que $\det(V)$ est non nul si et seulement si les $x_i$ sont tous distincts.

Soit $\mathcal{C}$ un code cyclique ; on rappelle que l'on nomme ensemble de définition $T$ du code $\mathcal{C}$ l'ensemble des $i$ tels que $\alpha^i$ est un zéro de $\mathcal{C}$. On dira que $T$ *contient un ensemble de* $s$ *éléments consécutifs* s'il existe un ensemble $\{b, b+1, \cdots, b+s-1\}$ d'entiers consécutifs tels que $\{b, b+1, \cdots, b+s-1\} \pmod n \subset T$.

**Théorème 1.71.** *Soit $\mathcal{C}$ un code cyclique de longueur $n$ sur $\mathbb{F}_q$ avec un ensemble de définition $T$. Si l'on note $d$ la distance minimale du code $\mathcal{C}$ et si $T$ contient $\delta-1$ éléments consécutifs, alors $\delta \leqslant d$.*

**Preuve.** Par hypothèse, il existe $b$ tel que les éléments $\alpha^b, \cdots, \alpha^{b+\delta-2}$ sont des zéros du code $\mathcal{C}$. Soit $c(x)$ un mot non nul du code de poids $w \leqslant \delta-1$. Le mot $c(x)$ est non nul sur $w$ coordonnées et s'écrit $c(x) = \sum_{j=1}^{w} c_{i_j} x^{i_j}$. Supposons que $w < \delta$. Comme $c(\alpha^i) = 0$ pour tout $b \leq i \leq b+\delta-2$, on a alors $M.u^t = 0$ pour

$$M = \begin{bmatrix} \alpha^{i_1 b} & \alpha^{i_2 b} & \cdots & \alpha^{i_w b} \\ \alpha^{i_1 (b+1)} & \alpha^{i_2 (b+1)} & \cdots & \alpha^{i_w (b+1)} \\ & \cdots & \cdots & \\ \alpha^{i_1 (b+w-1)} & \alpha^{i_2 (b+w-1)} & \cdots & \alpha^{i_w (b+w-1)} \end{bmatrix},$$

avec $u = (c_{i_1}, \cdots, c_{i_w})$. Mais, comme $w \leqslant \delta - 1$, on peut écrire $\det(M) = \alpha^{b(i_1+\cdots+i_w)}\det(V)$, où $V$ est la matrice de Vandermonde

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \cdots & \alpha^{i_w} \\ & \cdots & \cdots & \\ \alpha^{i_1 w-1)} & \alpha^{i_2(w-1)} & \cdots & \alpha^{i_w(w-1)} \end{bmatrix}$$

de déterminant non nul, puisque les $\alpha^{i_k}$ sont distincts pour $1 \leqslant k \leqslant w$. On en déduit que la matrice $M$ est inversible et donc que la solution de $M.u^t = 0$ est $u = 0$ ; or $u \neq 0$, soit une contradiction et donc $w \geqslant \delta$. ■

**Remarque.** On parle pour $\delta$ de *distance construite* du code. La valeur $\delta$ constitue une minoration de la vraie distance minimale $d$ du code. En pratique, il y a souvent très peu de différence entre $d$ et $\delta$.

**Test 1.31.**

Soit le code de Hamming ayant pour ensemble de définition $T = C_1 = \{1, 2, 4\}$. Montrer en utilisant la borne précédente que sa distance minimale vaut 3. Et si l'on ajoute 0 à l'ensemble de définition ? Soit le code binaire cyclique en longueur 23 d'ensemble de définition $T = C_1$, la 2-classe cyclotomique de 1 modulo 23 ; que donne alors la borne ?

## IV.3. Codes BCH

**Définition 1.72. (Codes BCH)** *Soient $n$ et $q$ deux entiers premiers entre eux, et soit $\alpha$ une racine $n$-ième de 1 sur $\mathbb{F}_q^{\mathrm{ord}_n(q)}$. Soit $\delta$ un entier avec $2 \leq \delta \leq n$. Un code BCH sur $\mathbb{F}_q$ de longueur $n$ et de distance construite $\delta$ est un code cyclique avec pour ensemble de définition*

$$T = C_b \cup C_{b+1} \cup \cdots \cup C_{b+\delta-2},$$

*où $C_i$ désigne la $q$-classe cyclotomique modulo $n$ contenant $i$.*

Pour différents $b$, on obtient des codes différents. Pour $b = 1$, on parle de *code BCH au sens strict*. Si de plus $n$ est de la forme $q^t - 1$ avec $t$ quelconque, on parle de *code BCH primitif*.

**Théorème 1.73.** *Soit $\mathcal{C}$ un code BCH $[n, k]_q$, de distance construite $\delta$, alors :*

1. $k \geq n - \mathrm{ord}_q(n).(\delta - 1)$ ;
2. *si $q = 2$ et si $\mathcal{C}$ est un code BCH au sens strict, alors on peut restreindre $\delta$ au cas où $\delta$ est impair ; de plus, si l'on écrit $\delta = 2w + 1$, alors $k \geq n - \mathrm{ord}_q(n).w$.*

**Preuve.** L'ensemble de définition de $\mathcal{C}$ est la réunion de $\delta - 1$ classes cyclotomiques. Comme chaque classe contient au plus $\mathrm{ord}_q(n)$ éléments, l'ensemble de définition du code contient au plus $\mathrm{ord}_q(n).(\delta - 1)$ éléments et la dimension du code est au moins $n - \mathrm{ord}_q(n).(\delta - 1)$. Pour le cas binaire, avec $\delta$ impair écrit sous la forme $\delta = 2w + 1$ et pour un code BCH au sens strict, on remarque que l'ensemble de définition $T = \cup_{i=1}^{2w} C_i$ est contenu dans la réunion $\cup_{i=1}^{w} C_{2i-1}$, puisque $C_{2a} = C_a$ pour $a$ quelconque, ce qui donne 2). ■

**Exemple 1.74.**

Prenons en longueur 15 sur $\mathbb{F}_2$, des codes BCH au sens strict qui sont primitifs.
- $\delta = 3$ donne $T = C_1 \cup C_2 = C_1 = \{1, 2, 4, 8\}$, soit $[15, 11, d \geq 3]$ ;

- $\delta = 4$ donne $T = C_1 \cup C_2 \cup C_3 = C_1 \cup C_3 = \{1, 2, 3, 4, 6, 8, 9, 12\}$, soit $[15, 7, d \geq 5]$ ;
- $\delta = 5$ n'ajoute rien par rapport au cas $\delta = 4$.
- $\delta = 6$ et $\delta = 7$ donnent $T = C_1 \cup C_3 \cup C_5$, soit $[15, 5, d \geq 7]$ ;
- $\delta = 8$ et plus donne $[15, 1, 15]$ car il n'y plus de classe cyclotomique à ajouter (à part 0).

Un calcul des polynômes générateur et le fait de construire un mot de poids précisement égal à $d$ dans chaque cas montre qu'en fait ces distances sont exactes. Par exemple, pour $\delta = 4$, on trouve, en prenant $x^4 + x + 1$ pour polynôme de construction de l'extension, que $g(x) = x^8 + x^7 + x^6 + x^4 + 1$.

## IV.4. Codes de Reed-Solomon cycliques

Les codes de Reed-Solomon ont été introduits originellement par I. S. Reed et G. Solomon en 1959 par une construction n'utilisant pas de cyclicité, comme on le verra plus loin, mais les codes de Reed-Solomon peuvent aussi s'interpréter comme des codes BCH particuliers (et donc cycliques). Ils ont longtemps été présentés sous cette forme car ils étaient aussi décodés en tant que codes BCH particuliers.

**Définition 1.75. (Codes de Reed-Solomon cycliques)** *On appelle codes de Reed-Solomon cycliques sur $\mathbb{F}_q$ les codes BCH de longueur $n = q - 1$.*

Dans ce cas, $\mathrm{ord}_q(n) = 1$ et donc tous les facteurs irréductibles de $x^n - 1$ sont de degré un. On a alors le théorème suivant.

**Théorème 1.76.** *Soit $\mathcal{C}$ un code de Reed-Solomon cyclique sur $\mathbb{F}_q$ de longueur $n = q - 1$ et de distance construite $\delta$, alors*

1. *$\mathcal{C}$ a pour ensemble de définition $T = \{b, b+1, \cdots, b+\delta-2\}$ ;*
2. *$\mathcal{C}$ a une distance minimale $d = \delta$ et une dimension $k = n - d + 1$ ;*
3. *$\mathcal{C}$ réalisent l'égalité dans la borne de Singleton et est MDS.*

**Preuve.** Pour un tel code, on a $\mathrm{ord}_q(n) = 1$ et donc $T = \{b, b+1, \cdots, b+\delta-2\} = \cup_{i=b}^{b+\delta-2} C_i$. La taille de l'ensemble de définition est exactement $\delta - 1$, soit $k = n - (\delta - 1) = n - \delta + 1$. Mais, comme $\delta \leqslant d$ et comme la borne de Singleton assure que $k \leqslant n - d + 1$, on a alors $d = \delta$, soit $k = n - \delta + 1 = n - d + 1$. Pour le point 3), on remarque qu'un code est MDS si précisément il vérifie l'égalité $k = n - d + 1$. ∎

**Test 1.32.**

Si l'on prend $\mathbb{F}_7$, construire un code de distance minimale 4. Quels sont les paramètres du code ?

## V. Décodage des codes BCH

Un premier algorithme de décodage a été proposé en 1960 par G. Peterson pour les BCH binaires, puis généralisé par G. Peterson, D. Gorenstein et N. Zierler la même année. En 1968, E. Berlekamp et J. Massey ont proposé un algorithme qui a permis d'améliorer les algorithmes précédents sur la partie de l'obtention du polynôme localisateur d'erreurs, que l'on verra par la suite et qui est une étape importante du décodage. Enfin, en 1974, a été proposé un décodage par l'algorithme d'Euclide étendu, d'une complexité asymptotique similaire à celle de Berlekamp-Massey, mais conceptuellement plus simple. C'est ce dernier algorithme que nous présentons dans cette section.

## V.1. Algorithme de Peterson

Soit $\mathcal{C}$ un code BCH sur $\mathbb{F}_q$ de longueur $n$ et de distance construite $\delta$. On va décoder jusqu'à au plus $t = \lfloor \frac{\delta-1}{2} \rfloor$ erreurs. On suppose que l'ensemble de définition $T$ du code contient un ensemble de $\delta - 1$ zéros consécutifs et que, sans perte de généralité, on peut prendre un code au sens strict, c'est-à-dire que $\{1, 2, \cdots, \delta - 1\} \subset T$. On considère une racine primitive $\alpha$ d'ordre $n$ de $\mathbb{F}_{q^m}$ avec $m = \mathrm{ord}_q(n)$.

On reçoit le vecteur $r(x) = c(x) + e(x)$ avec $e$ un vecteur d'erreurs de poids $w_H(e) \leq t$ et où $c(x)$ correspond au vecteur envoyé. On note $e(x) = e_{k_1} x^{k_1} + \cdots + e_{k_\nu} x^{k_\nu}$ ; le problème du décodage est de retrouver $e(x)$. Puisque $c(x) \in \mathcal{C}$, on a $c(\alpha^i) = 0, \forall i \in T$. En particulier, on a

$$r(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i), \forall i \in T.$$

On peut remarquer que $[1, ..., 2t] \subset T$, puisque $2t \leq \delta - 1$.
On note $S_i = r(\alpha^i)$ pour $1 \leq i \leq 2t$ l'ensemble des syndromes associés au mot reçu $\nu$.
On note maintenant

$$H = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \cdots & \alpha^{2(n-1)} \\ & \cdots & \cdots & \\ 1 & \alpha^t & \cdots & \alpha^{t(n-1)} \end{bmatrix}.$$

Soient $\nu = (\nu_0, \cdots, \nu_{n-1}) = \nu_0 + \nu_1 x + \cdots + \nu_{n-1} x^{n-1}$ et $S = (S_1, \cdots, S_t)$, alors on a $H.\nu^t = S^t$. Pour $e_{k_j}$ la valeur de l'erreur et $k_j$ l'emplacement de l'erreur, on note $E_j = e_{k_j}$ et $X_j = \alpha^{k_j}$, les valeurs des emplacements de l'erreur. On obtient alors

$$S_i = r(\alpha^i) = \sum_{j=1}^{\nu} E_j X_j^i \text{ pour } 1 \leq i \leq 2t.$$

Ceci nous donne le système

$$(\mathcal{S}_1) \ : \ \begin{cases} E_1 X_1 + \cdots + E_\nu X_\nu = S_1 \\ E_1 X_1^2 + \cdots + E_\nu X_\nu^2 = S_2 \\ E_1 X_1^3 + \cdots + E_\nu X_\nu^3 = S_2 \\ \cdots \\ E_1 X_1^{2t} + \cdots + E_\nu X_\nu^{2t} = S_{2t} \end{cases}.$$

Dans ce système, les $E_i$ et les $X_i$ sont des inconnues, seuls les $S_i$ sont connus ; le système est donc non linéaire et est compliqué à résoudre directement. On va alors introduire un polynôme localisateur d'erreurs qui permettra de trouver les $X_i$ et ainsi de se ramener à la résolution d'un système linéaire en les $E_i$.

