REPUBLIQUE DU CAMEROUN
****
Paix-Travail-Patrie
****
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
****
UNIVERSITE DE YAOUNDE I
****
FACULTE DES SCIENCES
****
DEPARTEMENT DE MATHEMATIQUES
****

REPUBLIC OF CAMEROON
****
Peace-Work-Fatherland
****
MINISTRY OF HIGHER EDUCATION
****
UNIVERSITY OF YAOUNDE I
****
FACULTY OF SCIENCE
****
DEPARTMENT OF MATHEMATICS
****

CODES CYCLIQUES DIVISIBLES SUR UN CORPS DE GALOIS PREMIER

Mémoire de Master de mathématiques

De

FOUOTSA TAKO Boris

Matricule : **11Y623**

Licencié en Mathématiques

Sous la direction de :

Pr MOUAHA Christophe

Maître de Conférences

Ecole Normale Supérieure de Yaoundé

Année académique : 2015-2016

CODES CYCLIQUES DIVISIBLES SUR UN CORPS DE GALOIS PREMIER

Mémoire de Master de mathématiques

De

FOUOTSA TAKO Boris

Matricule: **11Y623**

Licencié en Mathématiques

Sous la direction de :

Pr MOUAHA Christophe

Maître de Conférences

Ecole Normale Supérieure de Yaoundé

Année Académique 2015-2016

♠ Dédicace ♠

Je dédie ce mémoire à

la famille TAKO

♠ Remerciements ♠

Je remercie le Dieu tout puissant pour sa grâce, sa protection et sa miséricorde.

J'adresse mes vifs remerciements au Pr MOUAHA Christophe, qui au delà de ses multiples occupations m'a attribué un sujet et a guidé mes premiers pas dans la recherche.

Je tiens également à remercier le Docteur NDJEYA en particulier et les enseignants de la Faculté des Sciences en général, ainsi que ceux de l'École Normale Supérieure de Yaoundé qui m'ont suivi tout au long de mon cursus universitaire et professionnel.

Mes remerciements vont également à :

☞ Mes parents M. NAMEKONG Jean et Mme NAMEKONG Béatrice pour le soutien moral et financier sans faille qu'ils m'ont apporté toutes ces dernières années.

☞ A mes camarades de Master II pour leur soutien moral durant l'année académique.

☞ A ma famille toute entière pour son affection et son soutien.

♠ Résumé ♠

Ce travail est consacré à l'étude des codes cycliques sur un corps de Galois premier et à la caractérisation de ceux dont les poids des mots ont un diviseur commun distinct de 1. Nous utilisons le théorème de McEliece pour montrer qu'un code cyclique binaire non e-dégénéré C est divisible si et seulement si 1 est un non-zéro de C. Nous montrons que ce résultat n'est vrai qu'en caractéristique 2 et nous donnons une caractérisation générale des codes cycliques divisibles sur un corps de Galois premier. Nous montrons aussi que si un code cyclique binaire non e-dégénéré est divisible, alors son orthogonal est aussi divisible si et seulement s'il est e-dégénéré. A la fin de ce travail, nous proposons un programme écrit en matlab qui permet de calculer la distribution de poids d'un code cyclique et de dire si ce code est e-dégénéré ou pas, divisible ou pas

Mots clés : corps de Galois premier, poids de Hamming, code de Griesmer, code cyclique, code divisible, code e-dégénéré.

♠ Abstract ♠

This work is devoted to the study of cyclic codes on a prime Galois field and the characterisation of the class of those having code words with a common divisor larger than one. We use the McEliece theorem to prove that a non e-degenerated binary code C is a divisible code if and only if 1 is a non zero of C. We prove that it's true only for binary codes and we give a general charaterisation of cyclic divisible codes on a prime Galois field. We also prove that if a non e-degenerated cyclic code is divisible on a prime Galois field, then its orthogonal is also divisible if and only if it is e-degenerated. At the end of this work, we give a matlab program that calculates the weight distribution of a binary cyclic code and says if the code is e-degenerated or not, if it's divisible or not.

Key words : prime Galois field, Hamming weight, Griesmer code, cyclic code, divisible code, e-degenerated code.

♠ Table des matières ♠

♠ Liste des tableaux ♠

♠ Introduction ♠

La communication est au cœur des interactions humaines. Elle consiste en l'envoi d'un message par un émetteur à travers un canal de transmission, et à la réception du message envoyé par un récepteur. Le canal n'étant pas fiable à 100%, il peut y avoir des erreurs lors de la transmission. Le codage permet de détecter et de corriger ces erreurs. C'est ainsi que nous parvenons à photographier les planètes lointaines, à communiquer quelque soit la distance.

En théorie algébrique du codage, l'un des problèmes majeurs est d'améliorer à la fois le nombre d'erreurs corrigées par un code et la vitesse de transmission des messages. Mais des inégalités mathématiques (borne de singleton par exemple) montrent qu'en améliorant l'un de ces paramètres, on diminue l'autre. On a donc des codes dits optimaux pour lesquels ces inégalités deviennent des égalités. Des chercheurs tels que McEliece [1] dans [3], Harold Ward [2] dans [5] et tant d'autres, se sont intéressés aux poids des mots des codes et se sont rendus compte que certains codes avaient des distributions de poids particulières : les poids de tous les mots avaient un diviseur commun (distinct de 1). En 1981, Harold N. Ward les a appelés codes divisibles. En 1998, il a montré dans [6] que les codes de Griesmer (qui sont des codes optimaux) ont des propriétés de divisibilité très particulières.

Dans ce travail, il est question pour nous d'élucider les notions de code cyclique sur un corps de Galois et de divisibilité d'un code linéaire, et de caractériser, parmi les codes cycliques sur un corps de Galois premier, ceux qui sont divisibles. Pour y parvenir, nous organisons notre travail en trois chapitres.

1. mathématicien américain, né en 1942, gagnant de la médaille Alexander en 2009 ;
2. mathématicien américain, né en 1936, il a défini la notion de code linéaire divisible en 1981.

Le premier chapitre porte sur les corps finis et leur construction. Le deuxième chapitre porte sur la notion de code sur un corps de Galois. Nous y définissons la notion de code linéaire, de code cyclique. Nous y construisons trois codes cycliques binaires de longueur 15. Le troisième chapitre porte sur la divisibilité des codes cycliques sur un corps de Galois premier. Nous y définissons la notion de code linéaire divisible et donnons quelques résultats généraux sur la divisibilité des codes linéaires. Nous utilisons le théorème de McEliece sur la divisibilité des codes cycliques sur un corps de Galois premier pour établir les résultats cités dans le résumé, et étudions la divisibilité de trois codes cycliques binaires que nous construisons au chapitre 2. Dans l'annexe, nous présentons le code source de notre programme matlab qui calcule la distance minimale, la capacité correctrice, la distribution de poids, le degré de divisibilité d'un code cyclique binaire, et dit si ce code est divisible ou pas, s'il est e-dégénéré ou pas.

Chapitre Un

PRÉLIMINAIRES

Nous ne pouvons étudier une langue sans étudier son alphabet. Les corps de Galois sont les alphabets des codes linéaires que nous allons étudier. L'un des objectifs de ce chapitre est de décrire les corps finis, de donner leurs propriétés et de dire comment ils peuvent être construits.

1.1 Rappels sur les corps finis

Définition 1.1.1 : *Un corps est un anneau unitaire tel que tout élément non nul est inversible.*

Définition 1.1.2 : *Un corps fini est un corps qui a un nombre fini d'éléments.*
On l'appelle aussi corps de Galois[1].

Proposition 1.1.1 : *Tout corps est un anneau intègre.*

Preuve : Soit $\mathbb{K}$ un corps, soient $a, b \in \mathbb{K}$ tels que $a \neq 0$ et $ab = 0$.
Alors $b = 1b = (a^{-1}a)b = a^{-1}(ab) = a^{-1}0 = 0$. D'où $\mathbb{K}$ est intègre. ■

Théorème 1.1.1 : *(théorème de Wedderburn*[2]*, [4])*
Tout corps fini est commutatif.

Proposition 1.1.2 : *Soit A un anneau unitaire d'élément unité 1_A.*
L'application

$$\begin{array}{rccc} \psi : & \mathbb{Z} & \to & A \\ & n & \longmapsto & n.1_A \end{array}$$

1. Evariste Galois, mathématicien français (1811-1832), a posé les prémisses de la théorie de Galois.
2. Joseph Henry Maclagan Wedderburn, mathématicien écossais (1882-1948).

est un morphisme d'anneaux.

Preuve : Immédiate ■

Remarque 1.1.1 : Le noyau de cette application est donc un idéal de $\mathbb{Z}$. $\mathbb{Z}$ étant principal, cet idéal est de la forme $m\mathbb{Z}$ où m est un entier naturel. Donc $ker(\psi) = m\mathbb{Z}$.

Définition 1.1.3 : *L'entier naturel m de la remarque précédente est appelé la caractéristique de l'anneau A.*

Proposition 1.1.3 : *La caractéristique d'un corps fini est un nombre premier.*

Preuve : Soit $\mathbb{K}$ un corps fini de caractéristique m.
Comme le corps $\mathbb{K}$ a un nombre fini d'éléments, le morphisme ψ ne peut être injectif, donc son noyau est distinct de l'idéal nul. Ainsi, m est un entier naturel non nul et distinct de 1 (car si $m = 1$, alors $1_{\mathbb{K}} = 0_{\mathbb{K}}$, absurde).
Supposons que m ne soit pas premier, alors il existe deux entiers naturels a et b, non nuls et tous distincts de 1, tels que $m = ab$.
Puisque a et b sont non nuls et tous distincts de 1, alors $a.1_{\mathbb{K}}$ et $b.1_{\mathbb{K}}$ sont deux éléments non nuls de $\mathbb{K}$. On a :

$$(a.1_{\mathbb{K}}) \times (b.1_{\mathbb{K}}) = (ab).1_{\mathbb{K}} = m.1_{\mathbb{K}} = 0_{\mathbb{K}}$$

Ce qui contredit le fait que le corps $\mathbb{K}$ soit intègre.
Donc m est premier. ■

Proposition 1.1.4 : *Si $\mathbb{K}$ est un corps fini de caractéristique un nombre premier p, alors $\mathbb{K}$ contient un sous-corps $\mathbb{F}_p = Im(\psi)$ isomorphe au corps $\mathbb{Z}/p\mathbb{Z}$.*

Preuve : Soit $\mathbb{K}$ un corps fini de caractéristique un nombre premier p, alors $ker(\psi) = p\mathbb{Z}$.
D'après le premier théorème d'isomorphisme, $Im(\psi) \cong \mathbb{Z}/ker(\psi) = \mathbb{Z}/p\mathbb{Z}$.
D'où le résultat. ■

Définition 1.1.4 : *Le sous-corps $\mathbb{F}_p$ du corps fini $\mathbb{K}$ de la proposition précédente est appelé sous-corps premier de $\mathbb{K}$.*

Proposition 1.1.5 : *Le cardinal d'un corps fini est une puissance de sa caractéristique.*

Preuve : Soit $\mathbb{K}$ un corps fini de caractéristique p et $\mathbb{F}_p$ son sous-corps premier.

Le corps $\mathbb{K}$ peut être vu comme un $\mathbb{F}_p$-espace vectoriel. Puisque $\mathbb{K}$ est de cardinal fini, alors $\mathbb{K}$ peut être vu comme un $\mathbb{F}_p$-espace vectoriel de dimension fini l. Dans ce cas, $\mathbb{K} \cong (\mathbb{F}_p)^l$.

Comme $\mathbb{F}_p \cong \mathbb{Z}/p\mathbb{Z}$, alors $\mathbb{K} \cong (\mathbb{Z}/p\mathbb{Z})^l$.

Donc

$$card(\mathbb{K}) = card((\mathbb{Z}/p\mathbb{Z})^l) = (card(\mathbb{Z}/p\mathbb{Z}))^l = p^l$$

■

Définition 1.1.5 : *On appelle corps de Galois premier tout corps de Galois dont le cardinal est un nombre premier.*

Proposition 1.1.6 : *Soit $\mathbb{K}$ un corps fini de caractéristique p. Alors l'application*

$$\begin{array}{rccc} Fr & \mathbb{K} & \to & \mathbb{K} \\ & x & \longmapsto & x^p \end{array}$$

est un automorphisme de corps.

Preuve : Soit $\mathbb{K}$ un corps fini de caractéristique p.

Soient $a, b \in \mathbb{K}$.

Puisque le corps $\mathbb{K}$ est commutatif (car c'est un corps fini), on a :

$$Fr(ab) = (ab)^p = a^p b^p = Fr(a)Fr(b)$$

Soit i un entier tel que $1 \leq i \leq p-1$, alors $iC_p^i = pC_{p-1}^{i-1}$. Puisque p est premier et est distinct de i, alors p et i sont premiers entre eux. Donc p divise C_p^i. On peut donc écrire

$$\begin{aligned} Fr(a+b) &= (a+b)^p \\ &= \sum_{i=0}^{p} C_p^i a^i b^{p-i} \\ &= a^p + \sum_{i=1}^{p-1} C_p^i a^i b^{p-i} + b^p \\ &= a^p + b^p \quad \text{car } p\text{divise } C_p^i \forall i \in \{1; \cdots; p-1\} \\ &= Fr(a) + Fr(b) \end{aligned}$$

Donc Fr est un endomorphisme.

Soit $a \in \mathbb{K}$. Si $Fr(a) = 0$, alors $a^p = 0$, et comme le corps $\mathbb{K}$ est intègre, on a $a = 0$.

Ainsi, Fr est un endomorphisme injectif ; d'où Fr est un automorphisme. ■

Définition 1.1.6 : *L'automorphisme Fr de la proposition 1.1.6 est appelé automorphisme de Frobenius.*

Proposition 1.1.7 : *Si $\mathbb{K}$ est un corps fini de cardinal q, alors $\mathbb{K}^*$ est un groupe cyclique d'ordre $q-1$.*

Preuve : Soit $\mathbb{K}$ un corps fini de cardinal q, alors $\mathbb{K}^*$ est un groupe abélien multiplicatif d'ordre $q-1$.
Supposons que $\mathbb{K}^*$ ne soit pas cyclique.
Soit e son exposant et soit α un élément d'ordre e.
Puisque l'ordre du groupe est $q-1$, alors e divise $q-1$.
Comme e est l'exposant du groupe $\mathbb{K}^*$, alors l'ordre de chaque élément de $\mathbb{K}^*$ est un diviseur de e. Donc pour tout $x \in \mathbb{K}^*$, $x^e = 1$; c'est-à-dire que tous les éléments de $\mathbb{K}^*$ sont des racines du polynôme $x^e - 1$. Puisque $card(\mathbb{K}^*) = q-1$, alors $q-1 \leq e$.
Ainsi, $q-1 = e$. D'où α est un générateur de $\mathbb{K}^*$. ■

Définition 1.1.7 : *Soit $\mathbb{K}$ un corps fini. On appelle élément primitif de $\mathbb{K}$ ou racine primitive de $\mathbb{K}$ tout générateur du groupe multiplicatif $\mathbb{K}^*$ de $\mathbb{K}$.*

Notation 1.1.1 : Si $\mathbb{K}$ est un corps, $\mathbb{K}[X]$ désigne l'anneau des polynômes à coefficients dans $\mathbb{K}$.

Remarque 1.1.2 : On a :

$$\forall x \in \mathbb{K}^*, x^{q-1} - 1 = 0$$

Donc les éléments de $\mathbb{K}$ sont des racines du polynôme $X^q - X \in \mathbb{F}_p[X]$. Puisque le polynôme $X^q - X$ est de degré q, alors $\mathbb{K}$ est l'ensemble des racines du polynôme $X^q - X \in \mathbb{F}_p[X]$.

Dans la suite, $\mathbb{F}_q$ désigne un corps fini de cardinal q et de caractéristique p $(q = p^n)$.

