**République Algérienne Démocratique et Populaire**
**Ministère d'Enseignement Supérieur et de la Recherche Scientifique**
**Université Mohamed Seddik Ben Yahia Jijel – Algerie –**
**Faculté des Sciences Exactes et Informatique**
**Département de Mathématiques**

# Codes correcteurs d'erreurs 2

## Codage, décodage et cryptage

Par Dr:

MOURAD CHELGHAM

Juin 2024

# Avant-Propos

C'est avec grand enthousiasme que je présente ce polycopié aux étudiants de la deuxième année Master, option : Mathématiques Fondamentales et Discrètes, intitulé : "Théorie des codes 2 - codage, décodage et cryptage", dédié à la Théorie des Codes Correcteurs d'Erreurs. Ce polycopié est une suite du polycopié intitulé " Codes correcteurs d'erreurs 1 - codage, décodage et cryptage" distiné au étudiant de la première année Master.

Au cœur de la communication numérique et de la sécurité de l'information, cette discipline (Théorie de codes) joue un rôle fondamental dans la transmission (et même l'enregistrement) fiable et efficace des données à travers divers canaux et outils de transfert de données.

La Théorie des Codes Correcteurs d'Erreurs est une branche fascinante des mathématiques et de l'informatique, qui explore les mécanismes de l'encodage et du décodage des informations pour assurer leur intégrité et leur confidentialité. Elle trouve des applications essentielles dans de nombreux domaines tels que les télécommunications, la cryptographie, les systèmes de stockage, les transmissions sans fil, et bien d'autres encore.

Ce travail a été conçu dans l'optique d'offrir aux étudiants et enseignants une introduction complète et accessible à cette discipline. Nous aborderons progressivement les concepts fondamentaux, commençant par les outils algébriques, conçus pour des codes correcteurs d'erreurs simples et linéaires aux codes cycliques utilisés dans les communications modernes. On y découvrira les méthodes de détection et de correction (ou de décodage) d'erreurs, ainsi que les techniques de codage pour optimiser la capacité de transmission et garantir l'intégrité des données.

Mon objectif, en tant qu'auteur, est de guider le lecteur avec clarté à travers les notions essentielles, en évitant autant que possible l'excès de formalisme mathématique. Il est important de noter que ce polycopié a été élaboré pour être un outil pédagogique à part entière. Ainsi, le lecteur y trouvera des exemples concrets, des illustrations et des exercices pratiques à la fin de ce polycopié pour renforcer la compréhension.

Je tiens à exprimer ma gratitude envers tous ceux qui ont contribué à l'élaboration de ce polycopié. Je tiens à remercier mes collègues, le professeur Mohamed Kerada et le docteur Ali Boussayoud pour leurs observations et leurs directives.

Dr Mourad CHELGHAM.

01 Juin 2024

# Table des matières

# Introduction

La théorie des codes correcteurs est l'étude des méthodes permettant le transfert ou le stockage d'informations de façon efficace et précise, et ainsi les protéger contre d'éventuelles erreurs, qui peuvent être, par exemple, des rayures ou de la poussière sur un C.D., une perturbation de l'appareillage, des parasites dans une ligne téléphonique ou un champ magnétique dans l'espace, dans les communications par satellites.

On se concentre, dans ce polycopié, sur la classe des codes cycliques qui est la classe des codes correcteurs linéaires fondés sur la théorie des corps finis, et en particulier les extensions de Galois, ainsi que sur les propriétés algébriques spéciales des polynômes cycliques qui sont exploitées pour concevoir un mécanisme efficace de détection et de correction d'erreurs. La principale caractéristique des codes cycliques réside dans leur capacité à garantir la détection et la correction d'un certain nombre d'erreurs en utilisant des techniques de codage et de décodage relativement simples.

Le présent polycopié consiste à présenter, en premier lieu, les différents concepts mathématiques (algébriques) utilisés dans l'étude des codes correcteurs d'erreurs. En second lieu, à donner en détail les notions et concepts de base de la théorie des codes linéaires, ainsi que les méthodes de codage et de décodage qui y sont associées. Nous explorons, en troisième lieu, les principaux concepts et caractéristiques des codes cycliques, y compris leurs structures, leurs représentations polynomiales, leurs polynômes générateurs, leurs matrices génératrices, leurs matrices de contrôle, et leur technique de codage systématique en utilisant le syndrome polynomial.

On termine par des applications des codes cycliques en cryptographie en présentant deux cryptosystèmes à clés publique basés sur les codes cycliques, à savoir le "*cryptosystème de McEliese*" et le" *cryptosystème de Niederreiter*".

Enfin à la fin du dernier chapitre, on trouve un programme de décodage par syndrome d’un code cyclique (de Hamming) par Maple et une série d'exercices sans solutions.

# Notions d'algèbre.

## Introduction

Les codes cycliques sont étroitement liés à l'algèbre linéaire et à l'arithmétique des polynômes sur un corps fini. On va présenter dans ce chapitre quelques notions et concepts algébriques associées aux codes cycliques et qui seront utilisées tout au long des chapitres qui suivent.

1. Les corps fini : Les codes cycliques sont construits sur des corps finis, les éléments d'un corps fini peuvent être utilisés pour représenter les coefficients des polynômes dans la construction des codes cycliques.
2. Les polynômes : Les codes cycliques sont définis en utilisant des polynômes, qui sont des expressions algébriques de la forme $a_0 + a_1X + a_2X^2 + ... + a_nX^n$, où les $a_i$ sont les coefficients du polynôme et $X$ est l'indéterminé. Les polynômes jouent un rôle central dans la conception des codes cycliques, car ils déterminent les propriétés du code.
3. La division polynomiale : La division polynomiale est une opération essentielle dans la construction des codes cycliques. Pour coder les données, on effectue une division polynomiale du polynôme à coder par le polynôme générateur. Le reste de cette division détermine si le mot appartient ou non au code cyclique correspondant.
4. Les polynômes générateurs et matrices génératrices : Un code cyclique est défini par un polynôme générateur ou une matrice génératrice y associée. Ce polynôme est choisi pour avoir des propriétés particulières, notamment la cyclicité qui rend la détection et la correction d'erreurs plus efficaces. Le polynôme générateur cyclique est utilisé pour générer le code cyclique à partir des données.
5. Les sommes de contrôle cycliques (CRC) : Les codes cycliques sont souvent utilisés pour la détection d'erreurs à l'aide de CRC. Les CRC sont des sommes de contrôle calculées à l'aide de polynômes pour vérifier l'intégrité des données transmises.
6. La matrices de contrôle de parité : Dans certains contextes, les codes cycliques peuvent être représentés sous forme matricielle à l'aide de matrices de contrôle de parité. Ces matrices permettent de détecter et de corriger les erreurs dans les données transmises.

En résumé, pour comprendre pleinement les codes cycliques, il est important d'avoir des connaissances en algèbre linéaire, en arithmétique des polynômes et en théorie des corps finis. Ces concepts sont fondamentaux pour la conception, l'analyse et l'application des codes cycliques dans les systèmes de communication et de stockage de données.

## 1.1 Anneaux et anneau des polynômes

### 1.1.1 Anneaux

**Définition 1.1.**

Soit $A$ un ensemble muni de deux lois internes (+) ,(.), *alors* $(A, +, .)$ est un **anneau**, si et seulement s'il vérifie les conditions suivantes:

1. $(A, +)$ est un groupe abélien (d'élément neutre $0_A$ où 0 par la loi (+), dit **élément nul**.
2. La loi (.) associative et distributive sur la loi (+).
3. $A$ admet un élément neutre noté $1_A$ ou 1 par la loi (.)

- Dans certaines définitions, la condition (3) n'est pas nécessaire. Dans ce cas, on dit que $A$ est un **anneau unitaire**.
- Si la loi (.) est commutative alors $A$ est dit **anneau commutatif**.

**Définition 1.2.**

Soit $A$ un anneau, un élément $x \in A-\{0\}$ est dit **diviseur de zéro**, s'il existe un élément $y \in A-\{0\}$ tel que $x.y = 0$ ou $y.x = 0$. Un anneau $A$ est dit **anneau intègre**, s'il n'admet pas de diviseurs de zéro.

**Exemple 1.1.**

Les anneaux $(\mathbb{Z}, +, .)$, $(\mathbb{R}, +, .)$ et $(\mathbb{R}[X], +, .)$ sont des anneaux commutatifs intègres.

$(\mathbb{Z}/6\mathbb{Z}, +, .)$ et $(F(\mathbb{R}, \mathbb{R}) , +, .)$ sont des anneaux non intègres.

**Définition 1.3.**

1. Un élément $a \in A-\{0\}$ est dit **inversible** (ou **unité**) dans $A$, s'il existe un élément $b \in A-\{0\}$ tel que $a.b = b.a = 1$.
2. L'ensemble des unités de $A$ est noté $\mathcal{U}(A)$ ou $A^*$. i.e. $\mathcal{U}(A)= \{a \in A, a \text{ inversible}\}$.

**Proposition 1.1.**

$(\mathcal{U}(A), .)$ est un groupe multiplicatif, dit **groupe des unités** de $A$.

On a : $\mathcal{U}(\mathbb{Z})=\{1, -1\}$, $\mathcal{U}(\mathbb{R}[X])=\mathbb{R}-\{0\}$.

**Définition 1.4.**

1. $(\mathbb{K}, +, .)$ est un **corps** si, et seulement si, $(\mathbb{K}, +, .)$ est un anneau unitaire dans lequel tout élément non nul est inversible. Autrement dit $\mathbb{K}$ est un corps si, et seulement si $\mathcal{U}(K)= \mathbb{K}-\{0\}$.
2. Un corps $\mathbb{K}$ est dit **corps commutatif,** si la loi (.) est commutative.
3. Un corps $\mathbb{K}$ est dit **corps fini**, si son cardinal entant qu'ensemble est fini.

**Définition 1.5.**

1. Soit $(A, +, .)$ un anneau, $I \subset A$ est dit **idéal** de $A$ si et seulement si, $I$ vérifie :

*a.* $I \leq (A, +)$.

*b.* Pour tous $x \in I$, $a \in A$ : $xa \in I$ et $ax \in I$.

2. Si $A$ est commutatif, la condition 2. devient : Pour tous $x \in I$, $a \in A$ : $ax \in I$.

**Exemple 1.2.** Soient $A$ est un anneau commutatif et $a \in A$.

1. *{0} et A* sont des idéaux de *A*, dits **idéaux triviaux**.
2. L'ensemble $I=\{ax / x \in A\}$ est un idéal de *A*, dit **idéal principal** de générateur *a*, noté $<a>$ *ou aA*.
3. $I=n\mathbb{Z} = <n>=\{nk / k \in \mathbb{Z}\}$, ensemble des multiples de *n* dans $\mathbb{Z}$ est un idéal principal de $\mathbb{Z}$.

**Définition 1.6.**

1. Un idéal $I$ de $A$ est dit **idéal maximal**, si et seulement si, pour tout idéal $J$ de $A$ tel que $I \subset J$ alors $J=I$ ou $J=A$.
2. Un idéal $I$ de $A$ est dit **idéal premier**, si et seulement si, pour tous $x, y \in A$ tel que $xy \in I$ alors $x \in I$ ou $y \in I$.

**Définition 1.7.**

Un anneau commutatif et intègre $A$ est dit **anneau principal**, si tout idéal de $A$ est un idéal principal.

**Exemple 1.3.**

1. L'anneau $\mathbb{Z}$ des entiers relatifs est un anneau principal.

2. Soit $\mathbb{K}$ un corps, l'anneau $\mathbb{K}[X]$ des polynômes sur $\mathbb{K}$, est un anneau principal.

### 1.1.2 Morphisme d'anneaux, anneaux quotients et anneau Euclidiens.

**Définition 1.8.**

1. Soient $(A, +, .)$, $(A', +, .)$ deux anneaux, d'éléments neutres, respectivement, $1_A$ et $1_{A'}$ par la loi (.). Une application $f$ de $A$ dans $A'$ est dite **morphisme d'anneaux**, si et seulement si, pour tous $x, y \in A$ on a :
   a. $f(x+y)= f(x)+ f(y)$.
   b. $f(x.y)= f(x). f(y)$.
   c. $f(1_A)=1_{A'}$.
2. Le **noyau** du morphisme d'anneaux $f$, noté $Kerf$, est définit par :
$$Kerf = \{x \in A, f(x)=0_A\} \subset A.$$
3. L'**image** du morphisme d'anneaux $f$, noté $Imf$, est définit par :
$$Imf = \{ f(x), x \in A\} \subset A'.$$

**Proposition 1.2.**

Soient $A$, $A'$ deux anneaux et $f$ un morphisme d'anneaux de $A$ dans $A'$, on a alors :

1. $Kerf$ est un idéal de $A$.
2. $Imf$ est un sous-anneau de $A'$.
3. $f$ injective, si et seulement si, $Kerf = \{0_A\}$.
4. $f$ surjective, si et seulement si, $Imf = A'$.

**Théorème 1.1.** (Premier théorème d'isomorphismes d'anneaux)

Soient $A$, $A'$ deux anneaux et $f$ un morphisme d'anneaux de $A$ dans $A'$, on a alors :
$$A/Kerf \cong Imf.$$

**Définition 1.9.**

Soit $A$ un anneau commutatif et considérons l'application :

$$f : \mathbb{Z} \to A$$
$$k \mapsto f(k)= k.1 = \begin{cases} 1+1+\cdots+1 & \text{si} \quad k > 0 \\ 0 & \text{si} \quad k = 0 \\ -1-1-\cdots-1 & \text{si} \quad k < 0 \end{cases}$$

$f$ est un morphisme d'anneaux et donc $Kerf$ est un idéal de $(\mathbb{Z}, +)$, d'où $\exists\, n \in \mathbb{N}$ tel que $Kerf = n\mathbb{Z}$. L'entier est $n$ est appelé la **caractéristique** de $A$, noté car($A$)

**Remarque 1.1.**

1. Si *f* est injectif (donc $n=0$), *A* est dit anneau de **caractéristique nulle**.
2. Si *f* n'est pas injectif ($n\neq0$), on dit que *A* est de caractéristique *n* et on écrit car(*A*)=*n*.
3. S'il existe, *n* est le plus petit entier positif non-nul vérifiant : $n.1_A=0_A$, si non $n=0$.

**Proposition 1.3.**

Soit (*A*, +, .) un anneau, *I* un idéal de *A*, comme *I* est un sous-groupe normal dans le groupe (*A*, +), alors le groupe (*A/I*, +) est un groupe abélien, tel que la loi (+) est définie par : $\bar{x}+\bar{y}=\overline{x+y}$.

La loi (.) dans *A* est compatible avec la relation d'équivalence *R* définie par :

$x\ R\ y$ si, et seulement si, $x-y\in I$. On définit donc dans *A/I* la loi (.) par : $\bar{x}\ .\ \bar{y}=\overline{x.y}$.

et (*A/I*, +, .) est un anneau, dit **anneau quotient** de *A* par *I*.

**Exemple 1.4.**

Soit $A=(\mathbb{Z}, +, .)$, $I=n\mathbb{Z}$ avec $n\in\mathbb{N}^*$, alors $(\mathbb{Z}/n\mathbb{Z}, +, .)$ est l'anneau quotient de $\mathbb{Z}$ par $n\mathbb{Z}$ tel

que : pour tout $x, y\in\mathbb{Z}$ : $\bar{x}+\bar{y}=\overline{x+y}\Leftrightarrow(x+n\mathbb{Z})+(y+n\mathbb{Z})=(x+y)+n\mathbb{Z}$.

$\bar{x}\ .\ \bar{y}=\overline{x.y}\Leftrightarrow(x+n\mathbb{Z})\ .\ (y+n\mathbb{Z})=(x.y)+n\mathbb{Z}$.

**Proposition 1.4.**

Soit *A* un anneau et *I* un idéal de *A*, alors :

1. *A/I* est un corps, si et seulement si, *I* est un idéal maximal.
2. *A/I* est un anneau intègre, si et seulement si, *I* est un idéal premier.

**Proposition 1.5.**

1. Un élément *a* de *A* est dit **premier** si, est seulement si $<a>$ est un idéal premier.
2. $<a>$ est un idéal premier si, et seulement si, $A/<a>$ est un anneau intègre.
3. Un élément *a* de *A* est **irréductible** si, et seulement si $<a>$ est un idéal maximal.
4. $<a>$ est un idéal maximal si, et seulement si $A/<a>$ est un corps.

**Définition 1.10.**

Soit $A$ un anneau commutatif intègre, $A$ est dit **anneau Euclidien,** s'il existe une application $\varphi : A\text{-}\{0\} \to \mathbb{N}\text{-}\{0\}$ appelée **Stathme** vérifiant :

1. Pour tout $x, y \in A$ tel que $y \neq 0_A$, $\exists!\ q, r \in A \quad x = yq + r$ avec $r = 0$ ou $\varphi(r) < \varphi(y)$.
2. Pour tout $x, y \in A\text{-}\{0\}$ $(\exists\ z \in A$ tel que $x = yz) \Rightarrow \varphi(r) \leq \varphi(y)$.

L'opération de trouver $q$ et $r$ est dite **Division Euclidienne** de $x$ par $y$, $q$ est dit le **quotient** et $r$ est dit le **reste** de cette division.

**Remarque 1.2.**

1) Si le reste $r=0$, alors $x=yq$, on dit que $y$ **divise** $x$ ou $y$ est un **diviseur** de $x$ ou encore que $x$ est un **multiple** de $y$.
2) Deux éléments $a$ et $b$ de $A$ sont dits **associés,** si $a$ divise $b$ et $b$ divise $a$ , c'est-à-dire s'il existe $c$ et $d$ dans $A$ tels que $a=bc$ et $b=ad$. Cela revient encore à dire qu'il existe $u$ élément de $A$ inversible tel que $a=bu$.

**Proposition 1.6.**

Tout anneau Euclidien est un anneau principal.

**Preuve.**

Soit $I$ un idéal non nul de $A$ et soit $a \in I$ avec $a \neq 0$ tel que $\varphi(a)$ minimal dans $\varphi(A\text{-}\{0\}) \subset \mathbb{N}$).

[Rappelons que tout ensemble non vide de $\mathbb{N}$ possède un élément minimal.] Soit maintenant

$x \in I$. En effectuant la Division Euclidienne de $x$ par $a$, on peut écrire $x = aq + r$ avec $r = 0$ ou bien $\varphi(r) < \varphi(a)$. Il est clair que $r=x-aq \in I$. Si on suppose $\varphi(r) < \varphi(a)$ alors $\varphi(r)$ sera l'élément minimal dans $\varphi(r)$, ce qui est absurde, et on a donc nécessairement $r = 0$ et d'où $x = aq \in < a >$ et donc $I = < a >$ principal.

**Exemple 1.4.**

1. $\mathbb{Z}$ est un anneau Euclidien avec le Sthatme $\varphi$ est définie par: $x \in \mathbb{Z}$: $\varphi(x)=|x|$
2. Si $\mathbb{K}$ $\mathbb{R}[X]$ est un anneau Euclidien avec le Sthatme $\varphi$ est définie par: $P \in \mathbb{R}[X]$:

   $\varphi(P)=\deg(P)$.

### 1.1.3 Polynômes et Anneau des polynômes.

**Définition 1.11.**

Soit *A* un anneau commutatif. Un **polynôme** sur *A* est une suite $P=(a_i)_{i\in\mathbb{N}}$ d'éléments de *A,* qui sont tous nuls sauf un nombre fini, ces éléments sont dits **coefficients** de *P*.

c.à.d. $\exists\, n\in\mathbb{N}$ : $a_n\neq 0$ et pour tout $i > n$ : $a_n=0$. L'entier *n* est dit le **degré** de *P,* noté deg(*P*).

Par convention d°(0) = -∞. Le coefficient $a_n$est dit le **coefficient dominant**. Si $a_n=1$, le polynôme *P est* dit **polynôme unitaire** ou **normalisé.**

On note $A[X]=\{ P=(a_i)_{i\in\mathbb{N}} / a_i\in A\}$, l'ensemble des polynômes sur *A*, où *X*=(0, 1, 0, …,0..) est

dite **l'indéterminé**. On muni *A*[*X*] par les lois d'addition (+) et multiplication (.) définies par : $(a_i)_{i\in\mathbb{N}}+(b_i)_{i\in\mathbb{N}}=(a_i+b_i)_{i\in\mathbb{N}}$, $(a_i)_{i\in\mathbb{N}}.(b_i)_{i\in\mathbb{N}}=(c_i)_{i\in\mathbb{N}}$, avec $c_i=\sum_{j=0}^{i} a_j\, b_{i-j}$

**Remarque 1.2.**

On a $X^1=X$, $X^2=X.X$, (0, 0,1, …,0, 0, …) et pour tout entier *i*, $X^i$= (0, 0,…,0, 1, 0, …) où 1 se trouve en position *i*+1, et par convention $X^0$= (1, 0, 0,…, 0, …), et tout polynôme $P=(a_i)_{i\in\mathbb{N}}$ s'écrit sous la forme $P=\sum_{i\in\mathbb{N}} a_i\, X^i$.

**Proposition 1.7.**

L'anneau (*A*[*X*], +, .) est un anneau commutatif dit **anneau des polynômes** d'indéterminé *X* à coefficients dans *A*. D'éléments neutres 0=(0, 0, …, 0…) pour la loi (+), dit **polynôme nul** et 1=(1, 0, 0, …, 0, …) pour la loi (.).

**Exemple 1.5.**

1. Les polynômes de degré nul sont de la forme $P=a$ / $a\in A$-{0} dit **polynômes constants**.
2. Les polynômes de degré1 sont de la forme $P=aX+b$ / $a\in A$-{0} et $b\in A$.
3. Les polynômes de degré 2 sont de la forme $P=aX^2+bX+c$ / $a\in A$-*{0} et b, c* $\in A$.

**Proposition 1.8.**

*A*[*X*] est un anneau intègre si et seulement si *A* est un anneau intègre.

**Preuve** Laissé comme exercice.

**Remarque 1.3.**

Soit $A=\mathbb{K}$ un corps commutatif. On dit qu'un polynôme P de $\mathbb{K}[X]$ est **irréductible** s'il est non constant, et si ses seuls diviseurs sont les polynômes constants et les polynômes qui lui sont **associés**, c'est-à-dire les polynômes de la forme λP, avec $\lambda\in\mathbb{K}^*$.

**Proposition 1.9.**

Si $A=\mathbb{K}$ est un corps commutatif, alors l'anneau $\mathbb{K}[X]$ est un anneau commutatif intègre Euclidien. Son Stathme est l'application :

$$\varphi : \mathbb{K}[X]\text{-}\{0\} \to \mathbb{N}\text{-}\{0\}$$
$$P \to \varphi(P)= d°(P).$$

**Conséquence 1.1.**

Si $\mathbb{K}$ est un corps commutatif, alors $\mathbb{K}[X]$ est un anneau principal.

**Remarque 1.4.**

Si $I$ est un idéal de $\mathbb{K}[X]$, alors il est principal, engendré par tout polynôme non nul de $I$, de degré minimum. Le polynôme unitaire $P$ engendrant $I$ est dit **le générateur** de $I$ et on a:

$$I=<P>=\{ PQ / Q\in \mathbb{K}[X]\}.$$

Les éléments de $I$ sont les multiples de $P$.

**Exemple 1.6.**

Soit $I=\{P\in\mathbb{R}[X]: P(1)=0\}$, alors $I$ est un idéal de $\mathbb{R}[X]$ et on a:

$P\in I \Leftrightarrow P(1)=0 \Leftrightarrow (\exists\ Q\in\mathbb{R}[X] : P=(X\text{-}1)Q) \Leftrightarrow P\in <P_1=X\text{-}1>$, donc $I=<X\text{-}1>$ l'idéal engendré par le polynôme $P_1=X\text{-}1$.

### 1.1.4 Racines d'un polynôme et l'anneau quotient $\mathbb{K}[X]/(P)$

**Définition 1.12.**

1. Un élément α de l'anneau $A$ est dit **racine** du polynôme $P=\sum_{i=0}^{n} a_i X^i \in A[X]$ si, et seulement si, $P(\alpha)=\sum_{i=0}^{n} a_i \alpha^i=0_A$.
2. La racine α est dite **racine multiple** de $P$ d'**ordre** $k$, si $P=(X\text{-}\alpha)^k Q$ avec $Q\in A[X]$ et $Q(\alpha)\neq 0$. Si $k=1$, α est dite **racine simple** et si $k=2$, α est dite **racine double**.

**Exemple 1.7.**

$P=(X\text{-}1)^2(X+1)\in\mathbb{R}[X]$ admet 1 comme racine double et (-1) comme racine simple.

**Proposition 1.10.**

1. Si $\alpha$ est une racine d'ordre $k$ d'un polynôme $P \in \mathbb{K}[X]$, alors $\alpha$ est une racine d'ordre $k$-1 de son polynôme dérivé $P'$.
2. Si $P(\alpha)=0$ et $P'(\alpha) \neq 0$, alors $\alpha$ est une racine simple de $P$.

**Proposition 1.12.**

Si $\mathbb{K}$ est un corps commutatif et $I$ un idéal de $\mathbb{K}[X]$ de générateur un polynôme $P$, alors l'anneau quotient $\mathbb{K}[X]/I=<P>$ est un anneau commutatif, d'éléments neutres $\bar{0}=I$ par l'addition et $\bar{1}=1+I$ par la multiplication. L'anneau $\mathbb{K}[X]/<P>$ est dit **anneau quotient** de $\mathbb{K}[X]$ par l'idéal $I=<P>$.

**Conséquence 1.2.**

Si $P \in \mathbb{K}[X]$ est un polynôme irréductible, alors l'anneau quotient $\mathbb{K}[X]/<P>$ est un corps commutatif.

**Exercice 1.1.**

Soit $\mathbb{K}$ un corps commutatif, alors $(\mathbb{K}[X], +, ., \times)$ est une $\mathbb{K}$-algèbre, et l'ensemble $\mathbb{K}_{n-1}[X]=\{P \in \mathbb{K}[X]$ tel que $d°(P) \leq n-1\}$ est une $\mathbb{K}$-sous-algèbre de $\mathbb{K}[X]$.

Rappelons que $(\times)$ est la loi externe définie par : $\lambda \in \mathbb{K}$, $P=\sum_{i=0}^{n} a_i X^i \in \mathbb{K}[X]$ :

$$\lambda \times P=\sum_{i=0}^{n}(\lambda a_i) X^i.$$

## 1.2 Construction et existence des corps finis.

### 1.2.1 Construction des corps finis

**Proposition 1.12.**

Si $p$ est un entier premier et $P$ un polynôme irréductible sur $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ de degré $n$, alors le corps $\mathbb{F}_p[X]/(P)$ est un corps commutatif fini isomorphe à $(\mathbb{F}_p)_{n-1}[X]$ (l'anneau des polynômes sur $\mathbb{F}_p$ de degré inferieur ou égal à $n$-1) et de cardinal $p^n$.

**Preuve.**

Soit l'application $f$ définie par :

$$f : \mathbb{F}_p[X] \to (\mathbb{F}_p)_{n-1}[X],$$
$$A \longmapsto f(A)=R$$

tel que $R$ est le reste de la division Euclidienne de $A$ par $P$.

$f$ ainsi définie est un morphisme d'anneaux surjective car:

$Imf=\{f(A) / A\in \mathbb{F}_p[X]\}= \{R / R\in \mathbb{F}_p[X]$ et $d°(R) \leq n-1\}= (\mathbb{F}_p)_{n-1}[X]$.

On a : $A=QP+R$ avec $R=0$ ou $d°(R) \leq n-1$ et le noyau de $f$ est donné par :

$Kerf=\{A\in \mathbb{F}_p[X] / f(A)=0\}=\{A\in \mathbb{F}_p[X] / R=0\}=\{ PQ / Q\in \mathbb{F}_p[X] \}=<P>$, l'idéal engendré par le polynôme $P$. Selon le premier théorème d'isomorphisme: $\mathbb{F}_p[X]/ <P> \cong (\mathbb{F}_p)_{n-1}[X]$ or $(\mathbb{F}_p)_{n-1}[X]$ est un espace vectoriel de dimension $n$ sur $\mathbb{F}_p$, donc $(\mathbb{F}_p)_{n-1}[X]\cong (\mathbb{F}_p)^n$ et d'où $card(\mathbb{F}_p[X]/(P))= card((\mathbb{F}_p)^n )=p^n$.

**Proposition 1.13.**

Si $\mathbb{K}$ est un corps fini, alors la caractéristique de $\mathbb{K}$ est un entier premier $p$ et $\mathbb{K}$ admet un sous-corps isomorphe à $\mathbb{F}_p$ dit **sous-corps premier** de $\mathbb{K}$ et le cardinal de $\mathbb{K}$ est de la forme $p^n$ avec $n\in\mathbb{N}$.

**Preuve.**

Comme $\mathbb{K}$ est un corps, alors $\mathbb{K}$ est un anneau intègre, donc $car(\mathbb{K})=0$ ou $car(A)=p$ premier. Si on suppose $car(\mathbb{K})=0$, alors $\mathbb{K}$ est infini, ce qui est absurde car $\mathbb{K}$ est fini.

L'application $f:\mathbb{Z} \longrightarrow \mathbb{K}, k \mapsto f(k) = k.1_{\mathbb{K}}$, est un morphisme d'anneaux de noyau $Kerf=p\mathbb{Z}$ et $\mathbb{F}_p=\mathbb{Z}/p\mathbb{Z}\cong f(\mathbb{Z})$ or $f(\mathbb{Z})$ est un sous-corps de $\mathbb{K}$, donc $\mathbb{K}$ est considéré comme un espace vectoriel sur $\mathbb{F}_p$ de dimension fini car $\mathbb{K}$ est un corps fini . Si $\dim_{\mathbb{F}_p}\mathbb{K} = n$ alors $\mathbb{K}\cong \mathbb{F}_p^n$ et donc $card(\mathbb{K})=p^n$.

**Proposition 1.14.**

Soit $p$ est la caractéristique d'un corps commutatif fini $\mathbb{K}$ alors :

1. $p$ est le plus petit entier non nul vérifiant $p.1_{\mathbb{K}}= 0$.
2. Pour tout $x \in \mathbb{K}$: $p.x = 0$,
3. Pour tout $x, y \in\mathbb{K}$: $(x + y)^p = x^p + y^p$ et $(x.y)^p = x^p.y^p$.
4. Pour tout $i\in\mathbb{N}, x, y \in\mathbb{K}$: $(x + y)^{p^i} = x^{p^i} + y^{p^i}$ et $(x.y)^{p^i} = x^{p^i}.y^{p^i}$.

**Proposition 1.15.**

Si $\mathbb{K}$ est un corps commutatif fini de caractéristique $p$ et de cardinal $p^n$ alors :

1. $(\mathbb{K}^*= \mathbb{K}-\{0\}, .)$ est un groupe cyclique d'ordre $p^n-1$
2. Pour tout $x\in \mathbb{K}$: $x^{p^n} = x$.

**Preuve.**

$\mathbb{K}^*= \mathcal{U}(\mathbb{K})$ est un groupe cyclique multiplicatif de cardinal $p^n-1$. Donc pour tout $x\in\mathbb{K}^*$ :

$x^{p^n-1} = 1$, en multipliant par $x$, on trouve que pour tout $x \in \mathbb{K}^*$ : $x^{p^n} = x$ et comme cette égalité est vrai pour $x = 0$, alors pour tout $x \in \mathbb{K}$ : $x^{p^n} = x$.

**Définition 1.13.**

Soit $\mathbb{K}$ un corps fini de caractéristique un entier premier $p$ et $\beta \in \mathbb{K}$. Le **polynôme minimale** de $\beta$ noté $M_\beta$ est le polynôme générateur de l'idéal (principal) $I_\beta$ définie par:

$$I_\beta = \{P \in \mathbb{F}_p[X] : P(\beta)=0\}.$$

**Proposition 1.16.**

Le polynôme minimale $M_\beta$ vérifie les propriétés suivantes :

1. $M_\beta$ est un polynôme unitaire de $\mathbb{F}_p[X]$ qui s'annule en $\beta$.
2. Si $P \in I_\beta$ (i.e. $P(\beta) = 0$) alors $M_\beta$ divise $P$.
3. $M_\beta$ est un polynôme irréductible (premier) sur $\mathbb{F}_p$.

**Preuve.**

1) et 2) découlent de la définition de $M_\beta$.

Pour 3) soit $P, Q \in \mathbb{F}_p[X]$ : $PQ \in I \Leftrightarrow (PQ)(\beta)=0 \Rightarrow P(\beta)Q(\beta)=0$ et comme $\mathbb{F}_p[X]$ est intègre alors $P(\beta)=0$ ou $Q(\beta)=0 \Rightarrow P \in I_\beta$ ou $Q \in I_\beta$ d'où $I_\beta$ est premier donc $M_\beta$ est premier (irréductible) sur $\mathbb{F}_p$.

**Proposition 1.17.**

Si $\mathbb{K}$ est un corps commutatif fini, alors $(\mathbb{K}^*, .)$ est un groupe cyclique, et tout générateur $\alpha$ de $\mathbb{K}^*$ est dit **racine primitive** (ou **élément primitif**) de $\mathbb{K}$ et $\mathbb{K}=\{0, \alpha^i / 0 \leq i \leq p^n-2\}$.

Le polynôme minimal associé à cette racine $\alpha$ est dit **polynôme primitif** de $\mathbb{K}$, qu'on note $M_\alpha$ qui a les mêmes propriétés de la Proposition 1.16.

**Proposition 1.18.**

Soit $\mathbb{K}$ un corps commutatif fini et $\beta \in \mathbb{K}$, alors l'application

$$f : \mathbb{F}_p[X] \to \mathbb{K},$$
$$P \mapsto f(P)=P(\beta),$$

est un morphisme d'anneaux de noyau $I_\beta$ et le corps $\mathbb{F}_p[X]/I_\beta$ est isomorphe à $Im(f)$.

**Preuve.**

$Kerf=\{P \in \mathbb{F}_p[X] / P(\beta)=0\}=I_\beta=(M_\beta)$ et donc d'après le premier théorème d'isomorphisme on a : $\mathbb{F}_p[X]/Kerf \cong Im(f)$ . On a $Kerf = I_\beta$ et $Im(f)=\{P(\beta)/P \in \mathbb{F}p[X]\}$ qu'on note $\mathbb{F}_p[\beta]$**.** Donc le corps quotient $\mathbb{F}_p[X]/< M_\beta >$ est isomorphe à $\mathbb{F}_p[\beta]$ **(**l'ensemble des expressions polynômiales d'indéterminé $\beta$ sur $\mathbb{F}_p$) dit **extension** de $\mathbb{F}_p$ par $\beta$.

**Remarque 1.5.** (Exercice)

$\mathbb{F}_p[\beta]$ est le plus petit sous-corps de $\mathbb{K}$ contenant $\beta$ et $\mathbb{F}_p$.

**Proposition 1.19.**

Si d°($M_\beta$)=$n$ alors $\mathbb{F}_p[\beta]$ est un espace vectoriel sur $\mathbb{F}_p$ de dimension $n$ et admet la famille $B=\{1, \beta, \beta^2, ..., \beta^{n-1}\}$ comme base.

**Théorème 1.2. (Waderburn)**

Tout corps fini $\mathbb{K}$ est commutatif.

Le théorème ci-dessous nous permet de construire un corps fini $\mathbb{K}$ dont sa caractéristique et son polynôme primitif sont connus.

**Théorème 1.3.**

Tout corps fini $\mathbb{K}$ de caractéristique un entier premier $p$ et de racine primitive $\alpha$, est isomorphe au corps quotient $\mathbb{F}_p[X]/< M_\alpha >$ tel que $M_\alpha$ est son polynôme primitif.

**Preuve.**

D'après la Proposition 1.19 en prenant $\beta=\alpha$ alors on trouve que $\mathbb{F}_p[X]/ I_\alpha = < M_\alpha>$ est isomorphe à $\mathbb{F}_p[\alpha]$ (ensembles des expressions polynomiales en $\alpha$ à coefficients dans $\mathbb{F}_p$).

Montrons que $\mathbb{K}= \mathbb{F}_p[\alpha]$.

Il est clair que $\mathbb{F}_p[\alpha] \subset \mathbb{K}$. Soit $x\in\mathbb{K}$, si $x=0$ alors $x\in Imf=\mathbb{F}_p[\alpha]$. Soit $x\neq 0$ et $x \in \mathbb{K}^*=<\alpha>$, alors il existe $m\in\mathbb{N}^*$ : $x=\alpha^m$ ce qui montre que $x$ est un élément de $\mathbb{F}_p[\alpha]$ , d'où $\mathbb{K} \subset \mathbb{F}_p[\alpha]$.

Enfin comme $M_\alpha$ est irréductible alors $\mathbb{F}_p[X]/<M_\alpha>$ est un corps isomorphe au corps $\mathbb{K}$.

### 1.2.2 Existence des corps finis

Pour $p$ un entier premier et $n\in\mathbb{N}^*$, posons-nous les questions suivantes :

1. Si $P$ est un polynôme unitaire et irréductible sur $\mathbb{F}_p$ existe-il un corps fini $\mathbb{K}$ et une racine primitive $\alpha$ de $\mathbb{K}$ tel que $P=M_\alpha$.
2. Existe-il un polynôme unitaire et irréductible de degré $n$ sur $\mathbb{F}_p$.
3. Existe-il un corps fini $\mathbb{K}$ de cardinal $p^n$.

**Remarque 1.6.**

Si $L$ est un corps commutatif quelconque et $\mathbb{K}$ un sous-corps de $L$, en remplaçant $\mathbb{K}$ par $L$ et $\mathbb{F}_p$ par $\mathbb{K}$ on peut généraliser la définition de $I_\beta$ pour $\beta\in L$, et le polynôme minimal $M_\beta$ de $\beta$ sur $\mathbb{K}$ et ses propriétés comme dans le cas précédent.

**Théorème 1.4.**

Soit $\mathbb{K}$ un corps commutatif et $P \in \mathbb{K}[X]$ irréductible, unitaire et non constant alors, il existe une extension $L$ de $\mathbb{K}$ et $\alpha \in L$ tel que $P=M_\alpha$.

**Preuve**.

Il suffit de prendre $L= \mathbb{K}[X]/<P>$ qui est un corps commutatif et l'application $f : \mathbb{K} \to \mathbb{K}[X]/<P>$, $x \mapsto f(x) = \bar{x}$, $f$ est un morphisme de corps injectif, donc $\mathbb{K}$ est isomorphe à $f(\mathbb{K})$ qui est un sous-corps de $L= \mathbb{K}[X]/(P)$, d'où $L$ est une extension de $\mathbb{K}$. *Si* $P=\sum_{i=0}^{i=n} a_i X^i$, alors $P(\alpha)=\sum_{i=0}^{i=n} a_i \alpha^i = \sum_{i=0}^{i=n} a_i \bar{X}^i = \overline{\sum_{\iota=0}^{\iota=n} a_\iota X^\iota} = \bar{P} = \bar{0}$, alors $P$ vérifie la propriété 2. de la Proposition 1.17, donc $P= M_\alpha$.

**Définition 1.14.**

Une extension (ou sur-corps) $L$ d'un corps commutatif $\mathbb{K}$ est dite **corps de rupture** d'un polynôme $P \in \mathbb{K}[X]$, si et seulement si, $\exists \alpha \in \mathbb{K}$, $a \in L$, tel que $P=\alpha(X-a)Q$, avec $Q \in \mathbb{K}[X]$.

**Définition 1.15.**

Une extension (ou sur-corps) $L$ d'un corps commutatif $\mathbb{K}$ est dite **corps de décomposition** d'un polynôme $P \in \mathbb{K}[X]$, si et seulement si, $\exists \alpha \in \mathbb{K}$, $a_1, a_2, \ldots, a_n \in L$ tel que $P=\alpha \prod_{i=1}^{i=n}(X - a_i)$.

**Proposition 1.20.**

Soit $\mathbb{K}$ un corps commutatif et $P \in \mathbb{K}[X]$ avec $d°(P) \geq 1$ alors $P$ admet un corps de rupture et un corps de décomposition sur $\mathbb{K}$.

**Théorème 1.5.**

Si $n \in \mathbb{N}^*$ et $p$ entier premier alors il existe un corps fini $\mathbb{K}$ de cardinal $p^n$ et un polynôme irréductible $P$ sur $\mathbb{F}_p$ de degré $n$.

**Preuve.**

Soit $P=X^{p^n} - X$ le polynôme de degré $p^n$ sur $\mathbb{K}$ et soit $\mathbb{K}_1$ le corps de décomposition du polynôme $P_1 =X^{p^n-1} - 1$. Le polynôme dérivé de $P_1$ est $P_1'= -X^{p^n-2}$, qui n'admet que la racine nulle, qui n'est pas racine de $P$ et donc toutes les $p^n - 1$ racines de $P_1$ sont distinctes et différentes de 0, alors $P$ admet $p^n$ racines distinctes. Soit $\mathbb{K}$ l'ensemble des racines de $P$ dans $\mathbb{K}_1$. Montrons que $\mathbb{K} = \{x \in \mathbb{K}_1 : x^{p^n} - x = 0\}$ est un sous-corps de cardinal $p^n$ du corps $\mathbb{K}_1$. Il suffit de montrer qu'il est stable par les deux lois (+) et (.).

En effet : si $x, y \in \mathbb{K}$ alors on a : $(x+y)^{p^n} = x^{p^n} + y^{p^n} = x+y$, donc $x+y \in \mathbb{K}$ et $(x.y)^{p^n} = x^{p^n}.y^{p^n} = x.y$, donc $x.y \in \mathbb{K}$, donc $\mathbb{K}$ est un corps de cardinal $p^n$ et son polynôme primitif $M_\alpha$ est un polynôme irréductible de degré $n$ sur $\mathbb{F}_p$ .

**Proposition 1.21.**

Tous les corps finis de caractéristique un entier premier $p$ et de cardinal $p^r$ , $r \in \mathbb{N}^*$, sont isomorphes. Cet unique corps fini à isomorphisme prés est dit **corps de Galois** noté $\mathbb{F}_q = \mathbb{F}_{p^r}$.

**Remarque 1.6.**

Pour décrire le corps de Galois $\mathbb{K}$ de caractéristique $p$ sur le corps premier $\mathbb{F}_p$, il suffit de :

1- Soit connaitre un polynôme primitif de degré $r$ sur $\mathbb{F}_p$ (i.e. le polynôme minimal $M_\alpha$ d'une racine primitive α de $\mathbb{K}$), et $\mathbb{F}_q = \mathbb{F}_p(\alpha)$ ou $\mathbb{F}_q \approx \mathbb{F}_p[X] / <M_\alpha>$.