**Définition 1.77. (Polynôme localisateur d'erreurs)** *Pour $X_j$, $1 \leq j \leq \nu$, les emplacements des erreurs pour le mot reçu $\nu$, on appelle polynôme localisateur d'erreurs le polynôme $\sigma(x)$ défini par*

$$\sigma(x) \stackrel{\text{déf}}{=} (1 - xX_1)(1 - xX_2) \cdots (1 - xX_\nu) = 1 + \sum_{i=1}^{\nu} \sigma_i x^i,$$

*où les racines de $\sigma(x)$ sont les inverses des emplacements d'erreurs $X_j$.*

On obtient alors :

$$\sigma(X_j^{-1}) = 1 + \sigma_1 X_j^{-1} + \cdots + \sigma_\nu X_j^{-\nu} = 0 \text{ pour } 1 \leq j \leq \nu.$$

En multipliant les équations précédentes par $E_j X_j^{i+\nu}$, il vient

$$E_j X_j^{i+\nu} + \sigma_1 E_j X_j^{i+\nu-1} + \cdots + \sigma_\nu E_j X_j^i = 0, \text{ pour tout } i.$$

Si l'on somme alors sur $j$ ($1 \leqslant j \leqslant \nu$), on obtient

$$\sum_{j=1}^{\nu} E_j X_j^{i+\nu} + \sigma_1 \sum_{j=1}^{\nu} E_j X_j^{i+\nu-1} + \cdots + \sigma_\nu \sum_{j=1}^{\nu} E_j X_j^{i} = 0.$$

Pour $1 \leq i$ et $i + \nu \leq 2t$, ces sommes correspondent aux syndromes $S_i$ ; comme $\nu \leq t$, cela nous donne le système d'équations suivant :

$$S_{i+\nu} + \sigma_1 S_{i+\nu-1} + \cdots + \sigma_\nu S_i = 0, \text{ pour } 1 \leq i \leq \nu.$$

Les coefficients du polynôme localisateur d'erreurs sont solutions du système

$$\begin{pmatrix} S_1 & S_2 & \cdots & S_\nu \\ S_2 & S_3 & \cdots & S_{\nu+1} \\ \cdots & \cdots & \cdots & \cdots \\ S_\nu & S_{\nu+1} & \cdots & S_{2\nu-1} \end{pmatrix} \begin{pmatrix} \sigma_\nu \\ \sigma_{\nu-1} \\ \cdots \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} -S_{\nu+1} \\ -S_{\nu+2} \\ \cdots \\ -S_{2\nu} \end{pmatrix}.$$

Le système précédent peut se récrire sous la forme du système système linéaire $(\mathcal{S}_2)$ suivant :

$$\begin{pmatrix} S_1 & S_2 & \cdots & S_\nu & S_{\nu+1} \\ S_2 & S_3 & \cdots & S_{\nu+1} & S_{\nu+2} \\ \cdots & \cdots & \cdots & \cdots & \\ S_\nu & S_{\nu+1} & \cdots & S_{2\nu-1} & S_{2\nu} \end{pmatrix} \begin{pmatrix} \sigma_\nu \\ \sigma_{\nu-1} \\ \cdots \\ \sigma_1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}.$$

On peut donc retrouver les coefficients de $\sigma(x)$ en résolvant le système $(\mathcal{S}_2)$, qui permet de retrouver le polynôme localisateur d'erreurs $\sigma(x)$. On peut alors déduire les emplacements d'erreurs $X_j$ en factorisant $\sigma(x)$. On retrouve les valeurs des coefficients des erreurs $E_j$, en remplaçant dans le système $(\mathcal{S}_1)$ les $X_j$ par leurs valeurs. On obtient alors un système linéaire en les $E_j$, que l'on peut résoudre.

Dans tout ce que l'on a considéré, la valeur de $\nu$ n'est pas connue et la question se pose de connaître le nombre exact d'erreurs $\nu$. Ce problème est résolu par la proposition suivante.

**Proposition 1.78.** *Soit $\mu \leq 2t$ et soit*

$$S_\mu = \begin{pmatrix} S_1 & S_2 & \cdots & S_\mu \\ S_2 & S_3 & \cdots & S_{\mu+1} \\ \cdots & \cdots & \cdots & \cdots \\ S_\mu & S_{\mu+1} & \cdots & S_{2\mu-1} \end{pmatrix},$$

*alors la matrice $S_\mu$ est inversible si $\mu = \nu$ et non inversible si $\mu > \nu$, où $\nu$ désigne le nombre d'erreurs.*

**Preuve.** Soit $V_\mu$ la matrice de Vandermonde définie pour les emplacements d'erreurs $X_1, X_2, \ldots, X_\mu$ avec $X_j = 0$ pour $j > \nu$. En utilisant les valeurs des coefficients $E_j$ pour $1 \leqslant j \leqslant \nu$ et $E_j = 0$ pour $j > \nu$, on peut vérifier que $S_\mu = V_\mu D_\mu V_\mu^t$ avec

$$D_\mu = \begin{pmatrix} E_1 X_1 & 0 & \cdots & 0 \\ 0 & E_2 X_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & E_\mu X_\mu \end{pmatrix}.$$

La matrice $D_\mu$ est inversible si et seulement si $\mu \leq \nu$ (pour $\mu > \nu$, il existe au moins un terme de la diagonale de $D_\mu$ nul) ; comme la matrice $V_\mu$ est inversible en tant que matrice de Vandermonde, la matrice $S_\mu$ est inversible si et seulement si $\mu \leq \nu$. ■

On peut alors synthétiser l'algorithme :

**Méthode** **Algorithme de Peterson**

1. Calculer des syndromes $S_i = r(\alpha_i)$ pour $1 \leq i \leq 2t$ ;
2. À partir de $\mu = t, \mu = t-1, \ldots$, trouver le premier $\mu$ tel que $S_\mu$ est inversible, ce qui donne la valeur de $\nu$ ;
3. Résoudre le système $(S_2)$ en les $\sigma_i$ ;
4. Calculer les emplacements d'erreurs à partir de $\sigma(x)$ par la recherche des racines de $\sigma(x)$ ;
5. Résoudre le système linéaire $(S_1)$ pour retrouver les valeurs des erreurs $E_j$.

**Remarque.** Dans le cas d'un code binaire, l'étape 5 n'est pas nécessaire.

La complexité de ce dernier algorithme est $\mathcal{O}(n^3)$ puisque les points 3 et 5 nécessitent la résolution d'un système d'équations linéaires (voir le chapitre 5 du tome *Mathématiques L2*). En fait, il est possible d'améliorer certaines de ces étapes. Par exemple, la résolution de la partie 3 peut se faire en temps quadratique en $n$ par l'algorithme de Berlekamp-Massey ou par celui d'Euclide étendu que nous détaillons dans la section suivante (voir aussi les chapitres 4 et 5 du tome *Mathématiques L2*).

**Exemple 1.79.** Considérons un code BCH au sens strict primitif de longueur 15 et de distance construite $\delta = 5$ en prenant, pour définir $\mathbb{F}_{16}$, le polynôme $x^4+x+1$ (soit $\alpha^4 = \alpha+1$). Le code est un code $[15, 7, 5]$ de polynôme générateur $g(x) = 1+x^4+x^6+x^7+x^8$. Supposons que l'on reçoive le mot $r(x) = 1+x+x^5+x^6+x^9+x^{10}$. On commence par calculer les syndromes $S_1 = \alpha^2$, $S_2 = \alpha^4$, $S_3 = \alpha^{11}$ et $S_4 = \alpha^8$.

Pour la deuxième étape, on calcule le déterminant de

$$M_2 = \begin{pmatrix} S_1 & S_2 \\ S_2 & S_3 \end{pmatrix} = \begin{pmatrix} \alpha^2 & \alpha^4 \\ \alpha^4 & \alpha^{11} \end{pmatrix}$$

qui est non nul. Il y a donc 2 erreurs. On doit maintenant résoudre le système

$$\begin{pmatrix} S_1 & S_2 \\ S_2 & S_3 \end{pmatrix} \begin{pmatrix} \sigma_2 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} -S_3 \\ -S_4 \end{pmatrix},$$

soit

$$\begin{pmatrix} \alpha^2 & \alpha^4 \\ \alpha^4 & \alpha^{11} \end{pmatrix} \begin{pmatrix} \sigma_2 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} \alpha^{11} \\ \alpha^8 \end{pmatrix}.$$

La solution est $\sigma_1 = \alpha^2$ et $\sigma_2 = \alpha^{14}$ et l'on obtient, pour le polynôme localisateur d'erreurs, $\sigma(x) = 1+\alpha^2 x+\alpha^{14}x^2$, qui s'annule pour $\alpha^{11}$ et $\alpha^5$, d'où $X_1 = \alpha^4$ et $X_2 = \alpha^{10}$. Et donc le mot transmis est $c(x) = r(x)+x^4+x^{10} = 1+x+x^4+x^5+x^6+x^9$.

**Test 1.33.**

Soit $\mathcal{C}$ le code BCH au sens strict binaire primitif $[15, 5, 7]$ avec $\mathbb{F}_{16}$ défini comme dans l'exemple précédent. Quel est le polynôme générateur de $\mathcal{C}$ ? Corriger le vecteur reçu $r(x) = x^2+x^5+x^6+x^9+x^{10}+x^{11}+x^{12}+x^{13}$.

## V.2. Décodage par l'algorithme d'Euclide étendu

On suppose que $\mathcal{C}$ est un code cyclique sur $\mathbb{F}_q$ dont le polynôme générateur a un ensemble de zéros contenant les éléments $\alpha, \alpha^2, \ldots, \alpha^{2t}$, où $\alpha$ désigne une racine $n$-ième de l'unité dans une extension $\mathbb{F}_{q^m}$. On reçoit le mot $r$ (sous forme polynomiale) et les syndromes associés sont les $S_i = r(\alpha^i)$. On a vu lors de l'algorithme précédent que les coefficients du polynôme localisateur d'erreurs $\sigma_t, \sigma_{t-1}, \cdots, \sigma_1, \sigma_0$ avec $\sigma_0 = 1$, étaient solutions du système $(\mathcal{S}_2)$, mais dans l'ordre décroissant des coefficients. Si l'on définit $S(x) = \sum_{i=1}^{2t} S_i x^{2t-i}$ et $\sigma'(x) = x^t \sigma(x^{-1})$ le polynôme réciproque de $\sigma(x)$ (où l'ordre des coefficients est inversé), et que l'on calcule $S(x)\sigma'(x)$, alors le système $(\mathcal{S}_2)$ implique que les termes de degrés $t, t+1, \cdots, 2t-1$ de $S(x)\sigma(x)$ sont tous nuls et donc que $\deg(S(x)\sigma'(x) \pmod{x^{2t}}) < t$. D'autre part, si $\sigma'(x)$ (avec $\deg(\sigma'(x)) \leq t$) vérifie $\deg(S(x)\sigma'(x) \pmod{x^{2t}}) < t$, alors on a une solution de $(\mathcal{S}_2)$.

