UNIVERSITE NICE SOPHIA ANTIPOLIS
ECOLE DOCTORALE STIC

THÈSE DE DOCTORAT
EN
INFORMATIQUE

Présentée par

OLFA YEMEN

SUJET

Application des codes cycliques tordus

Soutenue le 19 janvier 2013 devant le jury composé des

Professeurs

Mongi Naimi	Président
Hédi Amara	Rapporteur
André Leroy	Rapporteur
Patrick Solé	Directeur de Thèse
Sami Omar	Directeur de Thèse
Bruno Martin	Examinateur

Remerciements

Je tiens tout d'abord à adresser mes plus sincères remerciements à mon directeur de thèse Patrick Solé, Directeur De Recherche au CNRS, qui, avec beaucoup de disponibilité et de gentillesse a dirigé mes travaux de recherche. Je lui exprime toute ma reconnaissance pour l'accueil chaleureux qu'il m'a accordé durant mes séjours à Nice et Paris. Ce fût pour moi l'occasion d'apprécier non seulement ses qualités scientifiques mais aussi ses non moins importantes qualités humaines.

J'exprime mes vifs remerciements à mon co-directeur de thèse, le Professeur Sami Omar, pour son attention et le soutien qu'il a su m'accorder pendant ces dernières années. Ses conseils et ses remarques constructives m'ont été d'une aide précieuse pour mener à bien cette thèse.

Je tiens à exprimer toute ma reconnaissance au Professeur Mongi Naimi , professeur de mes débuts, qui n'a jamais cessé de m'encourager. Son aide et ses conseils judicieux m'ont été d'un grand apport. Je suis honorée qu'il soit présent dans ce jury et qu'il ait accepté de le présider.

Je voudrais également remercier chaleureusement le professeur Hédi Amara qui a eu la patience et la gentillesse de lire ma thèse en vue d'en faire le rapport. Je suis ravie qu'il participe à mon jury.

Mon premier contact avec le Professeur André Leroy a eu lieu il y a quelques années à travers un mail que je lui ai envoyé pour avoir une clarification sur l'un de ses articles. Dés lors, je n'ai pu qu'apprécier la gentillesse et la disponibilité avec laquelle il a répondu à une doctorante qu'il ne connait même pas. Je lui adresse mes plus sincères remerciements pour avoir accepter d'être rapporteur de ma thèse.

Le professeur Bruno Martin est membre du laboratoire I3S de l'Université Sophia Antipolis ou j'ai fait les débuts de ma thèse. Je profite de cette occasion pour lui exprimer toute ma reconnaissance pour son accueil chaleureux et son aide précieuse durant mes séjours à Nice. Je le remercie vivement d'avoir eu la gentillesse d'accepter de faire partie du jury.

Un témoignage de ma profonde reconnaissance s'adresse à Adnen Sboui qui m'a présenté à Patrick Solé et m'a aidé à démarrer cette thèse. Il est difficile de trouver des qualificatifs assez forts pour souligner sa gentillesse, son humilité et sa grande volonté d'aide.

Je tiens aussi à remercier mes amis Lin Sok et Romar Dela Cruz que j'ai rencontré à Nice et à Paris pour leur extrême gentillesse, leur soutien et pour les nombreuses discussions scientifiques qu'on a échangé.

Un grand merci à tous les membres du laboratoire I3S de Sophia Antipolis chercheurs et administrateurs pour leur extrême gentillesse et disponibilité.

Je tiens à exprimer toute ma reconnaissance à mes amies Feriel Sassi , Nadia Ben Hamadi et Malek Zribi pour leur aide dans les préparatifs des dernières minutes. Merci pour leur amitié dévouée, leur complicité et les moments partagés.

Enfin et dans la crainte d'oublier quelqu'un, je remercie toutes les personnes qui ont contribué de près ou de loin à la réalisation de cette thèse et en particulier beaucoup de mes collègues de l'Institut Préparatoire Aux Etudes d'Ingénieurs El Manar.

Dédicace

Je dédie cette thèse

A Mes très chers parents.

Pour tous leurs sacrifices, amour et soutien inépuisable qui m'a guidé jusqu'à ce jour.
Et puis, pour cette certitude inébranlable que j'ai au fond de moi que quoi qu'il arrive, ils seront toujours là, autant dans les bons moments que dans les plus pénibles, un pilier fort et réconfortant sur lequel je sais que je peux compter.
Quoi que je dise, je ne pourrais leur exprimer l'immensité de mon amour et de ma gratitude pour leur présence qui a illuminé mon chemin.
Que dieu leur accorde longue vie, santé et bonheur.

A Mohamed.

Pour ce jour où, face à tous mes doutes et mes angoisses, il m'a dit :
"T'en fais pas. Ensemble, on fera les sacrifices nécessaires et tu l'auras ta thèse "
Son aide, son écoute et son grand soutien ont été pour moi une source inestimable où j'ai puisé courage et confiance.
Qu'il trouve ici toute ma reconnaissance pour son amour et sa grande générosité ainsi que l'expression de mon attachement et de ma tendresse.

A Roua et Maram.

Mes deux princesses que j'adore et la lumière de ma vie.
Une demande de pardon, pour tout ce temps qui aurait du être le leur, pour tout ce stress qu'elles ont du subir et pour cette question inquiète qui flottait de temps à autre dans leurs innocents yeux alors que des codes correcteurs se disputaient dans ma tête
" Maman,.... pourquoi t'es fâchée.. ? ! ! ! "
Mon cœur leur souhaite une longue vie illuminée par les étoiles du bonheur.

A tous ceux que j'aime

Et spécialement . . . . . .

A Yosra et Wael
Une profonde reconnaissance pour leur affection et leur générosité.

A tous mes ami(e)s.

Et, aux enseignants qui ont marqué mes études

L'amour, le soutien et les encouragements de tous ceux que j'ai cité étaient les bougies qui ont éclairé mon parcours de recherche.

Merci à tous et à toutes.
olfa

Table des matières

1 Introduction générale **9**

2 Généralités **12**
2.1 Introduction 12
2.1.1 Définitions 12
2.1.2 Dualité 13
2.1.3 Codes cycliques et quasi-cycliques 15
2.1.4 Anneau des polynômes d'Ore et codes θ-cycliques 16
2.1.5 Codes quantiques 19

I Des codes θ-cycliques aux codes quantiques sur $\mathbb{F}_4$ **23**

3 Factorisation des polynômes tordus et construction des codes θ-cycliques **24**
3.1 Introduction 24
3.2 Définitions et notations 24
3.3 Notions sur le ppcm dans $\mathbb{F}_q[X;\theta]$ 25
3.4 Factorisation dans $\mathbb{F}_4[X;\theta]$ 26
3.5 Construction des codes θ-cycliques 29

4 Codes θ-cycliques auto-orthogonaux et informatique quantique **33**
4.1 Introduction 33
4.2 Notions sur les $p-$polynômes 34
4.3 Caractérisation d'un code θ-cyclique auto-orthogonal 35
4.4 Code auto-orthogonal comme sous code d'un code auto dual 40
4.5 Construction d'un code quantique sur $\mathbb{F}_4$ 40
4.5.1 Résultats numériques 41

II Nouvelles constructions des codes $\theta-$cycliques et quasi-θ-cycliques

Applications à la construction des codes quantiques asymétriques 43

5 Des codes θ-cycliques aux codes quantiques asymétriques 44
5.1 Introduction . . . 44
5.2 Généralités . . . 45
5.3 L'application S sur les codes de $\mathbb{F}_4$ . . . 46
5.4 Application aux codes θ-cycliques . . . 47
5.5 Analyse du poids de distribution . . . 52
5.6 Codes quantiques asymétriques et application S . . . 53
5.7 Les Constructions . . . 55
5.7.1 Construction à partir du meilleur code linéaire connu . . . 55
5.7.2 Construction à partir des codes de Reed Solomon concaténés . . . 56

6 Nouvelle construction des codes quasi θ-cycliques sur $\mathbb{F}_4$ 60
6.1 Introduction . . . 60
6.2 codes quasi θ-cycliques . . . 60
6.3 L'application S sur les codes quasi θ-cycliques de $\mathbb{F}_4$ . . . 61
6.4 Conclusions . . . 64

7 Des codes θ-cycliques sur $\mathbb{F}_2 + v\mathbb{F}_2$ aux codes quasi-cycliques binaires 65
7.1 Introduction . . . 65
7.2 L'application Z . . . 66
7.3 Codes $\theta-$cycliques et application Z . . . 68
7.4 Construction des codes $\theta-$cycliques sur $\mathbb{F}_2 + v\mathbb{F}_2$ . . . 71
7.5 Construction de code $\theta-$cyclique de longueur paire . . . 71
7.6 Construction de codes θ-cycliques auto-duaux euclidiens sur $\mathbb{F}_2 + v\mathbb{F}_2$ . . . 72
7.7 Conclusion et problèmes ouverts . . . 72

III Polynômes tordus pour la construction des codes sur $M_2(\mathbb{F}_2)$ 73

8 Codes quasi-cycliques binaires comme codes cycliques sur $M_2(\mathbb{F}_2)$ 74
8.1 Introduction . . . 74
8.2 Préliminaires . . . 75
8.3 Quelques propriétés sur les Polynômes irréductibles de $\mathbb{F}_p[X]$ . . . 76
8.4 Factorisation dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$ . . . 81
8.5 Factorisation dans $M_2(\mathbb{F}_2[X])$ . . . 85
8.6 Construction des codes $\Omega(P)$ . . . 88
8.7 Construction des codes $\Omega(P)$ auto duaux . . . 90

9 Codes cycliques sur $M_2(\mathbb{F}_2)$ **93**
9.1 Introduction 93
9.2 Notations et définitions 94
9.3 Construction d'un code cyclique 94
9.4 L'application de Bachoc 96
9.4.1 Propriétés métriques 96
9.4.2 Propriétés de dualité 96
9.4.3 Propriétés cycliques 98
9.5 Codes cycliques auto-duaux 98
9.6 Codes cycliques auto-duaux de longueur impaire $n \leq 31$. 99
9.7 Conclusion et problèmes ouverts 103

Bibliographie **114**

Chapitre 1

Introduction générale

Cette thèse est axée sur trois thèmes séparés mais complémentaires : les polynômes d'Ore, les codes θ-cycliques et l'informatique quantique.
Les anneaux d'Ore ont vu le jour dans les années 30 par Oystein Ore [51]. Il s'agit d'un anneau de polynômes à coefficients dans un corps K muni d'un automorphisme non trivial θ et où la multiplication n'est pas commutative. En effet, l'anneau d'Ore est noté et défini par

$$K[X;\theta] = \{\sum_{i=0}^{n} a_i X^i \mid a_i \in K \ et \ n \in \mathbb{N}\},$$

où l'addition est définie de la manière usuelle et la multiplication suivant la règle

$$Xa = \theta(a)X.$$

On étend cette règle par associativité et distributivité par rapport à l'addition. L'anneau $K[X;\theta]$ est donc un anneau non commutatif euclidien à droite et à gauche et dont les idéaux à droite -respectivement à gauche - sont principaux.
Durant la dernière décennie, ces polynômes ont fait l'objet de travaux de plusieurs chercheurs dont nous citons en particulier André Leroy et ses co-auteurs [16],[40] et [41]. Ainsi, nous avons à ce jour une étude complète sur ces polynômes autant du côté algébrique qu'arithmétique tel que propriétés de racines, fonctions symétriques et factorisations...etc. Dans ce cadre, il est important de signaler qu'un des atouts majeurs des polynômes tordus est le fait qu'ils n'admettent pas de factorisation unique ce qui constitue une motivation extrême pour leur utilisation dans le domaine des codes correcteurs d'erreurs.
En effet, les anneaux de polynômes ont été l'un des outils privilégiés pour construire et étudier des familles des codes correcteurs et spécialement dans la classe des codes linéaires. Ainsi, l'utilisation des polynômes tordus dans ce domaine va permettre de créer un nombre important de codes linéaires pour une longueur donnée et avec de très bons paramètres de surplus. Cependant, ce n'est qu'au début des années 2000 et spécialement l'année 2007 que les polynômes d'Ore ont été utilisés dans la théorie des codes via l'apparition dans les travaux de Felix Ulmer et ses co-auteurs des codes θ-cycliques [12],[8], [9] et [10]. Les auteurs définissent un code θ-cyclique C comme étant un sous espace vectoriel

de $\mathbb{F}_q^n$ possédant la propriété suivante

$$(a_0, a_1, ..., a_{n-1}) \in C \Rightarrow (\theta(a_{n-1}), \theta(a_0), \theta(a_1), ..., \theta(a_{n-2})) \in C.$$

Ainsi, pour un polynôme P de $K[X;\theta]$ de degré multiple de l'ordre de θ et pour tout diviseur à droite Q de P l'idéal à gauche $(Q)_{/(P)}$ est la représentation polynomiale du code θ-cyclique C.
D'un autre côté, la richesse des codes cycliques tordus en nombre et paramètres incite à leur utilisation dans l'informatique quantique et spécialement dans la construction des codes quantiques.

En effet, durant les années quatre-vingts l'idée de construire un ordinateur quantique a vu le jour afin de résoudre le problème de la complexité à simuler des systèmes quantiques par un ordinateur classique [32]. Cependant, la théorie classique des codes correcteurs ne peut pas être mise en œuvre dans le cadre de l'informatique quantique à cause de détails techniques. Il était donc nécessaire de développer une théorie des codes quantiques afin de corriger les états quantiques lors de leur transmission sur un canal quantique. Les premiers travaux sur les codes quantiques ont commencé avec Shor, Sloane, Rains et Calderbank dans [13]. Ce fut spécialement la découverte de la construction CSS puis des codes stabilisateurs. Ces derniers [13] sont construits à base de codes sur le corps fini d'ordre 4 et utilisent des codes classiques comme les codes BCH par exemple. Les résultats récents dans ce domaine sont reportés par la table des records de Markus Grassel à la section QECC [61].
Très récemment, une nouvelle tendance est apparue dans le domaine des codes correcteurs quantiques [33], [37] et [23]. En effet, des expériences physiques récentes ont montré qu'il existe une nette différence entre les probabilités des opérateurs d'erreurs quantiques élémentaires. Ainsi, a vu le jour l'idée de construire des nouveaux codes quantiques qui prennent en compte cette asymétrie de transmission dans le canal quantique : nous citons les codes quantiques asymétriques [33]. En effet, les codes quantiques asymétriques ont la capacité de corriger beaucoup plus d'erreurs du type appelé phase que d'erreurs du type appelé bit ou phase-bit. Ainsi, afin de signaler le caractère asymétrique de ces codes, les nouvelles constructions leur attribuent deux paramètres de distances. On dit donc qu'un code asymétrique a une X distance d_x, respectivement Z distance d_z s'il peut détecter toutes les erreurs du type X ou bit de poids supérieur ou égale à $d_x - 1$, respectivement toutes les erreurs du type Z ou phase de poids supérieur ou égale à $d_z - 1$.
Dans cette thèse, nous avons travaillé sur les différents liens qui peuvent exister entre les trois thèmes cités ci-dessus à savoir : les polynômes tordus, les codes θ-cycliques et l'informatique quantique.

Cette thèse a fait l'objet de publications [chap 5 [21], chap 8 [58]] et de résultats soumis [chap 9].
Dans la première partie, nous avons étudié les polynômes d'Ore et nous avons élaboré un algorithme de factorisation afin de présenter via des calculs numériques sur Magma [49] des nouvelles constructions des codes $\theta-$cycliques auto-orthogonaux. En effet, bien que les codes auto-orthogonaux soient le point de départ de la construction des codes quantiques via la construction CSS, peu de résultats existent dans la littérature concer-

nant les codes $\theta-$cycliques auto-orthogonaux. Ainsi, les résultats de cette première partie vont nous permettre de retrouver les codes quantiques de meilleures distances connues dans les tables de Marcus Grassel [61]. D'un autre côté, une des difficultés rencontrées dans la construction des codes $\theta-$cycliques fût le problème de la factorisation des polynômes tordus. Ainsi et pour contourner cette difficulté surtout au niveau numérique nous avons introduit dans la deuxième partie de cette thèse diverses applications permettant le passage des codes θ-cycliques ou quasi θ-cycliques sur différents anneaux à des codes cycliques et quasi-cycliques et nous avons utilisé ce passage dans la construction des codes quantiques asymétriques. L'application S par exemple utilisée dans le chapitre 5 et appliquée sur des codes θ-cycliques de $\mathbb{F}_4^n$ nous a permis d'obtenir pour la première fois et d'une façon directe des constructions de codes quantiques asymétriques pour des valeurs particulières tels que $[[18, 2, 12/2]]$, $[[30, 3, 20/2]]$, $[[32, 2, 22/2]]$...etc. De plus, nous avons présenté dans les chapitres 6 et 7 des nouvelles isométries qui nous ont permis de faciliter la construction des codes quasi $\theta-$cycliques sur $\mathbb{F}_4$ (chapitre 6) et des codes $\theta-$cycliques sur le seul exemple d'anneau d'ordre 4 possédant un automorphisme non trivial et sur le quel on peut donc construire des codes $\theta-$cycliques à savoir l'anneau $R = \mathbb{F}_2 + v\mathbb{F}_2$ (chapitre 7).
Dans la troisième partie enfin, nous nous sommes intéressés à la construction des codes cycliques et quasi-cycliques sur les anneaux non commutatifs. L'exemple le plus concret de ces anneaux est l'anneau $A = M_2(\mathbb{F}_2)$ qui fût d'ailleurs le premier anneau non commutatif à avoir été utilisé comme alphabet pour des codes en blocs. La première motivation de l'utilisation de cet anneau est la construction des réseaux modulaires [4] et plus récemment la construction des codes espaces temps à partir de la concaténation des codes d'or [5]. De plus, des travaux récents de J.C.Belfiore, F.Oggier et P.Solé [5] ont révélé une relation entre l'anneau des matrices carrées d'ordre n et à coefficients dans $\mathbb{F}_2$ et l'anneau des polynômes tordus $\mathbb{F}_{2^n}[X;\theta]_{/X^n-1}$. Ainsi, on a l'isomorphisme d'anneaux et d'espaces vectoriels suivant

$$M_n(\mathbb{F}_2) \simeq \mathbb{F}_{2^n}[Y,\theta]_{/Y^n+1}.$$

Nous avons donc utilisé les polynômes tordus comme un outil de travail pour étudier la construction des codes cycliques sur l'anneau de matrices $M_2(\mathbb{F}_2)$. Plus précisément, nous avons eu recours à l'isomorphisme cité ci-dessus pour présenter une étude non exhaustive des polynômes de $M_2(\mathbb{F}_2)[X]$. Cette étude va nous permettre d'améliorer la construction des codes quasi-cycliques sur $M_2(\mathbb{F}_2)$ donnée dans [15] et de donner des nouvelles constructions de codes quasi-cycliques auto-duaux sur cet anneau.
Enfin, dans le dernier chapitre de cette thèse, nous avons attribué à l'anneau $M_2(\mathbb{F}_2)$ une structure d'anneau quotient sur l'anneau des polynômes tordus à coefficients dans le corps $\mathbb{F}_4$ via l'introduction d'une nouvelle matrice i et l'écriture de $M_2(\mathbb{F}_2)$ sous la forme $M_2(\mathbb{F}_2) = \mathbb{F}_4 + i\mathbb{F}_4$. Cette étude nous a permis de présenter une caractérisation des codes cycliques sur $M_2(\mathbb{F}_2)$ via leurs générateurs et une caractérisation arithmétique de ces générateurs afin qu'ils soient auto-duaux.

Chapitre 2

Généralités

2.1 Introduction

L'objet de ce chapitre est de rappeler quelques notions de bases sur les codes correcteurs d'erreurs et de présenter une étude non exhaustive sur certains outils utilisés tout au long de cette thèse tels que les codes cycliques et quasi-cycliques, les anneaux d'Ore et les codes quantiques.

2.1.1 Définitions

Soit $\mathbb{F}$ un anneau commutatif de cardinal fini q. Un code C de longueur n sur $\mathbb{F}$ est un sous ensemble de $\mathbb{F}^n$. On appelle mot de code tout élément du code C et on appelle taille N du code son cardinal.
Pour tout vecteur $x = (x_1, x_2, ..., x_n)$ de $\mathbb{F}^n$, on appelle poids de Hamming et on note $wt(x)$ le nombre de composantes x_i différentes de zéro. La distance minimale d'un code C est le nombre noté et défini par :

$$d(C) = \min\{wt(y - x) : x, y \in C \ \ et\ y \neq x\}.$$

Un code C est dit de paramètres (n, M, d) s'il est de longueur n, de taille M et de distance minimale d.
On considère maintenant $\mathbb{F}_q$ le corps fini de cardinal q, de caractéristique p et où $q = p^m$. On appelle code linéaire -respectivement code additif - de longueur n sur $\mathbb{F}_q$ tout $\mathbb{F}_q$- sous espace vectoriel de $\mathbb{F}_q^n$, respectivement tout $\mathbb{F}_p$- sous espace vectoriel de $\mathbb{F}_q^n$. On appelle dimension d'un code linéaire C sa dimension k en tant que $\mathbb{F}_q$-sous espace vectoriel , sa taille est alors égale à q^k. Dans le cas d'un code linéaire la distance minimale d'un code C est définie par

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

Un code linéaire de longueur n, de dimension k et de distance minimale d sur $\mathbb{F}_q$ est dit de paramètres $[n, k, d]_q$. La matrice génératrice d'un code linéaire C est toute matrice à

k lignes et n colonnes dont les lignes forment une base de C.
Une matrice génératrice G d'un code C est dite sous sa forme systématique si et seulement si elle s'écrit sous la forme :

$$G = [I_k|A],$$

où I_k est la matrice Identité d'ordre k. Dans ce cas la matrice

$$H = [-{}^tA|I_{n-k}]$$

est appelée une matrice de contrôle du code C. De plus, on a

$$C = \{x \in \mathbb{F}_q^n \mid Hx^T = 0\}.$$

Exemple 2.1.1 *La matrice*

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

est une matrice génératrice du code

$$\{(00000); (10000); (11010); (11101); (01010); (01101); (00111); (10111)\}.$$

Proposition 2.1.2 *Un code linéaire de paramètres $[n, k, d]$ sur $\mathbb{F}_q$ vérifie l'inégalité*

$$d \leq n - k + 1.$$

On dit que deux codes linéaires de même longueur sont équivalents si et seulement si l'un s'obtient à partir de l'autre par une permutation des coordonnées c'est à dire qu'il est l'image de l'autre par une permutation de S_n. On rappelle que deux codes équivalents ont les mêmes propriétés.

2.1.2 Dualité

Suivant le cardinal d'un corps $\mathbb{F}_q$, on peut définir plusieurs produits scalaires.

Définition 2.1.1 *On appelle produit scalaire euclidien sur $\mathbb{F}_q^n$ la forme bilinéaire symétrique qui à tout (x, y) de $(\mathbb{F}_q^n)^2$ associe l'élément*

$$< x, y >_e = \sum_{i=0}^{n} x_i.y_i.$$

Si l'entier q est une puissance paire d'un nombre premier p, on peut alors définir l'automorphisme involutif suivant :

$$\forall x \in \mathbb{F}_q \quad x \mapsto x^{\sqrt{q}}.$$

Dans ce cas on peut définir dans $\mathbb{F}_q$ le produit scalaire hermitien et le produit scalaire trace comme suit.

Définition 2.1.2 *On appelle produit scalaire hermitien sur $\mathbb{F}_q^n$ la forme qui à tout (x, y) de $(\mathbb{F}_q^n)^2$ associe l'élément :*

$$< x, y >_h = \sum_{i=0}^{n} x_i . y_i^{\sqrt{q}},$$

et

Définition 2.1.3 *Le produit scalaire trace sur $\mathbb{F}_q^n$ est la forme qui à tout (x, y) de $(\mathbb{F}_q^n)^2$ associe l'élément :*

$$< x, y >_{tr} = \sum_{i=0}^{n} (x_i . y_i^{\sqrt{q}} + x_i^{\sqrt{q}} y_i).$$

Dans toute la suite, la notation $< ., . >$ voudra dire un produit scalaire euclidien, hermitien ou trace dans $\mathbb{F}_q^n$. Le dual $C^\perp$ d'un code C de longueur n sur $\mathbb{F}_q$ est l'ensemble

$$C^\perp = \{x \in \mathbb{F}_q^n : \forall y \in C, < x, y >= 0\}.$$

Remarque 2.1.1 *Le dual d'un code quelconque C est un code linéaire. Si de plus C est linéaire, on a*

$$\dim C + \dim C^\perp = n.$$

La matrice génératrice du code $C^\perp$ est la matrice de contrôle du code C. Ainsi, si C est un code de paramètres $[n, k, d]$, $C^\perp$ sera de paramètres $[n, n - k, d]$.

Un code C est dit auto-orthogonal (respectivement auto-dual) si et seulement si $C \subset C^\perp$ (rspectivement $C = C^\perp$). Un code auto-dual est toujours de longueur paire et de dimension $k = \frac{n}{2}$. La distribution de poids d'un code C est la suite $A_0, A_1, ...A_n$ où chaque A_i désigne le nombre de mots dans C de poids i. Le polynôme énumérateur $W_C(X, Y)$ de C est le polynôme de degré n défini par :

Définition 2.1.4 *Le polynôme énumérateur $W_C(X, Y)$ d'un $(n, M = |C|, d)$-code C est le polynôme*

$$W_C(X, Y) = \sum_{i=0}^{n} A_i X^{n-i} Y^i, \tag{2.1.1}$$

où A_i est le nombre de mots de code de poids i dans C.

Le polynôme énumérateur d'un code est très important pour l'étude de certaines de ses propriétés. Ainsi, Le polynôme énumérateur du dual hermitien $C^{\perp_H}$ d'un $[n, k, d]$-code C est lié au polynôme énumérateur du code C par l'identité de MacWilliams :

$$W_{C^{\perp_H}}(X, Y) = \frac{1}{|C|} W_C(X + (q - 1)Y, X - Y). \tag{2.1.2}$$

2.1.3 Codes cycliques et quasi-cycliques

On appelle code cyclique de longueur n sur $\mathbb{F}_q$ tout code linéaire C stable par décalage circulaire, c'est-à-dire :

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

Dans toute la suite, on appelle " shift" l'application T définie dans $\mathbb{F}_q^n$ par

$$T((c_0, c_1, ..., c_{n-1})) = (c_{n-1}, c_0, c_1, ..., c_{n-2}).$$

Ainsi, un code cyclique est un code stable par le shift T. Il est commode d'identifier les vecteurs de $\mathbb{F}_q^n$ avec l'ensemble des polynômes de $\mathbb{F}_q[X]$ de degré inférieur ou égal à $n-1$ par la correspondance

$$(c_0, c_1, ..., c_{n-1}) \mapsto c(X) = c_0 + c_1 X + ... + c_{n-1} X^{n-1}.$$

L'ensemble des polynômes de degré inférieur ou égale à $n-1$ est lui-même un système de représentants distincts de l'anneau quotient $A = \mathbb{F}_q[X]/(X^n - 1)$. L'intérêt de cette identification est que le décalage circulaire est exactement la multiplication par X dans l'anneau A. Ainsi, on a le résultat suivant :

Proposition 2.1.3 *Les codes cycliques de $\mathbb{F}_q^n$ sont les idéaux de l'anneau $A = \mathbb{F}_q[X]/(X^n - 1)$.*

On appelle polynôme générateur du code cyclique C, le polynôme unitaire non nul de plus petit degré représentant un élément de C.

Théorème 2.1.1 *Soit C un code cyclique de longueur n sur $\mathbb{F}_q$ et soit $g(X)$ son polynôme générateur. Alors*
1. Le code C est l'idéal $< g(X) >$ de l'anneau A.
2. Le polynôme de $g(X)$ divise $X^n - 1$ dans $F_q[X]$.
3. $dimC$ est égale à $n - deg(g(X))$ et une base de C en tant que sous espace vectoriel de $\mathbb{F}_q^n$ est $(g(X), Xg(X), ..., X^{n-deg(g)-1}.g(X))$.

Soit C un code cyclique de dimension k, de longueur n et de polynôme générateur $g(X) = \sum_{i=0}^{n-k} g_i X^i$. La matrice génératrice de C est donnée par

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & ... & g_{n-k} & 0 & ... & 0 \\ 0 & g_0 & g_1 & .. & g_{n-k-1} & g_{n-k} & ... & ... \\ .. & .. & & .. & & .. & .. & \\ 0 & ... & .. & 0 & g_0 & .. & .. & g_{n-k} \end{pmatrix}.$$

Si

$$X^n - 1 = g(X).h(X)$$

où $h(X) = h_0 + h_1 X + ... + h^k X^k$, le polynôme générateur du code dual $C^\perp$ de C est alors le polynôme unitaire réciproque de $h(X)$ donné par

$$g^\perp(X) = h_0^{-1}(h_0 X^k + h_1 X^{k-1} + ... + h_{k-1} X + h_k) = h_0^{-1} X^k h(\frac{1}{X}).$$

Exemple 2.1.4 *Les codes BCH (Bose-Chaudhuri-Hocquenghem) sont des codes cycliques particuliers. On appelle code BCH binaire de longueur $n = 2^r - 1$ et de distance prescrite δ (δ entier tel que $0 < \delta \leq n$) le code cyclique de polynôme générateur g_Σ, où Σ est le plus petit sous-ensemble de $\mathbb{Z}_{/n\mathbb{Z}}$ contenant $1, 2, ..., \delta - 1$ et stable par multiplication par deux. Autrement dit, un polynôme $c(X) = \sum_{i=0}^{n-1} c_i X^i$ de $\mathbb{F}_2[X]$ appartient à ce code si et seulement si*

$$c(\alpha) = c(\alpha^2) = ...c(\alpha^{\delta-1}) = 0,$$

où α est une racine primitive n-ième de l'unité.

Définition 2.1.5 *Soit n un entier positif non nul et l un diviseur de n différent de n. Un $[n, k, d]_q$- code linéaire C est dit quasi-cyclique d' index l ou l-quasi-cyclique si et seulement si il est stable par la puissance l du shift T, c'est à dire :*
pour tout $v = (v_0, v_1, \ldots, v_{n-1}) \in C$, on a

$$v' = (v_{n-l}, v_{n-l+1}, \ldots, v_{n-1}, v_0, v_1, \ldots, v_{n-l-1}) \in C.$$

En particulier, un 1-quasi-cyclique code est un code cyclique. Comme pour les codes linéaires, on définit les codes additifs cycliques et les codes additifs quasi-cycliques d'une façon similaire en remplaçant le mot linéaire par additif.

2.1.4 Anneau des polynômes d'Ore et codes θ-cycliques

L'anneau des polynômes tordus a été introduit par Oystein Ore en 1933 [51]. Considérons un automorphisme θ de $\mathbb{F}_q$. L'anneau des polynômes tordus $\mathbb{F}_q[X;\theta]$ est l'ensemble des polynômes $\sum_{i=0}^{n} a_i X^i$ où l'addition est définie d'une façon naturelle et la multiplication suivant la règle

$$X.a = \theta(a)X.$$

On étend cette règle par associativité et distributivité par rapport à l'addition. Plus précisément la multiplication des polynômes dans cet anneau est définie par :

$$\sum_{i=0}^{n} a_i X^i . \sum_{j=0}^{p} b_j X^j = \sum_{i=0}^{n+p} a_i \theta^i(b_j) X^{i+j}.$$

L'anneau $\mathbb{F}_q[X;\theta]$ est un anneau non commutatif euclidien à droite et à gauche et dont les idéaux à droite (respectivement à gauche) sont principaux.

Pour $P_1, P_2 \in \mathbb{F}_q[X;\theta]$ tel que P_2 est non nul, il existe deux polynômes uniques $Q, R \in \mathbb{F}_q[X;\theta]$ tels que

$$P_1 = Q.P_2 + R \quad degR < degP_2.$$

Si $R = 0$, P_2 est appelé diviseur à droite de P_1 dans $\mathbb{F}_q[X;\theta]$. La notion de diviseur à gauche est définie d'une façon similaire. La notion de *pgcd* et *ppcm* existe dans $F_q[X;\theta]$ en utilisant l'algorithme d'Euclide.

Exemple 2.1.5 *Dans $\mathbb{F}_4 = \mathbb{F}_4(w)$ la division à droite et à gauche de $X + w$ par $wX + 1$ est donnée par :*

$$\begin{aligned} X + w &= w^2(wX + 1) + 1, \\ &= (wX + 1)w + 0. \end{aligned}$$

Dans toute la suite, on désigne par diviseur tout diviseur à droite d'un polynôme P de $\mathbb{F}_q[X;\theta]$. Pour plus d'informations concernant les polynômes tordus, le lecteur pourrait consulter [16], [40], [41] [51] et [45].

Définition 2.1.6 *On appelle code θ-cyclique de $\mathbb{F}_q^n$ tout code linéaire C de $\mathbb{F}_q^n$ vérifiant*

$$(a_0, a_1, ..., a_{n-1}) \in C \Rightarrow (\theta(a_{n-1}), \theta(a_0), \theta(a_1), ..., \theta(a_{n-2})) \in C.$$

Signalons que si l'automorphisme θ est égal à l'identité alors le code θ-cyclique C est un code cyclique.
Les travaux qui ont été faits sur les codes θ-cycliques ont été motivés par le fait que cette sous classe de codes est une importante classe de codes linéaires contenant les codes cycliques et qui contient aussi des codes avec de très bons paramètres. De plus, et comme pour les codes cycliques une correspondance a été établie entre les mots de codes d'un code θ-cyclique et les polynômes de $F_q[X;\theta]$ afin de présenter une structure algébrique de ces codes et de faciliter leur décodage.
D'une façon similaire à celle qui a été faite pour les codes cycliques, on peut identifier un mot de code d'un code C sur $\mathbb{F}_q^n$ à un polynôme de $F_q[X;\theta]_{/(X^n-1)}$ via la correspondance suivante

$$(c_0, c_1, ..., c_{n-1}) \mapsto c(X) = c_0 + c_1X + ... + c_{n-1}X^{n-1}.$$

On a le résultat suivant :

Lemme 2.1.6 *Soit $\mathbb{F}_q$ un corps fini à q éléments, θ un automorphisme d'ordre m de $\mathbb{F}_q$ et $\mathbb{F}_q^\theta$ le sous corps d'éléments de $\mathbb{F}_q$ stable par θ. Un polynôme P de $F_q[X;\theta]$ commute avec tous les éléments de $F_q[X;\theta]$ si et seulement si*

$$P = \sum_{i=0}^{n} c_i X^{im},$$

où les c_i sont des éléments de $\mathbb{F}_q^\theta$.

Notons que pour tout polynôme $P = \sum_{i=0}^{n} c_i X^{im} \in \mathbb{F}_q^\theta[X]$, l'idéal (P) est un idéal bilatère de $F_q[X;\theta]$. On utilisera dans la suite la notation $R = \mathbb{F}_q[X;\theta]$.

Proposition 2.1.7 *Soit F_q un corps fini à q éléments , θ un automorphisme de $\mathbb{F}_q$ et (P) un idéal bilatère de R. L'anneau $R_{/(P)}$ est un anneau principal où tout idéal à gauche est engendré par un diviseur à droite de P dans R.*

Proposition 2.1.8 *Soit $\mathbb{F}_q$ un corps fini à q éléments, θ un automorphisme de $\mathbb{F}_q$ et (P) un idéal bilatère de R. On appelle un θ-code ou code cyclique tordu tout idéal à gauche de $R_{/(P)}$ de la forme $(g)_{/(P)}$ et où g est un diviseur de P dans R.*

Signalons que si
1) $f = X^n - c$ où n est divisible par m l'ordre de θ et $c \in \mathbb{F}_q^{\theta}$ alors l'idéal $(g)_{/(P)}$ est appelé un θ-code consta-cyclique.
2) $f = X^n - 1$ où n est divisible par m l'ordre de θ alors l'idéal $(g)_{/(P)}$ est un code θ-cyclique et on a le résultat suivant :

Proposition 2.1.9 *Soit $F_q$un corps fini à q éléments, θ un automorphisme de $\mathbb{F}_q$ et n un entier divisible par l'ordre m de θ. C est un code θ-cyclique sur $\mathbb{F}_q^n$ si et seulement si la représentation polynomiale de C dans $R_{/(X^n-1)}$ est un idéal à gauche $(g)_{/(X^n-1)}$ de $R_{/(X^n-1)}$ où g est un diviseur de $X^n - 1$ dans R.*

La dimension d'un code θ-cyclique $C = (g)_{/(X^n-1)}$ est égale à $n - deg(g)$ et sa matrice génératrice est donnée par :

$$G = \begin{pmatrix} g_0 & g_1 & \dots & g_{r-1} & g_r & 0 & \dots & 0 \\ 0 & \theta(g_0) & \theta(g_1) & \dots & \theta(g_{r-1}) & \theta(g_r) & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \theta^{n-r-1}(g_0) & \dots & \theta^{n-r-1}(g_{r-1}) & \theta^{n-r-1}(g_r) \end{pmatrix}, \quad (2.1.3)$$

et qui ne dépend que du polynôme g et de l'entier n.