2- Soit choisir un polynôme $P$ de $\mathbb{F}_p[X]$, de degré $r$, irréductible sur $\mathbb{F}_p$ et $\mathbb{F}_q \approx \mathbb{F}_p[X]/<P>$

**Exemple 1.7.**

Construction d'un corps de Galois $\mathbb{K} = \mathbb{F}_9$, $q = 9 = 3^2$ donc $p$=car($\mathbb{K}$)=3, $r$=d°($M_\alpha$)=2, $\mathbb{F}_9 = \{0\} \cup \mathbb{F}_9^* = \{0, \alpha^i, 0 \le i \le 7\} = \{0, 1, \alpha, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\}$. Soit $M_\alpha$ le polynôme primitif de $\mathbb{F}_9$ (le polynôme minimal associe à α), $M_\alpha$ est de degré $n$=2, irréductible et unitaire sur $\mathbb{F}_3$. On peut prendre le polynôme primitif $M_\alpha = X^2+X+2$ ou $M_\alpha = X^2+2X+2$.

**1ere méthode.** Prenons $M_\alpha = X^2+X+2$. On a $M_\alpha(\alpha)=0 \Leftrightarrow \alpha^2+\alpha+2=0 \Leftrightarrow \alpha^2 = -\alpha-2 = 2\alpha+1$, $\alpha^3 = 2\alpha+2$, $\alpha^4 = 2$, $\alpha^5 = 2\alpha$, $\alpha^6 = \alpha+2$, $\alpha^7 = \alpha+1$, donc $\mathbb{F}_9 = \{0, 1, 2, \alpha, \alpha+1, \alpha+2, 2\alpha, 2\alpha+1, 2\alpha+2\}$.

**2eme méthode.** Ou encore, considérons le polynôme $P = X^2+1$qui est irréductible de degré 2 sur $\mathbb{F}_3$.

Alors le corps $\mathbb{F}_9$ est isomorphe au corps quotient $\mathbb{F}_3[X]/<P> = \mathbb{F}_3[X]/<X^2+1>$

Donc $\mathbb{F}_9 = \{a\bar{X} + b / a, b \in \mathbb{F}_3\}$. Posons $\bar{X} = \beta$, alors $\mathbb{F}_9 = \{0, 1, 2, \beta, 2\beta, \beta+1, \beta+2, 2\beta+1, 2\beta+2\}$.

**Exemple 1.8.**

Construction d'un corps de Galois de cardinal $q = 8 = 2^3$ donc $p$=car($\mathbb{K}$)=2, $\mathbb{F}_8 = \{0\} \cup \mathbb{F}_8^* = \{0, \alpha^i, 0 \le i \le 6\} = \{0, 1, \alpha, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}$. Soit $M_\alpha$ le polynôme primitif de $\mathbb{F}_8$ (le polynôme minimal associe à α), $M_\alpha$ est de degré $r$=3, irréductible et unitaire sur $\mathbb{F}_2$. On peut prendre le polynôme primitif $M_\alpha = X^3+X+1$ ou $M_\alpha = X^3+X^2+1$.

Prenons $M_\alpha = X^3+X+1$. On a $M_\alpha(\alpha)=0 \Leftrightarrow \alpha^3+\alpha+1=0 \Leftrightarrow \alpha^3 = \alpha+1$, $\alpha^4 = \alpha^2+\alpha$, $\alpha^5 = \alpha^3+\alpha^2 = \alpha^{2+}$ $\alpha+1$, $\alpha^6 = \alpha^3+\alpha^2+\alpha$, $\alpha^7 = 1$, donc $\mathbb{F}_8 = \{0, 1, \alpha, \alpha+1, \alpha^2, \alpha^2+1\ \alpha^2+\alpha, \alpha+1, \alpha+2, 2\alpha, 2\alpha+1, 2\alpha+2\}$.

**Exemple 1.9.**

(Exercice) Décrire le corps de Galois de cardinal $q$=16 ($\mathbb{F}_{16}$).

## 1.3 Groupe et corps des racines nièmes de l'unité et la décomposition du polynôme $X^n$ -1

Dans ce paragraphe, on va définir et voir comment construire le plus petit corps qui contient tous les racines du polynôme $X^n$-1, pour $n$ un entier non nul donné et qui nous permet de décomposer ce polynôme en produit de polynômes irréductibles sur $\mathbb{F}_p$.

### 1.3.1 Groupe des racines nièmes de l'unité

**Définition 1.16.**

Soit $n$ un entier non nul et $\mathbb{K}$ un corps commutatif, on appel **racine nième de l'unité** sur $\mathbb{K}$, tout élément $\alpha$ de $\mathbb{K}$ racine du polynôme $X^n-1 \in \mathbb{K}[X]$, c'est aussi l'ordre de $\alpha$ dans le groupe $\mathbb{K}^*$.

On note $G_n(\mathbb{K})$ l'ensemble des racines *nièmes* de l'unité dans $\mathbb{K}$ c.à.d.

$$G_n(\mathbb{K})=\{x\in\mathbb{K}-\{0\}: x^n-1=0\}$$

**Exemple 1.9.**

1- Les racines troisième de l'unité sur $\mathbb{C}$ sont $\alpha_1=1$, $\alpha_2=\frac{1}{2}+\frac{\sqrt{3}}{2}i$, $\alpha_3=\frac{1}{2}-\frac{\sqrt{3}}{2}i$ et sur $\mathbb{R}$ ou $\mathbb{Q}$ la seule racine troisième de l'unité est la racine triviale $\alpha=1$.

2- Les racines quatrièmes de l'unité sur $\mathbb{R}$ sont $\alpha_1=1$, $\alpha_2=-1$.

Rappelons le théorème suivant concernant les groupes cycliques :

**Théorème 1.6.**

Si $G$ est un groupe cyclique d'ordre $n$ et $H=\{x\in G, x^k=1\}$, l'ensemble des racines d'ordre $k$ de l'unité, alors $H$ est un sous-groupe (cyclique) de $G$ d'ordre $d$ = PGCD($n$, $k$).

**Théorème 1.7. Groupe des racines nième de l'unité**

Soit $\mathbb{K}=\mathbb{F}_q$ le corps de Galois de cardinal $q=p^r$ tel que $r\in\mathbb{N}^*$et de caractéristique p premier. L'ensemble $G_n(\mathbb{K})=\{x\in\mathbb{K}-\{0\}: x^n-1=0\}$ est un sous-groupe cyclique d'ordre $d$=PGCD($p^r-1$, $n$) dit **groupe des racines nièmes de l'unité** sur $\mathbb{F}_p$ et on a $G_n(\mathbb{K})=G_d(\mathbb{K})$.

**Preuve**.

Il suffit d'appliquer le théorème ci-dessus pour $G=\mathbb{K}^*=\mathbb{F}_q-\{0\}$ et $H=G_n(\mathbb{K})$

### 1.3.2 Corps des racines nièmes de l'unité sur $\mathbb{F}_p$

Soit $p$ un entier premier et $n\in\mathbb{N}^*$ et cherchons un corps de décomposition $\mathbb{K}$ de $X^n$-1 sur $\mathbb{F}_p$

c-à-d. un sur-corps $\mathbb{K}$ du corps premier $\mathbb{F}_p$ tel que $X^n$-1 se décompose en produit de polynômes

de premiers degré ( pas nécessairement tous différents) c.à.d. $X^n-1=\prod_{i=1}^{i=n}(X-a_i)$.

**Remarque 1.7.**

On peut écrire l'entier $n$ sous forme $n=Np^m$ où $N\wedge p=1$et $m\in\mathbb{N}$, on a 2 cas :

1. $n\wedge p=1$ ($n$ premier avec $p$), donc $m=0$ et $N=n$.
2. $n$ n'est pas premier avec $p$, dans ce cas $n=Np^m$ et $m\neq 0$ et $N\wedge p=1$. Ce cas peut être envoyer au premier cas. C.à.d. que ladécomposition de $X^n-1$ avec $n$ n'est pas premier avec $p$, se déduit de celle de $X^N-1$ avec $N$ premier avec $p$, en effet :

$X^n-1=0\Leftrightarrow X^{Np^m}-1=0\Leftrightarrow(X^n-1)^{p^m}=0\Leftrightarrow X^n-1=0$, et d'où $G_n(\mathbb{K})=G_N(\mathbb{K})$.

**Théorème 1.8. (**Construction du corps des racines nièmes de l'unité )

Soit $p$ un entier premier et $n\in\mathbb{N}$, il existe un unique (le plus petit ) corps de décomposition de $X^n-1$ sur $\mathbb{F}_p$ , c'est le corps $\mathbb{K}=\mathbb{F}_{p^r}$ où r est le plus petit entier non nul tel que $N$ divise $p^r-1$. Ce corps est dit **corps des racines nièmes de l'unité** sur $\mathbb{F}_p$.Et on a :

$X^n-1=\prod_{i=1}^{i=n}\left(X-\beta^i\right)^{p^m}$ ,tel que $\beta=\alpha^s$, avec $s=\frac{p^r-1}{N}$ et $\alpha$ une racine primitive de $\mathbb{K}$.

**Preuve.**

1. Si $n$ premier avec $p$**.** En effectuant la division Euclidienne de $p$ par $n$ on trouve :

$p=q.n+p_1$ avec $0<p_1<n\Rightarrow p\equiv p_1[n]$. On a $p\wedge n=1$ donc $p_1\wedge n=1$ et $p_1<n$ donc $\overline{p_1}$ est inversible dans $\mathbb{Z}/n\mathbb{Z}$. Soit $r$ est l'ordre de $\overline{p_1}$, donc r est le plus petit entier non nul tel que $p_1^r\equiv 1[n]$ et comme $p\equiv p_1[n]$ alors $p^r\equiv 1[n]$ donc $p^r-1\equiv 0[n]$. En résumé $r$ est le plus petit entier non nul tel que $n$ divise $p^r-1$. Considérons le corps $\mathbb{K}=\mathbb{F}_{p^r}$ où $r$ est l'entier définie ci-dessus. Selon le Théorème 1.7. ci-dessus le groupe cyclique $G_n(\mathbb{K})$ est d'ordre $d=\text{PGCD}(p^r-1,n)=n$ (car on a : $n$ divise $p^r-1$).

Si $\beta\in\mathbb{K}^*$ est un générateur de $G_n(\mathbb{K})$, alors $G_n(\mathbb{K})=\langle\beta\rangle=\{1,\beta,\beta^2,\ldots,\beta^{n-1}\}$ constitué de $n$ racines distinctes de $X^n-1$ et $X^n-1$ qui se décompose donc par :

$X^n-1=\prod_{i=1}^{i=n}\left(X-\beta^i\right)$ et donc $\mathbb{K}=\mathbb{F}_{p^r}$ est un corps de décomposition de $X^n-1$ sur $\mathbb{F}_p$.

**2.** Si $n$ n'est pas premier avec $p$. Soient $n=Np^m$ et $m\in 0$ tel que $N\wedge p=1$ et $r$ le plus petit entier non nul tel que $N$ divise $p^r-1$, alors le corps $\mathbb{K}=\mathbb{F}_{p^r}$ est le corps de décomposition de $X^N-1$ et donc celui de $X^n-1$ qui se décompose par :

$X^N-1=\prod_{i=0}^{i=N-1}\left(X-\beta^i\right)\Rightarrow X^n-1=(X^N-1)^{p^m}=\prod_{i=0}^{i=N-1}\left(X-\beta^i\right)^{p^m}$.

Si $\alpha$ est une racine primitive de $\mathbb{K}$ (c.à.d. générateur de $\mathbb{K}^*$) alors d'après les propriétés des groupes cycliques, on peut prendre $\beta$ (générateur de $G(\mathbb{K})$) de la forme $\beta=\alpha^s$ tel que

$s = \frac{p^r - 1}{n}$.

**Exercice 1.2.**

Le corps $\mathbb{K}=\mathbb{F}_{p^r}$ est le plus petit corps de décomposition de $X - 1$ sur $\mathbb{F}_p$.

**En effet.**

- Si $L=\mathbb{F}_{p^v}$ est un autre corps de décomposition de $X^n$-1 sur $\mathbb{F}_p$, alors $n$ divise $p^v - 1$, comme $r$ est le plus petit entier non nultel que $n$ divise $p^r - 1$ alors par division Euclidienne de $v$ par $r$ on trouve : $v=rq+t$ tel que $0 \leq t < r$, alors $t=0$, si non on aura : $p^v \equiv 1[n]$ et $p^r \equiv 1[n]$ donc $p^v \equiv 1[n]$ et $p^{-qr} \equiv 1[n]$ ce qui donne : $p^t = p^{v-qr} \equiv 1[n]$. Ceci montre que t est le plus petit entier non nul tel que $n$ divise $p^t-1$, ce qui est absurde, donc $t=0$ et $v=rq$ d'où $r$ divise $v$ et donc $\mathbb{K}=\mathbb{F}_{p^r}$ est un sous-corps de $L=\mathbb{F}_{p^v}$. Donc chaque corps de décomposition $L$ de $X^n$-1est un sur-corps de $\mathbb{K}$.
- Si $L$ est un sur-corps de $\mathbb{K}=\mathbb{F}_{p^r}$, alors $L$ est automatiquement un corps de décomposition de de $X^n$-1 sur $\mathbb{F}_p$. D'où $\mathbb{K}=\mathbb{F}_{p^r}$ est le plus petit corps de décomposition de $X^n$-1 sur $\mathbb{F}_p$.

**Exemple 1.10.**

Déterminer $\mathbb{K}$ le corps des racines 15-imes de l'unité sur $\mathbb{F}_2$ et décomposer $X^{30}$ -1 sur $\mathbb{F}_2$.

$N$=15 et $p$=2. Le corps concerné est $\mathbb{K}=\mathbb{F}_{2^r}$, tel que $r$ est le plus petit entier non nul tel que $N$=15 divise $2^r$ -1, on trouve $r$= 4 et donc $\mathbb{K}=\mathbb{F}_{16}$.

$X^{15} - 1 = \prod_{i=0}^{i=14}(X - \beta^i)$ et $X^{30} - 1 = (X^{15} - 1)^2 = \prod_{i=0}^{i=14}(X - \beta^i)^2$,

$\beta=\alpha$ est la racine primitive 15ieme de l'unité qui est une racine primitive de $\mathbb{K}$.

### 1.3.3 Décomposition de $X^n - 1$ en produit de polynômes irréductibles sur $\mathbb{F}_p$.

Soit $\mathbb{K}=\mathbb{F}_{p^r}$ corps des racines *nièmes* de l'unité sur $\mathbb{F}_p$ et $G_n(\mathbb{K}) = \{x \in \mathbb{K}^*: X^n - 1 = 0\}$

**Définition 1.17.**

On appel **racine nième primitive de l'unité**, tout générateur $\beta$ du groupe cyclique $G_n(\mathbb{K})$.

L'ensemble de ces racines nième primitive de l'unité noté $P_n(\mathbb{K})$ est donnée par:

$$P_n(\mathbb{K})=\{\beta \in G_n(\mathbb{K}) / \beta \text{ engendre } G_n(\mathbb{K})\}.$$

**Définition 1.18.**

Soit $\mathbb{K}=\mathbb{F}_{p^r}$ corps des racines nièmes de l'unité sur $\mathbb{F}_p$.

On appelle **Polynôme cyclotomique** d'indice $n$, le polynôme noté $\emptyset_n(X) \in \mathbb{F}_p[X]$ dont ses

racines sont les racines nièmes primitives de l'unité dans K. i.e.

$$\emptyset_n(X) = \prod_{\varepsilon \in P_n(\mathbb{K})} (X - \varepsilon).$$

**Définition 1.19.**

On appelle **Fonction indicatrice d'Euler** la fonction:

$$\varphi: \mathbb{N} \longrightarrow \mathbb{N}$$

$$n \longmapsto \varphi(n) = \text{card}\{\, i \in \mathbb{N}: i < n \; et \; i \wedge n = 1\}$$

**Exemple 1.11.**

Pour $n$=6, $\varphi(6) = \text{card}\{1,5\} = 2$.

Rappelons le théorème suivant concernant les générateurs d'un groupe cyclique :

**Théorème 1.9.**

Si $G$ est un groupe cyclique d'ordre $n$ engendré par $g$, alors :

$$(g^k \text{ engendre } G) \Leftrightarrow 1 \leq k \leq n-1 \text{ et } k \wedge n = 1.$$

L'ensemble des générateurs de $G$ est :

$$P = \{g^k, 1 \leq k \leq n-1 \text{ et } k \wedge n = 1\} \text{ et } \text{card}(P) = \varphi(n)$$

**Proposition 1.22.**

Soient $n \in \mathbb{N}$ et $\mathbb{K}=\mathbb{F}_{p^r}$ corps des racines nièmes de l'unité sur $\mathbb{F}_p$.

Si $\beta$ est une racine primitive nième de l'unité, Alors l'ensemble de tous les générateurs de $G_n(\mathbb{K})$ (les racine primitives nièmes de l'unité) est :

$$P_n(\mathbb{K}) = \{\beta^j / 1 \leq j \leq n-1 \text{ et } j \wedge n = 1\} \text{ et } \text{card}(P_n(\mathbb{K})) = \varphi(n).$$

**Preuve.**

Il suffit d'appliquer le théorème précèdent avec $G= \mathbb{K}^*$ et $g=\beta$.

**Proposition 1.23.**

Soient $n \in \mathbb{N}$ et $\mathbb{K}=\mathbb{F}_{p^r}$ le corps des racines *nièmes* de l'unité sur $\mathbb{F}_p$ Si $\beta$ est une racine primitive nième de l'unité alors, le polynôme cyclotomique $\emptyset_n(X)$ s'écrit :

$$\emptyset_n(X) = \prod_{\substack{1 \leq j \leq n \\ j \wedge n = 1}} \left(X - \beta^j\right) \text{ et } d^\circ(\emptyset_n(X)) = \varphi(n).$$

**Preuve.**

$\emptyset_n(X) = \prod_{\varepsilon \in P_n(\mathbb{K})}(X - \varepsilon)$ et $\varepsilon$ est de la forme $\varepsilon=\beta^j$ tel que $1 \leq j \leq n-1$ et $j \wedge n = 1$.

Donc $\emptyset_n(X) = \prod_{\substack{1 \leq j \leq n \\ j \wedge n = 1}} \left(X - \beta^j\right)$ et pour le degré on a :

d°(∅(*X*))=d°$\left(\prod_{\substack{1\le j\le n\\ j\wedge n=1}}(X-\beta^j)\right)=\sum_{\substack{1\le j\le n\\ j\wedge n=1}}$ d°$(X-\beta^j)$=$\sum_{\substack{1\le j\le n\\ j\wedge n=1}}1=card\left(P_n(\mathbb{K})\right)=\varphi(n)$.

**Proposition 1.24.**

Soient $n\in\mathbb{N}^*$ et *p* premier tel que *n* premier avec *p* et $\mathbb{K}=\mathbb{F}_{p^r}$ corps des racines nièmes de l'unité sur $\mathbb{F}_p$. Alors le polynôme $X^n$-1 se décompose par :

$$X^n-1=\prod_{d/n}\emptyset_d(X).$$

**Preuve**.

Comme $n\wedge p$=1(donc *n*=*N*), alors si β est une racine primitive nième de l'unité, le groupe $G_n(\mathbb{K})=\{1,\beta,\dots,\beta^{n-1}\}$ est de cardinal *n*. Si *d* divise *n*, on note $P_d(\mathbb{K})$ : l'ensemble des racines primitive d'ordre *d* de l'unité. Il est évident que $P_d(\mathbb{K})\subset G_n(\mathbb{K})$ et la famille $\{P_d(\mathbb{K})\}_{d/n}$ forme une partition de $G_n(\mathbb{K})$ (exercice) et donc :

$X^n-1=\prod_{j\in[0,n-1]}(X-\beta^j)=\prod_{\varepsilon\in}G_n(\mathbb{K})\ (X-\varepsilon)=\prod_{d/n}\prod_{\varepsilon\in}P_d(\mathbb{K})\ (X-\varepsilon)$,

or $\prod_{\varepsilon\in}P_d(\mathbb{K})\ (X-\varepsilon)$ n'est que le polynôme cyclotomique $\emptyset_d(X)$ d'où $X^n-1=\prod_{d/n}\emptyset_d(X)$.

**Conséquence 1.3.**

Si *n* n'est pas premier avec *p*, alors $n=N.p^m$ avec $N\wedge p$=1 et on a :

$$X^n-1=\prod_{d/n}(\emptyset_d(X))^{p^m}.$$

**Preuve**.

Il suffit de décomposer $X^N-1=\prod_{d/N}\emptyset_d(X)$. Et on a :

$$X^n-1=(X^N-1)^{p^m}\Rightarrow X^n-1=\prod_{d/n}(\emptyset_d(X))^{p^m}.$$

**Proposition 1.25**.

Si $n\in\mathbb{N}^*$ et *p* premier tel que *n* premier avec *p*. Alors les polynômes cyclotomiques $\emptyset_n(X)$sont des polynômes unitaires à coefficients dans $\mathbb{F}_p$.

**Preuve**.

On démontre par récurrence sur *n*. Si *n*=1, $\emptyset_1(X)=X-1$ donc unitaire et $\emptyset_1(x)\in\mathbb{F}_p[X]$. On suppose que pour tout m < n : $\emptyset_m(X)$ unitaire et $\emptyset_m(X)\in\mathbb{F}_p[X]$. On a :

$$X^n-1=\prod_{d/n}\emptyset_d(X).=\prod_{\substack{d/n\\ d\neq n}}\emptyset_d(X).\ \emptyset_n(X),$$

donc

$$X^n-1=P(x).\emptyset_n(X)\text{ avec }P(x)=\prod_{d/n,d\neq n}\emptyset_d(X).$$

On a : pour tout *d*/*n* : d < *n*, d'après le processus de récurrence , $\emptyset_d(X)$ est unitaire et $\emptyset_d(X)\in\mathbb{F}_p[X]$, donc *P*(*X*) est unitaire et $P(X)\in\mathbb{F}_p[X]$ et comme $X^n-1\in\mathbb{F}_p[X]$ est unitaire alors

de l'égalité $X^n-1= P(X).\emptyset_n(X)$, on déduit que $\emptyset_n(X)\in\mathbb{F}_p[X]$ et $\emptyset_n(X)$ unitaire.

### 1.3.4 Calcul direct des polynômes cyclotomiques.

**Proposition 1.26.**

Si $p$ est un entier premier alors :

$$\emptyset_p(X) = X^{p-1} + \dots + X+1.$$

**Preuve.** Comme $p$ premier, les diviseurs de $p$ sont 1 et $p$, donc on a

$X^p-1= \emptyset_1(X)\emptyset_p(X) = (X-1)\,\emptyset_p(X)$ donc $\emptyset_p(X) = \frac{X^p-1}{X-1} = X^{p-1} + X^{p-2} + \dots + X+1$.

**Exemple 1.12.**

$\emptyset_2(X) = X+1$ , $\emptyset_3(X)= X^2+X+1$, $\emptyset_5(X) = X^4+X^3+X^2+X+1$.

**Conséquence 1.4.**

Si $p$ un entier premier et $k\in\mathbb{N}^*$ alors :

$$\emptyset_{p^k}(X) = X^{(p-1)p^{k-1}}+X^{(p-2)p^{k-1}} +\dots+ X^{2p^{k-1}}+X^{p^{k-1}}+ 1= \frac{X^{p^k}-1}{X^{p^{k-1}}-1}.$$

**Preuve.**

$X^{p^k} - 1 = \prod_{d/p^k} \emptyset_d(X) = \emptyset_{p^k}(X) \prod_{d/p^{k-1}} \emptyset_d(X) \Rightarrow X^{p^k} - 1= \emptyset_{p^k}(X)\,(X^{p^{k-1}} - 1)$

Ce qui donne : $\emptyset_{p^k}(X) = \frac{X^{p^k}-1}{X^{p^{k-1}}-1}$.

**Remarque 1.8.**

$$\emptyset_{p^k}(X) = \frac{X^{p^k}-1}{X^{p^{k-1}}-1} = \frac{(X^{p^{k-1}})^p-1}{X^{p^{k-1}}-1} = \emptyset_p\left(X^{p^{k-1}}\right).$$

**Exemples 1.13.**

$\emptyset_8(X) = \emptyset_{2^3}(X) = \frac{X^8-1}{X^4-1} = \emptyset_2(X^4) = X^4+1$.

**Théorème 1.10.**

Soit $p$ entier premier et $n\in\mathbb{N}^*$ .

1. Si $p$ divise $n$ alors : $\emptyset_{np}(X)= \emptyset_n(X^p)$.
2. Si $p$ ne divise pas $n$ alors : $\emptyset_{np^k}(X)= \frac{\emptyset_n\left(X^{p^k}\right)}{\emptyset_n\left(X^{p^{k-1}}\right)}$ et en particulier $\emptyset_{np}(X)=\frac{\emptyset_n(X^p)}{\emptyset_n(X)}$.

**Théorème 1.11.**

Soient $p$ premier, $n\in\mathbb{N}^*$ et $\mathbb{K}=\mathbb{F}_{p^r}$, le corps des racines *nièmes* de l'unité sur $\mathbb{F}_p$. Alors $\emptyset_n(X)$ se décompose en produit de $\varphi(n)/r$ polynômes irréductibles de degré r et à coefficients dans

$\mathbb{F}_p$.

**Preuve.**

Soit $\mathbb{K}=\mathbb{F}_{p^r}$, le corps des racines *nièmes* de l'unité sur $\mathbb{F}_p$. Soit $\beta$ une racine *n*ieme primitive de l'unité et posons $I_n=\{i\in [1, n]: i\wedge n=1\}$ et on a: $p^r=1$ dans $\mathbb{Z}/n\mathbb{Z}$ et donc pour tout $i\in I_n$ : $p^r i = i$ dans $\mathbb{Z}/n\mathbb{Z}$ et comme $i$ est premier avec $n$ alors, les ensembles $J_i=\{ i, pi, \dots , p^{r-1} i\}$ où $i\wedge n=1$ dits **classes cyclotomiques**, sont de cardinal $r$ et forment une partition à $I_n$ c.à.d $I_n = \cup_{t=1}^{t=k} J_t$.

$\emptyset_n(X) = \prod_{\varepsilon\in P_n(\mathbb{K})}(X-\varepsilon) = \prod_{\substack{1\le j\le n \\ j\wedge n=1}}(X-\beta^j) = \prod_{j\in I_n}(X-\beta^j) = \prod_{j\in \cup_{t=1}^{t=k} J_t}(X-\beta^j)$

avec card($J_t$) = $r$ alors: $\emptyset_n(X) = \prod_{j\in J_1}(X-\beta^j)\prod_{j\in J_2}(X-\beta^j)\dots\prod_{j\in J_k}(X-\beta^j)$.

Si $j_i$ est le représentant de la classe $J_i$ , alors pour tout $i\in[1,k]$:

$\prod_{j\in J_i}(X-\beta^j) = \prod_{0\le l\le r-1}\left(X-\beta^{j_i p^l}\right)$.

**Lemme 1.1.**

Soit $\beta$ est un élément d'un corps fini $\mathbb{K}$ de caractéristique $p$ , et soit $M_\beta$ le polynôme minimal de $\beta$ de degré $r$ alors :

1. Les éléments $\beta, \beta^p, \dots, \beta^{p^{r-1}}$ (dits **conjugués** de $\beta$) sont distincts.
2. $M_\beta$ s'écrit : $M_\beta=\prod_{0\le l\le r-1}\left(X-\beta^{p^l}\right)$ et pour tout $l\in[1,r\text{-}1]$ : $M_\beta = M_{\beta^{p^l}}$. D'après le lemme ci-dessus : $\emptyset_n(X) = M_{\beta^{j_1}} M_{\beta^{j_2}} \dots M_{\beta^{j_k}}$, qui est le produit de $k=\varphi(n)/r$ polynômes irréductibles, car on a: $d°(\emptyset_n(X)) = k\,d°(M_{\beta^{j_1}}) \Rightarrow \varphi(n)= k.r \Rightarrow k= \varphi(n)/r$.

**Exemple 1.14.**

Soit $n$=15 et $p$=2, le corps des racines 15-èmes de l'unité est $\mathbb{K}=\mathbb{F}_{p^r}$ où $r$ est le plus petit entier non nul tel que $n$=15 divise $2^r$ -1, on trouve que $r$=4, donc $\emptyset_{15}(x)$ se décompose en produit de $\varphi(15)/r = 8/4 = 2$ polynômes irréductibles de degré $r$=4 c.à.d. $\emptyset_{15}(X) = (X^4 + aX^3 + bX^2 + cX + 1)(X^4 + a'X^3 + b'X^2 + c'X + 1)$. D'autre part $\emptyset_{15}(X)= \emptyset_{3.5}(X) = X^8 + X^7 + X^5 + X^4 + X^3 + X + 1$. Après développement et identification, on trouve : $a= b = b' = c' = 0$ et $c = a' = 1$ et donc $\emptyset_{15}(X) = (X^4 + X + 1)(X^4 + X^3 + 1)$.

**Conséquence 1.5.**

Soient $p$ premier, $n\in\mathbb{N}^*$ tel que $n\wedge p=1$ et soit $\mathbb{K}=\mathbb{F}_{p^r}$ le corps des racines nièmes de l'unité. 7 Si $\varphi(n)=r$ alors $\emptyset_n(X)$ est un polynôme irréductible.

**Preuve.**

Comme φ(*n*)=*r* alors *k*=1 et$\emptyset_n(X)$= $M_\beta$, qui est un polynôme irréductibles.

**Exemple 1.15.**

Soient *p*=3 et *n*=5. Le corps de décomposition de $X^5$ -1 sur $\mathbb{F}_3$ est $\mathbb{K}=\mathbb{F}_{3^r}$ où *r* est le plus petit entier non nul tel que *n*=5 divise $3^r$ -1, on trouve que r=4, et on a φ (5) = 4=*r*, donc $\emptyset_5(X) = X^4 + X^3 + X^2 + X + 1$ est irréductible sur $\mathbb{F}_3$.

**Exemple 1.16.**

La décomposition de $X^9$ -1 sur $\mathbb{F}_2$, donne:

$X^9 - 1 = \emptyset_1(X).\emptyset_3(X).\emptyset_9(X) = (X-1)(X^2+X+1)(X^6+X^3+1)$. L'ordre de *p*=2 dans $\mathbb{Z}/9\mathbb{Z}$ est *r*=6, donc le corps des racines $9^{iemes}$ de l'unité est $\mathbb{K}=\mathbb{F}_{2^6}$ et φ (9)=6=r, donc $\emptyset_9(X)$ est irréductible.

Le théorème ci-dessous nous permet de Décomposer le polynôme $X^n - 1$ en produit de polynômes irréductibles sur $\mathbb{F}_p$.

**Théorème 1.12.**

Soient *p* premier, $n \in \mathbb{N}^*$ tel que $n \wedge p = 1$. Alors le polynôme $X^n - 1$ se décompose en produit de polynômes irréductibles sur $\mathbb{F}_p$.

**Preuve.**

Il suffit d'appliquer la proposition 1.25 et sa Conséquence 1.3 et le Théorème 1.11 et sa Conséquence 1.5.

**Remarque 1.9.**

Si *n* n'est pas premier avec *p*, alors $n=Np^m$ avec $N \wedge p = 1$, on applique le théorème ci-dessus, en décomposant le polynôme $X^N - 1$, et on en déduit la décomposition de $X^n - 1 = (X^N - 1)^{p^m}$.

**Exemple 1.17.**

Décomposer le polynôme $X^5 - 1$ en polynômes irréductibles sur $\mathbb{F}_2$.

Soit $\mathbb{K}=\mathbb{F}_{2^r}$, le corps des racines *nièmes* de l'unité sur $\mathbb{F}_2$. *n*=5 premier avec *p*=2, le plus petit entier non nul *r* tel que *n*=5 divise $2^r$ -1 est *r*=4, donc $\mathbb{K}=\mathbb{F}_{16}$.

$X^5 - 1 = \prod_{d/5} \emptyset_d(X) = \emptyset_1(X).\emptyset_5(X) = (X-1)\,\emptyset_5(X)$. Décomposons $\emptyset_5(X)$ en produit de polynômes irréductibles sur $\mathbb{F}_2$. On a φ(5)=4=*r* et donc $\emptyset_5(X) = X^4 + X^3 + X^2 + X + 1$ est irréductible sur $\mathbb{F}_2$. Donc $X^5 - 1 = (X-1)(X^4 + X^3 + X^2 + X + 1)$ .

**Exercice 1.3.**

Décomposer les polynômes $X^{15}$ -1 , $X^9$ -1, $X^{18}$ -1 sur $\mathbb{F}_2$ et $\mathbb{F}_3$ et $\mathbb{F}_5$.

## 1.4 Matrice de permutation et ses propriétés.

### 1.4.1 Matrice de permutation

**Définition 1.20.**

Une **matrice de permutation** d'ordre *n* est une matrice carré *P* d'ordre *n* dont les colonnes (ou les lignes) sont une permutation des colonnes (ou des lignes) de la matrice identité

$$I_n=\begin{pmatrix}1&0&\dots&0\\0&1&\ddots&0\\\vdots&\vdots&&\\0&0&\dots&1\end{pmatrix}$$

Si $P = (p_{ij})_{1\le i,\, j\le n}$ ; $\exists\ \sigma\in S_n$ ($S_n$ le groupe symétrique d'indice *n*) tel que :

$$p_{ij} = \delta_{i,\sigma(j)} = \begin{cases}1, & si\ \ i=\sigma(j).\\ 0, & si\ non.\end{cases}$$

$\delta_{i,j}$ représente le **symbole de Kronecker**.

Si σ est la permutation associée à la matrice de permutation *P*, on note $P_\sigma$ au lieu de *P*.

### 1.4.2 Propriétés de la matrice de permutation

**Proposition 1.27.**

Si $P_\sigma$ la matrice de permutation associée à la permutation σ:

1. $\sigma, \tau\in S_n$ : $P_\sigma\, P_\tau = P_{\sigma o\tau}$, où (o) représente la loi de composition des applications.
2. L'ensemble des matrices de permutation d'ordre *n* noté $P_n$ forme un sous-groupe du groupe multiplicatif des matrices carrés d'ordre *n* isomorphe au groupe symétrique $S_n$. Cet isomorphisme est l'application $f : (S_n, o) \to (P_n, .),\ \ \sigma \to f(\sigma)= P_\sigma$.
3. $P_\sigma$ est une matrice inversible. Si σ est paire $\det(P_\sigma)=1$, si non $\det(P_\sigma)= -1$, et l'inverse de $P_\sigma$ est $P_\sigma^{-1}$ qui égale à ${}^tP_\sigma$ (la matrice transposée de $P_\sigma$) et si $P_\sigma$ est une matrice symétrique alors $P_\sigma^{-1}= P_\sigma$.
4. Multiplier une matrice *M* à droite par $P_\sigma$ revient à permuter les colonnes de la matrice *M* suivant la permutation σ.
5. Multiplier une matrice *M* à gauche par $P_\sigma$ revient à permuter les lignes de la matrice *M* suivant la permutation inverse $P_\sigma^{-1}= {}^tP_\sigma$.
6. Les colonnes de la matrice $P_\sigma$ sont les vecteurs de la base canonique de $\mathbb{R}^n$, dont on a modifié l'ordre. Si on note $e_1, e_2, \dots, e_n$ ces vecteurs, alors $P_\sigma(e_i)=e_{\sigma(i)}$. Ainsi $P_\sigma$ envoie une base orthonormale sur une base orthonormale, donc $P_\sigma$ est une **matrice orthogonale**.

**Exemple 1.18.**

Soit la matrice $A=\begin{pmatrix}1&0&1&1&1&0&0&0\\0&1&1&1&0&1&0&0\\1&1&0&1&0&0&1&0\\1&1&1&1&1&1&1&1\end{pmatrix}$ et la matrice de permutation $P_\sigma$ tel que $\sigma=\tau_{13}$ et soit la matrice $S=\begin{pmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{pmatrix}$ avec $S^{-1}=S$ alors :

Pour le produit $A.P_\sigma$ on permute la première et la troisième colonne de $A$ et pour le produit $S.A$ on permute la première ligne avec la quatrième et la troisième ligne avec la deuxième ligne.

On trouve $A.P_\sigma=\begin{pmatrix}1&0&1&1&1&0&0&0\\1&1&0&1&0&1&0&0\\0&1&1&1&0&0&1&0\\1&1&1&1&1&1&1&1\end{pmatrix}$ et $S.A=\begin{pmatrix}1&1&1&1&1&1&1&1\\1&1&0&1&0&0&1&0\\0&1&1&1&0&1&0&0\\1&0&1&1&1&0&0&0\end{pmatrix}$.

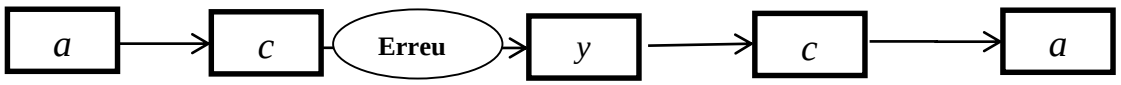
# Généralités sur les codes linéaires.

## Introduction.

La "*Théorie des codes correcteurs d'erreurs*" (T.C.C.E) appartient à un domaine plus vaste appelé "*Théorie de l'information*" du à *Claude Shannon* (fondateur), *Hamming* et *Golay*. Cette théorie est née en 1948, dont l'objet est la description et l'étude des systèmes de communication, où l'information est considérée comme une grandeur mathématique.

Un problème majeur de la transmission ou de l'enregistrement de l'information à travers un canal est celui des erreurs que peut introduire ce canal. Un message erroné (ou bruité) peut être par exemple une rayure ou de la poussière sur un C.D, une perturbation de l'appareillage, des parasites dans une ligne téléphonique, un champ magnétique dans l'espace (transmission par satellite) ... etc. Tous ces erreurs peuvent changer le message transmis c.à.d. des "0" qui seront changés en des "1" ou inversement.

Le but de la T.C.C.E est de détecter et même corriger (si possible) ces inévitable erreurs. Pour cela, on allonge le mot " $a$ " à transmettre (opération dite codage), en lui ajoutant des bits dites bits de control ou de redondance, de façon à contenir une information sur le message lui-même, pour obtenir ainsi un mot dit mot code " $c$ ", ce mot sera transmis via un canal de transmission (qui peut être une ligne téléphonique, un bus data dans un ordinateur ou un câble internet etc.…) pour avoir un mot reçu " $y$ " entaché d'erreurs, ensuite restituer "$c$" (opération de correction), en utilisant des méthodes de correction appropriés. Enfin obtenir le mot *"a"* (opération de décodage). Le schéma ci-dessous (modèle d'un système de communication) proposé par *C. Shannon* résume ces étapes :




Emetteur — Encodeur (Codage) — Canal (Bruits) — Correcteur (Correction) — Décodeur (Décodage) — Récepteur

Fig 2.1 Le schéma d'un système de communication.

Tout le problème de la T.C.C.E est la construction des codes qui détectent et corrigent le maximum d'erreurs possible, tout en allongeant les messages le moins possible (redondance minimale) et qui soient facile à décoder. Les spécialistes en ce domaine se sont alors mis à étudier de manière systématique les codes correcteurs et leurs propriétés, dans le but d'obtenir concrètement des codes aussi performant ou presque que le présidaient les résultats théoriques

de *C.* Shannon. Pour se faire, ils ont utilisé les ressources de l'algèbre tels que l'algèbre abstraite et élaborée, en liaison avec la théorie des corps finis, qui ont été utilisés pour construire des codes correcteurs très efficaces, adaptés à tel ou tel type de transmission d'information.

## 2.1 Généralités sur les codes correcteurs.

On commence ce chapitre par donner quelques concepts basiques et généralités sur les codes correcteurs définies sur un alphabet *A*.

### 2.1.1 Codes et codages.

**Définition 2.1.**

- Un **alphabet** *A* est un ensemble fini, dont les éléments sont appelés **symboles**.
  En pratique $A=\{0, 1\}$.
- Un **mot** de **longueur** un entier *n*, est un élément $x=(x_1, x_2, \ldots, x_n)$ de $A^n$, qu'on écrit comme suit : $x = x_1x_2 \ldots x_n$.
- Un **message** est la concaténation d'un nombre fini de mots.
- On appelle **code** de longueur *n*, tout sous ensemble *C* de $A^n$, et tout mot *c* de *C* est dit **mot code** de **taille** *n*.

Dans la pratique des codes correcteurs, on utilise principalement des codes dits **codes par blocs**, tel que le message à transmettre est découpé en blocs (mots) de même taille $k < n$. Chacun des blocs d'origine de taille *k* est codé séparément, tous les blocs codés doivent avoir la même taille *n*.

**Définition 2.2.**

Soient *A* un alphabet, *k*, *n* deux entiers tel que $k < n$. Un **codage par bloc,** est toute application injective $\phi$ de $A^k$ dans $A^n$ définie par:

$$\phi : A^k \longrightarrow A^n$$

$$x = (x_1, \ldots, x_k) \longmapsto c = \phi(x) = (f_1(x), \ldots, f_n(x)),$$

où $f_i : A^k \longrightarrow A$ , $i \in \{1, \ldots, n\}$ une application.

Le code *C* associé à $\phi$ est: $C=Im(\phi)$, est dit **code en bloc** de **taille** $(n, k)$ et noté $C(n, k)$, l'entier *n* est dit la **longueur** du code *C*. Les blocs codés sont transmis successivement.

### 2.1.2 Caractérisation des codes correcteurs.

Les codes correcteurs sont caractérisés par quatre données importantes :

#### 2.1.2.1 La longueur des mots de code.

Dans les codes par blocs, on coupe le message en mots de longueur *n*. On souhaite envoyer des blocs de messages relativement grands car dans ce cas, le nombre d'erreurs se rapproche de son espérance, ce qui implique que la probabilité d'erreurs est meilleure.

#### 2.1.2.2 Le rendement ou taux de codage.

Le rendement ou taux de codage correspondant au nombre de symboles d'information *k* des blocs du message divisé par le nombre de symboles des mots codes, c'est-à-dire *k/n*. Si ce taux est très petit, cela signifie que pour envoyer un petit bloc de message, il va être nécessaire de transmettre un mot de code très grand. Or ceci n'est pas souhaitable car la vitesse de transmission va être fortement réduite, et il faudra beaucoup de temps pour envoyer un court message. Le but étant d'optimiser cette vitesse de transmission, on cherchera à avoir le meilleur rendement possible.