**Rappel** **Algorithme d'Euclide étendu**

Étant donné deux polynômes $a(x)$ et $b(x)$ de $\mathbb{F}_q[x]$, l'algorithme d'Euclide étendu calcule leur PGCD $d(x)$ et deux polynômes $f(x)$ et $g(x)$ réalisant l'identité de Bézout

$$a(x)f(x) + b(x)g(x) = d(x),$$

avec $\deg(g(x)) < \deg(a(x))$ et $\deg(f(x)) < \deg(b(x))$.
L'algorithme procède par itérations successives de divisions euclidiennes. L'initialisation se fait par $r_{-1}(x) = a(x), f_{-1}(x) = 1, g_{-1}(x) = 0, r_0(x) = b(x), f_0(x) = 0, g_0(x) = 1$, puis pour $i \geq 1$, on effectue la division euclidienne $r_{i-2}(x) = q_i(x)r_{i-1}(x) + r_i(x)$ avec $\deg(r_i(x)) < \deg(r_{i-1}(x))$. On met à jour $f_i(x) = f_{i-2}(x) - q_i(x)f_{i-1}(x)$ et $g_i(x) = g_{i-2}(x) - q_i(x)g_{i-1}(x)$.
On a les résultats suivants à chaque étape :

1. $\mathrm{pgcd}(r_{i-1}(x), r_i(x)) = \mathrm{pgcd}(r_i(x), r_{i+1}(x))$ ;
2. $f_i(x)a(x) + g_i(x)b(x) = r_i(x)$ ;
3. $\deg(g_i(x)) + \deg(r_{i-1}(x)) = \deg(a(x))$ ;

**Théorème 1.80.** *Si l'on utilise l'algorithme d'Euclide étendu en prenant comme polynômes de départ* $S(x)$ *et* $x^{2t}$*, et en s'arrêtant dès que le reste* $r_i(x)$ *est tel que* $\deg(r_{i-i}(x)) \geq t$ *et* $\deg(r_i(x)) < t$*, alors on obtient le polynôme* $\sigma'(x)$ *(et donc* $\sigma(x)$*) par la relation* $\sigma'(x) = g_i(x)$*, en prenant* $g_i(x)$ *monique.*

**Preuve.** Si l'on arrête l'algorithme d'Euclide étendu à l'endroit précisé dans le théorème, on obtient, d'après le rappel sur l'algorithme d'Euclide étendu, $\deg(g_j(x)) = 2t - \deg(r_{j-1}(x)) \leq t$ et $f_j(x)x^{2t} + g_j(x)s(x) = r_j(x)$. Le polynôme $g_j(x)$ est un polynôme de degré au plus $t$ tel que $\deg(S(x)g_j(x) \pmod{x^{2t}}) = \deg(r_j(x) < t$, ce qui implique, d'après ce que l'on a vu plus haut, en tenant compte du fait que $\sigma_t = 1$, que $\sigma'(x) = g_j(x)$ (pris sous forme monique). ∎

La complexité de l'algorithme d'Euclide étendu est en $\mathcal{O}(n^2)$ (voire $\mathcal{O}(n \log(n))$ avec des méthodes avancées), ce qui est plus efficace pour trouver $\sigma(x)$ que l'algorithme classique d'inversion de matrice en $\mathcal{O}(n^3)$. Il existe aussi des méthodes pour accélérer le calcul des $E_j$ que nous ne décrivons pas ici.

**Exemple 1.81.** On considère à nouveau l'exemple 1.79 pour lequel on va appliquer l'algorithme d'Euclide étendu pour retrouver $\sigma(x)$. On a dans ce cas $2t = 4$ et $S(x) = S_1x^3+S_2x^2+S_3x+S_4$, soit, en reprenant les données précédentes $S(x) = \alpha^2x^3+\alpha^4x^2+\alpha^{11}x+\alpha^8$. On applique alors l'algorithme d'Euclide étendu entre $S(x)$ et $x^4$. On initialise l'algorithme par $f_{-1}(x) = 1, g_{-1}(x) = 0$, $r_{-1}(x) = x^4$ et $f_0(x) = 0, g_0(x) = 1$ et enfin $r_0(x) = S(x) = \alpha^2x^3 + \alpha^4x^2 + \alpha^{11}x + \alpha^8$. On effectue la division euclidienne de $r_{-1}(x) = x^4$ par $r_0(x) = S(x) = \alpha^2x^3 + \alpha^4x^2 + \alpha^{11}x + \alpha^8$, soit $r_{-1}(x) = (\alpha^{13}x + 1)r_0(x) + \alpha^{14}x^2 + \alpha x + \alpha^8$, qui donne $r_1 = \alpha^{14}x^2 + \alpha x + \alpha^8$ et $q_1(x) = \alpha^{13}x + 1$. On peut, alors, calculer $f_1(x) = 1$ et $g_1(x) = q_1(x)$. Comme $\deg(r_1) \geqslant t$, on refait une itération de division euclidienne : $r_0(x) = (\alpha^3x)r_1(x) + \alpha x + \alpha^8$, soit $r_2(x) = \alpha x + \alpha^8$ et $q_2(x) = \alpha^3x$. Comme cette fois $\deg(r_2) < t$, on arrête l'algorithme et on calcule $\sigma'(x) = g_2(x) = g_0(x) - q_2(x)g_1(x) = 1 - (\alpha^3x)(\alpha^{13}x + 1) = \alpha x^2 + \alpha^3x + 1$. On obtient alors $g_2(x) = \alpha x^2 + \alpha^3x + 1$ qui, sous forme monique, donne $\sigma'(x) = x^2 + \alpha^2x + \alpha^{14}$, soit $\sigma(x) = \alpha^{14}x^2 + \alpha^2x + 1$, comme dans l'exemple précédent 1.79.

**Test 1.34.**

Montrer qu'on peut retrouver les emplacements d'erreurs directement à partir du polynôme $g_j(x)$ obtenu par l'algorithme d'Euclide étendu sans passer par $\sigma(x)$.

**Test 1.35.**

En reprenant l'énoncé du test 1.33, montrer que le mot reçu conduit à une erreur de poids 2. Appliquer l'algorithme d'Euclide étendu avec $t = 2$ pour retrouver le polynôme localisateur d'erreurs.

# VI. Codes de Reed-Solomon, codes alternants et codes de Goppa

## VI.1. Codes de Reed-Solomon et algorithme de Welch-Berlekamp

Les codes de Reed-Solomon ont été introduits par I. S. Reed et G. Solomon en 1959. En raison de leurs paramètres optimaux et de leur facilité de décodage, ces codes ont été très utilisés dans l'industrie, par exemple pour les CD et DVD ainsi que pour les communications satellites. On a vu à la section IV comment ces codes pouvaient aussi être introduits en tant que codes BCH particuliers. La définition que nous donnons maintenant correspond à la définition originelle de ces codes, qui n'utilise pas la notion de cyclicité. On renvoie à l'exercice 1.13 pour la preuve que les codes de Reed-Solomon cycliques sont un cas particulier de la famille des codes que nous présentons maintenant.

**Définition 1.82. (Code de Reed-Solomon)** *Soit $\mathbb{F}_q$ un corps et soient $x_1, \cdots, x_n$, $n$ éléments distincts de $\mathbb{F}_q^*$. Pour $k \leq n$, on considère l'ensemble $\mathcal{P}_k$ des polynômes de $\mathbb{F}_q[x]$ de degré au plus $k-1$. Un code de Reed-Solomon est composé de l'ensemble des mots $c(f) = (f(x_1), \cdots, f(x_n))$ pour $f \in \mathcal{P}_k$.*
*On parlera de code $RS(n,k)_q$ pour désigner un code de Reed-Solomon de longueur $n$ sur $\mathbb{F}_q$ et de paramètre $k$.*

On a alors le théorème suivant.

**Théorème 1.83.** *Les codes de Reed-Solomon $RS(n,k)_q$ sont des codes $[n, k, n-k+1]_q$.*

**Preuve.** Par définition, les codes $RS(n,k)_q$ sont des codes de longueur $n$ ; la linéarité des codes découle de la linéarité de l'addition de polynômes dans $\mathcal{P}_k$. Le nombre de mots d'un code RS est inférieur ou égal au nombre de polynômes de $\mathcal{P}_k$, c'est-à-dire $q^k$. Supposons que deux

polynômes $f$ et $g$ de $\mathcal{P}_k$ donnent le même mot du code $c(f) = c(g)$ ; alors, on a $(f-g)(x) = 0$ pour tout $x \in x_1, \cdots, x_n$. Le polynôme $f-g$ est de degré au plus $k-1$ et s'annule alors en $n$ points. Comme $n > k-1 \leq \deg(f-g)$, on en déduit que $f-g = 0$ et donc que $f = g$, ce qui montre que la dimension du code est précisément $k$.

Montrons maintenant que la distance minimale est $n-k+1$. Soit $c(f)$ un mot du code non nul (i.e. $f \neq 0$), alors $c(f) = (f(x_1), \cdots, f(x_n))$ pour $\deg(f) \leq k-1$. Si $f$ s'annule en un nombre de points strictement supérieur à $\deg(f)$, alors $f$ est nécessairement le polynôme nul ; comme $\deg(f) \leq k-1$ et que $f \neq 0$, $f$ peut donc s'annuler en $k-1$ points au plus. On en déduit que $w_H(c(f)) \geq n-(k-1) = n-k+1$, soit $d \geqslant n-k+1$. Maintenant, la borne de Singleton assure que $d \leqslant n-k+1$ ; on a donc exactement $d = n-k+1$. ∎

**Remarque.** Comme les codes de Reed-Solomon satisfont l'égalité dans la borne de Singleton, ils sont MDS.

Nous présentons ici un algorithme de décodage élémentaire de ces codes.

Soit $r = c + e$ un mot reçu tel que $w_H(e) \leq t$ avec $t = \lfloor \frac{d-1}{2} \rfloor$. Le décodage utilise un polynôme bivarié particulier $Q(x,y)$ de la forme $Q(x,y) = Q_0(x) + yQ_1(y)$, tel que

1. $Q(x_i, r_i) = 0, i \in \{1, \cdots, n\}$ ;
2. $\deg(Q_0) \leq n-1-t$ ;
3. $\deg(Q_1) \leq n-1-t-(k-1)$ ;

Ce polynôme est en fait un polynôme d'interpolation (voir le chapitre 2) bivarié sur les points $(x_i, r_i)$.

**Proposition 1.84.** *Il existe au moins un polynôme $Q(x,y)$ non nul vérifiant les conditions précédentes.*

**Preuve.** Le polynôme $Q(x,y)$ peut être décrit à partir de ses degrés en $Q_0(x)$ et $Q_1(x)$ en prenant pour inconnues les coefficients de $Q_0$ et $Q_1$. On va simplement compter le nombre de contraintes et le nombre de degrés de liberté. La condition 1 donne $n$ contraintes linéaires ; les conditions 2 et 3 fixent les degrés de $Q_0$ et $Q_1$, ce qui donne $\deg(Q_0)+1+\deg(Q_1)+1$ degrés de libertés (les coefficients des monômes de $Q_0$ et $Q_1$). Il y a donc $n-1-t+1+n-1-t-(k-1)+1 = n-k-2t+1+n+1$ inconnues. Mais, comme $t = \lfloor (n-k+1)/2 \rfloor$, le nombre de degrés de liberté est supérieur ou égal à $n+1$. Comme le nombre $n+1$ d'inconnues est supérieur au nombre $n$ de contraintes, on est sûr qu'il existe au moins un tel polynôme $Q(x,y)$ non nul obtenu en résolvant un système à $n$ équations et $n+1$ inconnues. ∎

On peut alors en déduire comment décoder les codes de Reed-Solomon.

**Théorème 1.85.** *Soit $c = c(f)$, un mot d'un code $RS_(n,k)_q$, construit à partir de $f(x)$ avec $\deg(f) \leq k-1$. Soit maintenant un mot reçu $r = c + e$ avec une erreur $e$ de poids $w_H(e) \leqslant t = \lfloor \frac{d-1}{2} \rfloor$. Si l'on construit le polynôme $Q(x,y) = Q_0(x) + yQ_1(y)$, le polynôme d'interpolation bivarié défini précédemment à partir du mot reçu $r$, alors on peut retrouver le polynôme $f(x)$ associé à $c(f)$ par $f(x) = \frac{-Q_0(x)}{Q_1(x)}$.*

**Preuve.** Soient $c(f) = (f(x_1), \cdots, f(x_n))$ et $r = c+e$ pour $w_H(e) \leq t = \lfloor \frac{d-1}{2} \rfloor$. Par définition, on a $Q(x_i, r_i) = 0$ pour tout $i \in [1,..,n]$, mais $r_i = f(x_i) + e_i$ et donc $r_i = f(x_i)$ lorsque $e_i = 0$, ce qui se produit en au moins $n-t$ coordonnées. Ainsi le polynôme univarié $Q(x, f(x))$ a au moins $n-t$ zéros distincts. Mais, comme $\deg(Q_0(x)) \leq n-1-t$ et $\deg(f(x)Q_1(x)) \leq (k-1)+n-1-t-(k-1) = n-1-t$, on a $\deg(Q(x, f(x)) \leq n-1-t$. Le polynôme $Q(x, f(x))$

est de degré au plus $n-t-1$ et s'annule en au moins $n-t$ valeurs. C'est donc le polynôme nul, d'où $Q(x,f(x))=0$ et donc $Q_0(x)+f(x)Q_1(x)=0$, ce qui implique que $f(x)=-\frac{Q_0(x)}{Q_1(x)}$. ■

On en déduit l'algorithme de décodage suivant

**Méthode** **Algorithme de Welch-Berlekamp**

1. Pour un mot reçu $r$, construire le polynôme interpolé $Q(x,y)=Q_0(x)+yQ_1(x)$.
2. Décoder par division de $Q_0(x)$ par $Q_1(x)$.

Ce décodage est polynomial en $n$ puisque ces deux étapes peuvent s'effectuer en temps polynomial en $n$ ; plus précisément, la première étape (la plus coûteuse) peut se faire dans le pire des cas par inversion d'une matrice $n\times n$, soit en $\mathcal{O}(n^3)$ opérations. En fait, il existe des méthodes (que nous ne décrivons pas ici) qui permettent de le faire en $\mathcal{O}(n^2)$.