Proposition 2.1.10 *Soit $\mathbb{F}_q$ un corps fini à q éléments, θ un automorphisme sur F_q et C est un code θ-cyclique sur $\mathbb{F}_q^n$ engendré par le polynôme g vérifiant $X^n - 1 = gh$. Le dual euclidien du code C est un code θ-cyclique engendré par :*

$$g^{\perp} = h_{n-r} + \theta\left(h_{n-r-1}\right) X + ... + \theta^{n-r}\left(h_0\right) X^{n-r}. \quad (2.1.4)$$

Le dual hermitien du code C est un code θ-cyclique engendré par :

$$g^{\perp_H} = \theta(h_{n-r}) + \theta^2\left(h_{n-r-1}\right) X + ... + \theta^{n-r+1}\left(h_0\right) X^{n-r}. \quad (2.1.5)$$

Ces résultats ont permis à Ulmer et ses co-auteurs d'étudier et de donner des exemples de bons codes cycliques tordus sur $\mathbb{F}_4$. Cependant, la structure des codes θ-cycliques comme des idéaux de l'anneau quotient $R_{/(X^n-1)}$ liée aux propriétés arithmétiques de leurs générateurs impose des restrictions aux longueurs de ces codes. Citons l'exemple du corps $\mathbb{F}_4$ muni de l'automorphisme de Frobenius où l'on ne peut définir les codes θ-cycliques idéaux de $R_{/(X^n-1)}$ que ceux de longueur paire. Pour dépasser ces contraintes, Ulmer et Boucher ont introduit en 2009 les codes θ-cycliques comme des sous modules du module $R_{/(X^n-1)}$. Ils ont donné la définition suivante :

Définition 2.1.7 *Soit $\mathbb{F}_q$ un corps fini à q éléments, θ un automorphisme de $\mathbb{F}_q$ et n un entier naturel. Un code C est dit code θ-cyclique sur $\mathbb{F}_q^n$ si et seulement si la représentation polynomiale de C dans $R_{/(X^n-1)}$ est un R-sous module à gauche $(g)_{/(X^n-1)}$ de $R_{/(X^n-1)}$, où g est un diviseur à droite de X^n-1 dans R.*

Les résultats généraux concernant les matrices génératrices, les dimensions ou les générateurs des codes duaux sont les mêmes tant pour les codes θ-cycliques considérés comme des idéaux de l'anneau quotient $R_{/(X^n-1)}$ que ceux définis comme des sous modules du module $R_{/(X^n-1)}$. Toujours dans ce contexte, Somphong Jitman et Ling San ont montré en 2009 que pour $q=4$ et θ l'automorphisme de Frobenuis dans $\mathbb{F}_4$, il n'existe aucun code θ-cyclique de longueur impair sur $\mathbb{F}_4$ [35]. Plus généralement, ils ont montré dans [35] le résultat suivant :

Proposition 2.1.11 *Soit F_q un corps fini à q éléments, θ un automorphisme d'ordre m de $\mathbb{F}_q$. Si $pgcd(m,n)=1$, alors tout code θ-cyclique sur $\mathbb{F}_q^n$ est un code cyclique.*

2.1.5 Codes quantiques

On rappelle qu'un espace de Hilbert est un $\mathbb{C}$-espace vectoriel muni d'un produit scalaire et qu'il est complet par rapport à la norme associée à ce produit scalaire. Le foyer naturel des codes quantiques est l'espace de Hilbert $\mathbb{C}^{2^n}$. Cet espace est naturellement identifié au produit tensoriel de n copies de $\mathbb{C}^2$ noté $V=(\mathbb{C}^2)^{\otimes n}$. On commence cette partie par donner un aperçu du vocabulaire utilisé dans cet espace.
un vecteur u sera noté selon la notation de Dirac $\mid u >$ et on notera $< u \mid$ le vecteur transposé conjugué de u. On définit un quantum bit (qubit) par un vecteur non nul de l'espace vectoriel de dimension deux $\mathbb{C}^2$. On peut noter une base de $\mathbb{C}^2$ par $(\mid 0>, \mid 1>)$. Ainsi un qubit est exprimé par

$$\mid v >= \alpha \mid 0> +\beta \mid 1 \quad (\alpha,\beta \in \mathbb{C}, (\alpha,\beta) \neq (0,0)).$$

On définit un n-qubit par un vecteur non nul du produit tensoriel $V=(\mathbb{C}^2)^n=\mathbb{C}^{2^n}$. Nous choisissons la base suivante de V

$$\{\mid a >=\mid a_1a_2...a_n >=\mid a_1 > \otimes \mid a_2 > \otimes... \mid a_n >: a=(a_1,a_2,..a_n) \in \mathbb{F}_2^n\}.$$

Un n-qubit de V peut s'écrire de la façon suivante

$$\mid v >= \sum_{a\in\mathbb{F}_2^n} c_a \mid a >= \sum_{(a_1,a_2,..a_n)\in\mathbb{F}_2^n} c(a_1,a_2,..a_n) \mid a_1a_2...a_n >,$$

où $c_a=c(a_1,a_2,..a_n) \in \mathbb{C}$.
Nous travaillons dans V avec le produit scalaire hermitien défini par :
pour $\mid v >= \sum_{a\in\mathbb{F}_2^n} c_a \mid a >$, $\mid u >= \sum_{b\in\mathbb{F}_2^n} c_b \mid b >$ où ($c_a, c_b \in \mathbb{C}$), le produit scalaire hermitien de $\mid v >$ et $\mid u >$ est défini par :

$$< u \mid v >= \sum_{a\in\mathbb{F}_2^n} \overline{b_a} c_a \in \mathbb{C},$$

où $\overline{b_a}$ est le conjugué de b_a.

Définition 2.1.8 *Un code quantique Q est un sous espace de $V = (\mathbb{C}^2)^{\otimes n}$. L'entier n désigne la longueur du code Q.*

On note

$$K = \dim_{\mathbb{C}} Q, \quad k = \log_2 K.$$

Ainsi, $1 \leq K \leq 2^n$ et $0 \leq k \leq n$.
Comme dans le cas classique, un code quantique est caractérisé par trois paramètres : la longueur, la dimension et la distance minimale.

Définition 2.1.9 *Une erreur quantique sur V est un opérateur unitaire $\mathbb{C}$-linéaire agissant sur V qubit par qubit.*

Définition 2.1.10 *Il existe trois opérateurs d'erreurs élémentaires appelés matrices de Pauli, notés σ_x, σ_z et σ_y et définis par*

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

et

$$\sigma_y = i\sigma_x\sigma_z = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$

Les erreurs X, Z et Y sont appelés respectivement erreur bit, phase et erreur combinée bit-phase. L'ensemble $\{I, \sigma_x, \sigma_y, \sigma_z\}$ forme un groupe appelé groupe de Pauli, où I étant la matrice unité d'ordre 2. De plus, on a pour tout qubit $\mid v >= \alpha \mid 0 > +\beta \mid 1 > \in \mathbb{C}^2$

$$\sigma_x \mid v > \;=\; \beta \mid 0 > +\alpha \mid 1 > . \tag{2.1.6}$$
$$\sigma_z \mid v > \;=\; \alpha \mid 0 > -\beta \mid 1 > . \tag{2.1.7}$$
$$\sigma_y \mid v > \;=\; -i\beta \mid 0 > +i\alpha \mid 1 > . \tag{2.1.8}$$

Définition 2.1.11 *Une erreur quantique agissant sur $V = (\mathbb{C}^2)^{\otimes n}$ est de la forme*

$$e = i^{\lambda} w_1 \otimes w_2 \otimes ... \otimes w_{n-1},$$

où $0 \leq \lambda \leq 3$ et $w_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}$.

Les erreurs quantiques forment un groupe non abélien d'ordre 4^{n+1} noté

$$E_n = \{i^{\lambda}(w_1 \otimes w_2 \otimes ... \otimes w_{n-1}) \; ; \; 0 \leq \lambda \leq 3, w_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}\}.$$

Définition 2.1.12 *Le poids quantique d'une erreur quantique*

$$e = i^{\lambda} w_1 \otimes w_2 \otimes ... \otimes w_{n-1}$$

est défini par

$$w_Q(e) = \sharp\{i \; ; \; 1 \leq i \leq n, w_i \neq I\}.$$

Remarque 2.1.2 *Le poids quantique d'une erreur quantique est exactement le nombre de qubits sur lesquels les erreurs σ_x, σ_y ou σ_z agissent.*

Pour tout $0 \leq l \leq n$, nous definissons le sous ensemble $E_n(l)$ de E_n par

$$E_n(l) = \{e \in E_n \ ; \ w_Q(e) \leq l\}.$$

Définition 2.1.13 *La distance minimale d'un code quantique Q est définie comme étant le plus grand entier d vérifiant la propriété suivante :*
pour tout $\mid v_1 >, \mid v_2 > \in Q$, pour tout $e \in E_n(d-1)$, on a

$$< v_1 \mid v_2 >= 0 \Rightarrow < v_1 \mid e \mid v_2 >= 0.$$

Un code quantique de longueur n, de dimension K et de distance minimale d est noté (n, K, d) où $[[n, k, d]]$ avec $k = \log_2 K$.

Définition 2.1.14 *Un code quantique Q de paramètres (n, K, d) est dit pur si et seulement si pour tout $\mid v_1 >, \mid v_2 > \in Q$, pour tout $e \in E_n$ tel que $1 \leq w_Q(e) \leq d-1$, on a*

$$< v_1 \mid e \mid v_2 >= 0.$$

Beaucoup de recherches ont porté sur la construction des codes quantiques. La plus générale et la plus utilisée est la construction appelée : construction par code stabilisateur ou stabilizer code construction (SCC) qui construit un code quantique comme étant un sous espace propre d'un sous groupe abélien S du groupe d'erreurs E_n. Ces constructions ont été appliquées en $\mathbb{F}_2$ et $\mathbb{F}_4$. Pour donner une image de ces constructions, on a besoin d'introduire un nouveau produit scalaire sur $\mathbb{F}_2^{2n}$ appelé produit scalaire symplectique noté et défini par :

$$((a \mid b), (a' \mid b')) = ab' + ba'. \tag{2.1.9}$$

On a d'après [13]

Proposition 2.1.12 *Soit C un code linéaire de dimension $n-k$ de $\mathbb{F}_2^{2n}$ auto-orthogonal suivant le produit scalaire 2.1.9 et tel qu'il n'existe pas d'éléments dans $C^{\perp} \setminus C$ de poids $< d$. Il existe donc un code quantique Q transformant k qubits en n qubits et pouvant corriger $\frac{d-1}{2}$ erreurs.*

Pour parler de la construction des codes quantiques sur $\mathbb{F}_4^n$, on a besoin d'introduire les codes additifs . Rappelons qu'un code additif sur $\mathbb{F}_4$ est par définition un groupe additif de $\mathbb{F}_4^n$ ou un $\mathbb{F}_2$-espace vectoriel de $\mathbb{F}_4^n$. Rappelons de plus la définition de l'application $\mathbb{F}_2$-linéaire trace définie dans $\mathbb{F}_4$ par : pour tout $a \in \mathbb{F}_4$

$$Tr(a) = a + \overline{a},$$

où $\overline{a}$ est le conjugué de a dans $\mathbb{F}_4$. Le produit scalaire trace dans $\mathbb{F}_4^n$ est donné par : pour tout $u = (u_1, u_2, ..., u_n), v = (v_1, v_2, ..., v_n)$ de $\mathbb{F}_4^n$, on a

$$u * v = Tr(u.\overline{v}) = \sum_{i=1}^{n} (u_i \overline{v_i} + \overline{u_i} v_i). \tag{2.1.10}$$

Considérons dans ce qui suit l'application

$$\begin{array}{rcl} \phi : \mathbb{F}_2^{2n} & \to & \mathbb{F}_4^n \\ (a \mid b) & = & wa + \overline{w}b \end{array}$$

Signalons que pour tout C un sous espace vectoriel de $\mathbb{F}_2^{2n}$, $\phi(C)$ est un groupe additif de $\mathbb{F}_4^n$. On a d'après [13] le résultat suivant

Proposition 2.1.13 *Soit C un code additif auto-orthogonal pour le produit scalaire trace de $\mathbb{F}_4^n$ et tel qu'il n'existe pas d'éléments dans $C^\perp \setminus C$ de poids $< d$. Tout sous espace propre de $\phi^{-1}(C)$ est un code quantique additif de paramètres $[[n, k, d]]$.*

Dernièrement, il a été signalé dans plusieurs travaux que les probabilités d'avoir une erreur phase est nettement supérieure que celle d'avoir une erreur bit ou une erreur combiné bit-phase. En 2007, loffe et Mézard postulent dans [33] qu'il faut prendre en considération cette asymétrie dans la construction des codes quantiques. Depuis, des nouvelles études sur des constructions des codes quantiques asymétriques ont vu le jour. En effet, les codes quantiques asymétriques ont la capacité de corriger beaucoup plus d'erreurs phase que d'erreurs bit ou phase-bit. Notons dans la suite ρ_x, ρ_y, ρ_z les trois probabilités respectives pour qu'une erreur X, Y ou Z ait lieu. Il a été introduit de plus le terme $A = \rho_z/\rho_x$ pour signaler le caractère asymètrique de ces codes. On dit qu'un code asymètrique a une X distance d_x, respectivement Z distance d_z s'il peut détecter toutes les erreurs X de poids supérieur ou égal à $d_x - 1$, respectivement toutes les erreurs Z de poids supérieur ou égal à $d_z - 1$. Ainsi un code asymètrique quantique est noté $[[n, k, d_x/dz]]_q$.
La construction des codes asymétriques nécessite deux codes : un pour la correction de l'erreur bit X et un autre pour la correction de l'erreur phase Z. Pour deux codes C et D, nous utiliserons la notation suivante $wt(C \setminus D) = \min\{wt_H(u(\neq 0)) : u \in (C \setminus D)\}$. La construction est basée sur le résultat suivant :

Proposition 2.1.14 *[33] Soient deux codes linéaires C_x et C_z sur F_q^n de paramètres respectifs $[n, k_x]$ et $[n, k_z]$ et vérifiant $C_x^\perp \subseteq C_z$. Il existe alors un code quantique asymétrique de paramètres $[[n, k_x + k_z - n, d_x/d_z]]$ où $d_x = wt(C_x \setminus C_z^\perp)$ et $d_z = wt(C_z \setminus C_x^\perp)$.*

Si dans cette construction $d_x = wt(C_x)$ et $d_z = wt(C_z)$, le code sera dit pur. Pour plus de détails à propos de la construction des codes quantiques asymétriques, vous pouvez consulter [33] et [23].

Première partie

Des codes θ-cycliques aux codes quantiques sur $\mathbb{F}_4$

Chapitre 3

Factorisation des polynômes tordus et construction des codes θ-cycliques

3.1 Introduction

La motivation de l'étude des codes θ-cycliques était le nombre important pour une longueur donnée et les bons paramètres de ces codes. Cependant, la construction de cette classe de codes nécessite une bonne connaissance des polynômes tordus et de l'anneau $\mathbb{F}_q[X;\theta]$. Dans ce chapitre nous allons utiliser spécialement les résultats donnés dans les travaux de A.Leroy et ses co- auteurs ([16],[40],[41]) à propos de cette famille de polynômes. Ainsi nous allons élaborer via ces résultats, un algorithme de factorisation des polynômes tordus et nous allons donné une nouvelle construction d'un code θ-cyclique sur $\mathbb{F}_4$. Ces résultats seront utilisés dans le chapitre suivant pour donner des nouvelles constructions des codes θ-cycliques auto-orthogonaux en vue d'utilisation dans l'informatique quantique et spécialement pour construire des codes quantiques.

3.2 Définitions et notations

Soit $\mathbb{F}_q$ le corps fini à q éléments. Soient $a \in \mathbb{F}_q$, $c \in \mathbb{F}_q^*$ et θ un automorphisme de $\mathbb{F}_q$. On considère $\mathbb{F}_q[X;\theta]$ l'anneau des polynômes tordus. On utilise dans ce qui suit les notations et les définitions suivantes données dans [16]

$$a^c = \theta(c).a.c^{-1}. \tag{3.2.1}$$

On rappelle que pour $P = \Sigma_{i=0}^{n} a_i X^i \in \mathbb{F}_q[X;\theta]$, le reste de la division à droite de P par $X - a$ est

$$P(a) = \Sigma_{i=0}^{n} a_i N_i(a), \tag{3.2.2}$$

où

$$N_i(a) = \theta^{i-1}(a)\theta^{i-2}(a)....\theta(a).a \ \ pour \ \ i \geq 1, \tag{3.2.3}$$

$$N_0(a) = 1. \tag{3.2.4}$$

a est dite une racine à droite de P si et seulement si $P(a) = 0$.
Dans toute la suite, on désigne par racine de P toute racine à droite de P dans $\mathbb{F}_q[X;\theta]$. On a d'après [16, Lemma 2.1]

Lemme 3.2.1 *(formule produit) Soient $f, g \in \mathbb{F}_q[X;\theta]$, on a*

$$\begin{aligned}(fg)(a) &= 0 \quad si \;\; g(a) = 0, &(3.2.5)\\ &= f(a^{g(a)})g(a) \;\; si \;\; g(a) \neq 0. &(3.2.6)\end{aligned}$$

Preuve 3.2.2 *Effectuons la division euclidienne à droite de $g(X)$ par $X - a$. Nous obtenons*

$$g(X) = q_1(X)(X - a) + g(a),$$

où q_1 est un polynôme quelconque de $\mathbb{F}_q[X;\theta]$.
Dans le cas où $g(a) = 0$, le résultat est immédiat. Supposons maintenant que $g(a) \neq 0$. En effectuant la division euclidienne de $f(X)$ par $X - a^{g(a)}$, on obtient

$$f(X) = q_2(X)(X - a^{g(a)}) + f(a^{g(a)}),$$

$q_2 \in \mathbb{F}_q[X;\theta]$. Ainsi,

$$f(X)g(X) = (q_2(x)(X-a^{g(a)})+f(a^{g(a)}))q_1(X)(X-a)+q_2(X)(X-a^{g(a)})g(a)+f(a^{g(a)})g(a)).$$

Le résultat est immédiat compte tenu du fait que

$$(X - a^{g(a)})g(a) = \theta(g(a))(X - a).$$

3.3 Notions sur le ppcm dans $\mathbb{F}_q[X;\theta]$

Définition 3.3.1 *Soient $f, g \in \mathbb{F}_q[X;\theta]$. On appelle plus petit commun multiple de f et g le polynôme unitaire de $\mathbb{F}_q[X;\theta]$ noté $ppcm(f, g)$ ou $[f, g]$ et vérifiant*

$$Rf \cap Rg = R[f, g],$$

où l'on a noté $R = \mathbb{F}_q[X;\theta]$.

On a d'après [16, Lemma 2.1]

Lemme 3.3.1 *Soient $f, g \in \mathbb{F}_q[X;\theta]$ et $a \in \mathbb{F}_q$, on a*

$$\begin{aligned}ppcm(f, X - a) &= f \quad Si \;\; f(a) = 0, &(3.3.7)\\ &= (X - a^{f(a)})f \;\; Si \;\; f(a) \neq 0. &(3.3.8)\end{aligned}$$

Exemple 3.3.2 *Travaillons dans le corps $\mathbb{F}_4$ muni de l'automorphisme de Frobenius défini par $\theta(a) = a^2$. Dans $\mathbb{F}_4[X;\theta]$, considérons $f = X - 1$ et $g = X - w$. Signalons que dans le cas de $\mathbb{F}_4$ la notation 3.2.1 devient $a^c = ac$. On a donc*

$$\begin{aligned}[X-1, X-w] &= (X - wf(w)))(X-1), \\ &= (X - w(w-1))(X-1), \\ &= X^2 - 1.\end{aligned}$$

En notant $P = [X-1, X-w]$, nous vérifions bien d'après 3.2.3 que $P(1) = N_2(1) - N_0(1) = 0$ et $P(w) = N_2(w) - N_0(w) = 0$. D'un autre côté, en notant $Q = (X-1)(X-w) = X^2 + wX + w$, on voit bien $Q(1) = N_2(1) + wN_1(1) + wN_0(1) \neq 0$. L'explication de ce résultat réside dans le fait que $X - 1$ est un diviseur à gauche et non à droite de Q et donc 1 n'est pas une racine -explicitement une racine à droite - de ce polynôme.

Pour $x_1, x_2, ..., x_n \in \mathbb{F}_q$ et en utilisant le lemme 3.3.7, on peut définir le *ppcm* des polynômes $X - x_j$ de la façon suivante : $P_0(X) = 1$ et pour tout $1 \leq i \leq n$, on a

$$\begin{aligned}P_i(X) = ppcm(X - x_i, P_{i-1}(X)) &= P_{i-1}(X) \quad si \quad P_{i-1}(x_i) = 0, &(3.3.9)\\ &= (X - x_i^{P_{i-1}(x_i)})P_{i-1}(X) \quad si \quad P_{i-1}(x_i) \neq 0 &(3.3.10)\end{aligned}$$

Nous présentons dans la suite quelques définitions et résultats donnés dans [16] qui vont nous aider à présenter un algorithme de factorisation des polynômes tordus.

Définition 3.3.2 *Une famille $\{x_1, x_2, ..., x_n\}$ de $\mathbb{F}_q$ est dite P-indépendante si et seulement si $deg([X - x_i; 1 \leq i \leq n]) = n$.*

Remarque 3.3.1 *1) Il est clair d'après cette définition que la famille $\{x_1, x_2, ..., x_n\}$ est P-indépendante si et seulement $p_i(x_{i+1}) \neq 0$ pour tout $1 \leq i \leq n-1$.*
2) Dans le cas d'une famille $\{x_1, x_2, ..., x_n\}$ P-indépendante, on a

$$P_n(X) = [X - x_i; 1 \leq i \leq n] = (X - y_n)(X - y_{n-1})...(X - y_1),$$

où $y_i = x_i^{p_{i-1}(x_i)} \quad i \in \{1...n\}$.

3.4 Factorisation dans $\mathbb{F}_4[X;\theta]$

Soit l' automorphisme de Frobenius dans $\mathbb{F}_4$ défini par $\theta(a) = a^2$, n un entier > 0 et $q = 2^n$. Rappelons que dans le cas de $\mathbb{F}_4$ la notation 3.2.1 devient $a^c = ac$.

Proposition 3.4.1 *Soit $P = \Sigma_{i=0}^n a_i X^i \in \mathbb{F}_4[X;\theta]$. Alors, a est une racine de P si et seulement si a est une racine du polynôme $P = \Sigma_{i=0}^n a_i X^{2^i - 1} \in \mathbb{F}_4[X]$.*

Preuve 3.4.2 *La preuve est immédiate d'après la définition des $N_i's$.*

Remarque 3.4.1 *1) Notons que ce résultat reste vrai dans n'importe quel corps de caractéristique* 2 *muni de l'automorphisme de Frobenuis.*
2)On peut énoncer un résultat similaire dans le cas du corps $\mathbb{F}_{p^m}$ *muni de l'automorphisme* $\theta(x) = x^p$ *et où* p *est un nombre premier, à savoir : a est une racine de* $P = \Sigma_{i=0}^n a_i X^i$ *si et seulement si a est une racine du polynôme* $P = \Sigma_{i=0}^n a_i X^{\frac{p^i-1}{p-1}} \in \mathbb{F}_{p^m}[X]$.

Lemme 3.4.3 *Soit* $P = \Sigma_{i=0}^n a_i X^i \in \mathbb{F}_4[X;\theta]$. *Pour tout* $a \in \mathbb{F}_4 \setminus \{0\}$, *on a*

$$P(a) = \Sigma_{i=0,i\ pair}^n a_i + \Sigma_{i=0,i\ impair}^n a_i a.$$

Preuve 3.4.4 *Le résultat est immédiat compte tenu de 8.1.1 et du fait que*

$$\begin{aligned} N_i(a) &= \theta^{i-1}(a)\theta^{i-2}(a)....\theta(a).a \ \ pour \ \ i \geq 1, \\ &= a^2.a.a^2.a....a^2.a = 1 \ \ si\ i\ pair, \\ &= a.a^2.a.a^2.a....a^2.a = a \ \ si\ i\ impair. \end{aligned}$$

Remarque 3.4.2 *Pour tout* $P = \Sigma_{i=0}^n a_i X^i \in \mathbb{F}_4[X;\theta]$, *on a donc*
1. Si $\Sigma_{i=0,i\ impair}^n a_i \neq 0$ *alors P admet une racine unique dans* $\mathbb{F}_4$ *égale à*

$$\frac{\Sigma_{i=0,i\ pair}^n a_i}{\Sigma_{i=0,i\ impair}^n a_i}.$$

2. Si $\Sigma_{i=0,i\ impair}^n a_i = 0$ *et* $\Sigma_{i=0,i\ pair}^n a_i \neq 0$ *alors P n'admet aucune racine dans* $\mathbb{F}_4$.

Exemple 3.4.5 *Il résulte de ce qui prècède et compte tenu que les racines de* $X^n - 1$ *dans* $\mathbb{F}_4[X;\theta]$ *sont les racines du polynôme* $X^{2^n-1} - 1$, *que l'ensemble des racines de* $X^n - 1$ $\cup\{0\}$ *est l'ensemble* $\mathbb{F}_{2^n}$.

Le résultat suivant est une conséquence de l'étude complète faite dans [16] concernant la factorisation des polynômes dans les anneaux d'Ore.

Proposition 3.4.6 *Soit* $P = X^n - 1 \in \mathbb{F}_4[X;\theta]$ *et w la racine primitive* $2^n - 1$ *de l'unité. Pour toute* $\mathbb{F}_2$ *-famille libre* $(u_1, u_2, ..., u_n)$ *de* $\mathbb{F}_{2^n}$, *il existe une unique factorisation de* $X^n - 1$ *en facteurs linéaires donnée par :*

$$X^n - 1 = (X - y_n)(X - y_{n-1}).....(X - y_1), \tag{3.4.11}$$

où l'on a noté

$$y_i = P_{i-1}(x_i)x_i \ \ et \ \ x_i = w^{u_i} = \theta(u_i)wu_i^{-1} = u_i w, \tag{3.4.12}$$

et où la famille des polynômes P_i *est défini de la façon récurrente suivante :*

$$P_0(X) = 1, \quad P_1(X) = X - x_1,$$

et

$$P_i(X) = ppcm(X - x_i, P_{i-1}(X)) = (X - P_{i-1}(x_i).x_i)P_{i-1}(X) = (X - y_i).P_{i-1}(X). \tag{3.4.13}$$

Preuve 3.4.7 *Pour $q = 2^n$, on étend l'automorphisme θ à $\mathbb{F}_q$. On définit l'ensemble $\Delta(w) := \{w^x \,|\, x \in \mathbb{F}_q^*\}$, c'est à dire*

$$\Delta(w) := \{wx \,|\, x \in \mathbb{F}_q^*\}.$$

Il est clair que cet ensemble contient les racines de $X^n - 1$ sur $\mathbb{F}_q$. D'un autre côté, considérons l'ensemble

$$C(w) := \{x \in \mathbb{F}_q^* \,|\, w^x = w\} \cup \{0\}.$$

Ainsi,

$$C(w) := \mathbb{F}_2.$$

D'où, la famille $(u_1, u_2, ..., u_n)$ est $C(w)-$libre compte tenu du fait qu'elle est est $\mathbb{F}_2-$libre. Il vient alors d'après [16, Th. 5.3] que la famille $(x_1, x_2, ..., x_n)$ est $P-$ indépendante et par conséquent que le degré du ppcm des $X - w_i$ est n où l'on a tenu compte de [16, Def. 2.3]. Comme ce ppcm divise $X^n - 1$, il est donc égal à $X^n - 1$.
Il vient alors d'après [16, Cor. 5.4] que l'ensemble des racines de $X^n - 1 \cup\{0\}$ dans l'anneau des polynômes tordus est égal à

$$\oplus_{i=1}^n C(w)u_i = \oplus_{i=1}^n \mathbb{F}_2 u_i.$$

où $\oplus$ dénote la somme directe des espaces vectoriels. D'après [16, Th 6.4], il existe une correspondance bijective entre les drapeaux des sous espaces vectoriels de la somme $\oplus_{i=1}^n \mathbb{F}_2 u_i$ et la factorisation de $X^n - 1$, ce qui démontre le résultat.

Remarque 3.4.3 *Le nombre de factorisations du polynôme $X^n - 1$sur $\mathbb{F}_q$ est égal au nombre des familles $\mathbb{F}_2$-libres de $\mathbb{F}_q$ avec $q = 2^n$.*

On propose dans la suite deux algorithmes en application à ces résultats : le premier donne une factorisation en facteurs linéaires de $X^n - 1$ à partir d'une $\mathbb{F}_2$ -famille libre de $\mathbb{F}_q$ et le deuxième donne tous les diviseurs de $X^n - 1$ sur l'anneau des polynômes tordus $\mathbb{F}_q[X; \theta]$.

Algorithme de la factorisation
Soient $q = 2^n$ et w un élément primitif d'ordre $2^n - 1$ dans le groupe $\mathbb{F}_q^*$. Considérons une $\mathbb{F}_2$-famille libre $(u_1, u_2, ..., u_n)$ dans l'ensemble des racines de $X^n - 1$. On pose $w_i = u_i w$, $P_0 = 1,\ P_1 = X - w_1$, et on définit par induction et en utilisant la définition 3.3.9

$$P_i = ppcm(X - w_i, P_{i-1}),$$

pour $i = 2, \cdots, n$.
La factorisation de $X^n - 1$ est donnée alors par :

$$X^n - 1 = \prod_{i=1}^{n} (X - P_{i-1}(w_i)w_i).$$

Algorithme des diviseurs

Considérons une $\mathbb{F}_2$-famille libre $(u_1, u_2, ..., u_k)$ de $\mathbb{F}_q$. Le diviseur de $X^n - 1$ correspondant à cette famille est le $ppcm(X - w_i,\ i = 1, \cdots, k)$, où $w_i = u_i w$. Il est calculé en utilisant le procédé récurent suivant : on pose $P_0 = 1,\ P_1 = X - w_1$, et

$$P_i = ppcm(X - w_i, P_{i-1}).$$

On a d'après la formule 3.3.7 que ce *ppcm* est égal à P_{i-1} si $P_{i-1}(w_i) = 0$ et $(X - P_{i-1}(w_i)w_i)P_i$ sinon.

Remarque 3.4.4 *L'algorithme précédent nous permet de déterminer les diviseurs de $X^n - 1$ dans $\mathbb{F}_q[X;\theta]$ et non dans $\mathbb{F}_4[X;\theta]$. Nous présentons dans la deuxième partie de ce chapitre un théorème permettant de faire une sélection dans les familles $(u_1, u_2, ..., u_k)$ afin de ne retenir que celles qui donneront des diviseurs de $X^n - 1$ dans $\mathbb{F}_4[X;\theta]$.*

3.5 Construction des codes θ-cycliques

Proposition 3.5.1 *[9] Soit $P = \Sigma_{i=0}^{n} a_i X^i \in \mathbb{F}_4[X;\theta]$. L'ensemble des racines de $P \cup \{0\}$ est un $\mathbb{F}_2$-espace vectoriel de dimension $n - min\{i \ : a_i \neq 0\}$.*

On considère dans la suite le polynôme $P = X^n - 1 = gh = hg$ de $\mathbb{F}_4[X;\theta]$.

Définition 3.5.1 *Soit C le code θ-cyclique sur $\mathbb{F}_4$ engendré par le polynôme tordu g. On appelle ensemble de zéros du code C l'ensemble des racines du polynôme g.*

On note dans la suite $V(C)$ ou $V(g)$ l'ensemble de zéros du code $C \cup \{0\}$.

Corollaire 3.5.2 *Soit le polynôme $P = X^n - 1 = gh = hg$ de $\mathbb{F}_4[X;\theta]$ et C le code θ-cyclique engendré par g. $V(C)$ est un $\mathbb{F}_2$-sous espace vectoriel de $\mathbb{F}_{2^n}$ de dimension $deg(g)$.*

Proposition 3.5.3 *Pour tout $\mathbb{F}_2$-sous espace vectoriel de $\mathbb{F}_{2^n}$, on peut construire un code θ-cyclique de longueur n sur $\mathbb{F}_{2^n}$.*

Preuve 3.5.4 *Soit V un $\mathbb{F}_2$-sous espace vectoriel de $\mathbb{F}_{2^n}$ et soit $B = (x_1, ..., x_k)$ une base de V. On sait déjà d'après le chapitre précédent que $g = ppcm_{1 \leq i \leq k}(X - x_i)$ est un diviseur de $X^n - 1$ dans $\mathbb{F}_{2^n}[X;\theta]$. Ainsi le code $C =< g >$ est un code θ-cyclique de longueur n sur $\mathbb{F}_{2^n}$.*

Le résultat suivant est prouvé par Ulmer et ses co-auteurs dans [9]. Nous présentons dans ce qui suit une nouvelle démonstration de cette proposition.

Proposition 3.5.5 *Soit σ le générateur du groupe de Galois; de $[\mathbb{F}_{2^n}/\mathbb{F}_4]$ et soit une $\mathbb{F}_2$-famille libre $(x_1, ..., x_k)$ de $\mathbb{F}_{2^n}$. Si le $\mathbb{F}_2$ espace vectoriel engendré par $(x_1, ..., x_k)$ est stable par σ alors le $ppcm_{1 \leq i \leq k}(X - x_i)$ appartient à $\mathbb{F}_4[X;\theta]$.*

Preuve 3.5.6 *Montrons le résultat par récurrence. Si $k = 1$ on a*

$$ppcm_{1\leq i\leq k}(X - x_i) = X - x_1.$$

Comme $\sigma(x_1) = \epsilon x_1$ où $\epsilon = 0$ ou 1, on a si $\epsilon = 0$ alors $\sigma(x_1) = 0$ et donc $x_1 = 0 \in \mathbb{F}_4$ et si $\epsilon = 1$ alors $\sigma(x_1) = x_1$ et par suite $x_1 \in \mathbb{F}_4$.
Supposons maintenant que le résultat est vrai jusqu'à l'ordre p et montrons le pour l'ordre $p+1$. Posons tout d'abord $f_k = ppcm_{1\leq i\leq k}(X - x_i)$, on a d'après 3.3.7

$$\begin{aligned} f_{p+1} = ppcm(f_p, X - x_{p+1}) &= f_p \quad si \;\; f_p(x_{k+1}) = 0, && (3.5.14)\\ &= (X - x_{p+1}^{f_p(x_{p+1})})f_p \;\; sinon, && (3.5.15)\end{aligned}$$

où l'on a noté

$$x_{p+1}^{f_p(x_{p+1})} = \theta(x_{p+1})f_p(x_{p+1})x_{p+1}^{-1} = x_{p+1}f_p(x_{p+1}).$$

Dans le premier cas, on a d'après l'hypothèse de récurrence $f_{p+1} = f_p \in \mathbb{F}_4[X;\theta]$. Traitons maintenant le cas où $f_p(x_{p+1}) \neq 0$. Il vient

$$f_{p+1} = (X - x_{p+1}f_p(x_{p+1}))f_p.$$

Comme f_p est un polynôme de $\mathbb{F}_4[X;\theta]$ d'après l'hypothèse de récurrence, il suffit donc de montrer que $x_{p+1}f_p(x_{p+1}) \in \mathbb{F}_4$. En notant $f_p = \Sigma_{i=0}^p a_i X^i$, on a d'après 3.2.2

$$\begin{aligned} f_p(x_{p+1}) &= \Sigma_{i=0}^p a_i N_i(x_{p+1}), && (3.5.16)\\ &= \Sigma_{i=0}^p a_i \theta^{i-1}(x_{p+1})\theta^{i-2}(x_{p+1})....\theta(x_{p+1}).x_{p+1}, && (3.5.17)\\ &= \sum_{i=0}^{p} a_i x_{p+1}^{2^i-1}, && (3.5.18)\end{aligned}$$

où les a_i sont dans $\mathbb{F}_4$. Ainsi

$$x_{p+1}f_p(x_{p+1}) = \Sigma_{i=0}^p a_i x_{p+1}^{2^i}.$$

Pour montrer donc que $x_{p+1}f_p(x_{p+1}) \in \mathbb{F}_4$, il suffit de montrer que $\sigma(\Sigma_{i=0}^p a_i x_{p+1}^{2^i}) = \Sigma_{i=0}^p a_i x_{p+1}^{2^i}$. Comme les a_i sont dans $\mathbb{F}_4$ et que σ est invariant sur $\mathbb{F}_4$, on a

$$\sigma(\sum_{i=0}^{p} a_i x_{p+1}^{2^i}) = \Sigma_{i=0}^p a_i \sigma(x_{p+1})^{2^i}.$$

D'autre part, on sait par hypothèse que le $\mathbb{F}_2$ espace vectoriel engendré par $(x_1, ..., x_{p+1})$ est stable par σ. Ainsi

$$\sigma(x_{p+1}) = \sum_{j=0}^{p+1} \epsilon_j x_j,$$

où les ϵ_j sont égals à 0 ou 1. Par conséquent

$$\begin{aligned}\sigma(\sum_{i=0}^{p} a_i x_{p+1}^{2^i}) &= \sum_{i=0}^{p} a_i (\sum_{j=0}^{p+1} \epsilon_j x_j)^{2^i}, &(3.5.19)\\ &= \sum_{j=0}^{p+1}\sum_{i=0}^{p} a_i (\epsilon_j x_j)^{2^i}, &(3.5.20)\\ &= \sum_{i=0}^{p} a_i \epsilon_p (x_{p+1})^{2^i} + \sum_{j=0}^{p} (\epsilon_j (\sum_{i=0}^{p} a_i (x_j)^{2^i}). &(3.5.21)\end{aligned}$$

Or, comme f_p est défini par $f_p = ppcm_{1\leq j\leq p}(X - x_j)$, il vient que x_j est une racine de f_p pour tout $1 \leq j \leq p$ et ainsi

$$f_p(x_j) = 0 = \sum_{i=0}^{p} a_i x_j^{2^i-1}.$$

Il vient donc

$$\sum_{i=0}^{p} a_i (x_j)^{2^i} = 0,$$

et ce, pour tout $0 \leq j \leq p$. Ainsi, on a compte tenu du fait que $\epsilon_j = 0$ ou 1, que

$$\sigma(\sum_{i=0}^{p} a_i x_{p+1}^{2^i}) = \sum_{i=0}^{p} a_i \epsilon_p (x_{p+1})^{2^i}.$$

Deux cas se présentent : le premier si $\epsilon_p = 0$ et on a donc $\sigma(\sum_{i=0}^{p} a_i x_{p+1}^{2^i}) = 0$ ce qui implique conte tenu que σ est un automorphisme que $\sum_{i=0}^{p} a_i x_{p+1}^{2^i} = 0 \in \mathbb{F}_4$. Le deuxième cas est que $\epsilon_p = 1$ et on a alors $\sigma(\sum_{i=0}^{p} a_i x_{p+1}^{2^i}) = \sum_{i=0}^{p} a_i x_{p+1}^{2^i}$ et donc $\sum_{i=0}^{p} a_i x_{p+1}^{2^i} \in \mathbb{F}_4$ ce qu'on veut démontrer.

Exemple 3.5.7 *Dans tous ces exemples nous notons dans chaque cas de n, u un élément primitif de $\mathbb{F}_{2^n}$.*
1. Pour $n = 4$ l'ensemble des racines du polynôme $X^4 + 1 \cup\{0\}$ est $\mathbb{F}_{16}$. La famille $\mathbb{F}_2$ libre $(1, u, u^2, u^7)$ nous donne la factorisation suivante de X^4+1 en facteurs premiers sur $\mathbb{F}_4[X;\theta]$:

$$X^4 + 1 = (X + 1)(X + w^2)(X + w)(X + 1).$$

La famille $\mathbb{F}_2$ libre $(1, u^2, u^4, u^6)$ nous donne la factorisation suivante de $X^4 + 1$:

$$X^4 + 1 = (X + 1)(X + w)(X + w^2)(X + 1).$$

2. Pour $n = 5$ l'ensemble des racines du polynôme $X^5 + 1 \cup\{0\}$ est $\mathbb{F}_{2^5}$. La famille $\mathbb{F}_2$ libre $(1, u^{341})$ nous donne le diviseur de degré 2 de $X^4 + 1$ dans $\mathbb{F}_4[X;\theta]$ donné par

$$X^2 + 1 = (X + w^2)(X + w).$$

3. Pour $n = 8$ l'ensemble des racines du polynôme $X^8 + 1 \cup\{0\}$ est $\mathbb{F}_{2^8}$. La famille $\mathbb{F}_2$ libre $(1, u^2, u^6, u^7, u^{17}, u^{34}, u^{138}, u^{157})$ nous donne la factorisation suivante de $X^8 + 1$ en facteurs premiers sur $\mathbb{F}_4[X; \theta]$:

$$X^8 + 1 = (X + 1)(X + w^2)^3(X + w)^3(X + 1).$$

La famille $\mathbb{F}_2$ libre $(1, u^2, u^7, u^{17}, u^{20}, u^{25}, u^{31}, u^{34})$ nous donne la factorisation suivante de $X^8 + 1$ en facteurs premiers sur $\mathbb{F}_4[X; \theta]$:

$$X^8 + 1 = (X + 1)(X + w^2)(X + 1)^2(X + w^2)(X + w)^2(X + 1).$$

La famille $\mathbb{F}_2$ libre $(1, u^{17}, u^{34})$ nous donne le diviseur de degré 3 de $X^8 + 1$ dans $\mathbb{F}_4[X; \theta]$ $x^3 + x^2 + x + 1$.