#### 2.1.2.3 La probabilité d'erreurs du code.

Cette probabilité va dépendre de la distance minimale entre les mots de code. Il est en effet évident que plus les mots seront différents les uns des autres, plus la capacité à retrouver le message original va augmenter.

#### 2.1.2.3 La complexité de l'algorithme d'encodage et de décodage.

La complexité de ces algorithmes est une donnée importante car elle décidera du temps de calcul et des ressources nécessaires pour l'exécution des fonctions de codage et de décodage. Il s'agit donc d'une donnée à prendre en compte lors du choix d'un type de code.

### 2.1.3 Codage et code systématique.

**Définition 2.3.**

Un codage (en bloc) $\phi: A^k \longrightarrow A^n$ est dit **codage systématique**, si le mot à coder se retrouve en tête du mot codé. i.e. $x = (x_1, \dots, x_k) \longmapsto \phi(x) = (x_1, \dots, x_k, x_{k+1}, \dots, x_n)$, avec $x_i = f_i(x)$ pour tout $i \in \{k+1, \dots, n\}$.

Un codage systématique consiste juste à rajouter *n-k* symboles à la fin du mot à coder**.**

Les *k* premiers symboles $x_1, \dots, x_k$ du mot code sont les **symboles d'information**.

Les *n-k* derniers symboles ajoutés $x_{k+1}, \dots, x_n$ sont les **symboles de redondance** ou de **control**.

**Exemple 2.1.**

Code à bit de parité de taille (*k*, *k*+1) est un code binaire (i.e. $A=\{0, 1\}$) systématique où la redondance est la somme des *k* premier bits.

$\phi : A^k \longrightarrow A^{k+1}$

$x = (x_1, \dots, x_k) \longmapsto c = \phi(x) = (x_1, \dots, x_k, \sum_{i=1}^{k} x_i\ )$

Pour *k=2* : $\phi : A^2 \longrightarrow A^3$ où *A*={0, 1}

$x = (x_1, x_2) \longmapsto c = \phi(x) = (x_1, x_2, x_1 + x_2\ )$

Le code associé est *C*= {000, 101, 011, 110} dit *code à bit de parité*.

**Exemple 2.2.** *Code par répétition* de taille (1, *n*).

$\phi : A \longrightarrow A^n,\ x = x_1 \longmapsto c = \phi(x) = (x_1, \dots, x_1\ )$

### 2.1.4 Distance de Hamming et distance minimale.

**Définition 2.4.**

On appelle **distance de Hamming** entre deux mots $x = (x_1,\dots,x_n),\ y = (y_1, \dots, y_n) \in A^n$ et on note $d(x, y)$**,** le nombre de positions où $x$ et $y$ ont des symboles différents :

$d(x, y)$= *card*$\{i \in \overline{1.n}\ :\ x_i \neq y_i\}$

La distance de Hamming est une distance sur $A^n$ au sens d'espaces métriques.

**Définition 2.5.**

On appelle **distance minimale** (ou simplement **distance**) d'un code *C* et on la note $d_{\min}$ (ou tout simplement *d*) la plus petite des distances entre deux mots distincts de ce code. c.à.d.

$$d = d_{\min} = \min\{\mathrm{d}(x, y)/\ (x, y)\ \in \mathrm{C} \times \mathrm{C}, x \neq y\}.$$

**Remarque 2.1.**

Les mots du code *C* sont caractérisés par la distance *d*.

Soit $x \in A^n$, alors $x \in C$, si et seulement si, pour tout mot $y \in C$ : $d \leq d(x, y)$.

**Définition 2.6.**

Soit *C* un code sur l'alphabet *A*. On appelle **capacité de correction** (resp **détection**) du code *C*, le nombre d'erreurs *e* (resp *t*) de transmission, que ce code peut **corriger** (resp **détecter**), on dit dans ce cas que C est un code **e-correcteur** (resp **t-détecteur**).

**Théorème 2.1.**

Soit *C* un code de distance *d*, soit *y* un mot reçu, comportant au plus *k* erreurs de transmission par rapport au mot envoyé *x*.

1. Si $k \leq d-1$, le code *C* permet de détecter si le mot reçu *y* est erroné et *C* est un code $(d-1)$-détecteur.
2. Si $k \leq \left[\frac{d-1}{2}\right]$, le code *C* permet de corriger le mot reçu *y* et *C* est un code $\left[\frac{d-1}{2}\right]$-correcteur.

**Preuve.**

1. Cas $k \leq d\text{-}1 < d$ **:** On suppose qu'il y a au moins une erreur de transmission. Le mot reçu est détecté comme erroné, si ce n'est pas un mot du code. On a par hypothèse $d(x,y) \leq k$. Supposons par absurde que $y \in C$. $d(x,y) \leq k$ et $k < d$ donc $d(x,y) < d$ et cela contredit la définition de d et donc *y* ne peut pas être un mot du code et donc il est détecter comme erroné.
2. Cas $k \leq \left[\frac{d-1}{2}\right]$**.**

   Pour tout mot du code *z* différent de *x*, on a par définition $d \leq d(x,z)$. L'inégalité triangulaire donne: $d(x,z) \leq d(x,y) + d(y,z)$. Or on a par hypothèse:

$d(x,y) \leq k$ *et* $k \leq \left[\frac{d-1}{2}\right] \leq \frac{d-1}{2} < \frac{d}{2}$, donc $(x,y) < \frac{d}{2}$, d'où $d < \frac{d}{2} + d(y,z)$,

soit $\frac{d}{2} < d(y,z)$, comme $d(x,y) < \frac{d}{2}$ **,** on en déduit que pour tout mot z du code différent de $x$, $d(x,y) < d(z,y)$. Le mot reçu *y* est bien corrigé en *x*, car *x* est le mot du code le plus proche de *y*. □

**Exemple 2.3.**

1. Soit le code $C$ = {011100, 111011, 111111, 110001,011100}, $d$ = min {3, 5,6} = 3.

Ce code peut corriger jusqu'à $e = \left[\frac{d-1}{2}\right]$=1 erreur et il peut détecter jusqu'à $t=d\text{-}1$=2 erreurs.

2. Soit le code par répétition $C$={00000,11111}, $d$=5.

Il peut détecter jusqu'à $t$=4 erreurs (4-détecteur) et corriger jusqu'à $e$=2 erreurs (2-correcteur)

### 2.1.5 Codes équivalents.

**Définition 2.7.**

Soit *A* un alphabet quelconque et *n* un entier tel que $n \geq 1$, pour chaque permutation $\sigma \in S_n$ (où $S_n$ est le **groupe symétrique** d'indice *n*). On définit l'application $\bar{\sigma}$ de $A^n$ dans $A^n$ par :

$$\bar{\sigma}: (x_1,\ldots,x_n) \longmapsto (x_{\sigma(1)},\ldots,x_{\sigma(n)})$$

Deux codes *C, C'* de longueur *n* sur l'alphabet *A* sont dits **équivalents** s'il existe une permutation $\sigma \in S_n$tel que $C'=\bar{\sigma}(C)$.

**Proposition 2.1.**

Deux codes équivalents sur l'alphabet *A* ont les mêmes paramètres i.e la même distance, la même longueur, le même cardinal et même capacité de correction.

**Théorème 2.2.**

Tout code en bloc $C(n, k)$ est équivalent à un code systématique.

**Preuve** : Si le code *C* est systématique alors il est équivalent à lui-même. Supposons que *C* n'est pas un code systématique**.** En choisissant une permutation $\sigma \in S_n$ de telle sorte que $\bar{\sigma}: A^n \longrightarrow A^n$, $(x_1, \dots, x_n) \mapsto \bar{\sigma}(x_1, \dots, x_n) = (x_1, \dots, x_k, x_{\sigma(k+1)}, \dots, x_{\sigma(n)})$ on obtient ainsi un code systématique *C'* qui est l'image de *C* par l'application $\bar{\sigma}$ et qui est donc équivalent à *C*. □

## 2.2 Codes et codages linéaires.

Si l'alphabet *A*= $\mathbb{K}$ est un **corps de Galois** (corps fini) de cardinal *q* noté $\mathbb{F}_q$ tel que $q= p^r$ où *p* entier premier représentant la caractéristique de $\mathbb{K}$ , alors $\mathbb{K}^n$ est un $\mathbb{K}$-espace vectoriel de dimension *n* pour les lois habituelles (l'addition et la multiplication par un scalaire), dans la pratique, on prend $\mathbb{K} = \mathbb{F}_2 = \{0,1\}$.

Dans tous le chapitre $\mathbb{K} = \mathbb{F}_q$ tel que $q= p^r$, est un corps de Galois.

### 2.2.1 Définitions et propriétés.

**Définition 2.8.**

On appelle **code linéaire** (ou **code q-aire**) *C* de **longueur** *n* et de **dimension** *k* sur $\mathbb{K}$, tout sous-espace vectoriel du $\mathbb{K}$-espace vectoriel $\mathbb{K}^n$, de dimension k. Si la distance de *C* est *d*, on le note *C*(*n, k, d*) où *n, k, d* sont dits **paramètres** du code *C*.

**Remarque 2.2.**

1- *C* est un code linéaire, si et seulement si, pour tous $x_1, x_2 \in C$, $\alpha_1, \alpha_2 \in \mathbb{K}$, $\alpha_1 x_1 + \alpha_2 x_2 \in C$

2- Une application $\phi: \mathbb{K}^k \longrightarrow \mathbb{K}^n$ sur l'alphabet $\mathbb{K}$ est dite *codage linéaire*, si et seulement si l'application $\phi$ est une application linéaire injective.

**Exemples 2.4.**

1- $\{0\}$ et $\mathbb{K}^n$ sont des codes linéaires dits **codes triviaux**.

2- $\phi: A^k \longrightarrow A^{k+1}$ **codage par bit de parité**.

$x = (x_1, \dots, x_k) \longmapsto \phi(x) = (x_1, \dots, x_k, \sum_{i=1}^{k} x_i)$ est un codage linéaire car l'application $\phi$ est linéaire.

3- $\phi : A \longrightarrow A^n$ **codage à répétition.**

$x = x_1 \longmapsto c = \phi(x) = (x_1, \dots, x_1)$ est un codage linéaire.

4- Le code $\mathbb{K} = \mathbb{F}_2$

$C=\{(x_1, x_2, x_3, x_4, x_1 + x_2 + x_3,\ x_1 + x_2 + x_4, x_2 + x_3 + x_4) / (x_1, x_2, x_3, x_4) \in {F_2}^4\}$

est un code linéaire, car c'est un sous-espace vectoriel de $\mathbb{F}_2^4$, engendré par la base:

B =$\{L_1=(1, 0, 0, 0, 1, 1, 0), L_2=(0, 1, 0, 0, 1, 1, 1), L_3=(0, 0, 1, 0, 1, 0, 1), L_4=(0, 0, 0, 1, 0, 1, 1)\}$.

**Définition 2.9.**

Le **poids** d'un élément $x = (x_1, \dots, x_n) \in \mathbb{K}^n$ noté $\boldsymbol{w(x)}$, est le nombre de ses composantes non nulles. $w(x) = card\{i \in \{1,2, \dots, n\}: x_i \neq 0\}$.

**Exemple 2.5.**

Soit l'alphabet A =$\mathbb{F}_2$ et $x = (1,0,1,0) \in \mathbb{F}_2^4$, $w(x) = 2$

**Propriétés 2.1.**

Pour tous $x, y \in \mathbb{K}^n, \lambda \in \mathbb{K}$ on a:

1. $d(x, y) = w(x - y)$,
2. $w(x) = d(x, 0)$,
3. $w(x) = 0 \Leftrightarrow x = 0$,
4. $w(\lambda x) = w(x), \lambda \neq 0$,
5. $w(x + y) \leq w(x) + w(y)$.

**Définition 2.10.**

On appelle **poids minimum** d'un code linéaire *C*(*n, k*), le plus petit poids des mots non nul du code *C* et on le not $P_{min}$ ou simplement *P*.

**Exemple 2.6.**

*C*= {00000, 01111, 11000,10111} est un code linéaire de longueur 5 et de dimension 2 sur $\mathbb{F}_2$ et $P_{min}(C) = min\{4,2\} = 2$.

**Proposition 2.2.**

Si *C* est un code linéaire, l'ensemble des distances entre mots de *C* est égale à l'ensemble des poids des mots de *C*.

**Preuve.** *D*= {d(*x, y*) / *x, y*∈ *C*}= {w(*x-y*) / *x, y*∈ *C*}= {w(*z*) / *z= x-y*∈ *C*}, *z* ∈ *C* car *C* est un sous espace vectoriel de $\mathbb{K}$. □

**Conséquence 2.1.**

La distance minimale d'un code linéaire est égale au poids minimum des mots non nuls du code, $P_{min} = d_{min}$.

Chercher la distance minimale d'un code en calculant le poids minimum des mots non nul est plus facile (car moins long) que chercher la plus petite distance entre tous les mots du code deux à deux distincts.

### 2.2.2 Matrice génératrice.

**Définition 2.11.**

Une **matrice génératrice** d'un code linéaire $C(n, k)$ sur le corps fini $\mathbb{K}$ est une matrice de type $k \times n$ à coefficients dans $\mathbb{K}$, dont les lignes forment une base de $C$.

**Exemple 2.7.**

$G=\begin{pmatrix}1 & 0 & 1 & 1 & 1\\ 0 & 1 & 1 & 1 & 1\end{pmatrix}$ est une matrice génératrice du code linéaire C(5,2)

$C$= {00000, 01111, 11000,10111} sur $\mathbb{F}_2$. Car les vecteurs lignes $L_1$=10111 et $L_2$ = 01111 sont libres est donc forment une base de $C$ car on sait que la dimension de $C$ est $k$=2.

**Proposition 2.3.**

Si $G$ est une matrice génératrice d'un code linéaire $C(n, k)$ sur $\mathbb{K}$, alors :

Toute matrice génératrice de $C$ est de la forme $A\times G$, ou $A$ est une matrice carrée inversible d'ordre $k$ sur $\mathbb{K}$ (ou encore $A$ est une matrice carrée de rang $k$ sur $\mathbb{K}$).

**Exemple 2.8.**

$G=\begin{pmatrix}1 & 0 & 1 & 1 & 1\\ 0 & 1 & 1 & 1 & 1\end{pmatrix}$ une matrice génératrice du code $C$ sur le corps $\mathbb{F}_2$ dans l'exemple précèdent alors les matrices génératrices possibles de $C$ sont:

$G_0=I_2.G=G$, $G_1=\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}.\begin{pmatrix}1 & 0 & 1 & 1 & 1\\ 0 & 1 & 1 & 1 & 1\end{pmatrix}=\begin{pmatrix}1 & 1 & 0 & 0 & 0\\ 0 & 1 & 1 & 1 & 1\end{pmatrix}$,

$G_2=\begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}.\begin{pmatrix}1 & 0 & 1 & 1 & 1\\ 0 & 1 & 1 & 1 & 1\end{pmatrix}=\begin{pmatrix}0 & 1 & 1 & 1 & 1\\ 1 & 1 & 0 & 0 & 0\end{pmatrix}$, $G_3=\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}.G=\begin{pmatrix}1 & 0 & 1 & 1 & 1\\ 1 & 1 & 0 & 0 & 0\end{pmatrix}$,

$G_4=\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}.G=\begin{pmatrix}1 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 1 & 1\end{pmatrix}$ et $G_5=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}.G=\begin{pmatrix}0 & 1 & 1 & 1 & 1\\ 1 & 0 & 1 & 1 & 1\end{pmatrix}$.

### 2.2.3 Construction d'un code linéaire.

La proposition ci-dessous montre comment construire un code linéaire en utilisant sa matrice génératrice.

**Proposition 2.4.**

Si $G$ est la matrice génératrice d'un code linéaire $C(n, k)$ sur $\mathbb{K}$ alors, Le code $C$ est le sous-espace de $\mathbb{K}^n$ des mots $c$ de la forme : $c= x.G$, avec $x=(x_1, \dots, x_k)\in\mathbb{K}^k$.

**Remarque 2.3.**

Si $G$ est une matrice génératrice d'un code linéaire $C(n, k)$ sur $\mathbb{K}$, alors les mots de $C$ sont toutes les combinaisons linéaires des lignes $L_1, L_2, \dots, L_n$ de $G$.

$i.e$: pour tout $c \in C : c = \sum_{i=1}^{k} \alpha_i L_i$ / $\alpha_i \in \mathbb{K}$.

**Exemple 2.9.**

Soit *C* un code linéaire de type (2, 5) de matrice génératrice $G=\begin{pmatrix}1 & 0 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 0\end{pmatrix}$, alors le code linéaire associé est:

$C = \{(x_1, x_2).G / x_1, x_2 \in \{0,1\}\} = \{(x_1+x_2, x_2, x_1+x_2, x_1+x_2, x_1) / x_1, x_2 \in \{0,1\}\}$

d'où *C*={00000, 10111, 11110, 01001}.

**Exemple 2.10.**

Considérons la matrice :$G = \begin{pmatrix}1 & 1 & 1 & 1 & 0\\ 0 & 1 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 1\end{pmatrix}$

La matrice G est de rang 3, en effet les trois premières colonnes de G sont linéairement indépendantes, c'est la matrice génératrice d'un code linéaire C(5,3) sur $\mathbb{F}_2$={0,1} engendré par $L_1,\ L_2,\ L_3$ et donc ce code linéaire est le suivant:

$C = \{0,\ L_1,\ L_2,\ L_3\ ,\ L_1+L_2,\ L_1+L_3,\ L_2+L_3,\ L_1+L_2+L_3\ \}$, donc

C={ 00000, 11110,01011,00101,10101,11011,01110,10000}.

### 2.2.4 Matrice génératrice normalisé et code linéaire systématique.

Dans ce qui suit $\mathbb{K}$ est un corps fini de cardinal *q*.

**Définition 2.12.**

Une matrice génératrice $G_N$ d'un code linéaire $\mathrm{C}(n,k)$ sur le corps $\mathbb{K}$, est dite **normalisée** (ou **standard**), si la matrice formée par ses *k* premières colonnes est la matrice unité $I_k$.

Donc *G* est de la forme $G=(I_k|M)$ tel que $M \in M_{k,n-k}(\mathbb{K})$ est une matrice dite *de* **redondance**.

**Exemples 2.11.**

**1.** Soit *C* un code linéaire (5,2) sur $\mathbb{F}_2$ de matrice génératrice $G=\begin{pmatrix}1 & 0 & 1 & 1 & 1\\ 0 & 1 & 1 & 1 & 1\end{pmatrix}$. *G* est une matrice génératrice normalisée du code *C* et *C*={00000, 01111, 11000, 10111}.

**2.** $G=\begin{pmatrix}1 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 1\end{pmatrix}$ est une matrice génératrice normalisée d'un code linéaire *C*(5,3) sur $\mathbb{F}_2$ dont les mots codes sont :

*C*={10010, 01011, 0010, 11001, 10111, 01110, 11100, 00000}

**Remarque 2.4.**

Certains codes linéaires n'admettent pas de matrices génératrices standard, par exemple le code: *C*= {000, 001, 010,011} admet comme matrices génératrices de *C* les matrices,

$G_1=\begin{pmatrix}0&0&1\\0&1&0\end{pmatrix}$, $G_2=\begin{pmatrix}0&1&0\\0&0&1\end{pmatrix}$, $G_2=\begin{pmatrix}0&0&1\\0&1&1\end{pmatrix}$ $G_3=\begin{pmatrix}0&1&1\\0&0&1\end{pmatrix}$, $G_4=\begin{pmatrix}0&1&0\\0&1&1\end{pmatrix}$ et $G_5=\begin{pmatrix}0&1&1\\0&1&0\end{pmatrix}$. Donc $C$ n'admet pas de matrice génératrice normalisée.

**Définition 2.13.**

Un code linéaire $\mathrm{C}(n,k)$ est dit **systématique** (ou **standard**), s'il possède une matrice génératrice normalisée $G_N$ de la forme $G_N=(I_k|M)$ tel que $M \in M_{k,n-k}(\mathbb{K})$.

**Remarque 2.5.**

1. Quelques mathématiciens définissent la matrice génératrice normalisée par:

   $G_N = (M \mid I_k)$ tel que $M \in M_{k,n-k}(\mathbb{K})$.

2. Si $C$ est un code systématique linéaire de matrice génératrice normalisé $G_N=(I_k|M)$ alors $C=\{x.G_N / x\in \mathbb{K}^k\}=\{(x, x.M) / x\in \mathbb{K}^k\}$.

**Exemple 2.12.**

Considérons le code linéaire binaire de matrice génératrice normalisée:

$G=\begin{pmatrix}1&1&1&1&0\\0&1&0&1&1\\0&0&1&0&1\end{pmatrix}$ de la forme $G=A|B)$. Comme la matrice $A=\begin{pmatrix}1&1&1\\0&1&0\\0&0&1\end{pmatrix}$ est inversible alors $C$ est systématique de matrice génératrice normalisée, $G_N=\begin{pmatrix}1&0&0&0&0\\0&1&0&1&1\\0&0&1&0&1\end{pmatrix}$ de la forme $G_N=(I_3|M)$ avec $M=\begin{pmatrix}0&0\\1&1\\0&1\end{pmatrix}$ et donc le code.

$C=\{x.G_N / x\in \mathbb{F}_2^3\}=\{(x, x.M) / x=(x_1, x_2, x_3)\in \mathbb{F}_2^3\}=\{(x_1, x_2, x_3, x_2, x_2+x_3) / x_i \in \mathbb{F}_2\}$

Donc C = { 00000, 11110,01011,00101,10101,11011,01110,10000}.

**Proposition 2.5.**

Si elle existe, la matrice génératrice normalisée d'un code linéaire systématique $C$ est unique. On l'obtient en appliquant l'algorithme de GAUSS sur une matrice génératrice quelconque de $C$.

**Preuve.**

Soit $G=(A \mid B)$ ou $A$ est une matrice carré de rang k ( donc inversible), en appliquant l'Algorithme de Gauss sur les lignes de $G$ (donc de $A$) pour avoir la matrice unité $I_k$, alors on obtient une matrice $G_N$ ( équivalente à $G$) de la forme $G_N=(I_k \mid B')$ qui est une matrice normalisée de $C$. □

**Exemple 2.14.**

Le code binaire $C$ suivant de longueur $n$=7 et de dimension $k$=4, définit par:

$C = \{( x_1, x_2, x_3, x_4, x_2+x_3+x_4, x_1+x_3+x_4, x_1+x_2+x_4) / x_1, x_2, x_3, x_4 \in \{0,1\}\}$

est un code linéaire de base B={$L_1$=1000011, $L_2$=0100101, $L_3$=0010110, $L_4$=0001111} qui admet la matrice génératrice suivante:

$G_N = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$ comme matrice standard, ce code est de distance minimale d=3? à vérifier. $C$ est donc un code systématique.

**Exemple 2.15.**

Soit $C$ un code linéaire définit par sa matrice génératrice

$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$ de la forme $G$=($A$, $B$) où $A$= $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ inversible et donc $C$ est un code systématique, on applique l'algorithme de GAUSS à $G$, pour passer de $G$=($A$, $B$) à $G_N$=($I_3$, $B'$).

On a : $G$=$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$ $L_{2\leftarrow}L_2+L_1$ $G \sim$ $G_1$=$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$

$G_1 \sim G_2$ =$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$ $L_{1\leftarrow}L_1+L_3$, $G_2 \sim G_3$=$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$ $L_{1\leftarrow}L_2+L_1$,

$G_3 \sim G_N$=$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$, qui est la matrice génératrice normalisée de $C$.

Et le code est $C = \{( x_1, x_2, x_3, x_2, x_1+x_3, x_1+x_2+x_3) / x_1, x_2, x_3 \in \{0,1\}\}$.

Donc C = { 000000, 100011, 010101, 001011, 110110, 101000, 011110, 111101}.

**Remarque 2.6.**

1. Si le code linéaire $C(n, k)$ est systématique, alors pour chaque mot $x$= ($x_1, x_2, \dots, x_k$) de $\mathbb{K}^k$, il existe un mot c est un seul de $C$ de la forme : $c$ =($x_1, x_2, \dots, x_k, x_{k+1}, \dots, x_n$).
2. Si la matrice génératrice $G$ est de la forme G= ($A$ | $B$) où $A$ est inversible, alors le code associé est systématique de matrice génératrice normalisée $G_N$=($I_k$ | $A^{-1}B$).
3. Si la matrice génératrice $G$ est de la forme $G$= ($A$ | $B$) où $A$ n'est pas inversible, alors le code associé ne peut être systématique, mais équivalent à un code systématique.

**Théorème 2.2.**

Tout code linéaire $C(n, k)$ est équivalent à un code linéaire systématique.

**Preuve.**

Supposons que $C$ ne soit pas un code systématique. Soit $G$ une matrice génératrice du code $C$. Comme le rang de $G$ est égal à $k$, il existe un mineur $k \times k$ non nul. Par une permutation des colonnes de $G$, on amène ce mineur aux $k$ premières colonnes, on obtient ainsi une matrice génératrice d'un code linéaire systématique $C$' équivalent à $C$. □

**Exemple 2.16.**

Soit le code linéaire $C(3, 2)$ de la Remarque 2.4, $C$= {000, 001, 010,011} qui n'est pas systématique (car il n'admet pas de matrice standard), et soit la matrice génératrice $G=\begin{pmatrix}0 & 1 & 0\\ 0 & 1 & 1\end{pmatrix}$ en faisant les permutations sur les colonnes $C_1 \leftrightarrow C_2$ et $C_2 \leftrightarrow C_3$ on obtient la matrice $G_1=\begin{pmatrix}1 & 0 & 0\\ 1 & 1 & 0\end{pmatrix}$, et en appliquant l'algorithme de Gauss sur les lignes de $G_1$ ( $L_{2\leftarrow}L_{2+}L_1$) on obtient la matrice $G_N=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}$ qui est la matrice génératrice normalisée d'un code systématique C'= {000, 100, 010,110} équivalent à C.

**Exemple 2.17.**

Soit $\mathbb{K} = \{0,1\}$ et $G$ la matrice binaire de type (6, 4) suivante:

$G = \begin{pmatrix}1 & 0 & 1 & 1 & 0 & 1\\ 1 & 1 & 0 & 1 & 0 & 1\\ 0 & 1 & 1 & 1 & 0 & 1\\ 1 & 1 & 0 & 1 & 1 & 0\end{pmatrix}$. Le premier mineur $4 \times 4$ : $\begin{vmatrix}1 & 0 & 1 & 1\\ 1 & 1 & 0 & 1\\ 0 & 1 & 1 & 1\\ 1 & 1 & 0 & 1\end{vmatrix}$ est nul car la somme des deux premières colonnes est égale à la troisième. Par contre : $\begin{vmatrix}1 & 0 & 1 & 0\\ 1 & 1 & 1 & 0\\ 0 & 1 & 1 & 0\\ 1 & 1 & 1 & 1\end{vmatrix} \neq 0$, on en déduit que rg($G$)=4, donc $G$ est la matrice d'un code linéaire binaire $C(6,4,\mathrm{d})$ mais à cause de la remarque du début, $C$ n'est pas systématique, on permute les colonnes ( pour amener le mineur non nul aux 4 premières colonnes) par $\pi=\begin{pmatrix}1 & 2 & 3 & 4 & 5 & 6\\ 1 & 2 & 4 & 5 & 3 & 6\end{pmatrix}$, pour avoir

$G'=\begin{pmatrix}1 & 0 & 1 & 0 & 1 & 1\\ 1 & 1 & 1 & 0 & 0 & 1\\ 0 & 1 & 1 & 0 & 1 & 1\\ 1 & 1 & 1 & 1 & 0 & 0\end{pmatrix}$. Puis on applique l'algorithme de Gauss sur les lignes de $G$' et on obtient $G_N=\begin{pmatrix}1 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 0 & 0 & 1\\ 0 & 0 & 0 & 1 & 0 & 1\end{pmatrix}$ qui est la matrice génératrice normalisée d'un code systématique équivalent à $C$.

### 2.2.5 Bornes sur la distance minimale

Soit $C(n, k, d)$ un code linéaire sur un corps fini $\mathbb{K}$ de cardinal $q$. Les grandeurs $q^k$ et d "jouent" l'une contre l'autre.

En effet, si on a un très grand nombre de mots (i.e ; $k$ très grand), alors on aura une distance d faible. Alors que si on a $k$ est petit, alors la distance $d$ sera grande, permettant ainsi de détecter et de corriger plus d'erreurs.

Il va donc être nécessaire de trouver un compromis entre ces deux valeurs, afin d'avoir un nombre de mots suffisant et une distance minimale suffisamment grande pour pouvoir détecter et corriger un certain nombre d'erreurs. Pour cela, il existe des bornes qui caractérisent les grandeurs $k$ et $d$. Parmi ces bornes on trouve la borne de singleton, de Griesmer et de Hamming.

#### 2.2.5.1 Borne de Singleton et codes M.D.S.

**Théorème 2.3.**

Si d est la distance minimale d'un code linéaire $\mathrm{C}(n, k)$ alors :

$d \leq n - k + 1$. Cette borne est appelée *borne de singleton* du code $C$.

**Preuve.**

Soit le sous espace vectoriel $D$={ $x$= $(x_1, x_2, \dots, x_n) \in \mathbb{K}^n$ : pour tout $i \geq$ d : $x_i$=0}

$D$={ $x$= $(x_1, \dots, x_{d-1}, 0 \dots, 0) \in \mathbb{K}^n$}, alors dim($D$)=$d$-1et pour tout $x \in D$-{0}: w($x$)<d, donc $x \notin C$ et $C \cap D$={0}. On a dim($C$+$D$)=dim($C$) + dim($D$)-dim($C \cap D$)=$k$-($d$-1) et comme $C+D \subset \mathbb{K}^n$, donc dim($C \cap D$) ≤ $n$, d'où $d \leq n$-$k$+1.

Cette borne permet de trouver une borne maximale sur la distance minimale $d$ par rapport aux valeurs $n$ et $k$. □

**Définition 2.14.**

Un code linéaire $\mathrm{C}(n, k)$ est dit **code M.D.S** (maximum distance séparable) si sa distance minimal $d$ atteint la borne de singleton i.e. : $d = n - k + 1$.

**Exemple 2.18.**

Le code à répétition $C$($n$, $k$=1, d=$n$) est un code linéaire de matrice génératrice $G$=(1 1 … 1 1), est un code M.D.S car $n$-$k$+1= $n$ = $d$.

**Exemple 2.19.**

Pour le code à bit de parité $C$($n$, $k$=$n$-1)

$\phi$: $A^{n-1} \longrightarrow A^n$

$x = (x_1, \dots, x_k) \longmapsto \phi(x) = (x_1, \dots, x_k, \sum_{i=1}^{k} x_i)$. $C$ est un code M.D.S? A vérifier.

**Exemple 2.20.**

Soit $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$ une matrice génératrice normalisée d'un code linéaire $C(n=5, k=3)$ sur $\mathbb{F}_2$: Est-ce que le code $C$ est M.D.S?

### 2.2.5.2 Borne de Hamming ou d'empilement de sphères et codes code parfaits.

**Définition 2.15.**

Soit $\mathrm{C}(n, k, d)$ est un code linéaire sur un corps fini $\mathbb{K}$ tel que card($\mathbb{K}$)=$q$ et $r \in \mathbb{N}^*$. Pour tout $x \in \mathbb{K}^{\boldsymbol{n}}$, on définit $\mathrm{B}(x, r)=\{y \in \mathbb{K}^{\boldsymbol{n}} : \mathrm{d}(x, y) \leq r\}$, *la* **boule** de centre x et *de* **rayon** $r$ et $\mathrm{S}(x, i) = \{y \in \mathbb{K}^{\boldsymbol{n}} : d(x, y) = i\}$, *la* **sphère** de centre $x$ et de **rayon** $i$.

**Lemme 2.1.**

Pour tout $i \leq n$: pour tout $y \in \mathbb{K}^{\boldsymbol{n}}$ **:** $\mathrm{card}(\mathrm{B}(x, r)) = \sum_{i=0}^{i=r} C_n^i\,(q-1)^i$.

**Preuve.**

On a $\mathrm{B}(x, r)=\{y \in \mathbb{K}^{\boldsymbol{n}} : \mathrm{d}(x, y) \leq r\}$. Comme la distance $d$ est un entier alors on peut écrire $\mathrm{B}(x, r)= \bigcup_{i=0}^{i=r} \mathrm{S}(x, i)$, tel que les $\mathrm{S}(x, i)$ sont disjointes deux à deux et donc, $\mathrm{card}(\mathrm{B}(x, r)) = \sum_{i=0}^{i=r} card(\mathrm{S}(x, i))$. Pour $x$ et $i$ fixés, calculons $\mathrm{card}(\mathrm{S}(x, i))$ qui est le nombre des éléments $y \in \mathbb{K}^{\boldsymbol{n}}$ : dont le nombre de composantes distinctes de celles de $x$ est égale à $i$. Comme les mots sont de longueur $n$, il y a donc $\mathrm{C}_n^i$ ensembles d'indices à $i$ éléments possibles. Chaque composante a ($q$-1) possibilités et donc $\mathrm{card}(\mathrm{S}(x, i))= C_n^i(q-1)^i$.

D'où $\mathrm{card}(\mathrm{B}(x, r)) =\sum_{i=0}^{i=r} card(\mathrm{S}(x, i))= \sum_{i=0}^{i=r} C_n^i\,(q-1)^i$. □

**Théorème 2.4.**

Si $\mathrm{C}(n, k, d)$ est un code linéaire sur un corps fini $K$ tel que card($\mathbb{K}$)=$q$ et sa capacité de correction $e=[\frac{d-1}{2}]$ alors, $q^{n-k} \geq \sum_{i=0}^{i=e} C_n^i\,(q-1)^i$. Cette borne est dite **borne d'empilement de sphères** *(ou* **borne de Hamming***).*

**Preuve.**

On sait que dans code $C$ qui vérifie la condition de décodage d'ordre $e$ (i.e. $e$-correcteur) toutes les boules $\mathrm{B}(x, e) / x \in C$ (de rayon $e$ centrées en les mots de $C$) sont disjointes deux à deux et $\bigcup_{x \in \mathrm{C}} \mathrm{B}(x, e) \subset \mathbb{K}^{\boldsymbol{n}}$ et donc $\sum_{x \in \mathrm{C}} card(\mathrm{B}(x, e)) \leq q^n$. Or d'après le Lemme ci-dessus, pour chaque $x$ de C, $\mathrm{card}(\mathrm{B}(x, e)) = \sum_{i=0}^{i=e} C_n^i\,(q-1)^i$ donc,

$\sum_{x \in \mathrm{C}} card(\mathrm{B}(x, e))=\sum_{x \in \mathrm{C}} \sum_{i=0}^{i=e} C_n^i\,(q-1)^i$ et comme $\sum_{i=0}^{i=e} C_n^i\,(q-1)^i$ ne depond pas de $x$,

alors $\sum_{x\in C} card(B(x,e))= q^k \sum_{i=0}^{i=e} C_n^i (q-1)^i$, donc on trouve $q^k \sum_{i=0}^{i=e} C_n^i (q-1)^i \leq q^n$ et d'où $q^{n-k} \geq \sum_{i=0}^{i=e} C_n^i (q-1)^i$. □

**Remarque 2.7.**

Dans le cas d'un code binaire l'inégalité du théorème ci-dessus, sachant que la capacité $e=[\frac{d-1}{2}]$, s'écrit comme suit: $\sum_{i=0}^{i=e} C_n^i \leq 2^{n-k}$ .

La borne de Hamming nous permet de donner une autre définition d'un code parfait.

**Définition 2.16.**

Un code linéaire $C(n, k, d)$ sur un corps fini $\mathbb{K}$ de cardinal $q$ est dit **code parfait**, si sa borne de Hamming est atteinte, c'est-à-dire si : $\sum_{i=0}^{\left\lfloor \frac{d-1}{2} \right\rfloor} \binom{n}{i} (q-1)^i = q^{n-k}$.

## 2.3 Code orthogonal d'un code linéaires

Dans un code *e*-correcteur, pour décoder il suffit de comparer le mot reçu *y* à tous les mots du code, puisque d'après la condition de décodage d'ordre *e*, il est possible de trouver un mot code au plus à une distance fixe ne dépassant pas *e*, c'est le principe dit "**principe de maximum de vraisemblance voisin"**. Cependant si la taille des paramètres est grande alors cette méthode devient impraticable, car il faut calculer tous les distances du mot reçu à chacun des mots du code, par exemple pour le code de Reed-Solomon (*n*=255, *k*= 223), il faut calculer $8^{223}$ distances !

Il existe plusieurs méthodes de décodage des codes linéaires, certaines sont d'ordre générale et d'autres sont spécifique à certains codes comme par exemple les codes de Goppa ou de Reed-Muller et cela selon la structure algébrique de ces codes.

### 2.3.1 Code orthogonal et Matrice de contrôle.

**Définition 2.17.**

Soit $C(n, k)$ un code linéaire sur un corps fini $\mathbb{K}$. Le **code orthogonal** (ou **dual**) de *C*, noté $C^{\perp}$ est le sous-espace vectoriel orthogonal de *C* pour le produit scalaire usuel de $\mathbb{K}^n$, c-à-d. $C^{\perp} = \{x \in \mathbb{K}^n, \text{pour tout } y \in C : < x, y >= 0\}$.

**Proposition 2.6.**

Le code dual du code $C(n, k)$ est un code linéaire de longueur *n* et de dimension $k' = n - k$.

**Définition 2.18.**

On appelle **matrice de contrôle** (ou **de parité**) d'un code linéaire $C(n, k)$ sur un corps fini $\mathbb{K}$, toute matrice génératrice $H$ de son code dual $C^{\perp}$.

**Remarque 2.8.**

Toute matrice de contrôle $H$ d'un code $C$ est de type $(n, n-k)$ et de rang $n-k$.

Le théorème ci-dessous, nous permet de tester si un mot appartient ou non à un code linéaire en utilisant une de ses matrices de contrôle $H$.

**Théorème 2.5.**

Soit $H$ une matrice de contrôle d'un code linéaire $C(n, k)$ et $c= (c_1, c_2, \ldots, c_n) \in \mathbb{K}^n$ alors :

$c \in C$ si, et seulement, si $c.{}^tH=0$.

**Exemple 2.21.**

Dans l'exemple ci-dessus le mot $m_1=00110 \in C$ car

$m_1.{}^tH=0 = 00110 . \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} = 00$, par contre le mot $m_2= 10110 \notin C$ car

$m_2.{}^tH = (10110) . \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} = 10 \neq 00.$

Le théorème ci-dessous nous donne la condition nécessaire et suffisante pour qu'une matrice soit une matrice de contrôle d'un code linéaire connaissant sa matrice génératrice.

**Théorème 2.6.**

Soit $G$ une matrice génératrice d'un code linéaire $C(n, k)$ et $H \in M_{n-k,n}(\mathbb{K})$ de rang $n-k$, alors $H$ est une matrice de contrôle de $C$, si est seulement si, $H.{}^tG = 0$.

### 2.3.2 Construction d'une matrice de contrôle à partir d'une matrice génératrice.

Donnons un code systématique $C(n, k)$ de matrice génératrice $G_N$, on peut en construire une matrice de contrôle $H$ et vice versa.

**Théorème 2.7.**

1. Soient $C(n, k)$ un code linéaire systématique $C$, $G_N$ sa matrice génératrice normalisée, tel que $G = (I_k \mid M)$ alors $H = (-{}^tM \mid I_{n-k})$ est une matrice de contrôle de $C$.

2. Réciproquement, si $C(n, k)$ est un code linéaire et $H$ est une matrice de contrôle de $C$ de la forme $H = (A \mid I_{n-k})$, alors $C$ est un code systématique et sa matrice génératrice normalisée est donnée par: $G = (I_k \mid -{}^tA)$.

**Exemple 2.22.**

Soit $G=\begin{pmatrix}1&1&1&0&0\\1&1&0&1&0\\1&0&0&0&1\end{pmatrix}$ une matrice génératrice d'un code linéaire binaire $C(5, 3)$ , qui est de la forme $G=(A/B)$ avec $A=\begin{pmatrix}1&1&1\\1&1&0\\1&0&0\end{pmatrix}$ inversible donc $C$ est systématique avec matrice génératrice $G_N=\begin{pmatrix}1&0&0&0&1\\0&1&0&0&1\\0&0&1&0&0\end{pmatrix}$. Et donc $C$ admet comme matrice de contrôle la matrice $H=\begin{pmatrix}0&0&0&1&0\\1&1&0&0&1\end{pmatrix}$.

**Exemple 2.23.**

Soit $C(6, 2)$ un code linéaire ternaire c.à.d. sur le corps fini $\mathbb{K}=\mathbb{F}_3$ qui admet comme matrice de contrôle $H=\begin{pmatrix}2&1&0&2&1&0\\0&2&1&0&2&1\end{pmatrix}$.

Montrer que ce code est systématique et trouver sa matrice génératrice normalisée $G_N$.

$H$ est de la forme $H=(M/N)$ tel que $N=\begin{pmatrix}1&0\\2&1\end{pmatrix}$ est inversible et en faisant la transformation sur la deuxième ligne $L_2=L_2+L_1$ on obtient une autre matrice de contrôle $H'=\begin{pmatrix}2&1&0&2&1&0\\2&0&1&2&0&1\end{pmatrix}$ de la forme $(A/I_2)$ et donc d'après *ii*) du théorème précédent $C$ est systématique et admet comme matrice génératrice normalisée $G_N = (I_4/-{}^tA) = (I_4/2{}^tA)$ qui est égale à: $G_N=\begin{pmatrix}1&0&0&0&1&1\\0&1&0&0&2&0\\0&0&1&0&0&2\\0&0&0&1&1&1\end{pmatrix}$

**Remarque 2.9.**

Si $C_1, ...,C_n$ sont les colonnes de $H$, alors pour tout mot $x= (x_1, x_2, \ldots, x_n)\in \mathbb{K}^n$ on a :

$$x.\,{}^tH=x_1C_1 + x_2C_2 +... + x_nC_n.$$

**Théorème 2.8.**

Soit $H$ une matrice de contrôle d'un code linéaire $C$ et $C_1,C_2, ...,C_n$ les colonnes de $H$ alors, il existe un mot de $C$ de poids $r$, si et seulement s'il existe une combinaison linéaire nulle à coefficients non nuls de $r$ colonnes de $H$, c-à-d.

($\exists m\in C$ tel que $w(m) = r$)$\Leftrightarrow$ ($\exists \alpha_{i_j}\in\mathbb{K} / j\in\{1,..,r\}$ non nuls et $\alpha_{i_j}C_{i_1}+ \alpha_{i_2}C_{i_2}+...+ \alpha_{i_r}C_{i_r}= 0$).