**Exemple 1.86.** On considère un code de Reed-Solomon $[6,4,3]_7$ défini par les points $x_1,..,x_6$, avec $x_i=3^{i-1}$. Une matrice génératrice $G$ s'obtient en évaluant les $x_i$ pour les polynômes $1,x,x^2$ et $x^3$, soit

$$G=\begin{bmatrix}1&1&1&1&1&1\\1&3&2&6&4&5\\1&2&4&1&2&4\\1&6&1&6&1&6\end{bmatrix}.$$

Comme $d=3$, on peut décoder au plus une erreur. On veut maintenant décoder le mot reçu $r=(r_1,r_2,...,r_6)=(3,1,1,6,3,3)$ avec l'algorithme de Welch-Berlekamp.

Dans ce cas, le polynôme $Q_0(x)$ a degré au plus $n-1-t=6-1-1=4$ et le polynôme $Q_1(x)$ a degré $n-1-t-(k-1)=1$. On cherche donc un polynôme, non nul, $Q(x,y)=a_0+a_1x+a_2x^2+a_3x^3+a_4x^4+y(b_0+b_1x)$, tel que $Q(x_i,r_i)=0$ pour $1\leqslant i\leqslant 6$. En remplaçant les $x_i$ par $3^{i-1}$ et les $r_i$ par leur valeur dans $Q(x,y)$, pour $1\leqslant i\leqslant 6$, on obtient un système avec 6 équations et 7 inconnues : $a_0,a_1,...,a_4,b_0,b_1$. La résolution du système donne $Q_0(x)=3x+3x^2+6x^3$ et $Q_1(x)=2+x$, soit $f(x)=-\frac{Q_0(x)}{Q_1(x)}=x^2+x$. En évaluant $f$ sur $(x_1,x_2,x_3,x_4,x_5,x_6)=(1,3,2,6,4,5)$, on obtient comme mot décodé $c(f)=(3,1,1,6,3,0)$ et comme erreur $(0,0,0,0,0,3)$.

**Test 1.36.**

On considère le code de Reed-Solomon $[6,3,4]_7$, défini sur $\mathbb{F}_7$ par les points $x_1,..,x_6$ avec $x_i=3^{i-1}$. Donner une matrice génératrice du code. Décoder le mot reçu $r=(6,6,5,2,1,2)$ par l'algorithme de Welch-Berlekamp.

## VI.2. Codes alternants

Les codes alternants sont reliés aux codes de Reed-Solomon par le biais des codes de Reed-Solomon généralisés (RSG).

**Définition 1.87. (Codes de Reed-Solomon généralisés)** *Un code de Reed-Solomon généralisé* $RS_k(\alpha,v)$ *sur* $\mathbb{F}_q$ *de longueur* $n\leqslant q-1$ *et de dimension* $k$ *est défini par un vecteur* $v=(v_1,\cdots,v_n)$ *d'éléments non nuls de* $\mathbb{F}_q$ *et un ensemble* $\alpha=(\alpha_1,\cdots,\alpha_n)$ *d'éléments distincts non nuls de* $\mathbb{F}_q$. *Alors,*

$$RS(\alpha,v)=\{(v_1f(\alpha_1),\cdots,v_nf(\alpha_n))\,|\,f\in\mathbb{F}_q[x],\deg(f)<k\}.$$

Les codes de Reed-Solomon généralisés correspondent donc à un code de Reed-Solomon dont on multiplie les colonnes par des éléments non nuls de $\mathbb{F}_q$. La multiplication d'une colonne par un élément non nul ne modifiant pas la distance minimale, les codes de Reed-Solomon généralisés ont les mêmes paramètres que les codes de Reed-Solomon et sont MDS.

**Définition 1.88. (Codes alternants)** *On note* $RS_{n-r}(\alpha, \nu)$ *un code de Reed-Solomon généralisé* $[n, n-r, r+1]_{q^m}$ *construit sur une extension* $\mathbb{F}_{q^m}$ *de* $\mathbb{F}_q$.
*Un code alternant* $\mathcal{A}_k(\alpha, \nu)$ *est défini comme l'ensemble des mots d'un code de Reed-Solomon généralisé* $RS_{n-r}(\alpha, \nu)$*, dont les coordonnées sont à coefficients dans le sous-corps* $\mathbb{F}_q$.

**Proposition 1.89.** *Soit* $RS_{n-r}(\alpha, \nu)$ *un code de Reed-Solomon généralisé* $[n, n-r, r+1]_{q^m}$ *construit sur une extension* $\mathbb{F}_{q^m}$ *de* $\mathbb{F}_q$. *Le code alternant* $\mathcal{A}_k(\alpha, \nu)$ *correspondant (sur* $\mathbb{F}_q$*) est un code* $[n, k, d]_q$ *avec* $n - mr \leq k \leq n - r$ *et* $r + 1 \leq d$.

**Preuve.**

Le code alternant est un code linéaire puisque la somme de mots du sous-code sur $\mathbb{F}_q$ est stable par addition et multiplication externe. Le sous-code a une distance minimale au moins égale à celle du code. Soit $H = \{H_{ij}\}_{1 \leq i \leq n-r, 1 \leq j \leq n}$ une matrice de contrôle de parité du code $RS_{n-r}(\alpha, \nu)$ de paramètres $[n, n-r, r+1]_{q^m}$. On considère une base $\{e_1, \cdots, e_m\}$ de $\mathbb{F}_{q^m}$ sur $\mathbb{F}_q$. Chacun des $H_{ij}$ peut s'écrire comme $H_{ij} = \sum_{k=1}^{m} H_{ijk} e_k$. On peut alors définir une matrice $H'$ $rm \times n$ à partir de $H$, en écrivant chacun des $H_{ij}$ en colonne dans la base $\{b_1, \ldots, b_m\}$. Comme, par définition, les mots du code alternant $\mathcal{A}_k(\alpha, \nu)$ sont les mots de $\mathbb{F}_q^n$ contenus dans $RS_{n-r}(\alpha, \nu)$, on a pour $c \in \mathbb{F}_q^n$

$$c \in \mathcal{A}_k(\alpha, \nu) \Leftrightarrow Hc^t = 0.$$

Mais comme $c \in \mathbb{F}_q^n$, l'équivalence précédente reste vraie lorsque l'on développe $\mathbb{F}_{q^m}$ sur une base de $\mathbb{F}_q$, soit pour $c \in \mathbb{F}_q^n$

$$Hc^t = 0 \Leftrightarrow H'c^t = 0.$$

La matrice $H'$ engendre donc une matrice duale de $\mathcal{A}_k(\alpha, \nu)$ ; comme elle a $rm$ lignes, elle a pour rang au plus $rm$ et donc la dimension du code alternant est supérieure à $n - rm$ et inférieure à $n - r$ (la dimension du code sur l'extension). ■

En pratique, la dimension des codes alternants reste proche de $n - rm$. La classe générale des codes alternants peut se décoder en tant que sous-code d'un code de Reed-Solomon généralisé, qui se décode à partir des codes de Reed-Solomon (en multipliant par $\nu^{-1} = (\nu_1^{-1}, \cdots, \nu_n^{-1})$). La classe des codes alternants contient un grand nombre de sous-classes avec des paramètres et des décodages spécifiques.

## VI.3. Codes de Goppa binaires

Les codes de Goppa ont été introduits par V. D. Goppa en 1970. Ils forment une classe particulière des codes alternants et peuvent être vus comme une généralisation des codes BCH. Ils sont en particulier très bons en caractéristique 2, où ils ont le même type de paramètres que les codes BCH, mais sont beaucoup plus nombreux pour des paramètres donnés. Cette dernière propriété en fait des codes très intéressants pour une utilisation en cryptographie dans le cryptosystème de chiffrement à clé publique de R. McEliece (voir le chapitre suivant).

### VI.3.1. Une autre construction pour les codes BCH au sens strict

Soit $t = \mathrm{ord}_q(n)$ et soit $\alpha$ une racine $n$-ième de l'unité dans $\mathbb{F}_{q^t}$. Soit maintenant $\delta > 1$ la distance construite pour un code BCH au sens strict $\mathcal{C}$. On voit alors, en reprenant la section IV.3 sur les codes BCH, que $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ est dans $\mathcal{C}$ si et seulement si $c(\alpha^i) = 0$ pour tout $1 \leq i \leq \delta - 1$. On remarque que l'on peut écrire

$$(x^n - 1)\sum_{i=0}^{n-1} \frac{c_i}{x - \alpha^{-i}} = \sum_{i=0}^{n-1} c_i \sum_{k=0}^{n-1} x^k (\alpha^{-i})^{n-1-k} = \sum_{k=0}^{n-1} x^k \sum_{i=0}^{n-1} c_i (\alpha^{k+1})^i. \tag{1.1}$$

Comme $c(\alpha^i) = 0$ pour $1 \leq i \leq \delta - 1$, la somme $\displaystyle\sum_{i=0}^{n-1} c_i(\alpha^{k+1})^i$ est nulle pour $0 \leq k \leq \delta - 2$, et donc le dernier membre de l'égalité précédente peut s'écrire comme un certain polynôme $q(x)$ fois $x^{\delta-1}$. La dernière égalité devient donc

$$\sum_{i=0}^{n-1} \frac{c_i}{x - \alpha^{-i}} = \frac{x^{\delta-1} q(x)}{x^n - 1}, \tag{1.2}$$

que l'on peut aussi voir comme vérifiant l'égalité

$$\sum_{i=0}^{n-1} \frac{c_i}{x - \alpha^{-i}} \equiv 0 \pmod{x^{\delta-1}}, \tag{1.3}$$

où le modulo est considéré pour le polynôme $a(x)$ lorsque l'on écrit le membre gauche de l'égalité 1.2 sous forme de fraction rationnelle $\frac{a(x)}{b(x)}$ (avec $a(x)$ et $b(x)$ premiers entre eux).

L'égalité 1.3 (dérivée de l'égalité 1.1) montre que les mots de $\mathbb{F}_q$ qui la vérifient sont exactement les mots $c(x)$ dont les coefficients sont tels que $c(\alpha^i) = 0$ pour $1 \leqslant i \leqslant \delta - 1$. Et donc le code est le code BCH au sens strict de distance construite $\delta$.

Cette égalité peut être généralisée en prenant un autre polynôme que le polynôme $x^{\delta-1}$. Ce point de vue permet d'introduire les codes de Goppa.

### VI.3.2. Construction des codes de Goppa

Les codes de Goppa $\Gamma(L, G)$ de longueur $n$ sur $\mathbb{F}_q$ sont définis à partir d'un polynôme $G(x)$ de degré $r$, à coefficients dans une extension $\mathbb{F}_{q^m}$ de $\mathbb{F}_q$ pour un $m$ quelconque, ainsi que d'un ensemble $L = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ de $n$ éléments de $\mathbb{F}_{q^m}$ qui sont tels que $G(\alpha_i) \neq 0$ pour $\alpha_i \in L$. En géneral, on prend pour $L$ l'ensemble de tous les éléments de $\mathbb{F}_{q^m}$ qui ne sont pas racines de $G$. Le polynôme $G$ est appelé *polynôme de Goppa*.

**Définition 1.90. (Code de Goppa)** *Avec les notations ci-dessus, le code de Goppa $\Gamma(L, G)$ est composé de tous les mots $c(c_1, c_2, \cdots, c_n) \in \mathbb{F}_q$ vérifiant*

$$R_c(x) \stackrel{\text{déf}}{=} \sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} \equiv 0 \pmod{G(x)}.$$

*Lorsque le polynôme $G(x)$ est irréductible, on dit que le code est un code de Goppa irréductible.*

L'égalité modulo $G(x)$ est considérée comme précédemment en termes de fraction rationnelle par rapport à $a(x)$ pour $R_c(x) = \frac{a(x)}{b(x)}$ et $a(x)$ et $b(x)$ premiers entre eux.