Le résultat suivant est immédiat.

Corollaire 3.5.8 *Soit σ le générateur du groupe de Galois de $[\mathbb{F}_{2^n}/\mathbb{F}_4]$. On a*
1. Pour toute $\mathbb{F}_2$-famille libre $(x_1, ..., x_k)$ de $\mathbb{F}_{2^n}$ vérifiant que le $\mathbb{F}_2$ espace vectoriel engendré par $(x_1, ..., x_k)$ soit stable par σ, on peut construire un code θ-cyclique de longueur n et de dimension k sur $\mathbb{F}_4$.
2. Pour tout $\mathbb{F}_2$ espace vectoriel de $\mathbb{F}_{2^n}$ stable par σ et de dimension k, on peut construire un code θ-cyclique de longueur n et de dimension k sur $\mathbb{F}_4$.

Remarque 3.5.1 *Il est clair que ce code est le code θ-cyclique engendré par $ppcm_{1\leq i\leq k}(X - x_i)$.*

Chapitre 4

Codes θ-cycliques auto-orthogonaux et informatique quantique

4.1 Introduction

La richesse des codes cycliques tordus en nombre et paramètres incite à leur utilisation dans l'informatique quantique et précisément dans la construction des codes quantiques. Cependant, peu de résultats existent dans la littérature concernant les codes θ-cycliques auto-orthogonaux point de départ de la construction des codes quantiques. En effet, des constructions des codes θ-cycliques auto-duaux euclidiens et hermitiens ont été données dans les travaux effectués sur les codes θ-cycliques [12], [11] et [8] via l'utilisation des bases de Gröbner. Cependant, on ne peut pas utiliser d'une façon directe les bases de Gröbner pour construire des codes θ-cycliques auto -orthogonaux. Dans ce chapitre, nous avons essayé de résoudre ce problème en donnant des nouvelles constructions des codes θ-cycliques auto-orthogonaux. La première construction consiste à donner une caractérisation de l'ensemble de zéros d'un code θ-cyclique auto orthogonal. La motivation de cette construction provient du travail de Hufmann et Vera dans [31] où ils ont défini l'ensemble de zéros d'un code cyclique comme étant l'ensemble des racines de son générateur et ont donné une caractérisation de l'ensemble de zéros d'un code cyclique auto-orthogonal. Cette caractérisation va leur permettre de donner une construction d'un tel code. Pour donner une caractérisation analogue de l'ensemble des zéros d'un code $\theta-$cyclique auto-orthogonal, nous allons utiliserl'analogie qui existe entre les polynômes additives où les p-polynômes et les polynômes tordus de $\mathbb{F}_q[X;\theta]$. Plus précisément, nous utiliserons les résultats donnés dans [27] pour étudier l'ensemble de zéros d'un code θ-cyclique auto-orthogonal. En effet, nous définirons l'ensemble de zéros d'un code θ-cyclique et nous donnerons plusieurs caractérisations de cet ensemble permettant de donner une méthode pour la construction d'un code θ-cyclique auto-orthogonal sur $\mathbb{F}_4$. Pour une documentation complète sur cette classe de polynômes, à savoir les polynômes additives où les p-polynômes, le lecteur est invité à consulter [34],[27] et [52]. La deuxième construction présentée dans ce chapitre consiste à considérer un code auto orthogonal comme étant un sous code d'un code θ-cyclique auto dual sur $\mathbb{F}_4$ en se basant sur l'algorithme de

factorisation d'un polynôme tordu. Signalons enfin que les résultats de ce chapitre vont nous permettre de donner des nouvelles constructions de codes quantiques. Ce chapitre est divisé en cinq sections. Nous donnerons quelques notions sur les $p-$polynômes dans la première section et une caractérisation de l'ensemble de zéros d'un code $\theta-$cyclique auto-orthogonal dans la deuxième. La troisième section consiste à présenter une construction d'un code auto-orthogonal comme étant un sous code d'un code auto-dual. Enfin, nous utiliserons dans les deux dernières sections ces résultats pour construire des codes quantiques sur $\mathbb{F}_4$.

4.2 Notions sur les $p-$polynômes

Dans cette section, on considère un nombre premier p, r un entier ≥ 1 et $q = p^r$. Dans le corps fini $\mathbb{F}_q$ à q éléments, on considère l'automorphisme θ défini par $\theta(x) = x^p$. On a d'après [42].

Définition 4.2.1 *On appelle $p-$ polynôme tout polynôme L de la forme*

$$L(x) = \sum_{i=0}^{n} a_i x^{p^i},$$

où les a_i sont dans $\mathbb{F}_q$.

Compte tenu que l'endomorphisme θ est stable sur $\mathbb{F}_p$, il est clair que le polynôme L est $\mathbb{F}_p-$linéaire. Ainsi, on a pour tout $x, y \in \mathbb{F}_q$, $\alpha \in \mathbb{F}_p$

$$\begin{aligned} L(x+y) &= L(x) + L(y). \\ L(\alpha x) &= \alpha L(x). \end{aligned}$$

On a de plus le résultat suivant.

Proposition 4.2.1 *[42] L'ensemble des racines d'un $p-$polynôme est un $\mathbb{F}_p-$espace vectoriel de $\mathbb{F}_q$.*

Dans ce qui suit, on aura recours au déterminant de Moore introduit par E.H.Moore en 1896. On a la définition suivante :

Définition 4.2.2 *Soit k un corps contenant le corps fini $\mathbb{F}_p$, θ l'automorphisme de k défini par $\theta(x) = x^p$ et $\{w_1, w_2, ..., w_n\} \subset k$. On appelle déterminant de Moore le nombre noté et défini par*

$$\Delta = \Delta(w_1, ..., w_n) = \begin{vmatrix} \theta^0(w_1) & \cdots & \theta^0(w_n) \\ \theta(w_1) & \cdots & \theta(w_n) \\ & & \\ \theta^{n-1}(w_1) & & \theta^{n-1}(w_n) \end{vmatrix}, \tag{4.2.1}$$

$$= \begin{vmatrix} w_1 & \cdots & w_n \\ w_1^p & \cdots & w_n^p \\ & & \\ w_1^{p^{n-1}} & & w_n^{p^{n-1}} \end{vmatrix}. \tag{4.2.2}$$

Lemme 4.2.2 *Soient $w_1, ..., w_n \in \mathbb{F}_q$. On a*

$$\Delta(w_1, ..., w_n) = w_1 \prod_{j=1}^{n-1} \prod_{c_1,...,c_j \in \mathbb{F}_p} (w_{j+1} - \sum_{k=1}^{j} c_k w_k).$$

Le résultat précédent montre donc que le déterminant de Moore de $w_1, ..w_n$ est différent de 0 si et seulement si $w_1, ..w_n$ sont des éléments $\mathbb{F}_p-$indépendents de $\mathbb{F}_q$.
Pour plus d'informations concernant les $p-$polynômes, le lecteur peut consulter [42], [27] et [52]. D'un autre côté, notre intérêt pour cette classe de polynômes réside dans la relation qui existe entre les polynômes tordus, les $p-$polynômes et les équations aux différences de $\mathbb{F}_q$. En effet, en se situant dans l'ensemble des endomorphismes de $\mathbb{F}_q$, on peut considérer le sous espace vectoriel engendré par les puissances de θ et noté $(\mathbb{F}_q, \theta)$ et appelé ensemble des équations aux différences de $\mathbb{F}_q$. Nous définissons l'application suivante :

$$\Psi \quad : \quad \mathbb{F}_q[X;\theta] \mapsto (\mathbb{F}_q, \theta) \tag{4.2.3}$$

$$P \quad = \quad \sum_{i=0}^{n} a_i X^i \mapsto \Psi(P) = \sum_{i=0}^{n} a_i \theta^i. \tag{4.2.4}$$

Il est clair que l'application Ψ est une application $\mathbb{F}_p-$linéaire, surjective et non injective. De plus, la non commutativité de la multiplication dans $\mathbb{F}_q[X;\theta]$ correspond à la non commutativité de la composition dans $(\mathbb{F}_q, \theta)$. D'un autre côté, pour tout polynôme P de $\mathbb{F}_q[X;\theta]$, $\Psi(P(x)) = \sum_{i=0}^{n} a_i x^{p^i}$ est un $p-$polynôme et toute solution de l'équation $\sum_{i=0}^{n} a_i \theta^i(y) = 0$ est une racine du polynôme $\Psi(P(x))$ comme polynôme de $\mathbb{F}_q[X]$. On a d'après [9], le résultat suivant.

Proposition 4.2.3 *Soient un polynôme P de $\mathbb{F}_q[X;\theta]$ et un élément a d'une extension $\mathbb{F}_{q^s}$ de $\mathbb{F}_q$. Alors, a est une racine de P si et seulement $\frac{\Theta(a)}{a}$ est une solution de l'équation $\Psi(P(y)) = 0$, où Θ est l'extension de l'automorphisme θ à $\mathbb{F}_{q^s}$.*

4.3 Caractérisation d'un code θ-cyclique auto-orthogonal

On utilisera dans la suite les résultats de la section précédente avec $p = 2$, $q = 4$ et θ l'automorphisme de Frobenuis sur $\mathbb{F}_4$ défini par $\theta(x) = x^2$. Dans ce cas, le sous espace vectoriel $(\mathbb{F}_4, \theta)$ est tout simplement le F_2- espace vectoriel engendré par $\{Id, \theta\}$. Ainsi, la proposition 4.2.3 et la définition de l'automorphisme de Frobenuis impliquent que
a est une racine d'un polynôme P de $\mathbb{F}_4[X;\theta]$ si et seulement si a est une solution de l'équation $\Psi(P(y)) = 0$.

Comme pour les polynômes de $\mathbb{F}_q[X]$, on définit dans $\mathbb{F}_q[X;\theta]$ la résultante de deux polynômes. On a le résultat suivant.

Définition 4.3.1 *[20] Pour deux polynômes $P=\sum_{i=0}^{n}a_iX^i$, $Q=\sum_{i=0}^{m}b_iX^i$ de $\mathbb{F}_4[X;\theta]$, la résultante de P et Q est le nombre noté et défini par*

$$R(P,Q)=\begin{vmatrix} a_0 & a_1 & a_2 & .... & a_n & 0 & ... & 0 \\ 0 & \theta(a_0) & \theta(a_1) & . . & \theta(a_{n-1}) & \theta(a_n) & . . & 0 \\ & & & & & & & \\ 0 & 0 & 0 & 0 & \theta^{n-1}(a_0) & .. & .. & \theta^{n-1}(a_n) \\ b_0 & b_1 & b_2 & .... & b_n & 0 & ... & 0 \\ 0 & \theta(b_0) & \theta(b_1) & . . & \theta(b_{n-1}) & \theta(b_n) & . . & 0 \\ & & & & & & & \\ 0 & 0 & 0 & 0 & \theta^{n-1}(b_0) & .. & .. & \theta^{n-1}(b_n) \end{vmatrix}.$$

Dans [27], D.Goss a défini la résultante de deux éléments de $(\mathbb{F}_q,\theta)$. En effet pour deux polynômes unitaires et séparables P,Q de $\mathbb{F}_q[X;\theta]$ et en notant respectivement $(\alpha_1,...,\alpha_n)$ et $(\beta_1,...,\beta_m)$ les bases des ensembles de solutions de $\Psi(P)$ et $\Psi(Q)$, on a

$$R(\Psi(P),\Psi(Q))=\frac{\Delta(\alpha_1,...,\alpha_n,\beta_1,....,\beta_m)}{\Delta(\alpha_1,...,\alpha_n)\Delta(\beta_1,....,\beta_m)}.$$

Ainsi, en utilisant la proposition 4.2.3 pour $q=4$, nous obtenons que

$$R(P,Q)\neq 0 \Leftrightarrow R(\Psi(P),\Psi(Q))\neq 0. \tag{4.3.5}$$

Dans toute la suite, on utilisera les notations du chapitre précédent à savoir : pour f,g et h trois polynômes de $\mathbb{F}_4[X;\theta]$, on note $V(f),V(g)$ et $V(h)$ les ensembles des racines respectifs de f,g et $h\cup\{0\}$.

Lemme 4.3.1 *Avec les notations précédentes et si $f=gh$ et tel que le coefficient constant de f est $\neq 0$, on a*

$$R(g,h)\neq 0 \Leftrightarrow V(f)=V(g)\oplus V(h). \tag{4.3.6}$$

Preuve 4.3.2 *On a d'après la définition de la résultante, g et h n'ont pas de racines communes. Ainsi, $V(g)\cap V(h)=\{0\}$. D'un autre côté, et comme le coefficient constant de f est $\neq 0$, il en est de même pour g et h et on a donc par conséquent et compte tenu de 3.5.1* $\dim(V(g))=deg(g)$, $\dim(V(h))=deg(h)$ *et* $\dim(V(f))=deg(f)$. *Ainsi*

$$\dim(V(f))=\dim(V(g))+\dim(V(h)),$$

ce qui démontre le premier sens. Pour montrer le deuxième sens, nous considérons $(\alpha_1,...,\alpha_n)$ et $(\beta_1,...,\beta_m)$ les bases respectifs de $V(g)$ et $V(h)$. Ainsi, il suffit de remarquer que le fait que $V(f)=V(g)\oplus V(h)$ implique que $(\alpha_1,...,\alpha_n,\beta_1,...,\beta_m)$ est une base de $V(f)$. Il vient donc que $\Delta(\alpha_1,...,\alpha_n,\beta_1,...,\beta_m)\neq 0$, et par conséquent $R(g,h)\neq 0$.

Définition 4.3.2 *Soit $P=\sum_{i=0}^{n}a_iX^i\in\mathbb{F}_4[X,\theta]$. On appelle θ-adjoint de P le polynôme noté et défini par*

$$P_*=\sum_{i=0}^{n}\theta^{n-i}(a_i)\,X^{n-i}.$$

Remarque 4.3.1 *1) Notons que si θ est égal à l'identité, nous retrouvons $P_* = P^* = \sum_{i=0}^{n} a_i X^{n-i}$, le plynôme réciproque de P.*
2) On a compte tenu de 2.1.4 que si $P = X^n - 1 = gh$, alors $h_* = g^{\perp}$.

Nous utiliserons dans la suite les notations suivantes : soit $f \in \mathbb{F}_4[X, \theta]$ et $B = (\alpha_1, ..., \alpha_n)$ une base du $\mathbb{F}_2-$espace vectoriel $V(f)$. On note $\overline{B}$ l'ensemble défini par

$$\overline{B} = (\overline{\alpha_1}, ..., \overline{\alpha_n}),$$

où l'on a noté pour tout $1 \leq i \leq n$

$$\overline{\alpha_i} = \frac{\Delta(\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_n)}{\Delta},$$

et où Δ est donné par $\Delta = \Delta(\alpha_1, ..., \alpha_n)$.
De plus, pour une famille $\Gamma = \{x_1, ..., x_n\}$ d'un anneau quelconque $\mathbb{F}$, on notera Γ^2 la famille $\{x_1^2, ..., x_n^2\}$.

Remarque 4.3.2 *Puisque $(\alpha_1, ..., \alpha_n)$ est une base de $V(f)$, il est donc évident que $\Delta = \Delta(\alpha_1, ..., \alpha_n) \neq 0$.*

La proposition donnée ci-dessous est une nouvelle écriture des résultats de la section 1.7 de [27] dans le cas $q = 4$.

Proposition 4.3.3 *[27] Soit f un polynôme non nul, à coefficient constant non nul de $\mathbb{F}_4[X, \theta]$ et soit $B = (\alpha_1, ..., \alpha_n)$ une base du $\mathbb{F}_2-$espace vectoriel $V(f)$. La famille $\overline{B}^2$ est une $\mathbb{F}_2-$base de l'ensemble de racine du polynôme $\theta-$adjoint de f f_*.*

Nous sommes maintenant en mesure de donner une caractérisation d'un code $\theta-$cyclique auto-orthogonal à travers une caractérisation de son ensemble de racines.

Théorème 4.3.1 *Soient f, g et h trois polynômes de $\mathbb{F}_4[X, \theta]$ vérifiant*

$$f = X^n - 1 = hg.$$

Soit $C =< g >$ le code $\theta-$cyclique engendré par g et soit $B_0 = (\alpha_1, ..., \alpha_k)$ une base du $\mathbb{F}_2-$espace vectoriel $V(g)$. Alors, il existe une base B de $V(f)$ contenant B_0 et on a :
C est un code auto-orthogonal euclidien si et seulement si

$$\overline{\aleph}^2 \subset V(g),$$

où $\aleph$ est la famille définie par $\aleph = \{\beta_1 g(\beta_1), .., \beta_m g(\beta_m)\}$ et où les β_i sont les éléments de $B \setminus B_0$.

Preuve 4.3.4 *Il vient d'après 3.2.5 que $V(g) \subset V(f)$. Ainsi, comme B_0 est une famille libre, il existe donc une base B de $V(f)$ contenant B_0.*
On pose $B \setminus B_0 = \{\beta_1, ...\beta_m\}$. Il est clair que les β_i ne s'écrivent pas en combinaison linéaire des éléments de B_0 et par conséquent β_i n'appartient pas à $V(g)$ $\forall 1 \leq i \leq m$. On a ainsi compte tenu de 3.2.5 $\beta_i g(\beta_i)$ est une racine de h pour tout $1 \leq i \leq m$.
Montrons maintenant que la famille $\aleph = \{\beta_1 g(\beta_1), .., \beta_m g(\beta_m)\}$ est une base de $V(h)$. Supposons pour cela qu'elle est liée. Il existe donc $1 \leq p \leq m$ et $\alpha_1, ..., \alpha_{p-1}, \alpha_{p+1}, ..., \alpha_m \in \mathbb{F}_4$ tel que

$$\beta_k g(\beta_k) = \sum_{i=1; i \neq k}^{m} \alpha_i \beta_i g(\beta_i).$$

Le fait que $g(\beta_k) \neq 0$ implique que

$$\beta_k = \sum_{i=1; i \neq k}^{m} \alpha_i (g(\beta_k))^{-1} \beta_i g(\beta_i),$$

ce qui est absurde car $\{\beta_1, ...\beta_m\}$ est une famille libre comme sous famille d'une famille libre.
D'autre part, comme $f = X^n - 1 = gh$, on a compte tenu de 3.5.1

$$\dim(V(h)) = deg(h) = deg(f) - deg(g) = m,$$

et par conséquent $\aleph$ est une base de $V(h)$.
La proposition 4.3.3 implique donc que la famille $\overline{\aleph}^2$ est une base de l'ensemble de racines de $h_* = g^\perp$. Ainsi, si

$$\overline{\aleph}^2 \subset V(g),$$

on a, toute racine de $g^\perp$ est une racine de g. D'où $g^\perp$ divise g et par suite

$$< g >= C \subset < g^\perp >= C^\perp.$$

On a ainsi le premier sens. La réciproque est immédiate.

Pour donner une caractérisation semblable dans le cas d'un code auto-orthogonal hermitien nous considérons l'application suivante :

$$\Phi \quad : \quad \mathbb{F}_4[X; \theta] \mapsto \mathbb{F}_4[X; \theta], \tag{4.3.7}$$

$$\sum_{i=0}^{n} a_i X^i \mapsto \sum_{i=0}^{n} \theta(a_i) X^i. \tag{4.3.8}$$

Remarque 4.3.3 *1. Il est clair que l'application Φ est un morphisme d'anneaux.*
2. On a compte tenu de 2.1.5 : si $P = X^n - 1 = gh$, alors $\Phi(h_*) = g_H^\perp$.

Lemme 4.3.5 *Soit $P = \sum_{i=0}^{n} a_i X^i \in \mathbb{F}_4[X, \theta]$, k une extension de $\mathbb{F}_4$ et Θ l'extension de θ à k. Pour tout $a \in k$, on a :*
a est une racine de P si et seulement si $\Theta(a)$ est une racine de $\Phi(P)$.

Preuve 4.3.6 *Comme a est une racine de P, on a d'après 3.2.2*

$$\sum_{i=0}^{n} a_i N_i(a) = 0.$$

Il vient donc via 3.2.3

$$\begin{aligned}\sum_{i=0}^{n} \Theta(a_i) N_i(\Theta(a)) &= \sum_{i=0}^{n} \Theta(a_i)\Theta^i(a)\Theta^{i-1}(a)...\Theta(a),\\ &= \Theta\left(\sum_{i=0}^{n} a_i \Theta^{i-1}(a)...\Theta(a)\, a\right),\\ &= \Theta\left(\sum_{i=0}^{n} a_i N_i(a)\right) = \Theta(0) = 0.\end{aligned}$$

Ainsi $\Theta(a)$ est une racine de $\hat{P}$. La réciproque est immédiate étant donné que Θ est d'ordre 2.

Le résultat suivant est une caractérisation d'un code $\theta-$cyclique auto-orthogonal hermitien à travers une caractérisation de son ensemble de racines.

Théorème 4.3.2 *Soit f, g et h trois polynômes de $\mathbb{F}_4[X,\theta]$ vérifiant*

$$f = X^n - 1 = hg.$$

Soit $C =< g >$ le code $\theta-$cyclique engendré par g et soit $B_0 = (\alpha_1, ..., \alpha_n)$ une base du $\mathbb{F}_2-$espace vectoriel $V(g)$. Alors, il existe une base B de $V(f)$ contenant B_0 et on a :
C est un code auto-orthogonal hermitien si et seulement si

$$\Theta(\overline{\aleph}^2) \subset V(g),$$

où $\aleph$ est la famille définie par $\aleph = \{\beta_1 g(\beta_1), .., \beta_m g(\beta_m)\}$, les β_i sont les éléments de $B \setminus B_0$ et où Θ est l'extension de θ à $V(f)$.

Preuve 4.3.7 *On montre comme pour la preuve de 4.3.1 que la famille $\overline{\aleph}^2$ est une base de $h_* = g^{\perp}$. Comme Θ est un $\mathbb{F}_2$-automorphisme, l'image d'une base est une base. D'où, il vient compte tenu du lemme 4.3.5 que $\Theta(\overline{\aleph}^2)$ est une base de l'ensemble de racine de $\Phi(h_*) = g^{\perp_H}$. Ainsi, si*

$$\Theta(\overline{\aleph}^2) \subset V(g),$$

on a, toute racine de $g^{\perp_H}$ est une racine de g. D'où $g^{\perp_H}$ divise g et par suite

$$< g >= C \subset < g^{\perp_H} >= C^{\perp_H}.$$

On a ainsi le premier sens. La réciproque est immédiate.

4.4 Code auto-orthogonal comme sous code d'un code auto dual

La construction suivante est basée sur le fait qu'un code auto-orthogonal peut être considéré comme un sous code d'un code auto- dual. On a le résultat suivant :

Proposition 4.4.1 *Soit $\langle g\rangle$ un code θ-cyclique auto-dual hermitien de longueur n sur $\mathbb{F}_4$. Soit f un diviseur arbitraire de X^n-1 et soit $h = ppcm(f,g)$. Le code θ-cyclique de générateur h est un code auto-orthogonal pour le produit scalaire hermitien sur $\mathbb{F}_4$.*

Preuve 4.4.2 *$\langle g\rangle$ un code auto-dual hermitien sur $\mathbb{F}_4$. Comme g est un diviseur de h, on a*

$$\langle h\rangle \subset \langle g\rangle,$$

et par conséquent

$$\langle h\rangle \subset \langle g\rangle = \langle g\rangle^{\perp} \subset \langle h\rangle^{\perp}.$$

Ainsi le code $\langle h\rangle$ est un code auto-orthogonal pour le produit scalaire hermitien comme étant un sous code de l'auto-dual $\langle g\rangle$.

4.5 Construction d'un code quantique sur $\mathbb{F}_4$

Les résultats des sections précédentes vont nous permettre de construire des codes θ-cycliques auto-orthogonaux pour le produit scalaire trace sur $\mathbb{F}_4$. Ceci nous permettra par la suite de construire des codes quantiques sur $\mathbb{F}_4$ en utilisant la construction CSS [13]. Nous Rappelons en premier lieu ce théorème donné dans [13]

Théorème 4.5.1 *Un code C est auto-orthogonal pour le produit scalaire hermitien sur $\mathbb{F}_4$ si et seulement si c'est un code auto-orthogonal pour le produit scalaire trace sur $\mathbb{F}_4$.*

Les deux résultats suivants sont immédiats via le théorème 4.5.1 et les résultats des deux sections précédentes. On a via 4.3.2 et 4.5.1 et en utilisant les notations de 4.3.2

Corollaire 4.5.1 *Soit $C = \langle g\rangle$ le code $\theta-$cyclique engendré par g et soit $B_0 = (\alpha_1, ..., \alpha_k)$ une base du $\mathbb{F}_2-$espace vectoriel $V(g)$. Alors, il existe une base B de $V(X^n-1)$ contenant B_0 et on a : si*

$$\Theta(\overline{\aleph}^2) \subset V(g),$$

alors il existe un code quantique sur $\mathbb{F}_4$ de distance minimale celle de $(C^{\perp_{tr}} \setminus C)$.

On a via 4.5.1 et 4.4.1 le résultat suivant :

Corollaire 4.5.2 *Soit $\langle g\rangle$ un code θ-cyclique auto-dual hermitien de longueur n sur $\mathbb{F}_4$. Pour tout diviseur arbitraire f de X^n-1, il existe un code quantique sur $\mathbb{F}_4$ de distance minimale celle de $(C^{\perp_{tr}} \setminus C)$, où C est le code auto-ortogonal engendré par $ppcm(f,g)$.*

Preuve 4.5.3 *On a via 4.4.1, le code engendré par le ppcm de f et g est un code auto-orthogonal hermitien sur $\mathbb{F}_4$ et par conséquent c'est un code auto-orthogonal pour le produit scalaire trace où l'on a fait appel au théorème 4.5.1. La construction CSS [13] nous donne le code quantique voulu.*

4.5.1 Résultats numériques

Dans cette section nous considérons les générateurs des codes θ-cycliques auto-duaux pour le produit scalaire hermitien sur F_4 donnés dans [8] et nous construisons par magma des multiples de ces polynômes qui divisent $X^n - 1$ dans $\mathbb{F}_4[X;\theta]$. Ces derniers polynômes vont engendrer des codes θ-cycliques auto-orthogonaux comme étant des sous codes de codes θ-cycliques auto-duaux sur $\mathbb{F}_4$.Nous présentons dans la suite un algorithme permettant de trouver un générateur d'un code θ-cyclique auto-orthogonal sur $\mathbb{F}_4$ à partir d'un code θ-cyclique auto-dual. Une application de cet algorithme sur Magma (9.7) nous permettera de trouver les résultats numériques donné ci dessous.

Algorithme
1. On considère $P = X^n + 1$ et g un générateur d'un code θ-cyclique auto-dual sur $\mathbb{F}_4$.
2. On détermine les racines des polynômes p et g comme étant des polynômes tordus.
3. pour tout a racine de p on écrit

$$g = ppcm(g, X - a)$$

4. Si $g \in \mathbb{F}_4[X]$ alors g est un générateur d'un code θ-cyclique auto-orthogonal, sinon on répète l'opération.
5. On s'arrête quant le degré de g est égal à n.

Dans la suite nous présentons quelques exemples des meilleurs codes quantiques obtenus sur $\mathbb{F}_4$ via cette construction suite à une application du programme **1** de l'annexe sous Magma.

Exemple 4.5.4 *1. Pour $n = 4$, on considère le code $\theta-$cyclique auto-dual engendré par X^2+1 de meilleure distance minimale et donné dans [8]. Nous obtenons via ce programme le code $\theta-$cyclique auto-orthogonal sur $\mathbb{F}_4$ engendré par*

$$X^3 + X^2 + X + 1,$$

et par conséquent le code quantique sur F_4 de meilleur distance minimale et de paramètres [[4,2,2]].
2. Pour $n = 6$, on considère le code $\theta-$cyclique auto-dual engendré par $X^3+w^2X^2+wX+1$ de meilleur distance minimale et donné dans [8]. Nous obtenons le code $\theta-$cyclique auto-orthogonal sur $\mathbb{F}_4$ engendré par

$$X^4 + w^2X^3 + w^2X + 1$$

et d'où le code quantique sur F_4 de meilleure distance minimale et de paramètres [[6,2,2]].
3. Pour $n = 10$, le code $\theta-$cyclique auto-dual de meilleure distance minimale engendré par $X^5+X^4+w^2X^3+wX^2+X+1$ nous donne via cette construction un code $\theta-$cyclique auto-orthogonal sur $\mathbb{F}_4$ engendré par

$$X^6 + w^2X^4 + w^2X^2 + 1.$$

Ces résultats nous permettent de construire un code quantique sur F_4 de paramètres [[10,2,3]].
4. Pour $n = 12$, on considère le code $\theta-$cyclique auto-dual engendré par $X^6 + X^5 + wX^4 + wX^2 + X + 1$ de meilleure distance minimale et donné dans [8]. Nous obtenons via le programme précédent le code $\theta-$cyclique auto-orthogonal sur $\mathbb{F}_4$ engendré par

$$X^7 + wX^5 + wX^4 + w^2X^3 + w^2X^2 + 1,$$

et par conséquent le code quantique sur F_4 de meilleure distance minimale et de paramètres [[12,2,4]].
Nous obtenons aussi un code $\theta-$cyclique auto-orthogonal sur $\mathbb{F}_4$ engendré par

$$X^8 + X^7 + w^2X^6 + X^5 + X^3 + w^2X^2 + x + 1,$$

et par conséquent le code quantique sur F_4 de paramètres [[12,4,3]].

Deuxième partie

Nouvelles constructions des codes $\theta-$cycliques et quasi-θ-cycliques Applications à la construction des codes quantiques asymétriques

Chapitre 5

Des codes θ-cycliques aux codes quantiques asymétriques

5.1 Introduction

La motivation de ce chapitre est la capacité des codes quantiques asymétriques de corriger beaucoup plus d'erreurs phase que d'erreurs bit ou phase-bit. D'autant plus, des récentes expériences physiques ont montré qu'il existe une nette différence entre les probabilité d'erreurs phase, bit ou phase-bit. Pour une documentation complète sur ces codes et leurs constructions, le lecteur pourrait consulter [36] et [24]. D'autre part, les codes θ-cycliques sont désormais connus par leur grand nombre pour une longueur donnée et leurs bons paramètres ce qui encourage leur utilisation dans des nouvelles constructions dans le domaine de l'informatique quantique. Dans ce chapitre, nous présentons des nouvelles constructions de codes quantiques asymétriques en se basant sur la méthode donnée dans [23] et en introduisant une application S transformant les codes θ-cycliques de longueur n sur $\mathbb{F}_4$ en des codes additifs de longueur $2n$ invariants par une permutation σ. La permutation σ est un cycle si n est impair et un produit de deux cycles si n est pair.
Ainsi, le but de ce chapitre serait de révéler la relation entre les codes θ-cycliques et les codes additifs cycliques et quasi-cycliques - suivant la parité du code choisi- et en passant par l'application S. Les propriétés intéressantes de dualité et de conservation de distance de cette application vont nous permettre d'établir leur relation avec les codes quantiques asymétriques. Signalons enfin que compte tenu de la proposition 2.1.11, l'étude du cas de la longueur impair pourrait se montrer inutile.
Ce chapitre est organisé comme suit : dans un premier temps nous introduisons et étudions les propriétés de l'application S ainsi qu'une étude des images des codes θ-cycliques par cette application. Nous présentons dans la suite une étude du poids de distribution de l'image d'un code θ-cyclique C par l'application S en relation avec son poids de distribution initial. Cette étude est d'autant plus importante qu'elle nous permettra de comprendre les paramètres du code quantique asymétrique obtenu via l'application S. Enfin, deux constructions de codes quantiques asymétriques seront présentées à la fin de ce chapitre.

Signalons enfin que ce chapitre a fait l'objet d'un article publié [21].

5.2 Généralités

Définition 5.2.1 *Soit $\mathbb{F}_4 := \{0, 1, \omega, \omega^2 = \overline{\omega}\}$, le corps fini à 4 éléments. On note pour $x \in \mathbb{F}_4$, $\overline{x} = x^2$, le conjugué de x. Soit n un entier positif ≥ 1, $\mathbf{u} = (u_0, u_1, \ldots, u_{n-1})$ et $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ deux éléments de $\mathbb{F}_4^n$. On utilisera dans la suite les notations suivantes :*

1. *4^{H} la famille des codes linéaires sur $\mathbb{F}_4$ de longueur n et muni du produit scalaire hermitien défini par*
$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathrm{H}} := \sum_{i=0}^{n-1} u_i \cdot v_i^2. \tag{5.2.1}$$

2. *$4^{\mathrm{H}+}$ la famille des codes additifs ou $\mathbb{F}_2$-linéaires sur $\mathbb{F}_4$ de longueur n et muni du produit scalaire trace défini par*
$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathrm{tr}} := \sum_{i=0}^{n-1} (u_i \cdot v_i^2 + u_i^2 . \cdot v_i). \tag{5.2.2}$$

Rappelons que pour un code quelconque C et pour un produit scalaire choisi $*$, le dual $C^{\perp *}$ de C est défini comme d'habitude par :

$$C^{\perp *} := \{\mathbf{u} \in \mathbb{F}_q^n : \langle \mathbf{u}, \mathbf{v} \rangle_* = 0 \text{ pour tout } \mathbf{v} \in C\}.$$

Signalons que la motivation de l'étude des codes de la famille $4^{\mathrm{H}+}$ est son application dans la construction des codes stabilisateurs [38] et des codes quantiques [31, Sec. 9.10]. On sait déjà que pour un $(n, 2^k)$-code additif C sur $\mathbb{F}_4$, $C^{\perp_{\mathrm{tr}}}$ est un code additif de paramètres $(n, 2^{2n-k})$. Notons en plus qu'il vient d'après [13] que si C est un code linéaire sur $\mathbb{F}_4$ de paramètres $[n, k, d]_4$, alors

$$C^{\perp_{\mathrm{H}}} = C^{\perp_{\mathrm{tr}}},$$

compte tenu du fait que $C^{\perp_{\mathrm{H}}} \subseteq C^{\perp_{\mathrm{tr}}}$ et que le cardinal de $C^{\perp_{\mathrm{H}}}$ est égal à $4^{n-k} = 2^{2n-2k}$ qui est en fait le cardinal de $C^{\perp_{\mathrm{tr}}}$. Rappelons enfin [50, Sec. 2.3] que les familles 4^{H} et $4^{\mathrm{H}+}$ vérifient l'équation MacWilliams. On a donc pour tout code additif C sur $\mathbb{F}_4$

$$W_{C^{\perp_{\mathrm{tr}}}}(X, Y) = \frac{1}{|C|} W_C(X + 3Y, X - Y), \tag{5.2.3}$$

où $W_C(X, Y)$ est le polynôme énumérateur du code C donné par 2.1.1.

5.3 L'application S sur les codes de $\mathbb{F}_4$

Définition 5.3.1 *On définit dans $\mathbb{F}_4^n$ l'application :*

$$\begin{array}{rccc} S: & \mathbb{F}_4^n & \mapsto & \mathbb{F}_4^{2n} \\ & (x_1, x_2, ..., x_n) & \mapsto & (x_1, \overline{x_1}, x_2, \overline{x_2}..., x_n, \overline{x_n}). \end{array}$$

Lemme 5.3.1 *L'application S est une application $\mathbb{F}_2$ linéaire, injective non surjective.*

Exemple 5.3.2 *Illustrons par un exemple le fait que l'application S n'est pas $\mathbb{F}_4$ linéaire. Considérons pour cela $n = 2$ et $u = (w, w^2)$, on a*
$S(u) = (w, w^2, w^2, w)$ et $S(w.u) = S((w^2, 1)) = (w^2, w, 1, 1) \neq w.S(u) = (w^2, 1, 1, w^2)$.

On considère dans la suite un code linéaire C sur $\mathbb{F}_4$ de longueur n, de dimension k et de distance minimale d.

Lemme 5.3.3 *Pour tout $u \in C$, on a*

$$W_H(S(u)) = 2W_H(u),$$

et

$$d(S(C)) = 2d(C).$$

Lemme 5.3.4 *$S(C)$ est un code additif sur $\mathbb{F}_4$ de longueur $2n$ de cardinal $= card(C) = 4^k = 2^{2k}$ et de distance minimale $2d$.*

Remarque 5.3.1 *L'application S envoie un (n, M, d)-code additif C sur $\mathbb{F}_4$ en un code additif $S(C)$ sur $\mathbb{F}_4$ de paramètres $(2n, M, 2d)$. Par le théorème du rang appliqué à l'application S comme étant $\mathbb{F}_2$ linéaire, on obtient que*

$$dim_{\mathbb{F}_2} S(C) = dim_{\mathbb{F}_2} C = 2k.$$

Lemme 5.3.5 *Si C est un code cyclique additif de longueur n sur $\mathbb{F}_4$, alors $S(C)$ est un code quasi-cyclique additif sur $\mathbb{F}_4$ de longueur $2n$ et d'index 2.*

Preuve 5.3.6 *Comme C est cyclique,*

$$\mathbf{v} = (v_0, \ldots, v_{n-2}, v_{n-1}) \in C \text{ si et seulement si } \mathbf{v}' = (v_{n-1}, v_0, \ldots, v_{n-2}) \in C.$$

En appliquant S on obtient

$$\begin{aligned} S(\mathbf{v}) &= (v_0, \overline{v_0}, \ldots, v_{n-2}, \overline{v_{n-2}}, v_{n-1}, \overline{v_{n-1}}) \in S(C), \\ S(\mathbf{v}') &= (v_{n-1}, \overline{v_{n-1}}, v_0, \overline{v_0}, \ldots, v_{n-2}, \overline{v_{n-2}}) \in S(C). \end{aligned}$$

Ainsi, $S(C)$ est un code additif 2-quasi-cyclique.