Du théorème ci-dessus, on déduit le corollaire suivant :

**Corollaire 2.2.**

1. Soient $C$ un code linéaire et $H$ sa matrice de contrôle et $r$ un entier naturel non nul. Si on ne trouve pas $r$-1 ou moins colonnes linéairement dépendants, alors $d \geq r$.
2. Soit $C$ un code linéaire et $H$ sa matrice de contrôle. La distance minimale du code $d$ est le plus petit nombre de colonnes linéairement dépendants.

**Remarque 2.10.**

1. Dans le cas d'un code binaire, la distance minimale du code $C$ est le plus petit nombre de colonnes d'une matrice de contrôle $H$ dont la somme est nulle.
2. Ce dernier corollaire nous permet donc de calculer la distance minimale d'un code linéaire en connaissant une de ces matrices de contrôle.

**Exemple 2.24.**

Soit $G=\begin{pmatrix}1&1&1&0&0\\1&1&0&1&0\\1&0&0&0&1\end{pmatrix}$ une matrice génératrice d'un code linéaire binaire $C(5,3)$.

Cherchons une matrice de contrôle $H$ de $C$. $G$ est de la forme $G=(A/B)$ tel que $B=\begin{pmatrix}0&0\\1&0\\0&1\end{pmatrix}$

$A=\begin{pmatrix}1&1&1\\1&1&0\\1&0&0\end{pmatrix}$ qui est inversible est donc le code $C$ admet comme matrice génératrice

$G_N=(I_3/A^{-1}B)$. On trouve $G_N=\begin{pmatrix}1&0&0&0&1\\0&1&0&1&1\\0&0&1&1&0\end{pmatrix}$ et donc $C$ admet comme matrice de contrôle la matrice $H=\begin{pmatrix}0&1&1&1&0\\1&1&0&0&1\end{pmatrix}$. En utilisant 2. du corollaire ci-dessus comme $H$ n'a pas de colonnes nulle alors $d \geq 2$, et comme les colonnes $C_1$ et $C_5$ sont égaux alors la distance minimale est $d$=2.

## 2.4 Méthodes de décodage des codes linéaires.

### 2.4.1 Décodage des codes linéaires par tableau standard.

#### 2.4.1.1 Tableau standard

**Définition 2.19.**

Soit $C(n, k, d)$ un code linéaire sur un corps fini $\mathbb{K}$ de cardinal $q$.

La **classe latérale** de $x \in \mathbb{K}^n$, noté cl($x$), est la classe d'équivalence de $x$ par relation d'équivalence $R$ définit dans $\mathbb{K}^n$ par: $x, y \in \mathbb{K}^n$, $x\ R\ y$ si, et seulement si $x\text{-}y \in C$. La classe de $x$ est donné par: cl($x$)=$x$+C=$\{x+c \,/\, c \in C\}$.

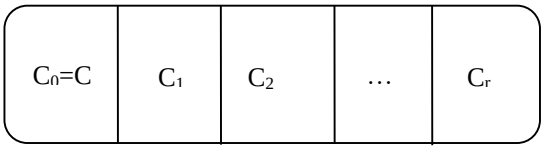



Figure 2.1 Classes latérales.

**Proposition 2.7.**

Il y $q^{n-k}$ classes latérales de $\mathbb{K}^n$ dont chaque classe contient $q^k$ mots.

**Définition 2.20.**

On appelle **tableau standard** d'un code linéaire C(*n*, *k*, *d*) sur un corps fini $\mathbb{K}$ de cardinal *q*, le tableau ci-dessous constitué de deux cases dont l'une (à droite) comporte les classes latérales et l'autre (à gauche) comporte les mots erreur de poids minimal dits chefs de classes.

| Chefs de classe | Classes latérales. |
|---|---|
| $u_0 = 0$ | $C_0 = C$ |
| $u_1$ | $C_1$ |
| $u_2$ | $C_2$ |
| … | … |
| $u_{r-1}$ | $C_{r-1}$ |

Tableau 2.2 Forme d'un tableau standard.

Les $u_i$ sont les représentants des classes $C_i$.

**2.4.1.2 Construction du tableau standard.**

1. On commence par remplir la première ligne à gauche par le mot nul $u_0 = 0$ et celle de droite par les mots codes $c_0=0, c_1, c_2, ..., c_S$.
2. On écrit en dessous du mot nul $u_0$ un mot $u_1 \notin C$ et de plus petit poids possible. Puis les mots $u_1$, $u_1 + c_1$, $u_1 + c_2$, ..., $u_1 + c_S$ sous la classe de *C*.
3. On recommence avec un autre mot $u_2 \notin C$ et $u_2 \notin C_1$ et de poids minimal et ainsi de suite jusqu'à épuisement de tous les mots de $\mathbb{K}^n$. On pose: $s=q^k-1$ et $r= q^{n-k}-1$, alors le tableau est comme suit :

| Mots erreurs (Chefs de classes). | Classes latérales. |
|---|---|
| $u_0 = 00\ldots00$ | $c_0=0, c_1, c_2, \ldots, c_S$ |
| $u_1 = r_1 0\ldots00$ | $u_1, u_1+c_1, u_1+c_2, \ldots, u_1+c_S$ |
| $u_2 = 0\ r_2\ldots00$ | $u_2, u_2+c_1, u_2+c_2, \ldots, u_2+c_S$ |
| … | … |
| $u_r = r_1\ r_2..\ r_e..00$ | $u_r, u_r+c_1, u_r+c_2, \ldots, u_r+c_S$ |

Tableau 2.3 Tableau standard.

**Remarque 2.11.**

Pour un code *e*-correcteur binaire $\mathbb{K}=\mathbb{F}_2$. On a le tableau suivant: posons $s=2^k-1$ et $r= 2^{n-k}-1$.

| Nb erreurs | Mots erreurs. | Classes latérales. |
|---|---|---|
| 1 | 100….00<br>010…00<br>…<br>000….01 | $u_1, u_1+c_1, u_1+c_2, \ldots, u_1+c_S$<br>$v_1, v_1+c_1, v_1+c_2, \ldots, v_1+c_S$<br>…<br>$w_1, w_1+c_1, w_1+c_2, \ldots, w_1+c_S$ |
| 2 | 110…00<br>101…00<br>…<br>000…11 | $u_2, u_2+ c_1, u_2+ c_2, \ldots, u_2+c_S$<br>$v_2, v_2+ c_1, v_2+ c_2, \ldots, v_2+c_S$<br>…<br>$w_2, w_2+ c_1, w_2+ c_2, \ldots, w_2+c_S$ |
| … | … | … |
| E | 111..1..00<br>011..1..00<br>…<br>000..1..11 | $u_e, u_e+ c_1, u_e+ c_e, \ldots, u_e+c_S$<br>$v_e, v_e+ c_1, v_e+ c_e, \ldots, v_e+c_S$<br>…<br>$w_e, w_e+ c_1, w_e+ c_e, \ldots, w_e+c_S$ |

Tableau 2.4 Tableau standard binaire ($K=F_2$).

**2.4.1.3 Principe de décodage par tableau standard.**

Soient $y\in\mathbb{K}^n$ le mot reçu , $c\in C$ le mot envoyé qu'on cherche et $\varepsilon\in\mathbb{K}^n$ le mot erreur avec le poids $w(\varepsilon) \leq e$. Pour décoder le mot y on suit les étapes suivantes:

1- Construire le tableau standard (T) comportant les erreurs et les classes latérales.

2- Chercher dans le tableau (T) le mot *y* parmi les mots de $\mathbb{K}^n$.

a. Si y se situe dans la première ligne alors il n'y a pas d'erreurs ($\varepsilon=0$) et $c=y$.

b. Si non, *y* se situe à l'intersection d'une ligne i et d'une colonne j dans le Tableau standard ci-dessous.

| Mots erreurs. | Classes latérale. |
|---|---|
| $u_0 = 0$ | $c_0=0\ c_1\ ..\ \ c_j\ \ldots\ c_S$ |
| $u_1$ | $u_1\ u_{1+}\ c_1\ \ u_{1+}\ c_2\ \ldots u_{1+}c_S$ |
| | |
| $u_i=\varepsilon_i$ | $u_i\ u_{i+}\ c_1\ ..\ y\ \ \ldots\ \ u_{i+}\ c_S$ |
| . . | . . |
| $u_r$ | $u_r\ \ u_{r+}\ c_1\ \ \ \ u_{r+}\ c_2\ \ \ldots\ \ \ u_{r+}\ c_S$ |

3. Le mot erreur $\varepsilon_i = u_i$ est le chef de classe $C_i$.
4. Le mot envoyé est $c=y - u_i=c_j$ le mot se trouvant à la colonne d'indice $j$.

Les mots erreurs qui pourront être corrigés sont précisément les chefs de classes, quel que soit le mot code envoyé. En choisissons des mots erreurs de poids minimal, en tant que chefs de classes, alors le tableau standard assure un décodage au plus proche voisin c'est ce qu'on appelle "**principe de maximum de vraisemblance voisin**".

**Exemple 2.25.** Soit le code binaire $C(5, 2)$ de matrice contrôle

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

Donc $C$ admet comme matrice génératrice la matrice

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

En appliquant le théorème qui permet de calculer $d$ à partir de $H$, les colonnes de $H$ sont non nulles et distinctes et $C_5=C_1+C_3$ et donc $d=3$. Le tableau standard est le suivant :

| Mots erreurs | Classes latérales | | | |
|---|---|---|---|---|
| 00000 | 00000 | **10110** | 01011 | **11101** |
| 10000 | 10000 | 00110 | 11011 | 01101 |
| 01000 | 01000 | 11110 | 10011 | 10101 |
| 00100 | 00100 | 10010 | 01111 | 11001 |
| 00010 | 00010 | 10100 | 01001 | 11111 |
| **00001** | 00001 | **10111** | 01010 | 11100 |
| 11000 | 11000 | 01110 | **10011** | 00101 |
| 10010 | 10010 | 00100 | 11001 | 01111 |

Tableau 2.5 Tableau de déchiffrement.

1. Si le mot reçu $y_1$=11101, c'est un mot de la première ligne, donc il n'y a pas d'erreurs et le mot envoyé est $x=y$=11101.
2. Si le mot reçu $y_2$=10111, $y_2 \notin C$. le mot erreur $\varepsilon_2$=00001 et le mot envoté $c_2$=10110.

3. Si le mot reçu $y_3$=10011, $y_3 \notin C$. le mot erreur $\varepsilon_3$=11000 et le poids w($\varepsilon_3$) > $e$=1, ce code ne peut pas corriger cette erreur car son poids dépasse la capacité de correction $e$.

### 2.4.2 Décodage des codes linéaires par syndrome.

#### 2.4.2.1 L'application syndrome

**Définition 2.21.**

Soient $C(n, k, d)$ un code linéaire sur un corps fini $\mathbb{K}$ de cardinal $q$, $H$ une matrice de contrôle de $C$ de type $n\text{-}k \times n$. On appelle **application syndrome** associée à $H$, l'application $h$ de $\mathbb{K}^n$ dans $\mathbb{K}^{n-k}$, dont $H$ est sa matrice dans les bases canoniques de $\mathbb{K}^n$ et $\mathbb{K}^{n-k}$. C.à.d.

$$h: \mathbb{K}^n \to \mathbb{K}^{n-k},$$

$$x \mapsto h(x) = x.{}^tH.$$

Le mot $h(x) \in \mathbb{K}^{n-k}$ est appelé **syndrome** de $x \in \mathbb{K}^n$.

**Exemple 2.26.** Soit le code binaire $C(5, 2)$ de matrice contrôle

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

L'application syndrome est:

$$h: \mathbb{F}_2^5 \to \mathbb{F}_2^3,$$

$x=(x_1, x_2, x_3, x_4, x_5) \mapsto h(x)= x.{}^tH=(x_1+ x_3, x_1+x_2+ x_4, x_2+ x_5)$.

| Les mots de $\mathbb{F}_2^5$ | | | | Syndrome |
|---|---|---|---|---|
| 00000 | **10110** | 01011 | **11101** | 000 |
| 10000 | 00110 | 11011 | 01101 | 110 |
| 01000 | 11110 | 10011 | 10101 | 011 |
| 00100 | 10010 | 01111 | 11001 | 100 |
| 00010 | 10100 | 01001 | 11111 | 010 |
| 00001 | **10111** | 01010 | 11100 | 001 |
| 11000 | 01110 | **10011** | 00101 | 101 |
| 10001 | 00111 | 11010 | 01100 | 111 |

Tableau 1.6 Syndromes des mots de $\mathbb{F}_2^5$.

**Proposition 2.8.**

Soient $C(n, k, d)$ un code linéaire sur un corps fini $\mathbb{K}$ de cardinal $q$,

$h$ l'application syndrome de $\mathbb{K}^n$ dans $\mathbb{K}^{n-k}$, alors $Ker h = C$ et $\mathbb{K}^n/C$ est isomorphe à $Im h$.

**Remarque 2.12.**

- Le quotient $\mathbb{K}^n/C$ est l'ensemble des classes latérales du code $C$.
- Les mots de la même classe latérale ont la même image par $h$ (i.e. le même syndrome).
- Les mots du code $C$ ont un syndrome nul.
- Si $y$ est un mot reçu associé à un mot envoyé $x$ alors $y=x+\varepsilon$ avec $\varepsilon$ le mot erreur alors, $h(y)=h(x)+h(\varepsilon)$ et comme $h(x)=0$ car $x \in C$ alors: $h(y)=h(\varepsilon)$, donc le mot reçu et le mot erreur ont le même syndrome.

Cette dernière remarque nous permet de connaitre la classe du mot reçu et donc du mot erreur, ce qui nous amène à trouver le mot erreur plus rapidement que dans le cas de la première méthode du tableau standard.

**Proposition 2.9.**

Soit $C(n, k, d)$ un code linéaire $e$-correcteur sur un corps fini $\mathbb{K}$ et $y_1, y_2 \in \mathbb{K}^n$.

Si $w(y_1) \leq e$ et $w(y_2) \leq e$, alors $h(y_1) = h(y_2)) \Rightarrow y_1 = y_2$.

**Preuve.**

Si $w(y_1) \leq t$ et $w(y_2) \leq t$, alors $w(y_1 - y_2) \leq 2t < d$. De plus $h(y_1) = h(y_2) \Rightarrow h(y_1 - y_2) = 0 \Rightarrow y_1 - y_2 \in C$. Alors $y_1 - y_2 \in C$ et $w(y_1 - y_2) < d$ et comme d est la distance minimale alors $y_1 - y_2 = 0$ d'où $y_1 = y_2$.

### 2.4.2.2 Tableau de déchiffrement par syndrome.

C'est le même tableau standard dans la méthode précédente sauf qu'on lui rajoute une troisième colonne à gauche comportant les syndromes des mots dans chaque classe latérale.

| Chefs de classe | Classes latérale | Syndrome |
|---|---|---|
| $u_0 = 00\ldots00 = 0$ | $c_0=0, c_1, c_2, \ldots, c_S$ | $h(0)$ |
| $u_1 = r_1 0\ldots00$ | $u_1, u_1+c_1, u_1+c_2, \ldots, u_1+c_S$ | $h(u_1)$ |
| $u_2 = 0\ r_2\ldots00$ | $u_2, u_2+c_1, u_2+c_2, \ldots, u_2+c_S$ | $h(u_2)$ |
| . . | . . | .. |
| $u_r = r_1\ r_2 .. r_e 00$ | $u_r, u_r+c_1, u_r+c_2, \ldots, u_r+c_S$ | $h(u_r)$ |

Tableau 2.7 Tableau de déchiffrement par syndrome.

**Remarque 2.13.** Dans le cas d'un code binaire e-correcteur *C* et si l'on est sur que le nombre d'erreurs ne dépasse pas la capacité de correction e alors le tableau standard est dans la page suivante:

| Nb erreurs | Mots erreurs. | Classes latérales. | Syndromes | |
|---|---|---|---|---|
| 0 | 000…00 | $c_0$=0, $c_1$, $c_2$, . . . , $c_S$ | h(0)= 0 | |
| 1 | 100….00<br>010…00<br>…<br>000….01 | $u_1$, $u_1$+$c_1$, $u_1$+$c_2$, . . . , $u_1$+$c_S$<br>$v_1$, $v_1$+$c_1$, $v_1$+$c_2$, . . . , $v_1$+$c_S$<br>…<br>$w_1$, $w_1$+$c_1$, $w_1$+$c_2$, . . . , $w_1$+$c_S$ | h($u_1$)=$C_1$<br>h($v_1$) =$C_2$<br>…<br>h($u_1$) =$C_n$ | Les colonnes de la matrice H. |
| 2 | 110…00<br>101…00<br>…<br>000…11 | $u_2$, $u_2$+ $c_1$, . . ., $u_2$+$c_S$<br>$v_2$, $v_2$+ $c_1$, . . . $v_2$+$c_S$<br>…<br>$w_2$, $w_2$+ $c_1$, . . . $w_2$+$c_S$ | h($u_2$)= $C_1$+ $C_2$<br>h($v_2$) = $C_1$+ $C_3$<br>…<br>h($w_2$) = $C_{n-1}$+ $C_n$ | Somme 2 à 2 des colonnes de la matrice H. |
| … | … | … | | |
| *i* | 111..1..00<br>011..1..00<br>…<br>000..1..11 | $u_e$, $u_e$+ $c_1$, . . ., $u_e$+$c_S$<br>$v_e$, $v_e$+ $c_1$, . . ., $v_e$+$c_S$<br>…<br>$w_e$, $w_e$+ $c_1$, . . , $w_e$+$c_S$ | h($u_e$)<br>h($v_e$)<br>…<br>h($w_e$) | Somme e à e des colonnes de la matrice H. |

Tableau 2.8 Tableau de déchiffrement binaire par syndrome.

**2.4.2.3 Principe du décodage par syndrome.**

Soient $y \in \mathbb{K}^n$ le mot reçu , c∈C le mot envoyé qu'on cherche et $\varepsilon \in \mathbb{K}^n$ le mot erreur avec le poids $w(\varepsilon) \leq e$. Pour décoder le mot *y* on suit les étapes suivantes:

1. Construire le tableau standard (T) comportant mots erreurs, classes latérales et syndromes.
2. Calcul du syndrome du mot reçu y i.e. $h(y)=y.{}^tH$.
3. Si $h(y)=0$ alors il n'y a pas d'erreurs ($\varepsilon=0$) et $x=y$.
4. Si non ($h(y)\neq 0$) alors $y \notin C$ , on détermine la classe latérale associé au mot *y*.
5. Rechercher dans cette classe le mot erreur ε de poids $w(\varepsilon) \leq e$.
6. Calculer le mot envoyé $x=y-\varepsilon$.

**Exemple 2.27.** Soit le code binaire C(5, 2) de matrice contrôle $H=\begin{pmatrix}1&0&1&0&0\\1&1&0&1&0\\0&1&0&0&1\end{pmatrix}$.

Le tableau de déchiffrement est le suivant:

| Mots erreurs | Classes latérale. | | | | Syndrome |
|---|---|---|---|---|---|
| 00000 | 00000 | **10110** | 01011 | **11101** | 000 |
| 10000 | 10000 | 00110 | 11011 | 01101 | 110 |
| 01000 | 01000 | 11110 | 10011 | 10101 | 011 |
| 00100 | 00100 | 10010 | 01111 | 11001 | 100 |
| 00100 | 00100 | 10100 | 01001 | 11111 | 010 |
| 00001 | 00001 | **10111** | 01010 | 11100 | 001 |
| 11000 | 11000 | 01110 | **10011** | 00101 | 101 |
| 10001 | 10001 | 00111 | 11010 | 01100 | 111 |

Tableau 2.9 Tableau standard binaire par syndrome

- Si le mot reçu $y_1$=11101, son syndrome $h(y)= C_1+C_2+C_3+C_5=000$ et si on suppose qu'il y a au plus une erreur alors $y_1$ est dans le code donc $x= y_1$=11101. Si on suppose qu'il y a au plus deux erreurs (i.e. le nombre d'erreurs dépasse $e$=1) alors même chose car la classe latérale de $y_1$ ne contient pas de mot de poids 2. Si on suppose qu' il y a au plus 3 erreurs alors le mot erreur est l'un des mots suivants: 00000, 10110, 01011et le mot envoyé peut être soit 11101, soit 01011, soit 10110. Dans le cas de 4 ou 5 erreurs donc C contient tout les mots de poids inferieur ou égal à 5 et donc chaque mot de C peut être le mot envoyé.
- Si le mot reçu $y_2$=10111, $h(y_2)= C_1+C_3+C_4+C_5=001= C_5=h(00001)$, dans le cas d'une erreur au plus alors l'erreur est $\varepsilon_2$=00001 et le mot envoyé $c_2$=10110. Dans le cas de 2 erreurs au plus le mot erreur est soit $\varepsilon_2$=00001, soit $\varepsilon_2$=01010 et le mot envoyé est soit $c_2$=10110, soit $c_2$=11101.

**Exemple 2.28.** Soit le code ternaire C(5, 2) de matrice contrôle $H=\begin{pmatrix}2&0&1&0&0\\1&2&0&1&0\\0&1&0&0&1\end{pmatrix}$.

La matrice génératrice normalisée $G_N= \begin{pmatrix}1&0&1&2&0\\0&1&0&1&2\end{pmatrix}$ et le tableau standard est le suivant:

| Nb erreurs | Mots erreurs. | Syndromes | Nb erreurs | Mots erreurs. | Syndromes |
|---|---|---|---|---|---|
| 0 | h(0)= 0 | 000…00 | | | |
| 1 | 10000 | $C_1$=210 | 1 | 00200 | 2 $C_3$=20 |
| | 20000 | 2$C_1$=120 | | 00010 | $C_4$=010 |
| | 01000 | $C_2$=021 | | 00020 | 2 $C_4$=020 |
| | 02000 | 2 $C_2$=012 | | 00001 | $C_5$=001 |
| | 00100 | $C_3$=100 | | **00002** | 2 $C_5$=**002** |

Tableau 2.10 Tableau ternaire ($K$= $F_3$) de déchiffrement par syndrome réduit.

Le code C est de distance $d$=3 (car il n'y a pas de colonnes nulle, ni deux colonnes linéairement dépendant alors que $C_1$+$C_3$= $C_4$) et donc le code est 1-correcteur, $e$=1)

Le code est C={00000, 10120, 01012, 20210, 02021, 11102, 22201, 12111, 21222 }

Soit le mot $y_1$= 01011 alors le syndrome de $y_1$ est $h(y_1)$=$C_2$+ $C_4$+$C_5$=002=2$C_5$=$h$(00002). Le mot erreur est donc $\varepsilon_1$=00002 et le mot envoyé est $x_1$= $y_1$-ε=y+2$\varepsilon_1$=01011+00001=01012.

**Exemple 2.29.** Soit le code de *Hamming ternaire* C(13, 10, 3).

De matrice de contrôle $H$=$\begin{pmatrix} 1 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 2 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 2 & 1 & 2 & 2 & 0 & 0 & 1 \end{pmatrix}$ et de distance $d$=3

La matrice génératrice normalisée est $G_N$=$(\ I_{10}\ /\ M)$ tel que $M$=$\begin{bmatrix} 2 & 2 & 2 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 0 \\ 2 & 1 & 0 \\ 2 & 0 & 2 \\ 2 & 0 & 1 \\ 0 & 2 & 2 \\ 0 & 2 & 1 \\ 2 & 2 & 1 \end{bmatrix}$

Soit $y$ =0 1 2 0 0 0 0 0 0 0 1 2 2, le mot reçu, son syndrome est:

$h(y)$=${}^tC_2$+2${}^tC_3$+${}^tC_{11}$+2${}^tC_{12}$+2${}^tC_{13}$ = (2 1 2)=2${}^tC_3$=$h$(0 0 2 0 0 0 0 0 0 0 0 0 0) et le mot erreur associée est: ε=0 0 2 0 0 0 0 0 0 0 0 0 0 et le mot envoyé est:

$x$ = $y$-ε = $y$+2ε = 0 1 0 0 0 0 0 0 0 0 122.

# Chapitre : Codes cycliques.

## Introduction

Les codes cycliques sont une classe importante et puissante de codes linéaires utilisés dans le domaine des communications numériques et de la théorie de l'information. Ils ont été largement étudiés et appliqués dans diverses technologies de communication, notamment dans les systèmes de télécommunications, les réseaux informatiques et les dispositifs de stockage de données.

Les codes cycliques sont des codes correcteurs linéaires qui se fondent sur la théorie des corps finis, et en particulier les extensions de Galois, ainsi que sur les propriétés algébriques spéciales des polynômes cycliques qui sont exploitées pour concevoir un mécanisme efficace de détection et de correction d'erreurs. La principale caractéristique des codes cycliques réside dans leur capacité à garantir la détection et la correction d'un certain nombre d'erreurs, en utilisant des techniques de codage et de décodage relativement simples.

Dans ce chapitre, nous explorerons les principaux concepts et caractéristiques des codes cycliques, y compris leurs structures, leurs représentations polynomiales, leurs polynômes générateurs, leurs matrices génératrices, leurs matrices de contrôles et leur technique de codage systématique en utilisant le syndrome polynomial.

## 3.1 Définition et description d'un code cyclique

### 3.1.1 Définition et exemples

**Définition 3.1.**

Un code linéaire $C$ de longueur *n* sur un corps fini $\mathbb{K}$ est dit **code cyclique** s'il vérifie la propriété suivante:

Pour tout $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{K}^n$ :

$$c = (c_0, c_1, \ldots, c_{n-1}) \in C \Rightarrow c' = (c_{n-1}, c_0, \ldots, c_{n-2}) \in C$$

● *c*' qui n'est que la permutation circulaire des composantes de *c* est appelée **shift** de *c*. C'est à dire que $c' = (c_{n-1}, c_0, \ldots, c_{n-2})$ est le shift de $c = (c_0, c_1, \ldots, c_{n-1})$.

**Exemple 3.1.**

1. $\{0\}$ et $\mathbb{K}^n$ sont des codes cycliques dits triviaux.

2. Le code $C = \{000,101,011,110\}$ est un code cyclique.

3. Tout code de Hamming est un code cyclique.

4. Le code $C = \{0000,1001,0110,1111\}$ n'est pas un code cyclique, car le mot $c$=1001$\in C$ et son shift $c$'=1100$\notin C$.

### 3.1.2 Représentation polynomial d'un code cyclique

Tout mot $c = (c_0, c_1, \ldots, c_{n-1})$ d'un code linéaire C sur un corps fini $\mathbb{K}$ peut être identifier à un polynôme $c(X) = c_0 + c_1X + \ldots + c_{n-1}X^{n-1}$ de $\mathbb{K}[X]$.

On associée au mot shift $c' = (c_{n-1}, c_0, \ldots, c_{n-2})$ du mot $c$ , le polynôme

$c'(X) = c_{n-1} + c_0X + \ldots + c_{n-2}X^{n-1}$ de $\mathbb{K}[X]$, ce polynôme peut être obtenu en calculant le produit $Xc(X)$ et en considérant que $X^n = 1$, c'est-à-dire en calculant modulo $X^n - 1$, et précisément ce calcul se fait dans l'anneau quotient $\mathbb{K}[X]/< X^n - 1 >$.

De ce qui précède, on obtient la proposition suivante :

**Proposition 3.1.**

Un code linéaire $C(n, k)$ est cyclique si, et seulement si, pour tout mot $c$ de $C$, le polynôme $Xc(X)$ calculé modulo $X^n - 1$, est le polynôme associée au mot $c' = (c_{n-1}, c_0, .., c_{n-2})$ de $C$.

**Définition 3.2.**

Soit $\mathbb{K}$ un corps fini et $n$ un entier non nul.

● On appelle **représentation polynomiale de $\mathbb{K}^n$**, l'application θ définie par

$$\theta : \mathbb{K}^n \to \mathbb{K}[X]/< X^n - 1 >$$

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

Le polynôme $c(X)$ est dit **représentation polynomiale du mot** $c$.

● On appelle **représentation polynomiale d'un code cyclique** $C$ de $\mathbb{K}^n$, l'ensemble des représentations polynomiales des mots du code $C$, c'est-à-dire :

$$\theta(C) = \{\, \theta(c) : \; c \in C\} = \; Im(\theta).$$

**Proposition 3.2.**

Soit $C$ un code linéaire de langueur *n* sur un corps fini $\mathbb{K}$.

$C$ est un code cyclique si, et seulement si, sa représentation polynomiale $\theta(C)$ est un idéal de l'anneau $\mathbb{K}[X]/< X^n - 1 >$.

**Preuve.**

Supposons que $C$ est un code cyclique et soit $c(X) = \sum_{i=0}^{n-1} c_i X^i$, $d(X) = \sum_{i=0}^{n-1} d_i X^i$ dans $\theta(C)$, donc $c = (c_0, c_1, \ldots, c_{n-1})$ et $d = (d_0, d_1, \ldots, d_{n-1}) \in C$, et comme *C* est un sous-espace vectoriel, alors pour $\alpha, \beta \in \mathbb{K}$, le mot $m = \alpha c + \beta d \in C$, ce qui donne en représentation polynomiale : $m(X) = \alpha c(X) + \beta d(X) \in \theta(C)$, d'où $\theta(C)$ est un espace vectoriel sur $\mathbb{K}[X]$. De plus soit $p(X) = \sum_{i=0}^{n-1} a_i X^i \in \mathbb{K}[X]/< X^n - 1 >$, alors comme $c(X) \in \theta(C)$ et $C$ cyclique alors $c(X), Xc(X), X^2c(X), \ldots, X^ic(X), \ldots \in \theta(C)$ et donc $a_0c(X) + a_1Xc(X) + a_2X^2c(X) + \ldots \in \theta(C)$, par la suite $p(X)c(X) \in \theta(C)$, d'où $\theta(C)$est un idéal de $\mathbb{K}[X]/< X^n - 1 >$.

Inversement, soit $\theta(C)$ un idéal de $\mathbb{K}[X]/< X^n - 1 >$. Alors si $c = (c_0, c_1, \ldots, c_{n-1}) \in C$ alors $c(X) = \sum_{i=0}^{n-1} c_i X^i \in \theta(C)$ et donc $Xc(X) \in \theta(C) \Rightarrow c' = (c_{n-1}, c_0, \ldots, c_{n-2}) \in C$, ce qui montre que $C$ est un code cyclique.

**Théorème 3.1.**

Soient $\mathbb{K}$ un corps fini et *n* un entier non nul, $C$ un code cyclique de longueur *n* non réduit à $\{0\}$. Alors $\theta(C)$ est un idéal principal de l'anneau $\mathbb{K}[X]/< X^n - 1 >$.

**Preuve**

Soit $g(X)$ un polynôme non nul et unitaire dans $\theta(C)$, de degré minimum. En effectuant la division Euclidienne de $X^n - 1$ par $g(X)$ (dans $\mathbb{K}[X]$), on trouve

$X^n - 1 = g(X)q(X) + r(X)$ avec $r(X) = 0$ ou $d°\big(r(X)\big) < d°\big(g(X)\big)$, donc dans $\mathbb{K}[X]/< X^n - 1 >$: $g(X)q(X) = -r(X) \in \theta(C)$, avec $d°\big(r(X)\big) < d°\big(g(X)\big)$, ce qui contredit la définition de $g(X)$ et donc $r(X) = 0$ et $X^n - 1 = g(X)q(X)$, d'où $g(X)$ divise $X^n - 1$.

Montrons que $\theta(C) =< g(X) >$. Soit $f(X) \in \theta(C)$, en divisant $f(X)$ par $g(X)$ dans $\mathbb{K}[X]$, alors il existe $q'(X), r'(X)$ dans $\mathbb{K}[X]$ : $f(X) = q'(X)g(X) + r'(X)$ avec $r'(X) = 0$ ou

$d°\big(r'(X)\big) < d°\big(g(X)\big)$, on trouve comme précédemment $r'(X) = 0$ et donc $f(X)$ est un multiple de $g(X)$ et $\theta(C)$ est un idéal principal engendré par $g(X)$.

### 3.1.3 Polynôme générateur et matrice génératrice d'un code cyclique

**Définition 3.3.**

Le polynôme $g(X)$ dans le Théorème 3.1 engendrant $\theta(C)$ est appelé le **polynôme générateur** du code cyclique *C*. $\theta(C)$ est l'idéal constitué des multiples de $g(X)$ dans $\mathbb{K}[X]/< X^n - 1 >$.

On a : $c \in C \Leftrightarrow c(X) \in \theta(C) \Leftrightarrow \exists\, q(X) \in \mathbb{K}[X]: c(X) = q(X)g(X)$. Du Théorème 3.1, on déduit les propriétés suivantes du polynôme générateur $g(X)$.

**Proposition 3.1.**

1. Le polynôme $g(X)$ est de degré minimale dans $\theta(C)$.
2. Le polynôme $g(X)$ est unitaire et unique.
3. Tout mot du code cyclique est multiple du polynôme générateur.
4. $g(X)$ divise $X^n - 1$.

**Exemple 3.2.**

Soit $C\,(3,2)$ un code cyclique tel que $\theta(C) = \{0, 1 + X, 1 + X^2, X + X^2\}$

Le polynôme $X + 1$ est le polynôme générateur du code $C$.

**Remarque 3.1.**

Pour trouver tous les codes cycliques de longueur *n* sur $\mathbb{F}_p$, il suffit de trouver tous les diviseurs du polynôme $X^n - 1$ sur $\mathbb{F}_p$. Pour cela il faut décomposer le polynôme $X^n - 1$ en produit de polynômes irréductibles sur $\mathbb{F}_p$.

**Exemple 3.3.**

Les codes cycliques non nuls de longueur *n=5* sur le corps $\mathbb{F}_2$ :

La décomposition de $X^5 - 1$ sur $\mathbb{F}_2$ en polynômes cyclotomiques donne

$$X^5 - 1 = \prod_{d/5} \phi_d(X)$$
$$= \phi_1(X)\phi_5(X)$$
$$= (X-1)(X^4 + X^3 + X^2 + X + 1)$$

Soit $\mathbb{K}=\mathbb{F}_{2^r}$, le corps des racines 5ieme de l'unité sur $\mathbb{F}_2$, où *r* est le plus petit entier non nul tel que *n*=5 divise $2^r - 1$, alors on trouve *r=4* et donc $\mathbb{K} = \mathbb{F}_{16}$. $\phi_5(X)$ est irréductible sur $\mathbb{F}_2$ car φ(5)=*r*=4. Chaque diviseur donne un générateur d'un code cyclique de longueur $n = 5$ sur$\mathbb{F}_2$
Si on note $g_i(X)$ le générateur du code $C_i$ on trouve :

$C_0: g_0(X) = X^5 - 1 = 0 \rightarrow C_0 = \{0\}$.

$C_1: g_1(X) = X - 1$.

$C_2: g_2(X) = X^4 + X^3 + X^2 + X + 1$.

$C_3: g_3(X) = 1 \rightarrow C_3 = \mathbb{K}^5$.

**Exemple 3.4.**

Les codes cycliques non nuls de longueur *n=7* sur le corps $\mathbb{F}_2$.

La décomposition de $X^7 - 1$ sur $\mathbb{F}_2$ en polynômes cyclotomiques donne :

$$X^7 - 1 = \prod_{d/7} \phi_d(X)$$
$$= \phi_1(X)\phi_7(X)$$
$$= (X-1)(X^6 + X^5 + X^4 + X^3 + X^2 + X + 1)$$

Soit $\mathbb{K}= \mathbb{F}_{2^r}$, le corps des racines 5ieme de l'unité sur $\mathbb{F}_2$, où *r* est le plus petit entier non nul tel que *n=7* divise $2^r - 1$, alors *r=3* et donc $\mathbb{K}=\mathbb{F}_8$.

Le degré de $\phi_7(X)$ est $\varphi(7) = card\{i \in \mathbb{N} \ / \ i < 7 \ et \ i \wedge 7 = 1\} = 6$ donc $\phi_7(X)$ n'est pas irreductible, mais il se décompose en produit de $\frac{\varphi(7)}{r} = 2$ polynômes irréductibles de degré *r=3* sur$\mathbb{F}_2$, donc $\phi_7(X) = X^6 + X^5 + X^4 + X^3 + X^2 + X + 1 = (X^3 + bX^2 + cX + 1)(X^3 + b'X^2 + c'X + 1)$, après identification on trouve : $b = c' = 0$ et $c = b' = 1$, donc

$\phi_7(X) = (X^3 + X^2 + X + 1)(X^3 + X^2 + X + 1)$, d'où la décomposition de $X^7 - 1$ est :

$X^7 - 1 = (X-1)(X^3 + X + 1)(X^3 + X^2 + 1)$ et chaque diviseur $g_i(X)$ de $X^7 - 1$ engendre un code cyclique $C_i$ de longueur *n=7* sur $\mathbb{F}_2$.

$C_0: g_0(X) = X^7 - 1 = 0, C_0 = \{0\}$ est le code cyclique trivial.

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

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

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

$C_4: g_4(X) = (X-1)(X^3 + X + 1) = X^4 + X^3 + X^2 + 1$

$C_5: g_5(X) = (X-1)(X^3 + X^2 + 1) = X^4 + X^2 + X + 1$

$C_6: g_6(X) = (X^3 + X + 1)(X^3 + X^2 + 1) = X^6 + X^5 + X^4 + X^3 + X^2 + 1$

Pour tout *i*, la représentation polynomiale $\theta(C_i)$ est formée par tous les multiples de $g_i(X)$ modulo $X^7 - 1$, c'est-à-dire par les produits $q(X)g_i(X)$, $q(X) \in \mathbb{K}[X]/< X^7 - 1 >$ et donc le code $C_i$ est formée par tous les mots correspondants à ces produits.

**Théorème 3.2.**

Soit $C$ un code cyclique de longueur *n* sur un corps fini $\mathbb{K}$ de polynôme générateur

$g(X) = g_0 + g_1X + g_2X^2 + \ldots + g_{t-1}X^{t-1} + X^t$ avec $d°g(X) = t$. Alors $\dim C = k = n - t$.

Et $C$ admet la matrice suivante $G$ comme matrice génératrice :

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \ldots & g_t & 0 & \ldots & \ldots & 0 \\ 0 & g_0 & g_1 & g_2 & \ldots & g_t & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & g_0 & g_1 & g_2 & \ldots & g_t & 0 \\ 0 & \ldots & \ldots & 0 & g_0 & g_1 & g_2 & \ldots & g_t \end{pmatrix}.$$

**Preuve.**

Soit $C$ un code cyclique de longueur n sur $\mathbb{K}$ et $g(X)$ le générateur de $C$, tel que $d°\big(g(X)\big) = t$. Tout polynôme $c(X)$ de la représentation $\theta(C)$ est de la forme :

$c(X) = a(X)g(X) = (a_0 + a_1X + a_2X^2 + \cdots + a_sX^s)\big(g(X)\big)$

$= a_0g(X) + a_1Xg(X) + a_2X^2g(X) + \ldots + a_sX^sg(X)$, avec $a_s \in \mathbb{K}$ et $0 \leq s \leq n-1$, les polynômes $g(X), Xg(X), \ldots, X^sg(X)$ forment donc une famille génératrice de $\theta(C)$. On va extraire de cette famille génératrice une base pour $\theta(C)$.

Soit $c(X) = a(X)g(X) \in \theta(C)$ et $h(X) = X^n - 1/g(X)$ dans $\mathbb{K}[X]$. En utilisant la division Euclidienne de $a(X)$ par $h(X)$ dans $\mathbb{K}$[X] , on obtient

$a(X) = q(X)h(X) + r(X)$, avec $d°r(X) < d°h(X) = n - t$, donc $r(X)$ est de la forme:

$$r(X) = r_0 + r_1X + \ldots + r_{n-t-1}X^{n-t-1},$$

en conséquence

$$a(X)g(X) = q(X)h(X)g(X) + r(X)g(X)$$
$$=(X^n - 1)q(X) + r(X)g(X).$$

En calculant dans l'anneau quotient $\mathbb{K}[X] /< X^n - 1 >$, on déduit que

$a(X)g(X) = r(X)g(X)$ $donc$ $c(X) = r_0 g(X) + r_1 X g(X) + \ldots + r_{n-t-1} X^{n-t-1} g(X)$, d'où la famille des polynômes $g(X), Xg(X), \ldots, X^{n-t-1}g(X)$ est une famille génératrice de $\theta(C)$.

Montrons que cette famille est libre ? Dans $\mathbb{K}[X] /< X^n - 1 >$ considérons l'égalité:

$$\alpha_0 g(X) + \alpha_1 X g(X) + \cdots + \alpha_{n-t-1} X^{n-t-1} g(x) = 0, \ldots(*)$$

avec $\alpha_i \in \mathbb{K}$ et $i \in \{0, 1, \ldots, n-t-1\}$. L'égalité (*) implique que dans $\mathbb{K}[X]$, on a:

$$(\alpha_0 + \alpha_1 X + \cdots + \alpha_{n-t-1} X^{n-t-1}) g(X) \equiv 0 (mod\ X^n - 1)$$

Posons $d(X) = (\alpha_0 + \alpha_1 X + \cdots + \alpha_{n-t-1} X^{n-t-1}) g(X)$, alors $d(X)$ est de degré au plus $n-1$ et $d(X)$ divisible par $X^n - 1$, cela conduit à $d(X) = 0$. Comme $\mathbb{K}[X]$ est intègre et $g(X)$ n'est pas nul, alors $\alpha_0 + \alpha_1 X + \alpha_{n-t-1} X^{n-t-1} = 0$, donc $\alpha_0 = \cdots = \alpha_{n-t-1} = 0$, et d'où la famille $\{g(X), Xg(X), \ldots, X^{n-t-1}g(X)\}$ est libre et donc elle forme une base de $\theta(C)$ et la dimension de $C$ est: *k= n-t*. Les mots $l_0, l_1, \ldots, l_{n-t}$ correspondants respectivement au polynômes $g(X), Xg(X), \ldots, X^{n-t-1}g(X)$ forme une base du code $C$ et donc la matrice $G$ dont les lignes sont les mots:

$l_0 = g_0 g_1 g_2 \ldots g_t 0 \ldots 0$, $l_1 = 0 g_0 g_1 g_2 \ldots g_t 0 \ldots 0$, $\ldots$, $l_{n-t-1} = 0 \ldots 0 g_0 g_1 g_2 \ldots g_t$ est une matrice génératrice de $C$.

