

Table des matières

Introduction	4
1 Les corps finis	5
1.1 <i>Introduction</i>	5
1.2 Corps finis	5
1.3 Caractéristique et cardinal	5
1.4 Polynômes sur les corps finis	7
1.4.1 L'anneaux $A[X]$ des polynômes	7
1.4.2 Polynômes irréductibles	9
1.4.3 Polynômes cyclotomiques	10
1.5 Factorisation de $x^n - 1$	11
1.6 Construction d'un corps fini	12
2 Généralités sur les Codes Linéaires et Cycliques	15
2.1 <i>Introduction</i>	15
2.2 Les Codes	15
2.3 Codes Linéaires	17
2.4 Codes Cycliques	22
3 Codes Optimaux $[n, \frac{n}{2}]$ sur F_2	26
3.1 <i>Introduction</i>	26
3.2 Calcul de la distance minimum des Codes Cycliques de Rendement $\frac{1}{2}$ sur F_2 pour $n \leq 50$	26

3.2.1	Codes Cycliques Optimaux sur $\text{GF}(2)$	26
3.3	ANNEXE	38
3.3.1	Programme de recherche de la distance minimaum d'un code cyclique sur $\text{GF}(2)$	38
3.4	Conclusion	43
3.5	BIBLIOGRAPHIE	44

NOTATIONS

- F_q : Un corps fini de cardinal q .
- $car(p)$: Caractéristique d'un corps fini
- F_q^* : Un groupe d'ordre $(q - 1)$.
- $A[X]$: L'ensemble des polynômes sur A .
- $\varphi(n)$: Fonction d'Euler .
- $C[n, k]$: Code correcteur de longueur n et dimension k .
- $d(x, y)$: Distance de Hamming.
- $wt(x)$: Poids de Hamming.
- $W(X, Y)$: Le polynôme énumérateur des poids d'un code.
- $\langle x, y \rangle$: Le produit scalaire de x et y .
- C^\perp : Le dual de code C .
- I_k : Matrice identique de taille $k \times k$.
- d_{\min} : La distance minimale .
- d_C : Maximal distance minimum d'un code cyclique.
- $(f(x))$: idéal engendré par f .
- \cong : isomorphe.
- A^* : $A \setminus \{0\}$.
- Q_n : le n^{ieme} polynôme cyclotomique.
- I : idéal de A .
- $F_q[x]$: anneau des polynômes à coefficients dans F_q .
- F_q^n : espace vectoriel des vecteurs de longueur n sur F_q .
- x^\perp : transposé du vecteur x .
- c : mot de code $\in C$.
- $F_q[x] \setminus (x^n - 1)$: L'anneaux quotient .
- $c(x)$: représentation polynômiale.
- $[x]$: la partie entière d'un réel x .

INTRODUCTION

-Les codes correcteurs d'erreurs sont présents aujourd'hui dans tous les réseaux, à des niveaux techniques plus ou moins complexes. La généralisation de l'usage des satellites de télécommunication dans les réseaux mondiaux augmentant le niveau de bruit, le niveau technique de la correction d'erreurs dans ces réseaux a tendance à augmenter sensiblement.

La théorie du codage vise à construire des codes correcteurs performant opérant au plus proche des limites théoriques établies par la théorie de l'information, dans ce contexte il s'agit de rechercher des codes optimaux ayant la meilleure distance minimale pour une longueur n pair et une dimension $\frac{n}{2}$.

Dans ce travail on s'intéresse aux codes optimaux $[n, \frac{n}{2}]$ sur le corps fini F_2 . En considérant les codes cycliques de paramètres $[n, \frac{n}{2}]$, pour n pair, nous avons recherché ces codes au sens de la distance minimum.

Déroulement du mémoire

Dans Le premier chapitre nous présentons les notions et propriétés fondamentales nécessaires pour la réalisation de ce travail concernant : les corps fini, polynômes sur les corps finis, factorisation de $x^n - 1$. Les notions citées dans ce chapitre représentant l'outil mathématique utilisé pour l'étude des codes correcteurs d'erreurs.

Le deuxième chapitre regroupe les définitions et les propriétés fondamentales des codes linéaires et des codes cycliques.

En fin, dans le dernier chapitre, on va calculer la distance minimale des codes cycliques de rendement $\frac{1}{2}$ sur F_2 pour $n \leq 50$, on utilise l'algorithme de Chen pour déterminer cette distance, on va choisir les codes optimaux parmi tous les codes $[n, \frac{n}{2}]$.

Chapitre 1

Les corps finis

1.1 *Introduction*

Dans cette partie on ne présente que des définitions et résultats sur les corps finis qui nous seront utiles pour la suite concernant les codes correcteurs. C'est bien sûr loin d'être exhaustif, il y manque même certains résultats essentiels pour l'agrégation.

1.2 Corps finis

Définition 1.1 *Un corps fini est un corps qui possède un nombre fini d'éléments, un corps fini de q éléments est notée par F_q ou $GF(q)$, [field of q éléments, Galois field of q éléments].*

Exemple 1.2 *si p est premier $F_p = \mathbb{Z} / p\mathbb{Z}$ est un corps fini.*

1.3 Caractéristique et cardinal

Définition 1.3 *Le nombre p est appelé la caractéristique du corps F_q , Il est noté par $\text{car}(F_q)$.*

Théorème 1.4 *Soit F_q un corps fini de cardinal q*

- 1) La caractéristique de F_q est un nombre premier p

2) F_q est un espace vectoriel de dimension n sur F_p et on a $q = p^n$.

Preuve. Comme \mathbb{Z} est infini, F_q ne peut être de caractéristique nulle. Donc il contient F_p avec p premier. Ainsi F_q est un espace vectoriel sur F_p , sa dimension n est finie, sinon F_q serait infini. En tant qu'espace vectoriel F_q est isomorphe à F_q^n , donc F_q à p^n éléments. ■

Théorème 1.5 (Théorème de Wedderburn)

Tout corps fini est commutatif.

Définition 1.6 Un groupe G est dit cyclique s'il existe $g \in G$ telque $G = \langle g \rangle$. L'élément g est un générateur du groupe G .

$$G = \{1, g, \dots, g^{n-1}\}$$

Théorème 1.7 Soit F_q un corps fini de cardinal q . Le groupe multiplicatif (F_q^*, \times) est cyclique d'ordre $q - 1$.

Théorème 1.8 Soit F_q un corps fini de cardinal q .

Pour tout $x \in F_q^*$ on a : $x^{q-1} = 1$, et pour tout $x \in F_q$ on a : $x^q = x$.

Preuve. D'après le théorème (1.7) l'ordre de F_q^* est $q - 1$ donc pour tout $x \in F_q^*$ on a : $x^{q-1} = 1$ et par conséquent pour tout $x \in F_q$ on a : $x^q = x$. Il en résulte du théorème, qu'un corps fini F_q à q éléments est l'ensemble des racines du polynôme $x^q - x$. ■

Définition 1.9 On appelle élément primitif de F_q tout générateur du groupe multiplicatif F_q^* . Soit α un élément primitif d'un corps fini F_q alors :

$$F_q = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{q-2}\}$$

Avec : $\alpha^{q-1} = 1$ de plus α^k est primitif si et seulement si k et $q - 1$ sont premiers entre eux.

Proposition 1.10 (Frobenius) Soit F un corps de caractéristique $p > 0$. L'application

$$x \in F \mapsto x^p \in F$$

est un homomorphisme de corps appelé homomorphisme de Frobenius.

Sous-corps

Proposition 1.11 $F_{p^n} \subset F_{p^m}$ ssi n/m .

Exemple 1.12 F_8 n'est pas un sous-corps de F_{64} .

1.4 Polynômes sur les corps finis

1.4.1 L'anneaux $A[X]$ des polynômes

Un anneau $(A, +, \times)$ est un ensemble A muni de deux opérations (lois), $+$ et \times , telles que $(A, +)$ est un groupe commutatif d'élément neutre noté 0 , et telles que \times est associative, distributive par rapport à $+$, et munie d'un élément neutre noté 1 . On note A^* le groupe multiplicatif de A , constitué des éléments inversibles (ou unités) de A . Ne pas confondre A^* avec $A \setminus \{0\}$: l'anneau A est un corps lorsque $A^* = A \setminus \{0\}$. On ne considérera que des anneaux commutatifs (i.e. dont la multiplication est commutative), de même lorsque nous parlerons de corps nous sous-entendrons qu'il s'agit d'un corps commutatif.