Lemme 5.3.7 *Pour tout $u, v \in C$, on a*

$$< u, v >_H = 0 \Rightarrow < S(u), S(v) >_{tr} = 0.$$

Proposition 5.3.8 *Soit un $(n, M, d)_4$- code additif C. On a, $S(C) \subseteq S(C)^{\perp_{\mathrm{tr}}}$.*

Preuve 5.3.9 *Soient $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}), \mathbf{u} = (u_0, u_1, \ldots, u_{n-1}) \in C$. Alors*

$$\langle S(\mathbf{v}), S(\mathbf{u}) \rangle_{\mathrm{tr}} = \sum_{i=0}^{n-1} (v_i \overline{u_i} + \overline{v_i} u_i) + \sum_{i=0}^{n-1} (\overline{v_i} u_i + v_i \overline{u_i}) = 2 \sum_{i=0}^{n-1} (v_i \overline{u_i} + \overline{v_i} u_i) = 0.$$

Remarque 5.3.2 *1. On a d'après [31] $(S(C))^{\perp_{tr}}$ est un code additif $(2n, 2^{4n-2k})$.*
2.Si C est auto dual pour le produit scalaire hermitien alors ceci n'implique pas que $S(C)$ soit auto dual pour les produits scalaires hermitien et trace. En effet si C est auto dual alors n est pair et $dim_{\mathbb{F}_4} C = \frac{n}{2}$. Ainsi $S(C)$ est un code additif $(2n, 2^n)$ et $(S(C))^{\perp_{tr}}$ est un code additif $(2n, 2^{4n-n}) = (2n, 2^{3n})$.
3. Soit C un code linéaire de $dim_{\mathbb{F}_4} = k$. Pour que $S(C)$ soit auto-dual, il faut que $k = 4n - k$. Ainsi le seul code C dont l'image par S est un code auto-dual est le code trivial $\mathbb{F}_4^n$.

5.4 Application aux codes θ-cycliques

Dans cette section nous considérons l'ensemble des polynômes tordus

$$\Re = \mathbb{F}_4[X, \theta] = \{a_0 + a_1 X + a_2 X^2 + ... + a_n X^n \;\; a_i \in \mathbb{F}_4\}$$

comme un $\mathbb{F}_4$ module à gauche. Dans [10] Boucher et Ulmer ont étudié les codes θ-cycliques comme des sous modules de $\Re_n = \Re_{/X^n - 1}$. Ils ont donné la définition suivante :

Définition 5.4.1 *Un sous ensemble C de $\mathbb{F}_4^n$ est appelé un code θ-cyclique de longueur n si et seulement si C est un $\Re-$sous module de $\Re_n$.*

On a donc C est un code θ-cyclique de $\mathbb{F}_4^n$ si et seulement si C est invariant par le $\theta-$shift T_θ c'est à dire :
si

$$c = (c_0, c_1, ..., c_{n-1}) \in C$$

alors

$$T_\theta(c) = (\theta(c_{n-1}), \theta(c_0), ..., \theta(c_{n-2})) \in C.$$

Dans la suite on considère l'automorphisme de Frobenuis défini dans $\mathbb{F}_4$ par $\theta(x) = x^2 = \overline{x}$ et C un code θ-cyclique de $\mathbb{F}_4^n$.

On note $[1..2n]$ l'ensemble $\{1, 2, \ldots, 2n\}$ et $\sigma = \tau \circ T^2$ la permutation dans $[1..2n]$ où T est le shift modulo $2n$ et $\tau = (12)(34)\ldots(2n-1, 2n)$. Comme T^2 et τ commutent, σ peut être vu comme $T^2 \circ \tau$. On dénote la permutation identité par (1).

Soit Σ la permutation sur les éléments de $\mathbb{F}_4^{2n}$ induite par σ. Ainsi, pour $\mathbf{x} = (x_1, x_2, \ldots, x_{2n}) \in \mathbb{F}_4^{2n}$, on a

$$\Sigma(\mathbf{x}) = \left(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(2n)}\right). \tag{5.4.4}$$

Lemme 5.4.1 *Soit C un code θ-cyclique de $\mathbb{F}_4^n$. On a, $S(C)$ est stable par la permutation Σ.*

Preuve 5.4.2 *Soit $\mathbf{v} = (v_1, v_2, \ldots, v_{2n}) \in S(C)$. Il existe donc $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in C$ tel que*

$$\mathbf{v} = (u_1, \overline{u_1}, u_2, \overline{u_2}, \ldots, u_n, \overline{u_n}) = S(\mathbf{u}).$$

Comme C est un code $\theta-$cyclique, on a

$$\overline{\mathbf{u}} := (\overline{u_n}, \overline{u_1}, \ldots, \overline{u_{n-1}}) \in C.$$

D'où,

$$\begin{aligned}\Sigma(\mathbf{v}) &= (\overline{u_n}, u_n, \overline{u_1}, u_1, \ldots, \overline{u_{n-1}}, u_{n-1}),\\ &= S((\overline{u_n}, \overline{u_1}, \ldots, \overline{u_{n-1}})),\end{aligned}$$

ce qui implique que $\Sigma(S(C)) \subset S(C)$.

Lemme 5.4.3 *L'ordre de σ est égal à $2n$ si n est impair et à n si n est pair.*

Preuve 5.4.4 *La permutation σ est explicitement défini par*

$$\sigma : i \mapsto \begin{cases} i+3 \pmod{2n} \text{ si } i \text{ est impair,} \\ i+1 \pmod{2n} \text{ si } i \text{ est pair}, \end{cases} . \tag{5.4.5}$$

où $1 \leq i \leq 2n$.
En calculant modulo $2n$, on remarque que si i est impair, alors $\sigma(i) = i+3$ et $\sigma^2(i) = \sigma(i+3) = i+4$. Si i est pair, alors $\sigma(i) = i+1$ et $\sigma^2(i) = \sigma(i+1) = i+4$. D'où, $\sigma^2 = T^4$. Ainsi si $n = 2p$ pour un entier positif p, on a

$$\sigma^n = \sigma^{2p} = T^{4p} = T^{2i} = (1).$$

De plus, pour tout $1 \leq k < n$, on a si $k = 2i$, alors

$$\sigma^k = \sigma^{2i} = T^{4i} \neq (1),$$

puisque $4i = 2k < 2n$.
Si $k = 2i+1$ alors

$$\sigma^k = \sigma^{2i+1} = T^{4i} \circ \tau \neq (1),$$

car $\sigma^k(1) = 4i+1 \neq 1$.

Par conséquent l'ordre de σ est n. Dans le cas ou n est impair on a

$$\sigma^{2n} = T^{4n} = id.$$

Pour montrer que $2n$ est l'entier minimum vérifiant cette propriété, on remarque d'abord que $\sigma^n = \tau$ compte tenu du fait

$$\sigma^n = \sigma^{2l+1} = T^{4l} \circ \sigma = T^{2n-2} \circ (\tau \circ T^2) = T^{2n-2} \circ (T^2 \circ \tau) = \tau.$$

D'autre part, on montre -comme pour le cas n pair- que pour $1 \leq k < n$, $\sigma^k \neq (1)$. Considérons maintenant le cas $n+1 \leq k < 2n$, on a

$$\sigma^k = \sigma^n \circ \sigma^{k-n} = \tau \circ \sigma^{k-n} \neq (1).$$

Ainsi l'ordre de σ est $2n$.

Remarque 5.4.1 *1. Si n est impair σ est un cycle de longueur $2n$. On note*

$$\sigma = (1, \sigma(1), \sigma^2(1), ..., \sigma^{2n-2}(1), \sigma^{2n-1}(1)). \tag{5.4.6}$$

2. Si n est pair, σ est un produit de deux cycles de longueur n chacun. En remarquant que $\sigma^k(1) \neq 2$ pour tout $0 \leq k \leq n-1$, on a

$$\sigma = (1, \sigma(1), \sigma^2(1), ..., \sigma^{n-1}(1))(2, \sigma(2), \sigma^2(2), ..., \sigma^{n-1}(2)). \tag{5.4.7}$$

Proposition 5.4.5 *Soit C un $[n,k,d]$-code θ-cyclique de $\mathbb{F}_4^n$.*
1. Si n est impair alors $S(C)$ est équivalent à un $(2n, 2^{2k}, 2d)$-code cyclique additif de $\mathbb{F}_4^{2n}$.
2. Si n est pair alors $S(C)$ est équivalent à un $(2n, 2^{2k}, 2d)$-code 2-quasi-cyclique additif de $\mathbb{F}_4^{2n}$.

Preuve 5.4.6 *1. Considérons d'abord le cas où n est impair, on a donc 5.4.6.*
On définit la permutation σ' par

$$\sigma' = \begin{pmatrix} 1 & 2 & .. & .. & 2n-1 & 2n \\ \sigma^{2n-1}(1) & \sigma^{2n-2}(1) & .. & .. & \sigma(1) & 1 \end{pmatrix}. \tag{5.4.8}$$

On remarque facilement que pour tout $1 \leq j \leq 2n$

$$\sigma'(j) = \sigma^{2n-j}(1). \tag{5.4.9}$$

Soit Σ' la permutation sur les éléments de $\mathbb{F}_4^{2n}$ induite par σ'. Pour $\mathbf{x} = (x_1, x_2, \ldots, x_{2n}) \in \mathbb{F}_4^{2n}$, on a

$$\Sigma'(\mathbf{x}) = \left(x_{\sigma'(1)}, x_{\sigma'(2)}, \ldots, x_{\sigma'(2n)}\right). \tag{5.4.10}$$

Pour montrer que $\Sigma'(S(C))$ est cyclique il suffit de montrer que pour chaque Y dans $S(C)$

$$T(\Sigma'(Y)) \in \Sigma'(S(C)).$$

Ainsi, en utilisant le fait que $S(C)$ est stable par Σ, il nous suffit de montrer que

$$T(\Sigma'(Y)) = \Sigma'(\Sigma(Y)). \tag{5.4.11}$$

On a d'après la définition de Σ

$$\begin{aligned} \Sigma(Y) &= (y_{\sigma(1)}, y_{\sigma(2)}, ..., y_{\sigma(2n)}), & (5.4.12)\\ &= (y'_1, y'_2, ..., y'_{2n}), & (5.4.13) \end{aligned}$$

où l'on a noté

$$y'_i = y_{\sigma(i)}. \tag{5.4.14}$$

Ainsi,

$$\begin{aligned} \Sigma'(\Sigma(Y)) &= (y'_{\sigma'(1)}, y'_{\sigma'(2)}, ..., y'_{\sigma'(2n)})\\ &= (y'_{\sigma^{2n-1}(1)}, y'_{\sigma^{2n-2}(1)}, ...y'_{\sigma^1(1)}, y'_{\sigma^0(1)}), \end{aligned}$$

où l'on a fait appel à 5.4.9.
Il vient donc en utilisant 5.4.14 que

$$\Sigma'(\Sigma(Y)) = (y_{\sigma^{2n}(1)}, y_{\sigma^{2n-1}(1)}, ...y_{\sigma^2(1)}, y_{\sigma^1(1)}). \tag{5.4.15}$$

D'autre part, on a d'après 5.4.9

$$\begin{aligned} \Sigma'(Y) &= (y_{\sigma'(1)}, y_{\sigma'(2)}, ..., y_{\sigma'(2n)})\\ &= (y_{\sigma^{2n-1}(1)}, y_{\sigma^{2n-2}(1)}, ...y_{\sigma^1(1)}, y_{\sigma^0(1)}). \end{aligned}$$

On a donc 5.4.11 compte tenu de 5.4.15.
Considérons maintenant le cas où n est pair et donc, on a 5.4.7.
Définissons dans la suite la permutation σ'' par

$$\sigma'' = \begin{pmatrix} 1 & 2 & 3 & 4 & .. & .. & 2n-1 & 2n \\ \sigma^{n-1}(1) & \sigma^{n-1}(2) & \sigma^{n-2}(1) & \sigma^{n-2}(2) & .. & .. & \sigma^0(1) & \sigma^0(2) \end{pmatrix}. \tag{5.4.16}$$

On a explicitement pour tout $1 \leq j \leq 2n$

$$\begin{aligned} \sigma''(j) &= \sigma^{n-p}(1) \quad si\ j = 2p-1\ , & (5.4.17)\\ &= \sigma^{n-p}(2) \quad si\ j = 2p\ . & (5.4.18) \end{aligned}$$

Soit Σ'' la permutation sur les éléments de $\mathbb{F}_4^{2n}$ induite par σ''. Pour $\mathbf{x} = (x_1, x_2, \ldots, x_{2n}) \in \mathbb{F}_4^{2n}$, on a

$$\Sigma''(\mathbf{x}) = \left(x_{\sigma''(1)}, x_{\sigma''(2)}, \ldots, x_{\sigma''(2n)}\right). \tag{5.4.19}$$

Soit Y dans $S(C)$, on a comme dans 5.4.12

$$\Sigma(Y) = (y'_1, y'_2, ..., y'_{2n}),$$

où l'on a utilisé la notation 5.4.14.
D'autre part

$$\begin{aligned}\Sigma''(\sigma(Y)) &= (y'_{\sigma''(1)}, y'_{\sigma''(2)}, ..., y'_{\sigma''(2n)}),\\ &= (y'_{\sigma^{n-1}(1)}, y'_{\sigma^{n-1}(2)}, y'_{\sigma^{n-2}(1)}, y'_{\sigma^{n-2}(2)}, ...y'_{\sigma^1(1)}, y'_{\sigma^1(2)}, y'_{\sigma^0(1)}, y'_{\sigma^0(2)}),\end{aligned}$$

où l'on a fait appel à 5.4.17.
Il vient donc de 5.4.14 que

$$\begin{aligned}\Sigma''(\Sigma(Y)) &= (y_{\Sigma^n(1)}, y_{\sigma^n(2)}, y_{\sigma^{n-1}(1)}, y_{\sigma^{n-1}(2)}, ...y_{\sigma^2(1)}, y_{\sigma^2(2)}, y_{\sigma^1(1)}, y_{\sigma^1(2)}),\\ &= (y_1, y_2, y_{\sigma^{n-1}(1)}, y_{\sigma^{n-1}(2)}, ...y_{\sigma^2(1)}, y_{\sigma^2(2)}, y_{\sigma(1)}, y_{\sigma(2)}).\end{aligned}$$

Or

$$\begin{aligned}\Sigma''(Y) &= (y_{\sigma''(1)}, y_{\sigma''(2)}, ..., y_{\sigma''(2n)}),\\ &= (y_{\sigma^{n-1}(1)}, y_{\sigma^{n-1}(2)}, y_{\sigma^{n-2}(1)}, y_{\sigma^{n-2}(2)}, ...y_{\sigma^1(1)}, y_{\sigma^1(2)}, y_1, y_2).\end{aligned}$$

Ainsi, on a

$$T^2(\Sigma''(Y)) = \Sigma''(\Sigma(Y)) \in \Sigma''(S(C)),$$

étant donné que $S(C)$ est stable par Σ.

Exemple 5.4.7 *Pour $n = 4$, on a*

$$\sigma = (1,4,5,8)(2,3,6,7) \text{ et}$$
$$\sigma'' = (1,8,2,7)(3,5,4,6).$$

Soit C le $[4,2,3]_4$-code θ-cyclique donné dans l'exemple 2 de [12] de matrice génératrice

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}. \tag{5.4.20}$$

On vérifie immédiatement que $S(C)$ est invariant par Σ.
Soit $\mathbf{u} = (1,0,1,0) \in C$. On pose $\mathbf{v} = S(\mathbf{u}) = (1,1,0,0,1,1,0,0)$. On a

$$\begin{aligned}\Sigma''\left(\Sigma(\mathbf{v})\right) &= \left(v_{\sigma^4(1)}, v_{\sigma^4(2)}, v_{\sigma^3(1)}, v_{\sigma^3(2)}, v_{\sigma^2(1)}, v_{\sigma^2(2)}, v_{\sigma(1)}, v_{\sigma(2)}\right),\\ &= (v_1, v_2, v_8, v_7, v_5, v_6, v_4, v_3) = (1,1,0,0,1,1,0,0).\end{aligned}$$

D'un autre côté

$$\begin{aligned}\Sigma''(\mathbf{v}) &= \left(v_{\sigma^3(1)}, v_{\sigma^3(2)}, v_{\sigma^2(1)}, v_{\sigma^2(2)}, v_{\sigma(1)}, v_{\sigma(2)}, v_1, v_2\right),\\ &= (v_8, v_7, v_5, v_6, v_4, v_3, v_1, v_2) = (1, \overline{w}, w, 0, 0, 1, 1).\end{aligned}$$

Et

$$\begin{aligned}\Sigma''\left(\Sigma(\mathbf{v})\right) &= \left(v_{\sigma^4(1)}, v_{\sigma^4(2)}, v_{\sigma^3(1)}, v_{\sigma^3(2)}, v_{\sigma^2(1)}, v_{\sigma^2(2)}, v_{\sigma(1)}, v_{\sigma(2)}\right),\\ &= (v_8, v_7, v_5, v_6, v_4, v_3, v_1, v_2) = (1,1,0,0,1,1,0,0).\end{aligned}$$

5.5 Analyse du poids de distribution

Nous présentons dans cette section une étude des polynômes énumérateurs de $S(C)$ et $S(C)^{\perp_{\mathrm{tr}}}$ et des relations qui existent entre ces deux polynômes. Cette analyse nous sera utile pour la détermination du paramètre d_x des codes quantiques asymétriques dans les constructions que nous présenterons dans la section suivante.

Soit A_i le nombre de mots de codes de poids i dans un $(n, M, d)_4$-code additif C. On a compte tenu de 5.3.3

$$W_{S(C)}(X, Y) = \sum_{i=0}^{n} A_i X^{2(n-i)} Y^{2i}. \tag{5.5.21}$$

De plus l'équation (5.2.3) nous permet d'écrire le polynôme énumérateur de $S(C)^{\perp_{\mathrm{tr}}}$ en fonction du polynôme énumérateur de $S(C)$ à savoir

$$W_{S(C)^{\perp_{\mathrm{tr}}}}(X, Y) = \frac{1}{|S(C)|} W_{S(C)}(X + 3Y, X - Y). \tag{5.5.22}$$

Plus explicitement, on a

$$W_{S(C)^{\perp_{\mathrm{tr}}}}(X, Y) = \frac{1}{M} \sum_{i=0}^{n} A_i L_i, \tag{5.5.23}$$

où L_i est défini par

$$\left(\sum_{j=0}^{n-i} \binom{n-i}{j} X^{n-i-j} (3Y)^j \right)^2 \left(\sum_{l=0}^{i} \binom{i}{l} X^{i-l} (-Y)^l \right)^2. \tag{5.5.24}$$

Notons le nombre de mots de codes de poids i dans le code $C^{\perp_{\mathrm{tr}}}$ par $A_i^{\perp_{\mathrm{tr}}}$. En utilisant *Pless power moments* avec $q = 4$ [31, p. 259], on a

$$\sum_{i=0}^{n} A_i = |C| = M, \tag{5.5.25}$$

$$\sum_{i=0}^{n} i A_i = \frac{M}{4} (3n - A_1^{\perp_{\mathrm{tr}}}), \tag{5.5.26}$$

$$\sum_{i=0}^{n} i^2 A_i = \frac{M}{4^2} \left\{ (9n^2 + 3n) - (6n - 2) A_1^{\perp_{\mathrm{tr}}} + 2 A_2^{\perp_{\mathrm{tr}}} \right\}. \tag{5.5.27}$$

En supposant $A_1^{\perp_{\mathrm{tr}}} = A_2^{\perp_{\mathrm{tr}}} = 0$, on a d'après (5.5.22) les résultats suivants :

1. le coefficient de $Y^0 X^{2n}$ est $\frac{1}{M} \sum_{i=0}^{n} A_i = 1$.

2. le coefficient de YX^{2n-1} est

$$\frac{1}{M}\sum_{i=0}^{n} A_i\left(2\cdot(n-i)\cdot 3-2i\right),$$
$$=\frac{1}{M}\sum_{i=0}^{n} A_i(6n-8i)=6n-4^{-1}\cdot 8(3n)=0,$$

où l'on a fait appel à l'équation (5.5.26).

3. Le coefficient de Y^2X^{2n-2} est

$$\frac{1}{M}\sum_{i=0}^{n} A_i\left(18n^2-48ni+32i^2-9n+8i\right),$$
$$=\frac{18n^2-9n}{M}\sum_{i=0}^{n} A_i+\frac{8-48n}{M}\sum_{i=0}^{n} iA_i+\frac{32}{M}\sum_{i=0}^{n} i^2A_i,$$
$$=3n,$$

où l'on a fait appel à (5.5.26) et (5.5.27).

En résumé, on a pour tout code additif C sur $\mathbb{F}_4$ de paramètres (n, M, d) et en utilisant la notation suivante :

$$W_{S(C)^{\perp_{\mathrm{tr}}}}(X,Y)=\sum_{i=0}^{2n} B_iX^{2n-i}Y^i. \tag{5.5.28}$$

* Si $d(C^{\perp_{\mathrm{tr}}})\geq 3$,alors $B_0=1, B_1=0$, et $B_2=3n$.
* Si $d(C^{\perp_{\mathrm{tr}}})=1$, alors $B_1=2A_1^{\perp_{\mathrm{tr}}}>0$.
* Si $d(C^{\perp_{\mathrm{tr}}})=2$, alors $B_1=0$ et $B_2=3n+4A_2^{\perp_{\mathrm{tr}}}>0$.

5.6 Codes quantiques asymétriques et application S

Avec les notations données dans le chapitre Généralités, nous commençons cette partie en présentant quelques résultats généraux sur les codes quantiques asymétriques et leurs constructions. Les références [36], [24] et [23] donneront une idée plus complète sur les codes quantiques asymétriques. Pour q et n deux entiers naturels, on notera dans la suite V_n le produit tensoriel $\mathbb{C}^{q^n}$ et nous rappelons la définition suivante :

Définition 5.6.1 *Soient d_x et d_z deux entiers naturels . Un code quantique Q de V_n de dimension $K\geq 2$ est appelé un code quantique asymétrique de paramètres $((n,K,d_z/d_x))_q$ ou $[[n,k,d_z/d_x]]_q$ et où $k=\log_q K$, si et seulement si Q détecte simultanément d_x-1 erreurs bit et d_z-1 erreurs phase.*

Le résultat suivant a été démontré dernièrement dans [23].

Proposition 5.6.1 *[23, Th. 4.5] Soit $q = p^2$ une puissance paire d'un entier premier p. Pour $i = 1, 2$, on considère C_i un code additif de paramètres (n, K_i, d_i) sur $\mathbb{F}_q$. Si $C_1^{\perp_{\mathrm{tr}}} \subseteq C_2$, alors il existe un code quantique asymétrique Q avec les paramètres $((n, \frac{|C_2|}{|C_1^{\perp_{\mathrm{tr}}}|}, d_z/d_x))_q$ où $\{d_z, d_x\} = \{d_1, d_2\}$.*

L'application S et l'analyse du polynôme énumérateur donnée dans la section précédente vont nous permettre de donner des nouvelles constructions de codes quantiques asymétriques. Il vient donc d'après 5.3.8 et l'analyse du polynôme énumérateur le résultat suivant :

Proposition 5.6.2 *Soit un code additif C sur $\mathbb{F}_4$ de paramètres (n, M, d) et vérifiant $d(C^{\perp_{\mathrm{tr}}}) \geq 2$. Il existe alors un code quantique asymétrique Q avec les paramètres $[[2n, \log_4\left(\frac{|S(C)^{\perp_{\mathrm{tr}}}|}{|S(C)|}\right), 2/2]]$.*

Preuve 5.6.3 *On a d'après proposition 5.3.8, $S(C) \subseteq S(C)^{\perp_{\mathrm{tr}}}$. On prend $C_1 = C_2 = S(C)^{\perp_{\mathrm{tr}}}$ et on applique le théorème 5.6.1. Il vient d'après l'analyse du polynôme énumérateur que $d_z = d_x = 2$.*

Signalons que les paramètres du code construit via la méthode donnée dans la proposition 5.6.2 ne sont pas d'excellents paramètres. Afin de les améliorer et de construire un nouveau code quantique avec des paramètres meilleurs, nous avons utilisé le fait que l'application S conserve l'inclusion et nous avons montré le résultat suivant :

Proposition 5.6.4 *Soit C un $(n, M_1, d_1)_4$-code additif tel que $d(C^{\perp_{\mathrm{tr}}}) \geq 2$ et D un $(n, M_2, d_2)_4$-code additif tel que $C \subseteq D$. Il existe alors un code quantique asymétrique Q avec les paramètres $[[2n, \log_4\left(\frac{M_2}{M_1}\right), 2d_2/2]]_4$.*

Preuve 5.6.5 *Dans le théorème 5.6.1 nous prenons $C_1 = S(C)^{\perp_{\mathrm{tr}}}$ et $C_2 = S(D)$. Le code $S(C)$ est un $(2n, M_1, 2d_1)_4$-code additif. De même, $S(D)$ est un code additif de paramètres $(2n, M_2, 2d_2)_4$. Les valeurs de d_z et de d_x résultent de l'analyse du polynôme énumérateur donné auparavant.*

Exemple 5.6.6 *On prend $C = D$ le $[n, 1, n]_4$-code de répétition engendré par le vecteur $\mathbf{1} = (1, \ldots, 1)$. On peut vérifier que $d(C^{\perp_{\mathrm{tr}}}) = 2$. D'où, et en utilisant le théorème 5.6.4, on obtient un code quantique asymétrique Q avec les paramètres $[[2n, 0, 2n/2]]_4$. Le code Q satisfait l'inégalité de la borne du singleton pour les codes quantiques à savoir $k \leq n - d_x - d_z + 2$.*

On prendra en compte dans la suite la définition suivante

Définition 5.6.2 *On appelle code quantique asymétrique MDS tout code quantique asymétrique Q satisfaisant la condition $k = n - d_x - d_z + 2$.*

Exemple 5.6.7 *Considérons le $[4, 2, 3]_4$-code θ-cyclique D de matrice génératrice G donné auparavant (5.4.20). Ce code contient le code de répétition C $[4, 1, 4]$ engendré par $\mathbf{1}$. En appliquant le théorème 5.6.1 avec $C = C_1^{\perp_{\mathrm{tr}}}$ et $D = C_2$, nous obtenons un $[[4, 1, 3/2]]_4$-code quantique asymétrique . D'un autre côté et en utilisant l'application S, et le théorème 5.6.4, nous obtenons le code quantique asymétrique de paramètres $[[8, 1, 6/2]]$ sur $\mathbb{F}_4$*

5.7 Les Constructions

Dans cette section, nous présentons deux constructions de codes quantiques asymétriques avec $d_z \geq d_x = 2$: l'une en utilisant la base de données BKLC et l'autre en appliquant l'application S sur les codes de Reed Solomon concaténés. Dans le but d'obtenir un code quantique optimal via l'application S nous allons essayer de baser notre construction sur les deux choix suivants :
1. le choix d'un code D de taille et de distance minimum relativement larges.
2. le choix du plus petit sous code possible C de D vérifiant $d(C^{\perp_{\mathrm{tr}}}) \geq 2$.

Notons qu'il n'existe pas de $(n, 2, d)_4$-code additif avec $d(C^{\perp_{\mathrm{tr}}}) \geq 2$. Le plus petit code additif avec $d(C^{\perp_{\mathrm{tr}}}) = 2$ est un $(n, 4, n) = [n, 1, n]$-code C sur $\mathbb{F}_4$ engendré par un mot de code $\mathbf{v}$ de poids n. Il vient du fait que C est un code MDS que, les paramètres de son dual $C^{\perp_{\mathrm{tr}}} = C^{\perp_{\mathrm{H}}}$ sont $[n, n-1, 2]$.

5.7.1 Construction à partir du meilleur code linéaire connu

Soient n et k deux entiers fixés avec $2 \leq k \leq n-1$. La construction consiste à considérer le meilleur code linéaire D de longueur n et de dimension k connu dans MAGMA et de vérifier s'il contient des mots de code de poids n et de rassembler ces mots de code dans un ensemble noté R. Si R est non vide, on choisit un mot de code arbitraire $\mathbf{v} \in R$ et on construit dans $\mathbb{F}_4$ un sous code $C \subset D$ de paramètres $[n, 1, n]$ engendré par $\mathbf{v}$.

A partir des codes C et D, on peut construire deux codes quantiques asymétriques. Le premier code Q est construit via le théorème 5.6.1 et sans l'utilisation de l'application S en prenant $C_1^{\perp_{\mathrm{tr}}} = C$ et $C_2 = D$, quant au deuxième code Q_S , il est obtenu via l'application de S et le théorème 5.6.4

Proposition 5.7.1 *Soit un entier $n \geq 3$, il existe un $[[n, n-2, 2/2]]$-code asymétrique quantique MDS sur $\mathbb{F}_4$.*

Preuve 5.7.2 *Une preuve de l'existence d'un $[[n, n-2, 2/2]]$-code asymétrique quantique MDS sur $\mathbb{F}_q$ est donnée dans [24, Cor. 3.4]. Nous présentons donc ici une construction de ce code dans le cas $q = 4$. En utilisant le polynôme $X+1$ comme polynôme générateur, on construit un $[n, n-1, 2]$-code cyclique sur $\mathbb{F}_4$. Le polynôme de contrôle de ce code est égal à $1 + X + \ldots + X^{n-1}$ et par conséquent sa distance minimale est 2. Il vient d'après [22, Th. 1], que D possède des mots de code de longueur n. Un de ces mots de code peut être choisi pour la construction d'un $[n, 1, n]$-code C sur $\mathbb{F}_4$. En appliquant le théorème 5.6.1 avec $C_1^{\perp_{\mathrm{tr}}} = C$ et $C_2 = D$ on obtient la construction voulue.*

Remarque 5.7.1 *Signalons que pour une longueur n fixée, l'existence d'un code linéaire optimal sur $\mathbb{F}_4$, de paramètres $[n, k, d]$ pour $k \in \{2, \ldots, n-2\}$, et contenant un mot de code de poids n n'est pas toujours vérifié. Par exemple, il n'existe aucun mot de code de poids 6 dans le meilleur code linéaire connu dans MAGMA de paramètres $[6, 4, 2]$ sur $\mathbb{F}_4$.*

La table 5.1 présente tous les codes quantiques résultants de cette méthode pour les longueurs n allant de 4 à 20 à partir des meilleurs codes linéaires connus avec les paramètres $[n,k]_4$ sous MAGMA. Signalons que nous n'avons pas traité le cas $k = n-1$ compte tenu de la proposition 5.7.1 et le cas $k = 1$ compte tenu de [23, Ex. 8.2].

Remarque 5.7.2 *Ces valeurs numériques montrent l'avantage de l'utilisation de l'application S. En effet, il fût impossible d'obtenir directement des codes quantiques asymétriques pour des valeurs de longueurs particulières sans l'utilisation de cette application. On peut citer l'exemples des codes* $[[18,2,12/2]]_4$, $[[30,2,22/2]]_4$, $[[30,3,20/2]]_4$, $[[32,3,22/2]]_4$, $[[38,4,22/2]]_4$, $[[40,4,24/2]]_4$, $[[42,4,26/2]]_4$, $[[44,4,28/2]]_4$, *et* $[[46,4,28/2]]_4$.

5.7.2 Construction à partir des codes de Reed Solomon concaténés

On considère un entier naturel m. La concaténation est une méthode de construction utilisée pour l'obtention de codes sur $\mathbb{F}_q$ à partir de codes sur une extension $\mathbb{F}_{q^m}$ de $\mathbb{F}_q$. Pour un aperçu général sur cette méthode les lecteurs peuvent consulter [43, Sec. 6.3] et [46, Ch. 10]. La construction présentée dans cette partie consiste à générer des codes emboités $C \subset D$ sur $\mathbb{F}_4$ à partir de codes emboités $A \subset B$ sur $\mathbb{F}_{4^m}$. Nous utiliserons alors les codes C et D et l'application S pour l'obtention d'un code quantique asymétrique Q.

En considérant le corps $\mathbb{F}_{4^m}$ comme un $\mathbb{F}_4$-espace vectoriel menu de la base $\{\beta_1 \ldots, \beta_m\}$, nous écrivons tout élément $x \in \mathbb{F}_{4^m}$ d'une façon unique comme

$$x = \sum_{j=1}^{m} a_j \beta_j \text{ avec } a_j \in \mathbb{F}_4.$$

Nous définissons l'application $\phi : \mathbb{F}_{4^m} \to \mathbb{F}_4^m$ par $x \mapsto (a_1, \ldots, a_m)$. L'application ϕ est un $\mathbb{F}_4$-isomorphisme d'espaces vectoriels et on peut la prolonger d'une façon naturelle à l'application ϕ^* :

$$\begin{array}{rl} \phi^* : & \mathbb{F}_{4^m}^N \to \mathbb{F}_4^{mN} \\ & (x_1, \ldots, x_n) \mapsto (\phi(x_1), \ldots, \phi(x_n)). \end{array} \tag{5.7.29}$$

Soit A un $[N,K,D]_{4^m}$-code et Soit $C = \phi^*(A)$. On peut facilement vérifier que C est un $[mN, mK, \geq D]_4$-code. De plus, la linéarité de l'application ϕ^* implique que si A est un $[N,K_1,D_1]_{4^m}$- sous code d'un $[N,K_2,D_2]_{4^m}$-code B, alors $C = \phi^*(A)$ est un sous code $D = \phi^*(B)$ comme codes de $\mathbb{F}_4$. Soit $q = 4^m$ et $\alpha_1, \ldots, \alpha_{q-1}$ des éléments non nuls de $\mathbb{F}_q$. On a d'après [46, Ch. 10 et Ch.11] la matrice de contrôle du $[q, k, q-k+1]_q$-code de Reed Solomon concaténé B est donnée par

$$H = \begin{pmatrix} 1 & 1 & \ldots & 1 & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_{q-1} & 0 \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_{q-1}^2 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_1^{q-k-1} & \alpha_2^{q-k-1} & \ldots & \alpha_{q-1}^{q-k-1} & 0 \end{pmatrix}. \tag{5.7.30}$$

Soit A un $[q,1,q]_q$-code de répétition engendré par $\mathbf{1}=(1,\ldots,1)$. Pour obtenir que A soit un sous code de B nous choisissons α comme un élément primitif de $\mathbb{F}_q$. En effet, pour tout $1\leq j\leq q-2$ et en notant $s=\sum_{l=1}^{q-1}\alpha_l^j$, on a $\alpha^j s=\sum_{l=1}^{q-1}(\alpha\cdot\alpha_l)^j=s$. Comme $\alpha^j\neq 1$, on conclut que $s=0$ et par conséquent $A\subset B$.