**Exemple 3.5.**

1. Le code de Hamming de paramètre $C(7, 4, 3)$ et de polynôme générateur : $g(X) = 1 + X + X^3$, admet comme matrice génératrice la matrice :

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

2. Pour les codes cycliques de longueur $n = 7$, on a le tableau suivant

| Le code cyclique | Le générateur | La dimension |
|---|---|---|
| $c_0$ | 0 | 0 |
| $c_1$ | $X-1$ | 6 |
| $c_2$ | $X^3+X+1$ | 4 |
| $c_3$ | $X^3+X^2+1$ | 4 |
| $c_4$ | $X^4+X^3+X^2+1$ | 3 |
| $c_5$ | $X^4+X^2+X+1$ | 3 |
| $c_6$ | $X^6+X^5+X^4+X^3+X^2+X+1$ | 1 |

Tableau 3.1 Table Codes cycliques de longueur *n*=7 et leurs dimensions.

3- Le code cyclique $C_4(7, 3, 3)$ admet comme générateur le polynôme $g_4(X) = X^4 + X^2 + X + 1$ et comme matrice génératrice la matrice :

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

## 3.2 Polynôme et matrice de contrôle d'un code cyclique

### 3.2.1 Polynôme d'un code cyclique

**Définition 3.4.**

Soit $C$ un code cyclique de longueur $n$ sur un corps fini $\mathbb{K}$ et de polynôme générateur $g(X)$.

Le polynôme $h(X) \in \mathbb{K}[X]$ tel que $h(X) = \frac{X^n-1}{g(X)}$ est dit **polynôme de contrôle** du code $C$.

- Le degré de $h(X)$ est donc $k = n - d°g(X) = n - t$.
- $c$ est un mot de $C$ si, et seulement si, $c(X)h(X) = 0$ dans $\mathbb{K}[X]/< X^n - 1 >$.

### 3.2.2 Matrice de contrôle d'un code cyclique

**Théorème 3.3.**

Soit $C$ un code cyclique de longueur $n$ sur un corps $\mathbb{K}$:

1. L'orthogonal $C^\perp$ d'un code cyclique $C$ est un code cyclique.

2. Si $h(X) = h_0 + h_1X + h_2X^2 + \ldots + h_kX^k$ est le polynôme de contrôle du code cyclique $C$ alors le générateur de $C^\perp$ est $h_1(X) = h_0^{-1}\bar{h}(X)$ où $\bar{h}(X) = X^k h(X^{-1})$ est le polynôme réciproque de $h(X)$.

3. La matrice $H_1$ suivante est une matrice de contrôle de $C$ :

$$H_1 = \begin{pmatrix} h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & \dots & \dots & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 \\ 0 & \dots & \dots & 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 \end{pmatrix}$$

**Preuve.**

On a $X^n - 1 = h(X)g(X)$ alors $h(X)g(X) = 0$ dans $\mathbb{K}[X] /< X^n - 1 >$. Soit

$a(X) = \sum_{i=0}^{n-1} a_i X^i \in \theta(C)$, donc $a(X)$ est un multiple de $g(X)$ dans $\mathbb{K}[X] /< X^n - 1 >$.

Si $h(X) = \sum_{i=0}^{n-1} h_i X^i$, alors $h(X)a(X) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_j h_{i-j} X^i = 0$ (où les différences

*i-j* sont calculées modulo *n*), on déduit que pour tout $i \in \{0, \dots, n-1\}$: $\sum_{j=0}^{i} a_j h_{i-j} = 0$.

En particulier, pour tout $i \in \{k, \dots, n-1\}$ on trouve les relations suivantes :

- Si *i=k*: $a_0 h_k + a_1 h_{k-1} + a_2 h_{k-2} + \cdots + a_k h_0 + a_{k+1} 0 + \cdots + a_{n-1} 0 = 0$ donc

$$(h_k, h_{k-1}, h_{k-2}, \dots, h_0, 0, \dots, 0) \perp (a_0, a_1, \dots, a_{n-1}).$$

- Si *i=k+1*: $a_0 0 + a_1 h_k + a_2 h_{k-1} + \cdots + a_k h_1 + a_{k+1} h_0 + \cdots + a_{n-1} 0 = 0$ donc

$$(0, h_k, h_{k-1}, h_{k-2}, \dots, h_0, 0, \dots, 0) \perp (a_0, a_1, \dots, a_{n-1}).$$

- Si *i=k+2*: $a_0 0 + a_1 0 + a_2 h_k + \cdots + a_k h_2 + a_{k+1} h_1 + \cdots + a_{n-1} 0 = 0$ donc

$$(0,0, h_k, h_{k-1}, \dots, h_0, 0, \dots, 0) \perp (a_0, a_1, \dots, a_{n-1}).$$

$$\vdots$$

- Si *i=n-1*: alors $a_0 . 0 + a_1 . 0 + a_2 . 0 + \cdots + a_{n-k-2} . 0 + a_{n-k-1} h_k + \cdots + a_{n-1} h_0 = 0$ donc

$$(0,0,0, \dots 0, h_k, h_{k-1}, \dots, h_1, h_0) \perp (a_0, a_1, \dots, a_{n-1}).$$

Les relations précédentes montrent que les shifts du mot $(h_k, h_{k-1}, h_{k-2}, \dots, h_0, 0, \dots, 0)$ sont orthogonaux au mot $a = (a_0, a_1, \dots, a_{n-1}) \in C$. .En d'autres termes ,les mots suivants :

$$(h_k, h_{k-1}, h_{k-2}, \dots, h_0, 0, \dots, 0)$$

$$(0, h_k, h_{k-1}, h_{k-2}, \dots, h_0, 0, \dots, 0)$$

$$\vdots$$

$$(0,0,0, \dots, h_k, h_{k-1}, \dots, h_1, h_0)$$

Sont orthogonaux au code $C$ donc ils appartient à $C^{\perp}$.

Soit $H_1 = \begin{pmatrix} h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & \dots & \dots & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 \\ 0 & \dots & \dots & 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 \end{pmatrix}$

la matrice dont les lignes sont ces mots ci-dessus. La matrice constitué des *t=n-k* premières colonnes de la matrice $H_1$ est une matrice triangulaire inversible car $h_k \neq 0$, ce qui prouve que la matrice $H_1$ est de rang t, donc ses lignes forment une base à $C^{\perp}$. Ce qui montre que $H_1$ est une matrice de contrôle du code $C$. Comme $h(X)$ divise $X^n - 1$, alors $h(0) = h_0 \neq 0$, et donc la matrice $H = h_0^{-1} H_1$ est aussi une autre matrice de contrôle de $C$, de plus le polynôme associé à la première ligne de *H* est le polynôme

$$h_1(X) = h_0^{-1}(h_k + h_{k-1}X + \cdots + X^k) = h_0^{-1}\bar{h}(X)$$

(où $\bar{h}(X)$ est le polynôme réciproque de $h(X)$) qui est un polynôme unitaire divisant $X^n - 1$, c'est donc « le » générateur du code cyclique $C^{\perp}$. Par ailleurs on constate que la matrice *H* est une matrice génératrice du code cyclique engendré par le polynôme $h_1(X)$.

**Exemple 3.6.**

Soit $C$ un code cyclique sur $\mathbb{F}_2$ de longueur *n*=7 et de polynôme générateur

$$g(X) = X^3 + X^2 + 1,$$

alors le polynôme de contrôle de $C$ est :

$$\begin{aligned} h(X) &= X^7 - 1/(X^3 + X^2 + 1) \\ &= X^4 + X^3 + X^2 + 1 \end{aligned}$$

L'orthogonal de $C$ est engendré par le polynôme : $h_1(X) = X^4 h(X^{-1}) = X^4 + X^2 + X + 1$

et donc $C$ admet comme matrice contrôle la matrice

$$H_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

## 3.3 Codes et codage cycliques systématiques

### 3.3.1 Codes cycliques systématiques

**Définition 3.5.**

Un code cyclique $C(n,k)$ sur un corps fini $\mathbb{K}$ est dit **code cyclique systématique**, s'il admet une matrice génératrice $G$ dite **normalisée** dont les *k* dernières colonnes forment la matrice identité $I_k$ et non pas les *k* premières colonnes dans le cas des codes linéaires. c-à-d $G = (M, I_k)$ où $M \in M_{k,n-k}(\mathbb{K})$.

**Exemple 3.7.**

Le code cyclique $C$ de longueur *n*=7 et de générateur $g(X) = X^3 + X + 1$ sur $\mathbb{F}_2$, admet comme matrice génératrice la matrice suivante G :

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

Qu'on peut mettre sous la forme normalisée

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

Donc $C$ est un code cyclique systématique.

### 3.3.2 Algorithme de codage systématique d'un code cyclique

Soit $C$ un code cyclique de longueur *n* sur un corps fini $\mathbb{K}$ de générateur le polynôme

$$g(X) = g_0 + g_1X + g_2X^2 + \ldots + g_tX^t, \text{ tel que } d°(g(X)) = t.$$

Le code $C$ admet une matrice génératrice (pas nécessairement normalisée) $G$ de la forme

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_t & 0 & \dots & \dots & 0 \\ 0 & g_0 & g_1 & g_2 & \dots & g_t & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & g_0 & g_1 & g_2 & \dots & g_t & 0 \\ 0 & \dots & \dots & 0 & g_0 & g_1 & g_2 & \dots & g_t \end{pmatrix}$$

$G = (M, T)$ où $M \in M_{k,n-k}(\mathbb{K})$ et $T$ est constitué des $k$=$n$-$t$ dernières colonnes de $G$. De plus $T$ est triangulaire inversible car $g_t = 1 \neq 0$.

La matrice génératrice normalisée de $C$ est la matrice $G_N = T^{-1}G = (N, I_k)$, tel que

$N = T^{-1}M$. La matrice $G_N$ est utilisée pour le codage systématique comme suit :

Soient le mot $(a_0, a_1, \dots, a_{k-1}) \in \mathbb{K}^k$, le mot codé $c$ est le produit du mot $a$ par la matrice $G$, donc $c = a.G = (aN, a) = [(a_0, a_1, \dots, a_{k-1})N, a_0, a_1, \dots, a_{k-1}]$. En posant

$(a_0, a_1, \dots, a_{k-1})N = (b_0, b_1, \dots, b_{n-k-1})$, alors en langage polynomial on obtient :

$$c(X) = b_0 + b_1X + \dots + b_{n-k-1}X^{n-k-1} + a_0X^{n-k} + a_1X^{n-k+1} + \dots + a_{k-1}X^{n-1}.$$

D'où $c(X) = b(X) + X^{n-k}a(X)$ où $b(X) = b_0 + b_1X + \dots + b_{n-k-1}X^{n-k-1}$ qu'il faut déterminer. Comme $c(X) \in \theta(C)$, alors $\exists u(X) \in \mathbb{K}[X]$ tel que $c(X) = u(X)g(X)$ et donc

$X^{n-k}a(X) = u(X)g(X) + (-b(X))$ avec $d°(-b(X)) < d°(g(X))$, cela veut dire que $(-b(X))$ n'est que $r(X)$ le reste de la division Euclidienne de $X^{n-k}a(X)$ par $g(X)$ et donc $b(X) = -r(X)$.

**Conséquence 3.1.**

Le codage systématique d'un mot $(a_0, a_1, \dots, a_{k-1}) \in \mathbb{K}^k$ se fait en représentation polynomiale par $a(X) \to c(X) = -r(X) + X^t a(X)$ où $r(X)$ est le reste de la division Euclidienne de $X^t a(X)$ par $g(X)$ tel que $t = d°(g(X))$.

**Exemple 3.8.**

On considère le code cyclique $C(7,4)$sur $\mathbb{F}_2$, de générateur $g(X) = X^3 + X^2 + 1$ et de matrice génératrice

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (M, T), \text{ où } M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \text{ et } T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

La matrice normalisé de $C$ est la matrice $G_N = (T^{-1}M, I_4)$ où

$$T^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \text{ et } T^{-1}M = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \text{ donc } G_N = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Le codage systématique d'un mot $a(X)$ se fait comme suit :

$a(X) \to c(X) = r(X) + X^3 a(X)$ où $r(X)$ est le reste de la division Euclidienne de $X^3 a(X)$ par $g(X)$.

Si on prend $a(X) = X^3$, alors si $r(X)$ est le reste de la division Euclidienne de $X^3 a(X) = X^6$ par $g(X)$. En utilisant un registre à décalage circulaire, on trouve que, $r(X) = X^3 + X$. Donc $c(X) = X^6 + X^2 + X$. En fin le codage du mot $a = 0001$ est le mot code $c = 0110001$.

## 3.4 Codes B.C.H et codes de Reed-Solomon

On présente dans cette partie, quelques codes cycliques particuliers utilisés en pratique tel que les codes B.C.H et les codes de Reed-Solomon.

### 3.4.1 Codes B.C.H

Les codes B.C.H sont des codes cycliques particuliers qui permettent de prévoit la distance minimale ( et donc la capacité de correction) avant la construction de ces codes.

Pour obtenir un code qui corrige au moins e erreur on peut choisir un code B.C.H de distance construire égale à $2e+1$ ou à $2e+2$. Il est plus économique de choisir un code B.C.H de distance construire égale à $2e + 1$, on obtient ainsi un polynôme générateur de degré plus petit et une dimension et un nombre de mots plus grand. On choisit donc dans la suite des codes B.C.H de distance construite $2e + 1$.

Un peu d'historique :

1959 : Découverte de ces codes par Hocquenghem

1960 : Découverte par Bose et Ray-Chaudhuri. Peterson prouve leur nature cyclique.

Peterson trouve un premier algorithme de décodage qui sera par la suite revu et généralisé par d'autres mathématiciens.

1981 : Gorenstein et Zierler généralisent ces codes aux alphabets à $p^n$ ($p$ premier) symboles.

**Proposition 3.4.**

Soient $C$ un code cyclique de longueur *n* sur le corps $\mathbb{K} = \mathbb{F}_{p^r}$ des racines nièmes de l'unité, de générateur $g(X)$ de degré *t* et de racines $\alpha_i$ / $i \in \{1,\ldots,t\}$.

Alors la matrice $H = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-2} & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-2} & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_t & \alpha_t^2 & \cdots & \alpha_t^{n-2} & \alpha_t^{n-1} \end{pmatrix}$ est une matrice de contrôle de *C*.

**Preuve.**

En effet on a : $c=c_0c_0 \ldots\ldots c_{n-1} \in C \Leftrightarrow c(x)=c_0 + c_1x + \cdots + c_{n-1}x^{n-1} \in \theta(C)$

$\Leftrightarrow \exists\, q(x) \in \mathbb{K}[X] : c(x)= q(x)g(x) \Rightarrow$ pour tout $i \in \{1,\ldots, t\} : c(\alpha_i)= q(\alpha_i)\, g(\alpha_i)=0.$

Ce qui donne :

$$c_0 + c_1\alpha_1 x + \cdots + c_{n-1}\alpha_1^{n-1}=0 \ldots (1)$$
$$c_0 + c_1\alpha_2 x + \cdots + c_{n-1}\alpha_2^{n-1}=0 \ldots (2)$$
$$\vdots$$
$$c_0 + c_1\alpha_t x + \cdots + c_{n-1}\alpha_t^{n-1}=0 \ldots (t)$$

L'écriture matricielle de ce système de *t* équations donne : $c.H^t = 0$ où *H* est la matrice

$$H = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-2} & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-2} & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_t & \alpha_t^2 & \cdots & \alpha_t^{n-2} & \alpha_t^{n-1} \end{pmatrix}$$

qui est une matrice de contrôle du code $C$.

**Théorème 3.4.**

Soit $\beta$ une racine $nièmes$ primitive de l'unité sur le corps $\mathbb{F}_q$. Soient $C(n, k, d)$ un code cyclique de longueur *n* et g(x) son polynôme générateur. Si pour un certain entier $b \geq 0$ et un certain entier $\delta \geq 2$, nous avons : $g(\beta^b)=g(\beta^{b+1})= \cdots =g(\beta^{b+\delta-2})=0$, alors $d \geq \delta$, et $\delta$ est dite distance construite du code $C$.

**Preuve.**

Si $c=c_0c_0 \ldots . c_{n-1} \in C$ un mot code alors $c(x)=c_0 + c_1x + \cdots + c_{n-1}x^{n-1} \in \theta(C)$

On a: $g(\beta^b)=g(\beta^{b+1})=\ldots =g(\beta^{b+\delta-2})=0$, alors d'après la proposition précédente

la matrice suivante:

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

est une matrice de contrôle du code $C$. Nous allons montrer que chaque $\delta - 1$ colonnes de $H$, sont linéairement indépendants sur $\mathbb{F}_q$. On va montrer que chaque matrice carrée

$$B_w = \begin{pmatrix} \beta^{j_1 b} & \beta^{j_2 b} & \cdots & \beta^{j_w b} \\ \beta^{j_1(b+1)} & \beta^{j_2(b+1)} & \cdots & \beta^{j_w(b+1)} \\ \vdots & \vdots & & \vdots \\ \beta^{j_1(b+w-1)} & \beta^{j_2(b+w-1)} & \cdots & \beta^{j_w(b+w-1)} \end{pmatrix}$$

d'ordre $w \leq \delta - 1$, extraite de $H$ est de déterminant non nul. En effet on a :

$$det(\mathrm{B}_w) = \beta^{(j_1+\cdots+j_w)b} det \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \beta^{j_1} & \beta^{j_2} & \cdots & \beta^{j_w} \\ \vdots & \vdots & & \vdots \\ \beta^{j_1(w-1)} & \beta^{j_2(w-1)} & \cdots & \beta^{j_w(w-1)} \end{pmatrix}.$$

$$= \beta^{(j_1+\cdots+j_w)b} \prod_{1\leq i\leq k\leq w} (\beta^{j_i} - \beta^{j_k}) \neq 0,$$

car pour tout $i, k \in \{1, 2, \ldots, w\}$ : $\beta^{j_i} \neq \beta^{j_k}$. Ainsi le résultat suit.

**Définition 3.6.**

Un **code B.C.H** de longueur $n$ et de distance construite $\delta$, est un code cyclique de longueur $n$, construit sur le corps $\mathbb{F}_{2^r}$ corps des racines nièmes de l'unité sur $\mathbb{F}_2$, où $r$ est l'ordre multiplicatif de $2$ modulo $n$, dont le polynôme générateur est le produit (sans répétition de facteur) des polynômes minimaux de $\beta^b$, $\beta^{b+1}$, …, $\beta^{b+\delta-2}$ où $\beta \in \mathbb{F}_{2^r}$ est une racine nième primitive de l'unité, $b$ un entier strictement positif.

Il existe deux cas importants:

Si $b$=1, le code B.C.H est appelé **code B.C.H** au **sens strict**.

Si la longueur du code $n = 2^r - 1$, $r$ étant un entier positif, on parle de **code B.C.H primitif**.

### 3.4.2 Construction d'un code B.C.H

La réalisation d'un code B.C.H ayant une capacité de correction $e$, peut se faire de la manière suivante :

1. Construire le corps $\mathbb{K}=\mathbb{F}_{2^r}$ des racines nièmes de l'unité.
2. Déterminer à l'aide d'un polynôme primitif $M_\alpha$ les éléments de $\mathbb{K}$.
3. Choisir $(\delta - 1 = 2e)$ racines de puissances successives $\beta^b$, $\beta^{b+1}$, …, $\beta^{b+\delta-2}$ du générateur $g(X)$.
4. Construire $g(X)$ le *PPCM* des polynômes minimaux des racines $\beta^b$, $\beta^{b+1}$, …, $\beta^{b+\delta-2}$, $c.\text{à}.d.\, g(X) = PPCM\left(M_{\beta^b}, M_{\beta^{b+1}}, \ldots, M_{\beta^{b+\delta-2}}\right)$.

**Exemples 3.9.**

Nous voulons construire un code B.C.H au sens strict de longueur égale à $n$=5 et de distance construite égale à $\delta$ =3 sur $\mathbb{F}_2$. Remarquons que nous sommes en présence d'un code B.C.H qui n'est pas primitif. Calculons en premier lieu les classes cyclotomiques de 2 modulo 5. Nous obtenons : $C_0$= {0} ,$C_1$= {1,2,3,4}. Donc $X^5$-1=($X$-1)( $X$-$\beta$)($X$-$\beta^2$)($X$-$\beta^3$)($X$-$\beta^4$).

Notre choix de $\delta = 3$, nous permet de prendre comme générateur le polynôme:

$$g(X)=(X-\beta)(X-\beta^2)(X-\beta^3)(X-\beta^4),$$

où β est une racine $5^{\text{ième}}$ primitive de l'unité. Comme le plus petit entier $r$ satisfaisant $5/2^r$-1 est $r$=4. On en déduit que $\beta \in \mathbb{F}_{16}$. De plus comme $s = \frac{2^r-1}{5}$=3, on en déduit que l'on peut prendre $\beta = \alpha^3$ où α est un élément primitif de $\mathbb{F}_{16}$ .On obtient alors :

$$g(X)=(X-\alpha^3)(X-\alpha^6)(X-\alpha^9)(X-\alpha^{12}),$$

en utilisant la construction de $\mathbb{F}_{2^4}$, ou encore $g(X)=\frac{X^5-1}{X-1}$, nous aurons :

$$g(X)=1+X+X^2+X^3+X^4.$$

Il est intéressant de remarquer que nous avons $\delta = 2e + 1 = 3$, comme $w[g(X)]$=5, on en déduit que la distance $d$=5.

**Exemple 3.10.**

Pour construire un code cyclique de longueur $n$=7, de capacité $e$=1, on choisit un code BCH binaire, de polynôme générateur $g(X)$ qui admet deux racines de puissances successives. On prend par exemple, $g(X) = X^3 + X + 1$, qui admet $\alpha$, $\alpha^2$ et $\alpha^4$, comme racines dans le corps des racines 7emes de l'unité $\mathbb{K}=\mathbb{F}_{2^3}=\mathbb{F}_8$, dont deux entre eux $\alpha$, $\alpha^2$ sont de puissances successives. Donc le code $C$ admet comme matrice de contrôle la matrice suivante :

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} \end{pmatrix}.$$

**Exemple 3.11.**

Nous chercherons à construire un code de longueur $n$ =15 et qui peut corriger jusqu'à $e$=2 erreurs c.à.d. avec une distance au moins égale à $\delta = 2e + 1$ =5. Pour cela, on choisit un code BCH dont le générateur admet au moins 4 racines successive. On commence par factoriser le polynôme $X^{15}$-1.

En utilisant les polynômes cyclotomiques, on obtient la décomposition du polynôme $X^{15}$-1 en produit de polynômes irréductibles sur $\mathbb{F}_2$ comme suit :

$$X^{15}\text{-}1=(X-1)(X^2+X+1)(X^4+X+1)(X^4+X^3+1)(X^4+X^3+X^2+X+1).$$

| Polynômes | Racines |
|---|---|
| $X-1$ | 1 |
| $X^2+X+1$ | $\alpha^5, \alpha^{10}$ |
| $X^4+\mathrm{X}+1$ | $\alpha, \alpha^2, \alpha^4, \alpha^8$ |
| $X^4+X^3+1$ | $\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$ |
| $X^4+X^3+X^2+X+1$ | $\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$ |

Table Facteurs irréductible de $X^{15}-1$ et leurs racines.

En combinant ces polynômes, on obtient des codes de distance et de dimension différentes.

Pour notre cas on prend $g(X) = (X^4 + X + 1)(X^4 + X^3 + X^2 + X + 1)$ qui admet $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^9, \alpha^{12}$ comme racines et parmi eux ils existent 4 racines $(\alpha, \alpha^2, \alpha^3, \alpha^4)$ de puissances successives, donc on peut prendre la distance *d* au moins égale à 5.

### 3.4.3 Codes Reed-Solomon

Les codes de Reed-Solomon forment un sous-ensemble de l'ensemble des codes cycliques. En fait, il s'agit de la sous-classe la plus importante des codes BCH. Ce sont de plus des codes M.D.S donc optimaux où ils nécessitent le minimum de redondance pour une capacité de correction fixée.

L'article sur les codes de Reed-Solomon a été soumis par Irving Reed et Gustave Solomon au Journal of the Society for Industrial and Applied Mathematics, le 21 janvier 1959 et a été publié en juin 1960 sous le titre « Polynomial Codes over Certain Finite Fields ».

Les code de Reed-Solomon sont les plus utilisés en pratique, ils sont utilisés dans la sauvegarde des données, par exemple pour les CD, DVD, dans la communication mobile, les réseaux sans fils (wireles), les communications satellitaires, les codes à barres bidimensionnels, la télévision et radio numériques ainsi que les modems ADSL.

**Définition 3.7.**

Soit $r \geq 2$ .Un **code de Reed-Solomon** de longueur $n = 2^r - 1$ est un code B.C.H primitif sur le corps de Galois $\mathbb{K} = \mathbb{F}_{2^r}$.

**Remarque 3.2.**

Tous les éléments non nuls du corps $\mathbb{K} = \mathbb{F}_{2^r}$ sont racines du polynôme $X^{2^r-1} - 1$. En conséquence, la décomposition sur $\mathbb{F}_{2^r}$ de $X^{2^r-1} - 1$ est la suivante:

$$X^{2^r-1} - 1 = \prod_{u \in \mathbb{F}_{2^r} - \{0\}} (X - u).$$

Si $\alpha$ est une racine primitive du corps $\mathbb{F}_{2^r}$, on obtient :

$$X^{2^r-1} - 1 = (X - 1)(X - \alpha)\dots\left(X - \alpha^i\right)\dots\left(X - \alpha^{2^r-2}\right).$$

Le générateur de degré *t* d'un code de Reed-Solomon est donc de la forme :

$$g(X) = (X - \alpha^i)(X - \alpha^{i+1})\dots(X - \alpha^{i+t-1}).$$

Qui admet *t* racines de puissances successives $\alpha^i$, $\alpha^{i+1}$, …, $\alpha^{i+t-1}$.

Pour un tel générateur, le code correspondant a pour dimension $k = 2^r - 1 - t$, de distance construite $\delta = t + 1$ .

**Proposition 3.5.**

Le code Reed-Solomon a pour paramètres :

- Longueur : $n = 2^r - 1$.
- Dimension : $k = 2^r - 1 - t$.
- Poids minimum : $d = t + 1 = n - k + 1$.

**Preuve.**

La longueur *n* et le dimension *k* viennent de la définition du code R-S. Pour la distance, on a d'une part d'après le borne de singleton $d \leq n\text{-}k\text{+}1\text{=}t\text{+}1$, et d'autre part d'après le théorème de la distance construite $d \geq \delta\text{=}t\text{+}1$ et donc $d\text{=}t\text{+}1$.

**Exemple 3.12.**

Soit $\mathbb{K} = \mathbb{F}_8$, la longueur de code R-S sur $\mathbb{K}$ est $n = 2^3 - 1 = 7$.

Construisons un code R-S qui corrige $e = 1$ erreur. Donc le générateur $g(X)$ est de degré $t = \delta - 1 = 2e = 2$, la distance $d$=3 et la dimension du code est $k = n - t = 5$. On peut prendre un code R-S au sens strict donc : $g(X) = (X - \alpha)(X - \alpha^2)$. Le polynôme primitif est le polynôme suivant : $M_\alpha(X) = X^3 + X + 1$, donc on aura : $\alpha^3 = \alpha + 1$, par la suite

$$g(X) = X^2 + (\alpha + \alpha^2)X + \alpha^3 = X^2 + \alpha^4 X + \alpha^3.$$

Le code *C* admet comme matrice génératrice la matrice $G$ :

$$G = \begin{pmatrix} \alpha^3 & \alpha^4 & 1 & 0 & 0 & 0 & 0 \\ 0 & \alpha^3 & \alpha^4 & 1 & 0 & 0 & 0 \\ 0 & 0 & \alpha^3 & \alpha^4 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha^3 & \alpha^4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \alpha^3 & \alpha^4 & 1 \end{pmatrix}.$$

**Exemple 3.13.**

Soit *C* le code R-S au sens strict de longueur *n=7* et corrigeant *e*=2 erreurs sur le corps de Galois $\mathbb{F}_8 = \{0, \alpha^i / 0 \leq i \leq 6\}$ et donc de polynôme générateur admet $\delta - 1$=t=4 racines successives $\alpha, \alpha^2, \alpha^3, \alpha^4$ est donné par :

$$g(X)=( X - \alpha)(X - \alpha^2)(X - \alpha^3)(X - \alpha^4).$$

Le développement de g(*X*) donne :

$$g(X)=X^4 + (\alpha^4 + \alpha^3 + \alpha^2 + a)X^3 + (\alpha^6 + \alpha^4 + \alpha^3 + 1)X^2 + (\alpha^6 + \alpha^2 + \alpha + 1)X + \alpha^3$$

En utilisant la construction du corps $\mathbb{F}_8$ on trouve :

$$g(X)=X^4 + \alpha^3 X^3 + X^2 + \alpha X + \alpha^3.$$

Une des matrices génératrices du code *C* est donc la suivante :

$$G=\begin{pmatrix} \alpha^3 & \alpha & 1 & \alpha^3 & 1 & 0 & 0 \\ 1 & \alpha^3 & \alpha & 1 & \alpha^3 & 1 & 0 \\ 0 & 0 & \alpha^3 & \alpha & 1 & \alpha^3 & 1 \end{pmatrix}.$$

**Exemple 3.14.**

Codes R-S pour la sonde spatiale Mariner 10.

Le 3 novembre 1973, la sonde spatiale Mariner 10 est lancée avec succès, elle avait pour mission le survol de la planète Vénus et de la planète Mercure.

Types de capteurs utilisés : deux caméras moyen angle avec enregistreur numérique, radiomètre infrarouge, plasma solaire, particules chargées, champs magnétiques, spectromètre à ultraviolets, occultation radio et mécanique céleste.

Après le survol de Vénus, la sonde se dirige vers Mercure. Elle réussit à réaliser 3 passages, accumulant de nombreuses photos d'une qualité sans précédent, et permettant de comprendre certains mystères de Mercure. Mariner 10 devient ainsi la première sonde à observer Mercure.

Le code utilisé par la NASA pour la sonde spatiale Mariner 10 est un code Reed-Solomon dont les parametres sont les suivants :

Le corps $\mathbb{K} = \mathbb{F}_{2^8} = \mathbb{F}_{256}$,

La longueur *n*=$2^8$-1= 255,

Le générateur $g(X) = \prod_{112 \leq i \leq 143}(X - \alpha^i)$, avec *t* = d°( $g(X)$) = 32,

La dimension $k=n-t=223$,

La distance $d=t+1=33$,

La capacité de correction $e = \left[\frac{t}{2}\right] = 16$.

La sonde spatiale Mariner 10

# Chapitre 4 Décodage des codes cycliques.

## Introduction

Les méthodes de décodage des codes cycliques peuvent être classées en deux grandes catégories : les méthodes à base de syndromes et les méthodes à base d'algorithmes de recherche. Les méthodes à base de syndromes exploitent les syndromes, qui sont des indicateurs de l'existence et de la localisation d'erreurs dans le message reçu. Les syndromes sont obtenus en comparant le message reçu avec les motifs cycliques prédéfinis. Les méthodes à base d'algorithmes de décodage tels que l'algorithme de Meggitt, l'algorithme de piégeage d'erreurs, l'algorithme de Transformation de Fourier Discrète et l'algorithme de Peterson-Gorenstein-Zierler sont utilisés pour déterminer les erreurs et les localiser dans le message, permettant ainsi leur correction.

En résumé, les méthodes de décodage des codes cycliques jouent un rôle crucial dans la garantie de la fiabilité des données transmises et stockées. Elles utilisent des techniques mathématiques avancées pour détecter et corriger les erreurs, améliorant ainsi les performances des systèmes de communication et de stockage. Que ce soit en utilisant des méthodes basées sur les syndromes ou des algorithmes de recherche, le décodage des codes cycliques demeure un domaine de recherche essentiel pour assurer l'intégrité des données dans un large éventail d'applications technologiques.

## 4.1 Décodage par syndrome polynômial.

Soit le code cyclique $C(n,k,d)$ sur le corps $\mathbb{F}_q = \mathbb{F}_{p^r}$, corps des racines *nièmes* de l'unité sur $\mathbb{F}_p$ , de polynôme générateur $g(X)$ avec $\deg(g(X)) = t$. Soit $e$ la capacité de $C$ et $y(X) \in \mathbb{F}_q[X]/< X^n - 1 >$. On dit que $y(X)$ est un mot reçu dont l'erreur est $\varepsilon(X)$, si $\omega(\varepsilon(X)) \leq e$ et s'il existe $c(X) \in C$ tel que $y(X) = c(X) + \varepsilon(X)$.

### 4.1.1 Syndrome polynômial.

**Définition 4.1.**

On appelle **syndrome polynômial** (ou **syndrome**) d'un mot $y(X) \in \mathbb{F}_q[X]/< X^n - 1 >$, qu'on note $S\big(y(X)\big)$, le reste de la division Euclidienne de $y(X)$ par $g(X)$ dans $\mathbb{F}_q[X]$.

**Proposition 4.1.**

Soit un mot $y(X) \in \mathbb{F}_q[X]/< X^n - 1 >$ alors $y(X) \in C$ si et seulement si $g(X)$ divise $y(X)$ dans $\mathbb{F}_q[X]$ c.à.d. $y(X) \in \theta(C) \Leftrightarrow S(y(X)) = 0$.

**Preuve.** Soit $y(X) \in \theta(C) \Leftrightarrow \exists\, y(X) \in \mathbb{F}_q[X] : y(X) = q(X)g(X)$

$\Leftrightarrow$ le reste de la division Euclidienne de $y(X)$ par $g(X)$=0

$\Leftrightarrow S(y(X)) = 0.$

**Définition 4.2.**

Un **registre à décalage** est une chaines d'éléments de mémoire. Un **décalage linéaire** transfère le contenu de chacune des cellules vers la cellule qui la suit immédiatement. Après un décalage le contenu de la première cellule est zéro.

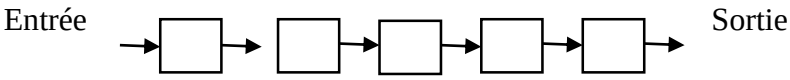



Fig 4.1 registre à décalage linéaire

Par convention les décalages s'effectuent de la gauche à la droite.

Dans la pratique, les calculs dans $\mathbb{F}_q[X]$, en particulier les divisions , s'effectuent au moyen de **registres à décalages circulaire** comme dans le schéma suivant qui représente un circuit à décalage circulaire de la divisions Euclidienne d'un polynôme $y(X) = \sum_{i=0}^{n} y_i X^i$ par un polynôme $g(X) = \sum_{i=0}^{t} g_i X^i$:

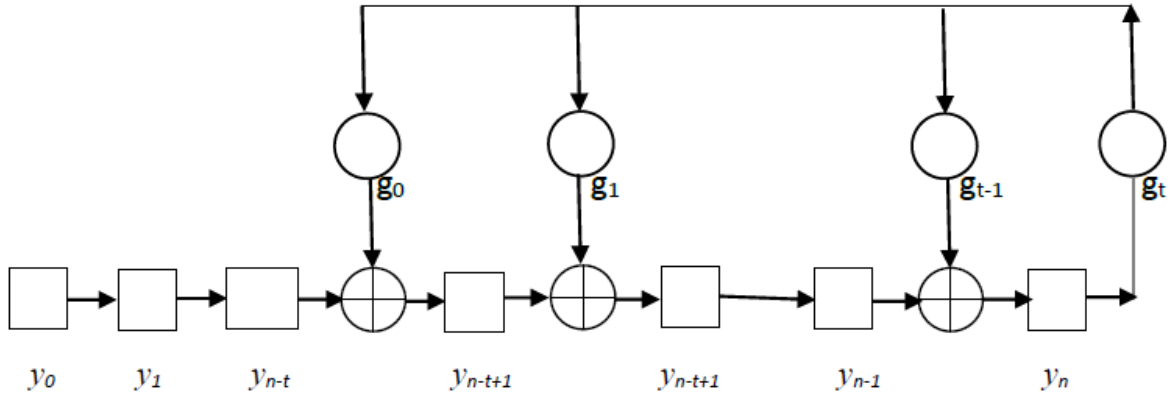



Fig 4.2 Circuit à décalage circulaire de la divisions Euclidienne.

Un bref aperçu sur les registres voir l'annexe (Appendice page 123).

### 4.1.2 Algorithme de décodage par la méthode de syndrome polynomial.

Soit $C(n,k,d)$un code cyclique sur $\mathbb{F}_q^n$ avec une capacité de correction égale à $e$. Si $y(X)$ est le mot reçu, alors l'algorithme de décodage est le suivant :

- Calcul du syndrome du mot reçu $S(y(X))$ avec un registre à décalage circulaire.
- Trouver l'erreur $\varepsilon(X)$ qui correspond au syndrome $S(y(X))$ et de poids $w(\varepsilon(X)) \leq e$.
- Soustraction de l'erreur au mot reçu, le mot envoyé est $c(X) = y(X) - \varepsilon(X)$.

**Exemple 4.1.**

Soit le code cyclique $C(7,4)$ sur $\mathbb{F}_2$ de polynôme générateur $g(X) = X^3 + X^2 + 1$, et soit le mot reçu $y = 0011001$ en représentation polynomiale $y(X) = \sum_{i=0}^{6} y_i X^i = X^6 + X^3 + X^2$.

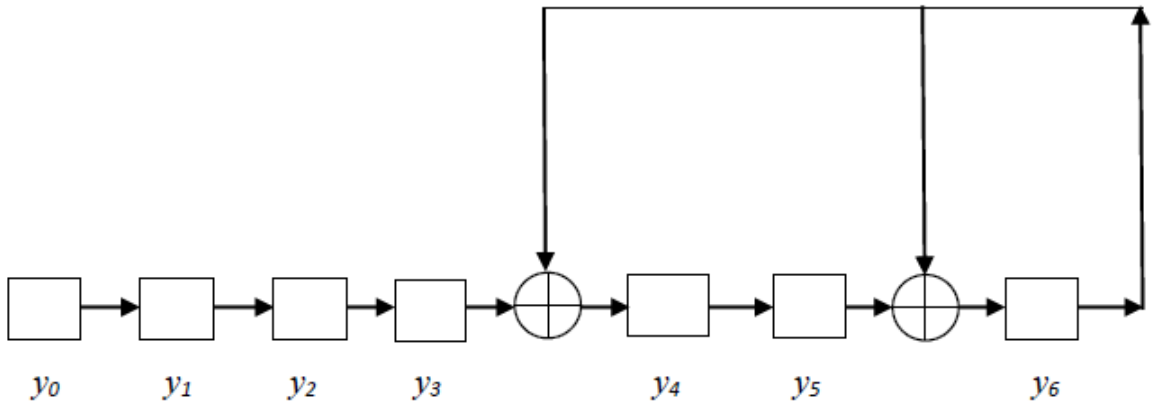



Fig 4.3 Registre à décalages circulaire

| Nombre décalage | $a_0$ | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ |
|---|---|---|---|---|---|---|---|
| 0 | O | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Reste | | | | | $r_0$ | $r_1$ | $r_2$ |

Tableau 4.4 Tableau des syndromes.

Alors : $S(y(X)) = r_2X^2 + r_1X + r_0 = X^2 + X + 1$.

Les syndromes des erreurs de poids 1, sont calculés dans le tableau suivant :

| $\varepsilon(X)$ | $S(\varepsilon(X))$ |
|---|---|
| $X^6$ | $X^2 + X$ |
| $X^5$ | $X + 1$ |
| $X^4$ | $X^2 + X + 1$ |
| $X^3$ | $X^2 + 1$ |
| $X^2$ | $X^2$ |
| X | $X$ |
| 1 | 1 |

Tableau 4.5 Tableau des syndromes-

Cherchons l'erreur $\varepsilon(X)$ de poids $w(\varepsilon(X)) = 1$ et de syndrome $S(\varepsilon(X)) = X^2 + X + 1$ dans le tableau des syndromes, on trouve le mot erreur est $\varepsilon(X) = X^4$ et le mot (polynôme) envoyé est : $C(X) = y(X) + \varepsilon(X) = X^6 + X^4 + X^3 + X^2$, et le mot envoyé $c = 0011101$.

## 4.2 Méthode de décodage de Meggitt.

Cette méthode tire son nom de l'ingénieur britannique Jack K. Meggitt, qui a développé cette approche dans les années 1960. Elle s'applique aux codes cycliques binaires, mais elle peut se généraliser au cas non binaire. L'idée de base consiste en l'utilisation de la cyclicité du code pour reteindre la table des syndromes et permettre des calculs récursifs. Le décodeur de Meggitt effectue un décodage symbole par symbole. On corrige d'abord une composante erronée du mot reçu au moyen de la méthode décrite ci-dessous, puis on applique de nouveau la méthode au nouveau mot reçu ainsi obtenu.

La méthode de Meggitt se distingue par sa capacité à détecter et à corriger plusieurs erreurs dans les codes cycliques, ce qui en fait une approche assez performante pour les environnements à forte distorsion. Cependant, il convient de noter que la méthode de Meggitt peut être complexe à mettre en œuvre en raison des calculs impliqués et de la nécessité de manipuler les générateurs de syndromes.

Les opérations à effectuer sont le shift et le calcul de syndrome on peut les réaliser au moyen de registres à décalage circulaire.

Un autre avantage de cette méthode réside dans le fait qu'on remplace le tableau de déchiffrement qui comporte tous les mots erreurs et tous les syndromes, par un autre tableau où ne figurent que les syndromes des mots erreurs dont le dernier symbole est erroné. On gagne ainsi beaucoup d'espace mémoire et de temps.

### 4.2.1 Suite des syndromes polynomiaux

La proposition suivante montre que les $j^{ieme}$ shifts du mot reçu et de l'erreur correspondante ont le même syndrome polynomial.

**Proposition 4.2.**

Soit $\varepsilon(X)$ le mot erreur du mot reçu $y(X)$. Alors, pour tout entier j, $0 \leq j \leq n-1$ :

1. le mot $X^j y(X)$) est un mot reçu dont l'erreur est $X^j \varepsilon(X)$.
2. S $(X^j \varepsilon(X))$ = S $(X^j y(X))$.

**Preuve.**

1. De $y(X)$ =c$(X)$+$\varepsilon(X)$, avec $c(X) \in \theta(X)$, on déduit $X^j y(X) = X^j c(X) + X^j \varepsilon(X)$. Le code C étant cyclique, on sait que $X^j c(X) \in \theta(X)$. D'autre part $w(X^j \varepsilon(X)) = w(\varepsilon(X))$, car la multiplication par $X^j$ ne modifie pas le poids d'un mot. L'égalité précédente montre que $X^j \varepsilon(X)$ est le mot erreur du mot reçu $X^j y(X)$.
2. Il existe c$(X)$ multiple de g(x) dans $\mathbb{F}_2[X]/<X^n - 1>$ tel que $y(X) = c(X) + \varepsilon(X)$, on obtient donc dans $\mathbb{F}_2[X]/<X^n - 1>$ une relation de la forme $X^j y(X) = X^j c(X) + X^j \varepsilon(X)$. Ceci implique, dans $\mathbb{F}_2[X]$, une égalité de la forme $X^j y(X) = X^j c(X) + X^j \varepsilon(X) + b(X)(X^n - 1)$. puisque $g(X)$ divise $(X^n - 1)$ dans $\mathbb{F}_2[X]$, on voit que $X^j y(X) = X^j \varepsilon(X)$ modulo $g(X)$, ce qui montre que $X^j y(X)$ et $X^j \varepsilon(X)$ ont le même reste dans la division par $g(X)$, c.à.d. le même syndrome.