Idéal d'un anneau

Définition 1.13 Un ensemble non vide I de A est un idéal si :

1. I est un sous groupe de $(A, +)$
2. $\forall a \in I, \forall b \in A, ab \in I$ et $ba \in I$.

Définition 1.14 L'ensemble des classes résiduelles d'un anneau A modulo un idéal I forme un anneau noté A/I dont les deux opérations sont définies par :

1. $(a + I) + (b + I) = (a + b) + I$
2. $(a + I)(b + I) = ab + I$.

Définition 1.15 Soit A un anneau, un polynôme f est une expression de la forme

$$f(x) = \sum_{i=0}^n a_i x^i = a_0 + a_1 x + \dots + a_n x^n.$$

Où n est un entier positif, les coefficients a_i , $0 \leq i \leq n$ sont des éléments de A et x un symbole appelé une indéterminée.

Soit $f(x) = \sum_{i=0}^n a_i x^i$ un polynôme tel que $a_n \neq 0$. Alors f est de degré n (on note $\deg(f) = n$), a_0 est le terme constant et a_n le coefficient de plus haut degré (leading coefficient en anglais).

On peut définir la somme et le produit de deux polynômes :

$$f(x) = \sum_{i=0}^n a_i x^i \text{ et } g(x) = \sum_{i=0}^m b_i x^i \text{ (} m \leq n \text{) :}$$

$$f(x) + g(x) = \sum_{i=0}^n (a_i + b_i) x^i$$

et

$$f(x)g(x) = \sum_{k=0}^{n+m} c_k x^k, \text{ où } c_k = \sum_{i+j=k} a_i b_j,$$

où $0 \leq i \leq n$ et $0 \leq j \leq m$.

L'ensemble des polynômes sur A muni de ces deux opérations admet une structure d'anneau noté $A[x]$. On montre facilement que pour tout $f, g \in A[x]$ (F étant un corps).

$$\deg(f + g) \leq \max \{ \deg(f), \deg(g) \}, \text{ et } \deg(fg) = \deg(f) + \deg(g).$$

Le polynôme g divise le polynôme f si l'existe un polynôme h tel que $f = gh$. Pour éviter tout problème, on ne considère ici que des polynômes à coefficients dans un corps.

Définition 1.16 Soit P un polynôme de $F[x]$. On définit l'idéal engendré par P , noté (P) par :

$$(P) = \{ PQ, Q \in F[x] \}.$$

C'est donc l'ensemble des polynômes multiples de P .

Définition 1.17 Un idéal engendré par un seul P , dit de type (P) , est appelé idéal principal.

Définition 1.18 Soit I un idéal de $F[x]$, on appelle générateurs de I les polynômes w tel que :

$$I = (w).$$

L'unique générateur d'un idéal I non nul est appelé polynôme minimal de l'idéal I .

1.4.2 Polynômes irréductibles

Définition 1.19 *Un polynôme $f \in F[x]$ qui n'admet pas des diviseurs propres est appelé polynôme irréductible.*

Exemple 1.20 $f(x) = (1 + x + x^3)$ est irréductible sur $F_2[x]$.

Théorème 1.21 *Tout polynôme $f \in F[x]$ peut s'écrire*

$$f = a f_1^{e_1} f_2^{e_2} \dots f_k^{e_k},$$

où $a \in F$, les f_i sont des polynômes irréductibles unitaires de $F[x]$ et les exposants e_i des entiers positifs. Cette factorisation est unique.

Définition 1.22 *Un élément α est une racine (ou un zéro) du polynôme f si $f(\alpha) = 0$.*

Remarque 1.23 *Si on a un tel polynôme et α une racine de f dans F_{q^n} , la famille*

$$\{1, \alpha, \dots, \alpha^{n-1}\}$$

est une base du F_q -espace vectoriel F_{q^n} .

Théorème 1.24 *Soit F_q un corps et $f(x) \in F_q[x]$, alors $F_q[x]/(f(x))$, est un corps si et seulement si $f(x)$ est irréductible sur F_q .*

Preuve. On note par I l'idéal principal $(f(x))$, supposons que $f(x)$ est irréductible sur F_q .

$F_q[x]/(f(x))$ est un anneau commutatif d'élément unité $I + e$ (où e est l'unité de F_q), il suffit de montrer que tout élément non nul de $F_q[x]/I$ admet un inverse dans $F_q[x]/I$. Soit $I + p(x) \in F_q[x]$ différent de zéro (c-à-d différent de I), donc $p(x) \notin I$, ce qui montre que $p(x)$ n'est pas multiple de $f(x)$, comme $f(x)$ est irréductible, alors $f(x)$ et $p(x)$ sont premiers entre eux et donc d'après le théorème de Bézout :

$$\exists u(x), v(x) \in F_q[x] \text{ tel que :}$$

$$f(x)u(x) + p(x)v(x) = e$$

alors on a :

$$e = p(x)v(x) = f(x)u(x) \in I$$

et par conséquent

$$I + e = I + p(x)v(x) = (I + p(x))(I + v(x)) = e$$

c-à-d $I + v(x)$ est l'élément inverse de $I + p(x)$. ■

1.4.3 Polynômes cyclotomiques

Racines $n^{\text{ièmes}}$ de l'unité

Définition 1.25 Soit F un corps. Une racine de $x^n - 1$ dans $F[x]$ est appelée une racine $n^{\text{ième}}$ de l'unité. L'ordre d'une racine $n^{\text{ième}}$ α de l'unité est le plus petit entier positif k tel que $\alpha^k = 1$. Une racine $n^{\text{ième}}$ de l'unité d'ordre n est dite primitive, le corps de décomposition S_n de $x^n - 1$ est appelé le corps cyclotomique associé. Dans la suite on aura besoin de la fonction φ de N dans N , dite d'Euler, qui est définie par

$$\varphi(n) = |\{m/1 \leq m \leq n \text{ et } (m, n) = 1\}|$$

pour m, n des entiers positifs. Si $n = p_1^{t_1} p_2^{t_2} \dots p_k^{t_k}$ où les p_i sont premier distincts, alors:

$$\varphi(n) = n(1 - \frac{1}{p_1}) \dots (1 - \frac{1}{p_k})$$

Polynômes cyclotomiques

Définition 1.26 Soit n un entier positif et F un corps dont la caractéristique ne divise pas n , et α une racine primitive $n^{\text{ième}}$ de l'unité. Le polynôme :

$$Q_n = (x - \alpha_1) \dots (x - \alpha_{\varphi(n)}) \in F(\alpha)[x]$$

où $\alpha_1, \dots, \alpha_{\varphi(n)}$ sont les racines primitives n^{ieme} de l'unité dans $F(\alpha)$, est appelé le n^{ieme} polynôme cyclotomique sur F . c'est-à-dire que Q_n a ses coefficients dans F_p . Soit α une racine primitive n^{ieme} de l'unité, alors il résulte que :

$$Q_n = \prod_i (x - \alpha^i)$$

où le produit est formé pour tout i avec $\text{pgcd}(i, n) = 1$. Le polynôme Q est de degré $\varphi(n)$. Soit $n = kd$ ainsi α^k d'ordre d , car $(\alpha^k)^d = \alpha^{kd} = \alpha^n = 1$, et est une racine primitive d^{ieme} de l'unité. Le d^{ieme} polynôme cyclotomique est de la forme :

$$Q_d = \prod_{\text{pgcd}(i, n)=1} (x - \alpha^{ik})$$

Toute racine n^{ieme} de l'unité est une racine primitive d^{ieme} de l'unité pour exactement un seul d .