**Théorème 1.91.** *Le code de Goppa $\Gamma(L, G)$ de longueur $n$ sur $\mathbb{F}_q$, défini à partir d'un polynôme $G(x)$ de degré $r$, à coefficients dans une extension $\mathbb{F}_{q^m}$ de $\mathbb{F}_q$ pour un $m$ quelconque, est un code linéaire de longueur $n$, de dimension $k \geq n - mr$ et de distance minimale $d \geq r+1$. Dans le cas binaire ($q = 2$), si $G(x)$ n'a pas de racine multiple, alors $d \geq 2r + 1$.*

**Preuve.** Le code $\Gamma(L,G)$ est clairement linéaire : si $c \in \Gamma(L,G)$, alors on a $\gamma c \in \Gamma(L,G)$ et si $c_1, c_2 \in \Gamma(L,G)$, alors on a $c_1 + c_2 \in \Gamma(L,G)$ d'après les propriétés de somme modulo $G(x)$ intervenant pour le calcul de $R_c(x)$. On va maintenant calculer le dual de $\Gamma(L,G)$. On remarque tout d'abord que

$$\frac{1}{x-\alpha_i} \equiv -G(\alpha_i)^{-1}\frac{(G(x)-G(\alpha_i))}{x-\alpha_i} \pmod{G(x)},$$

puisque $-G(\alpha_i)^{-1}(G(x)-G(\alpha_i)) = 1 - G(\alpha_i)G(x) \equiv 1 \pmod{G(x)}$. On obtient donc

$$c \in \Gamma \Leftrightarrow \sum_{i=1}^{n} c_i \frac{(G(x)-G(\alpha_i))}{x-\alpha_i} G(\alpha_i)^{-1} \equiv 0 \pmod{G(x)}. \tag{1.4}$$

Si l'on écrit $G(x) = \sum_{i=0}^{r} g_i x^i$ avec $g_i \in \mathbb{F}_{q^m}$ et $g_r \neq 0$, alors

$$\begin{aligned}\frac{(G(x)-G(\alpha_i))}{x-\alpha_i} G(\alpha_i)^{-1} &= g_r(x^{r-1} + x^{r-2}\alpha_i + \cdots + \alpha_i^{r-1}) \\ &\quad + g_{r-1}(x^{r-2} + x^{r-3}\alpha_i + \cdots + \alpha_i^{r-2}) + \cdots + g_2(x+\alpha_i) + g_1.\end{aligned}$$

En mettant les coefficients de $x^i$ à zéro dans 1.4, on obtient que $c \in \Gamma$ si et seulement si $H.c^t = 0$ avec

$$H = \begin{pmatrix} g_r G(\alpha_1)^{-1} & \cdots & g_r G(\alpha_n)^{-1} \\ (g_{r-1} + \alpha_1 g_r)G(\alpha_1)^{-1} & \cdots & (g_{r-1} + \alpha_n g_r)G(\alpha_n)^{-1} \\ & \cdots & \\ (g_1 + \alpha_1 g_2 + \alpha_1^{r-1} g_r)G(\alpha_1)^{-1} & \cdots & (g_1 + \alpha_1 g_2 + \alpha_1^{r-1} g_r)G(\alpha_n)^{-1} \end{pmatrix}.$$

Mais on peut décomposer $H$ sous la forme

$$H = \begin{pmatrix} g_r & 0 & 0 & \cdots & 0 \\ g_{r-1} & g_r & 0 & \cdots & 0 \\ g_{r-2} & g_{r-1} & g_r & \cdots & 0 \\ & \cdots & \cdots & & \\ g_1 & g_2 & g_3 & \cdots & g_r \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ & \cdots & \cdots & \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \cdots & \alpha_n^{r-1} \end{pmatrix}$$
$$\cdot \begin{pmatrix} G(\alpha_1)^{-1} & 0 & \cdots & 0 \\ 0 & G(\alpha_2)^{-1} & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & G(\alpha_n)^{-1} \end{pmatrix}.$$

Comme la première des trois matrices est une matrice carrée inversible (puisque $g_r \neq 0$), on en déduit qu'une matrice de contrôle de parité $H$ peut aussi s'écrire comme le produit des deux dernières matrices qui peut s'écrire sous la forme

$$\mathcal{H} = \begin{pmatrix} G(\alpha_1)^{-1} & \cdots & G(\alpha_n)^{-1} \\ \alpha_1 G(\alpha_1)^{-1} & \cdots & \alpha_n G(\alpha_n)^{-1} \\ & \cdots & \cdots \\ \alpha_1^{r-1} G(\alpha_1)^{-1} & \cdots & \alpha_n^{r-1} G(\alpha_n)^{-1} \end{pmatrix}.$$

Cette dernière écriture montre que le code de Goppa $\Gamma(L,G)$ est un code alternant avec un ensemble de points $\alpha = (\alpha_1, \ldots, \alpha_n)$ et $y = (G(\alpha_1)^{-1}, \ldots, G(\alpha_n)^{-1})$). On peut donc en déduire d'après les résultats sur les codes alternants que le code $\Gamma(L,G)$ a pour dimension $k \geq n - rm$ et $d \geq r+1$.

De manière similaire au cas des codes BCH binaires qui sont relativement meilleurs que les codes BCH généraux, on peut aussi faire mieux avec des codes de Goppa binaires. Soit $\Gamma(L, G)$ un code de Gopa binaire et soit $c(c_1, \ldots, c_n)$ un mot du code, non nul, de poids $w$ avec $c_{l_1} = \cdots = c_{l_w} = 1$. On introduit

$$f_c(x) \stackrel{déf}{=} \Pi_{i=1}^{w}(x - \alpha_{l_i}).$$

Si l'on dérive par rapport à $x$, on obtient

$$f_c'(x) = \sum_{i=1}^{w} \Pi_{j \neq i}(x - \alpha_{l_j})$$

et

$$R_c(x) = \sum_{i=1}^{w} \frac{1}{x - \alpha_{l_i}} = \frac{f_c'(x)}{f_c(x)}.$$

Par définition du code, les $\alpha_i$ sont distincts et donc $f_c'(x)$ et $f_c(x)$ n'ont pas de facteurs communs ; de plus, comme les $\alpha_i$ ne sont pas racines de $G(x)$, les polynômes $f_c(x)$ et $G(x)$ n'ont pas de facteur commun et sont premiers entre eux. On obtient donc

$$R_c(x) \equiv 0 \pmod{G(x)} \quad \text{si et seulement si} \quad G(x) | f_c'(x).$$

Comme on travaille sur une extension de $\mathbb{F}_2$, $f_c'(x)$ ne contient que des puissances paires et peut s'écrire sous la forme d'un carré parfait $f_c'(x) = p(x)^2$, d'un certain polynôme $p(x)$. En effet, $f_c'(x)$ est un polynôme de la forme $f_c'(x) = \sum_{i=0}^{l} a_i x^{2i}$ avec $a_i \in \mathbb{F}_{2^m}$ et, comme on est sur une extension de $\mathbb{F}_2$, tout élément $a_i$ de $\mathbb{F}_{2^m}$ peut s'écrire sous la forme d'un carré $a_i = b_i^2$ pour $b_i \in \mathbb{F}_{2^m}$, d'où $f_c'(x) = (\sum_{i=0}^{l} b_i x^i)^2$.

Comme $f_c'(x)$ est un carré parfait et que $G(x)$ divise $f_c'(x)$, alors, $f_c'(x)$ est aussi divisible par le polynôme de plus petit degré, qui est à la fois un carré parfait et est divisible par $G(x)$. Dans le cas où $G(x)$ n'a pas de racine multiple, le plus petit polynôme qui est un carré parfait divisible par $G(x)$ est $G(x)^2$. Donc, $G(x)^2 \mid f_c'(x)$ et, comme $\deg(f_c'(x)) = w - 1$ et $\deg(G(x)^2) = 2r$, il vient $w - 1 \geq 2r$ pour tout mot $c$ du code non nul. Dans le cas d'un code de Goppa binaire avec $G(x)$ sans racine multiple, on en déduit que $d \geq 2r + 1$. ■

Les codes de Goppa non binaires peuvent se décoder en tant que codes alternants. Pour le cas binaire, il existe un algorithme spécifique de N. J. Patterson que nous ne décrivons pas ici. Toujours pour le cas binaire, si l'on prend un polynôme $G(x)$ irréductible sur $\mathbb{F}_{q^m}$ (et il y en a beaucoup !), on obtient un polynôme sans racine multiple.

**Exemple 1.92.** Soit le corps $\mathbb{F}_8$ construit comme extension de $\mathbb{F}_2$ par le polynôme $x^3 + x + 1$. Soit $\alpha$ une racine primitive de $\mathbb{F}_8$. On veut construire le code de Goppa binaire avec pour L, l'ensemble des éléments de $\mathbb{F}_8$, pris dans l'ordre $\{\alpha, \alpha^2, ..., \alpha^6, 1, 0\}$ (avec $\alpha^7 = 1$) et pour polynôme $G(x) = 1 + \alpha x + x^2$. On vérifie tout d'abord que $G(x)$ ne s'annule pas sur $\mathbb{F}_8$. On construit alors la matrice $\mathcal{H}$ du dual sur $\mathbb{F}_8$ du code alternant induit par le polynôme de Goppa, donnée par

$$\mathcal{H} = \begin{pmatrix} G(\alpha)^{-1} & G(\alpha^2)^{-1} & \ldots & G(\alpha^6)^{-1} & G(1)^{-1} & G(0)^{-1} \\ \alpha G(\alpha)^{-1} & \alpha^2 G(\alpha^2)^{-1} & \ldots & \alpha^6 G(\alpha^6)^{-1} & 1.G(1)^{-1} & 0.G(0)^{-1} \end{pmatrix},$$

soit

$$\mathcal{H} = \begin{pmatrix} 1 & \alpha^5 & \alpha^6 & \alpha^5 & \alpha^2 & \alpha^2 & \alpha^6 & 1 \\ \alpha & 1 & \alpha^2 & \alpha^2 & 1 & \alpha & \alpha^6 & 0 \end{pmatrix}.$$

Pour obtenir le sous-code sur le sous corps, on choisit une base de $\mathbb{F}_8$ sur $\mathbb{F}_2$, comme par exemple $\{1, \alpha, \alpha^2\}$, et on écrit les éléments de $\mathcal{H}$ représentés en binaire et en colonne dans cette

base. On obtient dans cette base $\alpha^3 = 1+\alpha, \alpha^4 = \alpha+\alpha^2, \alpha^5 = 1+\alpha+\alpha^2$ et $\alpha^6 = 1+\alpha^2$, ce qui donne, pour la matrice du dual binaire

$$\mathcal{H}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Par exemple l'avant-dernière colonne correspond bien à $\alpha^6$, écrit deux fois en colonne dans la base $\{1, \alpha, \alpha^2\}$. Le code a pour dimension $k \geq 8-3.2 = 2$ ; comme le polynôme $G(x)$ n'a pas de zéro multiple, on doit trouver $d \geq 2*2+1 = 5$. Le calcul du code donne un code de dimension 2 engendré par les mots $(1,0,1,0,1,1,1,1)$ et $(0,1,1,1,1,0,0,1)$, soit un code binaire $[8,2,5]$.

**Test 1.37.**

En reprenant le même corps $\mathbb{F}_8$ que dans l'exemple, construire le code de Goppa binaire de polynôme $G(x) = 1+x+x^2$.

## VII. Décodage en liste des codes de Reed-Solomon

### VII.1. Introduction au décodage en liste

On a vu dans la section II que, pour un code $[n,k,d]_q$, on pouvait décoder de manière univoque jusqu'à la distance $t = \lfloor \frac{d-1}{2} \rfloor$. Mais que se passe-t-il pour une erreur $e$ de poids $\tau$ plus grande que $t$ ? Prenons par exemple le code $\mathcal{C}$ $[5,2,2]$ de matrice génératrice

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

Comme $d = 2$, on a $t = \lfloor (2-1)/2 \rfloor = 0$ et il n'y a pas de décodage univoque pour une erreur quelconque de poids 1 (en fait pour aucun poids d'erreur donné) (voir le théorème 1.26)

Considérons le mot reçu 11000 ; ce mot peut se décoder de manière univoque en 11100 avec une erreur de poids 1. Prenons maintenant le mot 00010 ; ce mot peut se décoder en 00000 ou 00011 avec une erreur de poids 1. On voit donc qu'au-delà de $t$, plusieurs cas peuvent se produire : certains mots auront un décodage univoque et d'autres non.

Considérons un mot $y$ de l'espace à distance $\tau$ d'un mot d'un code $[n,k,d]_q$. Si $\tau \leq t = \lfloor \frac{d-1}{2} \rfloor$, le nombre de mots du code à distance inférieure ou égale à $\tau$ sera 1. Si maintenant l'on prend $\tau$ tendant vers $n$, le nombre de mots du code à distance inférieure à $\tau$ tendra vers tous les mots du code, soit un nombre exponentiel $q^k$. Entre ces deux zones, $\tau \leq t$ et $\tau$ de l'ordre de $n$, il existe une zone dans laquelle le nombre de mots du code à distance inférieure à $\tau$ ne sera ni fixe ni exponentiel. En fait, on peut montrer qu'il existe une zone dans laquelle le nombre de mots est polynomial en $n$.

Le *décodage en liste* introduit par P. Elias en 1957 consiste, pour un mot de l'espace, à renvoyer une liste de tous les mots du code à distance inférieure ou égale à $\tau$. La longueur de la liste peut être variable, mais a priori elle n'est pas trop grande.