Proposition 1.1.8 : *L'anneau $\mathbb{F}_q[X]$ est principal.*

Preuve : $\mathbb{F}_q$ est un corps, donc un anneau intègre. Ainsi, le produit PQ de deux polynômes non nuls P et Q (c'est-à-dire de degrés supérieurs ou égaux à 0) est un polynôme de degré $deg(PQ) = deg(P) + deg(Q) \geq 0$, c'est-à-dire $PQ \neq 0$. D'où $\mathbb{F}_q[X]$ est un anneau intègre.
Soit I est un idéal non nul de $\mathbb{F}_q[X]$. Soit P_0 un polynôme unitaire et non nul de I, de degré minimal parmi les polynômes non nuls de I. Alors $P_0\mathbb{F}_p[X] \subset I$ car I est un idéal de $\mathbb{F}_q[X]$. Soit $P \in I$, par division euclidienne, on a $P = QP_0 + R$ avec $R = 0$ ou $R \neq 0$ et $deg(R) < deg(P_0)$. On a $R = P - QP_0 \in I$ car $P, P_0 \in I$. Si $R \neq 0$, alors R serait un polynôme non nul de I de degré strictement inférieur à celui de P_0, ce qui contredit le fait

que P_0 soit de degré minimal parmi les polynômes non nuls de I. Donc $R = 0$, c'est-à-dire $P = QP_0 \in P_0\mathbb{F}_q[X]$ et par suite $I = P_0\mathbb{F}_q[X]$. Ainsi, $\mathbb{F}_q[X]$ est principal. ■

Proposition 1.1.9 : *Soit $\beta \in \mathbb{F}_q$. L'ensemble I_β des polynômes à coefficients dans $\mathbb{F}_q$ dont β est une racine est un idéal principal de $\mathbb{F}_q[X]$.*

Preuve : Immédiate ■

Définition 1.1.8 : *Soit $\beta \in \mathbb{F}_q$.*
On appelle polynôme minimal de β le polynôme unitaire générateur de I_β.

Notation 1.1.2 : Le polynôme minimal de β est noté $m_\beta(X)$.

Proposition 1.1.10 : *Le polynôme minimal d'un élément $\beta \in \mathbb{F}_q$ est irréductible.*

Preuve : Soit $\beta \in \mathbb{F}_q$ et $m_\beta(X)$ son polynôme minimal.
Soient $m_1(X), m_2(X) \in \mathbb{F}_q[X]$ tels que $m_\beta(X) = m_1(X)m_2(X)$.
Comme $m_\beta(X)$ est le polynôme minimal de β, alors $m_\beta(\beta) = 0$. On a :

$$\begin{aligned}
m_\beta(\beta) = 0 &\Rightarrow m_1(\beta)m_2(\beta) = 0 \\
&\Rightarrow m_1(\beta) = 0 \text{ ou } m_2(\beta) = 0 \qquad \text{car } \mathbb{F}_q \text{ est intègre} \\
&\Rightarrow m_1(X) \in I_\beta \text{ ou } m_2(X) \in I_\beta \\
&\Rightarrow m_\beta(X) | m_1(X) \text{ ou } m_\beta(X) | m_2(X) \\
&\Rightarrow m_\beta(X) \text{ et } m_1(X) \text{ sont associés, ou } m_\beta(X) \text{ et } m_2(X) \text{ sont associés, car} \\
&\quad\ m_1(X) \text{ et } m_2(X) \text{ divisent } m_\beta(X)
\end{aligned}$$

D'où $m_\beta(X)$ est irréductible. ■

Définition 1.1.9 : *Soit $P(X) \in \mathbb{F}_q[X]$. On appelle corps de décomposition de $P(X)$ tout sur-corps de $\mathbb{F}_q$ dans lequel $P(X)$ se décompose en facteurs linéaires et qui soit minimal pour cette propriété.*

Proposition 1.1.11 : *([2]) Soit $P(X) \in \mathbb{F}_q[X]$. Le corps de décomposition de $P(X)$ existe et est unique (à isomorphisme près).*

Remarque 1.1.3 : Nous avons vu que $\mathbb{F}_q$ est l'ensemble des racines du polynôme $X^q - X \in \mathbb{F}_p[X]$, donc $\mathbb{F}_q$ est un corps de décomposition de $X^q - X$. Ainsi, tout corps fini de cardinal q est un corps de décomposition du polynôme $X^q - X$ sur un corps premier. Nous déduisons donc la proposition suivante.

Proposition 1.1.12 : *Pour tout nombre premier p et pour tout entier naturel n, il existe un unique corps fini de cardinal p^n (à isomorphisme près).*

Preuve : Soit p un nombre premier et n un entier naturel. Pour l'existence, il suffit de prendre un corps de décomposition du polynôme $X^{p^n} - X \in \mathbb{F}_p[X]$; c'est un corps fini de cardinal p^n.
Nous avons vu dans la remarque précédente que tout corps fini de cardinal p^n est un corps de décomposition du polynôme $X^q - X$ sur un corps premier. D'après la proposition précédente, nous déduisons l'unicité à isomorphisme près du corps fini de cardinal p^n. ■

Remarque 1.1.4 : La construction des corps se fait de plusieurs façons différentes : en considérant le corps de fractions d'un anneau intègre, en considérant le quotient d'un anneau par un de ses idéaux maximaux ou en étendant des corps pour avoir d'autres plus grands. Dans le cas des codes linéaires, les extensions sont les méthodes utilisées dans la construction des corps finis.

Proposition 1.1.13 : *Soit $\beta \in \mathbb{F}_{p^n}$ et $m_\beta(X) \in \mathbb{F}_p[X]$ son polynôme minimal. L'ensemble*

$$\mathbb{F}_p(\beta) = \{f(\beta), f(X) \in \mathbb{F}_p[X]\}$$

est un sur-corps de $\mathbb{F}_p$ contenant β et isomorphe à $\mathbb{F}_p[X]/(m_\beta(X))$.

Preuve : Soit $\beta \in \mathbb{F}_{p^n}$ et $m_\beta(X) \in \mathbb{F}_p[X]$ son polynôme minimal.
Le morphisme d'anneaux

$$\begin{aligned} \Phi : \ \mathbb{F}_p[X] &\to \mathbb{F}_{p^n} \\ f &\mapsto f(\beta) \end{aligned}$$

a pour noyau $(m_\beta(X))$ et pour image $\mathbb{F}_p(\beta)$. D'après le premier théorème d'isomorphisme, $\mathbb{F}_p[X]/ker(\Phi) = \mathbb{F}_p[X]/(m_\beta(X))$ est isomorphe à $Im(\Phi) = \mathbb{F}_p(\beta)$. Puisque $m_\beta(X)$ est un polynôme irréductible, $(m_\beta(X))$ est un idéal maximal de $\mathbb{F}_p[X]$, donc $\mathbb{F}_p[X]/(m_\beta(X))$ est un corps.
Ainsi, $\mathbb{F}_p(\beta)$ est un corps isomorphe à $\mathbb{F}_p[X]/(m_\beta(X))$.
$\mathbb{F}_p(\beta)$ contient $\mathbb{F}_p$ car les éléments de $\mathbb{F}_p$ peuvent être vus comme des polynômes constants.
$\beta \in \mathbb{F}_p(\beta)$ car $\beta = f(\beta)$ où $f(X) = X$. ■

Remarque 1.1.5 : Dans cette proposition, $\mathbb{F}_p[X]/(m_\beta(X))$ peut être vu comme un $\mathbb{F}_p$-espace vectoriel de dimension $deg(m_\beta(X))$, et donc $\mathbb{F}_p[X]/(m_\beta(X))$ a pour cardinal

$p^{deg(m_\beta(X))}$. Ceci nous permet de voir que, si q est une puissance d'un nombre premier et n est un entier naturel non nul, pour avoir un corps fini de cardinal q^n, il suffit d'avoir un polynôme irréductible de degré n à coefficients dans $\mathbb{F}_q$.

Définition 1.1.10 : *Le corps $\mathbb{F}_p(\beta)$ de la proposition 1.1.13 est appelé extension simple de $\mathbb{F}_p$ par adjonction de β.*

Proposition 1.1.14 : *Tout corps fini est une extension simple de son sous-corps premier.*

Preuve : En effet, il suffit de voir que si α est un élément primitif de $\mathbb{F}_{p^n}$ dont le polynôme minimal est de degré n, alors $\mathbb{F}_{p^n} = \mathbb{F}_p(\alpha)$.

1.2 Construction des corps finis

Comme nous l'avons déjà précisé, pour construire un corps fini de cardinal $q = p^n$, il suffit d'avoir un polynôme irréductible de degré n à coefficients dans $\mathbb{F}_p$. De la même façon, si $n = lm$, on peut aussi construire $\mathbb{F}_q$ en utilisant un polynôme irréductible de degré l à coefficients dans $\mathbb{F}_{p^m}$ ou en utilisant un polynôme irréductible de degré m à coefficients dans $\mathbb{F}_{p^l}$. L'existence des polynômes irréductibles de tout degré est garantie par l'existence des corps finis de cardinal p^n pour tout entier naturel non nul n ; mais la détermination de ces polynômes n'est pas facile lorsque leur degré est supérieur à 4 ou lorsque p est grand. Nous donnons dans la table suivante des polynômes irréductibles sur $\mathbb{F}_2$ ([8]).

$n = 1$	$X, \underline{X+1}$
$n = 2$	$\underline{X^2+X+1}$
$n = 3$	$\underline{X^3+X^2+1}, \underline{X^3+X+1}$
$n = 4$	$\underline{X^4+X^3+1}, \underline{X^4+X+1}, X^4+X^3+X^2+X+1$
$n = 5$	$\underline{X^5+X^4+X^3+X^2+1}, \underline{X^5+X^3+X^2+X+1}$ $\underline{X^5+X^4+X^3+X+1}, \underline{X^5+X^4+X^2+X+1}$ $\underline{X^5+X^3+1}, \underline{X^5+X^2+1},$
$n = 6$	$\underline{X^6+X^5+X^4+X+1}, \underline{X^6+X^5+X^2+X+1}$ $\underline{X^6+X^5+X^3+X^2+1}, \underline{X^6+X^4+X^3+X+1}$ $\underline{X^6+X^5+1}, \underline{X^6+X+1}$ $X^6+X^5+X^4+X^2+1, X^6+X^4+X^2+X+1$ X^6+X^3+1

TABLE 1.1 – Polynômes irréductibles sur $\mathbb{F}_2$ de degré inférieur à 6

Remarque 1.2.1 : Dans cette table, les polynômes non soulignés sont irréductibles, mais leurs racines ne sont pas des éléments primitifs du corps fini qu'ils permettent de construire.

Par exemple le polynôme $X^4+X^3+X^2+X+1$ est irréductible de degré 4 sur $\mathbb{F}_2$, mais est un diviseur de X^5-1 car $X^5-1=(X-1)(X^4+X^3+X^2+X+1)$. Donc toutes ses racines sont d'ordre 5 et ne peuvent pas être des éléments primitifs de $\mathbb{F}_{16}$.

Exemple : construction du corps fini $\mathbb{F}_{16}$

Le sous-corps premier de $\mathbb{F}_{16}$ est $\mathbb{F}_2=\{0;1\}$ car $16=2^4$.
On a : $16=2^4$. Donc pour construire $\mathbb{F}_{16}$, nous avons besoin d'un polynôme irréductible de degré 4 à coefficients dans $\mathbb{F}_2$. Choisissons un polynôme tel que toute racine soit un élément primitif de $\mathbb{F}_{16}$. D'après la table 1, nous avons le choix entre X^4+X^3+1 et X^4+X+1. Travaillons avec X^4+X+1.

Soit α une racine du polynôme X^4+X+1 ; alors $\alpha^4=\alpha+1$. Nous utilisons cette relation pour calculer les éléments de $\mathbb{F}_{16}$ qui sont l'élément nul 0 et les puissances de α.

0	$\alpha^3=\alpha^3$	$\alpha^7=\alpha^3+\alpha+1$	$\alpha^{11}=\alpha^3+\alpha^2+\alpha$
$\alpha^0=1$	$\alpha^4=\alpha+1$	$\alpha^8=\alpha^2+1$	$\alpha^{12}=\alpha^3+\alpha^2+\alpha+1$
$\alpha^1=\alpha$	$\alpha^5=\alpha^2+\alpha$	$\alpha^9=\alpha^3+\alpha$	$\alpha^{13}=\alpha^3+\alpha^2+1$
$\alpha^2=\alpha^2$	$\alpha^6=\alpha^3+\alpha^2$	$\alpha^{10}=\alpha^2+\alpha+1$	$\alpha^{14}=\alpha^3+1$

TABLE 1.2 – Les éléments du corps $\mathbb{F}_{16}$

Chapitre Deux

CODES LINÉAIRES SUR UN CORPS DE GALOIS

Après les corps finis, nous passons à présent à la notion de code linéaire sur un corps de Galois. Dans ce chapitre, nous allons d'abord donner les généralités sur les codes linéaires, ensuite nous nous attarderons sur les codes cycliques, enfin nous construirons trois codes cycliques binaires de longueur 15 dont nous étudierons la divisibilité dans le chapitre 3.

2.1 Généralités sur les codes linéaires

2.1.1 Définition et premières propriétés

Définition 2.1.1 : *Soit F un ensemble de cardinal q et n un entier naturel non nul.*
On appelle code C de longueur n sur F toute partie non vide C de F^n.
L'ensemble F est appelé alphabet du code.
Un élément $a = (a_0, a_1, \cdots, a_{n-1})$ de C est appelé mot du code.

Proposition 2.1.1 : *L'application $d_H : F^n \times F^n \longrightarrow \mathbb{N}$ définie par*

$$\forall a = (a_0; a_1; \cdots; a_{n-1}), b = (b_0; b_1; \cdots; b_{n-1}) \in F^n, d_H(a,b) = Card\{i \in \{0, \cdots, n-1\}/a_i \neq b_i\}$$

est une distance sur F^n.

Preuve : Soient $a, b, c \in F^n$.

On a : $d_H(a,b) = 0 \Leftrightarrow Card\{i \in \{0, \cdots, n-1\}/a_i \neq b_i\} = 0$

$\Leftrightarrow \{i \in \{0, \cdots, n-1\}/a_i \neq b_i\} = \varnothing$

$\Leftrightarrow \forall i \in \{0, \cdots, n-1\}, a_i = b_i\}$

$\Leftrightarrow a = b.$

$d_H(a,b) = Card\{i \in \{0,\cdots,n-1\}/a_i \neq b_i\} = Card\{i \in \{0,\cdots,n-1\}/b_i \neq a_i\} = d_H(b,a)$
Constatons que :

$$\{i \in \{0,\cdots,n-1\}/a_i = c_i\} \cap \{i \in \{0,\cdots,n-1\}/b_i = c_i\} \subset \{i \in \{0,\cdots,n-1\}/a_i = b_i\}.$$

Par passage au complémentaire dans $\{0,\cdots,n-1\}$, on obtient :

$$\{i \in \{0,\cdots,n-1\}/a_i \neq b_i\} \subset \{i \in \{0,\cdots,n-1\}/a_i \neq c_i\} \cup \{i \in \{0,\cdots,n-1\}/b_i \neq c_i\}.$$

Donc

$$Card(\{i \in \{0,\cdots,n-1\}/a_i \neq b_i\}) \leq Card(\{i \in \{0,\cdots,n-1\}/a_i \neq c_i\}) + Card(\{i \in \{0,\cdots,n-1\}/b_i \neq c_i\}).$$

D'où $d_H(a,b) \leq d_H(a,c) + d_H(c,b)$. ■

Définition 2.1.2 : *La distance d_H de la proposition 2.1.1 est appelée distance de Hamming*[1] *sur F^n.*

Définition 2.1.3 : *Soient $\mathbb{F}_q$ le corps de Galois de cardinal q et n un entier naturel non nul.*
On appelle code linéaire de longueur n et de dimension k sur $\mathbb{F}_q$ tout sous-espace vectoriel de $\mathbb{F}_q^n$ de dimension k.

Dans la suite, C un code linéaire de longueur n et de dimension k sur $\mathbb{F}_q$.

Définition 2.1.4 : *Soit $a = (a_0, a_1, \cdots, a_{n-1})$ un mot de $\mathbb{F}_q^n$.*
On appelle support du mot a le sous-ensemble $supp(a)$ de $[|0; n-1|]$ défini par

$$supp(a) = \{i \in \{0,1,\cdots,n-1\}, a_i \neq 0\}$$

Définition 2.1.5 : *Soit $a = (a_0, a_1, \cdots, a_{n-1})$ un mot de C.*
On appelle poids de Hamming de a l'entier $\omega_H(a)$ défini comme suit :

$$\omega_H(a) = Card(supp(a)).$$

Remarque 2.1.1 : Pour tous mots a et b de C, on a l'égalité suivante :

$$d_H(a,b) = \omega_H(a-b) \text{ et } \omega_H(a) = d_H(a,0).$$

1. Richard Wesley Hamming, mathématicien américain (1915-1998), prix Turing en 1968.

Définition 2.1.6 : *On appelle distance minimale de C l'entier d défini par :*

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

Notation 2.1.1 : On écrit $C(n,k,d)$ pour dire que C est un code de longueur n, de dimension k et de distance minimale d.