1.5 Factorisation de $x^n - 1$

La factorisation du polynôme $x^n - 1$ joue un rôle important dans la recherche de tous les codes cycliques de longueur n sur F_q . Comme le polynôme générateur d'un code cyclique de longueur n sur F_q est un diviseur de $x^n - 1$, on est intéressé à déterminer les facteurs irréductibles de ce polynôme.

$$x^n - 1 = \prod_{d \mid n} Q_d \quad (\text{décomposition cyclotomique})$$

Un résultat important se déduit pour le polynôme cyclotomique Q_{p^m} , pour p premier et m entier positif, à savoir :

Corollaire 1.27

$$Q_{p^m} = 1 + x^{p^{m-1}} + \dots + x^{(p-1)p^{m-1}}.$$

En effet le théorème précédent et sachant que les diviseurs de p^m , p premier, sont 1, p , p^2 , ..., p^m alors :

$$\begin{aligned} x^{p^m} - 1 &= \prod_{d \mid p^m} Q_d = Q_1 Q_p \dots Q_{p^m} \\ Q_{p^m} &= \frac{x^{p^m} - 1}{Q_1 Q_p \dots Q_{p^{m-1}}} \\ &= \frac{x^{p^m} - 1}{x^{p^{m-1}} - 1} \\ &= 1 + x^{p^{m-1}} + \dots + x^{(p-1)p^{m-1}}. \end{aligned}$$

Exemple 1.28 Soit dans F_2 , le polynôme $x^9 - 1$, donnons sa décomposition cyclotomique. Comme les diviseurs de 9 sont : 1, 3, 9 on a

$$x^9 - 1 = \prod_{d \mid 9} Q_d = Q_1 Q_3 Q_9$$

d'après le corollaire on a :

$$\deg Q_1 = \varphi(1) = 1 \text{ et } Q_1 = x + 1$$

$$\deg Q_3 = \varphi(3) = 2 \text{ et } Q_3 = \frac{x^3 - 1}{x - 1} = x^2 + x + 1$$

$$\deg Q_9 = \varphi(9) = 6 \text{ et } Q_9 = \frac{x^9 - 1}{x^3 - 1} = x^6 + x^3 + 1$$

D'où l'écriture du polynôme

$$x^9 - 1 = (1 + x)(1 + x + x^2)(1 + x^3 + x^6)$$

1.6 Construction d'un corps fini

Pour déterminer les éléments d'un corps fini F_q on utilise l'anneau quotients $F_q[x]/(f(x))$ où $f(x)$ est un polynôme irréductible sur F_q .

Soit F_q un corps fini et $f(x) \in F_q[x]$, un polynôme irréductible de degré n . Alors :

$$F_q[x]/(f(x)) = \{a_0 + a_1x + \dots + a_{n-1}x^{n-1} : a_i \in F_q\}$$

est un espace vectoriel sur F_q de dimension n et de base $\{1, \alpha, \dots, \alpha^{n-1}\}$, avec

$$\alpha = [x] + (f(x)) \text{ où } \bar{\alpha} = 0.$$

On sait que le corps fini F_{p^n} est un espace vectoriel de dimension n sur F_p , de plus, F_{p^n} est une extension simple, c-à-d $F_{p^n} = F_p(\alpha)$, est tous les $(n + 1)$ éléments de F_{p^n} seront linéairement dépendants .

Donc ils existent $a_0, a_1, \dots, a_n \in F_p$ tel que :

$$a_0 + a_1\alpha + \dots + a_n\alpha^n = 0$$

Ce qui montre que α est une racine de polynôme

$$a_0 + a_1x + \dots + a_nx^n \in F_p[x].$$

Soit $f(x)$ un polynôme minimal de α (irréductible unitaire).

$$F_{p^n} = F_p(\alpha) \cong F_p[x]/(f(x)).$$

On détermine un polynôme irréductible unitaire de degré n sur F_p , et on construit $F_p[x]/(f(x))$.

Exemple 1.29 construire de F_9 “corps finie à 9 éléments”

Dans F_3 , le polynôme $g(x) = x^2 + x + 2$ est irréductible, on détermine les éléments de F_{3^2} en le regardant comme extension est obtenue par adjonction à F_3 d'une racine de $g(x)$, ainsi $F_{3^2} = F_3[x]/(g(x))$, soit α une racine de $g(x)$, alors $\{1, \alpha\}$ est une base de F_{3^2} .

$$\begin{aligned} F_3[x]/(g(x)) &= \{a_0 + a_1\alpha \mid a_0, a_1 \in F_3\} \\ &= \{0, 1, 2, \alpha, 2\alpha, 1 + \alpha, 2 + \alpha, 1 + 2\alpha, 2 + 2\alpha\} \text{ (Représentation polynomiale).} \end{aligned}$$

Tout polynôme de corps $F_3[x]/(g(x))$, peut être modulo $g(x)$ en utilisant le fait que :

$$g(\alpha) = 0$$

Dans F_3 , c-à-d que : $\alpha^2 = 2\alpha + 1$, et on aura :

$$\begin{aligned} F_{3^2} &= F_3[x]/(g(x)) \\ &= \{0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\} \text{ (Représentation en puissance de } \alpha \text{).} \end{aligned}$$

Chapitre 2

Généralités sur les Codes Linéaires et Cycliques

2.1 *Introduction*

Par codes, on peut entendre plusieurs concepts bien distincts : cryptographie (RSA,...), codes de compression (Huffman,...), codes correcteurs d'erreurs.

Dans ce chapitre, on s'intéresse aux codes correcteurs d'erreur, plus précisément à la famille des codes linéaires et cycliques. Lorsqu'on envoie un message à travers un canal de transmission des données (par exemple : en téléchargeant ce cours sur internet), des erreurs de transmission peuvent se produire.

2.2 Les Codes

Soit $A = \{a_1, \dots, a_q\}$ un ensemble que nous appellerons un alphabet de code et soit A^n l'ensemble de toutes les chaînes de longueur n sur A .

Nous dirons que chaque sous-ensemble $C \subset A^n$ s'appelle un code. Chacune des chaînes c de C est appelée mot du code. De plus, nous dirons qu'un code $C \subset A^n$ est de cardinalité M si $M = |C|$. La dimension n de A^n est appelée la longueur du code. Un code de longueur n contenant M mots sera appelé un (n, M) -code. Le corps fini à q éléments sera noté F_q . L'espace vectoriel de dimension n sur F_q sera noté F_q^n .

Exemple 2.1 1) $C = \{011, 101, 110, 000\}$ est un code de longueur 3 sur $A = \{0, 1\} = F_2$.

2) $C = \{aa, ba, ab\}$ est un code de longueur 2 sur $A = \{a, b\}$ et de cardinale 3.

Distance de Hamming

Définition 2.2 La distance de Hamming entre deux mots $x = \{x_1, x_2, \dots, x_n\}$ et $y = \{y_1, y_2, \dots, y_n\}$ de A^n , et l'on note $d(x, y)$, est le nombre d'indice i de $\{1, 2, \dots, n\}$ telle que $x_i \neq y_i$. C'est-à-dire :

$$d(x, y) = |\{i \mid x_i \neq y_i\}|.$$

On remarque que la distance de Hamming est une vraie distance au sens numérique de terme. Rappelons brièvement les propriétés d'une distance $d(x, y)$, faciles à vérifier sur d .

- 1) $d(x, y) = d(y, x) \geq 0$
- 2) $d(x, y) = 0$ si et seulement si $x = y$
- 3) $d(x, y) \leq d(x, z) + d(z, y)$

Distance minimale d'un code

Définition 2.3 La distance minimale d'un code C est la distance minimum entre deux mots distincts de code. On la note d :

$$d = \min \{d(x, y) \mid x, y \in C \text{ et } x \neq y\}.$$

par exemple, $d(0011, 0010) = 1$, $d(1120, 2200) = 3$

Le poids de Hamming

Définition 2.4 Le poids de Hamming d'un mot $x = \{x_1, x_2, \dots, x_n\}$, noté $wt(x)$, est le nombre d'indice i telle que $x_i \neq 0$.

$$wt(x) = |\{i \mid x_i \neq 0\}| = d(x, 0)$$

par exemple, dans F_2^3 , nous avons $wt(101) = 2$ et $wt(001) = 1$.

2.3 Codes Linéaires

Dans notre étude des codes contenues dans F_q^n , nous allons concentrer sur les codes linéaire, c'est-à-dire ceux qui ont une structure d'espace vectoriels. Ce qui permet d'utiliser les outils de l'algèbre linéaire.