**Remarque 4.1.**

On voit donc d'après (b), que si l'on trouve S $(X^j$y$(X))$ dans une table de syndrome indiquant l'erreur correspondante, on peut retrouver $X^j \mathcal{E}(X)$ et donc aussi l'erreur $\mathcal{E}(X)$.

La proposition suivante montre comment on peut calculer S $(X^j$y$(X))$ à partir de S (y$(X)$) de manière récursive.

**Proposition 4.3.**

Avec les notations de la proposition précédente, soit $(S_j(X))_{j\in\mathbb{N}}$ la suite des polynômes de $\mathbb{F}_2[X]/< X^n-1 >$ définie par :

$$\begin{cases} S_0(X) = S(y(X)) \\ S_{j+1}(X) = S(XS_j(X)) \end{cases}$$

Alors pour tout entier $j$, $0 \le j \le n-1$, on a : $S_j(X) = S(X^j y(X))$.

**Preuve.**

Pour $j = 0$, la propriété est vraie par définition : $S_0(X) = S(y(X)) = S(X^0 y(X))$.

Démontrons la pour $j = 1$,

Soit $q(X)$ le quotient de la division de $y(X)$ par $g(X)$ dans $\mathbb{F}_2[X]$. D'après la définition du syndrome, on obtient : $y(X) = g(X)q(X) + S(y(X))$, ceci implique

$$Xy(X) = Xg(X)q(X) + XS(y(X)) \ldots(1)$$

Soit l'application $\varphi$: $\mathbb{F}_2[X] \to \mathbb{F}_2[X]/< X^n-1 >$ qui associe chaque polynôme de $\mathbb{F}_2[X]$ avec son reste de la division Euclidienne par $X^n-1$.

Soit $q'(X)$ le quotient de la division Euclidienne de $XS(y(X))$ par $X^n-1$ dans $\mathbb{F}_2[X]$.

On trouve : $XS(y(X)) = q'(X)(X^n-1) + \varphi(XS(y(X)))$.

Remplaçons dans (1); $Xy(X) = Xg(X)q(X) + q'(X)(X^n-1) + \varphi(XS(y(X)))$.

Puisque g(x) divise $X^n-1$, on obtient : $Xy(X) = \varphi(XS(y(X)))mod g(X)$.

Soit $q''(X)$ le quotient de la division Euclidienne de $Xy(X)$ par $X^n-1$,

$Xy(X) = q''(X)(X^n-1) + \varphi(Xy(X))$. Comme $g(X)$ divise $X^n-1$, on obtient :

$Xy(X) = \varphi(Xy(X))mod g(X)$. D'où $\varphi(XS(y(X))) = \varphi(Xy(X))\, mod g(X)$.

Ceci montre que $\varphi(XS(y(X)))$ et $\varphi(Xy(X))$ ont le même reste de division par $g(X)$ et donc le même syndrome. Cela signifie que $XS(y(X))$ et $Xy(X)$ calculés modulo $X^n-1$ ont le même syndrome. i.e. $S(XS(y(X))) = S(Xy(X))$. D'où $S_1(X) = S(Xy(X))$. Supposons que le

résultat est vrai pour $j = k$, c.à.d. $S_k(X) = S(X^k y(X))$, et montrons le pour $j = k + 1$.
D'après la définition de la suite, on a :

$$S_{k+1}(X) = S\big(XS_k(X)\big) = S\left(XS\Big(X^k y(X)\Big)\right) = S\left(XS\big(y_k(X)\big)\right) \text{ avec } y_k(X) = X^k y(X)\,.$$

En appliquant le résultat pour $j = 1$ à $y_k(X)$, on trouve

$S(XS(y_k(X))) = S\big(Xy_k(X)\big) = S(X.X^k y(X)) = S(X^{k+1} y(X))$. Cela achève la démonstration par récurrence.

### 4.2.2 Algorithme de décodage de Meggitt

Soit $(T)$ la table des syndromes des erreurs dont la composante d'indice $n-1$ est erronée. Soit $c(X)$ le mot envoyé, $y(X)$ le mot reçu , et $\varepsilon(X)$ le mot erreur avec $\omega(\varepsilon(X)) \leq e$.

La suite $S_i(X)$ est définie comme dans la Proposition 4.3. L'algorithme de décodage de Meggitt est le suivant :

1. Calcul de $S(y(X))$.
2. Si $S\big(y_k(X)\big) = 0$ alors $y(X) = c(X)$ et l'algorithme se termine.
3. Sinon ;

- On cherche le plus petit entier non nul $j$ tel que $S_j(X)$ se trouve dans la table $(T)$.
- Corriger la composante d'indice $n-1-j$ de $y(X)$, soit $y'(X)$, le nouveau mot obtenu.

4. Repartir au début de l'algorithme avec $y'(X)$.

**Exemple 4.2.**

Soit $C(7,4,3)$ le code de Hamming 1-correcteur de polynôme générateur $g(X) = X^3 + X + 1$. La table $(T)$ des syndromes des erreurs dont la composante d'indice $6$ est égale à $1$ se réduit au tableau suivant :

| Erreur | $X^6$ |
|---|---|
| Syndrome | $X^2 + 1$ |

Tableau 4.6 Tableau des syndrome $(T)$

Soit $y(X) = X^6 + X^5 + X^4$,le mot reçu, donc le syndrome de $y(X)$ est égale à $S(y(X)) = X^2$ ne figure pas dans la table $(T)$, On cherche le plus petit entier non nul $j$ tel que $S_j(X) \in (T)$, c.à.d. tel que $S_j(X) = S\left(X^j y(X)\right) = X^2 + 1$, on trouve que $j = 4$. Il y a donc une erreur en position $n - 1 - j = 7 - 1 - 4 = 2$. Avec l'hypothèse que la capacité de correction égale à 1 n'est pas dépassé, l'erreur est $\varepsilon(X) = X^2$ et le mot envoyé est $c(X) = y(X) + \varepsilon(X) = X^6 + X^5 + X^4 + X^2$.

**Exemple 4.3.**

Soit $C(15, 7, 5)$ le code cyclique avec polynôme générateur $g(X) = 1 + X^4 + X^6 + X^7 + X^8$. Alors la liste des polynômes erreurs $\varepsilon(X)$ avec $\omega\left(\varepsilon(X)\right) \leq e = 2$ et leurs syndromes est la suivante :

| $\varepsilon(X)$ | $S(\varepsilon(X))$ |
|---|---|
| $X^{14}$ | $X^3 + X^5 + X^6 + X^7$ |
| $X^{14} + X^{13}$ | $X^2 + X^3 + X^4 + X^7$ |
| $X^{14} + X^{12}$ | $X + X^4 + X^6 + X^7$ |
| $X^{14} + X^{11}$ | $1 + X^2 + X^4 + X^5 + X^6 + X^7$ |
| $X^{14} + X^{10}$ | $X + X^2 + X^3$ |
| $X^{14} + X^9$ | $1 + X + X^3 + X^4 + X^7$ |
| $X^{14} + X^8$ | $1 + X^3 + X^4 + X^5$ |
| $X^{14} + X^7$ | $X^3 + X^5 + X^6$ |
| $X^{14} + X^6$ | $X^3 + X^5 + X^7$ |
| $X^{14} + X^5$ | $X^3 + X^6 + X^7$ |
| $X^{14} + X^4$ | $X^3 + X^4 + X^5 + X^6 + X^7$ |
| $X^{14} + X^3$ | $X^5 + X^6 + X^7$ |
| $X^{14} + X^2$ | $X^2 + X^3 + X^5 + X^6 + X^7$ |
| $X^{14} + X$ | $X + X^3 + X^5 + X^6 + X^7$ |
| $X^{14} + 1$ | $1 + X^3 + X^5 + X^6 + X^7$ |

Tableau 4.7 Tableau des syndromes $(T)$ pour le code $C(15, 7, 5)$

Soit $y(X) = X^{12} + X^{10} + X^9 + X^7 + X^4 + 1$ le mot reçu.

En utilisant un registre à décalage circulaire, on calcule le syndrome du mot reçu on trouve : $S(y(X)) = X^5 + X^4 + X^3 + X^2 + X$.

On remarque que $S(y(X))$ ne figure pas dans la table, cherchons le plus petit entier $j$ tel que $S_j(X)$ soit dans la table $(T)$. On trouve $j = 2$, car après calcul on trouve

$$S(X^2y(X)) = X^7 + X^6 + X^5 + X^4 + X^3 = S(X^{14} + X^4).$$

L'erreur associée au mot $X^2y(X)$ est $X^2\varepsilon(X) = X^{14} + X^4$, d'où le mot erreur associée au mot $y(X)$ est $\varepsilon(X) = X^{12} + X^2$. D'où $c(X) = y(X) + \varepsilon(X) = X^{10} + X^9 + X^7 + X^4 + X^2 + 1$.

## 4.3 Décodage par piégeage d'erreur

Le décodage par piégeage d'erreur, également connu sous le nom de "Error Trapping Decoding" en anglais, est une approche utilisée pour décoder les codes correcteurs d'erreurs, y compris les codes cycliques. Cette méthode vise à détecter et à corriger les erreurs dans les données en identifiant et en isolant les erreurs potentielles à l'aide de techniques de syndromes et de calculs itératifs. Le décodage par piégeage d'erreur est particulièrement efficace pour les codes cycliques, car il exploite leurs propriétés spécifiques pour améliorer la correction d'erreurs.

### 4.3.1 Principe de la méthode de piégeage d'erreurs

Soit $C(n, k)$ un code cyclique $e$-correcteur sur le corps $\mathbb{F}_q$, de polynôme générateur $g(X)$. Supposons que $\mathrm{c}(X) \in C$ le mot transmis et $\mathrm{y}(X) = \mathrm{c}(X) + \varepsilon(X)$ est le mot reçu, où $\varepsilon(X)$ est le mot erreur, avec le poids $\omega(\varepsilon(X)) \leq e$. La méthode de décodage par piégeage d'erreur est une modification de la méthode de Meggitt, il s'agit de déplacer par décalage circulaire, c'est-à-dire « piéger » en quelque sorte, les composantes non nulles de l'erreur sur certaines positions. On considère un code binaire (on peut généraliser au cas non binaire), et on suppose comme cité ci-dessus, que le nombre d'erreurs ne dépasse pas la capacité de correction $e$.
Le principe du décodage par piégeage d'erreur s'appuie sur les résultats suivants :

**Lemme 4.1.**

Soit $\varepsilon(\mathrm{X})$ le mot erreur du mot reçu $\mathrm{y}(X)$, Si $\mathrm{d}^\circ(\varepsilon(\mathrm{X})) \leq \mathrm{n} - \mathrm{k} - 1$ alors :
$\varepsilon(X) = \mathrm{S}(\mathrm{y}(\mathrm{I}))$.

**Preuve.**

Dans $\mathbb{F}_\mathrm{q}[X]/< X^\mathrm{n} - 1 >$ on a $\mathrm{y}(X) = \mathrm{c}(X) + \varepsilon(\mathrm{X})$, avec $\varepsilon(X) \in \mathrm{C}$, soit encore dans $\mathbb{F}_\mathrm{q}[X]$ $\mathrm{y}(X) = \mathrm{a}(X)\mathrm{g}(X) + \varepsilon(X) + \mathrm{b}(X)(X^\mathrm{n} - 1)$. Puisque $\mathrm{g}(X)$ divise $X^\mathrm{n} - 1$, on trouve $\mathrm{y}(X) = \mathrm{d}(X)\mathrm{g}(X) + \varepsilon(X)$.

Si d°( $\varepsilon(X)) \leq n - k - 1$, alors d'après l'unicité de reste dans la division par $g(X)$, on obtient $S\big(y(X)\big) = \varepsilon(X)$.

**Lemme 4.2.**

Soit $\varepsilon(X)$ le mot erreur du mot reçu $y(X)$ , alors $\omega(S(y(X)) \leq e$ si et seulement si $S\big(y(X)\big) = \varepsilon(X)$.

**Preuve.**

La division dans $\mathbb{F}_q[X]$ de $y(X)$ par $g(X)$ s'exprime par $y(X) = g(X)a(X) + \varepsilon(X)$.
Les conditions sur le degré des polynômes intervenant dans cette égalité, font que celle si également vérifiée dans $\mathbb{F}_q[X]/< X^n - 1 >$. La décomposition d'un mot reçu comme somme d'un mot du code et d'un mot de poids inférieur ou égal à e est unique. Donc si $\omega(S(y(X)) \leq e$, alors $S\big(y(X)\big) = \varepsilon(X)$. Réciproquement si $S\big(y(X)\big) = \varepsilon(X)$, alors $\omega(S(y(X)) \leq e$ puisque le poids de l'erreur est au plus $e$.

**Théorème 4.2.** Soit $\varepsilon(X)$ le mot erreur du mot reçu $y(X)$.
Si $j$ est un entier $0 \leq j \leq n - 1$ tel que $\omega(S(X^j y(X)) \leq e$, alors $\varepsilon(X) = X^{-j} S\left(X^j y(X)\right)$.

**Preuve.**

Soit $y_1(X) = X^j y(X)$, c'est le mot erreur du mot reçu dont l'erreur est $\varepsilon_1(X) = X^j \varepsilon(X)$.d'aprée le lemme 2.15.2, si $\omega(S(y_1(X) \leq e$, alors $S\big(y_1(X)\big) = \varepsilon_1(X)$, c.-à-d. $S\left(X^j y(X)\right) = X^j \varepsilon(X)$ donc $\varepsilon(X) = X^{-j} S(X^j y(X))$.

**Théorème 4.3.**

Soit $\varepsilon(X)$ le mot erreur du mot reçu $y(X)$. Si $\varepsilon(X) = X^j \varepsilon_1(X)$ avec d °( $\varepsilon_1(X)) \leq n - k - 1$, alors $\omega(S(X^{-j} y(X)) \leq e$.

**Preuve.**

D'après le lemme 2.15.1, comme d ° $\varepsilon_1(X) \leq n - k - 1$ alors $\varepsilon_1(X) = X^{-j}\varepsilon(X) = S(X^{-j} y(X))$. On pose $y_1(X) = X^{-j} y(X)$ donc $\varepsilon_1(X) = S(y_1(X)$ d'aprés le lemme 2.15.2 on trouve $\omega(S(y_1(X)) \leq e$ alors $\omega(S(X^{-j} y(X)) \leq I$.

### 4.3.2 Algorithme de décodage par piégeage d'erreur

Soit $y(X)$ le mot reçu, $\varepsilon(X)$ le mot erreur avec $\omega(\varepsilon(X)) \leq e$.

1. Calcul de $S(y(X))$.
2. Si $S(y(X)) = 0$ alors $\varepsilon(X) = 0$.
3. Sinon,
   - Si $\omega(S(y(X)) \leq e$ alors $\varepsilon(X) = S(y(X))$.
4. Sinon on cherche le plus petit entier j tel que $\omega(S\left(X^j y(X)\right)) \leq e$, alors $\varepsilon(X) = X^{-j}S(X^j y(X))$.
5. Le mot envoyé est $c(X) = y(X) - \varepsilon(X)$.

**Exemple 4.4.**

Soit le code $C(7,4,3)$ sur $\mathbb{F}_2$, Soit $g(X)$ le polynôme générateur, $g(X) = X^3 + X^2 + 1$,
Soit $y(X) = X^5 + X^3 + X$, le mot reçu. Le syndrome est le reste de la division $y(X)$ par $g(X)$.
On a : $y(X) = (X^3 + X^2 + 1)(X^2 + X) + X^2$, donc $S(y(X)) = X^2$ et $\omega\left(S(y(X))\right) = 1$. En supposant que la capacité d'erreurs n'est pas dépassée, l'erreur est $\varepsilon(X) = S(y(X))$ et le mot transmis est donc : $c(X) = y(X) - \varepsilon(X) = X^5 + X^3 + X^2 + X$.

**Exemple 4.5.**

Soit le code $C(7.4,3)$ sur $\mathbb{F}_2$. Soit le polynôme générateur $g(X) = X^3 + X + 1$.
Soit $y(X) = X^5 + X^4 + X^2$, le mot reçu. Le syndrome est le reste de la division de $y(X)$ par $g(X)$. On a : $y(X) = (X^3 + X + 1)(X^2 + X + 1) + (X^2 + 1)$ Donc $S(y(X)) = X^2 + 1$ et $\omega\left(S(y(X))\right) = 2$. Puisqu'on suppose que le poids de l'erreur est au plus 1, $S(y(X)) \neq \varepsilon(X)$.
Comme $S(Xy(X)) = 1$, alors le plus petit entier non nul j tel que $\omega\left(X^j S(y(X))\right) \leq 1$ est $j = 1$. Donc le mot erreur est $\varepsilon(X) = X^{-1}S(Xy(X)) = X^{-1} * 1 = X^{7-1} = X^6$.
Donc $\varepsilon(X) = X^6$, et le mot envoyé est $c(X) = y(X) - \varepsilon(X)$ est $c(X) = X^6 + X^5 + X^4 + X^2$.

## 4.4 Décodage algébriques des codes B.C.H

Le décodage algébrique des codes BCH est une méthode qui repose sur des concepts et des techniques algébriques avancées pour localiser et corriger les erreurs, en exploitant les propriétés caractéristiques des codes BCH.

Voici un aperçu du décodage algébrique des codes BCH :

1. **Propriétés des codes BCH** : Les codes BCH sont construits à partir de polynômes cyclotomiques et ont la propriété remarquable de posséder des distances minimales relativement élevées. Cela signifie qu'ils peuvent détecter et corriger un grand nombre d'erreurs. Ils sont souvent utilisés dans les systèmes où la fiabilité des données est cruciale, tels que les systèmes de stockage et les communications.
2. **Syndromes et générateurs de syndromes** : Les syndromes sont des vecteurs associés à des erreurs spécifiques dans les données reçues. Les générateurs de syndromes sont des polynômes qui permettent de calculer ces syndromes. Le processus de décodage commence par le calcul des syndromes à partir des données reçues.
3. **Localisation d'erreurs avec les polynômes locaux** : Une caractéristique clé des codes BCH est l'utilisation de polynômes locaux pour localiser les erreurs. Ces polynômes permettent de déterminer les positions où les erreurs sont susceptibles de se trouver. Les positions d'erreurs probables sont appelées "positions d'erreur candidates".
4. **Localisation précise d'erreurs avec les équations de localisation** : Les équations de localisation sont des relations algébriques qui associent les syndromes aux positions d'erreur candidates. En résolvant ces équations, il est possible d'obtenir des informations plus précises sur les positions d'erreurs et leurs valeurs.
5. **Correction d'erreurs avec les équations de correction** : Les équations de correction sont utilisées pour calculer les valeurs correctes des bits en fonction des positions d'erreurs identifiées. En appliquant ces équations, les erreurs peuvent être corrigées, ce qui permet de récupérer les données d'origine avec une grande précision.
6. **Itérations et raffinements** : Dans certaines situations, des itérations et des raffinements peuvent être nécessaires pour améliorer davantage la correction d'erreurs. Cela peut impliquer de répéter les étapes précédentes avec des informations mises à jour pour parvenir à une solution plus précise.

Le décodage algébrique des codes BCH nécessite une compréhension approfondie des propriétés algébriques des codes et des techniques de manipulation de polynômes. Cette méthode est généralement plus complexe que les méthodes de décodage basées sur les syndromes, mais elle offre une excellente capacité de correction d'erreurs, ce qui la rend très utile dans des contextes où la fiabilité des données est primordiale.

### 4.4.1 Syndrome, localisateur et polynôme localisateur

Soit $C(n,k)$ un code BCH, de générateur $g(X)$ admettant $\delta - 1 = 2e$ (où $e$ est la capacité de correction et $\delta$ la distance construite) racines successives $\beta^j$, $b \leq j \leq b + \delta - 2$, $b$ un entier non nul. Soit $v$ le poids de l'erreur c.à.d. $v = w(\varepsilon(X))$, avec $v \leq e$.

**Définition 4.3.**

Si $y = (y_0, y_1, \dots, y_{n-1})$ est le mot reçu, $\varepsilon = (\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{n-1})$ le mot erreur de poids *v* tel que $v = w(\varepsilon(X)) \leq e$, et $H$ la matrice de contrôle du code *C*. Alor le syndrome de $y$ est :

$S = y * H^t$ donné par :

$$S=\left(\sum_{i=0}^{n-1} y_i \beta^{bi}, \sum_{i=0}^{n-1} y_i \beta^{(b+1)i}, \dots, \sum_{i=0}^{n-1} y_i \beta^{(b+2e-1)i}\right),$$

$$s_j = \sum_{i=0}^{n-1} y_i \beta^{ji} = y(\beta^j), \quad b \leq j \leq b + \delta - 2.$$

Le syndrome se calcule en cherchant les valeurs de $y(X)$ pour les $\delta - 1$ racines successives $\beta^j$ du polynôme générateur $g(X)$.

**Définition 4.4.**

On appelle **localisateur de la position** $i$, l'élément $X_i$ défini par $X_i = \beta^i$, pour $0 \leq i \leq n - 1$.

On note $I$ l'ensemble des positions de l'erreur, $I = \{i, 0 \leq i \leq n-1, \varepsilon_i \neq 0\}$, alors

$v = \text{card}(I)$.

• On appelle **valeur de l'erreur** dans la position $i$, l'élément $Y_i$, tel que $Y_i = \varepsilon_i$.

**Définition 4.5.** On appelle **polynôme localisateur de l'erreur**, le polynôme σ(X) définie par :

$$\sigma(X) = \prod_{i \in I} (1 - X_i X)$$

Par définition du polynôme localisateur, ces racines ne sont que les inverses des localisateurs.

**Remarque 4.2.**

Soit y(x) un mot reçu avec une erreur ε(X) de poids $w(\varepsilon(X)) \leq e$, alors :

$y(X) = c(X) + \varepsilon(X)$, avec $c(X) \in C$. On a :

Pour tout $i \in \{b, \ldots, b+\delta-2\}$: $y(\beta^i) = c(\beta^i) + \varepsilon(\beta^i) = \varepsilon(\beta^i)$, avec $\beta$ une racine nième primitive de l'unité. En d'autres termes, le mot reçu et le mot erreur ont le même syndrome.

$$s_j = y(\beta^j) = \varepsilon(\beta^j)$$

$$s_j = \sum_{i=0}^{n-1} y_i (\beta^j)^i = \sum_{i=0}^{n-1} \varepsilon_i (\beta^j)^i$$

$$= \sum_{i \in I} \varepsilon_i (\beta^i)^j = \sum_{i \in I} Y_i X_i^j$$

*Donc* $s_j = \sum_{i=1}^{v} Y_i X_i^j,\ b \le j \le b + 2e - 1$

Avec $X_i = \beta^i$ et $Y_i = \varepsilon_i$, et les $S_j$, sont calculés à partir du mot reçu y.

Ces équations sont non linéaires, elles ne peuvent pas être résolues directement, elles nécessitent l'utilisation de variables intermédiaires qui peuvent être calculées à l'aide du syndrome et qui permettent de déterminer les positions de l'erreur.

### 4.4.2 Principe de décodage des codes B.C.H

La méthode de décodage algébrique ou de **Peterson, Gorenstein et Zierler** se déroule en trois étapes :

1. Calcul du syndrome du mot reçu.
2. Détermination des positions de l'erreur.
3. Calcul de la valeur de l'erreur aux positions trouvées.

- **Détermination des positions de l'erreur**

Pour déterminer les positions de l'erreur, il suffit de déterminer les localisateurs, pour le faire il suffit de déterminer les racines du polynôme localisateur.

le polynôme localisateur $\sigma(X)$ est de la forme :

$$\sigma(X) = \sigma_v X^v + \sigma_{v-1} X^{v-1} + \cdots + \sigma_1 X + 1. \qquad (1)$$

Il s'agit dans un premier temps de déterminer les coefficients du polynôme localisateur de l'erreur.

$$\sigma(X) = (1 - X_1X) \dots (1 - X_vX)$$

En multipliant les deux membres de l'équation (1) par $Y_iX_i^{j+v}$ et en remplaçant $X$ par $X_i^{-1}$ l'équation devient

$$\sigma_vX_i^{-v} + \sigma_{v-1}X^{-v+1} + \cdots + \sigma_1X_i^{-1} + 1 = 0,$$

$$Y_iX_i^{j+v}(\sigma_vX_i^{-v} + \sigma_{v-1}X^{-v+1} + \cdots + \sigma_1X_i^{-1} + 1) = 0,$$

$$Y_i\left(\sigma_vX_i^{j} + \sigma_{v-1}X_i^{j+1} + \cdots + \sigma_1X_i^{j+v-1} + X_i^{j+v}\right) = 0.$$

En sommant sur $i, 1 \leq i \leq v$, on obtient

$$\sum_{i=1}^{v} Y_i\left(\sigma_vX_i^{j} + \sigma_{v-1}X_i^{j+1} + \cdots + \sigma_1X_i^{j+v-1} + X_i^{j+v}\right) = 0,$$

$$\sigma_v \sum_{i=1}^{v} Y_iX_i^{j} + \sigma_{v-1} \sum_{i=1}^{v} Y_iX_i^{j+1} + \cdots + \sigma_1 \sum_{i=1}^{v} Y_iX_i^{j+v-1} + \sum_{i=1}^{v} Y_iX_i^{j+v} = 0.$$

Et donc : $\sigma_vS_j + \sigma_{v-1}S_{j+1} + \cdots + \sigma_1S_{j+v-1} + S_{j+v} = 0.$

L'équation devient finalement

$$\sigma_vS_j + \sigma_{v-1}S_{j+1} + \cdots + \sigma_1S_{j+v-1} = -S_{j+v}.$$

Par la définition du syndrome tous les $S_j$ sont connus pour $b \leq j \leq b + 2e - 1$, de plus on a $v \leq e$ d'où le système suivant.

$$\begin{pmatrix} S_b & S_{b+1} & \cdots & S_{b+v-1} \\ S_{b+1} & S_{b+2} & \cdots & S_{b+v} \\ \vdots & \vdots & \vdots & \vdots \\ S_{b+v-1} & S_{b+v} & \cdots & S_{b+2v-2} \end{pmatrix} \begin{pmatrix} \sigma_v \\ \sigma_{v-1} \\ \vdots \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} -S_{b+v} \\ -S_{b+v+1} \\ \vdots \\ -S_{b+2v-1} \end{pmatrix} \qquad \text{(I)}$$

Ce système admet une solution unique. Donc les coefficients du polynôme localisateur sont déterminés par la résolution du système (I). Les coefficients du polynôme localisateur de l'erreur$\sigma_1, \sigma_2, \dots, \sigma_v$ sont maintenant connus, il s'agit de déterminer les localisateurs de l'erreur.

Soit $\sigma(X) = \sigma_vX^v + \sigma_{v-1}X^{v-1} + \cdots + \sigma_1X + 1$. Par définition, les racines de $\sigma(X)$ sont les inverses des localisateurs c.à.d. les $X_i^{-1}$ avec $X_i = \beta^i$ et $i \in I$. Parmi tous les éléments de $\mathbb{F}_{p^r}$, on cherche ceux qui sont racine de $\sigma(X)$. On cherche les éléments $\beta^i$ de $\mathbb{F}_{p^r}$, tel que

$\sigma(\beta^i) = 0$. Or $\sigma(\beta^i) = 0$ si et seulementsi $\varepsilon_{n-i} \neq 0$. Donc $\sigma(\beta^i) = 0$ si et seulement si $n - i$ est une position de l'erreur.

- **Calcul de la valeur de l'erreur aux positions trouvées**

Les localisateurs $X_i^j$ sont maintenant connus, il reste à résoudre le système suivant:

$$\sum_{i=1}^{v} Y_i X_i^j = S_j, b \leq j \leq b + 2e - 1$$

On prend *b*=1 sans perte de généralité et $1 \leq j \leq v$ :

$$\begin{pmatrix} X_1 & X_2 & \cdots & X_v \\ X_1^2 & X_2^2 & \cdots & X_v^2 \\ \vdots & \vdots & \vdots & \vdots \\ X_1^v & X_2^v & \cdots & X_v^v \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_v \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_v \end{pmatrix} \qquad \text{(II)}$$

La matrice du système (II) est une matrice de Vandermonde. Si le poids de l'erreur est *v*, alors les localisateurs $X_1, X_2, \ldots, X_v$, sont non nuls et distincts. D'après le théorème sur les matrices de Vandermonde, la matrice du système est inversible et le système admet donc une solution unique.

### 4.4.3 Algorithme de décodage algébrique des codes B. C. H

Soient y($X$) le mot reçu, ε(X) le mot erreur, avec $w(\varepsilon(X)) = v$ et $\beta^b$, $\beta^{b+1}$,...,$\beta^{b+\delta-2}$, avec $b >$0, les $\delta - 1 = 2e$ racines du polynôme générateur.

1. Calcul des composantes du syndrome à partir du mot reçu, $S_j = y(\beta^j)$ pour tout $j$ $b \leq j \leq b + 2e - 1$.

2. Résoudre le système (I) afin de déterminer les coefficients du polynôme localisateur de l'erreur, $\sigma_1, \sigma_2, \ldots, \sigma_v$ et ainsi le polynôme localisateur $\sigma(X)$.
3. Déterminer les localisateurs de l'erreur $X_i = \beta^i, 1 \leq i \leq v$, tel que $\sigma(\beta^{-i}) = 0$, avec $\beta^i \in \mathbb{F}_{p^r}$. alors $\varepsilon_i \neq 0$, l'indice $i$ est une position de l'erreur.
4. Substituer les valeurs de $X_1, X_2, \ldots, X_v$, trouvées, et résoudre le système (II) pour déterminer les valeurs de l'erreur, $Y_i = \varepsilon_i, 1 \leq i \leq v$.
5. Correction de l'erreur et trouver le mot transmis $c(X) = y(X) - \varepsilon(X)$.

**Exemples 4.6.**

Les deux seuls codes BCH avec $n$=7 sont le code de répétition pure $C$(7,1) avec $e$=3 et le code de Hamming $C$(7, 4) avec $e$=1.

1) **Exemple 4.6.1.** Soit $C$(7,1) un code BCH binaire de polynôme générateur : $g(X)$ =$(X^3+X+1)$ $(X^3+X^2+1)$ et soit $y(X)$= $\alpha^2X^6+ \alpha X^5$ le mot reçu contenant v=2 erreurs, tel que les racines successive de $g(X)$ sont $\alpha,\alpha^2,\alpha^3,\alpha^4,\alpha^5,\alpha^6$

Le polynôme localisateur $\sigma(X)= \sigma_2X^2+\sigma_1X+1$

1. **Calcul du syndrome S=($s_1, s_2, s_3, s_4, s_5, s_6$)**

$\mathbb{K}=\mathbb{F}_2{}^r$ le corps de racine 7$^{iéme}$ de l'unité sur $\mathbb{F}_2$, avec $r$=min{$t$ ,$n$=7/2$^r$-1}=3 et donc $\mathbb{K}=\mathbb{F}_2{}^3=\mathbb{F}_8$.

Si $\alpha$ est une racine primitive de $\mathbb{K}$ alors, $\mathbb{K}=\{0, \alpha^i / 0 \leq i \leq 6\}$. Le polynôme primitif de $\mathbb{K}$ est le polynôme $M_\alpha(X)=X^3+X+1$. On a $M_\alpha(\alpha)$=0 donc $\alpha^3=\alpha+1$, $\alpha^4=\alpha^2+\alpha$, $\alpha^5=\alpha^2+\alpha+1$, $\alpha^6= \alpha^2+1$ et $\alpha^7=1$. Le syndrome est donné par : $s_j=y(\alpha^j)$, $1 \leq j \leq 6$ alors :

$S_1=y(\alpha)=\alpha^8+\alpha^6=\alpha+\alpha^2+1=\alpha^5$

$S_2=y(\alpha^2)=\alpha^{14}+\alpha^{11}=1+\alpha^4=\alpha^2+\alpha+1=\alpha^5$

$S_3=y(\alpha^3)=\alpha^{20}+\alpha^{16}= \alpha^6+\alpha^2= \alpha^2+1+ \alpha^2=1$

$S_4=y(\alpha^4)=\alpha^{26}+\alpha^{21}= \alpha^5+1= \alpha^2+\alpha+1+1= \alpha^4$

$S_5=y(\alpha^5)=\alpha^{32}+\alpha^{26}= \alpha^4+\alpha^5= \alpha^2+\alpha+ \alpha^2+\alpha+1=1$

$S_6=y(\alpha^6)=\alpha^{38}+\alpha^{31}= \alpha^3+\alpha^3=0$

2. **Calcul du polynôme localisateur et des localisateurs**

Les coefficients $\sigma_2$, $\sigma_1$ sont les solutions du système :

$$\begin{pmatrix} s_1 & s_2 \\ s_2 & s_3 \end{pmatrix}\begin{pmatrix} \sigma_2 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} s_3 \\ s_4 \end{pmatrix} \qquad \text{(I)}$$

$$\text{(I)} \Leftrightarrow \begin{pmatrix} \alpha^5 & \alpha^5 \\ \alpha^5 & 1 \end{pmatrix}\begin{pmatrix} \sigma_2 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} 1 \\ \alpha^4 \end{pmatrix} \Leftrightarrow \begin{cases} \alpha^5\sigma_2 + \alpha^5\sigma_1 = 1 & (1) \\ \alpha^5\sigma_2 + \sigma_1 = \alpha^4 & (2) \end{cases}$$

En résolvant ce système on trouve $\sigma_1= \alpha$ et $\sigma_2= \alpha^4$.

Le polynôme localisateur est donc $\sigma(X)=\sigma_2X^2+ \sigma_1X+1= \alpha^4X^2+ \alpha X +1$

Cherchons les racines de $\sigma(X)$ dans le corps $\mathbb{F}_8$, on a : $\sigma(0) \neq 0$ , $\sigma(1) \neq 0$

$\sigma(\alpha) = \alpha^6+\alpha^2 +1 = \alpha^2 +1 + \alpha^2 +1 = 0$

$\sigma(\alpha^2) = \alpha^8+\alpha^3 +1= \alpha+\alpha^3 +1= \alpha +1 + \alpha +1 = 0$

Donc les racines sont $\alpha$ et $\alpha^2$ donc les localisateurs sont $X_5= \alpha^{-2}=\alpha^5$ et $X_6= \alpha^{-1}=\alpha^6$.

### 3. Calcul des valeurs des erreurs $y_i=\varepsilon_i$

On résoudre le système :

$$\begin{pmatrix} X_5 X_6 \\ X_5^2 X_6^2 \end{pmatrix}\begin{pmatrix} Y_5 \\ Y_6 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \qquad \text{(II)}$$

$$\text{(II)} \Leftrightarrow \begin{pmatrix} \alpha^5\alpha^6 \\ \alpha^3\alpha^5 \end{pmatrix}\begin{pmatrix} Y_5 \\ Y_6 \end{pmatrix} = \begin{pmatrix} \alpha^5 \\ \alpha^5 \end{pmatrix} \Leftrightarrow \begin{cases} \alpha^5 Y + \alpha^6 Y_6 = \alpha^5 & (1) \\ \alpha^3 Y_5 + \alpha^5 Y_6 = \alpha^5 & (2) \end{cases}$$

Après résolution du système dans le corps $\mathbb{F}_8$, on trouve : $y_5=\alpha$ et $y_6= \alpha^2$.

Donc le mot erreur est : $\varepsilon(X)=\alpha X^5+\alpha^2X^6=y(X)$ et le mot envoyé est $c(X) = y(X)+ \varepsilon(X) = 0$.

**2) Exemple 4.6.2.** Le code de Hamming C(7,4) ave e=1

Soit C(7, 4) un code BCH binaire sur le corps $\mathbb{K}=\mathbb{F}_8$, de polynôme générateur : $g(X) =X^3 + +X+1 =(X- \alpha)(X- \alpha^2 )(X- \alpha^4)$, qui admet 2 racines successives $\alpha$ et $\alpha^2$. Soit $y(X)=1+ X+X^2$ le mot reçu avec une erreur (c.à.d v=1), le polynôme localisateur est de degré 1 : $\sigma(X)= \sigma_1X+1$.

### 1. Calcul du syndrome $S=(s_1, s_2)$

$s_1=y(\alpha)= 1+\alpha+\alpha^2= \alpha^3+\alpha^2=\alpha^5$ et $s_2=y(\alpha^2)= 1+ \alpha^2+\alpha^4= \alpha^3$.

### 2. Calcul du polynôme localisateur et des localisateurs

On résout l'équation : $s_1\sigma_1= s_2 \Leftrightarrow \alpha^5\sigma_1= \alpha^3 \Leftrightarrow \sigma_{1=} \alpha^5$. Le polynôme localisateur $\sigma(X)= \alpha^5X+1$, qui admet une seule racine $\alpha^2$ et donc le localisateur est $X_5 = \alpha^5$. Il y a une erreur en position $i$=5.

### 3. Calcul de la valeur de l'erreur

Comme on considérer que les erreurs sont dans le cops $\mathbb{F}_2$ (binaire), alors la valeur de cette erreur est $Y_5$=1. Le mot erreur est : $\varepsilon(X)=X^5$.

### 4. Le mot envoyé

Le mot envoyé est $c(X)=y(X)+\varepsilon(X)=1+X+X^2+X^5$.

**Exemple 4.7.**

Soit $C(15.\ 7)$ un code BCH sur le corps $\mathbb{F}_{2^4}=\mathbb{F}_{16}$ , de polynôme générateur :

$g(X)=(X^4+X+1)(X^4+X^3+X^2+1)$ qui admet $\alpha,\alpha^2,\alpha^3,\alpha^4$ comme racines successives.

Soit $y(X)=X^2+X^8$ le mot reçu contenant v=2 erreurs.

### 1. Calcul du syndrome

Le syndrome $S=(s_1,s_2,s_3,s_4)$

$\mathbb{K}$ le corps de racine 15$^{\text{iéme}}$ de l'unité $\mathbb{K}=\mathbb{F}_{16}$. $F_{16}=\{0,\ \alpha^i/0\leq i\leq 14\}$.

Le polynôme primitif de $\mathbb{K}$ est le polynôme $M_\alpha(X)=X^4+X+1$. On a : $M_\alpha(\alpha)=0$ donc $\alpha^4=\alpha+1$

Donc : $\alpha^5=\alpha^2+\alpha$; $\alpha^6=\alpha^3+\alpha^2$; $\alpha^7=\alpha^3+\alpha+1$; $\alpha^8=\alpha^2+1$; $\alpha^9=\alpha^3+\alpha^2$ ; $\alpha^{10}=\alpha^2+\alpha+1$;

$\alpha^{11}=\alpha^3+\alpha^2+\alpha$; $\alpha^{12}=\alpha^2+1$; $\alpha^{13}=\alpha^3+\alpha^2+\alpha+1$; $\alpha^{14}=\alpha^3+1$.

On a $s_j=y(\alpha^j)$, $0\leq j\leq 4$.

$s_1=y(\alpha)=\alpha^2+\alpha^8=1$

$s_2=y(\alpha^2)=\alpha^4+\alpha^{16}=\alpha^4+\alpha=1$

$s_3=y(\alpha^3)=\alpha^6+\alpha^{24}=\alpha^6+\alpha^9=\alpha^5$

$s_4=y(\alpha^4)=\alpha^8+\alpha^{32}=\alpha^8+\alpha^2=1$

### 2. Calcul du polynôme localisateur et des localisateurs

Le polynôme localisateur $\sigma(X)=\sigma_2X^2+\sigma_1X+1$. Les coefficients $\sigma_2$, $\sigma_1$sont les solutions du système

$$\begin{pmatrix} s_1 & s_2 \\ s_2 & s_3 \end{pmatrix}\begin{pmatrix}\sigma_2\\ \sigma_1\end{pmatrix}=\begin{pmatrix}s_3\\ s_4\end{pmatrix} \qquad \text{(I)}$$

$$\text{(I)}\Leftrightarrow\begin{pmatrix}1 & 1\\ 1 & \alpha^5\end{pmatrix}\begin{pmatrix}\sigma_2\\ \sigma_1\end{pmatrix}=\begin{pmatrix}\alpha^5\\ 1\end{pmatrix}\Leftrightarrow\begin{cases}\sigma_2+\sigma_1=\alpha^5 & (1)\\ \sigma_2+\alpha^5\sigma_1=1 & (2)\end{cases}$$

Apres résolution du système dans le corps $\mathbb{F}_8$, on trouve : $\sigma_1=1$ et $\sigma_2= \alpha^{10}$, $\sigma(X)= \alpha^{10}X^2+X+1$.

On peut vérifier facilement que les racines de $\sigma(X)$: sont $\alpha^7$ et $\alpha^{10}$.

Donc les localisateurs sont $X_5 = \alpha^5 \; et \; X_8 = \alpha^8$.

**3. Calcul de la valeur de l'erreur**

Si on considérer que les erreurs sont dans le cops $\mathbb{F}_2$ (binaire), alors $\varepsilon(X)=X^5+X^8$ .

**4. Le mot envoyé**

Le mot envoyé est $c(X)=y(X)+ \varepsilon(X)= X^2+X^8 + X^5+X^8 =X^2+X^5$.

## 4.5 Méthode de décodage des codes Reed-Solomon par transformation de Fourier discrète

### 4.5.1 Transformation de Fourier discrète sur un corps fini

Soit F un corps fini de caractéristique p et $\alpha$ une racine primitive du corps $\mathbb{K}$ des racines nièmes de l'unité sur F, soit $a(X) = \sum_{i=0}^{n-1} a_i X^i$ un polynôme de $F[X]/\langle X^n - 1\rangle$ et $\hat{a}(X) = \sum_{j=0}^{n-1} \hat{a}_j X^{n-j}$ est un polynôme de $\mathbb{K}[X]/\langle X^n - 1\rangle$ tels que $\hat{a}_j = a(\alpha^j) = \sum_{i=0}^{n-1} a_i \alpha^{ij}$.