D'un autre côté, en choisissant une $\mathbb{F}_4$-base $\{\beta_1\ldots,\beta_m\}$ de $\mathbb{F}_q$ de façon que la matrice génératrice du code $C'=\phi^*(A)$ soit donnée par $m\times mq$ matrice $G=(I_m|I_m|\ldots|I_m)$ où I_m est la matrice identité d'ordre m, on obtient que les paramètres de C' sont $[mq,m,q]_4$. Comme $\mathbf{1}=(1,\ldots,1)\in C'$ nous pouvons définir C comme un $[mq,1,mq]_4$-sous code de répétition de C'. Le code $D=\phi^*(B)$ est un $[mq,mk,d'\geq(q-k+1)]_4$-code contenant C. La preuve du théorème 5.6.4 nous donne le résultat suivant :

Proposition 5.7.3 *Soit m un entier naturel, $q=4^m$ et $1\leq k\leq q$. Il existe alors un $[[2mq,mk-1,(\geq 2(q-k+1))/2]]_4$-code quantique asymétrique Q.*

Remarque 5.7.3 *Pour une valeur fixée de m et une base donnée $\{\beta_1,\ldots,\beta_m\}$, la distance $d'=d(D)$ peut être explicitement calculée. D'un autre côté et comme il a été signalé dans [46, Ch. 10], un changement de la base peut changer le poids de distribution et la distance minimale du code D.*

Exemple 5.7.4 *Pour $m=2$ et $1\leq k\leq 16$, nous obtenons le $[[64,k',d_z/2]]_4$-code quantique asymétrique donné dans la table 5.2.*

Exemple 5.7.5 *Pour $m=3$ et $1\leq k\leq 64$ nous obtenons le $[[384,k',d_z/2]]_4$-code quantique asymétrique donné dans la table 5.3.*

TABLE 5.1 – Codes Quantiques asymétriques QECC à partir des BKLC

n	**Code** Q	**Code** Q_S	n	**Code** Q	**Code** Q_S
4	$[[\mathbf{4,1,3/2}]]_{\mathbf{4}}$	$[[8,1,6/2]]_4$	15	$[[15,7,6/2]]_4$	$[[30,7,12/2]]_4$
5	$[[\mathbf{5,2,3/2}]]_{\mathbf{4}}$	$[[10,2,6/2]]_4$		$[[15,8,5/2]]_4$	$[[30,8,10/2]]_4$
6	$[[\mathbf{6,2,4/2}]]_{\mathbf{4}}$	$[[12,2,8/2]]_4$		$[[15,10,4/2]]_4$	$[[30,10,8/2]]_4$
7	$[[7,2,4/2]]_4$	$[[14,2,8/2]]_4$		$[[15,11,3/2]]_4$	$[[30,11,6/2]]_4$
	$[[7,3,3/2]]_4$	$[[14,3,6/2]]_4$	16	$[[16,2,12/2]]_4$	$[[32,2,24/2]]_4$
8	$[[8,1,6/2]]_4$	$[[16,1,12/2]]_4$		$[[16,3,11/2]]_4$	$[[32,3,22/2]]_4$
	$[[8,2,5/2]]_4$	$[[16,2,10/2]]_4$		$[[16,6,8/2]]_4$	$[[32,6,16/2]]_4$
	$[[8,3,4/2]]_4$	$[[16,3,8/2]]_4$		$[[16,7,7/2]]_4$	$[[32,7,14/2]]_4$
	$[[8,4,3/2]]_4$	$[[16,4,6/2]]_4$		$[[16,8,6/2]]_4$	$[[32,8,12/2]]_4$
9	$[[9,2,6/2]]_4$	$[[18,2,12/2]]_4$		$[[16,9,5/2]]_4$	$[[32,9,10/2]]_4$
	$[[9,3,5/2]]_4$	$[[18,3,10/2]]_4$		$[[16,11,4/2]]_4$	$[[32,11,8/2]]_4$
	$[[9,4,4/2]]_4$	$[[18,4,8/2]]_4$		$[[16,12,3/2]]_4$	$[[32,12,6/2]]_4$
	$[[9,5,3/2]]_4$	$[[18,5,6/2]]_4$	17	$[[17,5,9/2]]_4$	$[[34,5,18/2]]_4$
10	$[[10,3,6/2]]_4$	$[[20,3,12/2]]_4$		$[[17,8,7/2]]_4$	$[[34,8,14/2]]_4$
	$[[10,4,5/2]]_4$	$[[20,4,10/2]]_4$		$[[17,9,6/2]]_4$	$[[34,9,12/2]]_4$
	$[[10,5,4/2]]_4$	$[[20,5,8/2]]_4$		$[[17,10,5/2]]_4$	$[[34,10,10/2]]_4$
	$[[10,6,3/2]]_4$	$[[20,6,6/2]]_4$		$[[17,12,4/2]]_4$	$[[34,12,8/2]]_4$
11	$[[11,2,7/2]]_4$	$[[22,2,14/2]]_4$		$[[17,13,3/2]]_4$	$[[34,13,6/2]]_4$
	$[[11,4,6/2]]_4$	$[[22,4,12/2]]_4$	18	$[[18,5,10/2]]_4$	$[[36,5,20/2]]_4$
	$[[11,5,5/2]]_4$	$[[22,5,10/2]]_4$		$[[18,6,9/2]]_4$	$[[36,6,18/2]]_4$
	$[[11,6,4/2]]_4$	$[[22,6,8/2]]_4$		$[[18,8,8/2]]_4$	$[[36,8,16/2]]_4$
	$[[11,7,3/2]]_4$	$[[22,7,6/2]]_4$		$[[18,10,6/2]]_4$	$[[36,10,12/2]]_4$
12	$[[12,2,8/2]]_4$	$[[24,2,16/2]]_4$		$[[18,11,5/2]]_4$	$[[36,11,10/2]]_4$
	$[[12,3,7/2]]_4$	$[[24,3,14/2]]_4$		$[[18,12,4/2]]_4$	$[[36,12,8/2]]_4$
	$[[12,5,6/2]]_4$	$[[24,5,12/2]]_4$		$[[18,14,3/2]]_4$	$[[36,14,6/2]]_4$
	$[[12,7,4/2]]_4$	$[[24,7,8/2]]_4$	19	$[[19,4,11/2]]_4$	$[[38,4,22/2]]_4$
	$[[12,8,3/2]]_4$	$[[24,8,6/2]]_4$		$[[19,5,10/2]]_4$	$[[38,5,20/2]]_4$
13	$[[13,2,9/2]]_4$	$[[26,2,18/2]]_4$		$[[19,6,9/2]]_4$	$[[38,6,18/2]]_4$
	$[[13,4,7/2]]_4$	$[[26,4,14/2]]_4$		$[[19,8,8/2]]_4$	$[[38,8,16/2]]_4$
	$[[13,5,6/2]]_4$	$[[26,5,12/2]]_4$		$[[19,9,7/2]]_4$	$[[38,9,14/2]]_4$
	$[[13,6,5/2]]_4$	$[[26,6,10/2]]_4$		$[[19,11,6/2]]_4$	$[[38,11,12/2]]_4$
	$[[13,8,4/2]]_4$	$[[26,8,8/2]]_4$		$[[19,12,5/2]]_4$	$[[38,12,10/2]]_4$
	$[[13,9,3/2]]_4$	$[[26,9,6/2]]_4$		$[[19,13,4/2]]_4$	$[[38,13,8/2]]_4$
14	$[[14,2,10/2]]_4$	$[[28,2,20/2]]_4$		$[[19,15,3/2]]_4$	$[[38,15,6/2]]_4$
	$[[14,3,9/2]]_4$	$[[28,3,18/2]]_4$	20	$[[20,4,12/2]]_4$	$[[40,4,24/2]]_4$
	$[[14,4,8/2]]_4$	$[[28,4,16/2]]_4$		$[[20,5,11/2]]_4$	$[[40,5,22/2]]_4$
	$[[14,5,7/2]]_4$	$[[28,5,14/2]]_4$		$[[20,6,10/2]]_4$	$[[40,6,20/2]]_4$
	$[[14,6,6/2]]_4$	$[[28,6,12/2]]_4$		$[[20,7,9/2]]_4$	$[[40,7,18/2]]_4$
	$[[14,7,5/2]]_4$	$[[28,7,10/2]]_4$		$[[20,9,8/2]]_4$	$[[40,9,16/2]]_4$
	$[[14,9,4/2]]_4$	$[[28,9,8/2]]_4$		$[[20,10,7/2]]_4$	$[[40,10,14/2]]_4$
	$[[14,10,3/2]]_4$	$[[28,10,6/2]]_4$		$[[20,12,6/2]]_4$	$[[40,12,12/2]]_4$
15	$[[15,2,11/2]]_4$	$[[30,2,22/2]]_4$		$[[20,13,5/2]]_4$	$[[40,13,10/2]]_4$
	$[[15,3,10/2]]_4$	$[[30,3,20/2]]_4$		$[[20,14,4/2]]_4$	$[[40,14,8/2]]_4$
	$[[15,6,7/2]]_4$	$[[30,6,14/2]]_4$		$[[20,16,3/2]]_4$	$[[40,16,6/2]]_4$

TABLE 5.2 – $[[64, k', d_z/2]]_4$-code Q à partir des $[16, k, 16-k+1]_{16}$-codes de Reed-Solomon étendus

k	1	2	3	4	5	6	7	8
k'	1	3	5	7	9	11	13	15
$d_z \geq$	32	30	28	26	24	22	20	18
k	9	10	11	12	13	14	15	16
k'	17	19	21	23	25	27	29	31
$d_z \geq$	16	14	12	10	8	6	4	2

TABLE 5.3 – $[[384, k', d_z/2]]_4$-code Q à partir des $[64, k, 64-k+1]_{64}$-codes de Reed-Solomon étendus

k	1	2	3	4	5	6	7	8
k'	2	5	8	11	14	17	20	23
$d_z \geq$	128	126	124	122	120	118	116	114
k	9	10	11	12	13	14	15	16
k'	26	29	32	35	38	41	44	47
$d_z \geq$	112	110	108	106	104	102	100	98
k	17	18	19	20	21	22	23	24
k'	50	53	56	59	62	65	68	71
$d_z \geq$	96	94	92	90	88	86	84	82
k	25	26	27	28	29	30	31	32
k'	74	77	80	83	86	89	92	95
$d_z \geq$	80	78	76	74	72	70	68	66
k	33	34	35	36	37	38	39	40
k'	98	101	104	107	110	113	116	119
$d_z \geq$	64	62	60	58	56	54	52	50
k	41	42	43	44	45	46	47	48
k'	122	125	128	131	134	137	140	143
$d_z \geq$	48	46	44	42	40	38	36	34
k	49	50	51	52	53	54	55	56
k'	146	149	152	155	158	161	164	167
$d_z \geq$	32	30	28	26	24	22	20	18
k	57	58	59	60	61	62	63	64
k'	170	173	176	179	182	185	188	191
$d_z \geq$	16	14	12	10	8	6	4	2

Chapitre 6

Nouvelle construction des codes quasi θ-cycliques sur $\mathbb{F}_4$

6.1 Introduction

Dans [2], les auteurs ont introduit une nouvelle classe de codes à savoir les codes quasi θ-cycliques. L'intérêt de cette classe de codes est qu'elle contient des codes quasi-cycliques et qui contient -comme pour la classe des codes θ-cycliques- des codes avec de bons paramètres. Pour une documentation complète sur ces codes, leurs caractéristiques et leurs constructions, le lecteur pourrait consulter [2] et [1]. Dans ce chapitre, nous continuons l'utilisation de l'application S présentée dans le chapitre précédent sur cette nouvelle classe des codes. Cette application permettra en premier lieu de donner une nouvelle construction plus simple d'un code quasi θ-cyclique révélant ainsi une nouvelle relation entre les codes quasi θ-cycliques et les codes quasi-cycliques. En effet, nous montrons que l'image par l'application S d'un code quasi θ-cyclique d'index l sur $\mathbb{F}_4$ est équivalente à un code $2l$-quasi-cyclique. En utilisant les résultats du chapitre précédent, cette construction pourrait nous aider à donner des nouvelles constructions de codes quantiques.

6.2 codes quasi θ-cycliques

Soit $\mathbb{F}_q$ le corps fini de caractéristique p et de cardinal $q = p^{mt}$, θ un automorphisme de $\mathbb{F}_q$ d'ordre m et soit un entier $n = sl$ et tel que m/s.

Définition 6.2.1 *Un ensemble C de $\mathbb{F}_q^n$ est dit code quasi θ-cyclique d'index l si et seulement si C est un sous espace vectoriel de $\mathbb{F}_q^n$ vérifiant :*
si

$$c = (c_{0,0}, c_{0,1}, ...c_{0,l-1}, c_{1,0}, c_{1,1}, ..., c_{1,l-1}, ..., c_{s-1,0}, c_{s-1,1}, ..., c_{s-1,l-1}) \in C$$

alors

$$T_{\theta,l}(c) =$$

$$(\theta(c_{s-1,0}), \theta(c_{s-1,1}), ..., \theta(c_{s-1,l-1}), \theta(c_{0,0}), \theta(c_{0,1}), ..., \theta(c_{0,l-1}), ..., \theta(c_{s-2,0}), \theta(c_{s-2,1}), ..., \theta(c_{s-2,l-1})) \in C.$$

Remarque 6.2.1 *Dans le cas où l'automorphisme θ est l'identité, un code quasi θ-cyclique est tout simplement un code quasi-cyclique.*

On considère dans la suite l'anneau des polynômes tordus $\mathbb{F}_q[X;\theta]$ et l'anneau $\Re_s = \mathbb{F}_q[X;\theta]_{/X^s-1}$. En définissant dans l'anneau $\Re_s^l$ la multiplication

$$f(x)(g_1(x), g_2(x), ..., g_l(x)) = (f(x)g_1(x), f(x)g_2(x), ..., f(x)g_l(x)),$$

on obtient que l'anneau $\Re_s^l$ est un $\Re_s$-module.
Notons maintenant l'application Φ de $\mathbb{F}_q^n$ à $\Re_s^l$ définie par : pour tout

$$c = (c_{0,0}, c_{0,1}, ...c_{0,l-1}, c_{1,0}, c_{1,1}, ..., c_{1,l-1}, ..., c_{s-1,0}, c_{s-1,1}, ..., c_{s-1,l-1}) \in \mathbb{F}_q^n = \mathbb{F}_q^{sl},$$

on a

$$\Phi(c) = (c_0(x), c_1(x), ..., c_{l-1}(x)),$$

où $c_j(X) = \sum_{i=0}^{s-1} c_{i,j}X^i \in \mathbb{F}_q[X;\theta]_{/X^s-1}$ pour $j = 0, ..., l-1$.
Remarquons que l'application Φ est un isomorphisme d'espaces vectoriels assurant une correspondance terme par terme entre les éléments de $\mathbb{F}_q^n$ et ceux de $\Re_s^l$. On a donc le résultat suivant :

Proposition 6.2.1 *[2] Un sous ensemble C de $\mathbb{F}_q^n$ est un quasi-θ-cyclique code de longueur $n = sl$ et d'index l si et seulement si $\Phi(C)$ est un sous module à gauche de l'anneau $\Re_s^l$.*

Parmi les codes quasi-θ-cycliques, on distingue la classe des codes engendrés par un seul polynôme. On a la définition suivante :

Définition 6.2.2 *On appelle code quasi-θ-cyclique engendré par un seul polynôme tout sous module à gauche de $\Re_s^l$ ayant la forme*

$$C = \{f(X)(g_1(X), g_2(X), ..., g_l(X)) : f(X) \in \Re_s\}.$$

6.3 L'application S sur les codes quasi θ-cycliques de $\mathbb{F}_4$

Dans la suite, on considère l'automorphisme de Frobenuis défini dans $\mathbb{F}_4$ par $\theta(x) = x^2 = \overline{x}$. On rappelle la définition de l'application S donnée dans le chapitre précédent :

$$\begin{array}{rccc} S: & \mathbb{F}_4^n & \mapsto & \mathbb{F}_4^{2n} \\ & (x_1, x_2, ..., x_n) & \mapsto & (x_1, \overline{x_1}, x_2, \overline{x_2}..., x_n, \overline{x_n}). \end{array}$$

On note $[1..2n]$ l'ensemble $\{1, 2, \ldots, 2n\}$ et la permutation $\sigma = \tau \circ T^{2l}$ dans $[1..2n]$ où T est le shift modulo $2n$ et $\tau = (12)(34)\ldots(2n-1, 2n)$. On note toujours la permutation identité par (1).

On considère aussi Σ la permutation sur les éléments de $\mathbb{F}_4^{2n}$ induite par σ. Ainsi, pour $\mathbf{x} = (x_1, x_2, \ldots, x_{2n}) \in \mathbb{F}_4^{2n}$, on a

$$\Sigma(\mathbf{x}) = \left(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(2n)}\right). \tag{6.3.1}$$

Dans la suite on considère C un code quasi-θ-cyclique sur $\mathbb{F}_4$ d'index l et de longueur $n = 2sl$. On utilisera les propriétés de l'application S données dans le chapitre précédent.

Lemme 6.3.1 *$S(C)$ est stable par la permutation Σ.*

Preuve 6.3.2 *Rappelllons d'abord que dans le cas $l = 1$, on retrouve le cas d'un code $\theta-$cyclique et où $S(C)$ est stable par la permutation induite de $\tau \circ T^2$.*

Soit maintenant $\mathbf{v} = (v_1, v_2, \ldots, v_{2n}) \in S(C)$. Il existe donc $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in C$ tel que

$$\mathbf{v} = (u_1, \overline{u_1}, u_2, \overline{u_2}, \ldots, u_n, \overline{u_n}) = S(\mathbf{u}).$$

Comme C est un code quasi-$\theta-$cyclique d'index l, on a

$$\overline{\mathbf{u}} := (\overline{u_{n-(l-1)}}, \overline{u_{n-(l-2)}}, \ldots, \overline{u_n}, \overline{u_1 u_{n-l}}) \in C.$$

D'où,

$$\begin{aligned}\Sigma(\mathbf{v}) &= (\overline{u_{n-(l-1)}}, u_{n-(l-1)}, \overline{u_{n-(l-2)}}, u_{n-(l-2)} \ldots, \overline{u_n}, u_n\overline{u_1}, u_1, \overline{u_{n-l}, u_{n-l}}) \\ &= S((\overline{u_{n-(l-1)}}, \overline{u_{n-(l-2)}}, \ldots, \overline{u_n}, \overline{u_1 u_{n-1}})) = S(\overline{\mathbf{u}}),\end{aligned}$$

ce qui implique que $\Sigma(S(C)) \subset S(C)$.

Lemme 6.3.3 *L'ordre de σ est égal à $2s$.*

Preuve 6.3.4 *Signalons ici qu'on retrouve le résultat 5.4.3 en prenant $n = 2sl$ et $l = 1$. D' un autre côté, la permutation σ est explicitement définie par*

$$\sigma : i \mapsto \begin{cases} i + 2l + 1 \pmod{2n} \text{ si } i \text{ est impair} \\ i + 2l - 1 \pmod{2n} \text{ si } i \text{ est pair}, \end{cases} . \tag{6.3.2}$$

où $1 \leq i \leq 2n$.
En calculant modulo $2n$, on remarque que si i est impair, alors $\sigma(i) = i + 2l + 1$ et $\sigma^2(i) = \sigma(i+2l+1) = i+4l$. Si i est pair, alors $\sigma(i) = i+2l-1$ et $\sigma^2(i) = \sigma(i+1) = i+4l$. D'où, $\sigma^2 = T^{4l}$.
Ainsi,

$$\sigma^{2s} = T^{4ls} = T^{2n} = (1).$$

De plus, pour tout $1 \leq k < 2s$, on a si $k = 2i$, alors

$$\sigma^k = \sigma^{2i} = T^{4li} \neq (1),$$

puisque $4li = 2lk < 4ls = 2n$. *Si* $k = 2i + 1$ *alors*

$$\sigma^k = \sigma^{2i+1} = T^{4li+2l} \circ \tau = T^{2lk} \circ \tau \neq (1)?$$

car $T^{2lk}(1) = 1 + 2lk \leq 2n - 2l + 1$. *Par conséquent l'ordre de* σ *est* $2s$.

Remarque 6.3.1 *Comme* $\sigma^k(1) \neq 2$,$\sigma^k(2) \neq 3$... *et* $\sigma^k(2l-1) \neq 2l$ *pour tout* $0 \leq k \leq 2s-1$, σ *s'écrit en produit de* $2l$ *cycles de longueur* $2s$ *chacun :*

$$\sigma = (1, \sigma(1), \sigma^2(1), ..., \sigma^{2s-1}(1))(2, \sigma(2), \sigma^2(2), ..., \sigma^{2s-1}(2)).....(2l, \sigma(2l), \sigma^2(2l), ..., \sigma^{2s-1}(2l)). \tag{6.3.3}$$

Proposition 6.3.5 *Soit un entier* $n = 2sl$ *et* C *un* $[n, k, d]$*-code quasi-*θ*-cyclique de* $\mathbb{F}_4^n$. $S(C)$ *est équivalent à un* $(2n, 2^{2k}, 2d)$*-code* $2l$*-quasi-cyclique additif de* $\mathbb{F}_4^{2n}$.

Preuve 6.3.6 *Ce résultat est une généralisation de la proposition 5.4.5 où* $l = 1$. *Définissons dans la suite la permutation* σ' *par*

$$\sigma' = \tag{6.3.4}$$

$$\begin{pmatrix} 1 & 2 & ... & 2l & 2l+1 & 2l+2 & ... & 4l & ... & 2n-2l+1 & 2n-2l+2 \\ \sigma^{2s-1}(1) & \sigma^{2s-1}(2) & ... & \sigma^{2s-1}(2l) & \sigma^{2s-2}(1) & \sigma^{2s-2}(2) & ... & \sigma^{2s-2}(2l) & ... & \sigma^0(1) & \sigma^0(2) \end{pmatrix}$$

On rappelle que pour tout $1 \leq k \leq 2n$, *il existe deux entiers* q *et* r *tels que*

$$k = 2lq + r \ \ , \ \ 1 \leq r < 2l.$$

Ainsi, on a

$$\sigma'(k) = \sigma^{2s-(q+1)}(r). \tag{6.3.5}$$

Soit la permutation Σ' *définie sur les éléments de* $\mathbb{F}_4^{2n}$ *et induite par* σ'. *Pour* $\mathbf{y} = (y_1, y_2, \ldots, y_{2n}) \in \mathbb{F}_4^{2n}$, *on a*

$$\Sigma'(\mathbf{y}) = \left(y_{\sigma'(1)}, y_{\sigma'(2)}, \ldots, y_{\sigma'(2n)}\right). \tag{6.3.6}$$

Pour montrer que $\Sigma'(S(C))$ *est quasi-cyclique d'index* $2l$, *il suffit de montrer que pour chaque* Y *dans* $S(C)$

$$T^{2l}(\Sigma'(Y)) \in \Sigma'(S(C)).$$

Ainsi, en utilisant le fait que $S(C)$ *est stable par* Σ, *il nous suffit de montrer que*

$$T^{2l}(\Sigma'(Y)) = \Sigma'(\Sigma(Y)). \tag{6.3.7}$$

Soit maintenant Y *dans* $S(C)$. *On a*

$$\Sigma(Y) = (y_{\sigma(1)}, y_{\sigma(2)}, ..., y_{\sigma(2n)}), \tag{6.3.8}$$

$$= (y'_1, y'_2, ..., y'_{2n}), \tag{6.3.9}$$

où l'on a noté

$$y'_i = y_{\sigma(i)}. \tag{6.3.10}$$

Ainsi,

$$\begin{aligned} \Sigma'(\Sigma(Y)) &= (y'_{\sigma'(1)}, y'_{\sigma'(2)}, ..., y'_{\sigma'(2n)}), \\ &= (y_{\sigma(\sigma'(1))}, y_{\sigma(\sigma'(2))}, ..., y_{\sigma(\sigma'(2n))}). \end{aligned}$$

Or, il vient de 6.3.5 que,

$$\sigma(\sigma'(k)) = \sigma^{2s-q}(r).$$

Par suite, on a compte tenu de 6.3.8,

$$\Sigma'(\Sigma(Y)) = (y_{\sigma^{2s}(1)}, y_{\sigma^{2s}(2)}, ..., y_{\sigma^{2s}(2l)}, y_{\sigma^{2s-1}(1)}, y_{\sigma^{2s-1}(2)}...y_{\sigma^{2s-1}(2l)}, ..., y_{\sigma(1)}, y_{\sigma(2)}, ..., y_{\sigma(2l)}).$$

D'un autre côté, comme

$$\begin{aligned} \Sigma'(Y) &= (y_{\sigma'(1)}, y_{\sigma'(2)}, ..., y_{\sigma'(2n)}), \\ &= (y_{\sigma^{2s-1}(1)}, y_{\sigma^{2s-1}(2)}, ..., y_{\sigma^{2s-1}(2l)}, y_{\sigma^{2s-2}(1)}, y_{\sigma^{2s-2}(2)}...y_{\sigma^{2s-2}(2l)}, ..., y_{\sigma^{2s}(1)}, y_{\sigma^{2s}(2)}, ..., y_{\sigma^{2s}(2l)}), \end{aligned}$$

et que $\Sigma(S(C)) = S(C)$, on a

$$T^{2l}(\Sigma'(Y)) = \Sigma'(\Sigma(Y)) \in \Sigma'(S(C)).$$

Ainsi, $\Sigma'(S(C))$ est un $2l$-quasi-cyclique code, ce qu'il fallait démontrer.

6.4 Conclusions

1. Dans [2, Conclusion], les auteurs signalent qu'un des problèmes rencontrés pour bien étudier la structure des codes quasi cycliques tordus a un générateur est la factorisation du polynôme $X^n - 1$ dans l'anneau des polynômes tordus $\mathbb{F}_q[X;\theta]$. Nous notons ici que nous avons présenté deux solutions pour ce problème en caractéristique 2 dans ce travail : la première c'est l'algorithme de factorisation de $X^n - 1$ dans $\mathbb{F}_4[X;\theta]$ donné dans le troisième chapitre de cette thèse et la deuxième c'est la méthode de construction d'un code quasi θ-cyclique via l'application S donnée dans ce chapitre et donc le fait de ramener le problème à une factorisation de $X^n - 1$ dans $\mathbb{F}_q[X]$ et non dans $\mathbb{F}_q[X;\theta]$ pour $q = 4$.
2. Toujours dans [2], les auteurs ont donné sept nouveaux codes de paramètres optimaux. Ainsi, en appliquant la proposition 5.6.4 via l'application S sur ces codes, on pourrait trouver des nouveaux codes quantiques avec des bons paramètres. Par exemple, en prenant le code D dans 5.6.4 comme étant le $[48, 12, 24]$ trouvé dans [2] et C un sous code de D vérifiant $d(C^{\perp_{\mathrm{tr}}}) \geq 2$, nous obtenons un nouveau code quantique de longueur 96 et de paramètres $d_x = 24$ et $d_z = 2$.

Chapitre 7

Des codes θ-cycliques sur $\mathbb{F}_2 + v\mathbb{F}_2$ aux codes quasi-cycliques binaires

7.1 Introduction

Dans des récents papiers [12, 11, 8, 9], Ulmer et ses co-auteurs ont introduit et étudié la notion des codes θ-cycliques. Cette étude a révélé une classe intéressante de codes avec de très bons paramètres spécialement sur l'anneau $\mathbb{F}_4$. Après ces travaux, il fut naturel de penser à étudier les codes θ-cycliques sur d'autres anneaux de cardinal 4. Or les anneaux Z_4 et $S = \mathbb{F}_2 + u\mathbb{F}_2$ tel que $u^2 = 0$ ne possèdent pas un automorphisme non trivial, un élément nécessaire pour définir un code θ-cyclique. Ainsi, le seul anneau commutatif possédant un automorphisme non trivial est l'anneau $R = \mathbb{F}_2 + v\mathbb{F}_2 = \{0, 1, v, v+1\}$ où $v^2 = v$. En effet, en posant $\theta(0) = 0$, $\theta(1) = 1$, $\theta(v) = v + 1$ et $\theta(v + 1) = v$, il est clair que l'application θ est un automorphisme non trivial de l'anneau R. Signalons que l'anneau R est isomorphe à l'anneau $\mathbb{F}_2 \times \mathbb{F}_2$ via l'application z définie par $z(0) = (0, 0), z(1) = (1, 1), z(v) = (1, 0)$ et $z(v + 1) = (0, 1)$. Nous considérons deux différents poids sur les mots de codes de l'anneau R : le poids de Hamming et le poids de Lee. Le poids de Lee des éléments de R : $0, 1, v, v + 1$ est respectivement $0, 2, 1$ et 1. Le poids d'un mot de code est comme d'habitude la somme des poids de Lee de ses composants. La distance de Lee minimale d_L d'un code C est le plus petit poids de Lee d'un mot de code non nul de C.
Récemment, Taher Abualrub et P.Seneviratne ont présenté dans [3] une étude des codes θ-cycliques sur l'anneau R. L'approche utilisée dans ce papier -comme celle dans les travaux d'Ulmer et ses co-auteurs est purement algébrique : les auteurs utilisent la théorie d'idéaux dans les anneaux d'Öre. Dans ce chapitre, nous utiliserons une approche plus combinatoire de ce cas spécial à savoir les codes θ-cycliques sur l'anneau $R = \mathbb{F}_2 + v\mathbb{F}_2$. En effet, nous introduisons une application Z de R^n à $\mathbb{F}_2^{2n}$. Cette application est $\mathbb{F}_2$-linéaire et elle transforme les codes cycliques en des codes quasi-cycliques d'index 2. Elle est une isométrie transformant la distance de Lee dans R en la distance de Hamming sur $\mathbb{F}_2$. Si C est un code $\theta-$cyclique alors, on montre que $Z(C)$ est équivalent à un code cyclique si n est impair et à un code quasi-cyclique d'index 2 si n est pair. Une nouvelle construction des codes $\theta-$cycliques et des codes $\theta-$cycliques auto-duaux euclidiens de R^n est donnée. A la fin de

ce chapitre nous présentons quelques problèmes ouverts à propos de la construction des codes $\theta-$cycliques auto-duaux hermitiens et des codes $\theta-$ cycliques auto-duaux hermitiens de type IV sur R^n.

7.2 L'application Z

On note dans la suite $\theta(x)$ par $\overline{x}$.

Définition 7.2.1 *Soit $B = (v, v+1)$ une base de R comme $\mathbb{F}_2$-espace vectoriel. On définit dans R^n l'application suivante*

$$\begin{aligned} Z: \quad R^n &\mapsto \mathbb{F}_2^{2n} && (7.2.1)\\ (x_1, x_2, ..., x_n) &\mapsto (z(x_1), z(x_2), z(x_n)) = (x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}..., x_{n,1}, x_{n,2}), && (7.2.2)\end{aligned}$$

où $(x_{i,1}, x_{i,2})$ sont les coordonnées de x_i dans la base B.

Remarque 7.2.1 *On remarque que si $z(u) = (u_1, u_2)$ alors $z(\overline{u}) = (u_2, u_1)$.*

Lemme 7.2.1 *L'application Z est un $\mathbb{F}_2$-isomorphisme d' espaces vectoriels.*

Considérons dans la suite un code linéaire C sur R de longueur n, de dimension k et de distance minimale d.

Remarque 7.2.2 *Comme Z est un $\mathbb{F}_2$-isomorphisme, On a*

$$dim_{\mathbb{F}_2}(Z(C)) = dim_{\mathbb{F}_2}(C) = 2k.$$

Lemme 7.2.2 *$Z(C)$ est un code additif binaire de longueur $2n$ et de cardinal $= card(C) = 4^k = 2^{2k}$. On note comme c'est convenu ses paramètres par $(2n, 2^{2k})$.*

Lemme 7.2.3 *Pour tout $u \in C$, on a*
1. Pour la distance de Hamming dans R :

$$W_H(u) \leq W_H(Z(u)) \leq 2W_H(u),$$

et

$$d_H(C) \leq d_H(Z(C)) \leq 2d_H(C).$$

2. Pour la distance de Lee dans R :

$$W_L(u) = W_H(Z(u)),$$

et

$$d_L(C) = d_H(Z(C)).$$

Preuve 7.2.4 *Soit* $u = (x_1, x_2, ..., x_n) \in C$.
1. Si $x_i \neq 0$ *alors un parmi ses deux coordonnées est nécessairement* $\neq 0$. *D'où, on a le résultat.*
2. Comme $x_i \in R$, *on a* $W_L(x_i) = W_H(z(x_i))$.

Remarque 7.2.3 *Z est une isométrie qui transforme la distance de Lee dans R à la distance de Hamming dans* $\mathbb{F}_2$.

Soit $a, b \in R$ où $z(a) = (a_1, a_2)$ et $z(b) = (b_1, b_2)$. On définit dans R un produit scalaire naturel par

$$a.b = a_1.b_1 + a_2.b_2.$$

Pour tout $u = (x_1, x_2, ..., x_n)$ et $v = (y_1, y_2, ..., y_n)$ de R^n. On définit les deux produits scalaires suivants dans R^n :
le produit scalaire euclidien donné par

$$< x, y >= \sum_{i=0}^{n} x_i.y_i,$$

et le produit scalaire hermitien donné par

$$< x, y >_H = \sum_{i=0}^{n} x_i.\overline{y_i}.$$

Lemme 7.2.5 *Soit* $u = (x_1, x_2, ..., x_n)$ *et* $v = (y_1, y_2, ..., y_n)$ *in* R^n, *on a*

$$< Z(u), Z(v) >=< u, v > .$$

Preuve 7.2.6 *On a*

$$< Z(u), Z(v) > = \sum_{i=0}^{n} x_{i,1}.y_{i,1} + x_{i,2}.y_{i,2}, \tag{7.2.3}$$

$$= \sum_{i=0}^{n} x_i.y_i =< u, v > . \tag{7.2.4}$$

Proposition 7.2.7 *Soit C un* $[n, k, d]$ *code linéaire sur R, on a*

$$(Z(C))^{\perp} = Z(C^{\perp}).$$

Preuve 7.2.8 *Soit* $v \in Z(C^{\perp})$, *il existe donc* $u \in C^{\perp}$ *tel que* $v = Z(u)$. *Si* $v' \in Z(C)$, *alors il existe* $u' \in C$ *tel que* $v' = Z(u')$. *On a*

$$< v, v' >=< Z(u), Z(u') >=< u, u' >= 0,$$

puisque $u \in C^{\perp}$ *et* $u' \in C$. *D'un autre côté et comme* $dim_{\mathbb{F}_2}(Z(C)) = dim_{\mathbb{F}_2}(C) = 2k$, *on a*

$$dim_{\mathbb{F}_2}((Z(C))^{\perp}) = 2n - k = dim_{\mathbb{F}_2}(C^{\perp}) = dim_{\mathbb{F}_2}(Z(C^{\perp})).$$

Corollaire 7.2.9 *Pour tout code linéaire C sur R^n, le code $Z(C)$ est :*
1. un code auto-orthogonal pour le produit scalaire euclidien si et seulement si C est un code auto-orthogonal pour le produit scalaire euclidien.
2. un code auto-dual pour le produit scalaire euclidien si et seulement si C est un code auto-dual pour le produit scalaire euclidien.

Lemme 7.2.10 *Le code C est un code cyclique sur R^n si et seulement si $Z(C)$ est un $2-$quasi-cyclique code sur $\mathbb{F}_2^{2n}$.*

7.3 Codes $\theta-$cycliques et application Z

Considérons dans la suite un code $\theta-$cyclique C sur R^n . On note $[1..2n]$ l'ensemble $\{1, 2, \ldots, 2n\}$ et $\sigma = \tau \circ T^2$ la permutation dans $[1..2n]$ où T est le shift modulo $2n$ et $\tau = (12)(34)\ldots(2n-1, 2n)$. On notera comme dans les chapitres précédents la permutation identité par (1). Soit Σ la permutation sur les éléments de $\mathbb{F}_2^{2n}$ induite par σ. Ainsi, pour $\mathbf{x} = (x_1, x_2, \ldots, x_{2n}) \in \mathbb{F}_2^{2n}$, on a

$$\Sigma(\mathbf{x}) = \left(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(2n)}\right). \tag{7.3.5}$$

Lemme 7.3.1 *Le code $Z(C)$ est stable par la permutation Σ si et seulement si C est un code $\theta-$cyclique sur R.*

Preuve 7.3.2 *Soit C un code $\theta-$cyclique et Soit $\mathbf{y} = (y_1, y_2, ..., y_n) \in Z(C)$. Il existe donc $\mathbf{x} = (x_1, x_2, ..., x_n) \in C$ tel que $\mathbf{y} = (x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}..., x_{n,1}, x_{n,2})$ et où $(x_{i,1}, x_{i,2})$ sont les coordonnées de x_i dans la base B pour tout $1 \leq i \leq n$.*
C est un code $\theta-$cyclique, d'où

$$\overline{\mathbf{x}} = (\overline{x_n}, \overline{x_1}, ..., \overline{x_{n-1}}) \in C$$

Comme les coordonnées de $\overline{x_i}$ dans B sont $(x_{i,2}, x_{i,1})$, on a

$$\begin{aligned} \Sigma(\mathbf{y}) &= (x_{n,2}, x_{n,1}, x_{1,2}, x_{1,1}..., x_{n-1,2}, x_{n-1,1}), \\ &= Z((\overline{x_n}, \overline{x_1}, ..., \overline{x_{n-1}})), \end{aligned}$$

ce qui implique $\Sigma(Z(C)) \subset Z(C)$.
Considérons maintenant un code C' sur $\mathbb{F}_2^{2n}$ stable par la permutation Σ et soit $\mathbf{y} = (y_1, y_2, ..., y_n) \in Z^{-1}(C')$. Il existe donc $\mathbf{x} = (x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}..., x_{n,1}, x_{n,2}) \in C'$, où $(x_{i,1}, x_{i,2})$ sont les coordonnées de y_i dans la base B pour tout $1 \leq i \leq n$.
On note

$$\overline{\mathbf{y}} = (\overline{y_n}, \overline{y_1}, ..., \overline{y_{n-1}}).$$

Montrer que $\overline{\mathbf{y}} \in Z^{-1}(C')$ revient à montrer que $Z(\overline{\mathbf{y}}) \in C'$. Ainsi,

$$Z(\overline{\mathbf{y}}) = (x_{n,2}, x_{n,1}, x_{1,2}, x_{1,1}..., x_{n-1,2}, x_{n-1,1}).$$

D'un autre côté

$$\begin{aligned}\Sigma(Z(\mathbf{y})) &= (x_{n,2}, x_{n,1}, x_{1,2}, x_{1,1}..., x_{n-1,2}, x_{n-1,1}),\\ &= Z(\overline{\mathbf{y}}) \in C',\end{aligned}$$

car C' est stable par Σ.
Par suite,

$$\overline{\mathbf{y}} \in Z^{-1}(C').$$

D'où, on a le résultat.

Lemme 7.3.3 *[21] L'ordre de σ est égal à $2n$ si n est impair et à n si n est pair.*

Remarque 7.3.1 *1) On rappelle que la permutation σ est définie par*

$$\sigma(1) = 2n, \tag{7.3.6}$$
$$\sigma(2) = 2n-1, \tag{7.3.7}$$

et pour tout $3 \leq i \leq 2n$,

$$\sigma(i) = i-1 \quad \textit{si } i \textit{ est impair}, \tag{7.3.8}$$
$$= i-3 \quad \textit{sinon}. \tag{7.3.9}$$

2) Pour n impair, σ est un cycle de longueur $2n$. On note

$$\sigma = (1, \sigma(1), \sigma^2(1), ..., \sigma^{2n-2}(1), \sigma^{2n-1}(1)) \tag{7.3.10}$$

3) Pour n pair, σ est un produit de deux cycles de longueur n chacun. Comme $\sigma^k(1) \neq 2$ pour tout $0 \leq k \leq n-1$, on a

$$\sigma = (1, \sigma(1), \sigma^2(1), ..., \sigma^{n-1}(1))(2, \sigma(2), \sigma^2(2), ..., \sigma^{n-1}(2)). \tag{7.3.11}$$

De plus, on a

$$\sigma^i(1) = 2i+1[2n] \quad \textit{Si } i \textit{ est pair}, \tag{7.3.12}$$
$$= 2i+2[2n] \quad \textit{si } i \textit{ est impair}, \tag{7.3.13}$$

et

$$\sigma^i(2) = 2i+2[2n] \quad \textit{si } i \textit{ est pair}, \tag{7.3.14}$$
$$= 2i+1[2n] \quad \textit{si } i \textit{ est impair}. \tag{7.3.15}$$

Dans la suite, on définit la permutation σ' par

$$\sigma' = \begin{pmatrix} 1 & 2 & \dots & 2n-1 & 2n \\ \sigma^{2n-1}(1) & \sigma^{2n-2}(1) & \dots & \sigma(1) & 1 \end{pmatrix}, \tag{7.3.16}$$

si n est impair et la permutation σ'' par

$$\sigma'' = \begin{pmatrix} 1 & 2 & 3 & 4 & \dots & 2n-1 & 2n \\ \sigma^{n-1}(1) & \sigma^{n-1}(2) & \sigma^{n-2}(1) & \sigma^{n-2}(2) & \dots & \sigma^0(1) & \sigma^0(2) \end{pmatrix}. \tag{7.3.17}$$

si n est pair.
On dénote par Σ' et Σ'' les permutations sur les éléments de $\mathbb{F}_2^{2n}$ induites respectivement par σ' et σ''. Ce résultat est une conséquence immédiate de 5.4.5

Proposition 7.3.4 *Soit C un code stable par Σ sur $\mathbb{F}^{2n}$.*
1.Si n est impair alors $\Sigma'(C)$ est un code cyclique sur $\mathbb{F}_2^{2n}$.
2. Si n est pair alors $\Sigma''(C)$ est un code 2- quasi-cyclique sur $\mathbb{F}_2^{2n}$.

Proposition 7.3.5 *Soit C un code sur R^n et C' un code sur $\mathbb{F}_2^{2n}$. On a les propriétés suivantes :*
1. C est un code $\theta-$cyclique sur R^n si et seulement si $Z(C)$ est stable par Σ.
2. C' est stable par Σ si et seulement si $\Sigma'(C)$ est un code cyclique si n est impair et $\Sigma''(C)$ est un code 2- quasi-cyclique si n est pair.

Preuve 7.3.6 *Considérons en premier lieu le cas : n est pair. Soit la permutation σ_2 donnée par*

$$\sigma_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & \dots & 2n-1 & 2n \\ \sigma^{n-1}(2) & \sigma^{n-1}(1) & \sigma^{n-2}(2) & \sigma^{n-2}(1) & \dots & \sigma^0(2) & \sigma^0(1) \end{pmatrix}. \tag{7.3.18}$$

On a pour tout b entier vérifiant $1 \leq b \leq n$ et pour tout $1 \leq j \leq 2n$,

$$\sigma_2(j) = \begin{cases} \sigma^{n-b}(2) \text{ si } j = 2b-1 \\ \sigma^{n-b}(1) \text{ si } j = 2b \end{cases}. \tag{7.3.19}$$

Un calcul simple montre que

$$\sigma_2 = \sigma^{"-1}. \tag{7.3.20}$$

Soit Σ_2 la permutation sur les éléments de $\mathbb{F}_2^{2n}$ induite par σ_2. d'après ce qui précède, il suffit de montrer que pour tout 2-quasi-cyclique code sur $\mathbb{F}_2^{2n}$ $\Sigma_2(C)$ est stable par Σ. Soit maintenant $\mathbf{x} = (x_1, x_2, ..., x_{2n}) \in C$, on a

$$\begin{aligned} \Sigma(\Sigma_2(\mathbf{x})) &= \left(x_{\sigma_2(\sigma(1)}, x_{\sigma_2(\sigma(2)}, \dots, x_{\sigma_2(\sigma(2n)}\right), \\ &= \left(x_{\sigma_2(4}, x_{\sigma_2(3)}, \dots, x_{\sigma_2(2)}, x_{\sigma_2(1)}\right), \\ &= (x_{2n-3}, x_{2n-2}, ..., x_{2n}, x_{2n-1}). \end{aligned}$$

D'un autre côté

$$\begin{aligned} \Sigma_2(T^2(\mathbf{x})) &= \Sigma_2((x_{2n-1}, x_{2n}, x_1, x_2, ..., x_{2n-3}, x_{2n-2})), \\ &= (x_{2n-3}, x_{2n-2}, ..., x_{2n}, x_{2n-1}), \\ &= \Sigma(\Sigma_2(\mathbf{x})), \end{aligned}$$

ce qu'il fallait démontrer.