Définition 2.5 *Un code linéaire C de dimension k et de longueur n sur F_q est un sous-espace vectoriel de dimension k de F_q^n . Si $q = 2$, on dit que C est un code binaire. Pour un code linéaire C . On retrouve la distance de Hamming par la formule*

$$d(x, y) = w(x - y)$$

la distance minimale du code C définie par :

$$d(C) = wt(C) = \min \{wt(x), x \in C \text{ et } x \neq 0\}$$

Exemple 2.6 *Soit C le code linéaire de longueur 4 et de dimension 2.*

$$C = \{0000, 1011, 0101, 1110\}, \quad d = \min \{wt(x) \mid x \in C \text{ et } x \neq 0\} = 2.$$

borne du Singleton

Théorème 2.7 *Soit un code $C[n, k, d]$. On a l'inégalité suivante : $d \leq n - k + 1$.*

Preuve. Soit E le sous-espace vectoriel de F_q^n constitué des mots dont les $k-1$ dernières composantes non nulles. Alors :

$$\dim(E) = n - (k - 1) = n - k + 1$$

On en déduit : $\dim(C) + \dim(E) = n + 1 > n$, ce qui implique $C \cap E \neq \{0\}$.

Soit m un mot non nul de $C \cap E$, on a

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

■

Polynômes énumérateur des poids

La distribution des poids d'un code est le plus souvent représentée par un polynôme à deux indéterminées. La raison de ce choix est la relation de Mac Williams ci-dessous, qui est un résultat clé de la théorie des codes.

Définition 2.8 *Le polynôme des poids d'un code C , linéaire ou non linéaire, est :*

$$W_C(X, Y) = \sum_{i=0}^n A_i X^{n-i} Y^i, \text{ ou } A_i = |\{u \in C \mid w(u) = i\}|.$$

Le théorème de Mac Williams est le suivant :

Théorème 2.9 *Soit C est un code linéaire de longueur n sur F , où F désigne un corps fini. Alors les polynômes des poids de C et de son dual C^\perp sont liés par la relation suivante (dite relation de Mac Williams):*

$$F = GF(2) : W_{C^\perp}(X, Y) = \frac{1}{|C|} W_C(X + Y, X - Y) \quad \text{cas binaire}$$

$$F = GF(q) : W_{C^\perp}(X, Y) = \frac{1}{|C|} W_C(X + (q-1)Y, X - Y) \quad \text{cas général}$$

où $GF(q)$ est le corps fini à q éléments.

Exemple 2.10 $F = \{0, 1\}$, $C = \{0000, 1111\}$

$$W_C(X, Y) = X^4 + Y^4, C \text{ est un code } [4, 1, 4]$$

$$W_{C^\perp}(X, Y) = \frac{1}{2} [(X + Y)^4 + (X - Y)^4] = X^4 + 6X^2Y^2 + Y^4, C^\perp \text{ est un code } [4, 3, 2].$$

Matrice génératrice

Définition 2.11 *Soit C un code linéaire de dimension k et de longueur n . Une matrice génératrice de C est une matrice G à k lignes et n colonnes, dont les lignes forment une base de C . Il est immédiat de vérifier que le code C peut être représenté de la façon suivante:*

$$C = \{c \in F_q^n / \exists x \in F_q^k : c = xG\}.$$

La procédure de codage est alors triviale. Il suffit de coder le vecteur x de longueur k par le mot xG du code.

Exemple 2.12 Soit G la matrice génératrice du $[3, 2]$ -code binaire C telle que:

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

Déterminions

$$C : C = \{c_1(1, 1, 0) + c_2(0, 1, 1) / c_1, c_2 \in F_2\}$$

$$C = \{000, 011, 110, 101\}.$$

Ainsi le code C est de paramètres $[3, 2, 2]$ et $|C| = q^k = 2^2 = 4$, et par exemple le message 11 est codé par $c = 11G = 101$.

Code dual

L'espace vectoriel F_q^n peut être muni d'un produit scalaire de la façon suivante. Soient $x = x_1, x_2, \dots, x_n$ et $y = y_1, y_2, \dots, y_n$ alors le produit scalaire de x et y nous est donné par :

$$\langle x, y \rangle = x_1y_1 + x_2y_2 + \dots + x_ny_n.$$

A l'aide de ce produit scalaire nous allons définir l'un des concepts fondamentaux de la théorie du codage, soit le concept de code dual.

Définition 2.13 Soit C un $[n, k]$ -code. Le code C^\perp défini par :

$$C^\perp = \{x \in F_q^n : \forall c \in C : \langle x, c \rangle = 0\}$$

est appelé le code dual du code C .

Exemple 2.14 Le code dual du code $C = \{0000, 1111\}$ est,

$$C^\perp = \{000, 1100, 1010, 1001, 0110, 0101, 0011, 1111\}.$$

Matrice de contrôle

Définition 2.15 Soit C un code linéaire de dimension k et de longueur n . Une matrice de contrôle de C est une matrice H à $n - k$ lignes et n colonnes, dont les lignes forment une base de C^\perp .

Proposition 2.16 Soit C un code linéaire et soit H une matrice de contrôle de C . On a l'équivalence :

1. $c \in C$
2. $H^t c = 0$.

Exemple 2.17 Supposons $q = 2$, $n = 6$, $k = 3$ (donc $M = 2^3 = 8$).

Soit C le code de paramètres $[6, 3]$ donné par la matrice de contrôle

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

$$c = (c_1, c_2, c_3, c_4, c_5, c_6) \in C \Leftrightarrow H^t c = 0$$

$$\Leftrightarrow \begin{cases} c_1 + c_2 + c_4 = 0 \\ c_1 + c_3 + c_5 = 0 \\ c_2 + c_3 + c_6 = 0 \end{cases} \Leftrightarrow \begin{cases} c_4 = -c_1 - c_2 \\ c_5 = -c_1 - c_3 \\ c_6 = -c_2 - c_3 \end{cases}$$

$$c \in C \Leftrightarrow c = (c_1, c_2, c_3, -c_1 - c_2, -c_1 - c_3, -c_2 - c_3)$$

$$\Leftrightarrow c = c_1(1, 0, 0, -1, -1, 0) + c_2(0, 1, 0, 1, 0, 1) + c_3(0, 0, 1, 0, 1, 1)$$

$$\Leftrightarrow c = c_1 v_1 + c_2 v_2 + c_3 v_3.$$

Donc C est le sous espace de F_2^6 sur F_2 engendré par les vecteurs v_1, v_2, v_3 .

Si le message $a = 011$ est transmis, alors le mot de code correspondant est $c = 011110$.

Le code C contient 2^3 mots de code :

$$\{000000, 001011, 010101, 011110, 100110, 101101, 110011, 111000\}.$$

Codes systématiques

Définition 2.18 Un $[n, k, d]_q$ -code est systématique s'il possède une matrice génératrice quel'on peut écrire par bloc sous la forme :

$$G = (I_k \setminus B).$$

Ou I_k désigne la matrice unité à k lignes et k colonnes, et B est une matrice de taille $k \times (n - k)$.

Corollaire 2.19 Soit C un code systématique, et soit $G = (I_k \setminus B)$ une matrice génératrice normalisée de C . La matrice $H = (-{}^tB \setminus I_{n-k})$ est une matrice de contrôle de C .

Preuve. La matrice tG s'écrit par blocs sous la forme ${}^tG = \begin{pmatrix} I_k \\ {}^tB \end{pmatrix}$, on a donc :

$$H{}^tG = (-{}^tB \setminus I_{n-k}) \begin{pmatrix} I_k \\ {}^tB \end{pmatrix} = -{}^tB + {}^tB = 0. \blacksquare$$

Exemple 2.20 a) L'application $f(x_1, x_2, x_3) = (x_1, x_2, x_3, x_1 + x_3, x_2 + x_3, x_1 + x_2 + x_3, x_3)$, définit un code systématique C de paramètres $[7, 3]$ sur F_2 .

$$\text{L'écriture : } c = (c_1, c_2, c_3, c_4, c_5, c_6, c_7) = (x_1, x_2, x_3) \begin{bmatrix} 1 & 0 & 1 & \vdots & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & \vdots & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & \vdots & 1 & 1 & 1 & 1 \end{bmatrix}$$

met en évidence une matrice génératrice de C , et l'on déduit la matrice de contrôle

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

b) Soit C le code linéaire de paramètres $[7, 4, 3]$ dont la matrice génératrice est donnée par :

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

D'où la matrice de contrôle est donnée par :

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

2.4 Codes Cycliques

Dans la famille des codes linéaire, il existe une classe très important, celle des codes en blocs linéaires et cyclique, dans un code cyclique toute opération de décalage cyclique appliquée à un mot de code fournit un autre mot de code.