Définition 2.1.7 : *On appelle distribution de poids du code C la suite $(A_i)_{i=0,\cdots,n}$ où*

$$\forall i \in \{0,\cdots,n\}, A_i = card\{a \in C, \omega_H(a) = i\}.$$

Définition 2.1.8 : *On appelle polynôme énumérateur de poids du code C le polynôme.*

$$\omega_C(X) = \sum_{i=0}^{n} A_i X^i$$

où $(A_i)_{i=0,\cdots,n}$ est la distribution de poids du code C.

Théorème 2.1.1 : *(J. H. Griesmer 1960)*
Pour tout code linéaire non nul $C(n,k,d)$ sur $\mathbb{F}_q$, on a l'inégalité suivante :

$$\sum_{i=0}^{k-1} \lceil \frac{d}{q^i} \rceil \leq n$$

où pour tout $x \in \mathbb{R}$, $\lceil x \rceil$ est le plus petit entier relatif supérieur ou égal à x.

Définition 2.1.9 : *On appelle code de Griesmer*[2] *tout code linéaire $C(n,k,d)$ sur $\mathbb{F}_q$ tel que*

$$\sum_{i=0}^{k-1} \lceil \frac{d}{q^i} \rceil = n$$

Des exemples de ces code vous seront donnés à la page section 3.3.

2.1.2 Matrice génératrice et codes équivalents

Définition 2.1.10 : *On appelle matrice génératrice de $C(n,k,d)$ toute matrice $k \times n$ dont les lignes forment une base de C.*

2. James Hugo Griesmer, mathématicien américain (1929-2011).

Définition 2.1.11 : *Soit $f : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$ une application.*
On dit que f est une transformation monomiale s'il existe $(a_1, a_2, \cdots, a_n) \in \mathbb{F}_q^{*n}$ et il existe $\sigma \in S_n$ tels que :

$$\forall x = (x_1, x_2, \cdots, x_n) \in \mathbb{F}_q^n, f(x) = (a_1 x_{\sigma(1)}, a_2 x_{\sigma(2)}, \cdots, a_n x_{\sigma(n)}).$$

Définition 2.1.12 : *Deux codes linéaires C et C' sont dits équivalents s'il existe une transformation monomiale f telle que $C' = f(C)$ où $f(C) = \{f(x), x \in C\}$.*

Remarque 2.1.2 : Un code reste inchangé si on permute les lignes de sa matrice génératrice ou si on remplace une ligne par une combinaison linéaire de cette ligne avec d'autres lignes de cette matrice. En effectuant ces opérations sur les colonnes de la matrice génératrice d'un code, on obtient un code qui lui est équivalent.

Proposition 2.1.2 : *Deux codes équivalents C et C' ont la même longueur, la même dimension et la même distribution de poids.*

Preuve : Comme C et C' sont équivalents, il existe une transformation monomiale f telle que $C' = f(C)$. Puisque $f : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$ est une application, C et C' ont la même longueur. C étant un sous-espace vectoriel de $\mathbb{F}_q^n$ et f étant une transformation monomiale, f est un automorphisme de $\mathbb{F}_q^n$; donc $C' = f(C)$ est un sous-espace vectoriel de $\mathbb{F}_q^n$ de même dimension que C.

Prouvons que C et C' ont la même distribution de poids.
Soient $j \in \{1, 2, \cdots, n\}$ et $x = (x_1, x_2, \cdots, x_n) \in C$.
On a :

$$\begin{aligned}
\omega_H(x) = j &\Leftrightarrow card\{i \in \{1, 2, \cdots, n\}, x_i \neq 0\} = j \\
&\Leftrightarrow card\{i \in \{1, 2, \cdots, n\}, x_{\sigma(i)} \neq 0\} = j \qquad \text{car } \sigma \in S_n \\
&\Leftrightarrow card\{i \in \{1, 2, \cdots, n\}, a_i x_{\sigma(i)} \neq 0\} = j \quad \text{car } (a_1, a_2, \cdots, a_n) \in \mathbb{F}_q^{*n} \\
&\Leftrightarrow \omega_H(f(x)) = j
\end{aligned}$$

Donc C et C' ont la même distribution de poids. ■

2.1.3 Dual d'un code linéaire

Proposition 2.1.3 : *L'application* $(,) : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$ *définie par*

$$\forall x = (x_0, x_1, \cdots, x_{n-1}), y = (y_0, y_1, \cdots, y_{n-1}) \in \mathbb{F}_q^n, (x, y) = \left(\sum_{i=0}^{n-1} x_i y_i \right) \ mod\ p \quad \text{où } q = p^s$$

est un produit scalaire sur $\mathbb{F}_q^n$.

Preuve : Immédiate ■

Définition 2.1.13 : *Deux mots* x *et* y *de* $\mathbb{F}_q^n$ *sont dits orthogonaux s'ils le sont par rapport au produit scalaire* $(,)$.

Remarque 2.1.3 : Puisque C est un sous-espace vectoriel de $\mathbb{F}_q^n$ de dimension k, son orthogonal $C^\perp$ par rapport au produit scalaire $(,)$, est un sous-espace vectoriel de $\mathbb{F}_q^n$ de dimension $n-k$; donc un code linéaire de longueur n et de dimension $n-k$.

Définition 2.1.14 : *On appelle code dual du code linéaire* C *le code* $C^\perp$ *défini par*

$$C^\perp = \{y \in \mathbb{F}_q^n / \forall x \in C, (x, y) = 0\}.$$

Définition 2.1.15 : *Le code* C *est dit orthogonal ou faiblement autodual si* $C \subset C^\perp$.

Définition 2.1.16 : *Le code* C *est dit autodual si* $C = C^\perp$.

Définition 2.1.17 : *Toute matrice génératrice* H *du code dual* $C^\perp$ *de* C *est appelée matrice de contrôle de* C.

Proposition 2.1.4 : (i) *Si* G *est une matrice génératrice de* C, *alors*

$$C^\perp = \{x \in \mathbb{F}_q^n, x^t G = 0\}.$$

(ii) *Pour tout code Linéaire* C, $C^{\perp\perp} = C$.

(iii) *Si* H *est une matrice* $(n-k) \times k$ *de rang* $n-k$ *et* G *une matrice génératrice de* C, *alors* H *est une matrice de contrôle de* C *si et seulement si*

$$^tHG = {}^tGH = 0.$$

Preuve : (i) Si G est une matrice génératrice de C, alors les lignes de G forment une base de C.

$$x \in C^{\perp} \iff x \text{ est orthogonal à tous les vecteurs d'une base de } C$$

Soit $x \in C$, on a : $\iff x$ est orthogonal à toutes les lignes de G

$\iff x^t G = 0.$

D'où $C^{\perp} = \{x \in \mathbb{F}_q^n, x^t G = 0\}$.

(ii) Nous savons que si $dim(C) = k$, alors $dim(C^{\perp}) = n - k$.

Donc $dim(C^{\perp\perp}) = n - (n-k) = k = dim(C)$.

Puisque $C \subset C^{\perp\perp}$, alors $C^{\perp\perp} = C$.

(iii) Soit C' le code dont H est une matrice génératrice.

Si ${}^tHG = {}^t GH = 0$, alors les lignes de H, qui forment une base de C', sont toutes orthogonales aux lignes de G qui forment une base de C ; donc $C' \subset C^{\perp}$. Mais C' est de dimension $n - k = dim(C^{\perp})$, d'où $C' = C^{\perp}$.

Réciproquement, si H est une matrice de contrôle de C, alors les lignes de H forment une base de $C^{\perp}$ et celles de G forment une base de C. Par définition de $C^{\perp}$, on obtient ${}^tHG = {}^t GH = 0$. ■

2.2 Codes cycliques

2.2.1 Généralités

Définition 2.2.1 : *On appelle opérateur de shift (à droite) l'application*

$$\begin{array}{rccc} \tau : & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^n \\ & (x_0, x_2, \cdots, x_{n-1}) & \longmapsto & (x_{n-1}, x_0, \cdots, x_{n-2}) \end{array}$$

Définition 2.2.2 : *Le code linéaire C est dit cyclique si C est stable par l'opérateur de shift, c'est-à-dire que $\tau(C) = C$.*

Proposition 2.2.1 : *Si C est un code cyclique, alors son orthogonal $C^{\perp}$ est aussi un code cyclique.*

Preuve : Soit C un code linéaire cyclique, alors C est stable par le shift.

Soit $a \in \mathbb{F}_q^n$, on a :

$$\begin{aligned} a \in C^\perp &\Leftrightarrow a.c = 0 \ \forall c \in C \\ &\Leftrightarrow \tau(a).\tau(c) = 0 \ \forall c \in C \qquad \text{car } a.c = \tau(a).\tau(c) \\ &\Leftrightarrow \tau(a).c' = 0 \ \forall c' \in C \qquad \text{car C est cyclique} \\ &\Leftrightarrow \tau(a) \in C^\perp \end{aligned}$$

D'où $\tau(C^\perp) = C^\perp$, donc $C^\perp$ est un code linéaire cyclique. ■

Notation 2.2.1 : Notons $(X^n - 1)$ l'idéal de l'anneau $\mathbb{F}_q[X]$ engendré par $X^n - 1$. En considérant l'épimorphisme canonique π défini de $\mathbb{F}_q[X]$ vers $\mathbb{F}_q[X]/(X^n - 1)$, nous désignons par x l'image de X par π ; c'est-à-dire $x = X + (X^n - 1)$, par suite nous désignerons $\mathbb{F}_q[X]/(X^n - 1)$ par $\mathbb{F}_q[x]$. $\mathbb{F}_q[x]$ est en effet l'anneau des polynômes nuls ou de degré strictement inférieur à n, à coefficients dans $\mathbb{F}_q$. $\mathbb{F}_q[x]$ est aussi un $\mathbb{F}_q$-espace vectoriel.

Proposition 2.2.2 : *L'application*

$$\begin{aligned} \mathcal{R} : \quad \mathbb{F}_q^n &\longrightarrow \mathbb{F}_q[x] \\ a = (a_0, a_2, \cdots, a_{n-1}) &\longmapsto \mathcal{R}(a) = \sum_{i=o}^{n-1} a_i x^i \end{aligned}$$

est un isomorphisme de $\mathbb{F}_q$-espaces vectoriels.

Preuve : Immédiate ■

Proposition 2.2.3 : *Le code linéaire C est cyclique si et seulement si $\mathcal{R}(C)$ est idéal de $\mathbb{F}_q[x]$.*

Preuve : Remarquons que pour tout mot c de C, $\mathcal{R}(\tau(c)) = x.\mathcal{R}(c)$.

Supposons que C est cyclique.

$$\begin{aligned} C \text{ est cyclique} &\Longrightarrow \forall c \in C, \tau(c) \in C \\ &\Longrightarrow \forall c \in C, \mathcal{R}(\tau(c)) \in \mathcal{R}(C) \\ &\Longrightarrow \forall c \in C, x.\mathcal{R}(c) \in \mathcal{R}(C) \\ &\Longrightarrow x.\mathcal{R}(C) \subset \mathcal{R}(C) \\ &\Longrightarrow \forall j \in \{0, 1, \cdot, n-1\}, x^j.\mathcal{R}(C) \subset \mathcal{R}(C) \\ &\Longrightarrow \forall b(x) = \sum_{j=0}^{n-1} a_i x^i \in \mathbb{F}_q[x], b(x).\mathcal{R}(C) \subset \mathcal{R}(C) \\ &\Longrightarrow \mathcal{R}(C) \text{ est un idéal de } \mathbb{F}_q[x] \end{aligned}$$

Réciproquement, si $\mathcal{R}(C)$ est un idéal de $\mathbb{F}_q[x]$, alors $\forall c \in C, x.\mathcal{R}(c) \in \mathcal{R}(C)$, c'est-à-dire $\forall c \in C, \mathcal{R}(\tau(c)) \in \mathcal{R}(C)$. Comme $\mathcal{R}$ est un isomorphisme, alors $\forall c \in C, \tau(c) \in C$; donc C est cyclique. ■

2.2.2 Matrice génératrice et matrice de contrôle d'un code cyclique

Comme $\mathbb{F}_q$ est un corps, alors $\mathbb{F}_q[x]$ est un anneau principal. On obtient donc la proposition suivante.

Proposition 2.2.4 : *Le code linéaire C est cyclique si et seulement si $\mathcal{R}(C)$ est un idéal principal de $\mathbb{F}_q[x]$.*

Définition 2.2.3 : *Si C est cyclique, on appelle polynôme générateur de C le polynôme unitaire $g(x) \in \mathbb{F}_q[x]$ tel $\mathcal{R}(C)$ soit l'idéal de $\mathbb{F}_q[x]$ engendré par $g(x)$.*

Proposition 2.2.5 : *Si $C(n,k,d)$ est cyclique de polynôme générateur $g(x)$, alors la famille $\{1.g(x), x.g(x), \cdots, x^{k-1}.g(x)\}$ est une base de $\mathcal{R}(C)$.*

Preuve : La famille $\{1, x, \cdots, x^{k-1}\}$ est une famille libre de $\mathbb{F}_q[x]$, donc $\{1.g(x), x.g(x), \cdots, x^{k-1}.g(x)\}$ est aussi une famille libre de $\mathbb{F}_q[x]$ car $g(x) \neq 0$. Comme $\{1.g(x), x.g(x), \cdots, x^{k-1}.g(x)\} \subset \mathcal{R}(C)$, alors $\{1.g(x), x.g(x), \cdots, x^{k-1}.g(x)\}$ est une famille de k vecteurs libres de l'espace vectoriel $\mathcal{R}(C)$; mais $dim(\mathcal{R}(C)) = dim(C) = k$, donc $\{1.g(x), x.g(x), \cdots, x^{k-1}.g(x)\}$ est une base de C. ■

Remarque 2.2.1 : Par abus, on confond un code cyclique à son image $\mathcal{R}(C)$ par $\mathcal{R}$. Le code C s'obtient en multipliant les messages qui sont des polynômes de degré au plus égal à $k-1$ par $g(x)$. Puisque les mots de C sont des polynômes de degré au plus égal à $n-1$, alors $g(x)$ est un polynôme de degré $n-k$. De plus, C étant un idéal de $\mathbb{F}_q[x] = \mathbb{F}_q[X]/(X^n-1)$, alors $g(X)$ divise $X^n - 1$.

Proposition 2.2.6 : *Si C est cyclique de polynôme générateur $g(x) = \sum_{i=0}^{n-k} g_i x^i$, alors une matrice génératrice de C est :*

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

Preuve : Ceci est du au fait que les lignes de G sont, dans l'ordre, les éléments de la base $\{1.g(x), x.g(x), \cdots, x^{k-1}.g(x)\}$ de C. ■

Définition 2.2.4 : *Soit $h(x) \in \mathbb{F}_q[x]$ tel que $h(X)g(X) = X^n - 1$.*
Le polynôme $h(x)$ est appelé polynôme correcteur de C.

Proposition 2.2.7 : *Si $h(x)$ est un polynôme de contrôle de $C = (g(x))$, alors pour tout $f(x) \in \mathbb{F}_q[x]$, $f(x) \in C$ si et seulement si $f(x)h(x) = 0$.*

Preuve : En effet, si $f(x) \in \mathbb{F}_q[x]$,

$$\begin{aligned} f(x) \in C &\Leftrightarrow g(x) | f(x) \\ &\Leftrightarrow \exists l(x) \in \mathbb{F}_q[x] \ / \ f(x) = l(x)g(x) \\ &\Leftrightarrow \exists l(x) \in \mathbb{F}_q[x] \ / \ f(x)h(x) = l(x)g(x)h(x) \\ &\Leftrightarrow f(x)h(x) = 0 \qquad \text{car } g(x)h(x) = 0 \end{aligned}$$

■

Proposition 2.2.8 : *Si $h(x) = h_0 + h_1 x + \cdots + h_k x^k$ est le polynôme correcteur de C, alors*

$$H = \begin{pmatrix} 0 & 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 \\ 0 & 0 & \cdots & h_k & h_{k-1} & h_{k-2} & \cdots & h_0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix} \in M_{n-k,n}(\mathbb{F}_q)$$

est une matrice de contrôle de C.