En pratique, surtout pour les codes avec une grande distance minimale, cela permet souvent de décoder pour des erreurs de poids plus important que $\lfloor (d-1)/2 \rfloor$ : la probabilité de mal décoder existe, mais elle est très faible pour $\tau$ proche de $\lfloor (d-1)/2 \rfloor$ et elle grandit avec $\tau$.
Par exemple, prenons le code de Reed-Muller $\mathcal{R}(1,10)$ de paramètre $[1024, 11, 512]$. Si l'on s'en

tient au décodage classique, on peut décoder de manière univoque jusqu'à $w(e) = t = \lfloor (512-1)/2 \rfloor = 255$. Pourtant, en pratique, si l'on calcule de manière combinatoire la probabilité qu'un mot avec une erreur de poids donné $\tau$ soit décodable de manière univoque, on trouve que la probabilité qu'un mot avec une erreur aléatoire de poids 410 soit décodable de manière univoque est de l'ordre de $10^{-9}$ !
Lorsque l'on fait du décodage à distance bornée, on est sûr de décoder l'erreur si son poids est inférieur à une distance donnée (en général $\lfloor \frac{d-1}{2} \rfloor$). Maintenant, si l'on est prêt à admettre une probabilité d'erreur dans le décodage pour une erreur de poids fixé, il est possible de décoder beaucoup plus loin.

La première famille de codes pour laquelle il a été possible de trouver un décodage en liste avec une complexité polynomiale a été la famille des codes de Reed-Solomon pour laquelle M. Sudan a proposé un algorithme en 1996. Sa méthode a été généralisée avec de meilleurs résultats par V. Guruswami et M. Sudan en 1999. Par la suite, des algorithmes ont aussi été proposés pour les codes de Reed-Muller et certains codes alternants. Dans ce qui suit, nous présentons le premier algorithme de Sudan.

## VII.2. Algorithme de Sudan

L'algorithme de Sudan peut être vu comme une généralisation de de l'algorithme de Welch-Berlekamp décrit dans la section V. On reprend les notations du code de Reed-Solomon de la section précédente. On considère un code de Reed-Solomon sur $\mathbb{F}_q$, $[n, k, d]_q$ avec $d = n-k+1$. Soient $c = (c_1, \ldots, c_n) = (f(x_1), \ldots, f(x_n))$ le mot émis et $r = (r_1, \ldots, r_n) = c + e$ le mot reçu avec $e$ qui est un vecteur d'erreurs de poids au plus $\tau$.

L'algorithme utilise à nouveau un polynôme bivarié, mais de degré plus élevé en $y$. On cherche ici un polynôme bivarié $Q(x, y) = Q_0(x) + Q_1(x)y + Q_2(x)y^2 + \ldots + Q_l(x)y^l$, où $l$ est un entier, vérifiant :

1. $Q(x_i, r_i) = 0$, $\forall i \in \{1 \ldots n\}$ ;
2. $\deg(Q_j(x)) \leq n - \tau - 1 - j(k-1)$, $\forall j \in \{0, 1, \cdots l\}$ ;
3. $Q(x, y) \neq 0$.

Dans l'algorithme, $l$ est la taille de la liste renvoyée, c'est-à-dire le nombre de mots potentiellement solutions au problème de décodage. Le problème de trouver $Q(x, y)$ est un problème d'interpolation.

**Proposition 1.93.** *Si* $\tau < n\frac{l}{l+1} - \frac{l}{2}(k-1)$ *et* $n - \tau - 1 - l(k-1) \geq 0$, *alors un tel polynôme* $Q(x, y)$ *existe.*

**Preuve.** La première condition sur $Q(x, y)$ donne $n$ contraintes. La deuxième condition donne des degrés de liberté qui correspondent aux différents coefficients des polynômes $Q_i$. Le polynôme $Q(x, y)$ peut s'obtenir en résolvant un système en les coefficients des $Q_j(x)$. Plus précisément, chaque polynôme $Q_j$ a degré au plus $n - \tau - 1 - j(k-1)$, ce qui donne $1 + n - \tau - 1 - j(k-1)$ degrés de libertés. Un tel polynôme $Q(x, y)$ non nul existera donc si le nombre de degrés de liberté est strictement supérieur au nombre de contraintes, soit si

$$\sum_{j=0}^{l} (1 + n - \tau - 1 - j(k-1)) > n.$$

Le premier membre de l'inégalité peut se simplifier en

$$\sum_{j=0}^{l} (1 + n - \tau - 1 - j(k-1)) = \sum_{j=0}^{l} (n - \tau) - \sum_{j=0}^{l} j(k-1) = (l+1)(n-\tau) - \frac{1}{2}l(l+1)(k-1).$$

On peut donc trouver un tel polynôme si

$$(l+1)(n-\tau) - \frac{1}{2}l(l+1)(k-1) > n,$$

soit

$$(l+1)(n-\tau) > n + \frac{l}{2}(l+1)(k-1) \Leftrightarrow n - \tau > \frac{n}{l+1} + \frac{l}{2}(k-1)$$

qui devient

$$\tau < -\frac{n-n(l+1)}{l+1} - \frac{l}{2}(k-1) \Leftrightarrow \tau < n\frac{l}{l+1} - \frac{l}{2}(k-1),$$

ce qui est bien la première inégalité recherchée ; la deuxième inégalité vient du fait que le degré de $Q_l(x)$ ne doit pas être négatif. ■

Une fois un tel polynôme $Q(x,y)$ construit, on peut décoder à partir de la proposition suivante.

**Théorème 1.94.** *Si $Q(x,y)$ vérifie les contraintes précédentes et si le mot de code envoyé est $c(f) = (f(x_1), f(x_2), \cdots, f(x_n))$ avec $\deg(f(x)) < k$, alors*

$$(y - f(x)) \,|\, Q(x,y).$$

**Preuve.** Pour un polynôme $Q(x,y)$ vérifiant les conditions précédentes, on considère le polynôme univarié $Q(x, f(x))$. D'après les conditions sur les degrés des $Q_j(x)$, chacun des polynômes $f(x)^j Q_j(x)$ est de degré $n-\tau-1$ et donc le polynôme $Q(x, f(x))$ est de degré au plus $n-\tau-1$.

Maintenant, par hypothèse, on a $Q(x_i, r_i) = 0$ pour tout $i$ dans $\{1, 2, \cdots, n\}$, et comme $f(x_i) = r_i$ pour au moins $n-\tau$ valeurs de $i$ (l'erreur étant de poids $\tau$), on a finalement $Q(x_i, f(x_i)) = 0$ pour au moins $n-\tau$ valeurs de $i$. Le polynôme $Q(x, f(x))$ est donc un polynôme de degré $\leq n-\tau-1$, ayant au moins $n-\tau$ racines. C'est donc le polynôme nul.

Montrons qu'alors $(y - f(x)) \,|\, Q(x,y)$. Le polynôme $Q(x,y)$ s'écrit

$$Q(x,y) = Q_0(x) + Q_1(x)y + Q_2(x)y^2 + \cdots + Q_l(x)y^l.$$

Comme $Q(x, f(x)) = 0$, il vient

$$0 = Q_0(x) + Q_1(x)f(x) + Q_2(x)f(x)^2 + \cdots + Q_l(x)y^l,$$

soit

$$Q_0(x) = -(Q_1(x)f(x) + Q_2(x)f(x)^2 + \cdots + Q_l(x)y^l).$$

En remplaçant cette dernière expression de $Q_0(x)$ dans l'expression de $Q(x,y)$, on obtient

$$\begin{aligned} Q(x,y) &= -(Q_1(x)f(x) + \cdots + Q_l(x)f(x)^l) + Q_1(x)y + \cdots + Q_l(x)y^l \\ &= Q_1(x)(y - f(x)) + Q_2(x)(y^2 - f(x)^2) + \ldots + Q_l(x)(y^l - f(x)^l) \\ &= (y - f(x))\sum_{j=1}^{l}(Q_j(x)\sum_{p=0}^{j-1} y^p f(x)^{j-1-p}) \end{aligned}$$

et $Q(x,y)$ est bien divisible par $(y - f(x))$. ■

On en déduit l'algorithme de décodage en liste suivant.

**Méthode** **Décodage en liste de Sudan des codes de Reed-Solomon**

1. Pour un mot reçu $r$ avec une erreur de poids $\tau$, construire le polynôme interpolé $Q(x,y) = Q_0(x) + Q_1(x)y + Q_2(x)y^2 + \ldots + Q_l(x)y^l$.
2. Trouver les facteurs de la forme $(y - f(x))$ avec $\deg(f(x)) < k$ qui divisent $Q(x,y)$. Si $\tau < n\frac{l}{l+1} - \frac{l}{2}(k-1)$, alors le mot émis fait partie de cette liste.

Il existe plusieurs méthodes possibles pour chacune de ces étapes. L'étape la plus coûteuse est l'étape 1 : il est toujours possible de trouver le polynôme $Q(x,y)$ en résolvant un système linéaire, mais les méthodes de type interpolation multivariée donnent les meilleurs résultats en termes de complexité calculatoire.

Une analyse détaillée des paramètres de l'algorithme permet de montrer que cet algorithme améliore le décodage en $\lfloor (d-1)/2 \rfloor$ pour un taux $k/n < 1/3$. On peut aussi montrer qu'asymptotiquement pour $k/n$ petit, et $k$ et $n$ grands l'algorithme permet de décoder jusqu'à $n - \sqrt{2nk}$ erreurs, soit un taux d'erreurs décodables en $1 - \sqrt{\frac{2k}{n}}$, qui tend vers 1 lorsque $k/n$ tend vers zéro. Ce résultat est à comparer au décodade classique qui ne décode que jusqu'à un taux de $1/2$ lorsque $k/n$ tend vers zéro.

**Remarque.** L'algorithme de décodage de Guruswami-Sudan améliore ce dernier algorithme en introduisant plus de contraintes sur le polynôme $Q(x,y)$. À la différence de l'algorithme de Sudan, il améliore le décodage classique des codes de Reed-Solomon pour des taux quelconques et, pour $k/n$ petit, il permet de décoder jusqu'à $n - \sqrt{nk}$ erreurs.

**Exemple 1.95.** On considère le code de Reed-Solomon $[10, 2, 9]_{11}$ défini pour $x_i = 2^{i-1}$. Par décodage classique, on peut décoder jusqu'á $t = (9-1)/2 = 4$ erreurs. Prenons $l = 2$ dans la définition du polynôme interpolé $Q(x,y)$, alors on peut décoder jusqu'à une erreur de poids $\tau < n\frac{l}{l+1} - \frac{l}{2}(k-1) = 10.2/3 - 1$, soit jusqu'à $\tau = 5$. Considérons le mot à décoder $r = (7,6,5,3,8,7,9,0,2,5)$. On cherche un polynôme $Q(x,y) = Q_0(x) + Q_1(x)y + Q_2(x)y^2$ avec $Q_0(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$, $Q_1(x) = b_0 + b_1x + b_2x^2 + b_3x^3$ et $Q_2(x) = d_0 + d_1x + d_2x^2$, tel que $Q(x_i, r_i) = 0$ pour $1 \leqslant i \leqslant 10$. En écrivant les contraintes sur $Q(x,y)$, on obtient un système linéaire de 10 équations et 12 inconnues, les $a_0, \ldots, a_4, b_0, \ldots, b_3, d_0, d_1, d_2$. Une résolution du système amène un espace solution de dimension 2, engendré par les deux vecteurs $(1,0,0,2,2,4,6,0,7,1,10,6)$ et $(0,1,10,3,1,2,7,10,2,5,2,9)$. Prenons par exemple le premier vecteur. On peut retrouver un exemple de polynôme $Q(x,y)$ en écrivant $Q_0(x), Q_1(x)$ et $Q_2(x)$ à partir des coordonnées de ce vecteur, comme défini précédemment. Ce qui amène $Q(x,y) = 1 + 2x^3 + 2x^4 + y(4 + 6x + 7x^3) + y^2(1 + 10x + 6x^2)$, qui se factorise en $Q(x,y) = 7(y - (4x+3))(x^3 + 2x^2y + 7x^2 + 7xy + 9x + 4y + 10)$. Il existe un facteur de la forme $(y - f(x))$ qui divise $Q(x,y)$. On en déduit que le mot décodé correspond à $f(x) = 4x + 3$, soit $c(f) = (7,10,5,6,8,1,9,3,2,0)$. On vérifie de plus que l'erreur $(0,7,0,8,0,6,0,8,0,5)$ a poids 5.

**Test 1.38.**

En reprenant le même code que dans l'exemple, décoder le mot $(7,1,0,9,5,9,5,8,3,4)$ pour $\tau = 5$. Qu'observe-t-on ?

## VIII. Exercices

### 1.1. ⋆

1) Montrer que si $\mathcal{C}$ est un code auto-orthogonal dont la matrice génératrice est composée de mots de poids multiples de 4, alors tous les mots du code ont un poids multiple de 4.
2) Montrer que si $\mathcal{C}$ est auto-orthogonal sur $\mathbb{F}_3$ alors, pour tout $x \in \mathcal{C}$, $w_H(x) \equiv 0 \pmod 3$.