**Définition 4.6.**

Soit F un corps fini, $n$ un entier tel que $n \geq 1$, $\mathbb{K}$ le corps des racines nièmes de l'unité sur F, et $\alpha$ une racine primitive nième de l'unité sur F. La **transformation de Fourier discrète** (**TFD**) sur F est l'application $F_\alpha$ aussi appelée **transformation de Mattson-Solomon**, telle que :

$$F_\alpha \; : F[X]/\langle X^n - 1\rangle \; \rightarrow \; \mathbb{K}[X]/\langle X^n - 1\rangle$$

$$a(X) \mapsto \hat{a}(X)$$

Le polynôme $\hat{a}(X) = \sum_{j=0}^{n-1} \hat{a}_j X^{n-j}$ est appelé le polynôme de Mattson-Solomon .

**Remarque 4.3.**

Le polynôme $\hat{a}(X)$ s'écrit $\hat{a}(X) = \sum_{k=0}^{n-1} \hat{a}_{-k} X^k$, avec $\hat{a}_{-k} = a(\alpha^{-k})$ .

En effet, on fait un changement d'indice k=n-j sur $\hat{a}(X) = \sum_{j=0}^{n-1} \hat{a}_j X^{n-j}$, on obtient :

$$\hat{a}(X) = \sum_{k=1}^{k=n} \hat{a}_{n-k} X^k,$$

et comme les indices sont calculés modulo *n*, alors

$$\hat{a}(X) = \sum_{k=0}^{n-1} \hat{a}_{-k} X^k.$$

**Lemme 4.3.**

Soit $\lambda \in \mathbb{K}$ une racine nième de l'unité, alors :

$$\sum_{i=0}^{n-1} \lambda^i = \begin{cases} 0 \ si \ \ \lambda \neq 1 \\ n \ si \ \ \lambda = 1 \end{cases} .$$

**Preuve.**

Si $\lambda \neq 1$, comme $\lambda^n = 1$, alors

$$\sum_{i=0}^{n-1} \lambda^i = \frac{\lambda^n - 1}{\lambda - 1} = 0 \ .$$

**Théorème 4.4.** (Formule d'inversion)

Si $p \wedge n = 1$, alors la transformation de Mattson-Solomon $\hat{a}(X)$ admet une transformation inverse, i.e. si $\hat{a}(X) = F_\alpha\big(a(x)\big)$, alors

$$a(X) = \frac{1}{n} \sum_{i=0}^{n-1} \hat{a}(\alpha^i) X^i.$$

**Preuve** On a

$$\begin{aligned} \hat{a}(\alpha^i) &= \sum_{j=0}^{n-1} \hat{a}_j \, \alpha^{-ij} \\ &= \sum_{j=0}^{n-1} a(\alpha^j) \, \alpha^{-ij} \\ &= \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} a_k \alpha^{kj} \, \alpha^{-ij} \\ &= \sum_{k=0}^{n-1} a_k \sum_{j=0}^{n-1} \alpha^{j(k-i)} \\ &= n a_i. \end{aligned}$$

Donc, pour tout $i \in \{0,1,...,n-1\}$ .

$$a_i = \frac{1}{n} \hat{a}(\alpha^i).$$

D'où

$$a(X) = \frac{1}{n} \sum_{i=0}^{n-1} \hat{a}(\alpha^i) X^i.$$

**Remarque 4.4.**

Si $F = \mathbb{F}_2$ alors $a(X) = \sum_{i=0}^{n-1} \hat{a}(\alpha^i) X^i$.

**Définition 4.7.**

Soient $A(X), B(X)$ deux polynômes de $\mathbb{K}[X]/< X^n - 1 >$ tels que : $A(X) = \sum_{i=0}^{n-1} A_i X^i$ et $B(X) = \sum_{i=0}^{n-1} B_i X^i$, alors, le **produit d'Hadamard** de A(X) et B(X) est défini par :

$$A(X) \otimes B(X) = \sum_{j=0}^{j=n-1} A_j B_j X^j.$$

**Théorème 4.5.**

La transformation de Mattson-Solomon est un morphisme d'anneaux de $(F[X]/\langle X^n - 1\rangle, +, \times)$ dans $(\mathbb{K}[X]/< X^n - 1 >, +, \otimes)$.

En particulier,

$$F_\alpha\big(p(X) \times q(X)\big) = F_\alpha\big(P(X)\big) \otimes F_\alpha\big(q(X)\big).$$

**Preuve.**

Il suffit montrer la condition de la somme, les deux autres sont claires.

Soient $a(X) = \sum_{i=0}^{n-1} a_i X^i$, $b(X) = \sum_{j=0}^{n-1} b_j X^j$ de $F[X]/< X^n - 1 >$.

$$c(X) = a(X)b(x) = \left(\sum_{i=0}^{n-1} a_i X^i\right)\left(\sum_{j=0}^{n-1} b_j X^j\right) = \sum_{i=0}^{n-1} c_i X^i$$

On a $c_i$ est le coefficient de $X^i$, et comme le produit est dans $F[X]/< X^n - 1 >$, et en considérant que $b_{n+i} = b_i$ alors

$$c_0 = a_0b_0 + a_1b_{n-1} + a_2b_{n-2} + \ldots + a_{n-1}b_1 = \sum_{j=0}^{n-1} a_j b_{n-j+0}.$$

$$c_1 = a_0b_1 + a_1b_0 + a_2b_{n-1} + \ldots + a_{n-1}b_2 = \sum_{j=0}^{n-1} a_j b_{n-j+1}.$$

$$\vdots$$

$$c_i = a_0b_i + a_1b_{i-1} + a_2b_{i-2} + \ldots + a_{n-1}b_{i+1} = \sum_{j=0}^{n-1} a_j b_{n-j+i}.$$

On obtient,

$$a(X)b(X) = \sum_{i=0}^{n-1} c_i X^i = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_j b_{n-j+i} X^i.$$

$$F_\alpha(a(X)b(X)) = \sum_{k=0}^{n-1} \hat{c}_{-k} X^k$$

$$= \sum_{k=0}^{n-1} c(\alpha^{-k}) X^k,$$

D'où

$$F_\alpha(a(X)b(X)) = \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_j b_{n-j+1}\, \alpha^{-ki} X^k \ \dots (*)$$

D'autre part,

$$\begin{aligned}
\left(F_\alpha(a(X))\right) \otimes \left(F_\alpha(b(X))\right) &= \left(\sum_{i=0}^{n-1} \hat{a}_{-i} X^i\right) \otimes \left(\sum_{j=0}^{n-1} \hat{b}_{-j} X^j\right) \\
&= \sum_{k=0}^{n-1} \hat{a}_{-k} \hat{b}_{-k} X^k \\
&= \sum_{k=0}^{n-1} a(\alpha^{-k}) b(\alpha^{-k}) X^k \\
&= \sum_{k=0}^{n-1} \left(\sum_{i=0}^{n-1} a_i \alpha^{-ki}\right)\left(\sum_{j=0}^{n-1} b_j \alpha^{-kj}\right) X^k \\
&= \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_j \alpha^{-kj} b_{n-j+i}\, \alpha^{-k(n-j+i)}\, X^k \\
&= \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_j b_{n-j+1}\, \alpha^{-ki} X^k \\
&\overset{(*)}{=} F_\alpha(a(X)b(X)).
\end{aligned}$$

**Corollaire 4.1.**

Si $p \wedge n = 1$, ou $p$ est la caracteristique du corps fini F et $\mathbb{K}$ le corps des racines *nièmes* de l'unité sur *F*, alors la transformation de Mattson-Solomon est un isomorphisme d'anneaux de $(F[X]/\langle X^n - 1\rangle, +, \times)$ dans $(\mathbb{K}[X]/< X^n - 1 >, +, \otimes)$.

**Preuve.**

On a la transformation de Mattson-Solomon est injective, de plus elle est surjective (théorème d'inversion), donc d'après le théorème précédent la transformation de Mattson-Solomon est un isomorphisme d'anneaux.

### 4.5.2 Algorithme de décodage par transformation de Fourier discrète

Soit $C(\mathrm{n}, \mathrm{k}, \mathrm{d})$ un code Reed-Solomon sur un corps fini $\mathbb{K}=\mathbb{F}_{2^m}$ *e*-correcteur, et de polynôme générateur $g(X) = (X - \alpha^b)(X - \alpha^{b+1}) \dots (X - \alpha^{b+d-2})$ où $\alpha$ une racine primitive de l'unité du corps $\mathbb{K}=\mathbb{F}_{2^m}$ (corps des racines *n*iémes de l'unité sur $\mathbb{F}_2$).

Donc g(x) admet $\alpha^b, \alpha^{b+1}, \dots, \alpha^{b+d-2}$ $(b > 0)$ comme racines.

On prend F= $\mathbb{K} = \mathbb{F}_{2^m}$ .

- **Calcul du syndrome de l'erreur**

Soit y(X) le mot reçu, ε(X) le mot erreur avec $(\varepsilon(X)) \leq e$, le syndrome de l'erreur est donc :

$$y(\alpha^{b+j}) = \varepsilon(\alpha^{b+j}), 0 \leq j \leq d - 2, (b > 0) \ \ldots \ (**),$$

le mot erreur s'écrit :

$$\varepsilon(X) = \varepsilon_0 + \varepsilon_1 X + \varepsilon_2 X^2 + \cdots + \varepsilon_{n-1} X^{n-1}.$$

La transformée de Fourrier de ε(X) est égale à :

$\hat{\varepsilon}(X) = \sum_{i=0}^{n-1} \hat{\varepsilon}_{n-i} X^i = \hat{\varepsilon}_0 + \hat{\varepsilon}_{n-1} X + \hat{\varepsilon}_{n-1} X^2 + \cdots + \hat{\varepsilon}_2 X^{n-2} + \hat{\varepsilon}_1 X^{n-1}.$

D'après la relation $(**)$, les d $-$ 1 coefficients de $\hat{\varepsilon}(X)$ sont trouvés.

il reste donc de déterminer les autres coefficients de ε̂(X).

- **Détermination du polynôme localisateur de l'erreur**

Soit $v$ le poids de l'erreur c.à.d. $v = w(\varepsilon(X))$ avec $v \leq e$ . Les localisateurs des positions des erreurs, sont les $X_i = \alpha^i$, pour $1 \leq i \leq v$.

On note $I$ l'ensemble des positions de l'erreur : $I = \{i, 0 \leq i \leq n-1 / \varepsilon_i \neq 0\}$

Le polynôme localisateur de l'erreur est $\sigma(X) = \prod_{i \in l} (1 - X_i X)$.

Qu'on peut écrire : $\sigma(X) = \sigma_v X^v + \sigma_{v-1} X^{v-1} : + \ldots + \sigma_1 X + 1$

Les racines de $\sigma(X)$ sont les inverses des localisateurs des positions erronées, on a donc :

$$\text{si } i \in I : \ \sigma(X_i^{-1}) = \sigma(\alpha^{-i}) = 0 \text{ et } \varepsilon_i \neq 0$$

$$\text{si } i \notin I : \sigma(X_i^{-1}) = \sigma(\alpha^{-i}) \neq 0 \text{ et } \varepsilon_i = 0$$

Les $\sigma(\alpha^{-i})$ pour $i \in \{1, \ldots, n\}$ sont les coefficients de la transformée de Fourrier de $\sigma(X)$, ce qui permet d'écrire $\varepsilon(X) \otimes \hat{\sigma}(X) = 0$ , où $\hat{\sigma}(X)$ est la transformée de Fourrier de $\sigma(X)$. Alors,

$$\hat{\varepsilon}(X) \times \sigma(X) \equiv 0 \ (mod \ (X^n - 1))$$

En effet :

Calculons d'abord $\hat{\varepsilon}(X) \times \sigma(X)$

On a : $\hat{\varepsilon}(X) = \sum_{i=0}^{n-1} \hat{\varepsilon}_{n-i} X^i$

Et $\sigma(X) = \sum_{i=0}^{v} \sigma_i X^i = \sum_{i=0}^{n-1} \sigma_i X^i$, tels que : $\sigma_i = 0$ pour $v+1 \leq i \leq n$ .

Donc, $\hat{\varepsilon}(X) \times \sigma(X) = (\hat{\varepsilon}_0 + \hat{\varepsilon}_{n-1}X + \ldots + \hat{\varepsilon}_1 X^{n-1})(\sigma_0 + \sigma_1 X + \ldots + \sigma_{n-1}X^{n-1})$

$= \sum_{k=0}^{n-1} c_k X^k$

tels que : $c_0 = \hat{\varepsilon}_0\sigma_0 + \hat{\varepsilon}_{n-1}\sigma_{n-1} + \ldots + \hat{\varepsilon}_1\sigma_1 = \sum_{i=0}^{n-1} \hat{\varepsilon}_i\sigma_i$

$c_1 = \hat{\varepsilon}_0\sigma_1 + \hat{\varepsilon}_{n-1}\sigma_{0=n} + \hat{\varepsilon}_{n-2}\sigma_{n-1} + \ldots + \hat{\varepsilon}_1\sigma_2 = \sum_{i=0}^{n-1} \hat{\varepsilon}_i\sigma_{i+1}$

$c_2 = \hat{\varepsilon}_0\sigma_2 + \hat{\varepsilon}_{n-1}\sigma_{1=n+1} + \hat{\varepsilon}_{n-2}\sigma_{0=n} + \hat{\varepsilon}_{n-3}\sigma_{n-1} + \ldots + \hat{\varepsilon}_1\sigma_3 = \sum_{i=0}^{n-1} \hat{\varepsilon}_i\sigma_{i+2}$

$\vdots$

Et ainsi de suite, à la fin on trouve :

$$\hat{\varepsilon}(X) \times \sigma(X) = \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \hat{\varepsilon}_i \sigma_{i+k} X^k = \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} \hat{\varepsilon}_{j-k} \sigma_j X^k$$

$$= \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} \varepsilon(\alpha^{j-k}) \sigma_j X^k$$

$$= \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \varepsilon_i \alpha^{-ik} X^k \sum_{j=0}^{n-1} \sigma_j \alpha^{ij}$$

$$= \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \varepsilon_i \sigma(\alpha^i) \alpha^{-ik} X^k$$

$$= \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \varepsilon_i \hat{\sigma}_i \alpha^{-ik} X^k = \sum_{k=0}^{n-1} \varepsilon \otimes \hat{\sigma}(\alpha^{-k}) X^k$$

$$\equiv 0 \ (mod(X^n - 1))$$

Ceci donne un système d'équations à résoudre.

Supposons qu'il y a au plus *e* erreurs, le polynôme $\sigma$ a donc au plus v racines, il est donc au plus de degré v, et donc $\sigma_i = 0$, pour $v + 1 \leq i \leq n$ . Il reste à déterminer $\sigma_1, \sigma_2, \ldots, \sigma_v$ et $\hat{\varepsilon}_d, \ldots, \hat{\varepsilon}_n$.

Le système s'écrit, où les calculs sur les indices sont effectués modulo *n* comme suit:

$$\begin{cases} \hat{\varepsilon}_0 + \hat{\varepsilon}_1\sigma_1 + \hat{\varepsilon}_2\sigma_2 + \ldots + \hat{\varepsilon}_v\sigma_v = 0 \\ \hat{\varepsilon}_{n-1} + \hat{\varepsilon}_0\sigma_1 + \hat{\varepsilon}_1\sigma_2 + \ldots + \hat{\varepsilon}_{v-1}\sigma_v = 0 \\ \hat{\varepsilon}_{n-2} + \hat{\varepsilon}_{n-1}\sigma_1 + \hat{\varepsilon}_0\sigma_2 + \ldots + \hat{\varepsilon}_{v-2}\sigma_v = 0 \\ \vdots \\ \hat{\varepsilon}_{n-1} + \hat{\varepsilon}_{n-i+1}\sigma_1 + \hat{\varepsilon}_{n-i+2}\sigma_2 + \ldots + \hat{\varepsilon}_{v-i}\sigma_v = 0 \\ \vdots \\ \hat{\varepsilon}_1 + \hat{\varepsilon}_2\sigma_1 + \hat{\varepsilon}_3\sigma_2 + \ldots + \hat{\varepsilon}_{v+1}\sigma_v = 0 \end{cases}$$

Les $d-1$ coefficients de la transformée de Fourier de l'erreur. $\hat{\varepsilon}_0, \hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_{d-1}$sont connus. En considérant les dernières équations du système il est possible de déterminer les coefficients du polynôme localisateur de l'erreur $\sigma_1, \sigma_2, \ldots, \sigma_v$.

- **Détermination des autres coefficients de la transformée de Fourier de l'erreur**

Les $n-v$ premières équations du système permettent de déterminer $\hat{\varepsilon}_d, \hat{\varepsilon}_{d+1}, \ldots, \hat{\varepsilon}_n$. En remplaçant $\sigma_1, \sigma_2, \ldots, \sigma_v$ par les valeurs trouvées précédemment. La transformée de Fourier de l'erreur $\hat{\varepsilon}(X)$ est désormais connue.

- **Détermination de $\varepsilon(X)$ par transformation de Fourier inverse**

On applique la transformation de Fourier inverse pour trouver le mot erreur $\varepsilon(X)$ tel que :

$$\varepsilon(X) = \frac{1}{n}\sum_{i=0}^{n-1}\hat{\varepsilon}(\alpha^i)X^i.$$

L'Algorithme de Décodage par transformation de Fourier discrète se résume en cinq étapes :

Soient $y(X)$ le mot reçu, $\varepsilon(X)$ le mot erreur avec $w(\varepsilon(X)) \leq e$ et $g(X)$ le polynôme générateur. Soit les $\delta - 1$ racines de $g(X)$ d'exposants consécutifs. $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{\delta+b-2}$ avec $b > 0$ :

1) Calcul des $\delta - 1$ coefficients de la transformée de Fourier de l'erreur $\varepsilon(X)$ par le calcul des composantes du Syndromes.

2) Détermination des coefficients $\sigma_1, \sigma_2, \ldots, \sigma_v$ du polynôme localisateur de l'erreur, par la résolution des v dernières équations du système.

$$\hat{\varepsilon}(X) \times \sigma(X) \equiv 0 \left(mod\ (X^n - 1)\right)$$

3) Détermination de tous les coefficients de la transformée de Fourier de l'erreur $\hat{\varepsilon}(X)$ par la résolution des $n-v$ premières équations du même système.

4) Détermination de $\varepsilon(X)$ par la transformation de Fourier inverse de $\hat{\varepsilon}(X)$.

5) Correction de l'erreur i.e. trouver le mot envoyé c(X).

$$c(X) = y(X) - \varepsilon(X).$$

**Exemple 4.8.**

Soit le code de Reed-Solomon C(3,1,3) sur $\mathbb{F}_4$ de polynôme générateur $g(X) = X^2 + X + 1$, ce polynôme a 2 racines : $\alpha$ et $\alpha^2$. Soit $y(X) = X^2 + 1$, le mot reçu avec *v*=1 erreur.

1. Calcul du syndrome :

$$s_1 = y(\alpha) = \alpha$$
$$s_2 = y(\alpha^2) = \alpha^2$$

Le syndrome donne les 2 premiers coefficients de la transformée de Fourier de $\varepsilon(X)$ :

$\hat{\varepsilon}_1 = \alpha$et $\hat{\varepsilon}_2 = \alpha^2$. On a donc:

$$\hat{\varepsilon}(X) = \sum_{i=0}^{2} \hat{\varepsilon}_{3-i} X^i = \hat{\varepsilon}_0 + \hat{\varepsilon}_2 X + \hat{\varepsilon}_1 X^2.$$

Le polynôme localisateur de l'erreur est de degré v=1. Il est de la forme :

$$\sigma(X) = \sigma_1 X + 1.$$

L'équation suivante :

$$\hat{\varepsilon}(X) \times \sigma(X) = \sum_{k=0}^{2} \sum_{j=0}^{1} \hat{\varepsilon}_{j-k} \sigma_j X^i \equiv 0(\mathrm{mod}(X^3 - 1))$$

Donne le système suivant :

$$\begin{cases} \hat{\varepsilon}_0 + \hat{\varepsilon}_1 \sigma_1 = 0 \\ \hat{\varepsilon}_2 + \hat{\varepsilon}_0 \sigma_1 = 0 \\ \hat{\varepsilon}_1 + \hat{\varepsilon}_2 \sigma_1 = 0 \end{cases}$$

2. Détermination du polynôme localisateur de l'erreur :
   Les coefficients $\hat{\varepsilon}_1$ et $\hat{\varepsilon}_2$ sont connus ($\hat{\varepsilon}_1 = \alpha$ et $\hat{\varepsilon}_2 = \alpha^2$) .
   La résolution de système précédent donne $\sigma_1 = \alpha^2$.
3. Détermination des autres coefficients de la transformée de Fourier de l'erreur :
   $\sigma_1$ est remplacés par leur valeur dans les autres équations du système nécessaire à la détermination de $\hat{\varepsilon}_0$. La résolution de ce nouveau système donne : $\hat{\varepsilon}_0 = \alpha^3 = 1$.

   Donc la transformée de Fourier de $\varepsilon(X)$ est : $\hat{\varepsilon}(X) = 1 + \alpha^2 X + \alpha X^2$.

   En utilisant la transformation de Fourier inverse : $\varepsilon(X)$.

On trouve : $\hat{\varepsilon}(1) = 0, \hat{\varepsilon}(\alpha) = 1$ et $\hat{\varepsilon}(\alpha^2) = 0$. Donc $\varepsilon(X) = X$.

Et le mot envoyé est $c(X) = y(X) - \varepsilon(X)$.

Donc $c(X) = X^2 + X + 1$.

**Exemple 4.9.**

Soit le code de Reed-Solomon C(15,9,7) sur le corps $\mathbb{F}_{16}$, de polynôme générateur

$$g(X) = X^6 + \alpha^{10}X^5 + \alpha^{14}X^4 + \alpha^4X^3 + \alpha^6X^2 + \alpha^9X + \alpha^6.$$

Ce polynôme a 6 racines : $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5$ et $\alpha^6$. Soit le mot reçu $y(X) = \alpha X^{14} + \alpha^2X^{12} + \alpha^{13}X^4$, avec $w(\varepsilon(X)) = 3$.

1) Construction du corps $\mathbb{F}_{16}$

$\mathbb{F}_{16}$=*{0,*$\alpha^i / 0 \leq i \leq 14$*}*, son polynôme primitif est : $M_\alpha(X)=X^4+X+1$.

On a : $M_\alpha(\alpha)=0$ *donc* $\alpha^4=\alpha+1$, *donc :* $\alpha^6=\alpha^3+\alpha^2$; $\alpha^5=\alpha^2+\alpha$; $\alpha^7=\alpha^3+\alpha+1$; $\alpha^8=\alpha^2+1$;

$\alpha^9=\alpha^3+\alpha^2$; $\alpha^{10}=\alpha^2+\alpha+1$; $\alpha^{11}=\alpha^3+\alpha^2+\alpha$; $\alpha^{12}=\alpha^2+1$; $\alpha^{13}=\alpha^3+\alpha^2+\alpha+1$; $\alpha^{14}=\alpha^3+1$.

2) Calcul du syndrome :

On a $s_j = y(\alpha^j)$ , $1 \leq j \leq 6$.

$$s_1 = y(\alpha) = \alpha^6$$
$$s_2 = y(\alpha^2) = \alpha^7$$
$$s_3 = y(\alpha^3) = \alpha^{12}$$
$$s_4 = y(\alpha^4) = 0$$
$$s_5 = y(\alpha^5) = \alpha$$
$$s_6 = y(\alpha^6) = \alpha^8$$

Le syndrome donne les 6 premiers coefficients de la transformée de Fourier de $\varepsilon(X)$ ;

$$\hat{\varepsilon}_1 = \alpha^6, \hat{\varepsilon}_2 = \alpha^7, \hat{\varepsilon}_3 = \alpha^{12}, \hat{\varepsilon}_4 = 0, \hat{\varepsilon}_5 = \alpha, \hat{\varepsilon}_6 = \alpha^8,$$

la transformée de Fourier de $\varepsilon(X)$ :

$$\hat{\varepsilon}(X) = \sum_{i=0}^{14} \hat{\varepsilon}_{15-i}X^i = \hat{\varepsilon}_0 + \hat{\varepsilon}_{14}X + \hat{\varepsilon}_{13}X^2 + \ldots + \hat{\varepsilon}_2X^{13} + \hat{\varepsilon}_1X^{14}$$

Le polynôme localisateur de l'erreur est de degré v=3. Il est de la forme :

$\sigma(X) = \sigma_3 X^3 + \sigma_2 X^2 + \sigma_1 X + 1.$

L'équation suivante : $\hat{\varepsilon}(X) \times \sigma(X) = \sum_{k=0}^{14} \sum_{j=0}^{3} \hat{\varepsilon}_{j-k}\, \sigma_j X^k \equiv 0(\mathrm{mod}(X^{15} - 1))$

Donne le système suivant :

$$\begin{cases} \hat{\varepsilon}_0 + \hat{\varepsilon}_1\sigma_1 + \hat{\varepsilon}_2\sigma_2 + \hat{\varepsilon}_3\sigma_3 = 0 \\ \hat{\varepsilon}_{14} + \hat{\varepsilon}_0\sigma_1 + \hat{\varepsilon}_1\sigma_2 + \hat{\varepsilon}_2\sigma_3 = 0 \\ \hat{\varepsilon}_{13} + \hat{\varepsilon}_{14}\sigma_1 + \hat{\varepsilon}_0\sigma_2 + \hat{\varepsilon}_1\sigma_3 = 0 \\ \hat{\varepsilon}_{12} + \hat{\varepsilon}_{13}\sigma_1 + \hat{\varepsilon}_{14}\sigma_2 + \hat{\varepsilon}_0\sigma_3 = 0 \\ \hat{\varepsilon}_{11} + \hat{\varepsilon}_{12}\sigma_1 + \hat{\varepsilon}_{13}\sigma_2 + \hat{\varepsilon}_{14}\sigma_3 = 0 \\ \hat{\varepsilon}_{10} + \hat{\varepsilon}_{11}\sigma_1 + \hat{\varepsilon}_{12}\sigma_2 + \hat{\varepsilon}_{13}\sigma_3 = 0 \\ \hat{\varepsilon}_9 + \hat{\varepsilon}_{10}\sigma_1 + \hat{\varepsilon}_{11}\sigma_2 + \hat{\varepsilon}_{12}\sigma_3 = 0 \\ \hat{\varepsilon}_8 + \hat{\varepsilon}_9\sigma_1 + \hat{\varepsilon}_{10}\sigma_2 + \hat{\varepsilon}_{11}\sigma_3 = 0 \\ \hat{\varepsilon}_7 + \hat{\varepsilon}_8\sigma_1 + \hat{\varepsilon}_9\sigma_2 + \hat{\varepsilon}_{10}\sigma_3 = 0 \\ \hat{\varepsilon}_6 + \hat{\varepsilon}_7\sigma_1 + \hat{\varepsilon}_8\sigma_2 + \hat{\varepsilon}_9\sigma_3 = 0 \\ \hat{\varepsilon}_5 + \hat{\varepsilon}_6\sigma_1 + \hat{\varepsilon}_7\sigma_2 + \hat{\varepsilon}_8\sigma_3 = 0 \\ \hat{\varepsilon}_4 + \hat{\varepsilon}_5\sigma_1 + \hat{\varepsilon}_6\sigma_2 + \hat{\varepsilon}_7\sigma_3 = 0 \\ \hat{\varepsilon}_3 + \hat{\varepsilon}_4\sigma_1 + \hat{\varepsilon}_5\sigma_2 + \hat{\varepsilon}_6\sigma_3 = 0 \\ \hat{\varepsilon}_2 + \hat{\varepsilon}_3\sigma_1 + \hat{\varepsilon}_4\sigma_2 + \hat{\varepsilon}_5\sigma_3 = 0 \\ \hat{\varepsilon}_1 + \hat{\varepsilon}_2\sigma_1 + \hat{\varepsilon}_3\sigma_2 + \hat{\varepsilon}_4\sigma_3 = 0 \end{cases}$$

3) Détermination du polynôme localisateur de l'erreur :

Les coefficients $\hat{\varepsilon}_1, \hat{\varepsilon}_2, \hat{\varepsilon}_3, \hat{\varepsilon}_4, \hat{\varepsilon}_5, \hat{\varepsilon}_6$ sont connus tels que $\hat{\varepsilon}_1 = \alpha^6, \hat{\varepsilon}_2 = \alpha^7, \hat{\varepsilon}_3 = \alpha^{12}, \hat{\varepsilon}_4 = 0, \hat{\varepsilon}_5 = \alpha$ et $\hat{\varepsilon}_6 = \alpha^8$

La résolution du système constitué par les trois dernière equations donne :

$$\begin{cases} \sigma_1 = \alpha^2 \\ \sigma_2 = \alpha^8 \\ \sigma_3 = 1 \end{cases}$$

Le polynôme localisateur est : $\sigma(X) = X^3 + \alpha^8 X^2 + \alpha^2 X + 1.$

4) Détermination des autres coefficients de la transformée de Fourier de l'erreur :

$\sigma_1, \sigma_2, \sigma_3$ sont remplacés par leurs valeurs dans les autres équations du système.

Après les calculs en trouve :

$\hat{\varepsilon}_0 = \alpha^7, \hat{\varepsilon}_{14} = \alpha^3, \hat{\varepsilon}_{13} = \alpha^7, \hat{\varepsilon}_{12} = \alpha^{12}, \hat{\varepsilon}_{11} = 0, \hat{\varepsilon}_{10} = \alpha^{13}, \hat{\varepsilon}_9 = \alpha^{11}, \hat{\varepsilon}_8 = 1, \hat{\varepsilon}_7 = \alpha^9$

Donc la transformée de Fourier de $\varepsilon(X)$ est :

$$\hat{\varepsilon}(X) = \alpha^7 + \alpha^3 X + \alpha^7 X^2 + \alpha^{12} X^3 + \alpha^{13} X^5 + \alpha^{11} X^6 + X^7 + \alpha^9 X^8 + \alpha^8 X^9 + \alpha X^{10} + \alpha^{12} X^{12} + \alpha^7 X^{13} + \alpha^6 X^{14}.$$

En utilisant la transformation de Fourier inverse, on trouve:

$$\varepsilon(X) = \frac{1}{15} \sum_{i=0}^{14} \hat{\varepsilon}(\alpha^i)\, X^i = \sum_{i=0}^{14} \hat{\varepsilon}(\alpha^i)\, X^i.$$

On trouve :

$$\hat{\varepsilon}(1) = \hat{\varepsilon}(\alpha) = \hat{\varepsilon}(\alpha^2) = \hat{\varepsilon}(\alpha^3) = \hat{\varepsilon}(\alpha^5) = \hat{\varepsilon}(\alpha^6) = \hat{\varepsilon}(\alpha^7) = \hat{\varepsilon}(\alpha^8) = \hat{\varepsilon}(\alpha^9) = \hat{\varepsilon}(\alpha^{10}) = \hat{\varepsilon}(\alpha^{11}) = \hat{\varepsilon}(\alpha^{13}) = 0.$$

Et $\hat{\varepsilon}(\alpha^4) = \alpha^{13}, \hat{\varepsilon}(\alpha^{12}) = \alpha^2, \hat{\varepsilon}(\alpha^{14}) = \alpha$.

Donc $\varepsilon(X) = \alpha X^{14} + \alpha^2 X^{12} + \alpha^{13} X^4 = y(X)$.

Et le mot envoyé est : $c(X) = y(X) - \varepsilon(X)$=0 (le mot nul).

# Chapitre 5 Application des codes cycliques à la cryptographie.

## Introduction

L'application des codes correcteurs en cryptographie représente une convergence fascinante entre la correction d'erreurs et la sécurité de l'information. La cryptographie, qui concerne la science de la sécurisation des données et des communications, joue un rôle crucial dans notre monde numérique interconnecté. Les codes correcteurs d'erreurs, quant à eux, sont des outils mathématiques qui permettent de détecter et de corriger les erreurs dans les données transmises ou stockées. Lorsque ces deux domaines se rejoignent, cela crée une puissante alliance pour protéger les informations confidentielles, garantir l'intégrité des données et sécuriser les communications.

Les codes correcteurs d'erreurs sont essentiellement utilisés pour garantir que les données transmises ou stockées restent intactes, même dans des environnements sujets à des perturbations, des interférences ou des attaques. Ils fonctionnent en ajoutant une certaine quantité de redondance aux données d'origine, ce qui permet de détecter et, dans de nombreux cas, de corriger les erreurs. Cependant, leur application en cryptographie va au-delà de la simple correction d'erreurs accidentelles. Les codes correcteurs d'erreurs sont intégrés dans des protocoles cryptographiques pour diverses raisons :

1. **Confidentialité**: Lors de la transmission de données sensibles, il est essentiel de s'assurer qu'elles ne peuvent pas être interceptées ou comprises par des tiers non autorisés. Les codes correcteurs d'erreurs peuvent être utilisés pour ajouter un niveau de confidentialité supplémentaire en cryptant les données avant de les transmettre.
2. **Intégrité des données**: La modification ou la corruption de données est une menace courante en ligne. Les codes correcteurs d'erreurs permettent de détecter ces altérations et, dans certains cas, de les corriger automatiquement pour garantir l'intégrité des données.
3. **Authentification**: Les codes correcteurs d'erreurs peuvent être utilisés pour générer des signatures numériques, qui sont des empreintes digitales cryptographiques associées à des messages. Ces signatures permettent de vérifier l'authenticité des messages et de s'assurer qu'ils proviennent bien de la source prétendue.

4. **Résilience aux attaques**: Les codes correcteurs d'erreurs renforcent la résistance aux attaques par force brute et à d'autres attaques cryptographiques. Ils rendent plus difficile pour un attaquant de manipuler les données sans être détecté.
5. **Sécurisation des canaux de communication**: Lors de la transmission de données via des canaux non fiables, comme Internet, les codes correcteurs d'erreurs garantissent que les données atteignent leur destination de manière fiable, même en présence de bruit ou d'interférences.

En résumé, l'application des codes correcteurs d'erreurs en cryptographie est une discipline essentielle pour sécuriser les données et les communications dans le monde numérique. Elle repose sur les principes de la correction d'erreurs pour garantir la confidentialité, l'intégrité, l'authenticité et la fiabilité des informations échangées, renforçant ainsi la sécurité des transactions financières, des communications gouvernementales, de la confidentialité des données personnelles et de bien d'autres aspects de notre vie quotidienne numérique.

## 5.1 Notions de Cryptographie.

**Définition 5.1.**

Un **système de cryptographie** ou **cryptosystème** est constitué de:

1. Un **alphabet** $A$. En pratique $A = \{0, 1\}$. Les éléments de $A$ sont dits **symboles**.
2. Un ensemble M composé de chaînes de symboles de l'alphabet A est appelé **espace de messages clairs**. Un élément de M est appelé **texte clair**.
3. Un ensemble C constitué de chaînes de symboles d'un alphabet B, qui peut être différente de l'alphabet $A$ est appelé **espace de messages chiffrés** ou **cryptogrammse**.
4. Un ensemble K dit **espace des clés**. Un élément $e$ de K est dit **clé**.
5. Pour chaque clé $e$ de K, on définit une bijection $f_e$ de $M$ dans $C$, dite **fonction de chiffrement**. Si $m \in M$ alors $f_e(m)=c \in C$.
6. Pour chaque clé $d$ de K, on définit une bijection $f_d$ de $C$ dans $M$, dite **fonction de déchiffrement**. Si $c \in C$ alors $f_d(c)=m \in M$.

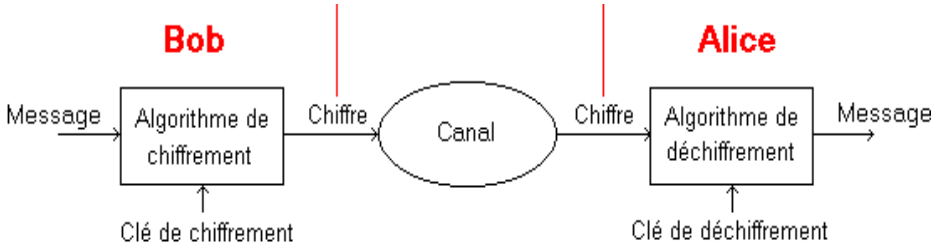



Fig 5.1 Schéma général d'un cryptosystème.

Il existe deux types de cryptographie:

### 5.1.1 Cryptographie symétrique.

**Définition 5.2.**

En **cryptographie symétrique**, également appelée **cryptographie à clé secrète**, une seule clé suffit pour le chiffrement et le déchiffrement. les clés *e* et *d* sont identiques.

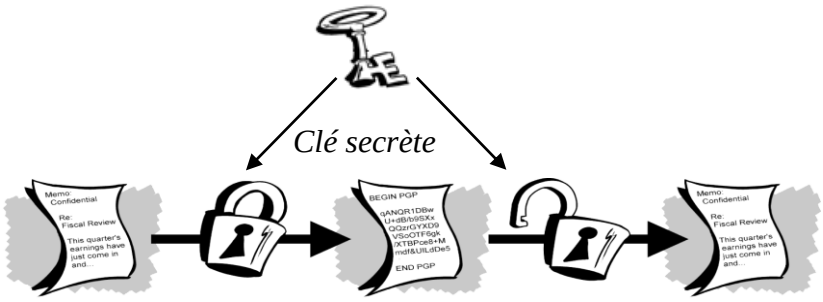



Fig 5.2 Chiffrement et déchiffrement symétrique.

### 5.1.2 Cryptographie asymétrique.

**Définition 5.3.**

En **cryptographie asymétrique**, également appelée **cryptographie à clé publique**, admet deux clés, une clé *e* (dite **publique**) sert pour le chiffrement et une autre clé *d* différente de la clé *e* (dite **secrète**) sert pour le déchiffrement.

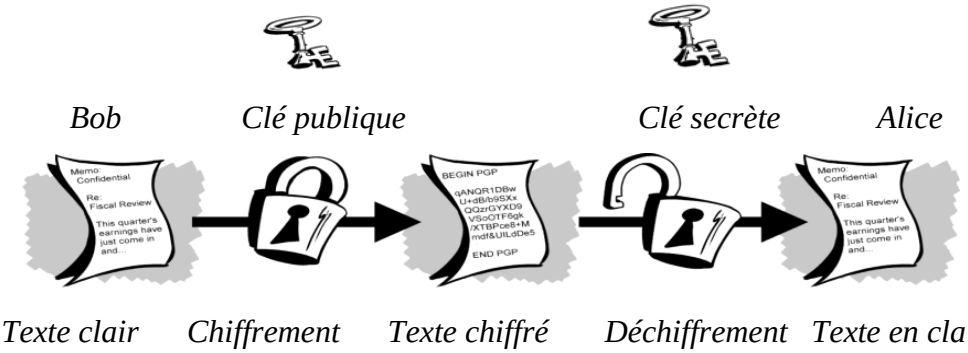



Fig 5.3 Chiffrement et déchiffrement asymétrique.

## 5.2 Cryptosystème de McEliece.

### 5.2.1 Historique et principe du Cryptosystème de McEliece.

Le cryptosystème de McEliece, développé par l'informaticien américain Robert J. McEliece en 1978, est l'un des premiers systèmes de chiffrement à clé publique basés sur la théorie de la correction d'erreur. Il diffère considérablement des systèmes de chiffrement à clé publique plus couramment utilisés, tels que le RSA ou le chiffrement basé sur les courbes elliptiques. Le cryptosystème de McEliece est apprécié pour sa résistance théorique aux attaques de déchiffrement quantique, ce qui en fait une option attrayante pour la sécurité à long terme.

L'idée fondamentale derrière le cryptosystème de McEliece repose sur les codes correcteurs d'erreurs, qui sont traditionnellement utilisés pour corriger les erreurs de transmission dans les communications numériques. Au lieu d'utiliser ces codes pour corriger des erreurs, McEliece a conçu son système de chiffrement en exploitant la difficulté de déterminer les erreurs dans les données chiffrées.

Voici une brève introduction au fonctionnement du cryptosystème de McEliece :

1. **Génération des clés** : Tout d'abord, l'utilisateur génère une paire de clés, une clé publique et une clé privée. La clé publique est destinée à chiffrer les messages, tandis que la clé privée est utilisée pour déchiffrer les messages.
2. **Construction de la matrice de génération de code** : Un élément clé du cryptosystème de McEliece est une matrice binaire de grande taille appelée matrice de génération de code. Cette matrice est générée de manière aléatoire et est utilisée pour encoder les messages en texte chiffré.
3. **Chiffrement** : Pour chiffrer un message, l'émetteur encode le message en utilisant la matrice de génération de code et ajoute ensuite un vecteur d'erreur aléatoire. Le résultat est le texte chiffré, qui est envoyé au destinataire.
4. **Déchiffrement** : Pour déchiffrer le message, le destinataire utilise la clé privée, qui consiste en une description de la matrice de génération de code ainsi que des informations pour déterminer et corriger les erreurs introduites lors du chiffrement. En appliquant des techniques de correction d'erreur, le destinataire peut retrouver le message d'origine.

Ce qui distingue le cryptosystème de McEliece, c'est sa résistance présumée aux attaques quantiques, notamment aux attaques de factorisation quantique qui menacent la sécurité du systèmes cryptographiques traditionnels comme le RSA. Cependant, le principal inconvénient de ce cryptosystème réside dans la taille relativement grande de la clé publique, ce qui peut rendre les opérations de chiffrement et de déchiffrement plus lentes par rapport à d'autres méthodes. Malgré cela, le cryptosystème de McEliece reste un sujet de recherche actif en cryptographie, en particulier pour ses avantages en matière de sécurité quantique.

### 5.2.2 Algorithme Cryptosystème de McEliece.

**1. Génération de clé**

On commence par générer un code cyclique $C(n, k, d)$ en donnant son polynôme générateur $g(X)$ et donc sa matrice génératrice $G$ de taille $k \times n$. On va mélanger cette matrice pour la rendre indistinguable d'une matrice aléatoire, pour cela on a besoin de :

a. Une matrice de permutation aléatoire $P_\sigma$ de taille $n \times n$ associée à une permutation $\sigma$ de $S_n$.

b. Une matrice inversible aléatoire $S$ de taille $k \times k$.

La clé publique sera le couple $(G', e)$ tel que $G' = S.G.P_\sigma$ qui est indistinguable d'une matrice aléatoire et $e$ la capacité de correction de C.

La clé secrète est composée des trois matrices $S$, $P_\sigma$ et $G$ qui permettent de retrouver la structure du code C et donnent donc accès à l'algorithme de décodage.

**2. Chiffrement.**

Soit $m$ un message de $k$ bits que l'on veut chiffrer. On ne dispose pour cela que de la clé publique $G'$. On commence par calculer le mot de code c, de longueur $n$, associé à $m$ : $c = m.G'$. Ensuite on génère une *erreur aléatoire* $\varepsilon$ de longueur $n$ et de poids $t \leq e$. Le texte chiffré sera simplement le mot bruité : $c' = c + \varepsilon$.