Définition 2.21 *Un code linéaire C de longueur n sur F_q^n est dit cyclique si l'ensemble de ses mots est invariant par décalage circulaire :*

$$(c_0, c_1, \dots, c_{n-1}) \in C \Rightarrow (c_{n-1}, c_0, c_1, \dots, c_{n-2}) \in C.$$

Exemple 2.22 *i) Le code binaire $C = \{000, 101, 011, 110\}$ est cyclique.*

ii) Le code binaire $C = \{0000, 1001, 0110, 1111\}$ n'est pas cyclique. Il est cependant équivalent à un code cyclique (il faut échanger les troisième et quatrième coordonnées).

On va identifier F_q^n à l'algèbre $F_q[x]/(x^n - 1)$ via l'application

$$(c_0, c_1, \dots, c_{n-1}) \rightarrow c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1} \text{ (représentation polynômiale).}$$

L'intérêt de cette identification est que le shift à droite s'identifie à la multiplication par x dans $F_q[x]/(x^n - 1)$. Ainsi, Un code linéaire est cyclique si et seulement si c'est un idéal de $F_q[x]/(x^n - 1)$.

C , qui est un espace-vectoriel stable par tout itéré de shifts à droite, s'identifie à un sous-ensemble de $F_q[x]/(x^n - 1)$ stable par multiplication par les polynômes de $F_q[x]/(x^n - 1)$, il s'identifie donc à un idéal de $F_q[x]/(x^n - 1)$.

Proposition 2.23 *Un code linéaire est cyclique si et seulement si c'est un idéal de*

$$F_q[x]/(x^n - 1).$$

Définition 2.24 *On appelle polynôme générateur du code cyclique $C \subset F_q^n$, le polynôme unitaire non nul de plus petit degré représentant un élément de C .*

Exemple 2.25 *Soit C un $[7, 3]$ -code cyclique. $g(x) = 1 + x^2 + x^3 + x^4$, est un polynôme générateur de C .*

Théorème 2.26 *Soit C un code cyclique de longueur n et soit $g(x)$ son polynôme générateur, Alors :*

1. *Le code C est l'idéal $\langle g(x) \rangle$ de l'anneau $F_q[x]/(x^n - 1)$*
2. *Le polynôme $g(x)$ divise $x^n - 1$ dans $F_q[x]$ (et dans $F_q[x]/(x^n - 1)$).*

Preuve. 1. Un quelconque élément de C peut être représenté par un polynôme $c(x)$ de degré $\leq n - 1$. La division euclidienne de $c(x)$ par $g(x)$ s'écrit :

$$c(x) = q(x)g(x) + r(x)$$

où $\deg r(x) < \deg g(x)$. Mais dans $F_q[x]/(x^n - 1)$ on a $r(x) = c(x) - q(x)g(x)$ où $c(x)$ et $q(x)g(x)$ sont tous les deux dans C . On en déduit que $r(x) \in C$ et que $r(x) = 0$ par hypothèse de minimalité du degré de $g(x)$.

2. L'argument est similaire à celui du 1. On écrit la division euclidienne dans $F_q[x]$ de $x^n - 1$ par $g(x)$:

$$x^n - 1 = q(x)g(x) + r(x)$$

où $\deg r(x) < \deg g(x)$. On en déduit donc que $r(x) = q(x)g(x) \bmod (x^n - 1)$, donc que $r(x) \in C$, ce qui implique $r(x) = 0$ par hypothèse de minimalité du degré de $g(x)$.

■

Représentation matricielle

On a vu que tout mot $c \in C$ peut s'obtenir en multipliant g (de degré r) par un polynôme a sans avoir à réduire modulo $x^n - 1$: $c(x) = a(x)g(x)$. Puisque $\deg(c(x)) < n$ et $\deg(g(x)) = r$, on obtient $\deg(a(x)) < n - r$. Utilisons maintenant la notation matricielle. On a

$$c = aG$$

où $c = (c_0, \dots, c_{n-1})$, $a = (a_0, \dots, a_{n-r})$ et G est une matrice circulante $(n - r) \times n$ dont la i ème ligne contient le mot $x^{i-1}g$

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \dots & g_r & 0 \\ & g_0 & g_1 & \dots & g_{r-1} & \\ & & & \dots & & \\ & & & \dots & & \\ & & & \dots & & \\ & & & \dots & & \\ 0 & & g_0 & \dots & \dots & g_r \end{bmatrix}$$

Son rang est $n - r$. G est une matrice génératrice du code qui a $\dim(C)$ lignes. Ainsi $\deg(g) = n - \dim(C)$.

Dual d'un code cyclique

Théorème 2.27 *Si g est le polynôme générateur de C , alors $h = (x^n - 1)/g$ est le polynôme générateur de C^\perp .*

Preuve. Soit $c \in C$ et $c' \in \langle h \rangle$. On peut écrire $c' = a'g$ et $c' = a'h$. Ce qui implique que $cc' = aa'gh = 0$ puisque $gh = 0$. donc $\langle h \rangle \subset C^\perp$. Il reste à montrer que $\dim(C^\perp) = n - k$ si $\dim(C) = k$. On sait que $\deg(h) = n - \deg(g) = k$ donc $\dim(h) = n - \deg(h) = n - k$. ■

Exemple 2.28 Soit $C = \langle g \rangle$ avec $g = x^3 + x + 1$, Le polynôme générateur de C^\perp est $(x^3 + x^2 + 1)(x + 1) = x^4 + x^2 + x + 1$. La matrice génératrice de C^\perp est :

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

On remarque que les colonnes de la matrice correspondent à tous les 3-uplets binaires non nuls. Le code C est donc le code de Hamming.

Construction d'un code cyclique

Il est maintenant possible d'exhiber tous les codes cycliques de longueur n grâce à la recherche de tous les diviseurs de $(x^n - 1)$. Le résultat suivant permet ensuite de trouver les mots du codes :

Soit C un code cyclique de longueur n et $g(x)$ son polynôme générateur tel que $d^\circ(g) = t$. La famille suivante

$$\{g(x), x \cdot g(x), \dots, x^{n-t-1} \cdot g(x)\}$$

est une base de C et la dimension du code est $n - t$.

Exemple 2.29 construction d'un code cyclique (7, 4)

Nous avons vu que

$(x^7 - 1) = (x - 1)(x^3 + x^2 + 1)(x^3 + x + 1)$ dans F_2 , ce qui nous conduit à :

$$C_0 : g_0(x) = x^7 - 1$$

$$C_1 : g_1(x) = x - 1$$

$$C_2 : g_2(x) = x^3 + x + 1$$

$$C_3 : g_3(x) = x^3 + x^2 + 1$$

$$C_4 : g_4(x) = g_1 \cdot g_2 = x^4 + x^3 + x^2 + 1$$

$$C_5 : g_5(x) = g_1 \cdot g_3 = x^4 + x^3 + x + 1$$

$$C_6 : g_6(x) = g_2 \cdot g_3 = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$$

$$C = \{C_0, C_1, C_2, C_3, C_4, C_5, C_6\}$$

$$= \{10000001, 11000000, 11010000, 10110000, 10111000, 11011000, 11111110\}.$$

Chapitre 3

Codes Optimaux $[n, \frac{n}{2}]$ sur F_2

3.1 *Introduction*

dans cette partie nous considérons la distance minimale d_{\min} d'un code cyclique de longueur n pair (sa dimension $\frac{n}{2}$) et de rendement $\frac{1}{2}$, nous établissons par calcul électronique en utilisant l'algorithme de Chen, la valeur de la distance minimale pour $n \leq 50$.