Preuve : On peut écrire $h(x) = \sum_{j=0}^{k} h_j x^j = \sum_{j=0}^{n-1} h_j x^j$ où $h_j = 0$ pour $k+1 \leq j$.
Soit $f(x) = \sum_{i=0}^{n-1} f_i x^i \in \mathbb{F}_q[x]$, alors d'après la proposition 2.2.7, $f(x) \in C$ si et seulement si $f(x)h(x) = 0$.
Calculons $f(x)h(x)$.

$$\begin{aligned}
h(x)f(x) &= \sum_{j=0}^{n-1} h_j x^j \sum_{i=0}^{n-1} f_i x^i \\
&= \sum_{j=0}^{n-1}\sum_{i=0}^{n-1} h_j f_i x^{i+j} && \text{où } x^{i+j} = x^{i+j-n} \text{ si } i+j \geq n \\
&= \sum_{l=0}^{2n-1}\sum_{i=0}^{n-1} h_{l-i} f_i x^l && \text{où } l = i+j \\
&= \sum_{l=0}^{n-1}\sum_{i=0}^{n-1} h_{l-i} f_i x^l + \sum_{l=n}^{2n-1}\sum_{i=0}^{n-1} h_{l-i} f_i x^l \\
&= \sum_{l=0}^{n-1}\sum_{i=0}^{n-1} h_{l-i} f_i x^l + \sum_{l'=0}^{n-1}\sum_{i=0}^{n-1} h_{l'+n-i} f_i x^{l'} \\
&= \sum_{l=0}^{n-1}\left(\sum_{i=0}^{n-1} h_{l-i} f_i + \sum_{i=0}^{n-1} h_{l+n-i} f_i\right) x^l
\end{aligned}$$

La deuxième somme sur i dans la dernière ligne est nulle dès que $l \geq k$ puisque $l+n-i \geq k+1$ et donc $h_{l+n-i} = 0$ (car le polynôme h est de degré k).

Donc pour $l \in \{k, \cdots, n-1\}$, on n'a qu'une seule somme et en revenant à $h(x)f(x) = 0$, on obtient :

$$\sum_{i=0}^{n-1} h_{l-i} f_i = 0, \ \forall l \in \{k, \cdots, n-1\}$$

Ainsi, une condition nécessaire consiste à un système de $k-n$ équations linéaires

$$\sum_{i=0}^{n-1} h_{l-i} f_i = 0, \ l \in \{k, \cdots, n-1\}$$

De plus, si $i \leq l-k-1$, alors $k+1 \leq l-i$, donc $h_{l-i} = 0$.

Ce système de $k-n$ équations s'écrit donc :

$$(S)\left\{\begin{array}{lllll}
(l = n-1) & 0.f_0 + \cdots + 0.f_{n-k-2} + h_k.f_{n-k-1} + \cdots + h_{k-1}.f_{n-k} + \cdots + h_0 \cdot f_{n-1} & = & 0 \\
(l = n-2) & 0.f_0 + \cdots + h_k.f_{n-k-2} + h_{k-1}.f_{n-k-1} + \cdots + h_0.f_{n-2} + 0.f_{n-1} & = & 0 \\
\cdots & \cdots\cdots\cdots\cdots\cdots\cdots & \cdots & \cdots \\
(l = k) & h_k.f_0 + \cdots + h_0.f_k + 0.f_{k+1} + \cdots + 0.f_{n-2} + 0.f_{n-1} & = & 0
\end{array}\right.$$

ou bien

$$\begin{pmatrix}
0 & 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 \\
0 & 0 & \cdots & h_k & h_{k-1} & h_{k-2} & \cdots & h_0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & \cdots & \cdots & 0
\end{pmatrix}
\begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ \vdots \\ f_{n-1} \end{pmatrix} = 0$$

Le code C est donc inclus dans l'ensemble solution du système (S). Comme h est un polynôme de degré k, alors $h_k \neq 0$. Donc la matrice H est de rang $n-k$, d'où l'ensemble solution de (S) est un espace vectoriel de dimension $n-(n-k) = k$. Comme C est de

dimension k, alors l'ensemble solution de (S) est exactement le code C.
Ainsi, H est une matrice de contrôle de C. ■

Remarque 2.2.2 : Nous avons vu à la proposition 2.2.1 que le dual $C^{\perp}$ d'un code cyclique C est un code cyclique. Toutefois, son polynôme générateur n'est pas le polynôme correcteur de C.

Proposition 2.2.9 : *Si C est un code cyclique de polynôme correcteur $h(x) = \sum_{i=0}^{k} h_i x^i$, alors le polynôme générateur de $C^{\perp}$ est :*

$$g^{\perp}(x) = h_k + h_{k-1}x + \cdots + h_0 x^k = x^k h(\frac{1}{x})$$

Preuve : En reversant l'ordre des lignes de la matrice de contrôle de C donnée à la proposition 2.2.8, le code dual $C^{\perp}$ de C reste inchangé et on obtient la matrice

$$H' = \begin{pmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 \end{pmatrix}$$

qui est encore une matrice génératrice de $C^{\perp}$. Donc $\{g^{\perp}(x), xg^{\perp}(x), \cdots, x^{n-k-1}g^{\perp}(x)\}$ est une base de $\mathcal{R}(C^{\perp})$. D'où $g^{\perp}(x)$ est un polynôme générateur de $C^{\perp}$. ■

2.2.3 Ensemble de définition d'un code cyclique et la borne BCH

Dans cette section, $C(n, k, d)$ est un code cyclique sur $\mathbb{F}_p$ (avec $n \wedge p = 1$) de polynôme générateur $g(x)$ et β est une racine $n^{ième}$ primitive de l'unité.

Définition 2.2.5 : *On appelle zéro du code C toute racine de son polynôme générateur $g(x)$.*

Remarque 2.2.3 : Rappelons que le polynôme générateur d'un code cyclique de longueur n est un diviseur de $X^n - 1$. Ainsi, les zéros d'un code cyclique de longueur n sont toujours des racines $n^{ième}$ de l'unité. Puisque β est une racine $n^{ième}$ primitive de l'unité, alors chaque zéro de C correspond à une unique puissance de β.

Définition 2.2.6 : *On appelle ensemble de définition du code cyclique C le sous-ensemble T de $\{0;1;\cdots;n-1\}$ défini par :*

$$T = \{i \in \{0;1;\cdots;n-1\}; \beta^i \text{ zéro de } C\}.$$

Définition 2.2.7 : *On appelle ensemble de parité du code cyclique C, le complémentaire de son ensemble de définition dans $\{0;1;\cdots;n-1\}$.*

Notation 2.2.2 : L'ensemble de parité est noté U.

Définition 2.2.8 : *On appelle non-zéro du code cyclique C toute racine $n^{\text{ième}}$ de l'unité qui n'est pas un zéro de C.*

Proposition 2.2.10 : *Si C est un code cyclique d'ensemble de parité U, alors l'ensemble de définition de son dual $C^\perp$ est :*

$$T^\perp = \{t | \exists s \in U, -s \equiv t \mod n\}.$$

Preuve : Soit C un code cyclique de polynôme correcteur $h(x) = \sum_{i=0}^{k} h_i x^i$. Puisque $X^n - 1 = g(X)h(X)$ et $X^n - 1$ n'a pas de racines multiples, alors l'ensemble des racines de $h(x)$ est $\{\beta^i, i \in U\}$. D'après la proposition 2.2.9, le polynôme générateur de $C^\perp$ est $g^\perp(x) = x^k h(\frac{1}{x})$. D'où l'ensemble des racines de $g^\perp(x)$ est $\{\beta^{-i}, i \in U\}$. Ainsi, le domaine de définition de $C^\perp$ est $T^\perp = \{t \mid \exists s \in U, -s \equiv t \bmod n\}$. ■

Remarque 2.2.4 : Puisque un polynôme est entièrement déterminé par la donnée de ses racines et leurs ordres de multiplicité, alors un code cyclique peut être défini par la donnée de son ensemble de définition T ou son ensemble de parité U.

Proposition 2.2.11 : *Si $T = \{i_1;\cdots;i_l\}$ est l'ensemble de définition du code cyclique C, alors*

$$H = \begin{pmatrix} 1 & \beta^{i_1} & \beta^{2i_1} & \cdots & \beta^{(n-1)i_1} \\ 1 & \beta^{i_2} & \beta^{2i_2} & \cdots & \beta^{(n-1)i_2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \beta^{i_l} & \beta^{2i_l} & \cdots & \beta^{(n-1)i_l} \end{pmatrix}$$

est une matrice de contrôle de C.

Preuve : Soit $c(x) = \sum_{j=0}^{n-1} c_j x^j \in \mathbb{F}_q[x]$ et $c = (c_0; \cdots; c_{n-1})$. Alors

$$
\begin{aligned}
c \in C &\Leftrightarrow g(x) | c(x) \\
&\Leftrightarrow \forall k \in \{1; \cdots; l\}, \beta^{i_k} \text{ est une racine de } c(x) \text{ car } g(x) \text{ n'a pas de racines multiples} \\
&\Leftrightarrow \forall k \in \{1; \cdots; l\}, c(\beta^{i_k}) = 0 \\
&\Leftrightarrow \forall k \in \{1; \cdots; l\}, \sum_{j=0}^{n-1} c_j \beta^{j i_k} \\
&\Leftrightarrow \begin{cases} c_0 + c_1\beta^{i_1} + c_2\beta^{2i_1} + \cdots + c_{n-1}\beta^{(n-1)i_1} = 0 \\ c_0 + c_1\beta^{i_2} + c_2\beta^{2i_2} + \cdots + c_{n-1}\beta^{(n-1)i_2} = 0 \\ \cdots + \cdots + \cdots + \cdots + \cdots = 0 \\ c_0 + c_1\beta^{i_l} + c_2\beta^{2i_l} + \cdots + c_{n-1}\beta^{(n-1)i_l} = 0 \end{cases} \\
&\Leftrightarrow cH^t = 0
\end{aligned}
$$

D'où le résultat. ■

Théorème 2.2.1 : *(de la borne BCH[3])*

Soit C un code cyclique dont l'ensemble de définition contient δ éléments consécutifs. Alors la distance minimale de C est supérieure ou égale à $\delta + 1$.

Lemme 2.1 : *Soit C un code linéaire de matrice de contrôle H et δ un entier naturel non nul. Si δ colonnes quelconques de H sont toujours linéairement indépendantes, alors la distance minimale de C est supérieure ou égale à $\delta + 1$.*

Preuve : Soit C un code linéaire de matrice de contrôle H et δ un entier naturel non nul tel que δ colonnes quelconques de H soient toujours linéairement indépendantes.

Posons $H = (C_0 C_1 \cdots C_{n-1})$ où C_i est la $(i+1)^{ième}$ colonne de H.

Nous allons procéder par l'absurde.

Supposons qu'il existe un mot non nul $c \in C$ tel que $\omega_H(c) < \delta + 1$.

On a :

$$\begin{aligned} c \in C &\Leftrightarrow Hc^t = 0 \\ &\Leftrightarrow (C_0 C_1 \cdots C_{n-1})c^t = 0 \\ &\Leftrightarrow c_0 C_0 + c_1 C_1 + \cdots + c_{n-1} C_{n-1} = 0 \qquad (\star) \end{aligned}$$

Puisque le mot c est de poids $\omega_H(c) < \delta + 1$, alors dans la ligne $(\star)$ on a à faire à une combinaison linéaire (à coefficients non tous nuls) d'au plus δ colonnes de H qui est nulle ; ce qui contredit le fait que δ colonnes quelconques de H soient linéairement indépendants. ■

Preuve : (du théorème 2.2.1)

Soit C un code cyclique dont l'ensemble de définition contient δ éléments consécutifs $b; b+1, \cdots; b+\delta-1$. On peut donc écrire $T = \{b; b+1, \cdots; b+\delta-1; i_{\delta+1}; \cdots; i_l\}$.

D'après la proposition 2.2.11, une matrice de contrôle de C est

$$H = \begin{pmatrix} 1 & \beta^{b} & \beta^{2b} & \cdots & \beta^{(n-1)b} \\ 1 & \beta^{b+1} & \beta^{2(b+1)} & \cdots & \beta^{(n-1)(b+1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \beta^{b+\delta-1} & \beta^{2(b+\delta-1)} & \cdots & \beta^{(n-1)(b+\delta-1)} \\ 1 & \beta^{\delta+1} & \beta^{2(\delta+1)} & \cdots & \beta^{(n-1)(\delta+1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \beta^{i_l} & \beta^{2i_l} & \cdots & \beta^{(n-1)i_l} \end{pmatrix}$$

3. Raj **B**ose (né en Inde, nationalité américaine), Ray-**C**haudhuri (américain), Alexis **H**ocquenghem (francais)

Montrons que δ colonnes quelconques de H sont linéairement indépendantes et utilisons le lemme précédent pour conclure.

Soient $j_1; j_2; \cdots; j_\delta \subset \{0; \cdots; n-1\}$ les numéros de δ colonnes quelconques de H. Pour montrer que ces colonnes sont linéairement indépendantes, il suffit de montrer que la matrice $M = (C_{j_1} C_{j_2} \cdots C_{j_\delta})$ formée par celles-ci est de rang δ, c'est-à-dire en extraire une sous-matrice inversible de dimension $\delta \times \delta$.

Considérons la matrice A formée par les δ premières lignes de M ; alors

$$A = \begin{pmatrix} \beta^{j_1 b} & \beta^{j_2 b} & \cdots & \beta^{j_\delta b} \\ \beta^{j_1 (b+1)} & \beta^{j_2 (b+1)} & \cdots & \beta^{j_\delta (b+1)} \\ \cdots & \cdots & \cdots & \cdots \\ \beta^{j_1 (b+\delta-1)} & \beta^{j_2 (b+\delta-1)} & \cdots & \beta^{j_\delta (b+\delta-1)} \end{pmatrix}.$$

Puisque les éléments de chaque colonne de la matrice A suivent une progression géométrique, alors le déterminant de la matrice A est de Vandermonde [4] et est donné par l'égalité

$$|A| = \beta^{(\sum_{k=1}^{\delta} j_k b)} \prod_{v<u} (\beta^{j_u} - \beta^{j_v}).$$

Soient $u, v \in \{1; \cdots; \delta\}$ tels que $v < u$; comme β est une racine primitive de l'unité, alors $\beta^{j_u} - \beta^{j_v} \neq 0$. Ainsi, $\prod_{v<u} (\beta^{j_u} - \beta^{j_v}) \neq 0$; et par conséquent $|A| \neq 0$. D'où A est de rang δ ; par conséquent M aussi.

En appliquant le lemme, on conclut que la distance minimale du code C est supérieure ou égale à $\delta + 1$. ■

Remarque 2.2.5 : Le théorème de la borne BCH permet de construire des codes corrigeant un certain nombre d'erreurs fixées dès le départ. Nous pouvons donc construire des codes corrigeant tant d'erreurs que nous le souhaitons, mais la longueur peut parfois être un facteur gênant dans la mise en pratique.

2.2.4 Construction d'un code cyclique sur $\mathbb{F}_p$

Définition 2.2.9 : *Soient n un entier naturel non nul et distinct de 1, p un entier premier et $j \in \{0; 1; \cdots; n-1\}$. On appelle classe $p-$cyclotomique de j modulo n le sous-ensemble*

4. déterminant d'une matrice dont les lignes (ou colonnes) suivent des progressions géométriques.

$\Gamma_p(j)$ *de* $\{0;\cdots;n-1\}$ *défini par*

$$\Gamma_p(j) = \{j; jp; jp^2; \cdots jp^{s-1}\}$$

où s *est le plus petit entier tel que* $jp^s \equiv j \mod n$.