7.4 Construction des codes $\theta-$cycliques sur $\mathbb{F}_2 + v\mathbb{F}_2$

On considère dans la suite l'application suivante définie dans [18].

$$\Phi : R \to \mathbb{F}_2 \times \mathbb{F}_2,$$

où $\Phi(x) = (x_1, x_2)$: les coordonnées de x dans la base B. Φ est un isomorphisme d'anneau et on peut l'étendre à R^n d'une façon naturelle. Soit C un code sur R, il vient d'après [18] qu' il existe deux codes binaires C_1 et C_2 tel que $C = \Phi^{-1}(C_1, C_2)$ et on note C par $CRT(C_1, C_2)$. Nous notons que C_1 et C_2 sont déterminés d'une façon unique pour tout $CRT(C_1, C_2)$. D'un autre côté on définit l'application

$$\begin{array}{rcl} \Psi & = & \mathbb{F}_2^{2n} \mapsto \mathbb{F}_2^n \times \mathbb{F}_2^n \\ (y_1, y_2, ..., y_{2n}) & \mapsto & ((y_1, y_3, ..., y_{2n-1}), (y_2, y_4, ..., y_{2n})) \end{array}$$

Remarque 7.4.1 *1. On a*

$$\Psi \circ Z = \Phi.$$

2. Pour tout code C sur R

$$Z(C) = \Psi^{-1}((C_1, C_2)).$$

Proposition 7.4.1 *Soit C_1 et C_2 deux codes binaires sur $\mathbb{F}_2^n$ et $C = CRT(C_1, C_2)$. Les propriétés suivantes sont équivalentes :*
1. C est un code cyclique sur R.
2. $Z(C) = \Psi^{-1}((C_1, C_2))$ est un code 2-quasi cyclique sur $\mathbb{F}_2^{2n}$.
3. C_1 et C_2 sont deux codes cycliques sur $\mathbb{F}_2^n$.

Avec les notations précédentes, on a le résultat suivant :

Proposition 7.4.2 *Soit C un code de longueur paire n sur R et C_1 et C_2 deux codes binaires sur $\mathbb{F}_2^n$ tel que $\Sigma''(Z(C)) = \Psi^{-1}((C_1, C_2))$. Les propriétés suivantes sont équivalentes :*
1. C est un code $\theta-$cyclique sur R.
2. $\Sigma''(Z(C)) = \Psi^{-1}((C_1, C_2))$ est un code 2-quasi cyclique code sur $\mathbb{F}_2^{2n}$.
3. C_1 et C_2 sont deux codes cycliques sur $\mathbb{F}_2^n$.

7.5 Construction de code $\theta-$cyclique de longueur paire

La construction est basée sur le plan suivant :
– Considérons C_1 et C_2 deux codes cycliques sur $\mathbb{F}_2^n$.

– On applique Ψ^{-1} et on obtient un 2-quasi cyclique code C' sur $\mathbb{F}_2^{2n}$.

– On applique $Z^{-1} \circ \Sigma_2$ à C'. On obtient un code $\theta-$cyclique C sur R^n.

Exemple 7.5.1 *Considérons $n = 4$. Soit C_1 et C_2 deux codes cycliques sur $\mathbb{F}_2$ et $x = (x_1, x_2, x_3, x_4) \in C_1$ $y = (y_1, y_2, y_3, y_4) \in C_2$. On a*

$$u = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4) \in \Psi^{-1}((C_1, C_2)),$$

et

$$v = (x_4, y_4, x_1, y_1, x_2, y_2, x_3, y_3) \in \Psi^{-1}((C_1, C_2)).$$

d'après ce qui précède, on a

$$\sigma_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 6 & 5 & 7 & 8 & 2 & 1 \end{pmatrix}, \tag{7.5.21}$$

et d'où

$$Z^{-1} \circ \Sigma_2(u) = (x_2 v + y_2(v+1), y_3 v + x_3(v+1), x_4 v + y_4(v+1), y_1 v + x_1(v+1))$$

et

$$Z^{-1} \circ \Sigma_2(v) = (x_1 v + y_1(v+1), y_2 v + x_2(v+1), x_3 v + y_3(v+1), y_4 v + x_4(v+1)) = \overline{u}.$$

Ainsi, $Z^{-1} \circ \Sigma_2[\Psi^{-1}((C_1, C_2))]$ est un code $\theta-$cyclique sur R^n.

7.6 Construction de codes θ-cycliques auto-duaux euclidiens sur $\mathbb{F}_2 + v\mathbb{F}_2$

Proposition 7.6.1 *Soit Γ la permutation sur les éléments de $\mathbb{F}_2^{2n}$ induite par une permutation γ sur l'ensemble $\{1...2n\}$. Soit C un code sur R^n et C_1 et C_2 deux codes binaires sur $\mathbb{F}_2^n$ tels que $\Gamma(Z(C)) = \Psi^{-1}((C_1, C_2))$. Si C_1 et C_2 sont deux codes auto-duaux pour le produit scalaire euclidien alors C est un code auto-dual sur R pour le produit scalaire euclidien.*

Preuve 7.6.2 *Le résultat est immédiat étant donné que $\Psi^{-1}((C_1, C_2))$ est un code auto-dual euclidien et compte tenu du fait qu'une permutation préserve la dualité.*

7.7 Conclusion et problèmes ouverts

Dans ce chapitre, on a introduit la décomposition d'un code C sur l'anneau R en deux codes binaires C_1 et C_2 et on a utilisé cette décomposition pour donner de nouvelles constructions des codes $\theta-$cycliques et codes $\theta-$cycliques auto-duaux sur R via l'application Z. Les résultats de ce chapitre peuvent être généralisés. En effet, on pourrait penser à trouver des conditions nécessaires et suffisantes sur les codes C_1 et C_2 pour que le code C soit un code $\theta-$cyclique auto-dual ou auto dual euclidien ou hermitien de type IV.

Troisième partie

Polynômes tordus pour la construction des codes sur $M_2(\mathbb{F}_2)$

Chapitre 8

Codes quasi-cycliques binaires comme codes cycliques sur $M_2(\mathbb{F}_2)$

8.1 Introduction

Les codes quasi cycliques sont une remarquable généralisation des codes cycliques vu qu'ils sont définis par stabilité par une puissance du shift T. On trouve dans la littérature plusieurs constructions des codes quasi cycliques. Dans [44], P.Solé et S.Ling présentent les codes quasi cycliques de longueur lm et d'index l sur $\mathbb{F}_q$ comme des codes linéaires sur l'anneau des polynômes $\mathbb{F}_q[Y]_{/Y^m-1}$. Cette construction utilise la décomposition de l'anneau $R = \mathbb{F}_q[Y]_{/Y^m-1}$ en somme directe de corps finis. Ainsi tout code quasi cyclique de longueur lm et d'index l sur $\mathbb{F}_q$ se décompose en la somme directe de " petits codes" linéaires sur un corps fini. Dans [39] K.Lally donne une autre construction des codes quasi cycliques en les identifiant aux $\mathbb{F}_q[X]$ -sous modules de $\mathbb{F}_q^l[X]_{/X^m-1}$. Récemment, P.L.Cayrel, C.Chabot et A.Necer ont présenté dans [15] une nouvelle approche. En travaillant sur les suites récurrentes à coefficients matricielles, les auteurs ont présenté les codes quasi cycliques de longueur lm et d'index l comme étant des codes cycliques sur $M_l(\mathbb{F}_q)$. Ils ont en effet montré que pour tout polynôme réversible f de $M_l(\mathbb{F}_q)[X]$, l'ensemble

$$\Omega(f) = \{u \in (F_q^l)^{\mathbb{N}} / f.u = 0\} \tag{8.1.1}$$

est un code quasi cyclique d'index l et de longueur lm sur $\mathbb{F}_q$, où m est la période du polynôme f. Cependant, les difficultés rencontrées par les auteurs dans ce travail sont d'ordre algorithmique. En effet, il paraît difficile de factoriser un polynôme de $M_l(\mathbb{F}_q)[X]$. Pour contourner le problème, les auteurs ont testé tout polynôme de $M_l(\mathbb{F}_q)[X]$ pour savoir s'il est un diviseur ou non de $X^m + 1$. Dans ce contexte, il devient impossible de construire en utilisant les bases de Gröbner des codes auto-duaux à partir d'une certaine longueur compte tenu du grand nombre de variables introduites.
Dans ce travail, nous essayons de résoudre ce problème en présentant une étude non exhaustive des polynômes de $M_2(\mathbb{F}_2)[x]$. En effet, nous démontrons des théorèmes et nous élaborons des algorithmes permettant de donner tous les diviseurs de $X^m + 1$ dans

$M_2(\mathbb{F}_2)[X]$ et réciproquement de déterminer la période de tout polynôme réversible de cet anneau de polynômes. De plus, nous donnons une nouvelle démonstration -autre que celle de [15]- pour montrer l'existence de la période d'un polynôme réversible à coefficients matriciels. Les algorithmes notés ci-dessus vont nous permettre de faciliter la construction des codes $\Omega(P)$ introduit dans [15]. Enfin, nous améliorons dans ce travail la construction des codes quasi-cycliques auto duaux d'index 2 sur $\mathbb{F}_2$ donnée dans [15]. En effet, nous présentons une nouvelle construction des codes auto duaux $\Omega(P)$ via la résolution d'un système polynomial de variables $< m$ et d'un nombre d'équations $< 2m$ au lieu d'un système à $2m$ variables et $4m$ équations dans [15]. Ce chapitre est organisé ainsi : dans la première partie nous exposons quelques préliminaires concernant les suites récurrentes linéaires à coefficients matricielles et les anneaux d'Ore. La deuxième partie est consacrée à l'étude des polynômes irréductibles de $\mathbb{F}_p[X]$ dans une extension particulière à savoir $\mathbb{F}_{p^{2^l}}[X]$. Les résultats présentés dans cette partie pris dans un cas particulier vont nous permettre de caractériser ultérieurement les diviseurs de X^m+1 dans $M_2(\mathbb{F}_2)[X]$. La troisième et la quatrième partie sont consacrées aux résultats permettant de retrouver toutes les factorisations de X^m+1 dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$ où θ est l'automorphisme de Frobenuis et dans $M_2(\mathbb{F}_2)[X]$. Enfin, dans la quatrième et la cinquième partie nous présentons de nouvelles constructions des codes $\Omega(f)$ et des codes $\Omega(f)$ auto duaux sur $\mathbb{F}_2$.
Signalons enfin que ce chapitre a fait l'objet d'un article publié [58].

8.2 Préliminaires

On considère le corps fini $\mathbb{F}_q$ à q éléments et de caractéristique p. Soit l un entier supérieur ou égal à 1. On note $A = \mathbb{F}_q^l$, $A^{\mathbb{N}}$ l'ensemble des suites d'éléments de A et $M_l(\mathbb{F}_q)$ l'ensemble des matrices carrées d'ordre l à coefficients dans $\mathbb{F}_q$. On définit avec l'application suivante une structure de $M_l(\mathbb{F}_q)[X]$-module à gauche sur $A^{\mathbb{N}}$:

$$\begin{aligned} M_l(\mathbb{F}_q)[X] \times A^{\mathbb{N}} &\longmapsto A^{\mathbb{N}} \\ (P(X), V) &\longmapsto P * V = \big(\sum_{i=0}^{degP} p_i v_{n+i} \big)_n, \end{aligned}$$

où $P(X) = \sum_{i=0}^{degP} p_i X^i$. Une suite $(u_n)_{n\in\mathbb{N}}$ de $A^{\mathbb{N}}$ est dite récurrente linéaire à coefficients matriciels s'il existe $A_1, A_2 .... A_h$ de $M_l(\mathbb{F}_q)$ telles que :

$$u_{n+h} = A_1 u_{n+h-1} + A_2 u_{n+h-2} + ...A_h u_n \quad \forall n \geq 0.$$

On définit de même pour tout $m \geq 1$ une structure de $M_l(\mathbb{F}_q)[X]$-module à gauche sur A^m avec l'application suivante :

$$\begin{aligned} M_l(\mathbb{F}_q)[X] \times A^m &\longmapsto A^m \\ (P(X), c) &\longmapsto P * c = \big(\sum_{i=0}^{degP} p_i c_{n+i \ mod\, m} \big)_{n=0,1,...m-1}, \end{aligned}$$

où $P(X) = \sum_{i=0}^{degP} p_i X_i$ et $c = (c_0, c_1, ...c_{m-1})$.
Soit $E \subset A^m$. On note

$$Ann(E) = \{P \in M_l(\mathbb{F}_q)[X] / \forall u \in E; P * u = 0\} \tag{8.2.2}$$

et

$$\Omega(P) = \{u \in A^m / P * u = 0\}. \tag{8.2.3}$$

Un polynôme P de $M_l(\mathbb{F}_q)[X]$ est dit réversible si et seulement si ses coefficients dominant et constant sont inversibles. On considère un automorphisme θ dans $\mathbb{F}_q$. On rappelle qu'on appelle anneau d'Ore ou anneau des polynômes tordus et on note $\mathbb{F}_q[X;\theta]$ l'anneau des polynômes à coefficients dans $\mathbb{F}_q$ dont l'addition est définie de la façon traditionnelle et la multiplication est définie à l'aide de la règle suivante :

$$Xa = \theta(a)X,$$

et en faisant l'extension à tous les éléments de $\mathbb{F}_q[X;\theta]$ en utilisant l'associativité et la distributivité de la multiplication par rapport à l'addition. On peut trouver une documentation complète sur ces anneaux dans [45] et [51] et sur leurs applications à la construction des codes correcteurs dans [12] et [8].
Outre les résultats donnés dans la partie Généralités concernant l'anneau des polynômes tordus $\mathbb{F}_q[X;\theta]$, on utilisera dans ce chapitre le théorème suivant :

Théorème 8.2.1 *[5] Considérons l'algèbre cyclique $A = (\mathbb{F}_{2^n}/\mathbb{F}_2, \theta, 1)$ avec $A \simeq \mathbb{F}_{2^n} \oplus \mathbb{F}_{2^n} e \oplus ...\mathbb{F}_{2^n} e^{n-1}$. Par une identification de Y et e, on a un isomorphisme d'anneaux et un $\mathbb{F}_2$-isomorphisme d'espaces vectoriels Φ*

$$M_n(\mathbb{F}_2) \simeq \mathbb{F}_{2^n}[Y,\theta]_{/Y^n+1} \tag{8.2.4}$$

8.3 Quelques propriétés sur les Polynômes irréductibles de $\mathbb{F}_p[X]$

Soit p un nombre premier. Dans cette section nous étudions les polynômes P de $\mathbb{F}_p[X]$ vérifiant la propriété suivante :

$$P = Q\overline{Q} \quad Q \in \mathbb{F}_q[X] \quad (*), \tag{8.3.5}$$

où $\mathbb{F}_q[X]$ est une extension de $\mathbb{F}_p[X]$. En effet, nous montrons que pour tout polynôme irréductible P de degré pair de $\mathbb{F}_p[X]$ il existe une infinité d' entiers q qu'on explicitera tel que P vérifie la propriété (*). De plus, nous donnons une caractérisation des autres polynômes de $\mathbb{F}_p[X]$ vérifiant cette propriété. Ces propriétés appliquées dans un cas particulier vont nous permettre ultérieurement de donner une caractérisation des diviseurs de $X^m + 1$ dans $M_l(\mathbb{F}_q)[X]$.

Lemme 8.3.1 *Soit b un entier et C_i^b la b-classe cyclotomique contenant i modulo n.*

1. Si card C_i^b est impair alors

$$C_i^b = C_i^{b^2}. \tag{8.3.6}$$

2. Si card C_i^b est pair alors

$$C_i^b = C_i^{b^2} \bigcup C_{ib}^{b^2}. \tag{8.3.7}$$

Preuve 8.3.2 *On a*

$$C_i^b = \{ib^k;\ \ 0 \leq k \leq m-1\},$$

avec

$$ib^m \equiv i[n]. \tag{8.3.8}$$

Considérons d'abord le cas où card $C_i^b = m$ est impair. On a, si $k = 2k'$ alors

$$ib^k = i(b^2)^{k'};\quad 0 \leq k' \leq \frac{m-1}{2}.$$

Si k est impair et comme

$$ib^k \equiv ib^{m+k}[n],$$

et $m+k$ est pair, on a

$$ib^{m+k} \equiv i(b^2)^{\frac{m+k}{2}}[n],$$

et ceci pour $1 \leq \frac{m+1}{2} \leq \frac{m+k}{2} \leq \frac{2m-2}{2} = m-1$.
De plus, on a

$$ib^{2m-2} \equiv ib^{m-2+m} \equiv ib^{m-2}[n].$$

Il découle donc de 8.3.8

$$i(b^2)^m = ib^{2m-2}b^2 \equiv ib^m \equiv i[n],$$

et par conséquent

$$C_i^b = \{i(b^2)^k;\ \ 0 \leq k \leq m-1\} = C_i^{b^2}.$$

Si maintenant card $C_i^b = m$ est pair, on a pour k pair($k = 2k'$)

$$ib^k = i(b^2)^{k'};\quad 0 \leq k' \leq \frac{m-2}{2}.$$

Si k est impair, il vient

$$ib^k = ib(b^2)^{k'},$$

et ceci pour $0 \leq k' \leq \frac{m-2}{2}$. De plus, 8.3.8 implique

$$(b^2)^{\frac{m-2}{2}+1} = b^m \equiv 1[n].$$

D'où

$$\begin{aligned} C_i^b &= \{i(b^2)^k;\ \ 0 \leq k \leq \frac{m-2}{2}\} \bigcup \{ib(b^2)^k;\ \ 0 \leq k \leq \frac{m.-2}{2}\}, \\ &= C_i^{b^2} \bigcup C_{ib}^{b^2}. \end{aligned}$$

Proposition 8.3.3 *Soit b un entier et C_i^b la b-classe cyclotomique contenant i modulo n. On pose*

$$card(C_i^b) = m = 2^l(2l'+1). \tag{8.3.9}$$

On a les résultats suivants

i. Si $l = 0$ alors

$$C_i^b = C_i^{b^{2^z}} \qquad \forall z \geq 1. \tag{8.3.10}$$

ii. Si $l = 1$ alors

$$C_i^b = C_i^{b^{2^z}} \bigcup C_{ib}^{b^{2^z}} \qquad \forall z \geq 1. \tag{8.3.11}$$

iii. Si $l \geq 2$ alors on a

$$C_i^b = (\bigcup_{k=0}^{a} C_{ib^{2k}}^{b^{2^l}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{2k+1}}^{b^{2^l}}), \tag{8.3.12}$$

où $a = \sum_{i=0}^{l-2} 2^i$, card $C_{ib^{2k}}^{b^{2^l}}$ et $C_{ib^{2k+1}}^{b^{2^l}}$ sont impairs. De plus, on a pour tout $z \geq l$

$$C_{ib^{2k}}^{b^{2^l}} = C_{ib^{2k}}^{b^{2^z}}, \tag{8.3.13}$$

et

$$C_{ib^{2k+1}}^{b^{2^l}} = C_{ib^{2k+1}}^{b^{2^z}}. \tag{8.3.14}$$

Preuve 8.3.4 *i. On peux montrer suite à une récurrence facile et en utilisant 8.3.6 que l' on ait pour pour $l = 0$*

$$C_i^2 = C_i^4 = C_i^{16} = ...C_i^{2^{2^z}}.$$

ii. Le résultat pour $l = 1$ est une conséquence immédiate de 8.3.7 et de 8.3.10.
iii. Montrons par récurrence le cas $l \geq 2$. Notons que le résultat est vérifiable facilement pour $l = 2$ et supposons qu'il est vrai à l'ordre l. Prenons $m = 2^{l+1}(2l'+1)$, on a donc

$$m = 2.2^l(2l'+1) = 2m'.$$

Il vient d'après 8.3.7

$$C_i^b = C_i^{b^2} \bigcup C_{ib}^{b^2},$$

où

$$card(C_i^{b^2}) = card(C_{ib}^{b^2}) = m' = 2^l(2l'+1).$$

Ainsi, on a via l'hypothèse de récurrence

$$C_i^{b^2} = (\bigcup_{k=0}^{a} C_{ib^{4k}}^{b^{2^{l+1}}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{4k+2}}^{b^{2^{l+1}}}),$$

et

$$C_{ib}^{b^2} = (\bigcup_{k=0}^{a} C_{ib^{4k+1}}^{b^{2^{l+1}}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{4k+3}}^{b^{2^{l+1}}}),$$

où $a = \sum_{i=0}^{l-2} 2^i$ *et le cardinal de* $C_{ib^{4k+j}}^{b^{2^{l+1}}}$ *est impair pour* $0 \leq j \leq 3$.

Par conséquent

$$\begin{aligned} C_i^b &= (\bigcup_{k=0}^{a} C_{ib^{2.2k}}^{b^{2^{l+1}}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{2.(2k+1)}}^{b^{2^{l+1}}}) \\ &\cup (\bigcup_{k=0}^{a} C_{ib^{2.(2k+1)}}^{b^{2^{l+1}}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{2.(2k+1)+1}}^{b^{2^{l+1}}}). \end{aligned}$$

En remarquant que pour $0 \leq k \leq a = \sum_{i=0}^{l-2} 2^i$, *on a*

$$0 \leq 2k \leq \sum_{i=1}^{l-1} 2^i,$$

et

$$0 \leq 2k+1 \leq \sum_{i=1}^{l-1} 2^i + 1 = \sum_{i=0}^{l-1} 2^i.$$

D'où

$$(\bigcup_{k=0}^{a} C_{ib^{2.2k}}^{b^{2^{l+1}}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{2.(2k+1)}}^{b^{2^{l+1}}}) = (\bigcup_{k=0}^{a'} C_{ib^{2k}}^{b^{2^{l+1}}}),$$

et

$$(\bigcup_{k=0}^{a} C_{ib^{2.(2k+1)}}^{b^{2^{l+1}}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{2.(2k+1)+1}}^{b^{2^{l+1}}}) = (\bigcup_{k=0}^{a'} C_{ib^{2k+1}}^{b^{2^{l+1}}}),$$

où

$$a' = \sum_{i=0}^{l-1} 2^i.$$

Ainsi on a

$$C_i^b = (\bigcup_{k=0}^{a'} C_{ib^{2k}}^{b^{2^{l+1}}}) \bigcup (\bigcup_{k=0}^{a'} C_{ib^{2k+1}}^{b^{2^{l+1}}}),$$

ce qu'on veut démontrer.

Signalons enfin que comme card $C_{ib^{2k}}^{b^{2^l}}$ *et* $C_{ib^{2k+1}}^{b^{2^l}}$ *sont impairs, 8.3.13 et 8.3.14 sont des conséquences immédiates du 8.3.10.*

Lemme 8.3.5 *Soit* $\mathbb{F}_q$ *un corps fini de caractéristique* p. *Les deux classes cyclotomiques* C_i^q *et* C_{ip}^q *correspondent à deux polynômes irréductibles de* $\mathbb{F}_q[X]$ *conjugués deux à deux.*

Preuve 8.3.6 *On a*

$$C_i^q = \{iq^k;\ \ 0 \leq k \leq m-1\}$$

et

$$C_{ip}^q = \{ipq^k;\ \ 0 \leq k \leq m-1\}.$$

Soit f et g les polynômes correspondant à C_i^q et C_{ip}^q et w une racine primitive nieme de l'unité dans $\mathbb{F}_q$. On a

$$f = \prod_{k=0}^{m-1} (X - w^{iq^k})$$

et

$$g = \prod_{k=0}^{m-1} (X - (w^{iq^k})^p) = \overline{f},$$

ce qu'il faut démontrer.

Proposition 8.3.7 *Soient p un entier premier et P un polynôme irréductible de $\mathbb{F}_p[X]$. On a*
i. Si $\deg P$ est impair alors P est irréductible dans $F_{p^{2^z}}[X]$ pour tout $z \geq 1$.
ii. Si $\deg P$ est pair c'est à dire si $\deg P = 2^l(2l'+1)$, on a
1.Si $l = 1$ alors

$$P = Q\overline{Q}, \tag{8.3.15}$$

où Q est irréductible dans $F_{p^{2^z}}[X]$ pour tout $z \geq 1$.
2. Si $l \geq 2$ alors

$$P = Q_1 Q_2 ....Q_a.\overline{Q_1 Q_2}....\overline{Q_a} = Q\overline{Q}, \tag{8.3.16}$$

où les Q_i sont des polynômes irréductibles dans $F_{p^{2^z}}[X]$ pour tout $z \geq l$ et où $a = \sum_{i=0}^{l-2} 2^i$.

Preuve 8.3.8 *Comme P est irréductile de $\mathbb{F}_p[X]$ alors P correspond à une seule 2- classe cyclotomique C_i^p et on a $\deg P = card(C_i^p)$. Ainsi, dans le cas où $\deg P = card(C_i^p)$ est impair, on a d'après 8.3.10, P correspond à une seule p^{2^z}- classe cyclotomique $C_i^{p^{2^z}}$ et donc P est irréductible dans $F_{p^{2^z}}[X]$ pour tout $z \geq 1$.*
Si maintenant $\deg P = card(C_i^p) = 2^l(2l'+1)$, on a pour $l = 1$

$$P = Q\overline{Q},$$

où Q est irréductible dans $F_{p^{2^z}}[X]$ pour tout $z \geq 1$ et où l'on a tenu compte de 8.3.11. Pour le cas $l \geq 2$, il vient de la propriété 8.3.12 que

$$C_i^b = (\bigcup_{k=0}^{a} C_{ib^{2k}}^{b^{2^z}}) \bigcup (\bigcup_{k=0}^{a} C_{ib^{2k+1}}^{b^{2^z}}),$$

et du fait que toutes les classes $C_{ib^{2k}}^{b^{2^z}}$ et $C_{ib^{2k+1}}^{b^{2^z}}$ ont un cardinal impair qu 'il existe $2a$ polynômes correspondant chacun à une de ces classes et donc tous irréductibles dans $F_{p^{2^z}}[X]$ et qui sont conjugués deux à deux. Par suite, on a 8.3.16.

Corollaire 8.3.9 *Soit p un entier premier et P un polynôme irréductible de $\mathbb{F}_p[X]$. Pour tout $n \geq 1$ il existe un polynôme $P_1 \in F_{p^{2^n}}[X]$ tel que*

$$P = P_1\overline{P_1}. \tag{8.3.17}$$

Preuve 8.3.10 *Le résultat est immédiat compte tenu de 8.3.16.*

Proposition 8.3.11 *Soient p un nombre premier, $h \in \mathbb{F}_p[X]$ et $q = p^{2^z}$. Il existe $f \in F_q[X]$ tel que*

$$h = f\overline{f}$$

si et seulement si tout facteur irréductible de h sur $\mathbb{F}_p[X]$ de degré impair a une multiplicité paire.

Preuve 8.3.12 *Supposons que $h = f\ \overline{f}$ et soit P un facteur irréductible de h sur $\mathbb{F}_p[X]$. D'où*

$$P|h = f\overline{f}.$$

Comme P est irréductible, on a $P|f$ ou $P|\overline{f}$. Ainsi, si $P|f$ avec une multiplicité m alors $\overline{P} = P|\overline{f}$ avec la même multiplicité m et par conséquent $P|h$ avec une multiplicité $2m$.
Inversement si tout facteur irréductible de h sur $\mathbb{F}_p[X]$ de degré impair a une multiplicité paire, alors l'écriture de h en facteurs irréductibles sur $\mathbb{F}_p[X]$ s'écrit de cette façon :

$$h = \prod_{\substack{P\ irréductible\ sur \mathbb{F}_p[X] \\ \deg P\ impair}} P^{\alpha_p} \prod_{\substack{Q\ irréductible\ sur \mathbb{F}_p[X] \\ \deg Q\ pair}} Q^{\beta_q}$$

Or, il résulte du théorème 4.4 que P est irréductible dans $F_q[X]$. Ainsi, h s'écrit

$$\begin{aligned} h &= \Big(\prod_{P\ irréductible\ sur F_q[X]} P^{m_p} \prod_{q \in F_q[X]} q^{\beta_q}\Big)\Big(\prod_{P\ irréductible\ sur F_q[X]} P^{m_p} \prod_{q \in F_q[X]} \overline{q}^{\beta_q}\Big), \\ &= f\overline{f}, \end{aligned}$$

où l'on a tenu compte de 8.3.17 et du fait que $\alpha_p = 2m_p$.

8.4 Factorisation dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$

Dans toute la suite, on considère $\mathbb{F}_4$ le corps fini à 4 éléments et de caractéristique 2 et θ l'automorphisme de Frobenius dans $\mathbb{F}_4$.

Lemme 8.4.1 *On définit dans $\mathbb{F}_4[X]$ l'application*

$$\begin{aligned} \Theta : \quad \mathbb{F}_4[X] &\mapsto \mathbb{F}_4[X] \\ P = \sum_{i=0}^{n} a_i X^i &\mapsto \sum_{i=0}^{n} \theta(a_i) X^i = \overline{P} \end{aligned}$$

L'application Θ est un automorphisme d'anneaux vérifiant $\Theta = \Theta^{-1}$. De plus on a
i. Si P/Q alors

$$\Theta(\frac{P}{Q}) = \frac{\Theta(P)}{\Theta(Q)}.$$

ii.

$$P\overline{P} \in \mathbb{F}_2[X].$$

Preuve 8.4.2 *La fonction Θ est bijective compte tenu du fait que la fonction θ l'est aussi. Soient $P, Q \in \mathbb{F}_4[X]$ avec $P = \sum_{i=0}^{n} a_i X^i$, $Q = \sum_{j=0}^{p} b_j X^j$, on a*

$$\Theta(PQ) = \sum_{k=0}^{n+p} \theta(\sum_{i=0}^{k} a_i b_{k-i}) X^k = \sum_{k=0}^{n+p} (\sum_{i=0}^{k} \theta(a_i)\theta(b_{k-i}) X^k = \Theta(P)\Theta(Q).$$

Les derniers résultats sont donc immédiats.

Proposition 8.4.3 *Soient $f_1, f_2 \in \mathbb{F}_4[X] \setminus \{0\}$ tel que pgcd $(f_1, f_2) = 1$. On a*

$$f_1 + f_2 Y | X^m + 1 \tag{8.4.18}$$

dans $\mathbb{F}_4[Y, \theta]_{/Y^2+1}[X]$, si et seulement si

$$f_1\overline{f_1} + f_2\overline{f_2} | X^m + 1 \tag{8.4.19}$$

dans $\mathbb{F}_2[X]$.

La proposition 8.4.3 est une version simple du théorème suivant que nous démontrons.

Théorème 8.4.1 *Soient $f_1^{'}, f_2^{'} \in \mathbb{F}_4[X] \setminus \{0\}$ et $g_1, g_2 \in \mathbb{F}_4[X]$. On a*

$$X^m + 1 \ = (f_1^{'} + f_2^{'} Y)(g_1 + g_2 Y) \tag{8.4.20}$$

dans $\mathbb{F}_4[Y, \theta]_{/Y^2+1}[X]$, si et seulement si

$$X^m + 1 \ = fg \tag{8.4.21}$$

dans $\mathbb{F}_2[X]$, avec

$$f \ = \ f_1\overline{f_1} + f_2\overline{f_2}, \tag{8.4.22}$$
$$g \ = \ AB, \tag{8.4.23}$$

où

$$A = f_1^{'} \wedge f_2^{'} \tag{8.4.24}$$

dans $\mathbb{F}_4[X]$, $f_1^{'} = Af_1$, $f_2^{'} = Af_2$ et

$$B = \frac{\overline{g_2}}{\overline{f_2}} = \frac{g_1}{\overline{f_1}}. \tag{8.4.25}$$

Preuve 8.4.4 *Il vient d'après 8.4.20 et 8.4.24 que*

$$X^m + 1 = Af_1g_1 + Af_2\overline{g_2} + (Af_1g_2 + Af_2\overline{g_1})Y. \tag{8.4.26}$$

D'où

$$X^m + 1 = A(f_1g_1 + f_2\overline{g_2}),$$

et

$$f_1g_2 + f_2\overline{g_1} = 0.$$

Comme $f_1g_2 = f_2\overline{g_1}$, *il vient compte tenu du fait que* $f_1 \wedge f_2 = 1$

$$f_1 \mid \overline{g_1} \quad , \quad f_2 \mid g_2,$$

et

$$\overline{g_1} = \frac{f_1g_2}{f_2}.$$

On peut donc poser

$$B = \frac{\overline{g_2}}{\overline{f_2}} = \frac{g_1}{\overline{f_1}},$$

où l'on a utilisé le fait que $\overline{\overline{g_1}} = \overline{(\frac{f_1g_2}{f_2})} = \frac{\overline{f_1}\overline{g_2}}{\overline{f_2}} = g_1$. *En remplaçant dans 8.4.26, on a*

$$X^m + 1 \ = AB(f_1\overline{f_1} + f_2\overline{f_2}) = fg.$$

Inversement, si

$$X^m + 1 \ = fg$$

dans $\mathbb{F}_4[X]$, *avec* f, g, *et* B *vérifiant 8.4.22, 8.4.23 et 8.4.25 et* A *un polynôme quelconque de* $\mathbb{F}_4[X]$, *on a*

$$(Af_1 + Af_2Y)(g_1 + g_2Y) = A(f_1g_1 + f_2\overline{g_2} + (f_1g_2 + f_2\overline{g_1})Y).$$

Or, 8.4.25 implique

$$g_1 = \overline{(\frac{f_1g_2}{f_2})} \in F_4[X].$$

De plus, il résulte du fait que la fonction Θ *est bijective, que* $\frac{f_1g_2}{f_2} \in F_4[X]$ *et par conséquent*

$$\overline{g_1} = \overline{\overline{(\frac{f_1g_2}{f_2})}} = \frac{f_1g_2}{f_2}.$$

Ainsi on a, $f_1g_2 + f_2\overline{g_1} = 0$ *et*

$$\begin{aligned}(Af_1 + Af_2Y)(g_1 + g_2Y) &= A(f_1g_1 + f_2\overline{g_2}),\\ &= AB(f_1\overline{f_1} + f_2\overline{f_2}),\\ &= fg = X^m + 1.\end{aligned}$$

Dans la suite de cette section, nous présentons deux petits résultats, le premier donne une factorisation de X^m+1 dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$ à partir d'une factorisation dans $\mathbb{F}_4[X]$ et non dans $\mathbb{F}_2[X]$ et le deuxième donne tous les diviseurs explicites de X^m+1 dans $\mathbb{F}_2[Y,\theta]_{/Y^2+1}[X]$.

Corollaire 8.4.5 *Soient $f,g \in \mathbb{F}_4[X] \setminus \mathbb{F}_2[X]$. On a*

$$X^m+1 = fg \quad dans \quad \mathbb{F}_4[X]$$

si et seulement si

$$X^m+1 \ = fg \quad ou \qquad X^m+1 \ = (fY)(\overline{g}Y)$$

dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$.

Preuve 8.4.6 *La condition étant clairement nécessaire, on montrera uniquement la condition suffisante. Pour cela considérons f_1+f_2Y un diviseur de X^m+1 dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$. Si f_1 et f_2 sont non nuls, on pose $A = f_1^{'} \wedge f_2^{'}$. On a alors d'après le théorème 8.4.26 $f,g \in \mathbb{F}_2[X]$, ce qui contredit l'hypothèse. Ainsi, on a soit*

$$f_2 = 0 \text{ et donc } g_2 = 0 \text{ et } X^m+1 \ = f_1g_1,$$

soit

$$f_1 = 0 \text{ et donc } g_1 = 0 \text{ et } X^m+1 \ = f_2\overline{g_2}.$$

Corollaire 8.4.7 *Soient $f,g \in \mathbb{F}_2[X]$ tel que*

$$X^m+1 = fg.$$

Alors, pour tout $A,B \in F_2[X]$ tel que $A+B=1$ et

$$f_1 = Af \ \ f_2 = Bf \ \ g_1 = Af \ \ g_2 = Bf$$

on a,

$$X^m+1 \ = (f_1+f_2Y)(g_1+g_2Y)$$

dans $\mathbb{F}_2[Y,\theta]_{/Y^2+1}[X]$.

Preuve 8.4.8 *Soient $A,B \in F_2[X]$ tel que $A+B=1$, $f_1 = Af$, $f_2 = Bf$, $g_1 = Af$ et $g_2 = Bf$. On a*

$$\begin{aligned}(f_1+f_2Y)(g_1+g_2Y) &= (f_1g_1+f_2g_2)+(f_2g_1+f_1g_2)Y,\\ &= (A^2+B^2)fg+(ABfg+ABfg)Y,\\ &= (A+B)^2fg = fg.\end{aligned}$$

8.5 Factorisation dans $M_2(\mathbb{F}_2[X])$

Rappelons en début de cette section qu'il existe d'après 8.2.4 un isomorphisme Φ entre $M_2(\mathbb{F}_q)$ et $\mathbb{F}_{q^2}[Y,\theta]_{/Y^2+1}$. On fera l'extension de cet isomorphisme à l'application suivante

$$\begin{aligned} \Psi : M_2(\mathbb{F}_q)[X] &\mapsto \mathbb{F}_{q^2}[Y,\theta]_{/Y^2+1}[X] \\ P = \sum_{i=0}^{n} P_i X^i &\mapsto \sum_{i=0}^{n} \Phi(P_i) X^i. \end{aligned}$$

On a le résultat suivant :

Proposition 8.5.1 *L'application Ψ est un isomorphisme d'anneaux et un $\mathbb{F}_q$-isomorphisme d'espaces vectoriels.*

Preuve 8.5.2 *La fonction Ψ est bijective par construction. Soient maintenant $P, Q \in \mathbb{F}_{q^2}[Y,\theta]_{/Y^2+1}[X]$ avec $P = \sum_{i=0}^{n} P_i X^i$, $Q = \sum_{j=0}^{p} Q_j X^j$. On a compte tenu du fait que Φ est isomorphisme d'anneaux*

$$\begin{aligned} \Psi(PQ) &= \sum_{k=0}^{n+p} \Phi(\sum_{i=0}^{k} P_i Q_{k-i}) X^k, \\ &= \sum_{k=0}^{n+p} (\sum_{i=0}^{k} \Phi(P_i)\Phi(Q_{k-i})) X^k = \Psi(P)\Psi(Q). \end{aligned}$$

On montre de la même façon que Ψ est un $\mathbb{F}_q$-isomorphisme d'espaces vectoriels.