### 1.2. ⋆

Montrer à partir de la table des syndromes du code de Hamming $[7,4,3]$ que le code décode de manière univoque toutes les erreurs de poids 1.

### 1.3. Construction des codes de Hamming sur $\mathbb{F}_q$ ⋆

On veut généraliser la construction des codes de Hamming sur $\mathbb{F}_q$. Pour $m$ fixé, montrer qu'en prenant pour colonnes du code des droites vectorielles (on prend une colonne à une constante multiplicative non nulle près) plutôt que tous les éléments non nuls de $\mathbb{F}_q^m$, on obtient plus généralement par cette construction des codes de paramètres $[\frac{q^r-1}{q-1}, \frac{q^r-1}{q-1} - r, 3]_q$.

### 1.4. Construction du code de Golay ternaire ⋆⋆

On repart de la construction du code de Golay binaire. Considérer pour $A$ une matrice $5 \times 5$ dont les colonnes sont indicées de 0 à 4. La première ligne de $A$ est construite en prenant 0 pour la première coordonnée puis 1 pour un indice de colonne qui est un carré modulo 5 et $-1$ sinon. Montrer que la construction précédente, avec cette nouvelle matrice $A$, permet de construire un code $[12,6,6]_3$, appelé *code de Golay étendu ternaire* et le *code de Golay ternaire* $[11,6,5]_3$.

### 1.5. ⋆

Montrer que les deux codes de Golay $[23,12,7]$ et $[11,6,5]_3$ sont parfaits.

### 1.6. ⋆

Montrer qu'il ne peut pas exister de code binaire $[12,7,5]$.

### 1.7. ⋆⋆

Montrer que la famille infinie des codes de Reed-Muller $\mathcal{R}(m, \frac{m-1}{2})$ pour $m$ impair quelconque est une famille de codes de taux fixe $1/2$. Cette famille de codes est-elle asymptotiquement bonne ?

### 1.8. ⋆

Montrer que pour un code cyclique binaire, ajouter $\{0\}$ à l'ensemble de définition du code revient à prendre le sous-code des mots de poids pair du code.

### 1.9. ⋆

Combien y-a-t-il de codes cycliques binaires possibles en longueur 23 ? Idem sur $\mathbb{F}_3$ en longueur 11 ? Quels sont-ils ? (On ne cherchera pas à trouver la distance minimale).

### 1.10. ⋆

Trouver les codes BCH et leurs paramètres pour $q = 3$ et $n = 13$.

### 1.11. BCH non binaires ⋆⋆

En utilisant le même type de raisonnement que pour les codes BCH binaires, montrer qu'il existe des codes BCH ternaires de paramètres $[3^m - 1, k \geq 3^m - 1 - 2mt, d \geq 3t + 1]_3$. Généraliser cette construction au cas $q$-aire.

### 1.12. ⋆

On considère un code de Reed-Solomon $[7,3,5]_8$. On encode $x_i = \alpha^{i-1}$ pour $\alpha$ un élément primitif. Prendre un mot du code, ajouter deux erreurs et le décoder. Peut-on savoir directement à partir des syndromes s'il y a une ou deux erreurs ?

### 1.13. Codes de Reed-Solomon cycliques ⋆⋆⋆

Soit $\alpha$ un élément primitif de $\mathbb{F}_q$. Soit $RS_k$ le code de Reed-Solomon, de longueur $n = q - 1$ et de dimension $k$, défini sur un ensemble $x_i = \alpha^{i-1}$ pour $i = 1, \ldots, q-1$ par $c(f) = (f(x_1), f(x_2), \cdots, f(x_{q-1}))$ pour $f \in \mathcal{P}_k$ avec $\mathcal{P}_k = \{f \in \mathbb{F}_q[x] | \deg(f) < k\}$. Montrer que $RS_k$ est un code cyclique de longueur $n$, de dimension $k$ sur $\mathbb{F}_q$ et de polynôme générateur $g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{n-k})$.

### 1.14. Codes BCH binaires et codes de Reed-Solomon ⋆⋆

Soit $\mathcal{C}$ un code BCH binaire primitif au sens strict, de longueur $n = 2^m - 1$ et de distance construite $\delta$. Montrer que $\mathcal{C}$ est exactement le sous-code binaire sur le sous-corps du code de Reed-Solomon cyclique $RS_{n-\delta-1}$, comme défini à l'exercice précédent.

### 1.15. ⋆⋆

Pour $l = 1$ ou $l = 2$, quand est-ce que le décodage en liste de Sudan améliore le décodage classique ?

# Première partie
# Solutions des tests

# Deuxième partie

## Solutions des tests

**0.1.** S'il y a une erreur dans un des quatre bits d'information, il y aura deux erreurs sur une ligne et une colonne différente. Si l'erreur n'est pas sur un des quatre bits d'information, il y aura, soit une ligne, soit une colonne avec une erreur, et donc l'erreur sera dans la ligne ou colonne extérieure.

**0.2.** On suppose qu'il y a exactement 1 erreur. Il y a plusieurs cas possibles : si l'erreur est sur $p_1, p_2$ ou $p_3$, deux des cercles auront une somme paire et le troisième correspondant à l'un des $p_i$ aura une somme impaire. Si l'erreur est en $d_1, d_2$ ou $d_3$, deux des cercles auront une somme incorrecte et l'erreur sera à l'intersection de ces deux cercles (mais pas $d_4$). Enfin, si les trois cercles ont une somme incorrecte, l'erreur sera $d_4$.

**0.3.** Si l'on part d'une matrice identité avec des 1 sur la diagonale, cette matrice engendre tout l'espace. On considère les vecteurs $e_i$ de longueur $n$ valant 0 partout sauf sur la coordonnée $i$ où ils valent 1. Les $e_i$ sont indépendants et engendrent $\mathbb{F}_2^n$. Considérons maintenant $e'_i = e_i + e_n$ pour $1 \leqslant i \leqslant n-1$. Les $e'_i$ sont indépendants, et comme la somme de deux mots avec un nombre pair de 1 a aussi un nombre pair de 1, on en déduit que le code engendré par les $e'_i$ est contenu dans le code de parité. Ce code a pour dimension $n-1$. Comme le code de parité ne peut avoir dimension $n$ car sinon il contiendrait $e_1$, ce qui n'est pas possible, alors ce code est exactement le code de parité de dimension $n-1$.

**0.4.** Les mots du code sont les mots $(d_1, d_2, d_3, d_4, p_1, p_2, p_3)$ satisfaisant les conditions de parité pour les trois cercles. Si deux mots appartiennent au code alors, d'après les propriétés de parité, ces conditions seront aussi vérifiées pour la somme des coordonnées de ces mots. Le code est donc linéaire.
Pour trouver une matrice génératrice du code, par linéarité il suffit de regarder les valeurs des redondances $p_1, p_2$ et $p_3$ pour de l'information fixée. On peut par exemple regarder la valeur des $p_i$ pour des valeurs $(d_1, d_2, d_3, d_4)$ valant respectivement $(1000), (0100), (0010)$ et $(0001)$ qui forment des vecteurs indépendants. On trouve donc comme matrice génératrice

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

**0.5.**

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

**0.6.** C'est la définition du dual.

**0.7.** Sur $\mathbb{F}_2$, lorsque l'on fait la somme des mots, la somme de deux 1 s'annule : ceci donne le résultat. Si $x.y = 0$ alors le nombre de 1 en commun sur une même coordonnée est pair et donc, le poids de $x * y$ est pair, ce qui implique que $2w(x * y)$ est multiple de 4.

**0.8.** Cela vient directement de la linéarité : $d(x, y) = w(x - y)$ et $x - y \in \mathcal{C}$ par linéarité.

**0.9.** Si l'on calcule tous les mots possibles, on trouve $d = 3$.

**0.10.** Ces codes sont distincts puisque l'on peut trouver un mot dans l'un qui n'est pas dans l'autre. Si l'on permute les colonnes 2 et 4 pour un code, on obtient l'autre.

**0.11.** On n'utilise jamais la notion de linéarité dans la preuve.

**0.12.** Le syndrome vaut $(100)^t$ avec un représentant principal unique : $(00100)$. Le mot se décode en $(11110)$. Pour le second mot, le syndrome est $(101)^t$. Dans ce cas (voir la table des syndromes), il y a trois représentants principaux potentiels (il y a trois mots de poids 2 dans le translaté associé) ; un seul a été choisi pour la table, mais le mot d'origine peut aussi correspondre à l'un des deux autres.

**0.13.** La distance minimale vient de la remarque précédente. L'énumérateur des poids est $W_{\mathcal{H}_8}(x, y) = x^8 + 14x^4y^4 + y^8$.

**0.14.** On a vu que $\mathcal{C}_1 \oplus \mathcal{C}_2 = \{\{(c_1|c_2)\, c_1 \in \mathcal{C}_1\}\, c_2 \in \mathcal{C}_2\}$. D'après la définition du polynôme énumérateur, si l'on a par exemple $c_1 = (00101)$ et $c_2 = (1110)$, on leur associe respectivement les monômes $x^3y^2$ et $x^1y^3$ et pour la concaténation de $c_1$ et $c_2$, on obtient $(001011110)$ de monôme $x^4y^5$, exactement le produit des deux monômes associés à $c_1$ et $c_2$. Soit $W_{\mathcal{C}_1}(x, y)$ l'énumérateur des poids de $\mathcal{C}_1$. Par définition, $\mathcal{C}_1 \oplus \mathcal{C}_2 = \{\{(c_1|c_2)\, c_1 \in \mathcal{C}_1\}\, c_2 \in \mathcal{C}_2\}$. D'après ce que l'on vient de voir, la contribution de $\{(c_1|c_2)\, c_1 \in \mathcal{C}_1\}$ sera le monôme associé à $c_2$ fois $W_{\mathcal{C}_1}(x, y)$ et le résultat en découle en faisant varier $c_2$.

**0.15.** C'est une simple généralisation du cas binaire ; on prend tous les vecteurs de $\mathbb{F}_q^m$ à une constante près. Par exemple, sur $\mathbb{F}_3$ avec $m = 2$, on a comme colonnes $(10)^t, (01)^t, (11)^t$ et $(12)^t$ ; on considère que la colonne $(20)$ est associée à $(10)^t$. On obtient un code $[4, 2, 3]_3$.

**0.16.** Il suffit d'écrire les $a_{ij}$ en fonction des $a_i$.

**0.17.** Il suffit de calculer tous les mots possibles, modulo un facteur non nul. Par exemple, $(100111)$ est dans le code, donc $\omega(100111)$ et $\omega^2(100111)$, aussi, avec le même poids. Soit, en tout, simplement vingt et une possibilités $((64-1)/3)$ en ne comptant pas le mot nul tout à 0.

**0.18.** Le code $\mathcal{R}(1,2)$ est un code $[4,3,2]$, le code $\mathcal{R}(0,2)$ est le code engendré par le mot $(1111)$. On applique ensuite la construction de Plotkin.

**0.19.** C'est une application directe du théorème.

**0.20.** Faire le calcul.

**0.21.** Non, car il existe des mots de l'espace qui sont à distance $r$ de deux mots du codes (par exemple le mot nul et un mot de poids $2r$). Il n'y a donc pas de décodage univoque pour ces éléments et le code n'est pas parfait.

**0.22.** Les boules de rayon $r_e(\mathcal{C})$ ne s'intersectent pas, alors que les boules de rayon $r_c(\mathcal{C})$ recouvrent tout l'espace par définition, donc $r_e(\mathcal{C}) \leq r_c(\mathcal{C})$. Ces deux rayons sont égaux uniquement pour des codes parfaits.

**0.23.** En reprenant la même idée que dans le lemme 1.52, on peut montrer que, si le rayon de recouvrement d'un code linéaire $\mathcal{C}$ avec $B_2(n,d)$ mots, est supérieur ou égal à $d$, alors on peut construire un code *linéaire*, $\mathcal{C}' = \mathcal{C} \cup x + \mathcal{C}$, de distance minimale $d$, pour un $x$ choisi comme dans le lemme, ce qui montre que le rayon de recouvrement de $\mathcal{C}$ est au plus $d-1$. Le reste de la démonstration est identique à la preuve de la borne de Gilbert-Varshamov.

**0.24.** $C_0 = \{0\}, C_1 = \{1,2,4,8\}, C_3 = \{3,6,12,9\}, C_5 = \{5,10\}, C_7 = \{7,14,13,11\}$.

**0.25.** $ord_2(9) = 6, 2^6-1 = 9\times 7$, $C_0 = \{0\}, C_1 = \{1,2,4,8,7,5\}, C_3 = \{3,6\}$, $x^9-1$ s'écrit donc sous la forme $x^9-1 = (x-1).p_1(x).p_2(x)$ où $\deg(p_1(x)) = 6$ et $\deg(p_2(x)) = 2$. Comme il n'y a qu'un seul polynôme irréductible de degré 2 sur $\mathbb{F}_2$ : $x^2+x+1$, on a $p_2(x) = x^2+x+1$ et $p_1(x) = (x^9-1)/((x-1).(x^2+x+1)) = x^6+x^3+1$.