Remarque 2.2.6 : Si m est la plus petite puissance de p telle que $n|p^m - 1$, alors $\mathbb{F}_{p^m}$ est le corps de décomposition de $X^n - 1$ sur $\mathbb{F}_p$ (cela provient du fait que $X^n - 1$ se décompose entièrement dans $\mathbb{F}_{p^k}$ si et seulement $X^n - 1$ divise $X^{p^k-1} - 1$, si et seulement si $n|p^k - 1$).

Pour construire un code cyclique de longueur n sur $\mathbb{F}_p$, on peut procéder comme suit :

i. chercher le plus petit entier m tel que $n|p^m - 1$;

ii. déterminer un élément primitif de $\mathbb{F}_{p^m}$;

iii. dresser les tables d'addition et de multiplication sur $\mathbb{F}_{p^m}$;

iv. déterminer les classes p–cyclotomiques modulo n ;

v. écrire $X^n - 1$ comme produit de facteurs linéaires, regrouper ces facteurs linéaires selon les classes p–cyclotomiques modulo n ;

vi. utiliser les tables d'addition et de multiplication sur $\mathbb{F}_{p^m}$ pour retrouver les facteurs irréductibles de $X^n - 1$;

vii. constituer le polynôme générateur $g(X)$ du code en choisissant ses facteurs irréductibles parmi ceux de $X^n - 1$;

viii. développer le polynôme $g(X)$ et dresser une matrice génératrice du code.

Exemple : construction de codes cycliques binaires de longueur 15

Nous sommes dans le cas où $p = 2$ et $n = 15$.

i. Le plus petit entier m tel que $15|2^m - 1$ est $m = 4$.

ii. Nous choisissons comme élément primitif de $\mathbb{F}_{16}$ une racine α de $X^4 + X + 1$.

iii. Pour la multiplication, c'est facile : pour tous $i, j \in [|0; 14|], \alpha^i \times \alpha^j = \alpha^k$ où $i + j \equiv k \mod 15$. Pour l'addition, nous avons la table suivante :

+	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	1	0	α^4	α^8	α^{14}	α	α^{10}	α^{13}	α^9	α^2	α^7	α^5	α^{12}	α^{11}	α^6	α^3
α	α	α^4	0	α^5	α^9	1	α^2	α^{11}	α^{14}	α^{10}	α^3	α^8	α^6	α^{13}	α^{12}	α^7
α^2	α^2	α^8	α^5	0	α^6	α^{10}	α	α^3	α^{12}	1	α^{11}	α^4	α^9	α^7	α^{14}	α^{13}
α^3	α^3	α^{14}	α^9	α^6	0	α^7	α^{11}	α^2	α^4	α^{13}	α	α^{12}	α^5	α^{10}	α^8	1
α^4	α^4	α	1	α^{10}	α^7	0	α^8	α^{12}	α^3	α^5	α^{14}	α^2	α^{13}	α^6	α^{11}	α^9
α^5	α^5	α^{10}	α^2	α	α^{11}	α^8	0	α^9	α^{13}	α^4	α^6	1	α^3	α^{14}	α^7	α^{12}
α^6	α^6	α^{13}	α^{11}	α^3	α^2	α^{12}	α^9	0	α^{10}	α^{14}	α^5	α^7	α	α^4	1	α^8
α^7	α^7	α^9	α^{14}	α^{12}	α^4	α^3	α^{13}	α^{10}	0	α^{11}	1	α^6	α^8	α^2	α^5	α
α^8	α^8	α^2	α^{10}	1	α^{13}	α^5	α^4	α^{14}	α^{11}	0	α^{12}	α	α^7	α^9	α^3	α^6
α^9	α^9	α^7	α^3	α^{11}	α	α^{14}	α^6	α^5	1	α^{12}	0	α^{13}	α^2	α^8	α^{10}	α^4
α^{10}	α^{10}	α^5	α^8	α^4	α^{12}	α^2	1	α^7	α^6	α	α^{13}	0	α^{14}	α^3	α^9	α^{11}
α^{11}	α^{11}	α^{12}	α^6	α^9	α^5	α^{13}	α^3	α	α^8	α^7	α^2	α^{14}	0	1	α^4	α^{10}
α^{12}	α^{12}	α^{11}	α^{13}	α^7	α^{10}	α^6	α^{14}	α^4	α^2	α^9	α^8	α^3	1	0	α	α^5
α^{13}	α^{13}	α^6	α^{12}	α^{14}	α^8	α^{11}	α^7	1	α^5	α^3	α^{10}	α^9	α^4	α	0	α^2
α^{14}	α^{14}	α^3	α^7	α^{13}	1	α^9	α^{12}	α^8	α	α^6	α^4	α^{11}	α^{10}	α^5	α^2	0

TABLE 2.1 – Addition dans $\mathbb{F}_{16}$

iv. Les classes 2–cyclotomiques modulo 15 sont :

$$\Gamma_2(0) = \{0\}, \Gamma_2(1) = \{1;2;4;8\}, \Gamma_2(3) = \{3;6;;9;12\}, \Gamma_2(5) = \{5;10\}, \Gamma_2(7) = \{7;11;13;14\}.$$

v. Décomposition de $X^{15}-1$ en produit de facteurs linéaires :

$$X^{15}-1 = \prod_{i=0}^{14}(X-\alpha^i).$$

En regroupant les facteurs linéaires selon les classes cyclotomiques, on obtient :

$$X^{15}-1 = (X-1)\prod_{i\in\{1;2;4;8\}}(X-\alpha^i)\prod_{i\in\{3;6;9;12\}}(X-\alpha^i)\prod_{i\in\{5;10\}}(X-\alpha^i)\prod_{i\in\{7;11;13;14\}}(X-\alpha^i)$$

vi. On développe chaque bloc de l'écriture ci-dessus en utilisant la table d'addition sur $\mathbb{F}_{16}$. En caractéristique 2, le signe importe peu. On a :

$$\begin{aligned}\prod_{i\in\{1;2;4;8\}}(X-\alpha^i) &= (X+\alpha)(X+\alpha^2)(X+\alpha^4)(X+\alpha^8)\\ &= X^4+(\alpha+\alpha^2+\alpha^4+\alpha^8)X^3+(\alpha^3+\alpha^5+\alpha^9+\alpha^6+\alpha^{10}+\alpha^{12})X^2+(\alpha^7+\alpha^{11}+\alpha^{13}+\alpha^{14})X+\alpha^{1+2+4+8}\\ &= X^4+X+1\end{aligned}$$

$$\begin{aligned}\prod_{i\in\{3;6;9;12\}}(X-\alpha^i) &= (X+\alpha^3)(X+\alpha^6)(X+\alpha^9)(X+\alpha^{12})\\ &= X^4+(\alpha^3+\alpha^6+\alpha^9+\alpha^{12})X^3+(\alpha^9+\alpha^{12}+1+1+\alpha^3+\alpha^6)X^2+(\alpha^3+\alpha^6+\alpha^9+\alpha^{12})X+\alpha^{3+6+9+12}\\ &= X^4+X^3+X^2+X+1\end{aligned}$$

$$\begin{aligned}\prod_{i\epsilon\{5;10\}}(X-\alpha^i) &= (X+\alpha^5)(X+\alpha^{10})\\ &= X^2+(\alpha^5+\alpha^{10})X+\alpha^{5+10}\\ &= X^2+X+1\end{aligned}$$

$$\begin{aligned}\prod_{i\epsilon\{7;11;13;14\}}(X-\alpha^i) &= (X+\alpha^7)(X+\alpha^{11})(X+\alpha^{13})(X+\alpha^{14})\\ &= X^4+(\alpha^7+\alpha^{11}+\alpha^{13}+\alpha^{14})X^3+(\alpha^3+\alpha^5+\alpha^9+\alpha^6+\alpha^{10}+\alpha^{12})X^2+(\alpha+\alpha^2+\alpha^4+\alpha^8)X\\ &\quad +\alpha^{7+11+13+14}\\ &= X^4+X^3+1\end{aligned}$$

Donc

$$X^{15}+1=(X+1)(X^2+X+1)(X^4+X+1)(X^4+X^3+X^2+X+1)(X^4+X^3+1)$$

vii. Le polynôme générateur du code est un diviseur de $X^{15}+1$. Nous choisissons trois polynômes $g_1(X)=(X+1)(X^2+X+1)(X^4+X+1)(X^4+X^3+X^2+X+1)$, $g_2(X)=(X-1)(X^2+X+1)(X^4+X+1)(X^4+X^3+1)$ et $g_3(X)=(X^2+X+1)(X^4+X^3+X^2+X+1)(X^4+X^3+1)$ qui nous conduiront à trois codes cycliques $C_1=(g_1(x))$, $C_2=(g_2(x))$ et $C_3=(g_3(x))$

viii. En développant les polynômes $g_1(x)$, $g_2(x)$ et $C_3=(g_3(x))$ on obtient $g_1(X)=x^{11}+x^{10}+x^9+x^8+x^6+x^4+x^3+1$, $g_2(X)=x^{11}+x^{10}+x^6+x^5+x+1$ et $g_3(X)=x^{10}+x^9+x^8+x^6+x^5+x^2+1$.

D'après la proposition 2.2.6, des matrices génératrices des codes C_1, C_2 et C_3 sont respectivement :

$$G_1=\begin{pmatrix}1&0&0&1&1&0&1&0&1&1&1&1&0&0&0\\0&1&0&0&1&1&0&1&0&1&1&1&1&0&0\\0&0&1&0&0&1&1&0&1&0&1&1&1&1&0\\0&0&0&1&0&0&1&1&0&1&0&1&1&1&1\end{pmatrix}$$

$$G_2=\begin{pmatrix}1&1&0&0&0&1&1&0&0&0&1&1&0&0&0\\0&1&1&0&0&0&1&1&0&0&0&1&1&0&0\\0&0&1&1&0&0&0&1&1&0&0&0&1&1&0\\0&0&0&1&1&0&0&0&1&1&0&0&0&1&1\end{pmatrix}$$

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

Chapitre Trois

CODES CYCLIQUES DIVISIBLES SUR UN CORPS DE GALOIS PREMIER

Dans ce chapitre, nous nous intéressons d'abord à la notion de divisibilité, ensuite nous énonçons le théorème de McEliece, l'utilisons conjointement avec le théorème de Harold N. Ward pour caractériser les codes cycliques divisibles sur un corps fini premier. Enfin nous étudions la divisibilité des codes cycliques binaires que nous avons construits.

3.1 Définition et généralités sur la divisibilité

Dans cette section, $C(n,k,d)$ est un code linéaire sur un corps de Galois $\mathbb{F}_q$.

Définition 3.1.1 : *Le code linéaire C est dit $l-$divisible, $l \in \mathbb{N}^* \setminus \{1\}$, si les poids de tous les mots de C sont divisibles par l.*
On dit que l est un diviseur du code C.

Définition 3.1.2 : *Le code linéaire C est dit divisible lorsqu'il existe un entier l non nul et différent de 1 tel que le code C soit $l-$divisible.*

Proposition 3.1.1 : *Si le code linéaire C est divisible, alors C est un code $\Delta-$divisible où*

$$\Delta = PPCM\{l, C \text{ est } l\text{-divisible}\}.$$

Preuve : Soit $c \in C$ un mot de C, alors pour tout entier l non nul et distinct de 1, tel que C soit $l-$divisible, l divise $\omega_H(c)$. Donc $\Delta = PPCM\{l, C \text{ est l-divisible}\}$ divise $\omega_H(c)$.
c étant arbitrairement choisi, C est $\Delta-$divisible. ■

Définition 3.1.3 : *Si le code linéaire C est divisible, alors on appelle degré de divisibilité du code linéaire C l'entier*

$$\Delta = PPCM\{l, C \text{ est l-divisible}\}.$$

Corollaire 3.1.1 : *Si le code linéaire C est $l-$divisible, $1 < l$, alors son degré de divisibilité est un multiple de l.*

Preuve : Cela provient du fait que le degré de divisibilité d'un code linéaire divisible est

$$\Delta = PPCM\{l, C \text{ est l-divisible}\}.$$ ■

Proposition 3.1.2 : *Soit $l \in \mathbb{N}^* \setminus \{1\}$.*
Si C est un code linéaire $l-$divisible, alors tout diviseur propre de l (diviseur de l qui est distinct de 1 et de l) dans $\mathbb{N}$ est un diviseur du code C.

Preuve : Soit t un diviseur propre de l, alors t est différent de 1.
Puisque t divise l et l divise les poids de tous les mots de C, alors par transitivité de la division dans $\mathbb{N}$, t divise les poids de tous les mots de C, donc t est un diviseur de C.

Proposition 3.1.3 : *Si C est un code divisible de degré de divisibilité Δ, alors*

$$\Delta = PGCD\{\omega_H(c), c \in C\}$$

Preuve : Posons $\mu = PGCD\{\omega_H(c), c \in C\}$.
Si C est $l-$divisible, alors les poids de tous les mots de C sont multiples de l. Donc l divise μ. l étant un diviseur quelconque de C, on déduit que Δ divise μ.
Réciproquement, montrons que μ divise Δ.
Il suffit de montrer que μ est un diviseur de C.
Par définition de μ, μ divise les poids de tous les mots de C, donc μ est un diviseur de C.
Ce qui prouve que $\Delta = \mu$. ■

Proposition 3.1.4 : *Soient C et C' deux codes linéaires équivalents. Si C est divisible, alors C' l'est aussi et les diviseurs de C' sont exactement ceux de C.*

Preuve : Cela découle de la proposition 2.1.2. ■

Proposition 3.1.5 : *Pour $q \in \{2;3\}$, tout code linéaire sur $\mathbb{F}_q$ faiblement auto-dual est $q-$divisible.*

Preuve : Remarquons que pour $q \in \{2;3\}$, $\mathbb{F}_q^*$ est un groupe multiplicatif d'ordre 1 ou 2. Donc pour tout $x \in \mathbb{F}_q^*$, $x^2 = 1$.
Soit C un code linéaire faiblement auto-dual, alors $C \subset C^{\perp}$. Donc pour tous mots c et c' de C, $c.c' = 0$. En particulier pour tout mot c de C, $c.c = 0$.
Mais

$$\begin{aligned}
c.c = 0 &\Leftrightarrow \sum_{i=0}^{n-1} c_i c_i = 0 \\
&\Leftrightarrow \sum_{i=0}^{n-1} c_i^2 = 0 \\
&\Leftrightarrow \sum_{i=0;c_i\neq 0}^{n-1} c_i^2 = 0 \\
&\Leftrightarrow \sum_{i=0;c_i\neq 0}^{n-1} 1 = 0 \\
&\Leftrightarrow card(\{i \in [|0;n-1|]/c_i \neq 0\}) = 0 \\
&\Leftrightarrow \omega_H(c) = 0 \\
&\Leftrightarrow \omega_H(c) \text{ est un multiple de } q
\end{aligned}$$

Ainsi, C est $q-$divisible. ■

Théorème 3.1.1 : *(H. N. Ward 1981, [5])*
Soit C un code linéaire de longueur n sur $\mathbb{F}_p$, $\Delta-$divisible, avec $\Delta \wedge p = 1$.
Alors C est équivalent à un code obtenu en prenant un code linéaire sur $\mathbb{F}_p$, en répétant chaque coordonnée Δ fois et en complétant avec des 0 pour obtenir la longueur n voulue.