3.2 Calcul de la distance minimum des Codes Cycliques de Rendement $\frac{1}{2}$ sur F_2 pour $n \leq 50$

3.2.1 Codes Cycliques Optimaux sur GF(2)

Soit le corps fini $F_2 = \{0, 1\}$, un code $C[n, k]$ linéaire binaire est un sous espace vectoriel de dimension k sur F_2^n . Les éléments de C sont appelés mots de code, et le poids $w(x)$ d'un mot de code x est le nombre de positions où x diffère de zéro. La distance de Hamming $d(x, y)$ entre deux mots de codes est définie par : $d(x, y) = w(x - y)$. La distance minimale d'un code linéaire C est :

$$d(C) = \min \{d(x, y) \mid x, y \in C, x \neq y\}.$$

Si C un code linéaire alors $d(C)$ est égale au poids minimum de tous ses mots de code non nuls. Le problème fondamentale de la théorie du codage est :

3.2. Calcul de la distance minimum des Codes Cycliques de Rendement $\frac{1}{2}$ sur F_2 pour $n \leq 50$

- trouver $d_q(n, k)$, la plus grande valeur de d pour laquelle un code $C[n, k, d]_q$ existe .

Un code qui atteint cette valeur est appelé un code optimal .

Codes Cycliques $C[2, 1]_2$:

L'algorithme de Chen nous permet de calculer les distances minimumes. En utilisant Mathematica 5.2, la décomposition en facteurs irréductibles du :

$$x^2 - 1 = (1 + x)^2$$

sur le corps F_2 nous donne un seul polynôme générateur $g(x) = 1 + x$. On enregistre dans la table 1 le résultat de calcul du poids et mot de code pour le polynôme générateur $g(x)$:

Table 1

g(x)	mot de code (a)	wt(a)
11	11	2

Codes Cycliques $C[4, 2]_2$:

De même la factorisation de $x^4 - 1$ nous donne un seul choix pour le polynôme générateur de degré 2 :

$$x^4 - 1 = (1 + x)^2(1 + x)^2.$$

Le mot de code et leurs poids dans la table qui suit :

Table 2

g(x)	mot de code (a)	wt(a)
101	1010	2

Codes Cycliques $C[6, 3]_2$:

La factorisation de $x^6 - 1$ donne un polynôme de degré = 1, et un polynôme de degré = 2

$$x^6 - 1 = (1 + x)(1 + x + x^2)$$

on a un seul polynôme générateur $g(x)$ de degré 3. On enregistre dans la table 3 le résultat de calcul du poids de mot de code pour le polynôme générateur $g(x) = (1 + x)(1 + x + x^2)$.

Table 3

g(x)	mot de code (a)	wt(a)
1001	100100	2

Codes Cycliques $C[8, 4]_2$:

Pour les codes cycliques $C[8, 4]$, la factorisation de $x^4 - 1$ nous donne 2 choix possibles pour le polynôme générateur de degré 4:

$$x^8 - 1 = (x + 1)(x + 2)(1 + x^2)(1 + x^4).$$

D'où la table des mots et leur poids correspondants :

Table 4

g(x)	mot de code (a)	wt(a)
10001	10001000	2
11111	10000100	2

Codes Cycliques $C[10, 5]_2$:

Dans ce cas on a un seul choix pour $g(x)$ et la factorisation de $x^{10} - 1$ en facteurs irréductibles donne :

$$x^{10} - 1 = (1 + x)^2(1 + x + x^2 + x^3 + x^4)^2.$$

Ainsi on aura le mot de code et leur poids :

Table 5

g(x)	mot de code (c)	wt(a)
100001	1000010000	2

Codes Cycliques $C[12, 6]_2$:

Pour les codes $C[12, 6]$, on a trois choix pour le polynôme générateur de degré 6 :

$$x^{12} - 1 = (1 + x)^4(1 + x + x^2)^4.$$

On note dans la table (6) les mots de codes et leurs poids correspondant:

Table 6

g(x)	mot de code (a)	wt(a)
1000001	100000100000	2
1110111	101000101000	4
1101011	001000100010	3

Codes Cycliques $C[14, 7]_2$:

La décomposition en facteurs irréductibles de $x^{14} - 1$ comrise 3 choix pour le polynôme générateur de degré 7 :

$$x^{14} - 1 = (1 + x)^2(1 + x + x^3)^2(1 + x^2 + x^3)^2.$$

Par conséquent on a la table qui nous donne tous les mots de codes et leurs poids correspondant :

Table 7

g(x)	mot de code (a)	wt (a)
11110011	10001000001010	4
11001111	10100000100010	4
10000001	10000001000000	2

Codes Cycliques $C[16, 8]_2$:

La factorisation de $x^{16} - 1$ nous donne 1 seul choix pour le polynôme générateur de degré 8 :

$$x^{16} - 1 = (1 + x^8)(1 + x^8).$$

Par conséquent on a la table ou on note le mot de code et leur poids correspondant :

Table 8

g(x)	mot de code (a)	wt (a)
100000001	1000000010000000	2

Codes Cycliques $C[18, 9]_2$:

De même, la factorisation de $x^{18} - 1$ nous donne aussi un seul choix pour le polynôme

générateur de degré 9 :

$$x^{18} - 1 = (1 + x)^2(1 + x + x^2)^2(1 + x^3 + x^6)^2.$$

On note dans la table (9) le mot de code et leur poids correspondant :

Table 9

g(x)	mot de code (a)	wt(a)
1000000001	10000000001000000000	2

Codes Cycliques $C[20, 10]_2$:

De même, la factorisation de $x^{20} - 1$ nous donne une seule possibilité pour le polynôme générateur de degré 10 :

$$x^{20} - 1 = (1 + x)^4(1 + x + x^2 + x^3 + x^4)^4.$$

On enregistre dans la table (10) le mot de code et leur poids :

Table 10

g(x)	mot de code (a)	wt(a)
10000000001	1000000000010000000000	2

Codes Cycliques $C[22, 11]_2$:

Ainsi la factorisation de $x^{22} - 1$ nous donne un seul choix pour le polynôme générateur de degré 11 :

$$x^{22} - 1 = (1 + x)(1 + x + x^2 + x^3 + x^4 + \dots + x^8 + x^9 + x^{10}).$$

Et Par conséquent on note dans la table (11) le mot de code et leur poids correspondant:

Table 11

g(x)	mot de code (a)	wt(a)
1000000000001	10000000000001000000000000	2

Codes Cycliques $C[24, 12]_2$:

Pour les codes cycliques de paramètres $[24, 12]_2$, nous avons 5 choix possibles pour le polynôme génératur $g(x)$ de degré 12 :

$$x^{24} - 1 = (1 + x)^8(1 + x + x^2)^8.$$

On résume les paramètres de code dans la table qui suit :

Table 12

g(x)	mot de code (a)	wt(a)
1010001000101	0000100000000100000001000	3
1010100010101	1000100000000100010000000	4
10000000000001	1000000000000100000000000	2
1110110110111	1010000000000101000000000	4
1101010101011	1001000000000100100000000	4

Codes Cycliques $C[26, 13]_2$:

La décomposition de $x^{26} - 1$ en facteurs irréductibles nous donne un seul choix du polynôme générateur $g(x)$ de degré 13

$$x^{26} - 1 = (1 + x)(1 + x + x^2 + x^3 + \dots + x^8 + x^9 + x^{10} + x^{11} + x^{12}).$$

Dans la table 13 on note le mot de code et leur poids correspondant :

Table 13

g(x)	mot de code (a)	wt(a)
100000000000001	10000000000000010000000000000	2

Codes Cycliques $C[28, 14]_2$:

Pour les codes cycliques binaires $[28, 14]$, la factorisation de $x^{28} - 1$ nous donne 5 choix possibles pour le polynôme générateur de degré 14 :

$$x^{28} - 1 = (x + 1)^4(1 + x + x^3)^4(1 + x^2 + x^3)^4.$$

Par conséquent, on résume les paramètres des codes dans la table qui suit :

Table 14

$g(x)$	mot de code (a)	wt(a)
101010100000101	1000000010000000000010001000	4
101000001010101	10001000000000000100000001000	4
1000000000000001	10000000000000010000000000000	2
110011101001111	100000010000000100000001000000	4
111100101110011	100000010000000100000001000000	4

Codes Cycliques $C[30, 15]_2$:

La factorisation de $x^{30} - 1$ nous donne setp choix pour le polynôme générateur de degré 15 :

$$x^{30} - 1 = (1 + x)^2(1 + x + x^2)^2(1 + x + x^4)^2 \\ (1 + x^3 + x^4)^2(1 + x + x^2 + x^3 + x^4)^2.$$

Les mots de codes et son poids sont consignés dans la table (15)

Table 15

$g(x)$	mot de code (a)	wt(a)
1010100101001011	1100000000000100110000000000100	6
1101111000100111	1100000000000100110000000000100	6
1101001010010101	1100100000000000110010000000000	6
1110010001111011	1100100000000000110010000000000	6
1010001111010111	1000010000000000100001000000000	4
1110101111000101	1000010000000000100001000000000	4
1000000000000001	1000000000000000100000000000000	2

Codes Cycliques $C[32, 16]_2$:

Dans ce cas on a un seul choix pour $g(x)$ et la factorisation de $x^{32} - 1$ en facteurs irréductibles donne :

$$x^{32} - 1 = (1 - x^{16})(1 + x^{16}).$$

Et Par conséquent on note dans la table (16) le mot de code et leur poids correspondant:

Table 16

$g(x)$	mot de code (a)	wt(a)
100000000000000001	1000000000000000001000000000000000	2

Codes Cycliques $C[34, 17]_2$:

La factorisation de $x^{34} - 1$ nous donne trois possibilité pour le polynôme générateur de degré 17 :

$$x^{34} - 1 = (1 + x)^2(1 + x^3 + x^4 + x^5 + x^8)^2 \\ (1 + x + x^2 + x^4 + x^6 + x^7 + x^8)^2.$$

On résume les paramètres de code dans la table qui suit :

Table 17

$g(x)$	mot de code (a)	wt(a)
100000000000000001	1000000000000000001000000000000000	2
110000111111000011	1010000000100000000000100000001010	6
111111001100111111	1010000010000000000000001000001010	6

Codes Cycliques $C[36, 18]_2$:

Pour les codes cycliques binaires $C[36, 18]$, la factorisation de $x^{36} - 1$ nous donne 5 choix possibles pour le polynôme générateur de degré 18 :

$$x^{36} - 1 = (x + 1)^4(1 + x + x^2)^4(1 + x^3 + x^6)^4.$$

On note les mots de code et leurs poids correspondant dans la table (18) :

Table 18

g(x)	mot de code (a)	wt(a)
1001000001000001001	0000000100000000000010000000000001000000	3
10010010000001001001	10000001000000000000010000001000000000000	4
1101010101010101011	1001000000000000000010010000000000000000	4
100000000000000000001	1000000000000000000001000000000000000000	2
1110110110110110111	1010000000000000000010100000000000000000	4

Codes Cycliques $C[38, 19]_2$:

La décomposition de $x^{38} - 1$ en facteurs irréductibles nous donne un seul choix pour le polynôme générateur $g(x)$ de degré 19 :

$$x^{38} - 1 = (1 + x)^2(1 + x + x^2 + x^3 + x^4 + \dots + x^{16} + x^{17} + x^{18})^2.$$

Et Par conséquent on note dans la table (19) le mot de code et leur poids correspondant:

Table 19

g(x)	mot de code (a)	wt(a)
100000000000000000001	1000000000000000000001000000000000000000	2

Codes Cycliques $C[40, 20]_2$:

Pour les codes cyclique de paramètres $[40, 20]_2$, nous donne trois choix possibles pour le polynôme générateur de degré 20 :

$$x^{40} - 1 = (1 + x)^8(1 + x^2 + x^3 + x^4)^8.$$

Par conséquent, on résume les paramètres de code dans la table qui suit :

Table 20

g(x)	mot de code (a)	wt(a)
11110111011101111111	100000000010001000000001000000100001000000	6
110010100101001010011	10001000000000000000010001000000000000000	4
1000000000000000000001	10000000000000000000010000000000000000000	2

Codes Cycliques $C[42, 21]_2$:

Pour les codes cycliques $[42, 21]$, la factorisation de $x^{42} - 1$ nous donne 19 possibilités

pour le polynôme générateur de degré 21 :

$$x^{42} - 1 = (1+x)^2(1+x+x^2)^2(1+x+x^3)^2(1+x^2+x^3)^2 \\ (1+x+x^2+x^4+x^6)^2(1+x^2+x^4+x^5+x^6)^2.$$

Dans la table 21 on note les mots de code et leurs poids correspondant :

Table 21

g(x)	mot de code(a)	wt(a)
1000000000000000000001	10000000000000000000010000000000000000000	2
1011110110110100101101	101010100000000000000000101000100010000000	8
1001000000001001001001	100000100000000000000000100000000000100000	4
1100011011111100111001	100000010000000000000010000001000000000000	4
1001110011111101100011	100000010000000000000010000001000000000000	4
1011110100100100001001	101010000000000000000000101000000000000010	6
1001000000001001001001	100000100000000000000000100000000000100000	4
1011110110110100101101	101010100000000000000000101000100010000000	8
1000101011001110111101	1001000000000001000001001000000000000100000	6
1100000111101110011011	1001000000000001000001001000000000000100000	6
1101100111011110000011	100100000100000000000100100000100000000000	6
1011110111001101010001	100100000100000000000100100000100000000000	6
1010010010010010010011	10000001000000000000010000001000000000000	4
1100100100100100100101	10000001000000000000010000001000000000000	4
1110010011100111101001	101000000000000100000000100000100000001000	6
1101001101000000001101	110000010000000000000000100000010000100000	6
1011000000001011001011	1010000000000001000000000100000100000001000	6
1001011110011100100111	1000001000000000010000000000010100010000000	6
1001001001000000001001	1000000000000100000000000000000100000100000	4

Codes Cycliques $C[44, 22]_2$:

Pour les codes cycliques binaires $[44, 22]$, la factorisation de $x^{44} - 1$ nous donne un seul choix pour le polynôme générateur de degré 22 :

$$x^{44} - 1 = (x + 1)^4(1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10})^4.$$

Dans la table (22) on note le mot de code et leur poids correspondant:

Table 22

g(x)	mot de code (a)	wt(a)
10000000000000000000000001	100000000000000000000000001000000000000000000000000	2

Codes Cycliques $C[46, 23]_2$:

Pour les codes cycliques binaires $[46, 23]$, nous avons trois choix possibles pour le polynôme générateur $g(x)$ de degré 23 :

$$\begin{aligned} x^{46} - 1 &= (x + 1)^2(1 + x + x^5 + x^6 + x^7 + x^9 + x^{11})^2 \\ &= (1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11})^2. \end{aligned}$$

Par conséquent, on résume les paramètres de code dans la table qui suit :

Table 23

g(x)	mot de code (a)	wt(a)
110011001111110000001111	101010100000000000000000000000001000100000100000010	8
111100000011111100110011	1010101000000000000000000000000010000000001000101000	8
1000000000000000000000000001	100000000000000000000000001000000000000000000000000	2

Codes Cycliques $C[48, 24]_2$:

La décomposition de $x^{48} - 1$ en facteurs irréductibles nous donne 8 polynômes générateurs possibles de degré 24 :

$$x^{48} - 1 = (1 + x)^{16}(1 + x + x^2)^{16}.$$

Et Par conséquent on note dans la table (24) les mots de code et leurs poids correspondant:

Table 24

$g(x)$	mot de code (a)	wt(a)
1110001101100011011000111	100000000000010000000000001000000000001000000000000	4
1010001000100010001000101	100000100000000000000000001000001000000000000000000	4
1101010101010101010101011	100100000000000000000000001001000000000000000000000	4
100000000000000000000000001	100000000000000000000000001000000000000000000000000	2
1110110110110110110110111	101000000000000000000000001010000000000000000000000	4
1010100010100010100010101	100010000000000000000000001000100000000000000000000	4
1101100011010101100011011	100000001000000000000000001000000010000000000000000	4
1000100010000000100010001	10000000100000000000000000100000001000000000000000	4
1000100000001000000010001	00000000100000000000000000100000000000000010000000	3

Codes Cycliques $C[50, 25]_2$:

La factorisation de $x^{50} - 1$ en facteurs irréductibles nous donne un seul choix du polynôme générateur $g(x)$ de degré 25 :

$$x^{50} - 1 = (1 + x)^2(1 + x + x^2 + x^3 + x^4)^2(1 + x^5 + x^{10} + x^{15} + x^{20})^2.$$

On note dans la table (25) le mot de code et leur poids correspondant :

Table 25

$g(x)$	mot de code (a)	wt(a)
100000000000000000000000001	100000000000000000000000001000000000000000000000000	2

Tableau récapitulatif de la distance optimale des codes de paramètres $[n, \frac{n}{2}]$,

n pair et ≤ 50 :

n	2	4	6	8	10	12	14	16	18	20	22	24	26
d_C	2	2	2	2	2	4	4	2	2	2	2	4	2

n	28	30	32	34	36	38	40	42	44	46	48	50
d_C	4	6	2	6	4	2	6	8	2	8	4	2

3.3 ANNEXE

3.3.1 Programme de recherche de la distance minimaum d'un code cyclique sur GF(2)

Principe de l'algorithme : si le code possède un mot de poids w , alors il existe forcément un décalage cyclique de ce meme mot possédant $r = \text{floor}(w \cdot k/n)$ coordonnées non nulles sur la partie information. Il suffit donc de générer tous les mots ayant un poids d'information égal à r , et de vérifier si l'un de ces mots possède un poids total de w (auquel cas $d_{\min} = w$). La recherche se fait par ordre de poids w croissant .

Pour compiler sous Linux Unix :

```
gcc-O2 dmin_f2-o dmin_f2
*/
#include <stdio.h >
#include <stdio.h >
/*Renseigner ici les paramètres du code */
#define N      42    /* longueur du code */
#define DEG_G  21    /* degré du gérateur */
#define W0      1    /* borne inf dmin */
/* Tableau des coefficients du polynome gérateur dans l'ordre g_0, g_1, ....., g_{N-K} */
int G[DEG_G+1]={1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1};
#define K      (N-DEG_G)
int pos_nz[K];
int val_nz[K];
int message[K];
int paritie [N-K];
/* Génère toutes les combinaisons de k éléments parmi n dans l'ordre lexicographique */
int next_comb (int n, int k, int *c)
{
```

```
int i, j;
j=k-1;
while (j >=0 && c[j] == (n-k+j) ) j--;
if (j == -1) return 0 ;
c[j] ++;
for (i = j+1; i < k; i ++ )
c[i] = c[i-1] + 1 ;
return 1;
}
/* Réalise l'encodage cyclique d'un message */
void encode ( int *data, int *parity )
{
int i, t ;
for (i = 0 ; i < N-K; i ++ )
parity [i] = 0;
for (t= 0; t < K; t ++ )
{
int feedback = (parity[0] + data[t] % 2;
for (i = 0; i < N-K-1; i ++ )
{
int j =(feedback * (2-G[N-K-1-i])) % 2;
parity [i] =(parity[i+1]+ j) % 2;
}
parity [N-K-1]= (feedback * (2-G[0] )) % 2;
}
}
/ * Programme principal */
int main ( )
{
int i,w;
```

```
/* Affiche quelques infos sur le code */
printf (" Code cyclique (%d, %d) sur GF(2) \ n ", N, K);
printf("Polynome g n rateur G = ");
for (i = 0; i <= DEG_G; i++)
printf ("%d", G[i]);
printf("\n");
printf("Borne inf rieure sur dmin :%d\n\n", W0 );
/* Initialise le message d'info   z ro */
for (i = 0; i < K; i++)
message [i] = 0
for (i = 0; i <N- K; i++)
parite [i] = 0 ;
/* Boucle infinie sur les poids w croissants */
for (w = W0; w++)
{
int weight, r ;
/* Calcul du poids r sur la partie info */
r = (w*K)/N ;
printf (" Recherche de mots de poids w=%d (r=%d)\n", w, r);
fflush (stdout);
if (r! = 0)
{
/* Boucle sur les combinaisons de r  l ments non nuls parmi K*/
for (i = 0; i < r ; i++)
pos_nz[i]=i;
do
{
/* Boucle sur les diff rents messages possibles */
for (i = 0 ; i < r; i++)
val_nz[i] = 1;
```



```
do {
/* Encodage d'un message */
for (i = 0; i < r ; i++)
message [ pos_nz[i] ]=val_nz[i];
encode(message, parite );

/* Calcul du poids du mot */
weight = r ;
for (i = 0; i < N-K; i++)
if (parite [i] != 0)
weight ++;
/* C'est terminé si on obtient le poids recherché . */
if (weight == w)
{
printf ("Un mot de poids w =%d a été trouvé\n", w );
/* Affiche le mot dans l'ordre c_{n-1},.....,c_0 */
printf ("Mot : ");
for (i = 0 ; i < K; i++)
printf("%d", message [i] );
for (i =0; i < N-K; i++)
printf("%d", (2-parite[i] % 2 ) );
printf ("\n");
return 0 ;
}
/* Génère le message suivant (s'il en reste ) */
val_nz[0] = (val_nz[0] + 1) % 2;
i = 0;
while (i < r && val_nz[i] == 0)
{
val_nz[i] = 1;
if (i < r-1 )
```

```
val_nz[i+1]=(val_nz[i+1]+1) % 2;
i ++;
}
} while (i != r);
/* Sinon, on continue en remettant le message à zéro */
for ( i =0; i < r; i++)
message [pos_nz[i]] = 0;
} while (next_comb(K, r, pos_nz) !=0);
}
/* Augmente le poids w si nécessaire */
printf("Aucun mot de poids w=%d n'a été trouvé \n\n", w);
}
return 0;
}
```

3.4 Conclusion

Le travail de ce mémoire entre, en générale dans le cadre de la classification des codes linéaires optimaux sur un corps fini F_q , En particulier nous avons étudié les codes cycliques optimaux de rendement $\frac{1}{2}$ sur F_2 . Dans ce contexte nous avons pré calculer la distance minimale optimale des codes de paramètres $[n, n/2]$ sur le corps fini F_2 jusqu'à la longueur 50 du code moyennant L'algorithme de Chen [José Felipe VOLOCH].

3.5 BIBLIOGRAPHIE

- [1] **Cherif Mihoubi**, Classification des Codes linéaires tertiaires optimaux $[n, n/2]$, These présenté pour l'obtention du diplôme de Doctorat, Université Hadj Lakhdar Batna, 2012 .
- [2] **Hans Bherer**, Théorie Algébrique du Codage, Mémoire présenté à la Faculté des études supérieures de l'université Laval .
- [3] **Christine Bachoc**, Codes et cryptologie(Cours et TD) .
- [4] **Robert Rolland**, Introduction à l'étude des Corps fini .
- [5] **Heboub Lakhdar**, Etude de Techniques de décodage des codes linéaires, Mémoire présenté pour l'obtention du diplôme de Magistère, Université de M'sila 2009/2010.
- [6] **A.A.Pantchichkine**, Magistère de Mathématique (L'ENS de Lyon) 2005 .
- [7] **A.Bonnecaze**, Introduction à l'algèbre pour les Codes Cycliques .
- [8] **Claude Carlet**, Cours de Codes Correcteurs d'Erreurs (et fonctions booléennes), D.E.A de mathématiques et d'informatique de Bamako Année 2007.
- [9] **Gilles Zémor**, Master CSI, Arithmétique 1: corps finis et applications .
- [10] **Nicolas Bruyère**, Eléments de théorie des corps finis. Application : les codes correcteurs, Université de Rouen Agrégation de mathématiques 2005-2006 .
- [11] **Nelly Burrin**, Finite Fields And Error-Correcting Codes, Master en systèmes de communication, Prge E'cole Polytechnique fédérale de Lansanne .
- [12] Théorie et Codage de l'information, Les codes de Hamming et les codes cycliques (Cours et TD sur Internet).
- [13] **Cherif Mihoubi · Patrick Solé**, Optimal and isodual ternary cyclic codes of rate $1/2$.
- [14] **José Felipe VOLOCH**, "Computing the minimal distance of cyclique codes ".
- [15] **Sara Djail**, Etude de l'équivalence de deux codes correcteurs d'erreurs par isometrie, Mémoire présenté pour l'obtention du diplôme de Master, Université de M'sila.