Dans le cas particulier où $q = 2$, on définit Φ^{-1} par la substitution dans $P_i = a_i + b_i Y$, par

$$Y \mapsto J = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{8.5.27}$$

$$w \mapsto W = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}. \tag{8.5.28}$$

On a le résultat suivant :

Théorème 8.5.1 *Soit $P \in M_2(\mathbb{F}_2)[X]$. Les propriétés suivantes sont équivalentes :*

$$\begin{aligned} &i) \quad P | X^m + 1 \quad dans \quad M_2(\mathbb{F}_2)[X]. \\ &ii) \quad f_1 + f_2 Y | X^m + 1 \quad dans \quad \mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]. \\ &ii) \quad f_1' \overline{f_1'} + f_2' \overline{f_2'} | X^m + 1 \quad dans \quad \mathbb{F}_2[X]. \end{aligned}$$

Où f_1 et f_2 sont données par

$$\Psi(P) = f_1 + f_2 Y,$$

et f_1' et f_2' sont les deux polynômes premiers entre eux obtenus par les divisions respectives de f_1 et f_2 par $f_1 \wedge f_2$.

Preuve 8.5.3 *Les propriétés ii et iii sont clairement équivalentes d'après le théorème 8.4.26. Il nous reste donc juste à démontrer l'équivalence $i \Leftrightarrow ii$. On a*

$$\begin{aligned} P|X^m+1 \quad dans \quad M_2(\mathbb{F}_2)[X] &\Leftrightarrow X^m+1 = P.Q \quad dans \quad M_2(\mathbb{F}_2)[X], \\ &\Leftrightarrow \Psi(X^m+1) = \Psi(P).\Psi(Q), \\ &\Leftrightarrow \Phi(I)X^m + \Phi(I) = X^m+1 = (f_1+f_2Y).\Psi(Q), \\ &\Leftrightarrow f_1+f_2Y|X^m+1 \quad dans \ \ \mathbb{F}_4[Y,\theta]_{/Y^2+1}[X], \end{aligned}$$

où l'on a utilisé le fait que Ψ est un morphisme d'anneaux.

Lemme 8.5.4 *Soit $P \in M_2(\mathbb{F}_2)[X]$ vérifiant $\Psi(P) = f_1 + f_2Y$. P est un polynôme réversible dans $M_2(\mathbb{F}_2)[X]$ si et seulement si $deg(f_1) \neq deg(f_2)$ et l'un seulement des deux coefficients constants de f_1 ou f_2 est nul.*

Preuve 8.5.5 *On pose*

$$f_1 + f_2Y = \sum_{i=0}^{n} (a_i + b_iY)X^i.$$

Il découle donc de la définition de Ψ que

$$\Psi^{-1}(f_1 + f_2Y) = \sum_{i=0}^{n} \Phi^{-1}(a_i + b_iY)X^i.$$

Or, on peut vérifier facilement que pour tout $a, b \in \mathbb{F}_4$: si a et b sont tous deux non nuls alors $\Phi^{-1}(a+bY)$ est une matrice non inversible. Ce qui termine la démonstration.

Le résultat suivant mentionnant que tout polynôme réversible de $M_2(\mathbb{F}_2)[X]$ possède un exposant a été donné dans [15]. Nous présentons dans ce qui suit une nouvelle démonstration plus simple de cette existence.

Proposition 8.5.6 *Soit P un polynôme réversible de $M_2(\mathbb{F}_2)[X]$ de degré ≥ 1, il existe alors un entier $m \geq 1$ tel que $P|X^m+1$ dans $M_2(\mathbb{F}_2)[X]$.*

Preuve 8.5.7 *On a*

$$\Psi(P) = f_1 + f_2Y.$$

Posons

$$h = f_1'\overline{f_1'} + f_2'\overline{f_2'},$$

où f_1' et f_2' sont les deux polynômes premiers entre eux obtenus par les divisions respectives de f_1 et f_2 par $f_1 \wedge f_2$.
Comme P est réversible, il découle du lemme 8.5.4 que $deg(P) = sup(deg(f_1), deg(f_2)) \geq 1$.
On a par conséquent, $deg(h) \geq 1$ et le coefficient constant de h est non nul. Ainsi, il existe un entier $m \geq 1$ tel que

$$h|X^m+1 \ \ dans \ \ \mathbb{F}_2[X].$$

Il résulte donc de 8.5.1 que

$$P|X^m+1 \ \ dans \ \ M_2(\mathbb{F}_2)[X].$$

le résultat suivant est donc une conséquence immédiate.

Corollaire 8.5.8 *Soit P un polynôme réversible de $M_2(\mathbb{F}_2)[X]$ de degré ≥ 1 où $\Psi(P) = f_1 + f_2Y$. La période de P est égale à la période du polynôme de $\mathbb{F}_2[X]$*

$$h = f_1'\overline{f_1'} + f_2'\overline{f_2'},$$

où f_1' et f_2' sont les deux polynômes premiers entre eux obtenus par les divisions respectives de f_1 et f_2 par $f_1 \wedge f_2$.

En utilisant les théorèmes 9.24 et 9.25 de [62] -explicitant la calcul de la période d'un polynôme de $\mathbb{F}_q[X]$- nous avons élaboré un programme 2 de l'annexe sous Magma permettant de construire des polynômes à coefficiens matricielles de $M_2(\mathbb{F}_2)[X]$ et de déterminer leurs périodes.

Exemple 8.5.9 *Dans ce qui suit, nous présentons quelques polynômes à coefficients matriciels et leurs périodes obtenus via l'application du programme 2.*

1. $P = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^5 + \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} X^4 + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} X^3 + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} X^2 + \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} X + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ *de période 868.*

2. $P = X^5 + \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} X^4 + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^3 + \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} X + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ *de période 1022.*

3. $P = X^4 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^3 + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} X^2 + \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} X + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ *de période 510.*

4. $P = X^4 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^3 + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ *de période 30.*

5. $P = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^4 + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^3 + \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ *de période 510.*

6. $X^6 + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} X^5 + \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} X^4 + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} X^3 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} X + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ *de période 60.*

7. $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^7 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^6 + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^5 + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} X^4 + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} X^3 + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ *de période 16380.*

8. $X^7 + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^6 + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} X^4 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^3 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X^2 + \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} X + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ *de période 3556.*

9. $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^8 + X^7 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^6 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X^5 + \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} X^4 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X^3 + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} X + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ *de période 3720.*

8.6 Construction des codes $\Omega(P)$

Théorème 8.6.1 *[15] Soient P et Q deux polynômes de $M_2(\mathbb{F}_2)[X]$ tel que $X^m+1 = PQ$ et $\Omega(P)$ est l'ensemble défini par 8.1.1. L'ensemble $\Omega(P)$ est un code quasi-cyclique de longueur $2m$ et d'index 2 de $\mathbb{F}_2$.*

Pour plus d'amples connaissances sur les codes $\Omega(P)$ consulter [15].

Théorème 8.6.2 *Soit f un diviseur de degré pair de $X^m + 1$ dans $\mathbb{F}_2[X]$. Pour tout $f_1 \in \mathbb{F}_4[X]$ de deg $\leq m$, il existe f_2 dans $\mathbb{F}_4[X]$ tel que*

$$f_1 + f_2 Y | X^m + 1 \quad dans \quad \mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$$

si et seulement si tout facteur irréductible de $\mathbb{F}_4[X]$ et de degré impair du polynôme

$$h = f + f_1\overline{f_1}$$

a une multiplicité paire.

Preuve 8.6.1 *Il résulte de la proposition 8.3.11 que le fait que tout facteur irréductible du polynôme h de degré impair a une multiplicité paire est une condition nécessaire et suffisante pour que le polynôme h s'écrit sous la forme $f_2\overline{f_2}$.*
Ainsi le polynôme $f_1\overline{f_1} + f_2\overline{f_2}$ est un diviseur de $X^m + 1$ dans $\mathbb{F}_2[X]$. Le résultat est immédiat compte tenu du théorème 8.4.26.

Dans la suite nous présentons deux algorithmes permettant d'effectuer la première étape de la construction d'un code quasi- cyclique de la forme $\Omega(f)$.
Le premier algorithme permettra la détermination d' une factorisation de $X^m + 1$ dans $M_2(\mathbb{F}_2)[X]$ et par conséquent la construction d'un code de longueur m connue. Quant au deuxième algorithme, il déterminera la période d'un polynôme f de $M_2(\mathbb{F}_2)[X]$ et il permettra donc la construction d'un code de dimension connue.

Algorithme 8.6.2 *1. On factorise $X^m + 1$ sur $\mathbb{F}_2[X]$.*

2. Pour tout diviseur f de degré pair de $X^m + 1$ dans $\mathbb{F}_2[X]$, on fait
i)Pour tout $f_1 \in \mathbb{F}_4[X]$ tel que deg $f_1 < m$, on pose

$$h = f + f_1\overline{f_1}.$$

ii) On factorise h sur $F_4[X]$.
iii) Pour tout diviseur l de h de degré $\frac{degh}{2}$, on vérifie si

$$l/h.$$

Si $h = l\bar{l}$ alors on prend $l = f_2$, sinon, on arrête.
Ainsi on obtient un polynôme $P = \sum_{i=0}^{n} P_i X^i = f_1 + f_2 Y$, diviseur de $X^m + 1$ dans $F_4[Y,\theta]_{/Y^2+1}[X]$ et vérifiant pour tout $1 \leq i \leq n$

$$P_i = a_i + b_i Y,$$

où les a_i et b_i sont dans $F_4 = \mathbb{F}_2(w)$.
Pour tout $1 \leq i \leq n$, on substitue dans $P_i = a_i + b_i Y$

$$Y \mapsto j = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

$$w \mapsto W = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

3. On obtient un polynôme $\sum_{i=0}^{n} A_i X^i$ à coefficients matriciels et diviseur de X^m+1 dans $M_2(\mathbb{F}_2)[X]$.

Remarque 8.6.1 *Pour une factorisation complète de $X^m + 1$ dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$, on peut continuer l'algorithme de la façon suivante :*
pour tout diviseur B de $\frac{X^m+1}{f}$ dans $\mathbb{F}_4[X]$, on prends :

$$\begin{aligned} g_1 &= \overline{f_1} B, \\ g_2 &= f_2 \overline{B}. \end{aligned}$$

6.On calcule $A = \frac{X^m+1}{fB}$. On obtient

$$X^m + 1 = (Af_1 + Af_2 Y)(g_1 + g_2 Y).$$

Dans ce qui suit nous présentons une table des meilleurs codes $\Omega(P)$ obtenus via l'application des programmes **3** et **4** de l'annexe.

f_1	f_2	P	code
$X^4 + X^2$	$X^3 + X^2 + X + 1$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X^3 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X^2 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	(12,8,3)
$X^3 + X^2$	$X^2 + 1$	$X^3 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X^2 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	(12,6,4)
$X^2 + X + w^2$	X	$X^2 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	(14,4,7)
$w^2.X^2 + X + 1$	X	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	(14,4,7)
$w^2.X^5 + w^2.X^4 + X^3 + X + w$	$X^4 + X^2 + X$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} X^5 + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} X^4 + X^3 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$	(14,10,3)
$w.X^5 + X^4 + X^2 + w^2.X + w^2$	$X^4 + X^3 + X$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^5 + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} X^4 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X^3 + X^2 + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} X + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	(14,10,3)
$w.X^3$	$X^2 + w$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} X^3 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X^2 + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	(16,6,6)

L'algorithme suivant permet la construction d'un code $\Omega(P)$ de dimension k donnée.

Algorithme 8.6.3 *On considère $f_1 \in \mathbb{F}_4[X]$ de degré k.*
1. Pour tout $f_2 \in \mathbb{F}_4[X]$ de degré $< k$, on pose

$$f = f_1\overline{f_1} + f_2\overline{f_2}.$$

2. On détermine la période m de f.
3. On obtient que le polynôme $P = \sum_{i=0}^{n} P_i X^i = f_1 + f_2 Y$ est un diviseur de $X^m + 1$ dans $\mathbb{F}_4[Y,\theta]_{/Y^2+1}[X]$ vérifiant pour tout $1 \leq i \leq n$,

$$P_i = a_i + b_i Y,$$

où les a_i et b_i sont dans $F_4 = \mathbb{F}_2(w)$.
4. Pour tout $1 \leq i \leq n$, on substitue dans $P_i = a_i + b_i Y$

$$\begin{aligned} Y &\mapsto j = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\ w &\mapsto W = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}. \end{aligned}$$

4. On obtient alors le polynôme $\sum_{i=0}^{n} A_i X^i$ à coefficients matriciels et diviseur de $X^m + 1$ dans $M_2(\mathbb{F}_2)[X]$.

8.7 Construction des codes $\Omega(P)$ auto duaux

On a le résultat suivant :

Théorème 8.7.1 *[15] Si $X^m + 1 = PQ$ dans $M_2(\mathbb{F}_q)[X]$, alors*

$$\Omega(P)^{\perp} = \Omega({}^tQ^*). \tag{8.7.29}$$

Lemme 8.7.1 *Soit $A \in M_2(\mathbb{F}_2)$. Si*

$$\Phi(A) = \alpha + \beta Y,$$

alors

$$\Phi({}^tA) = \alpha + \overline{\beta} Y.$$

Preuve 8.7.2 *Avec les notations 8.4.22 et en notant I la matrice identité de $M_2(\mathbb{F}_2)$, on a que (I, W, J, WJ) est une base de $M_2(\mathbb{F}_2)$. Ainsi, il existe a, b, c et d dans $\mathbb{F}_2$ tels que*

$$A = aI + bW + cJ + dWJ.$$

Il résulte du fait que Φ est un $\mathbb{F}_2$-isomorphisme d'espaces vectoriels et un isomorphisme d'anneaux que

$$\Phi(A) = a + bw + (c + dw)Y = \alpha + \beta Y,$$

et

$$\begin{aligned}\Phi({}^tA) &= a\Phi({}^tI) + b\Phi({}^tW) + c\Phi({}^tJ) + d\Phi({}^t(WJ)),\\ &= a\Phi(I) + b\Phi(W) + c\Phi(J) + d\Phi({}^tJ{}^tW),\\ &= a\Phi(I) + b\Phi(W) + c\Phi(J) + d\Phi(J)\Phi(W),\\ &= a + bw + cY + dYw = a + bw + (c + dw^2)Y = \alpha + \overline{\beta}Y,\end{aligned}$$

où l'on a utilisé le fait que $Yw = w^2Y$ *dans* $\mathbb{F}_4[Y,\theta]_{/Y^2+1}$.

Proposition 8.7.3 *Soit* P *un polynôme de degré* ≥ 1 *de* $M_2(\mathbb{F}_q)[X]$. *On a*

$$X^m + 1 = P{}^tP^* \quad dans \quad M_2(\mathbb{F}_q)[X], \tag{8.7.30}$$

si et seulement si

$$X^m + 1 \ = (f_1 + f_2Y)(f_1^* + \overline{f_2^*}Y) \quad dans \quad \mathbb{F}_4[Y,\theta]_{/Y^2+1}[X], \tag{8.7.31}$$

où f_1 *et* f_2 *sont données par :*

$$\Psi(P) = f_1 + f_2Y.$$

Preuve 8.7.4 *Montrer ce résultat revient à montrer que*

$$\Psi({}^tP^*) = f_1^* + \overline{f_2^*}Y.$$

Posons $P = \sum_{i=0}^n A_iX^i$, *ainsi* $P^* = \sum_{i=0}^n A_{n-i}X^i$. *Il résulte de la définition de l'application* Ψ *que*

$$\begin{aligned}\Psi({}^tP^*) &= \sum_{i=0}^n \Phi({}^tA_{n-i})X^i,\\ &= \sum_{i=0}^n (a_{n-i} + \overline{b_{n-i}}Y)X^i,\\ &= \sum_{i=0}^n a_{n-i}X^i + (\sum_{i=0}^n \overline{b_{n-i}}X^i)Y,\\ &= f_1^* + \overline{f_2^*}Y,\end{aligned}$$

où l'on a noté $f_1 = \sum_{i=0}^n a_iX^i$ *et* $f_2 = \sum_{i=0}^n b_iX^i$.

Il découle des résultats précédents le fait que : déterminer un code auto dual $\Omega(P)$ revient à déterminer un polynôme P de $M_2(\mathbb{F}_q)[X]$, réversible, de degré $p = \frac{m}{2}$ et vérifiant $X^m+1 = P{}^tP^*$. En utilisant l'isomorphisme Ψ, on a donc

$$\Psi(X^m + 1) = X^m + 1 = (f_1 + f_2Y)(f_1^* + \overline{f_2^*}Y),$$

où f_1 et f_2 sont données par

$$\Psi(P) = f_1 + f_2 Y.$$

Il vient donc

$$\begin{aligned} f_1 f_1^* + f_2 f_2^* &= X^m + 1. \\ f_1 \overline{f_2^*} + f_2 \overline{f_1^*} &= 0. \end{aligned}$$

Comme P est réversible, on peut supposer sans perte de généralité que $deg(f_1) = p > deg(f_2) = q$. Ainsi, en posant

$$f_1 = \sum_{i=0}^{p} a_i X^i \quad f_2 = \sum_{i=0}^{q} b_i X^i,$$

on a le système polynomial suivant :

$$\begin{aligned} a_0 a_p + b_0 b_q &= 1. \\ \sum_{i=0}^{m} (a_i a_{p-k+i} + b_i b_{q-k+i}) &= 1. \\ \sum_{i=0}^{k} (a_i a_{p-k+i} + b_i b_{q-k+i}) &= 0 \quad 1 \leq k \leq m-1. \\ \sum_{i=0}^{k} (a_i b_{q-k+i}^2 + b_i a_{p-k+i}^2) &= 0 \quad 0 \leq k \leq p+q. \end{aligned}$$

Ainsi la détermination d'un code auto dual pour le produit scalaire euclidien revient à la résolution de ce système d'équations à un nombre d'inconnues égal à $p+q < m$ et à un nombre d'équations égal à $p+q+m+1 \leq 2m$, en utilisant les bases de Gröbner. Nous présentons dans la suite quelques exemples de codes auto duaux obtenus via la résolution du système précédent à l'aide du programme **5** de l'annexe sous Magma.

Exemple 8.7.5 *Pour $m = 4$ et 12, nous obtenons deux codes équivalents aux codes résidus quadratiques étendus de paramètres $[8, 4, 4]$ et $[24, 12, 8]$ comme donné dans [46, Ch. 16,§6].*

Exemple 8.7.6 *On sait déjà d'après [57, p.366,colonne de gauche] qu'il n'existe pas de codes cycliques auto-duaux pour $m = 6$, 8 et 10. Par contre, nous obtenons par équivalence un unique code 2-quasi-cyclique de longueur 12 et de distance minimale 4. Il existe au moins deux non équivalents codes $[16, 8, 4]$ ayant respectivement 12 et 28 mots de codes de poids 4.*

Exemple 8.7.7 *Pour $m = 14$, $p = 7$, $q = 5$, nous obtenons plusieurs codes cycliques auto-duaux binaires de longueur 28. Nous pouvons citer les deux codes de distance minimale 4 donnés dans [57] et quatre autres codes de distance minimale 2.*

Chapitre 9

Codes cycliques sur $M_2(\mathbb{F}_2)$

9.1 Introduction

En 1994, les auteurs résolvent dans [14] le problème de la dualité formelle des codes de Kerdock et Preparata en utilisant la dualité sur l'anneau $\mathbb{Z}_4$, un anneau de quatre éléments qui n'est pas un corps. Depuis, plusieurs travaux ont été publié sur les codes sur des anneaux. Cependant, peu d'entres eux traitaient le cas spécial des codes sur les anneaux non commutatifs. L'exemple le plus concret de ces anneaux est $A = M_2(\mathbb{F}_2)$. D'ailleurs, il fût le premier anneau non commutatif à avoir été utilisé comme alphabet pour des codes en blocs. La première motivation de l'utilisation de cet anneau est la construction des réseaux modulaires [4]. Récemment, une nouvelle application a vu le jour, à savoir la construction des codes espaces temps à partir de la concaténation des codes d'or [5].
L'atout de cet anneau est l'existence d'une application analogue à l'application de Gray, à savoir l'application de Bachoc [14] qui est une isométrie qui transforme l'anneau A muni d'une distance spéciale appelée distance de Bachoc en deux copies du corps de Galois $\mathbb{F}_4$ muni de la distance de Hamming [4, §6.2]. L'idée consiste à introduire deux matrices appelées ω et i et dont les polynômes caractéristiques respectifs sont $X^2 + X + 1$ et $X^2 + 1$ et satisfaisant à la relation $i\omega = w^2 i$. Ainsi l'anneau A peut être écrit comme $A = \mathbb{F}_4 + i\mathbb{F}_4$, en considérant $\mathbb{F}_4$ comme $\mathbb{F}_2[\omega]$. Ceci mène à attribuer à l'anneau A une structure d'anneau quotient sur l'anneau des polynômes tordus à coefficients dans le corps $\mathbb{F}_4$. D'un autre côté, notons que, pour une matrice non nulle M, le poids de Bachoc de M est 1 si M est inversible et 2 si elle est un diviseur de zéro. Ainsi, il parait logique d'étudier les codes cycliques sur cet exemple d'anneau fini non commutatif. Nous présentons donc une caractérisation des codes cycliques sur A via leurs générateurs. Nous montrons que leur duaux sont également cycliques et nous donnons une expression du générateur d'un dual d'un code C en fonction du générateur de C. Nous donnons une caractérisation arithmétique pour qu'un code cyclique sur A soit auto-dual pour le produit scalaire euclidien et nous montrons qu'il n'existe aucun code cyclique auto-dual de longueur impaire pour le produit scalaire hermitien. Nous montrons également que l'image de Bachoc d'un code auto- dual pour le produit scalaire euclidien est formellement auto-

dual. Notre nouvelle expression de l'image de Bachoc d'un code comme la somme de Plotkin des codes résidue et torsion nous permet de calculer les paramètres de plusieurs exemples de codes quaternaires formellement auto- duaux de longueur $n \leq 31$.

9.2 Notations et définitions

Notons $A = M_2(\mathbb{F}_2)$ l'anneau des matrices carrées d'ordre 2 à coefficients dans $\mathbb{F}_2$. Nous considérons dans A deux matrices notées ω et i de polynômes caractéristiques respectifs $X^2 + X + 1$ et $X^2 + 1$ et satisfaisant la relation $i\omega = \omega^2 i$.

Nous avons d'après [4]

$$A = \mathbb{F}_2[\omega] \oplus i\mathbb{F}_2[\omega].$$

Un choix possible des deux matrices i et ω est

$$i = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \omega = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Nous notons aussi u l'élément nilpotent défini par $u = 1 + i$. En identifiant $\mathbb{F}_2[\omega]$ avec $\mathbb{F}_4$, on peut écrire

$$A = \mathbb{F}_4 \oplus u\mathbb{F}_4.$$

Considérons μ la projection sur $\mathbb{F}_4$ parallèlement à $u\mathbb{F}_4$. Cette application est $\mathbb{F}_4-$ linéaire mais elle n'est pas un morphisme d'anneaux. Elle peut être prolongée d'une façon naturelle à une application définie de $A[X]$ dans $\mathbb{F}_4[X]$. Considérons maintenant un code linéaire C de longueur n, comme un A- sous module de A^n. Nous construisons les codes quaternaires R et T de même longueur n vérifiant $R = \mu(C)$ et T est le plus grand code D vérifiant $uD \subseteq C$. Les codes R et T sont appelés respectivement code résidu et code torsion. En utilisant uC, on peut montrer que $R \subseteq T$, avec égalité si et seulement si C est un $A-$module libre.

On note dans toute la suite $R_n = A[X]/(X^n - 1)$. Signalons que les idéaux bilatéraux de R_n représentent les codes cycliques de longueur n de A.

9.3 Construction d'un code cyclique

Considérons un entier n impair et l'écriture en facteurs irréductibles sur $\mathbb{F}_4$ du polynôme $X^n - 1 = \prod_{j=1}^{t} f_j$. Notons que si l'idéal (f_j) n'est pas bilatère, l'ensemble $A_j = A[X]/(f_j)$ n'est pas un anneau mais il est juste un A-module à droite. Nous commençons cette partie par énoncer les deux lemmes suivants. Pour cela nous allons utiliser un anologue du théorème des restes chinois dans le cas des modules.

Lemme 9.3.1 *En utilisant la notation des $A-$modules quotients $A_j = A[X]/(f_j)$ nous pouvons écrire R_n comme une somme directe de modules à droites A_j.*

$$R_n = \oplus_{j=1}^{t} A_j.$$

Preuve 9.3.2 *Ce résultat est une conséquence immédiate de [60, 9.12].*

Le résultat suivant est un analogue du Lemme 3 de [54] où $\mathbb{F}_4$ joue le rôle de $\mathbb{F}_2$ par rapport à A qui joue le rôle de $\mathbb{Z}_4$.

Lemme 9.3.3 *Soit f est un polynôme irréductible sur $\mathbb{F}_4$, les seuls $A-$modules à droites de $R(f) = A[X]/(f)$ sont (0), (u), (1). En particulier, l'anneau quotient est un anneau à chaine non commutatif.*

Preuve 9.3.4 *Soit $I \neq (0)$ un idéal de $R(f)$ et soit g un élément de $A[X]$ tel que $g+(f) \in I$, mais $g \notin (f)$. Comme f est irréductible le pgcd de $\mu(g)$ et f ne peut prendre que les valeurs 1 ou f. Dans le premier cas g est inversible mod f et $I = (1) = R(f)$. Dans le second cas, $I \subseteq u + (f)$. Considérons d'un autre côté $g = ur$ avec $ur + (f) \subseteq I$ et $ur + (f) \neq 0$. On peut supposer en se basant sur la dernière condition que $\mu r \notin (f)$. Comme f est irréductible, $pgcd(\mu r, f) = 1$. Il existe donc $a, b, c \in A[X]$ tels que*

$$ra + fb = 1 + uc,$$

et en multipliant par u des deux côtés, on a

$$ura = u + ufb.$$

La deuxième inclusion provient du fait que le membre à gauche de l'égalité appartient à I qui est un idéal bilatère.

Nous pouvons maintenant énoncer le théorème suivant

Théorème 9.3.1 *Soit une factorisation de $X^n - 1$ en trois facteurs deux à deux premiers entre eux : $X^n - 1 = fgh$ sur $\mathbb{F}_4[X]$. On peut construire un code cyclique*

$$C = (fh) + u(fg).$$

Les codes Résidu et Torsion de C sont deux codes cyclique quaternaires de longueur n et de polynômes générateurs respectifs fh et f, et de dimensions respectives $deg(g)$ et $deg(g) + deg(h)$. Inversement, tout $A-$code cyclique de longueur n peut être construit de cette façon.

Preuve 9.3.5 *Comme $(fg) \subseteq (f)$, il vient que le code ainsi construit est un idéal à droite de R_n. D'un autre côté, en écrivant $x^n - 1 = \prod_{j=1}^{t} f_j$, où les f_j sont des polynômes irréductibles su $\mathbb{F}_4[X]$, les lemmes 9.3.1 et 9.3.3 montrent que tout code cyclique est une somme d'idéaux de la forme A_j où les A_j sont soit égals à $(\widehat{f_j})$ ou $u(\widehat{f_j})$.*
Ainsi, tout code cyclique est obtenu par une construction multi niveau. Cette situation est différente du cas $\mathbb{Z}_4$ mais similaire au cas $\mathbb{F}_2 + u\mathbb{F}_2$, où comme pour A le corps résiduel est un sous anneau [6].

9.4 L'application de Bachoc

9.4.1 Propriétés métriques

L'application de Bachoc est définie ainsi :
pour tout élément de la forme $a + bi$ avec a, $b \in \mathbb{F}_4$, on a

$$\phi(a + ib) = (a, b).$$

En utilisant le fait que $u = 1 + i$ on voit que

$$\phi(a + ub) = (a + b, b).$$

Un prolongement de cette application à A^n révèle une relation avec la construction $(u, u+v)$ ou la somme de Plotkin de deux codes [55]. En effet, soient C_1 et C_2 deux codes quaternaires de longueur n, la somme de Plotkin de C_1 et C_2 est définie par

$$C_1 P C_2 = \{(u, u + v) |\, u \in C_1,\, v \in C_2\}.$$

Si C_1 et C_2 sont de dimension k_1, k_2 et de distance minimales d_1, d_2, alors les paramètres de la somme de Plotkin de C_1 et C_2 sont $[2n, k_1 + k_2, \min(2d_1, d_2)]$.

Proposition 9.4.1 *L'image d'un code cyclique par l'application de Bachoc est équivalente la somme de Plotkin de ses codes Torsion et Résidu.*

Preuve 9.4.2 *Le théorème 9.3.1 montre que tout code cyclique s'écrit sous la forme*

$$C = R + uT$$

où R et T sont les codes résidu et torsion. Le résultat est donc immédiat via la définition de l'application de Bachoc.

9.4.2 Propriétés de dualité

On définit comme dans [4, Prop. 2.1 (2)] une **conjugaison** sur A par

$$\overline{a + ib} = \overline{a} + ib,$$

pour a, $b \in \mathbb{F}_4$, et où $\overline{a} = a^2$. En prolongeant à A^n, on peut définir la forme hermitienne $\sum_j x_j \overline{y_j}$. Il est clair via [4, Lemma 6.4] que l'application de Bachoc est compatible avec le produit scalaire Hermitien. Nous avons le résultat suivant :

Proposition 9.4.3 *Soit C un code auto-orthogonal de A^n pour le produit scalaire e hermitien. On a, $\phi(C)$ est un code auto-orthogonal pour le produit scalaire hermitien classique de $\mathbb{F}_4^{2n}$.*

Preuve 9.4.4 *La preuve est immédiate via l'égalité*

$$(a+bi)\overline{(a'+b'i)} = a\overline{a'} + b\overline{b'} + (ba' + ab')i.$$

Considérons maintenant le produit scalaire euclidien sur A_n et définissons le polynôme énumérateur de Bachoc d'un code $C \subseteq A^n$ par

$$bwe_C(a,b,c) = \sum_{c \in C} a^{n_0(c)} b^{n_1(c)} c^{n_2(c)},$$

où $n_j(c)$ est le nombres de coordonnées de mots de codes c de poids de Bachoc j.
Rappelons que pour un code quartenaire de longueur N le polynôme énumérateur de Hamming est le polynôme défini par

$$W_Q(x,y) = \sum_{q \in Q} x^{N-|q|} y^{|q|}.$$

En particulier, un code quartenaire est formellement auto-dual pour le polynôme des poids de Hamming si et seulement si ce polynôme est un point fixe de la transformation de MacWilliams :

$$W_Q(x,y) = W_Q(\frac{x+3y}{2}, \frac{x-y}{2}).$$

Le résultat suivant est une conséquence de [14], qui est la clé de la dualité formelle des codes Kerdock et Preparata.

Proposition 9.4.5 *Soit C un code de A^n. Si C est auto-dual pour la forme euclidienne, alors $\phi(C)$ est formellement auto-dual pour le polynôme du poids.*

Preuve 9.4.6 *Le polynôme énumérateur de Hamming de $\phi(C)$ est obtenu à partir du polynôme énumérateur de Bachoc de C par*

$$W_{\phi(C)}(x,y) = bwe_C(x^2, xy, y^2).$$

*La relation de MacWilliams ([4, Th. 4.2 (2)]) sur A^n nous permet de donner l'expression du **bwe** de C en fonction du **bwe** de son dual par*

$$bwe_C(a,b,c) = \frac{1}{|C|} bwe_C(a+6b+9c, a+2b-3c, a-2b+c).$$

En éliminant les bwe's par $a = x^2$, $b = xy$, $c = y^2$ et en utilisant l'homogénéité et le fait que $|C| = 4^n = 2^{2n}$, on obtient l'identité

$$W_{\phi(C)}(x,y) = W_{\phi(C)}(\frac{x+3y}{2}, \frac{x-y}{2}),$$

ce qu'on veut démontrer.

9.4.3 Propriétés cycliques

La cyclicité de l'image de Bachoc d un code cyclique proviendra d'un théorème de structure sur les codes cycliques à racines multiples sur $\mathbb{F}_4$.

Lemme 9.4.7 *Soient C_1 et C_2 deux codes quaternaires cycliques de longueur impaire n et de polynômes générateurs g_1 et g_1g_2. Le code cyclique de longueur $2n$ et de générateur $g_1^2g_2$ est équivalent à la somme de Plotkin de C_1 et C_2.*

Preuve 9.4.8 *Ce résultat est une conséquence directe de la preuve du théorème 1 dans [59].*

Proposition 9.4.9 *L'image de Bachoc d'un code cyclique sur A de longueur impaire n est équivalente à un code cyclique de longueur $2n$ et de générateur $g_T^2g_R$ où g_R et g_T sont les générateurs respectifs des codes résidu et torsion.*

Preuve 9.4.10 *Ce résultat est une conséquence immédiate du lemme 9.4.7 et la proposition 9.4.1.*

9.5 Codes cycliques auto-duaux

Nous allons commencer cette partie par la caractérisation du générateur du dual euclidien d'un code cyclique. Notons f^* le polynôme unitaire associé au polynôme réciproque de f. Par exemple $(x+\omega)^* = x+\omega^2$.

Lemme 9.5.1 *Soit $C = (fh, ufg)$ un code cyclique de longueur impaire n tel que $X^n - 1 = fgh$. Le dual euclidien de C est $C^\perp = (g^*h^*, ug^*f^*)$.*

Preuve 9.5.2 *Une inclusion et l'égalité des dimensions permettent d'avoir le résultat.*

Théorème 9.5.1 *Soit $C = (fh, ufg)$ un code cyclique de longueur n tel que $X^n - 1 = fgh$. C est un code auto-dual euclidien si et seulement si $h = h^*$ et $g = f^*$. Il n'est jamais auto-dual hermitien.*

Preuve 9.5.3 *La condition suffisante est obtenue via le lemme précédent. Pour montrer la condition nécessaire, identifions les générateurs de la façon suivante*

$$fh = g^*h^*,$$

et

$$fg = g^*f^*.$$

En multipliant la première égalité par g et la deuxième par h nous obtenons

$$gh^* = hf^*.$$

Puisque h et g sont premiers entre eux et h et h^* ont le même degré, il vient que $h = \epsilon h^*$ pour $\epsilon = 1, \omega,$ ou ω^2. D'un autre côté, comme $x+1$ doit diviser h, on obtient que $\epsilon = 1$. En raisonnant de la même façon pour le cas hermitien, on obtient que $f = g$ ce qui est impossible pour n impair.

L'existence d'un triplet de polynômes satisfaisants le théorème 9.5.1 implique des conditions sur les classes cyclotomiques. Rappelons que l'ordre d'un entier a modulo b est le plus petit entier j tel que $a^j \equiv 1 \pmod{b}$.

Corollaire 9.5.4 *Il existe des codes cycliques auto-duaux euclidiens de longueur n si et seulement si il n'existe pas j tel que $4^j \equiv -1 \pmod{n}$, ce qui veut dire que l'ordre multiplicatif de 4 mod n est impair.*

Preuve 9.5.5 *Si -1 est une puissance de 4 modulo n, alors toutes les 4-classes cyclotomiques sont symétriques et il n'existe pas f et g non triviaux vérifiant les hypothèses du théorème 9.5.1. Réciproquement, supposons qu'il existe des 4-classes cyclotomiques non symétriques. Donnons la partition de $\mathbb{Z}_n$ en $T \bigcup -T \bigcup U$, avec $U = -U$ et T leur réunion. Soit f le polynôme dont les racines correspondent à T et h le polynôme dont les racines correspondent à U. Puisque f est non trivial, le code cyclique correspondant aux polynômes f, f^*, h est auto-dual et non trivial.*

Exemple 9.5.6 *Pour $n = 5$, on a $4 \equiv -1 \pmod{5}$ et il n'existe pas de codes cycliques non triviaux. En effet, la factorisation de X^n+1 est donnée en trois polynômes réciproques*

$$X^5 + 1 = (X + 1)(X^2 + \omega x + 1)(X^2 + \omega^2 X + 1).$$

L'annexe de [53] contient une étude détaillée sur les conditions que doit vérifier un entier n afin qu'il divise $2^k + 1$ pour un entier k. Posons $N(x)$ le nombre d'entiers premiers $p \leq x$ tel que l'ordre multiplicatif de 4 mod p est impair. On sait d'après [48, Th. 1] que pour x assez grand, on a

$$N(x) \sim \frac{7x}{12 \log x}.$$

En particulier, il existe d'une façon arbitraire des codes cycliques auto-duaux et non triviaux sur A.

9.6 Codes cycliques auto-duaux de longueur impaire $n \leq 31$.

Nous présentons dans la suite une classification des codes cycliques auto-duaux en prenant en compte la symétrie entre f et g dans le théorème 9.5.1. Puisque $g = f^*$, échanger f et g produit des codes équivalents a inversion de l ordre des coordonnées prés. Notons que h doit être divisible par le produit de tous les polynômes auto-réciproques qui divise $X^n + 1$. Ces résultats ont été trouvés via l'application des programmes **6** et **7** donnés dans l'annexe sous Magma.

$n = 3$: on trouve deux codes cycliques auto-duaux avec $h = X + 1$ et f, $g = X + \omega$.

$n = 5$: le corollaire 9.5.4 montre qu'il n'existe aucun code auto-dual non trivial.

$n = 7$: on trouve deux codes cycliques auto-duaux avec $h = X+1$ et f, $g = X^3+X+1$.

h	$X+1$
f	$X+w$
d_R	3
d_T	2
$\min(2d_T, d_R)$	3

TABLE 9.1 – Longueur 3

h	$X+1$
f	X^3+X+1
d_R	4
d_T	3
$\min(2d_T, d_R)$	4

TABLE 9.2 – Longueur 7

h	$X+1$
f	$X^5+w*X^4+X^3+X^2+w^2*X+1$
d_R	6
d_T	5
$\min(2d_T, d_R)$	6

TABLE 9.3 – Longueur 11

$n = 11$: on trouve deux codes cycliques auto-duaux avec $h = X + 1$ et f, $g = X^5 + X^5 + \omega X^4 + X^3 + X^2 + \omega^2 X + 1$.

$n = 13$: comme 13 divise $4^3 + 1$, le corollaire 9.5.4 montre qu'il n'existe aucun code auto-dual non trivial.

$n = 15$: la factorisation de $X^n + 1$ est de la forme $(X^5 + 1)f_1 f_1^* f_2 f_2^* f_3 f_3^*$, avec $f_1 = X^2 + X + \omega$, $f_2 = X^2 + X + \omega^2$, et $f_3 = X + \omega$. Nous discutons suivant le nombre de facteurs de h.