**0.26.** On peut partir du polynôme irréductible de degré 3 : $x^3+x+1$. On peut prendre comme élément primitif $\alpha = x$. Dans ce cas par exemple $\alpha^6 = x^6 \pmod{x^3+x+1} = x^2+1 = (1+x)^2 = (\alpha^3)^2$.

**0.27.** On obtient un code $[7,4,3]$ (le code de Hamming).

**0.28.** On connaît la factorisation de $x^7-1 = (x+1)(x^3+x^2+1)(x^3+x+1) = g_1(x).g_2(x).g_3(x)$. On calcule les différents cas possibles ($2^3 = 8$). Par exemple, pour le polynôme générateur $g_2(x).g_1(x)$, la dimension est $7-4 = 3$ et la distance minimale 4.

**0.29.** Prendre $\alpha^i$ au lieu de $\alpha$ revient à permuter les colonnes du code par l'application qui envoie la colonne $j$ sur la colonne $ij \pmod n$. Cette opération est bien une permutation puisque $pgcd(i,n) = 1$.

**0.30.** $h(x) = x^4+x^3+x^2+1$. Si l'on définit $\mathbb{F}_2^3$ à partir du polynôme $x^3+x+1$, on a $T = \{1,2,4\}$ pour le code et $T = \{0,1,2,4\}$ pour le dual.

**0.31.** $T = C_1 = \{1,2,4\}$. Il y a 2 éléments consécutifs 1 et 2 donc la distance est au moins 3. En calculant le polynôme générateur, on trouve un mot de poids 3, donc la distance est exactement 3. Si l'on ajoute 0 à $T$, on trouve $d = 4$. Pour la longueur 23, le code est un code $[23,12,d]$ ; on trouve $d \geq 5$ car il y a quatre éléments qui se suivent dans $C_1$ : $\{1,2,3,4\}$. En fait, ce code est le code de Hamming binaire, $d = 7$ et la borne n'est pas optimale.

**0.32.** Prenons par exemple $\alpha = 3$ comme racine primitive de $\mathbb{F}_7$. On obtient $g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3) = x^3+3x^2+x-1$. Le code est un code $[6,3,4]_7$.

**0.33.** $g(x) = x^{10}+x^8+x^5+x^4+x^2+x+1$, l'erreur est $e(x) = x^8+x^9$.

**0.34.** Les zéros de $\sigma(x)$ sont les inverses des emplacements des erreurs. Comme $\sigma'(x) = x^t\sigma(x^{-1})$, les zéros de $\sigma'(x)$ sont les inverses des zéros de $\sigma(x)$, soit les emplacements d'erreurs. Comme $g_j(x)$ est multiple de $\sigma'(x)$, à une constante non nulle près, les racines de $g_j(x)$ sont exactement les emplacements d'erreurs.

**0.35.** $\sigma(x) = \alpha^2x^2+\alpha^{12}x+1$.

**0.36.** Une matrice génératrice reprend les trois premières lignes de la matrice de l'exemple précédent. Le mot envoyé correspond à $f(x) = 1+2x+3x^2$ et l'erreur vaut $(0,0,2,0,0,0)$.

**0.37.** On obtient à nouveau un code $[8,2,5]$.

**0.38.** Il y a deux mots du code possibles et l'algorithme renvoie une liste. Les mots à distance 5 sont $(7,1,0,9,5,8,3,4,6,10)$ et $(6,10,7,1,0,9,5,8,3,4)$.

# Troisième partie
# Solutions des exercices

# Quatrième partie
# Solutions des exercices

**0.39.** 1) Pour un code auto-orthogonal, le produit $x * y$ est pair et donc, pour le calcul du poids de la somme, $2w(x * y) \equiv 0 \pmod 4$ ce qui donne le résultat.
2) Sur $\mathbb{F}_3$, on a $1^2 = (-1)^2 = 1$. De plus, si le code est auto-orthogonal, alors $x.x = 0$, ce qui implique que le nombre de coordonnées non nulles est un multiple de 3.

**0.40.** On peut faire la table des syndromes, sinon cela vient directement du théorème 1.26.

**0.41.** Puisque l'on choisit les colonnes à une constante non nulle près, le nombre de colonnes (la longueur du code) est $(q^m - 1)/(q - 1)$. Le nombre de lignes dans le dual est toujours $m$. Maintenant, deux colonnes du dual ainsi construit sont toujours indépendantes par construction et donc la distance minimale est plus grande que 3 (et exactement 3 en exhibant une relation entre 3 colonnes).

**0.42.** On montre d'abord, comme pour le cas binaire, que le code associé à la matrice est un code autodual sur $\mathbb{F}_3$ et que tout mot de poids 6 s'écrit $(a|b)$ mais que, par symétrie, on a aussi $(b|a)$. Comme $x.x = 0$ pour tout $x$ du code, le poids de chaque mot est un multiple de 3. Pour montrer que le code a pour distance minimale 6, il suffit donc de vérifier qu'il n'a pas de mots de poids 3.

**0.43.** Faire le calcul.

**0.44.** Les paramètres de tout code doivent vérifier la borne de Hamming. Ce n'est pas le cas pour un code avec ces paramètres, donc il ne peut exister.

**0.45.** Dans le cas $m$ impair, la symétrie des coefficients binomiaux assure que la dimension est exactement $2^m/2$. Pour ces codes, la distance vaut $2^{m-\frac{m-1}{2}} = 2^{\frac{m+1}{2}}$ qui est de l'ordre de $\sqrt{2^m}$ avec $n = 2^m$ pour ces codes. Cette famille n'est donc pas asymptotiquement bonne. En fait, elle est même plutôt mauvaise si l'on considère uniquement le critère de la distance, puisque la distance croît en racine carrée et non linéairement par rapport à la longueur.

**0.46.** Dire que $\{0\}$ fait partie de l'ensemble de définition signifie que, pour tout mot du code $c(x)$, $\alpha^0 = 1$ est solution donc $c(1) = 0$ pour tout $c \in \mathcal{C}$. Cela signifie que $c$ a un nombre pair de coordonnées non nulles.

**0.47.** À chaque fois, il y en a $2^3$. Il s'agit des codes de Golay quand la dimension correspond.

**0.48.** On trouve des codes $[13, 10, 3]_3$, $[13, 7, 4]_3$, $[13, 4, 7]_3$ et $[13, 1, 13]_3$.

**0.49.** On recherche un tel code sous la forme d'un code BCH ternaire primitif au sens strict de longueur $n = 3^m - 1$. La taille des 3-classes cyclotomiques est un diviseur de $m$. Pour avoir une distance construite supérieure à $3t + 1$, il suffit d'avoir une séquence $\{1, 2, \ldots, 3t\}$ de $3t$ valeurs consécutives dans l'ensemble de définition du code.
Considérons le code d'ensemble de définition $T = C_1 \cup C_2 \cup \ldots \cup C_{3t}$. Par construction, ce code a bien une distance minimale $d \geq 3t+1$. Dans le cas ternaire, $C_i = C_{3i}$ donc $T$ s'écrit aussi simplement comme la réunion des classes $C_i$ pour $1 \leq i \leq 3t$ et $i$ non divisible par 3. L'ensemble $T$ peut donc s'écrire comme réunion de $2t$ classes cyclotomiques, chacune ayant au plus $m$ éléments. Le code obtenu a donc une dimension $k$ supérieure ou égale à $n - 2tm$ et une distance minimale construite de $3t + 1$.
Dans le cas $q$-aire, on obtient des codes $[q^m - 1, k \geq q^m - 1 - (q - 1)mt, d \geq qt + 1]_q$. Cette construction s'avère être particulièrement intéressante pour des alphabets de petite taille comme $q = 2$, $q = 3$ ou $q = 4$.

**0.50.** Pour le décodage, on peut suivre la méthode de Welch-Berlekamp. On peut connaître le nombre d'erreurs en calculant le rang de la matrice des syndromes.

**0.51.** Le code est cyclique car si $c(f) = (f(x_1), f(x_2, \cdots, f(x_n)) = (f(\alpha^0), f(\alpha), \cdots, f(\alpha^{n-1}))$ est un mot du code pour $\deg(f) < k$, alors le mot décalé de une position $c' = (f(\alpha^{n-1}), f(\alpha^0), f(\alpha), \cdots, f(\alpha^{n-2}))$ est aussi un mot du code. En effet, on peut remarquer que $c' = (f_1(\alpha^0), f_1(\alpha), \cdots, f_1(\alpha^{n-1}))$ pour $f_1(x) = f(\alpha^{-1}x)$ car $\alpha^n = 1$. Comme $f$ est de degré $k$, $f_1$ est aussi de degré $k$ et donc $c'$ appartient au code, qui est donc cyclique. Maintenant, on peut remarquer que, pour tout $1 \leqslant j \leqslant n - 1$, la somme $S = \sum_{i=0}^{n-1} (\alpha^j)^i$ vérifie $S = 0$ car $\alpha^j S - S = (\alpha^j)^n - 1 = 0$ et $1 - \alpha^j \neq 0$. Le code $RS_k$ est engendré par les vecteurs de la forme $e_l = (\alpha^l, (\alpha^2)^l, \cdots, (\alpha^{n-1})^l)$ pour $0 \leqslant l \leqslant k - 1$. En utilisant le fait que la somme $S$ est nulle pour $1 \leqslant j \leqslant n - 1$, on peut alors en déduire que, pour la matrice $H$ définie par

$$H = \begin{bmatrix} 1 & \alpha & \ldots & \alpha^{n-1} \\ 1 & \alpha^2 & \ldots & \alpha^{2(n-1)} \\ & \ldots & \ldots & \\ 1 & \alpha^{n-k} & \ldots & \alpha^{(n-k)(n-1)} \end{bmatrix},$$

le produit $H.u_l^t = 0$ pour tout $0 \leqslant l \leqslant k - 1$. Cela montre que $H$ est contenue dans le dual de $RS_k$. De plus, elle est de rang $n - k$ car on peut en extraire une sous-matrice $(n - k) \times (n - k)$ inversible sous forme de

matrice de Vandermonde. C'est donc exactement une matrice du dual de $RS_k$. En reprenant la démonstration de la borne BCH, on en déduit que si $c(x) = (c_1, \cdots, c_n)$ est un mot du code, alors $H.c^t = 0$ implique $c(\alpha^i) = 0$ pour $1 \leqslant i \leqslant n-k$. Comme $RS_k$ a pour dimension k, cela montre que $RS_k$ est un code cyclique de zéros $\alpha, \alpha^2, \cdots, \alpha^{n-k}$ et donc de polynôme générateur $g(x) = (x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{n-k})$.

**0.52.** Le sous-code sur le sous corps binaire de $RS_{n-\delta-1}$ est linéaire et cyclique puisque le code $RS_{n-\delta-1}$ est cyclique. D'après l'exercice précédent, ce code a dans l'ensemble de ses zéros $\alpha, \alpha^2, \ldots, \alpha^{\delta-1}$. Soit un ensemble de définition T qui contient au moins $\{1, 2, \ldots, \delta-1\}$. Comme le code est binaire cyclique, l'ensemble de définition contient aussi tous les éléments des 2-classes cyclotomiques modulo n associées aux éléments de l'ensemble de définition, donc $C_1 \cup \ldots \cup \mathbb{C}_{\delta-1} \subset T$. Réciproquement, tout mot de code binaire ayant pour zéros $\alpha, \alpha^2, \ldots, \alpha^{\delta-1}$ (soit le code BCH binaire avec les mêmes zéros) est contenu dans le sous-code sur le sous-corps. On peut donc en déduire que le sous-code sur le sous-corps de $RS_{n-\delta-1}$ est le code BCH binaire au sens strict de distance construite $\delta$.

**0.53.** Si $l = 1$, le décodage en liste peut décoder jusqu'à $\tau < \frac{n}{2} - \frac{1}{2}(k-1)$ erreurs, ce qui correspond au cas du décodage classique, il n'y a pas d'amélioration. Si $l = 2$, alors, pour avoir $\tau > (n-k+1)/2$, il faut $\frac{2}{3}n - k + 1 > (n-k+1)/2$, soit après simplification, $k/n \leq 1/3 + 1/n$.

# Index

Code
    code linéaire, 3
    construction de Plotkin, 12
    cyclique, 22
    de Golay, 14
    de Reed-Muller, 15
    distance minimale, 7
    dual d'un, 5
    matrice génératrice, 4
    mots de, 3
    Syndrome, 9
Code correcteur, 3

Hamming
    distance de, 6
    borne de, 17
    premier code de, 3

Reed-Solomon
    code de, 32

Singleton
    borne de, 18