Définition 3.1.4 : *Soient $m \in \mathbb{N}^* \setminus \{1\}$ et C un code linéaire de longueur n. C est dit m-dégénéré s'il est obtenu en répétant m fois un code de plus petite longueur et en complétant avec des 0 pour avoir la longueur n voulue.*

Définition 3.1.5 : *Un code linéaire C est dit dégénéré s'il existe $m \in \mathbb{N}^* \setminus \{1\}$ tel que C soit m-dégénéré.*

Définition 3.1.6 : *Un code linéaire C est dit e-dégénéré s'il est équivalent à un code dégénéré.*

Corollaire 3.1.2 : *Soit C un code linéaire sur $\mathbb{F}_p$ $\Delta-$divisible avec $\Delta = p^j\Delta'$, $p \wedge \Delta' = 1$, $\Delta' \neq 1$, $j \in \mathbb{N}$. Alors C est e-dégénéré.*

Preuve : Soit C un code linéaire sur $\mathbb{F}_p$ $\Delta-$divisible avec $\Delta = p^j\Delta'$, $p \wedge \Delta' = 1$, $\Delta' \neq 1$, $j \in \mathbb{N}$. Alors C est aussi Δ'-divisible car Δ' divise Δ. Puisque $p \wedge \Delta' = 1$, en appliquant le théorème 3.1.1, on conclut que C est e-dégénéré. ■

Corollaire 3.1.3 : *Si $C(n;k)$ est code Δ-divisible sur $\mathbb{F}_p$ avec $\Delta \wedge p$, alors $k \leq \lfloor \frac{n}{\Delta} \rfloor$.*

Preuve : Si $C(n;k)$ est code Δ-divisible sur $\mathbb{F}_p$ avec $\Delta \wedge p$, alors d'après le théorème 3.1.1 C est équivalent à un code obtenu en prenant un code linéaire C' de longueur n' sur $\mathbb{F}_p$, en répétant chaque coordonnée Δ fois et en complétant avec des 0 pour obtenir la longueur n voulue. On a donc $k = dimC = dimC'$. Lorsqu'on répète Δ fois le code C', on obtient un code de longueur $\Delta n'$. De ce fait, $\Delta n' \leq n$; c'est-à-dire $n' \leq \frac{n}{\Delta}$. Puisque $k \leq n'$, alors $k \leq \frac{n}{\Delta}$. D'où $k \leq \lfloor \frac{n}{\Delta} \rfloor$. ■

Divisibilité des codes de Griesmer

Théorème 3.1.2 : *(H. N. Ward 1998, [6])*
Soit $C(n;k;d)$ un code de Griesmer sur $\mathbb{F}_p$. Alors la plus grande puissance de p divisant d est la plus grande puissance de p divisant le code.

Critère de divisibilité des codes binaires

Proposition 3.1.6 : *Soient n un entier naturel non nul ; $a, b \in \mathbb{F}_2^n$, alors*

$$\omega_H(a+b) = \omega_H(a) + \omega_H(b) - 2\omega_H(a * b)$$

où $a * b \in \mathbb{F}_2^n$ et $a * b = (a_i b_i)_{0 \leq i \leq n-1}$.

Preuve : Soient n un entier naturel non nul ; $a, b \in \mathbb{F}_2^n$.
On sait que $\omega_H(a) = card(\{i \in [|0; n-1|]; a_i \neq 0\})$. Pour i parcourant $[|0; n-1|]$, on a :

$$\begin{aligned}
\{i; a_i + b_i \neq 0\} &= \{i; a_i + b_i = 1\} \\
&= \{i; a_i = 1 \text{ et } b_i = 0\} \cup \{i; a_i = 0 \text{ et } b_i = 1\} \\
&= (\{i; a_i = 1\} \setminus \{i; a_i = 1; b_i = 1\}) \cup (\{i; b_i = 1\} \setminus \{i; b_i = 1; a_i = 1\}) \\
&= (\{i; a_i = 1\} \setminus \{i; a_i b_i = 1\}) \cup (\{i; b_i = 1\} \setminus \{i; a_i b_i = 1\})
\end{aligned}$$

En passant aux cardinaux on obtient :

$$card(\{i; a_i + b_i \neq 0\}) = [card(\{i; a_i = 1\}) - card(\{i; a_i b_i = 1\})] + [card(\{i; b_i = 1\}) - card(\{i; a_i b_i = 1\})]$$

C'est-à-dire

$$\omega_H(a+b) = \omega_H(a) - \omega_H(a*b) + \omega_H(b) - \omega_H(a*b)$$

D'où

$$\omega_H(a+b) = \omega_H(a) + \omega_H(b) - 2\omega_H(a*b)$$

■

Proposition 3.1.7 : *Soient n et m deux entiers naturels non nuls tels que $2 \leq m$, $(a^i)_{1\leq i\leq m}$ une famille de m éléments de $\mathbb{F}_2^n$. Alors*

$$\omega_H(\sum_{i=0}^{m} a^i) = \sum_{\varnothing \neq S \subset [|1;m|]} (-2)^{|S|-1} \omega_H(*_{i\in S} a^i)$$

Preuve : Nous allons raisonner par récurrence sur m.

Le cas $m = 2$ est vrai car il correspond à la proposition 3.1.6.

Soit m un entier supérieur ou égal à 2. Supposons que pour toute famille $(a^i)_{1\leq i\leq m}$ de m éléments de $\in \mathbb{F}_2^n$,

$$\omega_H(\sum_{i=0}^{m} a^i) = \sum_{\varnothing \neq S \subset [|1;m|]} (-2)^{|S|-1} \omega_H(*_{i\in S} a^i)$$

Soit $(a^i)_{1\leq i\leq m+1}$ une famille de $m+1$ éléments de $\in \mathbb{F}_2^n$, alors $\sum_{i=0}^{m+1} a^i = \sum_{i=0}^{m} a^i + a^{m+1}$.

Donc

$$\begin{aligned}
\omega_H(\sum_{i=0}^{m+1} a^i) &= \omega_H(\sum_{i=0}^{m} a^i + a^{m+1}) \\
&= \omega_H(\sum_{i=0}^{m} a^i) + \omega_H(a^{m+1}) - 2\omega_H((\sum_{i=0}^{m} a^i) * a^{m+1}) \qquad \text{d'après la proposition 3.1.6} \\
&= \omega_H(\sum_{i=0}^{m} a^i) + \omega_H(a^{m+1}) - 2\omega_H(\sum_{i=0}^{m}(a^i * a^{m+1})) \qquad (\star)
\end{aligned}$$

En appliquant l'hypothèse de récurrence aux termes de la ligne $(\star)$ on obtient :

$$\begin{aligned}
\omega_H(\sum_{i=0}^{m} a^i) &= \sum_{\varnothing \neq S \subset [|1;m|]} (-2)^{|S|-1} \\
&= \sum_{\substack{S \subset [|1;m+1|], |S| \geq 1 \\ a^{m+1} \notin S}} (-2)^{|S|-1} \omega_H(*_{i\in S} a^i) \qquad (I)
\end{aligned}$$

et

$$\begin{aligned}\omega_H(a^{m+1}) - 2\omega_H(\sum_{i=0}^{m}(a^i * a^{m+1})) &= \omega_H(a^{m+1}) - 2\sum_{\varnothing \neq S \subset [|1;m|]} (-2)^{|S|-1}\omega_H(*_{i\in S}(a^i * a^{m+1})) \\ &= \sum_{\substack{S \subset [|1;m+1|] \\ |S|=1 \\ a^{m+1}\in S}} (-2)^{|S|-1}\omega_H(*_{i\in S}a^i) + \sum_{\substack{S \subset [|1;m+1|] \\ |S|>1 \\ a^{m+1}\in S}} (-2)^{|S|-1}\omega_H(*_{i\in S}a^i) \\ &= \sum_{\substack{S \subset [|1;m+1|], |S|\geq 1 \\ a^{m+1}\in S}} (-2)^{|S|-1}\omega_H(*_{i\in S}a^i) \qquad (II)\end{aligned}$$

En remplaçant les valeurs obtenues en I et en II dans $(\star)$, on obtient

$$\begin{aligned}\omega_H(\sum_{i=0}^{m+1} a^i) &= \sum_{\substack{S \subset [|1;m+1|], |S|\geq 1 \\ a^{m+1}\notin S}} (-2)^{|S|-1}\omega_H(*_{i\in S}a^i) + \sum_{\substack{S \subset [|1;m+1|], |S|\geq 1 \\ a^{m+1}\in S}} (-2)^{|S|-1}\omega_H(*_{i\in S}a^i) \\ &= \sum_{\varnothing \neq S \subset [|1;m+1|]} (-2)^{|S|-1}\omega_H(*_{i\in S}a^i)\end{aligned}$$

D'où le résultat. ■

Le théorème suivant a été énoncé par Ward dans [7]. Nous proposons une preuve de ce théorème.

Théorème 3.1.3 : *Soient C un code linéaire sur $\mathbb{F}_2$ de matrice génératrice G et e un entier naturel non nul. Alors 2^e est un diviseur du code C si et seulement si pour tout entier naturel non nul $m \leq e$, le mot obtenu en multipliant composante par composante m lignes de G (avec répétitions possibles) a un poids divisible par 2^{e-m+1}.*

Preuve : Soit C un code linéaire sur $\mathbb{F}_2$ de matrice génératrice G et e un entier naturel non nul.

Supposons que 2^e est un diviseur de C. Soient $m \leq e$ un entier naturel non nul, L_1, L_2, $\cdots$, L_m m lignes de G (avec répétitions possibles) et $L^m = L_1 * L_2 * \cdots * L_m$.

Montrons par induction sur m que L^m a un poids divisible par 2^{e-m+1}

Si $m = 1$, alors L^m est une ligne de G. Puisque C est 2^e-divisible, alors alors L a un poids multiple de $2^e = 2^{e-1+1} = 2^{e-m+1}$.

Supposons maintenant que $m < e$ et que pour tout $k \in [|1;m|]$, L^k a un poids divisible par 2^{e-k+1} et montrons que le poids de L^{m+1} est divisible par 2^{e-m}.

D'après la proposition 3.1.7, on a :

$$\begin{aligned}\omega_H(\sum_{i=1}^{m+1} L_i) &= \sum_{\varnothing\neq S\subset[|1;m+1|]} (-2)^{|S|-1}\omega_H(*_{i\in S}L_i)\\ &= \sum_{\varnothing\neq S\subset[|1;m+1|],|S|<m+1} (-2)^{|S|-1}\omega_H(*_{i\in S}L_i) + (-2)^m\omega_H(*_{i\in[|1;m+1|]}L_i)\\ &= \sum_{\varnothing\neq S\subset[|1;m+1|],|S|<m+1} (-2)^{|S|-1}\omega_H(L^{|S|}) + (-2)^m\omega_H(L^{m+1})\end{aligned}$$

Donc

$$(-2)^m\omega_H(L^{m+1}) = \omega_H(\sum_{i=1}^{m+1} L_i) - \sum_{\varnothing\neq S\subset[|1;m+1|],|S|<m+1} (-2)^{|S|-1}\omega_H(L^{|S|}) \qquad (\star)$$

Le mot $\sum_{i=1}^{m+1} L_i$ est un mot du code, donc son poids est divisible par 2^e.
Pour toute partie non vide S de $[|1;m+1|]$ telle que $|S| < m+1$, l'hypothèse d'induction nous rassure que $\omega_H(L^{|S|})$ est divisible par $2^{e-|S|+1}$, donc $(-2)^{|S|-1}\omega_H(L^{|S|})$ est divisible par $2^{|S|-1}\times 2^{e-|S|+1} = 2^e$. D'où le terme

$$\sum_{\varnothing\neq S\subset[|1;m+1|],|S|<m+1} (-2)^{|S|-1}\omega_H(L^{|S|})$$

est un multiple de 2^e.
En revenant à l'équation $\star$ on déduit que 2^e divise $(-2)^m\omega_H(L^{m+1})$; donc 2^{e-m} divise $\omega_H(L^{m+1})$.

Réciproquement, supposons que pour tout entier naturel non nul $m \le e$, L^m a un poids divisible par 2^{e-m+1} et montrons que 2^e est un diviseur de C.
Soit c un mot de C. Puisque G est une matrice génératrice de C, alors c est une combinaison linéaire sur $\mathbb{F}_2$ des t lignes de G avec $1 \le t$.
Posons $c = \sum_{i=1}^{t} L_i$. Alors d'après la proposition 3.1.7, on a :

$$\begin{aligned}\omega_H(c) &= \omega_H(\sum_{i=1}^{t} L_i)\\ &= \sum_{\varnothing\neq S\subset[|1;t|]} (-2)^{|S|-1}\omega_H(L^{|S|})\\ &= \sum_{\varnothing\neq S\subset[|1;t|],|S|\le m} (-2)^{|S|-1}\omega_H(L^{|S|}) + \sum_{\varnothing\neq S\subset[|1;t|],m<|S|} (-2)^{|S|-1}\omega_H(L^{|S|})\\ &= \sum_{\varnothing\neq S\subset[|1;t|],|S|\le e} (-2)^{|S|-1}\omega_H(L^{|S|}) + (-2)^e \sum_{\varnothing\neq S\subset[|1;t|],e+1\le|S|} (-2)^{|S|-1-e}\omega_H(L^{|S|}) \qquad (\star\star)\end{aligned}$$

Pour toute partie S de $[|1;t|]$ de cardinal inférieur ou égal à e, l'hypothèse nous rassure que $\omega_H(L^{|S|})$ est divisible par $2^{e-|S|+1}$; donc $(-2)^{|S|-1}\omega_H(L^{|S|})$ est divisible par $2^{|S|-1}\times 2^{e-|S|+1} = 2^e$.

D'où le terme

$$\sum_{\varnothing \neq S \subset [|1;t|], |S| \leq e} (-2)^{|S|-1} \omega_H(L^{|S|})$$

est un multiple de 2^e.

Ainsi, en revenant à l'égalité $(\star\star)$, on déduit que $\omega_H(c)$ est un multiple de 2^e.

D'où 2^e est un diviseur du code. ■

3.2 Divisibilité des codes cycliques sur un corps de Galois premier

Il s'agit ici de donner le théorème de McEliece sur la divisibilité des codes cycliques sur $\mathbb{F}_p$ et de l'utiliser pour démontrer les résultats cités dans le résumé.

Dans toute la suite, on se place dans le cas semi-simple, c'est-à-dire $pgcd(n, p) = 1$.

Théorème 3.2.1 : *(McEliece 1972)*

Soit C un code cyclique de longueur n sur $\mathbb{F}_p$. Alors la plus grande puissance de p divisant C est p^e où $m = (p-1)(e+1)$ est le plus petit multiple de $p-1$ tel que le produit de m non-zéros de C (avec répétitions possibles) soit 1.

Remarque 3.2.1 : En prenant $p = 2$ dans le théorème précédent, on obtient $p - 1 = 1$ et $m = e + 1$. Dans le cas binaire on a donc une version plus simplifiée.

Corollaire 3.2.1 : *Soit C un code cyclique de longueur n (impair) sur $\mathbb{F}_2$. Alors la plus grande puissance de 2 divisant C est 2^e où $e+1$ est le plus petit entier tel que le produit de $e+1$ non-zéros de C (avec répétitions possibles) soit 1.*

Proposition 3.2.1 : *Soit C un code cyclique de longueur n sur $\mathbb{F}_p$ d'ensemble de parité U. Alors $\underbrace{(1;1;\cdots;1)}_{n \text{ fois}}$ est un mot de C si et seulement si $0 \in U$.*

Preuve : Soit C un code cyclique de longueur n sur $\mathbb{F}_p$ d'ensemble de parité U (avec $n \wedge p = 1$) et soit $g(x)$ son polynôme générateur.

Si $(1;1;\cdots;1)$ est un mot de C, alors sa représentation polynomiale $X^{n-1} + X^{n-2} + \cdots + 1$ est un multiple de $g(X)$. Puisque $X^n - 1 = (X-1)(X^{n-1} + X^{n-2} + \cdots + 1)$ et que $X^n - 1$ n'a pas de racine multiple, alors 1 n'est pas une racine de $X^{n-1} + X^{n-2} + \cdots + 1$; d'où 1 n'est pas

une racine de $g(x)$ car $X^{n-1} + X^{n-2} + \cdots + 1$ est un multiple de $g(X)$. Notons β une racine $n^{ième}$ primitive de l'unité. Alors $1 = \beta^0$ n'est pas une racine de $g(x)$ d'où $0 \notin T$(ensemble de définition de C) ; donc $0 \in U$.

Réciproquement, si $0 \in U$, alors 1 n'est pas une racine de $g(x)$, donc $x - 1$ et $g(x)$ sont premiers entre eux. Puisque $g(X)$ divise $X^n - 1 = (X - 1)(X^{n-1} + X^{n-2} + \cdots + 1)$, on en déduit que $g(X)$ divise $X^{n-1} + X^{n-2} + \cdots + 1$. Donc $(1; 1; \cdots; 1)$est un mot de C. ■

Proposition 3.2.2 : *Soit C un code cyclique non e-dégénéré de longueur n sur $\mathbb{F}_p$ et d'ensemble de parité U. Si C est divisible, alors $0 \notin U$.*
Autrement dit, si $0 \in U$, alors C n'est pas divisible.

Preuve : Soit C un code cyclique de longueur n sur $\mathbb{F}_p$ d'ensemble de parité U tel que C soit non e-dégénéré.