– En prenant $h = X^5 + 1$ nous obtenons quatre choix pour f et g : $f = f_1 f_2 f_3$, $f = f_1 f_2 f_3^*$, $f = f_1 f_2^* f_3$, $f = f_1 f_2^* f_3^*$.
– En prenant $h = (X^5 + 1)f_1 f_1^*$ nous obtenons deux choix pour f, g : $f = f_2 f_3$, $f = f_2 f_3^*$ et d'une façon similaire deux choix pour f, g dans le cas où $h = (X^5 + 1)f_2 f_2^*$, et deux choix de f, g dans le cas où $h = (X^5 + 1)f_3 f_3^*$.
– En prenant $h = (X^5 + 1)f_1 f_1^* f_2 f_2^*$ nous avons un seul choix de f à savoir $f = f_3$. Il en est de même pour $h = (X^5 + 1)f_1 f_1^* f_3 f_3^*$ et $h = (X^5 + 1)f_2 f_2^* f_3 f_3^*$

$n = 17$: 17 divides $4^2 + 1$ et par conséquent il n'existe aucun code auto-dual non trivial via le corollaire 9.5.4.

$n = 19$: il existe deux codes cycliques ato-duaux avec $h = X + 1$ et f, $g = X^9 + \omega X^8 + \omega X^6 + \omega X^5 + \omega^2 X^4 + \omega^2 X^3 + \omega^2 X + 1$.

$n = 21$: la factorisation de $X^n + 1$ est de la forme $(X^3 + 1)f_0 f_0^* f_1 f_1^* f_2 f_2^*$, avec $f_i = X^3 + \omega^i X + 1$. Si h est un multiple de $X^3 + 1$, la discussion est la même que pour la longueur 15. Si $h = X + 1$,on prend $f = (X + \omega)f_0^a f_1^b f_2^c$ avec a, b, c dans $\{1, *\}$.

$h/(X^5+1)$	1	1	1	1	$f_1f_1^*$	$f_1f_1^*$	$f_2f_2^*$	$f_2f_2^*$	$f_3f_3^*$	$f_3f_3^*$
f	$f_1f_2f_3$	$f_1f_2f_3^*$	$f_1f_2^*f_3$	$f_1f_2^*f_3^*$	f_2f_3	$f_2f_3^*$	f_1f_3	$f_1f_3^*$	f_1f_2	$f_1f_2^*$
d_R	8	8	6	3	9	11	11	9	8	6
d_T	3	3	3	2	2	3	3	2	3	2
$\min(2d_T, d_R)$	6	6	6	3	4	6	6	4	6	4

TABLE 9.4 – Longueur 15

$h/(X^5+1)$	$f_1f_1^*f_2f_2^*$	$f_1f_1^*f_3f_3^*$	$f_2f_2^*f_3f_3^*$
f	f_3	f_2	f_1
d_R	15	12	12
d_T	2	2	2
$\min(2d_T, d_R)$	4	4	4

TABLE 9.5 – Suite longueur 15

$h/(X^3+1)$	1	1	1	1	$f_0f_0^*$	$f_0f_0^*$	$f_1f_1^*$	$f_1f_1^*$	$f_2f_2^*$	$f_2f_2^*$
f	$f_0f_1f_2$	$f_0f_1f_2^*$	$f_0f_1^*f_2$	$f_0f_1^*f_2^*$	f_1f_2	$f_1f_2^*$	f_0f_2	$f_0f_2^*$	f_0f_1	$f_0f_1^*$
d_R	4	6	6	8	8	6	8	12	8	12
d_T	3	3	3	5	3	2	3	3	3	3
$\min(2d_T, d_R)$	4	6	6	8	6	4	6	6	6	6

TABLE 9.6 – Longueur 21

$h/(X^3+1)$	$f_0f_0^*f_1f_1^*$	$f_0f_0^*f_2f_2^*$	$f_1f_1^*f_2f_2^*$
f	f_2	f_1	f_0
d_R	12	12	12
d_T	2	2	2
$\min(2d_T, d_R)$	4	4	4

TABLE 9.7 – Suite longueur 21

abc	111	11$*$	$1*1$	$1**$	$*11$	$*1*$	$**1$	$***$
d_R	4	6	3	8	8	6	3	4
d_T	4	6	3	5	5	6	3	4
$\min(2d_T, d_R)$	4	6	3	8	8	6	3	4

TABLE 9.8 – Longueur 21 et $h = X + 1$

$n = 23$: on trouve deux codes cycliques ato-duaux avec $h = X + 1$ et f, $g = X^{11} + X^9 + X^7 + X^6 + X^5 + X + 1$.

$n = 25$: 25 divise $4^5 + 1$ et par conséquent il n'existe aucun code cyclique auto-dual non trivial via le corollaire 9.5.4.

$n = 27$: la factorisation de $X^n + 1$ est de la forme $(X + 1)f_1f_1^*f_2f_2^*f_3f_3^*$, avec $f_1 = X + \omega$, $f_2 = X^3 + \omega$, $f_3 = X^9 + \omega$. On peut refaire la même discussion que celle du cas $n = 15$.

$h/(X+1)$	1	1	1	1	$f_1f_1^*$	$f_1f_1^*$	$f_2f_2^*$	$f_2f_2^*$	$f_3f_3^*$	$f_3f_3^*$
f	$f_1f_2f_3$	$f_1f_2f_3^*$	$f_1f_2^*f_3$	$f_1f_2^*f_3^*$	f_2f_3	$f_2f_3^*$	f_1f_3	$f_1f_3^*$	f_1f_2	$f_1f_2^*$
d_R	3	3	3	3	3	3	3	3	9	9
d_T	3	3	3	3	3	3	3	3	2	2
$\min(2d_T, d_R)$	3	3	3	3	3	3	3	3	4	4

TABLE 9.9 – Longueur 27

$h/(X+1)$	$f_1f_1^*f_2f_2^*$	$f_1f_1^*f_3f_3^*$	$f_2f_2^*f_3f_3^*$
f	f_3	f_2	f_1
d_R	3	9	27
d_T	2	2	2
$\min(2d_T, d_R)$	3	4	4

TABLE 9.10 – Suite longueur 27

$n = 29$: Il existe deux codes cycliques auto-duaux avec $h = X + 1$ et

$$f,\ g = X^{14}+\omega X^{13}+\omega X^{11}+\omega^2 X^{10}+X^9+\omega^2 X^8+\omega X^7+\omega^2 X^6+X^5+\omega^2 X^4+\omega X^3+\omega X+1.$$

$n = 31$:

$$X^{31} + 1 = (X + 1)\prod_{j=1}^{3} f_j f_j^*,$$

avec

$$f_1 = X^5 + X^2 + 1,\ f_2 = X^5 + X^3 + X^2 + X + 1,\ f_3 = X^5 + X^2 + X + 1.$$

$h/(X+1)$	1	1	1	1	$f_1f_1^*$	$f_1f_1^*$	$f_2f_2^*$	$f_2f_2^*$	$f_3f_3^*$	$f_3f_3^*$
f	$f_1f_2f_3$	$f_1f_2f_3^*$	$f_1f_2^*f_3$	$f_1f_2^*f_3^*$	f_2f_3	$f_2f_3^*$	f_1f_3	$f_1f_3^*$	f_1f_2	$f_1f_2^*$
d_R	6	6	6	6	8	8	6	6	6	6
d_T	6	6	6	6	4	4	4	4	5	5
$\min(2d_T, d_R)$	6	6	6	6	8	8	6	6	6	6

TABLE 9.11 – Longueur 31

$h/(X+1)$	$f_1f_1^*f_2f_2^*$	$f_1f_1^*f_3f_3^*$	$f_2f_2^*f_3f_3^*$
f	f_3	f_2	f_1
d_R	10	8	6
d_T	2	3	3
$\min(2d_T, d_R)$	4	6	6

TABLE 9.12 – Suite longueur 31

9.7 Conclusion et problèmes ouverts

Dans ce chapitre, nous avons étudié la théorie des codes cycliques sur l'anneau non commutatif de matrices d'ordre 2 sur $\mathbb{F}_2$. En particulier, nous avons donné une caractérisation des codes cycliques et de leurs duaux en fonction de leurs deux générateurs. Nous avons prouvé l'existence d'une infinité de codes cycliques auto-duaux et non triviaux pour le produit scalaire e euclidien et pour une longueur impair. Leurs images par l'application de Bachoc sont des codes quaternaires formellement auto-duaux. Naturellement, on se pose la question à ce stade de la généralisation pour une longueur paire. Il parait que cette généralisation est difficile compte tenu des résultats dans [17] pour l'anneau $\mathbb{Z}_4$. Cependant, une motivation de cet effort est la construction des codes auto duaux hermitiens qui pourraient donner des réseaux par la construction de [4].

Annexe

Nous présentons dans la suite quelques programmes sous magma qui ont été utilisés tout au long de cette thèse et qui nous ont permis de trouver les valeurs numériques illustrant nos résultats.

1. Le programme **1** est le programme utilisé dans la section 4.5.1 du chapitre 4. On consière ici un exemple du programme pour $n = 10$ et pour le générateur $g = X^5 + X^4 + w * X^3 + w * X^2 + X + 1$ du code θ-cyclique auto-dual $[10, 5, 4]$ sur $\mathbb{F}_4$ donné dans [8]. Nous avons trouvé via ce programme un polynôme h de $\mathbb{F}_4[X; \theta]$ multiple de g et diviseur de $X^n - 1$ et donc un générateur d'un code θ-cyclique auto-orthogonal sur $\mathbb{F}_4$.

Programme 1

$k\langle w\rangle$:= GF(4) ;
$PG\langle x\rangle$:= PolynomialRing(k) ;
$p := x^{10} - 1$;
n :=function(f) ;
return Degree(f) ;
end function ;
a :=function(i) ;
return 2^i-1 ;
end function ;
b :=function(f,j) ;
if exists(t) i :i in [0..n(f)] | j eq a(i) then
return Coefficient(f,t) ;
else return 0 ;
end if ;
end function ;
Sp :=[b(p,j) : j in [0..2^n(p) -1]] ;
P :=Polynomial(Sp) ;
P ;
P1 :=RootsInSplittingField(P) ;
k1 :=SplittingField(P) ;
$PG1\langle x\rangle$:= PolynomialRing(k1) ;
n1 :=function(f) ;
return Degree(f) ;
end function ;

b1 :=function(f,j) ;
if exists(t) i : i in [0..n(f)] | j eq a(i) then
return Coefficient(f,t) ;
else return 0 ;
end if ;
end function ;
$g := x^5 + x^4 + w * x^3 + w * x^2 + x + 1$;
repeat
$i := Random(1, card(P1))$;
i ;
S1 :=$[(Coefficient(g, j))^2$:j in [0..n1(g)]] ;
g1 :=Polynomial(S1) ;
Sg :=[b1(g,j) : j in [0..$2^{n1(g)}$ -1]] ;
G :=Polynomial(Sg) ;
c :=function(f,j) ;
return Evaluate(f,P1[j][1])*P1[j][1] ;
end function ;
c(G,i) ;
if c(G,i) eq 0 then
g ;
else
g :=x*g1+ c(G,i)*g ;
g ;
end if ;
if exists(t) i : i in [0..n1(g)] | Coefficient(g,i) notin k then false ;
else PG !g ;
end if ;
until Degree(g) eq n(p) ;

2. le programme **2** est le programme utilisé dans la section 8.5 du chapitre 8. Il permet de construire un polynôme de $M_2(\mathbb{F}_2)[X]$ via deux polynômes quelconques de $\mathbb{F}_4[X]$ puis de déterminer sa période.

Programme 2

G<w> := FiniteField(4) ;
PG<x> := PolynomialRing(G) ;
K := FiniteField(2) ;
P<x> := PolynomialRing(K) ;
M2F2 :=MatrixRing(GF(2),2) ;
Pmat<X> :=PolynomialRing(M2F2) ;
PG5 := ! f in PG | Degree(f) lt 6 ! ; m :=map<G -> G | t :-> t^2> ;
m1 :=map<PG5 -> PG5 | f :-> PG ![m(Coefficient(f,0)),m(Coefficient(f,1)),m(Coefficient(f,2)),
m(Coefficient(f,3)),m(Coefficient(f,4)),m(Coefficient(f,5))]> ;
f1 := PG ![Random(G), Random(G), Random(G),Random(G), Random(G),Random(G)] ;

f1 ;
f2 := PG ![Random(G), Random(G), Random(G),Random(G), Random(G)] ;
if Degree(f2) lt Degree(f1) then f2 ;
if Coefficient(f1,0) eq 0 or Coefficient(f2,0)eq 0 then p := f1*m1(f1)+f2*m1(f2) ;
p ;
F :=Factorization(p,P) ;
F ;
f :=function(i) ;
return F[i,1] ;
end function ;
n :=function(i) ;
return Degree (f(i)) ;
end function ;
m :=function(i) ;
return F[i][2] ;
end function ;
k :=function(i) ;
return $2^n(i) - 1$;
end function ;
N :=function(i) ;
if k(i) ne 1 then return Factorization(k(i)) ;
else return 0 ;
end if ;
end function ;
r :=function(i) ;
if k(i) ne 1 then return card (N(i)) ;
else return 0 ;
end if ;
end function ;
s :=function(i,j) ;
if r(i) ne 0 then return N(i)[j,1] ;
else return 0 ;
end if ;
end function ;
e :=function(i,j) ;
if r(i) ne 0 then return N(i)[j][2] ;
else return 0 ;
end if ;
end function ;
d :=function(i,j) ;
if k(i) ne 1 then a_i :=ExactQuotient(k(i), $s(i,j)^{e(i,j)}$) ;
if x_i^amod(f(i)) ne 1 then l :=0 ;
repeat l :=l+1 ;

k_j :=ExactQuotient(k(i), $s(i,j)^l$) ;
until x_j^k mod(f(i)) ne 1 ;
l :=l-1 ;
else l :=e(i,j) ;
end if ;
return $s(i,j)^{(e(i,j)-l)}$;
else return 1 ;
end if ;
end function ;
pf :=function(i) ;
t :=0 ;
pf :=1 ;
if r(i) ne 0 then repeat t :=t+1 ;
pf :=pf*d(i,t) ;
until t eq r(i) ;
return pf ;
else return 1 ;
end if ;
end function ;
pf(1) ;
pf(2) ;
c :=function(i) ;
t :=-1 ;
repeat t :=t+1 ;
until 2^t gt m(i) ;
return 2^t ;
end function ;
s :=[pf(i) : i in [1..card F]] ;
s ;
b :=Lcm(s) ;
b ;
s1 :=[c(i) : i in [1..card F]] ;
s1 ;
b1 := Maximum(s1) ;
b1 ;
pf :=b*b1 ;
pf ;
ExactQuotient($x^{pf}+1$, p) ;
I2 :=Matrix(GF(2), 2, 2, [1,0,0,1]) ;
Y :=Matrix(GF(2), 2, 2, [0,1,1,0]) ;
W :=Matrix(GF(2), 2, 2, [0,1,1,1]) ;
Z :=Matrix(GF(2), 2, 2, [0,0,0,0]) ;
m3 :=map<G -> M2F2 | <0, Z>, <1, I2>, <w, W>, <w^2, W^2> > ;

```
m2 :=map<PG5 -> Pmat | f :-> Pmat ![m3(Coefficient(f,0)),m3(Coefficient(f,1)),m3(Coefficient(f,2)),
m3(Coefficient(f,3)),m3(Coefficient(f,4)),m3(Coefficient(f,5))]> ;
D :=m2(f1)+m2(f2)*Y ;
D ;
else false ;
end if ;
else false ;
end if ;
```

3. Le programme **3** est le programme utilisé dans la section 8.6 du chapitre 8. Il permet de déterminer un diviseur du polynôme $X^n - 1$ dans $M_2(\mathbb{F}_2)[X]$ pour $n = 6$ par exemple.

Programme 3

```
K := FiniteField(2) ;
P<x> := PolynomialRing(K) ;
M2F2 :=MatrixRing(GF(2),2) ;
Pmat<X> :=PolynomialRing(M2F2) ;
p :=x^6 + 1 ;
F :=Factorization(p) ;
C := CartesianProduct([ [0..f[2]] : f in F ]) ;
P :=[ produit *[F[i][1]^e[i] : i in [1..card F] ] : e in C] ;
for f in P do if IsEven(Degree(f)) then f ;
G<w> := FiniteField(4) ;
PG<x> := PolynomialRing(G) ;
PG5 :=! f in PG | Degree(f) lt 6 ! ;
m :=map<G -> G | t :-> t^2> ;
m1 :=map<PG5 -> PG5 | f :-> PG ![m(Coefficient(f,0)),m(Coefficient(f,1)),m(Coefficient(f,2)),
m(Coefficient(f,3)),m(Coefficient(f,4)),m(Coefficient(f,5))]> ;
f1 := PG ![ Random(G), Random(G),Random(G)] ;
f1 ;
h :=f+f1*m1(f1) ;
k :=Degree(h) ;
if IsEven(k) then F1 :=Factorization(h) ;
C1 := CartesianProduct([ [0..f[2]] : f in F1 ]) ;
P1 :=[ produit *[F1[i][1]^e[i] : i in [1..card F1] ] : e in C1] ;
if exists(l) f : f in P1 | Degree(f) eq k/2  then l ;
if l*m1(l)eq h then f2 :=l ;
if Degree(f2) lt Degree(f1) then
if Coefficient(f1,0) eq 0 or Coefficient(f2,0)eq 0 then f2 ;
g := ExactQuotient(p,f) ;
if g eq 1 then b :=1 ;
else F2 :=Factorization(g) ;
```

C2 := CartesianProduct([[0..f[2]] : f in F2]) ;
P2 :=[produit *[$F2[i][1]^{e[i]}$: i in [1..card F2]] : e in C2] ;
b :=Random(P2) ;
end if ;
g1 := m1(f1)* b ;
g2 :=f2*m1(b) ;
a := ExactQuotient(p,f*b) ;
p1 :=a*f1*g1+ g2*m1(a)*m1(f2) ;
p2 :=m1(a)*m1(f1)*g2+f2*a*g1 ;
I2 :=Matrix(GF(2), 2, 2, [1,0,0,1]) ;
Y :=Matrix(GF(2), 2, 2, [0,1,1,0]) ;
W :=Matrix(GF(2), 2, 2, [0,1,1,1]) ;
Z :=Matrix(GF(2), 2, 2, [0,0,0,0]) ;
m3 :=map<G -> M2F2 | <0, Z>, <1, I2>, <w, W>, $<w^2, W^2>$ > ;
m2 :=map<PG5 -> Pmat | f :-> Pmat ![m3(Coefficient(f,0)),m3(Coefficient(f,1)),
m3(Coefficient(f,2)),m3(Coefficient(f,3)),m3(Coefficient(f,4)),m3(Coefficient(f,5))]> ;
D :=m2(g1)+m2(g2)*Y ;
D ;
else "false" ;
end if ;
else "false" ;
end if ;
else " false" ;
end if ;
else " false" ;
end if ;
else " false" ;
end if ;
else " false" ;
end if ;
end for ;

4. Le programme **4** donné ci dessus permet de déterminer le code $\Omega(P)$ associé au diviseur D de $X^n - 1$ dans $M_2(\mathbb{F}_2)[X]$ trouvé via le programme 3.

Programme 4

G<w> := FiniteField(4) ;
Y<y> := PolynomialRing(G) ;
X<x> := PolynomialRing(Y) ;
M2F2 :=MatrixRing(GF(2),2) ;
I2 :=Matrix(GF(2), 2, 2, [1,0,0,1]) ;
Y :=Matrix(GF(2), 2, 2, [0,1,1,0]) ;
W :=Matrix(GF(2), 2, 2, [0,1,1,1]) ;
Z :=Matrix(GF(2), 2, 2, [0,0,0,0]) ;

m :=map<G -> M2F2 | <0, Z>, <1, I2>, <w, W>, $<w^2, W^2>$ > ;
M :=6 ;
f1 :=w ;
f2 := x^3 ;
h :=f1+f2*y ;
h ;
g1 :=w^2 ;
g2 :=x^3 ;
g :=g1+g2*y ;
g ;
L :=Transpose(m(Coefficient(Coefficient(h,0),0))+ m(Coefficient(Coefficient(h,0),1))*Y) ;
for i :=1 to Degree(h) do
L :=HorizontalJoin(L,Transpose(m(Coefficient(Coefficient(h,i),0))+
m(Coefficient(Coefficient(h,i),1))*Y)) ;
end for ;
R :=L ;
for i :=Degree(h)+2 to M do R :=HorizontalJoin(R,Z) ;
end for ;
H :=R ;
for j :=1 to Degree(g)-1 do LR :=ColumnSubmatrix(R,(2*M-2)) ;
R :=HorizontalJoin(Z,LR) ;
H :=VerticalJoin(H,R) ;
end for ;
C :=LinearCode(H) ;

5. Le programme **5** donné ci dessus est un programme permettant la résolution d'un système polynomial via l'utilisation des bases de Gröbner en vue d'obtenir un code auto-dual $\Omega(P)$.

Programme 5

p :=4 ;q :=4 ;
K<w> := FiniteField(4) ;
p1<a0,a1,a2,a3,a4,b0,b1,b2,b3,b4,X> := PolynomialRing(K,p+q+3) ;
a :=[a0,a1,a2,a3,a4] ;
b :=[b0,b1,b2,b3,b4] ;
f := +[a[i+1]*X^i : i in [0..p]] ;
f12 := +[a[p-i+1]*X^i : i in [0..p]] ;
f123 := +[$a[p-i+1]^2 * X^i$: i in [0..p]] ;
g1 :=+[b[i+1]*X^i : i in [0..q]] ;
g12 :=+[b[q-i+1]*X^i : i in [0..q]] ;
g123 :=+[$b[q-i+1]^2 * X^i$: i in [0..q]] ;
l :=f*f12+g1*g12 ;
g := f*g123 +g1*f123 ;
B :=[Coefficient(l,X,i) : i in [0..2*q]] ;

```
B1 :=[Coefficient(g,X,i) : i in [0..p+q]] ;
B[1] :=B[1]-1 ;
B[2*q+1] :=B[2*q+1]-1 ;
BB :=B cat B1 ;
I := lideal<p1| BB> ;
BB := GroebnerBasis(I) ;
I := rideal<p1| BB> ;
GroebnerBasis(I) ;
Variety(I) ;
card Variety(I) ;
G<X> :=PolynomialRing(K) ;
for j :=1 to card Variety(I) do
S :=[Variety(I)[j][i] : i in [1..p+1]] ;
f1 :=Polynomial(G,S) ;
S1 :=[Variety(I)[j][i] : i in [p+2..p+q+2]] ;
f2 :=Polynomial(G,S1) ;
Y<y> := PolynomialRing(K) ;
X<X> := PolynomialRing(Y) ;
M2F2 :=MatrixRing(GF(2),2) ;
I2 :=Matrix(GF(2), 2, 2, [1,0,0,1]) ;
Y :=Matrix(GF(2), 2, 2, [0,1,1,0]) ;
W :=Matrix(GF(2), 2, 2, [0,1,1,1]) ;
Z :=Matrix(GF(2), 2, 2, [0,0,0,0]) ;
m :=map<K -> M2F2 | <0, Z>, <1, I2>, <w, W>, <w^2, W^2> > ;
M :=2*p ;
h :=f1+f2*y ;
L :=Transpose(m(Coefficient(Coefficient(h,0),0))+ m(Coefficient(Coefficient(h,0),1))*Y) ;
for i :=1 to Degree(h) do L :=HorizontalJoin(L,Transpose(m(Coefficient(Coefficient(h,i),0))
+ m(Coefficient(Coefficient(h,i),1))*Y)) ;
end for ;
R :=L ;
for i :=Degree(h)+2 to M do R :=HorizontalJoin(R,Z) ;
end for ;
H :=R ;
for j :=1 to p-1 do LR :=ColumnSubmatrix(R,(2*M-2)) ;
R :=HorizontalJoin(Z,LR) ;
H :=VerticalJoin(H,R) ;
end for ;
C :=LinearCode(H) ;
end for ;
```

6. les programmes **6** et **7** sont les programmes qui ont été utilisés sous magma pour obtenir les valeurs numériques de la section 9.6 du chapitre 9. Le programme **6** permet

l'écriture de X^n+1 en trois facteurs f_1, f_2 et f_3. D'un autre côté, il résulte du programme **7** une classification des codes cycliques auto-duaux de longueur impaire $n \leq 31$.

Programme 6

```
F<w> :=GF(4) ;
p<x> :=PolynomialRing(F) ;
fp :=[Factorization(x^i + 1) : i in [7..31]] ;
x111 :=[] ;
for x1 in fp do x11 :=x1[i][1] : i in [1..card x1] ;
x111 :=x111 cat [x11] ;
end for ;
dif :=function(S1,S2) ;
return [x : x in S1| x in S2 eq false] ;
end function ;
mult :=function(S) ;
s :=[a : a in S] ;
ss :=s[1] ;
for i in [2..card s] do ss :=ss*s[i] ;
end for ;
return ss ;
end function ;
mult1 :=function(S) ;
s :=Subsets(S,card S-2) ;
s111 :=[] ;
for s1 in s do s11 :=[[mult(s1)] cat dif(S,s1)] ;
s111 :=s111 cat s11 ;
end for ;
return s111 ;
end function ;
```

Programme 7

```
G<w> := FiniteField(4) ;
P<x> := PolynomialRing(G) ;
h := x^5 + 1 ;
f1 := x^2 + x + w ;
f2 := x^2 + x + w^2 ;
f3 := x + w ;
a1 := LeadingCoefficient(ReciprocalPolynomial(f1)) ;
g1 :=a1^-1*ReciprocalPolynomial(f1) ;
a2 := LeadingCoefficient(ReciprocalPolynomial(f2)) ;
g2 :=a2^-1*ReciprocalPolynomial(f2) ;
a3 := LeadingCoefficient(ReciprocalPolynomial(f3)) ;
g3 :=a3^-1*ReciprocalPolynomial(f3) ;
f := f1*f2*f3 ; C1 := CyclicCode(15, f*h) ;
```

```
C2 := CyclicCode(15, f);
d1 := MinimumDistance(C1);
d2 := MinimumDistance(C2);
Minimum(2*d1, d2);
```

Conclusion

Les polynômes tordus forment une classe particulière et extrêmement intéressante des polynômes. En effet, la règle spécifique de multiplication dans l'ensemble de ces polynômes fait de lui un anneau non commutatif et attribue ainsi un atout majeur à cette classe de polynômes à savoir qu'ils n'admettent pas de factorisation unique. C'est en fait cette propriété qui a incité à leur utilisation dans le domaine de la théorie des codes correcteurs d'erreurs compte tenu du fait que la classe des codes tordus serait une classe riche en nombre et en bons paramètres. Cette richesse nous a en plus encouragé à l'utiliser dans le domaine de l'informatique quantique.

Dans la première partie de cette thèse, nous avons présenté une étude des polynômes tordus et élaboré des algorithmes de factorisation de ces polynômes. Ces résultats nous ont permis d'obtenir des nouvelles constructions des codes θ-cycliques auto-orthogonaux et par conséquent de retrouver des codes quantiques des meilleurs paramètres connus. D'un autre côté, nous avons présenté de nouvelles constructions à caractère combinatoire des codes θ-cycliques sur $\mathbb{F}_4$ et sur d'autres anneaux de cardinal 4. En effet, le problème de la factorisation des polynômes tordus a toujours été un handicap pour la construction des codes tordus surtout au niveau numérique. Pour contourner cette difficulté, nous avons introduit de diverses applications permettant le passage des codes θ-cycliques et θ-quasi-cycliques aux codes cycliques et quasi-cycliques. Ainsi, nous avons obtenu des nouvelles constructions combinatoires des codes cycliques tordus. Ces constructions ont été appliquées dans la construction des codes quantiques asymétriques, une nouvelle tendance qui a vu le jour récemment et nous ont permis d'obtenir des nouveaux codes quantiques asymétriques.

Dans la dernière partie de cette thèse nous avons utilisé les polynômes tordus comme un outil pour étudier les codes cycliques et quasi-cycliques sur des anneaux finis. En particulier, nous avons utilisé un isomorphisme entre $M_2(\mathbb{F}_q)$ et l'anneau quotient $\mathbb{F}_{q^2}[Y,\theta]_{/Y^2+1}$ pour présenter une étude des codes quasi-cycliques de $\mathbb{F}_2$.

En résumé, nous avons essayé dans cette thèse de présenter un tour d'horizon sur les polynômes tordus et leurs diverses utilisations dans le domaine de la théorie des codes correcteurs.

Bibliographie

[1] I. Siap, T. Abualrub, N. Aydin, et P. Seneviratne, "Skew quasi-cyclic codes of arbitrary length", International Journal of Information and Coding Theory, Vol 2, No 1, pp 10-20, 2011.

[2] T. Abualrub, A. Ghrayeb, N. Aydin , et I. Siap," On the construction of skew quasi-cyclic codes", IEEE Transactions on Information Theory, Vol. 56, No. 5, pp. 2081-2090, May 2010.

[3] Taher Abualrub, Nuh Aydin et Padmapani Seneviratne, "Skew Cyclic Codes Over $\mathbb{F}_2 + v\mathbb{F}_2$", Proceedings of ALCOMA 2010, Thurnau, Germany (2010).

[4] C. Bachoc, "Applications of coding theory to the construction of modular lattices", J. Comb. Th A 78-1 (1997) 92-119.

[5] J.C. Belfiore, F. Oggier, P. Solé, "Codes over Matrix Rings for Space-Time Coded Modulations", IEEE International Symposium on Information Theory vol. 58, no 2, February 2012.

[6] Bonnecaze, A.Udaya, P.GECT, "Cyclic Codes and Self-Dual Codes Over $F_2 + uF_2$", IEEE Transactions on Information Theory(1999) 1250-1255.

[7] W. Bosma, J. Cannon et C. Playoust, "The Magma algebra system. I. The user language", J. Symbolic Comput., **24** (1997), 235-265.

[8] D. Boucher, F. Ulmer, "Coding with skew polynomial rings", Journal of Symbolic Computation, 44, 1644-1656 (2009).

[9] D. Boucher, L. Chausade, P. Loidreau, F. Ulmer, "Skew codes of prescribed distance or rank", Designs, Codes and Cryptography, 50(3) (2009) 267-284.

[10] D. Boucher et F. Ulmer, "Codes as modules over skew polynomial rings", Proceedings of the 12th IMA Conference on Cryptography and Coding, Cirencester, Lecture Notes in Computer Science, **5921** (2009), 38-55.

[11] D. Boucher, P. Solé, F. Ulmer, Skew Constacyclic Codes over Galois Rings, Advances in Mathematics of Communications, 2, (2008) 273-292.

[12] D. Boucher, W. Geiselmann et F. Ulmer, "Skew-cyclic codes", Applied Algebra in Engineering, Communication and Computing, **18** (2007), 379-389.

[13] A. R. Calderbank, E. M. Rains, P. W. Shor, et N. J. A. Sloane, "Quantum Error Correction Via Codes Over GF(4)", IEEE Trans. Information Theory, 44 (1998), pp. 1369-1387.

[14] A. R. Calderbank, A. R. Hammons Jr, P. Vijay Kumar, N. J. A. Sloane et P. Sole, "The $\mathbb{Z}_4-$Linearity of Kerdock, Preparata, Goethals and Related Codes", IEEE Trans. Information Theory, 40 (1994), pp. 301-319.

[15] Pierre-lois Cayrel , Cristophe Chabot et Abdelkader Necer," quasi-cyclic codes over ring of matrices",Finite Fields and their Applications, number 16(2), p 100-115, 2010.

[16] J. Delenclos, A. Leroy," Non commutative symmetric functions and $W-$polynomials", Journal of Algebra and its Applications, 6 (5) (2007), 815-837.

[17] S. Dougherty, S. Ling, "Cyclic codes over $\mathbb{Z}_4$ of even length ", Designs, Codes and Cryptography,(2006), 127–153.

[18] Steven T. Dougherty , Philippe Gabority, Masaaki Harada, Akihiro Munemasa et Patrick Solé, "Type IV Self-Dual Codes over Rings", IEEE Trans. Inform. Theory 45 (1999), no. 7, 2345-2360.

[19] S. Dougherty, M. Harada, P. Solé, "Codes over Rings and the Chinese remainder theorem", Hokkaido J. of Math., 28 (1999) 253–283.

[20] Eric.Aleksandra Lj, "The resultant of non commutative polynomials", Mat. Vesnik 60 (2008), no. 1, 3–8.

[21] Martianus Frederic Ezerman , San Ling, Patrick Solé, Olfa Yemen, "From $\theta-$cyclic codes to asymmetric quantum codes", Advances in Mathematics of Communications, V 5, (2011) p 41-57.

[22] M. F. Ezerman, M. Grassl et P. Solé, "The weights in MDS codes", IEEE Transactions on Information Theory, vol. 57, no. 1, (2011), p. 392-396.

[23] M. F. Ezerman, S. Ling et P. Solé, "Additive asymmetric quantum codes", IEEE Transactions on Information Theory, vol. 57, no. 8, (2011), p. 5536-5550.

[24] K.Feng, S.Ling, C.Xing et L.Wang, "Asymmetric quantum codes : characterization and constructions", IEEE Trans. Inf. Theory, V 56 (2010) p 2938 - 2945.

[25] K. Feng, S. Ling et C. Xing, "Asymptotic bounds on quantum codes from algebraic geometry codes", IEEE Trans. Inf. Theory, **52** (2006), 986–991.

[26] E. M. Gabidulin, "Theory of codes with maximum rank distance", Probl. Peredach. Inform. (in Russian), **21** (1985), 3–16 ; pp. 1-12.

[27] D.Goss, *Basic structures of function field arithmetic*, Ergebnisse der Mathematik und ihrer Grenzgebiete, V35, Springer, (1996).

[28] M. Grassl, "Bounds on the minimum distance of linear codes and quantum codes", Online available at http ://www.codetables.de, accessed on April 3, 2010.

[29] P. Gaborit, A.M. Natividad, P. Solé, "Eisenstein lattices, Galois rings, and quaternary codes", Int. J. of Number Theory, Vol. 2, No 2 (2006)289–303.

[30] M. Greferath, S. E. Schmidt , "Linear Codes and Rings of Matrices", Proceedings of AAECC 13 Hawaii, Springer LNCS 1719 (1999) 160–169.

[31] W.Gary Huffman et Véra Pless, *Fundamentals of error-correcting codes*, Cambridge University Press, Cambridge, 2003.

[32] T.Hey, "Richard Feynman and computation",Contemporary Physics, (1999), volume 40, number 4, pages 257- 265

[33] L. Ioffe et M. M´ezard, "Asymmetric quantum error-correcting codes", Phys. Rev. Lett. A, 75 :032345, 2007.

[34] N.Jacobson, *Finite dimensional divisions algebra over fields*, Springer, (1996).

[35] Somphong Jitman et Ling San (April 6, 2010).

[36] A.Klappenecker, P.K.Sarvepalli, et M.Rötteler, "Asymmetric quantum codes : constructions, bounds and performance", Proc. of the Royal Soc. A, **465** (2009), 1645–1672.

[37] Andreas Klappencker, Pradeep Kiran Sarvepalli, Martin Rotteler, Asymmetric quantum LDPC codes, to appear (2008).

[38] J. L. Kim et V. Pless, "Designs in additive codes over $GF(4)$", Design, Codes and Cryptography, **30** (2003), 187–199.

[39] K.Lally. Quasicyclic codes of index l over F_q viewed as $\mathbb{F}_q[X]$ submodules of $\mathbb{F}_q^l[X]_{/X^m-1}$, AAECC-15, May (2003).

[40] T.Y.Lam et A. Leroy, "Algebraic conjugacy classes and skew polynomial rings", perspectives in ring theory, proceedings of the 1987 NATO workshop in Antwerp, Reidel, 1988.

[41] A. Leroy, "Pseudo linear transformation and evaluation in Ore extensions", Bull. Soc. Math. Belg. Vol. 2, n° 3, May 1995, 321-347.

[42] R.Lidl et H.Niederreiter, *Finite Fields* , Encyclopedia of Mathematics and its Applications Vol. 20, Amsterdam : Addison-Wesley. (1956).

[43] S. Ling et C. P. Xing, *Coding Theory. A First Course*, Cambridge University Press, Cambridge,(2004).

[44] S.Ling et Patrick Solé, On the algebraic structure of quasi-cyclic codes I : Finite fields.IEEE Trans.Inform .Theory,47 : (2001)2751-2760.

[45] B. R. Mcdonald, *Finite Rings with Identity*, Marcel Dekker Inc, (1974).

[46] (MR0465509) F. J. MacWilliams et N. J. A. Sloane, "The Theory of Error-Correcting Codes," North-Holland, Amsterdam, (1977).

[47] P. Moree, "On the divisors of $a^k + b^k$," Acta Arithmetica 80 (1997), 197–212.

[48] P. Moree, "Asymptotically exact heuristics for prime divisors of the sequence $\{a^k + b^k\}_{k=1}^{\infty}$", J. Integer Seq. 9 (2006), Article 06.2.8, pp. 15 (electronic).

[49] `http://magma.maths.usyd.edu.au/magma/`

[50] G. Nebe, E. M. Rains et N. J. A. Sloane, "Self-Dual Codes and Invariant Theory", Algorithms and Computation in Mathematics vol. 17, Springer-Verlag, Berlin Heidelberg, 2006.

[51] O.Ore, "Theory of non-commutative polynomials", Ann. of Math., 34 :480-508, 1933.

[52] O.Ore, " On a special class of polynomials," Transaction of the American Mathematical Society **Vol 35** (1933) 559-584.

[53] V. Pless, P. Solé, Z. Qian, "Cyclic Self-Dual $\mathbb{Z}_4$-Codes", Finite Fields and Their Applications 3, 48-69 (1997)

[54] V. Pless, Z. Qian, "Cyclic Codes and Quadratic Residue Codes over $\mathbb{Z}_4$", IEEE Trans. Inform. Theory vol. 42, no. 5 (1996) pp 1594-1600.

[55] M. Plotkin, "Binary codes with specified minimum distance", IRE Trans, vol. IT-6, pp. 445–450, 1960.

[56] E. M. Rains et N. J. A. Sloane, "Self-dual codes", Handbook of Coding Theory I, Elsevier, (1998), 177–294.

[57] N. J. A. Sloane et J. G. Thompson, "Cyclic Self-Dual Codes", IEEE Trans. Information Theory, IT-29 (1983), pp. 364-366.

[58] P.Solé et O.Yemen, "Binary quasi-cyclic codes of index 2 and skew polynomial rings", Finite Fields and Their Applications,(2012),http ://www.sciencedirect.com/science/article/pii/S1071579712000226.

[59] J.H. van Lint, "Repeated root cyclic codes", IEEE Trans. on Information Theory, IT-37 (1991) 343-345.

[60] R. Wisbauer, *Foundations of module and ring theory*, Gordon and Breach (1991).

[61] www.codestables.de/

[62] Zhe-Xian Wan, "Lectures on finite fields and Galois rings", World Scientific, (2003) 342 pages.