Procédons par l'absurde : Supposons que C est divisible et que $0 \in U$.

C étant non e-dégénéré, d'après le théorème 3.1.2, tout diviseur de C est une puissance de p. Soit p^e $(e \neq 0)$ un diviseur de C. Comme $0 \in U$, d'après la proposition 3.2.1, $(1; 1; \cdots; 1)$ est un mot de C. Donc $\omega_H((1; 1; \cdots; 1)) = n$ est un multiple de p^e, c'est-à-dire que n est un multiple de p. Cela contredit le fait que $n \wedge p = 1$. ■

Théorème 3.2.2 : ***(Divisibilité du dual d'un code cyclique)***
Soit C un code cyclique divisible non e-dégénéré. Alors son dual est divisible si et seulement s'il est e-dégénéré.

Preuve : Soit C un code cyclique divisible non e-dégénéré d'ensemble de parité U.

Si son dual est e-dégénéré, alors il est divisible par définition.

Réciproquement, supposons que $C^\perp$ soit divisible.

Si $C^\perp$ est non e-dégénéré, alors d'après la proposition 3.2.2, 0 n'est pas un élément de l'ensemble de parité de $C^\perp$, donc 0 appartient à l'ensemble de définition de $C^\perp$ (attention : cela n'est vrai que pour 0, en général le complémentaire de l'ensemble de définition n'est pas l'ensemble de parité). D'après la proposition 2.2.10, l'ensemble de définition de $C^\perp$ est $T^\perp = \{t \mid \exists s \in U, -s \equiv t \ mod \ n\}$, d'où $0 \in U$. Ce qui contredit la proposition 3.2.2 car C est cyclique non e-dégénéré et divisible.

Donc $C^\perp$ est e-dégénéré. ■

Théorème 3.2.3 : *(Caractérisation des codes cycliques binaires divisibles)*
Soit C un code cyclique binaire non e-dégénéré d'ensemble de parité U. Alors C est divisible si et seulement si $0 \notin U$.

Preuve : Soit C un code cyclique binaire non e-dégénéré d'ensemble de parité U.
Si C est divisible, alors d'après la proposition 3.2.2, $0 \notin U$.
Réciproquement, supposons que $0 \notin U$, alors 1 n'est pas un non-zéro de C, donc le plus petit entier m tel que le produit de m non-zéros de C (avec répétitions possibles) soit 1 est supérieur ou égal à 2. D'après le corollaire 3.2.1, la plus grande puissance de 2 divisant C est 2^e où $m = e + 1$. Comme $m \geq 2$, alors $e \geq 1$. Donc 2 est un diviseur de C ; d'où C est divisible. ■

Les propositions suivantes montrent que cette caractérisation n'est valide que pour les codes binaires.

Proposition 3.2.3 : *Si $p \neq 2$, tout code cyclique non e-dégénéré sur $\mathbb{F}_p$ de longueur paire n et d'ensemble de parité U tel que $\frac{n}{2} \in U$ est non divisible.*

Preuve : Soit p un nombre premier distinct de 2. Soit C un code cyclique non e-dégénéré sur $\mathbb{F}_p$ de longueur paire n et d'ensemble de parité U tel que $\frac{n}{2} \in U$. Pour montrer que C est non divisible, il suffit de montrer que la plus grande puissance de p divisant C est 1 car C est non e-dégénéré.
Soit β une racine $n^{ième}$ primitive de l'unité, alors $\theta = \beta^{\frac{n}{2}}$ est un non-zéro de C. $\theta^2 = (\beta^{\frac{n}{2}})^2 = \beta^n = 1$.
Puisque p est un nombre premier distinct de 2, alors p est impair, donc $p-1$ est pair. D'où $\theta^{p-1} = (\theta^2)^{\frac{p-1}{2}} = 1^{\frac{p-1}{2}} = 1$. Donc le produit de $p-1$ non-zéros de C (avec répétitions possibles) est égal à 1. En revenant aux notations du théorème de McEliece, on a
$p - 1 = \omega = (p-1)(e+1)$. Donc $e = 0$, d'après le théorème de McEliece, la plus grande puissance de p divisant C est 1.
D'où C est non divisible. ■

Proposition 3.2.4 : *Si $C(n,k)$ est un code linéaire m-e-dégénéré, alors $k \leq \lfloor \frac{n}{m} \rfloor$.*

Preuve : Identique à celle du corollaire 3.1.3. ■

Théorème 3.2.4 : *Si $p \neq 2$, alors il existe au moins un code cyclique C sur $\mathbb{F}_p$ d'ensemble de parité U tel que $0 \notin U$ et C est non divisible.*

Preuve : Soit p un nombre premier distinct de 2. Alors p est impair, donc $n = p^2 - 1$ est pair. Soit β une racine $n^{ième}$ primitive de l'unité et soit $m(X)$ son polynôme minimal. Posons $g(X) = (X-1)m(X)$, c'est un polynôme de degré 3.
Soit C le code cyclique de longueur n sur $\mathbb{F}_p$ de polynôme générateur $g(x)$ et soit U son ensemble de parité. Alors la dimension de C est $k = n - 3$ et $0 \notin U$.

C est un code non e-dégénéré :
En effet, si C est m-e-dégénéré avec $n \neq m$ (ce cas n'est pas intéressant), alors d'après la proposition 3.2.4,$k \leq \lfloor \frac{n}{m} \rfloor$, d'où $k \leq \frac{n}{m}$ (car $\lfloor \frac{n}{m} \rfloor \leq \frac{n}{m}$) ; c'est-à-dire $mk \leq n$. Puisque $m \in \mathbb{N}^* \setminus \{1\}$, alors $2 \leq m$, d'où $2k \leq n$. Comme $k = n - 3$, alors $2(n-3) \leq n$, donc $n \leq 6$. En remplaçant n par sa valeur, on obtient $p^2 - 1 \leq 6$, c'est-à-dire $p \leq \sqrt{7}$, ce qui contredit le fait que p est un nombre premier distinct de 2.

L'ensemble de définition de C est $\{0; 1; p\}$, donc $\frac{n}{2} \in U$. En appliquant la proposition 3.2.3, on conclut que C est non divisible. ■

Théorème 3.2.5 : ***(Caractérisation des codes cycliques divisibles sur un corps de Galois premier)***
Soit C un code cyclique non e-dégénéré sur un corps de Galois premier $\mathbb{F}_p$; alors C est divisible si et seulement si le produit de $p-1$ non-zéros de C (avec répétitions possibles) est toujours distinct de 1.

Preuve : Supposons que C soit divisible. Comme C est non e-dégénéré, alors il existe un entier naturel non nul e tel que p^e soit la plus grande puissance de p divisant C. D'après le théorème McEliece, $m = (p-1)(e+1)$ est le plus petit multiple de $p-1$ tel que le produit de m non-zéros de C (avec répétitions possibles) soit 1. Puisque $p - 1 < m$, alors le produit de $p-1$ non-zéros de C (avec répétitions possibles) est distinct de 1.

Réciproquement, supposons que le produit de $p-1$ non-zéros de C (avec répétitions possibles) est toujours distinct de 1.
Soit $m = (p-1)(e+1)$ le plus petit multiple de $p-1$ tel que alors le produit de $p-1$ non-zéros de C (avec répétitions possibles) soit 1, alors $1 \leq e$ sinon $e = 0$ et $m = p - 1$, ce qui est absurde. D'après le théorème de McEliece, p^e est un diviseur de C ; donc C est divisible. ■

3.3 Divisibilité des codes C_1, C_2 et C_3

Nous allons étudier la divisibilité de ces codes en utilisant tour à tour les critères et théorèmes abordés précédemment.

3.3.1 Cas du code cycliques C_1

Le polynôme générateur de C_1 est

$$g_1(x) = (x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+x^2+x+1)$$

Comme $g_1(x)$ est un polynôme de degré 11, alors C_1 est un $(15;4;d)-$code cyclique de matrice génératrice

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$\star$) C_1 est un code de Griesmer

En effet, son ensemble de définition est $T_1 = \{0;1;2;3;4;5;6;8;9;10;12\}$, qui contient la suite $0;1;2;3;4;5;6$ constituée de 7 entiers consécutifs. D'après le théorème de la borne BCH, $7+1 \leq d$, donc $8 \leq d$. Mais les mots de la matrice génératrice G_1 ont pour poids 8 ; donc $d = 8$.

Ainsi,

$$\sum_{i=0}^{k-1} \lceil \frac{d}{2^i} \rceil = \sum_{i=0}^{4-1} \lceil \frac{8}{2^i} \rceil = \lceil \frac{8}{1} \rceil + \lceil \frac{8}{2} \rceil + \lceil \frac{8}{4} \rceil + \lceil \frac{8}{8} \rceil = 8+4+2+1 = 15$$

D'où C_1 est un code de Griesmer.

D'après le théorème 3.1.2, la plus grande puissance de 2 divisant C_1 est celle qui divise $d = 8$. D'où C_1 est 8−divisible.

$\star\star$) Appliquons le théorème de McEliece

L'ensemble de Parité de C_1 est $U_1 = \{7;11;13;14\}$ et les non-zéros de C_1 sont : $\alpha^7;\alpha^{11};\alpha^{13}$ et α^{14}. Le fait que $0 \notin U_1$ montre déjà que le code C_1 est divisible. De ce fait

on calcule le produit de k non-zéros de C_1 (avec répétitions possibles), k allant de 2 au cardinal de U_1. Pour que le produit de k non-zéros de C_1 (avec répétitions possibles) soit égal à 1, il faut et il suffit que la somme de leurs exposants (un exposant est répété autant de fois que le non-zéro auquel il correspond) soit un multiple de 15.

Après avoir effectué les calculs, on constate que pour $k \leq 3$ le produit de k non-zéros de C_1 est distinct de 1. Et, on a :

$$\alpha^7\alpha^{11}\alpha^{13}\alpha^{14} = \alpha^{7+11+13+14} = \alpha^{45} = 1$$

Donc le plus petit entier m tel que le produit de m non-zéros de C_1 (avec répétitions possibles) soit égal à 1 est $m = 4$. Puisque $4 = 3 + 1$, d'après le corollaire 3.2.1, la plus grande puissance de 2 divisant C_1 est $2^3 = 8$. D'où C_1 est 8−divisible.
Nous pouvons aussi remarquer qu'aucun nombre, premier avec deux, n'est diviseur du code C_1, car sinon il diviserait la distance minimale du code qui est 8 ce qui est absurde.

Conclusion : C_1 est un code cyclique divisible non e-dégénéré de degré de divisibilité $8 = 2^3$.

3.3.2 Cas du code cycliques C_2

Le polynôme générateur de C_2 est

$$g_2(x) = (x-1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)$$

Comme $g_2(x)$ est un polynôme de degré 11, alors C_2 est un $(15; 4; d)$−code cyclique.

Une matrice génératrice de C_2 est :

$$G_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

On constate directement que $d \leq 6$ et on a donc :

$$\sum_{i=0}^{k-1} \lceil \frac{d}{2^i} \rceil \leq \sum_{i=0}^{4-1} \lceil \frac{6}{2^i} \rceil = \lceil \frac{6}{1} \rceil + \lceil \frac{6}{2} \rceil + \lceil \frac{6}{4} \rceil + \lceil \frac{6}{8} \rceil = 6 + 3 + 2 + 1 = 12 < 15$$

D'où C_2 n'est pas un code de Griesmer.

La matrice génératrice de C_2 est constituée de trois blocs identiques ; donc C_2 est un code dégénéré ; il s'obtient en répétant 3 fois chaque mot du code cyclique C_2^* de longueur 5, de polynôme générateur $g_2^*(x) = x + 1$, de dimension 4 et de matrice génératrice

$$G_2^* = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Donc 3 est un diviseur du code C_2.

Intéressons nous au code C_2^* :

$\star$) C_2^* est un code de Griesmer

En effet, tout code cyclique non trivial (distinct du code nul et de $\mathbb{F}q^n$) a une distance minimale supérieure ou égale à 2 (c'est une conséquence du théorème de la borne BCH). Comme C_2^* a des mots de poids 2, alors sa distance minimale est 2. On a donc

$$\sum_{i=0}^{k-1} \lceil \frac{d}{2^i} \rceil = \sum_{i=0}^{4-1} \lceil \frac{2}{2^i} \rceil = \lceil \frac{2}{1} \rceil + \lceil \frac{2}{2} \rceil + \lceil \frac{2}{4} \rceil + \lceil \frac{2}{8} \rceil = 2 + 1 + 1 + 1 = 5$$

D'où C_2^* est un code de Griesmer.

D'après le théorème 3.1.2, la plus grande puissance de 2 divisant C_2^* est celle qui divise 2. D'où C_2^* est 2−divisible.

$\star\star$) Appliquons le théorème de McEliece

Soit θ une racine $5^{ième}$ de l'unité distincte de 1. Le polynôme générateur de C_2^* est $g_2^*(x) = x + 1$ et a pour racine $1 = \theta^0$, donc l'ensemble de parité de C_2^* est $U_2^* = \{1; 2; 3; 4\}$ et les non-zéros de C_2^* sont θ, θ^2, θ^3 et θ^4. Comme $0 \notin U_2^*$, alors U_2^* est divisible. On remarque immédiatement que $\theta^2\theta^3 = \theta^5 = 1$ ou que $\theta\theta^4 = \theta^5 = 1$, donc 2 est le plus petit des entiers m tels que le produit de m non-zéros de C_2^* (avec répétitions possibles) soit égal à 1. Puisque $2 = 1 + 1$, d'après le corollaire 3.2.1, la plus grande puissance de 2 divisant C_2^* est $2^1 = 2$. D'où C_2^* est 2−divisible.

Ainsi, C_2 est une 3−réplique du code C_2^* qui est 2−divisible ; donc 2 et 3 sont des diviseurs de C_2. Comme $2 \wedge 3 = 1$, alors d'après la proposition 3.1.1, $6 = PPCM(2; 3)$ est

aussi un diviseur du code C_2.

Conclusion : C_2 est un code cyclique divisible dégénéré de degré de divisibilité $6 = 2^1 \times 3$.

3.3.3 Cas du code cyclique C_3

Le polynôme générateur de C_3 est $g_3(X) = (x^2+x+1)(x^4+x^3+x^2+x+1)(x^4+x^3+1)$
Comme $g_3(x)$ est un polynôme de degré 10, alors C_3 est un $(15;5;d)-$code cyclique.

Une matrice génératrice de C_3 est :

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

Le domaine de définition de C_3 est $T_3 = \{3;5;6;9;10;11;12;13;14\}$, qui contient la suite $9;10;11;12;13;14$ constituée de 6 entiers consécutifs. D'après le théorème de la borne BCH, $6+1 \leq d$, donc $7 \leq d$. Mais les mots de la matrice génératrice ont pour poids 7 ; donc $d = 7$.

Le code C_3 est non e-dégénéré

En effet, si C_3 était m-e-dégénéré ($m \in \mathbb{N}^* \setminus \{1\}$), alors m serait un diviseur de 7. Donc on aurait $m = 7$, car si $m = 1$, C_3 serait non e-dégénéré. Dans ce cas C_3 serait un code $7-$divisible ; cela est impossible car la somme des deux premières lignes de la matrice génératrice de C_3 donne le mot $(1;1;1;1;0;1;0;1;1;0;0;1;0;0;0)$ qui a pour poids 8 et 7 ne divise pas 8.

Aucune puissance de 2 (distincte de 1) ne divise le code C_3

Puisque 2 ne divise pas $d = 7$, alors 2 ne divise pas le code C_3 car tout diviseur d'un code divise sa distance minimale. Donc aucune puissance de 2 distincte de 1 ne divise C_3.
On peut aussi partir du fait que $0 \in U_3 = \{0;1;2;4;8\}$ et utiliser le corollaire 3.2.1 pour montrer que la plus grande puissance de 2 divisant C_3 est 1.

Conclusion : C_3 est un code cyclique non divisible.

♠ Conclusion ♠

Dans notre travail, nous avons d'abord examiné la divisibilité des codes cycliques sur un corps de Galois premier, nous avons ensuite étudié la divisibilité des trois codes cycliques binaires de longueur 15 et nous avons enfin conçu un programme matlab qui calcule la distribution de poids d'un code cyclique binaire et nous dit si ce code est divisible ou pas, s'il est e-dégénéré ou pas. Il en ressort qu'un code cyclique binaire C non e-dégénéré est divisible si et seulement si 0 est un élément de son ensemble de définition. Dans le cas général où p est un nombre premier quelconque, nous avons prouvé qu'un code cyclique non e-dégénéré sur $\mathbb{F}_p$ est divisible si et seulement si le produit de $p-1$ non-zéros de ce code (avec répétitions possibles) est toujours distinct de 1.

Nous allons dans les travaux futurs nous intéresser aux codes cycliques divisibles sur des corps de Galois (non premier). Et, nous allons aussi regarder les avantages de la divisibilité sur le décodage.

♠ Annexe ♠

Description du programme

Notre programme matlab prend en entrée le polynôme générateur d'un code cyclique binaire et la longueur de ce code. Il retourne la dimension du code, sa distance minimale, sa capacité correctrice, une de ses matrices génératrices, son polynôme énumérateur de poids, le degré de divisibilité du code. Il nous dit aussi si le code est divisible ou pas, s'il est e-dégénéré ou pas.

Il obtient la dimension en retranchant le degré du polynôme générateur de la longueur du code. Il obtient une matrice génératrice en shiftant le polynôme générateur. Il calcule tous les mots du code de façon exhaustive et récupère leurs poids à l'aide d'un vecteur (distribution de poids du code). Il utilise ce vecteur pour déterminer la distance minimale, la capacité correctrice, le pgcd des poids des mots du code (que nous notons d).
Si $d = 1$, alors notre code est non divisible.
Sinon, notre code est divisible. Si d est une puissance de 2, il conclut que le code est non e-dégénéré. Si d n'est pas une puissance de 2, alors le code est e-dégénéré.

Présentation du code source

1) MACRO.m

```
disp('   ***  Mémoire de Master  2016 de FOUOTSA TAKO Boris  ***' );
disp(' ');
disp('       ********  BIENVENU  DANS  NOTRE PROGRAMME   *******');
disp(' ');
disp(' ');
disp(' Il prend en entrée la longueur et le polynôme générateur d'' un code
cyclique ');
disp(' binaire C et permet de: ');
disp(' ');
disp(' --  déterminer  ses paramètres (longueur, dimension,  distance minimale');
disp('       et capacité correctrice) et une de ses matrices génératrices;');
disp(' ');
disp(' --  déterminer sa distribution de poids et dire si''il est dégénéré ou
pas,');
disp('     s''il est divisible ou pas.');

tout_a_propos_du_code_avec_contraintes;
```

2) Des sous-programmes utiles

a) dimension_du_code.m

Prend en entrée la longueur et le polynôme générateur et retourne la dimension du code.

```
function [k]= dimension_du_code(g,n)
          k=n-length(g)+1;
return
```

b) Matrice_génératrice_du_code.m

Prend en entrée la longueur et le polynôme générateur et retourne une matrice génératrice du code.

```
function [G] = matrice_generatrice_du_code(g,n)
         k = dimension_du_code(g,n);
         G = zeros(k,n);
         l=length(g);
         for i = 1:k
             G(i,i:i+l-1) = g;
         end
return
```

c) le_mot_est_il_binaire.m

Pend en entrée un vecteur et retourne 1 si les composantes de ce vecteur sont binaires et 0 sinon.

```
function [y] = le_mot_est_il_binaire(g)
      n=length(g);
      i=1;
      while ( (i<n+1) && ((g(i)==0) || (g(i)==1))    )
          i=i+1;
      end
      if i-1==n
         y=true;
      else
          y=false;
      end
end
```

d) test_de_division.m

Prend en entrée un polynôme et un entier n, retourne 1 si le polynôme divise $X^n - 1$ et 0 sinon.

```
function [test] = test_de_division(g,n)
   l=length(g);
    a=zeros(1,n);
    for i=1:l    % pour utiliser un vecteur comme polynôme, la première
      a(i)=g(l-i+1); % composante doit être le coefficient du monôme de plus
    end        % haut  degré, de ce fait nous inversons l'ordre des coefficients du
                    % polynôme générateurs ;
    p=zeros(1,n+1);
    p(n+1)=1;
    p(1)=1;
    [q,r]= gfdeconv(p,a);
    if r==0
      test = true;
    else
      test = false;
    end
end
```

e) countvect.m

Prend en entrée un entier n et un vecteur x dont les composantes sont comprises entre 0 et n, retourne le nombre de fois que chaque composante apparait sous forme de vecteur.

```
 function [t] = countvect(x,n)
     m=length(x);
     t=zeros(1,n+1);
     for i= 1:n+1
          for j=1:m
              if x(j)==i-1
                 t(i)=t(i)+1;
              else
              end
          end
     end
end
```

f) pgcd_d_une_suite_d_entiers.m

Prend en entrée une suite finie d'entiers et retourne leur pgcd.

```
function [u] = pgcd_d_une_suite_d_entiers(x)
   n=length(x);
   u=x(1);
    for i = 1:n-1
        u=gcd(u,x(i+1));
     end
return
```

g) puissance_de_deux_qui_divise.m

Prend en entrée un entier et retourne la plus grande puissance de 2 qui le divise.

```
function [m] = puissance_de_deux_qui_divise(n)
       m=1;
        while floor( n/(2*m))== n/(2*m)
        m= 2*m;
        end
return
```

h) poids_d_un_mot_binaire.m

Prend en entrée un mot binaire et retourne son poids de Hamming.

```
function [w] = poids_d_un_mot_binaire(x)
    n= length(x);
    w=0;
    for i = 1:n
         w=w+x(i);
     end
return
```

i) vecteur_modulo_2.m

Prend en entrée un vecteur et retourne un vecteur dont les composantes sont les restes de composantes du premier vecteur dans la division par 2.

```
 function [y] = vecteur_modulo_2(x)
     n = length(x);
     y = 1:n;
     for i=1:n
         y(i) = rem(x(i),2);
     end
 return
```

j) distribution_sans_zeros

Prend en entrée le vecteur de distribution de poids et la longueur du code et affiche le polynôme énumérateur de poids.

```
function  distribution_sans_zeros(x,n)
       fprintf('1 ');
       for i=2:n+1
           if x(i)==0
           elseif x(i)==1
                  fprintf('+ x^%d ',i-1);
           else
                   fprintf('+ %dx^%d ',x(i),i-1);
           end
       end
      disp(' ')
end
```

k) generateur_espace_vectoriel_binaire

Prend en entrée un entier naturel non nul k et génère une matrice F $2^k \times k$ contenant tous les vecteurs de l'espace vectoriel $\mathbb{F}_2^k$.

```
function  [F]=generateur_espace_vectoriel_binaire(k)
         Mat=zeros(2^k,k);
           Nb_ligne=2^k;

           j=1; posi=1;  val=0;
           ligne=1;
           while(j<=k)

               puis=(2^k)/(2^j);
               amp=puis;
                while(ligne<=Nb_ligne)
```

```
                    for i=posi:puis

                        Mat(i,j)=val;
                    end
                    posi=puis+1;
                    puis=puis+amp;
                    ligne=posi;

                    if(val==0)
                        val=1;
                    else
                        val=0;
                    end

                end
                ligne=1; posi=1; j=j+1;
                val=0;
            end
           F=Mat;
end
```

3) tout_a_propos_du_code.m

Prend en entrée la longueur du code et son polynôme générateur, retourne sa dimension, distance minimale, sa capacité correctrice, sa matrice génératrice, son degré de divisibilité et sa distribution de poids.

Nous ne présentons ici que les calculs pour le cas où la dimension du code est 4.

```
function  [n, k, d, e, G, t, dist]=tout_a_propos_du_code(g,n)

k = dimension_du_code(g,n);         % nous récupérons la dimension du code à partir
de son polynome générateur

G = matrice_generatrice_du_code(g,n); % G est la matrice génératrice du code

  F=generateur_espace_vectoriel_binaire(k);
  C=zeros(2^k-1,n);
  d=1:2^k-1;

   for i=1:2^k-1     %  nous utilisons l'espace vectoriel et la matrice
      C(i,:)=F(i+1,:) *G; génératrice G pour calculer tous les mots non
      C(i,:)=vecteur_modulo_2(C(i,:));  %  nuls du code.
      d(i) =  poids_d_un_mot_binaire(C(i,:)); %  calcul des poids des mots du code.
   end

     dist=countvect(d,n); % calcul de la distribution de poids
     t = pgcd_d_une_suite_d_entiers(d);   % calcul du degré de divisibilité du code
       d = min(d);       % calcul de la distance minimale du code
       e= floor((d-1)/2); % calcul de la capacité correctrice du code
end
```

4) tout_a_propos_du_code_avec_contraintes.m

Utilise les contraintes pour contrôler le polynôme générateur entré par l'utilisateur et calcule tous les paramètres du code, une matrice génératrice, la distribution de poids du code en utilisant tout_a_propos_du_code.m

```
n=input(' Entrez la longueur de votre code: \n');
while n <1
    disp(' ');
    n=input(' La longueur de votre code doit être un entier naturel non nul. \n
        Entrez à nouveau la longueur de votre code: \n');
end
  disp(' ');
  g=input(' Entrez votre polynôme générateur à coefficients binaires entre
          crochets:  \n  ');
 disp(' ');
 while ( any(g) ==0 || g(1)==0 || g(length(g))==0 || le_mot_est_il_binaire(g)==0
        || test_de_division(g,n) ==0 )
     if (any(g) ==0)
         disp(' ');
         g=input(' Votre polynôme est nul, entrez un polynôme non nul: \n');

     elseif (g(1)==0 || g(length(g))==0)
         disp(' ');
        disp(' Il est impossible que le terme constant ou le coefficient dominant
              de votre polynôme soit nul');
         g=input(' Entrez un polynôme dont le terme constant et le coefficient
                   dominant sont non nuls: \n');

     elseif (le_mot_est_il_binaire(g)==0)
         disp(' ');
          disp(' Votre polynôme n''est pas à coefficients binaires.');
          g=input(' Entrez un polynôme à coefficients binaires: \n');

     elseif (test_de_division(g,n) ==0)
        fprintf(' Votre polynôme ne divise pas X^%d -1, il n''est donc pas le
                générateur d''un code cyclique de longueur %d. \n', n,n);
        fprintf(' Veuillez entrer le polynôme générateur d''un code cyclique
             binaire de longueur %d. \n', n);
        g=input(' ');
        fprintf(' \n');
    end
 end
  disp(' ');
  disp(' Votre polynôme est correct !');
  disp(' ');

  [n, k, d, e, G, t, dist]=tout_a_propos_du_code(g,n);

  choix_des_informations;
```

5) choix_des_informations.m

Permet de choisir les informations que nous recherchons.

```
disp(' ');
disp(' ');
disp(' Tapez : 1 si vous voulez les paramètres
      et une matrice génératrice du code;');
disp('         2 si vous voulez les informations sur la divisibilité et la
           dégénération du code;');
disp('         3 si vous voulez toutes les informations qui précèdent;');
disp('         4 si vous voulez entrer un nouveau code;');
disp('         0 si vous voulez sortir.');
disp(' ');
 x=input(' Votre choix:  ');
disp(' ');
  while (x~=1 && x~=2 && x~=3 && x~=4 && x~=0)
         x=input('Choix incorrect, rechoisissez:  ');
         disp(' ');
  end
   if (x==1)
        fprintf(' La longueur de ce code est: %d; \n',n);
        fprintf(' Sa dimension est: %d; \n',k);
        fprintf(' Sa distance minimale est: %d; \n',d);
        fprintf(' Sa capacité correctrice est : %d. \n',e);
        disp(' ');
        disp(' Une matrice génératrice de votre code est :'); disp(G);
        disp(' ');
        choix_des_informations;

   elseif  (x==2)

       disp(' ');
       fprintf(' Le polynôme énumérateur de poids de votre code est:    ');
       distribution_sans_zeros(dist,n);
       disp(' ');
        if (t==1) % si la capcité correctrice est 1
              disp(' Votre code n''est pas divisible, donc n''est pas
               e-dégénéré.');
              disp(' ');
        elseif  (floor(log2(t))~= log2(t) )  % si la capacité correctrice
                    est distincte de 1 et n'est pas une puissance de 2
                red = t/puissance_de_deux_qui_divise(t) ;
                fprintf(' Votre code est e-dégénéré; il est obtenu en
                 répétant %d fois un code de plus petite longueur. \n', red);
                  if puissance_de_deux_qui_divise(t) == 1
                     fprintf(' Aucune puissance de deux ne divise votre code.
                            Il est divisible et son degré de divisibilité est
                             donc %d. \n', t);
                     disp(' ');
                  else
                      disp(' ');
                      fprintf(' La plus grande  puissance de deux  divisant
                       votre code est %d. \n', puissance_de_deux_qui_divise(t));
                      disp(' ');
                      fprintf(' Ainsi, il est divisible et son  degré de
                              divisibilité est: %d = %d * %d. \n', t, red,
                                      puissance_de_deux_qui_divise(t));
                      disp(' ');
                  end
          else
               fprintf(' Votre code est divisible et non e-dégénéré. Son degré de
```

```
                    divisibilité est %d. \n', t);
            disp(' ');
      end

      choix_des_informations;

elseif(x==3)
      fprintf(' La longueur de ce code est: %d; \n',n);
      fprintf(' Sa dimension est: %d; \n',k);
      fprintf(' Sa distance minimale est: %d; \n',d);
      fprintf(' Sa capacité correctrice est : %d. \n \n',e);

     disp(' ');
     disp('Une matrice génératrice de votre est :'); disp(G);
     disp(' ');

     disp(' ');
     fprintf(' Le polynôme énumérateur de poids de votre code est:   ');
      distribution_sans_zeros(dist,n);
     disp(' ');
        if (t==1) % si la capacité correctrice est 1
            disp(' Votre code n''est pas divisible, donc n''est pas
                  e-dégénéré.');
            disp(' ');
        elseif  (floor(log2(t))~= log2(t) )  % si la capacité correctrice est
                 distincte de 1 et n'est pas une puissance de 2
                red = t/puissance_de_deux_qui_divise(t) ;
                fprintf(' Votre code est e-dégénéré; il est obtenu en répétant %d
                          fois un code de plus petite longueur. \n', red);
                    if puissance_de_deux_qui_divise(t) == 1
                      fprintf(' Aucune puissance de deux ne divise votre code.
                              Il est divisible et son degré de divisibilité est
                               donc %d. \n', t);
                      disp(' ');
                   else
                       disp(' ');
                       fprintf(' La plus grande  puissance de deux  divisant
                                  votre code est %d. \n',
                                puissance_de_deux_qui_divise(t));
                       disp(' ');
                       fprintf(' Ainsi, il est divisible et son  degré de
                              divisibilité est: %d = %d * %d. \n', t, red,
                               puissance_de_deux_qui_divise(t));
                       disp(' ');
                   end
        else
            fprintf(' Votre code est divisible et non e-dégénéré. Son degré de
                      divisibilité est %d. \n', t);
            disp(' ');
       end
        choix_des_informations;
elseif (x==4)
    tout_a_propos_du_code_avec_contraintes;
else
   disp(' ');
   disp(' ');
   disp(' Nous espérons que vous avez été satisfaits !       Merci ');
   disp(' ');
   disp(' ');
   disp('                    ********  FIN  DE  NOTRE PROGRAMME   *********');
end
```

♠ Bibliographie ♠

[1] J. H. Griesmer (1960), *A bound for error-correcting codes*, IBM Journal of Research and Development, vol 4, no. 5, pp. 532-542.

[2] P. Lissy (2010), *Polynômes irréductibles. Corps de rupture. Exemples et applications*, page 2.

[3] R. J. McEliece (1972), *Weight congruences for p-ary cyclic codes*, Discrete Mathematics 3, pages 177–192.

[4] D. Perrin (1996), *Cours d'algèbre*, Ellipses.

[5] H. N. Ward (1981) *Divisible codes*, Archiv der Mathematik, vol 36, pages 485–494.

[6] H. N. Ward (1998), *Divisibility of codes meeting the Griesmer bound*, Journal Combinatory Theory Serdica A, vol 83, pages 79–93.

[7] H. N. Ward (2001), *Divisible codes - A survey*, Serdica Mathématical Journal, vol 27, pages 263-278.

[8] A. Warusfel (1971), *Structures algébriques finies, Groupes, Anneaux, Corps*, Classiques Hachette, page 205.