3. **Déchiffrement.** Pour déchiffrer le texte c', en connaissant $P_\sigma$, $S$ et $G$, il suffit de calculer : $r' = c'.P_\sigma^{-1} = m.G'.P_\sigma^{-1} + \varepsilon.P_\sigma^{-1} = m\,S.G + \varepsilon.P_\sigma^{-1}$. Le mot $r = m.S.G$ est un mot du code C et $\varepsilon' = \varepsilon.P_\sigma^{-1}$ est une erreur de poids $t$ (car P est une permutation et conserve donc le poids des mots), donc on peut corriger cette erreur et retrouver le message initial $m' = m\,S$ et le message clair $m = m'.S^{-1}$.

**Exemple 5.1.**

Soit C(*7*, 4, *d*) un code de Hamming (cyclique) de longueur *n*=7, de polynôme générateur $g(X) = X^3 + X + 1$, donc sa matrice génératrice est donnée par : $G=\begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,0\,1\,1\,0\,1 \end{pmatrix}$.

En appliquant la méthode de Gauss sur les lignes de G on trouve la matrice génératrice normalisé $G_N$,

$$G_N=\begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,1 \end{pmatrix}$$

et donc C admet comme matrice génératrice

$$H=\begin{pmatrix} 1\,0\,0\,1\,0\,0\,0 \\ 0\,1\,0\,1\,1\,0\,0 \\ 0\,0\,1\,0\,1\,1\,0 \end{pmatrix}$$

En utilisant la matrice *H*, on peut montrer que la distance *d*=3 et donc la capacité *e*=1.

Soit *m* =1010 $\in \mathbb{F}_2^4$ un message clair. Supposons que Bob veut envoyer ce message à Alice.

1. **Génération des clés.** Alice génère les clés suivants :

a. La clés secrète : La matrice normalisée $G_N$, une matrice de permutation $P_\sigma$ de type $7 \times 7$, une matrice inversible et aléatoire *S* de type $4 \times 4$ par exemple on choisit:

$$P_\sigma=\begin{pmatrix} 0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0\,0 \end{pmatrix} \text{ et } S=\begin{pmatrix} 0\,1\,0\,0 \\ 1\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}$$

b. La clé publique est le couple (*G'*, *e*) tel que $G'=S.G_N.P_\sigma$. Donc

$$G'=\begin{pmatrix} 0\,0\,1\,1\,0\,0\,1 \\ 1\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,0\,1\,0\,0 \end{pmatrix}$$

et la capacité *e*=1.

**Le Chiffrement.** Bob chiffre le message *m* en calculant c'= *mG'*+ε, avec par exemple l'erreur ε=0100000 ∈ $\mathbb{F}_2^7$, de poids w(ε)=1 ≤ *e*=1. Donc le texte chiffré est : c'= *m.G'*+ε

c'=0011010+0100000 = 0111010.

**Le déchiffrement**

On a c'=$m.G'+\varepsilon=m.S.G_N.P_\sigma+\varepsilon \Longrightarrow r'$=c'. $P_\sigma^{-1}$ =$m.S.G_N+\varepsilon.P_\sigma^{-1}=m'+\varepsilon'$ tel que $m' \in$C et

$w(\varepsilon')$=1(car $P_\sigma$ est une permutation et conserve donc le poids des mots)

On calcule $r'$=c'. $P_\sigma^{-1}$= 0111010.$\begin{pmatrix} 0&0&0&1&0&0&0\\ 1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0 \end{pmatrix}$ =1100110. $r'$ est un mot entaché d'erreurs.

En représentation polynomiale le mot $r'$=1100110 correspond au polynôme $r'(X) = X^5 + X^4 + X + 1$. En utilisant la méthode de décodage de Meggitt. On peut corriger le mots $r'(X)$ d'erreur ε'.

Le syndrome de *r'(X)* est le reste de la division Euclidienne de *r'(X)* par $g(X)$. En utilisant un registre à décalage circulaire, on trouve S(*r'(X)*)=*X* et w(S(*r'(X)*)) =1. D'apres l'algorithme de Meggitt L'erreur est $\varepsilon(X) = \mathrm{S}\big(r'(X)\big) = X$. Donc le mot code est $r(X) = r'(X) + \varepsilon(X) = X^5 + X^4 + 1$, le mot correspondant au polynôme $r(X)$ est $r = 1000110$= *m.S.G*$_N$.

Le décodage de *r* (en enlevant la redondance les trois bits de gauche) on obtient le mot

*m'*=*mS*=0110 et donc le message initial *m*= *m'*.*S*$^{-1}$ = 0110.$\begin{pmatrix} 0&1&0&0\\ 1&0&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{pmatrix}$ = 1010. Qui est le message clair transmis.

**Exemple 5.2.** Soit C (*n*=15, *k*=7, d) un code cyclique de polynôme générateur

$$g(X) = X^8 + X^4 + X^2 + X + 1.$$

et comme matrice de contrôle

*H*=( $I_8$/*M*) tel que *M*=$\begin{pmatrix} 1&0&0&0&1&0&1\\ 1&1&0&0&1&1&1\\ 1&1&1&0&1&1&0\\ 0&1&1&1&0&1&1\\ 1&0&1&1&0&0&0\\ 0&1&0&1&1&0&0\\ 0&0&1&0&1&1&0\\ 0&0&0&1&0&1&1 \end{pmatrix}$

On trouve que la distance $d \geq 3$ et donc la capacité $e \geq 1$.

Soit $m$ =1010110 $\in \mathbb{F}_2^7$, un message clair. Supposons que Bob veut envoyer ce message à Alice.

2. **Génération des clés.** Alice génère les clés suivants :

La clés secrète : les éléments de la première ligne de la matrice génératrice sont les coefficients du générateur et les autres lignes sont les shifts de la première ligne donc :

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

La matrice normalisée est comme suit :

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

et comme matrice de contrôle

$H$=( $I_8$/$M$) tel que $M$= $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$

Une matrice de permutation $P_\sigma$ de type $15 \times 15$, associée à la permutation

$\sigma$ = (3,11,4,6,13,5,14,2,10,9,12,7,8,1,15).

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

Sa matrice inverse est :

$$\mathrm{P}_\sigma^{-1} = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{pmatrix}.$$

Une matrice inversible et aléatoire *S* de type $7 \times 7$ par exemple:

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

Sa matrice inverse est :

$$S^{-1} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Apres calcul, on trouve que la clé publique ($G'=S.G_N.P_\sigma$) est la suivante:

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

On choisit comme vecteur d'erreur par exemple ε=100000000000000, tel que w(ε)=1 ≤ *e*.

**Le Chiffrement.** Bob chiffre le message *m* en calculant c'= *mG'*+ε de poids w(ε)=1.

Apres calcul, on trouve le message chiffré

c'= *m.G'*+ε=101111111010100+100000000000000=001111111010100.

**Le déchiffrement**.

On a c'=*m.G'*+ε=*m.S.G.*$P_\sigma$+ε ⟹ r'=c'. $P_\sigma^{-1}$=*m.S.G*+ε. $P_\sigma^{-1}$=*m' G* +ε' tel que w (ε′)=1 (car $P_\sigma$ est une permutation et conserve donc le poids des mots).

On trouve *r'*=c'. $P_\sigma^{-1}$=111111000101100. En représentation polynomiale

$r'(X) = X^{12} + X^{11} + X^9 + X^5 + X^4 + X^3 + X^2 + X + 1$, mot entaché d'erreur $\varepsilon'(X)$.

En utilisant la méthode de décodage par syndrome polynomial. On peut corriger le mot $r'$ d'erreur ε'.

Le syndrome de $r'(X)$ est $S(r'(X)) = X^7 + X^6 + X^3 + X^2 + X$, qui représente le mot

$$h(r') = 01110011 = C_{14} = h(000000000000010) = h(\varepsilon').$$

Et w(ε')=1. L'erreur est donc ε'=000000000000010. Et le mot code est *r*=*m.S.G* = r'+ε',

*r* = 111111000101100 + 000000000000010 = 111111000101110.

On a *: r= m.S.G*=111111000101110. On enlève la redondance c.à.d. les huit premier bits, on trouve *m.S* =0101110= *m*' et donc le message initial :

$m=m'.S^{-1}=(0101110)\begin{pmatrix}0&0&1&1&1&0&1\\1&0&0&0&0&0&0\\1&0&1&0&0&0&0\\0&1&1&0&0&1&0\\0&1&1&0&0&0&1\\0&0&1&0&1&0&1\\0&1&0&1&0&0&0\end{pmatrix}$=1010110, qui est le mot clair.

## 5.3 Cryptosystème de Niederreiter

### 5.3.1 Historique et principe du Cryptosystème de Niederreiter

Le cryptosystème de Niederreiter est un système de chiffrement à clé publique, inventé par Harald Niederreiter en 1978 et qui a été mis au point en 1986. Ce système appartient à la famille des cryptosystèmes basés sur les codes à clé publique et est similaire au cryptosystème de McEliece, mais il diffère dans les détails de sa mise en œuvre. Le cryptosystème de Niederreiter est réputé pour sa robustesse théorique contre les attaques quantiques et sa sécurité basée sur des problèmes mathématiques difficiles.

Le cryptosystème de Niederreiter fonctionne de la même manière que celui de McEliece:

1. Génération des clés
2. Construction de la matrice de décodage.
3. Chiffrement.
4. Déchiffrement.

La force principale du cryptosystème de Niederreiter réside dans sa résistance supposée aux attaques quantiques, notamment aux attaques de factorisation quantique qui menacent la sécurité de systèmes cryptographiques classiques. De plus, contrairement au cryptosystème de McEliece, la clé publique du cryptosystème de Niederreiter est généralement plus petite, ce qui le rend plus rapide et plus efficace en termes de performances de chiffrement et de déchiffrement.

Cependant, la mise en œuvre du cryptosystème de Niederreiter peut être complexe et nécessite des calculs mathématiques avancés, ce qui peut limiter son adoption dans certaines applications. Néanmoins, en raison de sa sécurité théorique et de sa résistance aux attaques quantiques, il reste un sujet de recherche actif en cryptographie, en particulier à mesure que les technologies quantiques continuent de se développer.

### 5.3.2 Algorithme du Cryptosystème de Niederreiter

**1. Génération de clé**

On commence par générer un code cyclique $C(n, k, d)$ et une matrice de contrôle H de taille $n\text{-}k \times n$. On va mélanger cette matrice pour la rendre indistinguable d'une matrice aléatoire, pour cela on a besoin de :

1. La clé secrète est composée des trois matrices :
   a. Une matrice de contrôle $H$ (et donc la matrice normalisée $H_N$) du code C.
   b. Une matrice de permutation aléatoire $P_\sigma$ de taille $n \times n$ associée à une permutation $\sigma$ du groupe symétrique $S_n$.
   c. Une matrice inversible aléatoire $M$ de taille $n\text{-}k \times n\text{-}k$.
2. La clé publique sera le couple $(H', t)$ où $H' = M.H_N.P_\sigma$ et $t \leq e$ tel que $e$ est la capacité de correction de $C$.

**2. Chiffrement.**

Soit $y$ un message de $n$ bits que l'émetteur Bob veut chiffrer et de poids $w(y)= t \leq e$. Le mot chiffré est le mot $z = y.{}^tH'$ de type $1 \times n\text{-}k$.

3. **Déchiffrement.**

Pour déchiffrer en connaissant $P_\sigma$, $M$ et $H_N$, Alice suit les étapes suivantes :

a. Calcule: $s= z.{}^tM^{-1}=(y.{}^tP_\sigma).{}^tH_N$ de type $1 \times n\text{-}k$ qui est un mot syndrome.
b. En utilisant un algorithme de décodage des codes cycliques, Alice retrouve le mot $z'= y.{}^tP_\sigma$ correspondant au syndrome $s$.
c. Le récepteur retrouve enfin le mot $y=z'.{}^tP_\sigma^{-1}$.

**Exemple 5.3.** Soit C(7, 4, $d$) un code de Hamming (cyclique) de longueur $n$=7, de polynôme générateur $g(X) = X^3 + X + 1$, donc son polynôme de control est :

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

Le polynôme générateur du code dual du code $C$ est :

$$h_1(X) = X^4 h(X^{-1}) = X^4 + X^3 + X^2 + 1.$$

et donc C admet comme matrice de contrôle

$$H = \begin{pmatrix} 1\,0\,1\,1\,1\,0\,0 \\ 0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{pmatrix}$$

La matrice normalisée $H_N$ associée à $H$ est :

$$H_N = \begin{pmatrix} 1\,0\,0\,1\,0\,1\,1 \\ 0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{pmatrix}$$

En utilisant la matrice $H_N$, on peut montrer que la distance $d$=3 et donc la capacité $e$=1.

Soit $m$ =1010 $\in \mathbb{F}_2^4$ un message clair. Supposons que Bob veut envoyer ce message à Alice.

3. **Génération des clés.** Alice génère les clés suivantes :

c. La clés secrète : La matrice normalisée $H_N$, une matrice de permutation $P_\sigma$ de type $7 \times 7$, une matrice inversible et aléatoire $M$ de type $3 \times 3$ par exemple on choisit:

$$P_\sigma = \begin{pmatrix} 0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0\,0 \end{pmatrix} \text{ et } M = \begin{pmatrix} 0\,1\,0 \\ 1\,0\,0 \\ 0\,0\,1 \end{pmatrix}$$

L'inverse de M est elle-même, car M est une matrice de permutation symétrique.

d. La clé publique est le couple ($H'$, $e$) tel que $H'=M.H_N.P_\sigma$.

Apres calcul on trouve

$$H' = \begin{pmatrix} 1\,0\,1\,1\,0\,1\,0 \\ 1\,1\,0\,0\,1\,1\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{pmatrix}$$

et la capacité $e$=1.

**Le Chiffrement.** Soit $y(X) = X^4 + X^3 + X + 1$ qui correspond au mot $y$=0001000 le mot reçu dont le poids de l'erreur ne dépasse pas 1. Bob chiffre le message $y$, en calculant le mot chiffré $z = y.{}^tH'$. On trouve que le texte chiffré est :

$$z = 1\,0\,0$$

**Le déchiffrement**

Pour le déchiffrement on calcule: $s = z.{}^tM^{-1} = z.{}^tM = 1\,0\,0 \begin{pmatrix} 0\,1\,0 \\ 1\,0\,0 \\ 0\,0\,1 \end{pmatrix} = 0\,1\,0 \neq 000$.

Théoriquement $s=y.{}^t P_\sigma.{}^tH_N$. c.à.d. $s$ représente le syndrome du mot $y' = y.{}^tP_\sigma$ et comme $s \neq 0$, donc y' n'est pas un mot de C et donc nécessite une correction.

Pour la correction de y', on va utiliser la méthode de décodage par syndrome. On a

s = $C_2$ (s est la deuxième colonne de $H_N$) donc s=$h$(ε) tel que ε=010000000000000 et le poids w(ε)=1=$e$. Donc les mots $y' = y.{}^tP_\sigma$ et ε ont le même syndrome et le même poids=1 alors, d'après la Proposition 2.8, on déduit que $y' = \varepsilon$ d'où $y.{}^tP_\sigma=\varepsilon$, ce qui donne que $y = \varepsilon.{}^tP_\sigma{}^{-1}$,

comme P est une matrice de permutation symétrique alors, $P_\sigma^{-1} = {}^tP_\sigma = P_\sigma$ et le mot déchiffré est : $y = \varepsilon . P_\sigma$=0001000, qui est bien le mot transmis.

**Exemple 5.4.** Considérons le code ($n$ = 15; $k$ = 7; $d$ = 5) de longueur $n$ = 15 sur le corps de Galois $F_{16}$, de racine primitive α et de polynôme primitif $M_\alpha(X) = X^4+X+1$.

On a $F_{16} = \{0, \alpha^i / 0 \le i \le 14\}$ avec $\alpha^4 = \alpha + 1$. Considérons la matrice de contrôle $H$:

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

qu'on peut mettre sous forme systématique : $H_N=(I_8 /A)$ tel que

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

On considère la matrice

$$M=\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&0\\ 1&1&1&1&0&0&0&0\\ 1&1&1&1&1&0&0&0\\ 1&1&1&1&1&1&0&0\\ 1&1&1&1&1&1&1&0\\ 1&1&1&1&1&1&1&1 \end{pmatrix},$$

d'inverse

$$M^{-1}=\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&1&1&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&1&1 \end{pmatrix}$$

et la matrice de permutation $P = P_\sigma$, tel que σ est la transposition $\tau_{17}$.

Après calcul de $H' = M.H.P$, on trouve

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

La clé publique est ($H'$, $e$=2). Soit le texte clair qu'on veut chiffrer et envoyer : $y$=010000000000001 de poids $t$=2.

Le chiffrement de $y$ est le mot $z$ donné par : $z = y.{}^tH'$. En remplaçant $y$ et $H'$ on trouve: $z$=10000000.

Pour le déchiffrement on calcule: $s= z.{}^tM^{-1}$=11000000 ≠ 0. Théoriquement $s=y.{}^tP.{}^tH_N$. c.à.d. $s$ représente le syndrome du mot $y' = y.{}^tP$ et comme $s \neq 0$ donc $y'$ n'est pas un mot de C.

Pour la correction de $y'$, on va utiliser la méthode de décodage par syndrome. On a $s = C_1+C_2$ (s est la somme de la première et la deuxième colonne de $H_N$) donc $s=h(\varepsilon)$ tel que $\varepsilon$=110000000000000 et le poids $w(\varepsilon)=2=e$. Donc les mots $y' = y.{}^tP$ et $\varepsilon$ ont le même syndrome et le même poids alors, d'après la Proposition 2.8, on déduit que $y' = \varepsilon$ d'où $y.{}^tP=\varepsilon$ ce qui donne que $y = \varepsilon.{}^tP^{-1}$, comme P est une matrice de permutation symétrique alors, $y = \varepsilon. P$=010000000000001, qui est bien le mot transmis.

## 5.4 Comparaison des cryptosystèmes McEliece, Niederreiter et RSA.

| | McEliece | Niederreiter | RSA |
|---|---|---|---|
| Taille de la clé publique. | *Kn* | *k(n-k)* | *2n* |
| | *67072 octets* | *32750 octets* | 256 octets |
| Nombre de bits d'information transmis par chiffrement. | *K* | *a=log2*($C_n^e$) | *N* |
| | *512* | *276* | 1024 |
| Taux de transmission. | *k/n* | *Log2*($C_n^e$)*/n-k* | 1 |
| | 51,17% | 56,81% | 100% |
| Nombre d'opérations binaires du chiffrement par bit d'information. | n/2+ *n/k* | *(n-k)ke/an+n/a* | $125.3^{m-1}/2^m$ |
| | 513,9 | 50,1 | 2402,7 |
| Nombre d'opérations binaires du déchiffrement par bit d'information. | *B/k* | *C/a* | $25.3^{m-1}/2$ |
| | 5140 | 7863,3 | 738112,5 |
| $B=n+mnt+4m^2t^2+2mt+mn(2t+1)+k^2/2$ et $C=2n+4m^2t^2+2m^2t+mn(2t+1)+(n-k)^2/2$. | | | |

Tableau 5.1 Tableau de comparaison des cryptosystèmes McEliece, Niederreiter et du RSA.

Du tableau ci-dessus on déduit que la taille de la clé au cryptosystème RSA est meilleur que celle de Niederreiter et cette dernière et meilleur que celle de McEliece.

Le taux de transmission dans RSA est paré et il est deux fois meilleur que celui des deux autres cryptosystèmes qui se rapprochent.

Concernant le nombre d'opérations binaires du chiffrement est très couteux dans RSA est le moins couteux dans Niederreiter.

# Appendice

## 1. Programme de décodage d'un code de Hamming par Maple.

```
restart; with(linalg) :

elmin :=proc(ens, p, n) local Pmin, Emin, i, mot, Pmot, t; Pmin := n;
     Emin := op(1, ens); for i from 1 to nops(ens) do; mot := op(i,
     ens); Pmot := 0; for t from 1 to n do; if op(t, mot) ≠ 0 then Pmoi
     := Pmot + 1 fi; od; if Pmot < Pmin then Pmin := Pmot; Emin
     := mot fi; od; Emin; end;

syndrome :=proc(p, n, k, G, H) local An, C, i, t, b, x, S, ens, u, j, el,
     s; An := array[1 ..p··n]; C := array[1 ..p··k]; for i from 0 to p··n
     − 1 do; t := convert(i, base, p); b := [seq(0, j = 1 ..n
     − nops(t))]; An[i + 1] := [op(t), b]; if i < p··k then b
     := [seq(0, j = 1 ..k − nops(t))]; x := convert([op(t), b],
     vector); C[i + 1] := convert(evalm(x&·G), list) mod p; fi; od;
     An := convert(An, set); C := convert(C, set); S := array[1 ..p·
     ·(n − k), 1 ..2]; S[1, 1] := [seq(0, s = 1 ..n)]; S[1, 2]
     := [seq(0, s = 1 ..n − k)]; ens := An minus C; for i from 2 to p·
     ·(n − k) do; S[i, 1] := convert(evalm(H&·u), list) mod p; for j
     from 1 to p··k do; el := S[i, 1] + op(j, C) mod p; ens := ens
     minus {el}; od;od; convert(S, Matrix); end ;

Ldecode :=proc(Y, p, H, S) local s, flag, i, u; s := convert(evalm(H&
     ·y), list) mod p; flag := 1; i := 1; while flag = 1 do; if S[i, 2] = s
     then u := S[i, 1]; flag := 0; fi; i := i + 1;od; Y − u mod p; end;

 p := 2; n := 7; k := 4;

id := Y→diag(seq(1, i = 1 ..Y));

id1 := id(k);

B := matrix([[1, 1, 1], [0, 1, 1], [1, 1, 0], [1, 0, 1]]);

"la matrice Génératrice"; G := concat(id1, B); u := vector(k); uG
     := evalm(u&·G);

"la matrice de controle";

tB := evalm(transpose(B));

id2 := id(n − k);

H := concat(tB, id2);

tH := evalm(transpose(H));

"le code C est";

Lcode :=proc(G) local C, i, t, b, x, j; C := array[1 ..p··k]; for i
     from 0 to p··k − 1 do; t := convert(i, base, p); b := [seq(0, j = 1
     ..k − nops(t))]; x := convert([op(t), b], Vector); C[i + 1]
     := convert(evalm(x&·G), list) mod p; od; convert(C, set); end;
```

```
C := Lcode(G);

"La distance minimale"; Lpoids :=proc(C) local Cstar, Pmin, i, mot,
    Pmot, t; Cstar := C minus {[seq(0, j = 1 ..n)]}; Pmin := n; for i
    from 1 to p··k − 1 do; mot := op(i, Cstar); Pmot := 0; for t
    from 1 to n do; if op(t, mot) ≠ 0 then Pmot := Pmot + 1 fi; od;
    if Pmot < Pmin then Pmin := Pmot fi; od; Pmin; end;

d := Lpoids(C); "représentants des classes latérales et syndrome";

S := Syndrome(p, n, k, G, H); with(LinearAlgebra :-Modular) :

"Le mot reçu"; Y := [1, 1, 0, 1, 1, 0, 0];

YH := evalm(Y&·tH);

"Le Syndrome du mot reçu"; SY := Mod(2, Vector[row]([4, YH]),
    integer[ ]); x := Ldecode (Y, p, H, S);

"Le mot erreur"; e := x − Y;

E := Mod(2, Vector[row] (7, e), integer[ ]);

"Le mot envoyé"; x := Ldecode (Y, p, H, S);
```

**L'exécution du programme**

**Les paramètres du code**

$p := 2$

$n := 7$

$k := 4$

**La matrice de redondance**

$$d1 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B := \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

**La matrice génératrice**

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

**Codage d'un mot.**

$$uG := \left[ u_1 \; u_2 \; u_3 \; u_4 \; u_1 + u_3 + u_4 \; u_1 + u_2 + u_3 \; u_1 + u_2 + u_4 \right]$$

**La matrice de contrôle.**

$$id2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ et } \quad tB := \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

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

**la matrice transposée de H.**

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

**Le code C.**

$$C := \{[0,0,0,0,0,0,0],[0,0,0,1,1,0,1],[0,0,1,0,1,1,0],[0,0,1,1,0,1,1],[0,1,0,0,0,1,1],[0,1,0,1,1,1,0],[0,1,1,0,1,0,1],[0,1,1,1,0,0,0],[1,0,0,0,1,1,1],[1,0,0,1,0,1,0],[1,0,1,0,0,0,1],[1,0,1,1,1,0,0],[1,1,0,0,1,0,0],[1,1,0,1,0,0,1],[1,1,1,0,0,1,0],[1,1,1,1,1,1,1]\}$$

**La distance minimale**

$$d := 3$$

**Représentants des classes latérales et syndromes (Tableau standard réduit)**

$$S := \begin{bmatrix} [0,0,0,0,0,0,0] & [0,0,0] \\ [1,0,0,0,0,0,0] & [1,1,1] \\ [0,1,0,0,0,0,0] & [1,1,0] \\ [0,0,1,0,0,0,0] & [1,1,0] \\ [0,0,0,1,0,0,0] & [1,0,1] \\ [0,0,0,0,1,0,0] & [1,0,0] \\ [0,0,0,0,0,1,0] & [0,1,0] \\ [0,0,0,0,0,0,1] & [0,0,1] \end{bmatrix}$$

**Le mot reçu.**

$$Y := [1,1,0,1,1,0,0]$$

**Le syndrome du mot reçu.**

$$SY := [1,0,1]$$

**Le mot erreur.**

$$e := [0,0,0,1,0,0,0]$$

**Le mot envoyé.**

$$x := [1,1,0,0,1,0,0]$$

## 2. Liste des polynômes primitifs de degré *n* sur $F_p$ .

2.1. Quelques polynômes primitifs *P(X)* de degré *n* sur le corps $F_2$ qui nous aident à décrire des corps de Galois $F_{2^n}$ sont donnés au tableau ci-dessous :

| *n* | *P(X)* | *n* | *P(X)* |
|---|---|---|---|
| 2 | $X^2 + X + 1$ | 14 | $X^{14} + X^{10} + X^6 + X + 1$ |
| 3 | $X^3 + X + 1$ | 15 | $X^{15} + X + 1$ |
| 4 | $X^4 + X + 1$ | 16 | $X^{16} + X^{12} + X^3 + X + 1$ |
| 5 | $X^5 + X^2 + 1$ | 17 | $X^{17} + X^3 + 1$ |
| 6 | $X^6 + X + 1$ | 18 | $X^{18} + X^7 + 1$ |
| 7 | $X^7 + X^3 + 1$ | 19 | $X^{19} + X^5 + X^2 + X + 1$ |
| 8 | $X^8 + X^4 + X^3 + X^2 + 1$ | 20 | $X^{20} + X^3 + 1$ |
| 9 | $X^9 + X^4 + 1$ | 21 | $X^{21} + X^2 + 1$ |
| 10 | $X^{10} + X^3 + 1$ | 22 | $X^{22} + X + 1$ |
| 11 | $X^{11} + X^2 + 1$ | 23 | $X^{23} + X^5 + 1$ |
| 12 | $X^{12} + X^6 + X^4 + X + 1$ | 24 | $X^{24} + X^7 + X^2 + X + 1$ |
| 13 | $X^{13} + X^4 + X^3 + X + 1$ | 25 | $X^{25} + X^3 + 1$ |

Tableau A.1 Quelques polynômes primitifs de degré *n* sur le corps $F_2$

2.2. Quelques polynômes primitifs *P(X)* de degré *n* sur le corps $F_3$ qui nous aident à décrire des corps de Galois $F_{3^n}$ sont donnés au tableau ci-dessous :

| *n* | *P(X)* |
|---|---|
| 2 | $X^2 + X + 2$ |
| 3 | $X^3 + 2X + 1$ |
| 4 | $X^4 + X + 2$ |
| 5 | $X^5 + 2X + 1$ |
| 6 | $X^6 + X + 2$ |
| 7 | $X^7 + X^2 + 2X + 1$ |
| 8 | $X^8 + X^3 + 2$ |
| 9 | $X^9 + 2X^3 + X^2 + 1$ |
| 10 | $X^{10} + X^3 + X$+2 |
| 11 | $X^{11} + X^2 + 2X + 1$ |

Tableau A.2 Quelques polynômes primitifs de degré *n* sur le corps $F_3$

## 3. Registres à décalage

### 3.1 Bascule et registre

Une **bascule**, ou **élément de mémoire**, est l'élément de base de la logique séquentielle qui prend la forme d'un rectangle ou d'un carré et qui permet de mémoriser un seul bit.

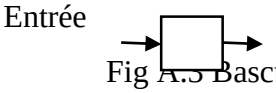



Fig A.3 Bascule

Un **registre à décalage** de *n* bascules est une chaine ordonnée de *n* éléments de mémoire, qui permet de mémoriser une information ou un message de *n* bits.

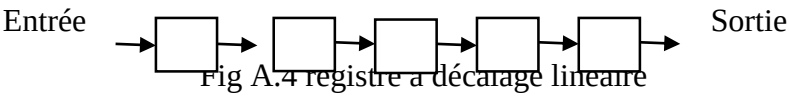



Fig A.4 registre à décalage linéaire

Un décalage transfère le contenu de chacune des cellules vers la cellule qui la suit immédiatement. Après un décalage le contenu de la première cellule est zéro.

Par convention les décalages s'effectuent de la gauche à la droite.

### 3.2 Types de registres

Il existe plusieurs types de registres :

#### 3.2.1 Registre à entrées parallèles et sorties parallèles

- Il peut charger une information sur n bits en même temps.
- Les n bascules changement d'états en même temps.
- Chaque bascule $B_i$ prend la valeur de l'information i.

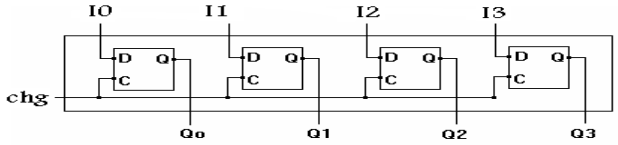



Fig A.5 registre à entrées parallèles et sorties parallèles

#### 3.2.2 Registre à entrée série et sortie série

- L'information est introduite bit par bit ( en série).
- L'ensemble du registre est décalé d'une position ( $B_i$, $B_{i+1}$) et la bascule $B_0$ reçoit une nouvelle entrée ES.
- Un tel registre est appelé registre à entrée série à gauche et à sortie série à droite.

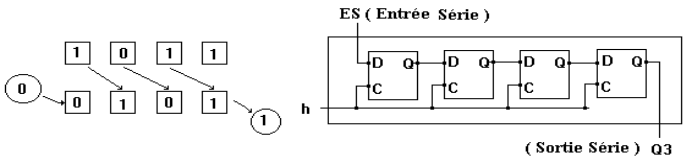



Fig A.6 registre à entrées série et sorties série

### 3.2.3 **Registre à entrée série et sortie parallèle**

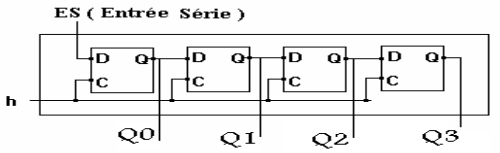



Fig A.7 registre à entrées série et sorties parallèle

### 3.2.4 **Registre à entrée parallèle et sortie série**

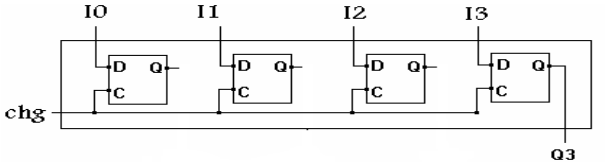



Fig A.8 registre à entrées parallèle et sorties série

### 3.2.5 **Registre à décalage circulaire**

• C'est un registre qui effectue un décalage vers la gauche en répercutant la sortie de la dernière bascule vers l'entrée de la dernière bascule.
• Le décalage peut être un décalage droite (circulaire droite) ou gauche ( circulaire gauche).

Contrairement à un registre à décalage linéaire, où les bits sont déplacés de gauche à droite ou de droite à gauche, un registre à décalage circulaire déplace les bits de manière circulaire, de sorte que le bit le plus à droite est déplacé à la position la plus à gauche sans perdre d'information.

Voici comment cela fonctionne :

1. **Initialisation** : Tout d'abord, le registre à décalage circulaire est initialisé avec une valeur de données.
2. **Décalage** : Ensuite, lorsqu'une opération de décalage est effectuée, chaque bit dans le registre est déplacé d'une position vers la gauche ou la droite, en fonction de la direction du décalage.
3. **Bit circulaire** : Le bit qui est déplacé hors de la position la plus à gauche (lors d'un décalage à gauche) ou de la position la plus à droite (lors d'un décalage à droite) réapparaît à l'opposé. C'est ce qui rend le décalage "circulaire".
4. **Répétition** : Ce processus de décalage circulaire peut être répété autant de fois que nécessaire.

Les registres à décalage circulaires sont utilisés dans diverses applications, notamment en cryptographie, pour l'implémentation de mémoires tampons circulaires, de générateurs de séquences pseudo-aléatoires, de filtres numériques et d'autres systèmes de traitement de données séquentielles où le décalage circulaire des données est nécessaire.

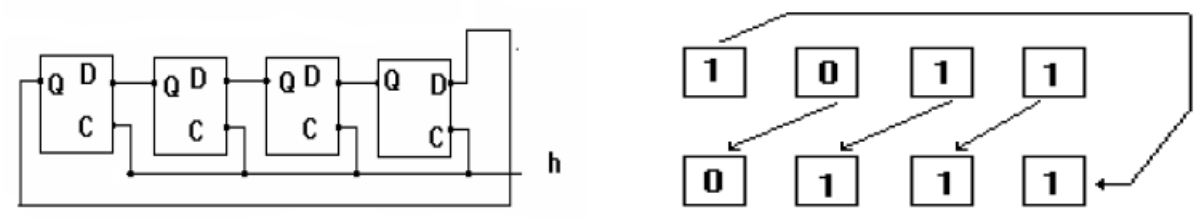



Fig A.9 registre à décalage circulaire

Les registres à décalages circulaires sont utilisés dans le calcul polynomial, en particulier pour le produit et la division Euclidienne de deux polynômes.

Le produit $c(X) = \sum_{i=0}^{n} c_i X^i$, des polynômes $y(X) = \sum_{i=0}^{n} y_i X^i$ et $g(X) = \sum_{i=0}^{t} g_i X^i$, se fait selon le schéma séquentiel suivant :

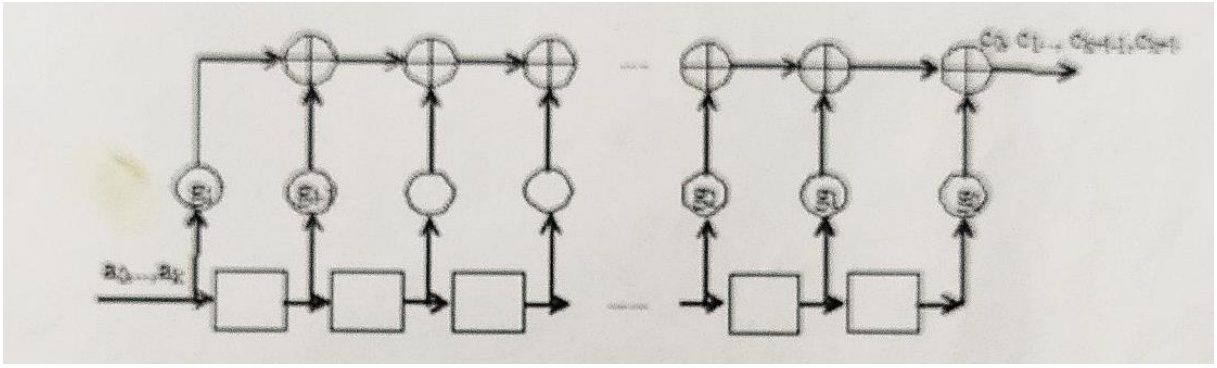


Fig A.10 Circuit à décalage circulaire du produit.

La divisions Euclidienne d'un polynôme $y(X) = \sum_{i=0}^{n} y_i X^i$ par un polynôme $g(X) = \sum_{i=0}^{t} g_i X^i$, se fait selon le schéma séquentiel :

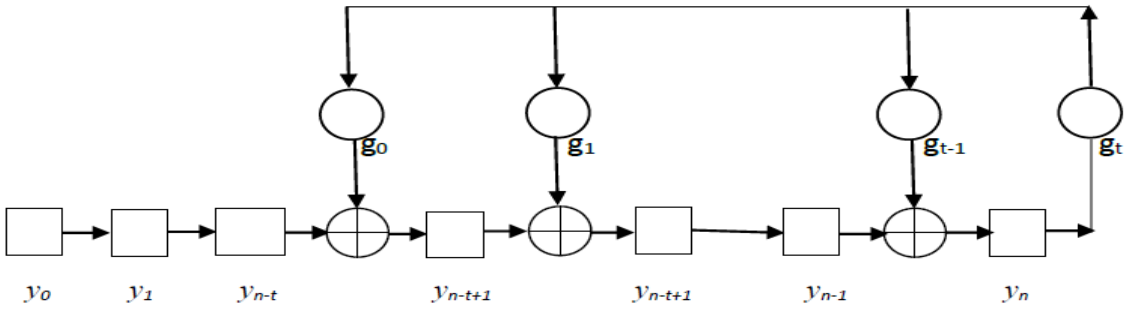



Fig A.9 Circuit à décalage circulaire de la divisions Euclidienne.

# Exercices

# Exercices

**Les exercices de 1 à 19 traitent les codes linéaires et ceux de 20 à 40 sont sur les codes cycliques.**

**Exercice1.**

Soit le code binaire $C$ = {0000, 1011, 0101, 1110}.

1. Quelle est la longueur du code ?
2. Quelle est la distance minimale du code ?
3. Ce code $C$ vérifie-t-il la condition de décodage d'ordre $e$ = 1 ? c.à.d. est-il 1-correcteur ?

**Exercice 2.**

Soit le code ternaire $C$ = {0000, 0121, 1110, 0212, 2220, 1201, 2102, 1022, 2011}.

1. Montrer que C est un code linéaire et donner ces paramètres (n, k, d)?
2. Quelle est la capacité de correction $e$ de ce code ?
3. Déterminer une matrice génératrice G et une matrice de contrôle de C.

**Exercice 3.**

On considère le code linéaire binaire $C(n, k, d)$ définie par une matrice de contrôle

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

1. Quelles sont la longueur $n$, la dimension $k$ et la distance d de ce code?
2. Montrer que $C$ est systématique et donner sa matrice génératrice $G_N$.
3. Trouver le mot code $c$ provenant du mot $x$=1010 par $G_N$.
4. Décoder si possible les mots $y_1$=11111000, $y_2$=11000001, $y_3$=01011101

**Exercice 4.**

Soit $C(n, k, d)$ un code linéaire de matrice de contrôle $H$ et $r \in \mathbb{N}^*$.

Montrer que : $d \geq (r+1)$ si et seulement si tout sous-ensemble de $r$ colonnes de $H$ est libre.

**Exercice 5.**

On considère le code linéaire trinaire (sur le corps $F_3$ ) $C(n, k, d)$ définie par son code orthogonal

$$C^{\perp} = \{(x_1 + x_3, 2x_1 + 2x_2 + x_3,\ x_1 + 2x_3,\ x_2,\ x_3) / \ x_i \in F_3\ \}$$

1. Déterminer une base de $C^{\perp}$ et déduire la longueur n et la dimension $k$ du code $C$?

2. Déduire une matrice de contrôle $H$ de $C$ et sa distance minimale $d$.
3. Montrer que $C$ est systématique et donner sa matrice génératrice normalisée $G_N$, et construire ce code.
4. Décoder si possible les mots $y_1$=22021, $y_2$=21211, $y_3$=11120, $y_4$=11110

**Exercice 6.**

Soit $C(n, k, d)$ un code binaire, et $H$ sa matrice de contrôle.

1. Montrer que la distance minimale $d$ est le plus petit nombre de colonnes de $H$ telles que leur somme soit nulle.
2. Déduire que si tout sous-ensemble de $r$-1 colonnes de $H$ est libre, alors $d \geq r$.

**Exercice 7.**

Soit $d$ la distance minimale d'un code linéaire $C$. Montrer que si $d \geq 3$ alors $C$ vérifie la condition de décodage d'ordre 1.

**Exercice 8.**

Soit $C(n, k, d)$ le code binaire de matrice génératrice $G$ définie par :

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

1. Déterminer les paramètres $n$ et $k$. Montrer que $C$ n'est pas systématique.
2. Montrer que $C$ est équivalent à un code systématique $C_s$ (en appliquant la permutation $\tau_{14}$) qu'on détermine sa matrice génératrice normalisée $G_N$. Construire le code $C_s$.
3. Déduire une matrice de contrôle $H_N$ du code $C_s$. Calculer par deux méthodes la distance $d$.
4. Corriger si possible les mots suivants : $y_1$=111100, $y_2$=111101, $y_3$=100010.

**Exercice 9.**

Soit $C(n, k, d)$ le code binaire de matrice de contrôle $G$ définie par :

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

1. Montrer que $C$ est systématique et déterminer les paramètres $n$ et $k$.
2. Détermine sa matrice génératrice normalisée $G_N$. Construire le code $C$.
3. Corriger si possible les mots suivants : $y_1$=0001110, $y_2$=0010100.

**Exercice 10.**

Soit $C(n, k, d)$ un code binaire sur l'alphabet $A$ (c.à.d. $A$={0, 1}, $q$=|$A$|=2), $r \in \mathbb{N}^*$

1. Pour tout $x \in A^n$, on note $B(x, r)=\{y \in A^n : d(x, y) \leq r\}$ la boule de centre $x$ et de rayon $r$ et

$S(x, i) = \{y \in A^n : d(x, y) = i\}$ la sphère de centre $x$ et de rayon $i$.

Montrer que $|B(x, r)| = \sum_{i=1}^{i=r} |S(x, i)| = \sum_{i=0}^{i=r} C_n^i$.

2. Si le code $C$ est de capacité e et Sachant que : $\bigcup_{x \in C} B(x, e) \subset A^n$.

Montrer que: $|C| \leq \frac{2^n}{\sum_{i=0}^{i=e} C_n^i}$ .

**Remarques.** 1) | . | représente le cardinal. 2) $C_n^i = \frac{n!}{i!(n-i)!}$

**Exercice 11.**

Soit $C(n, k, d)$ un code linéaire binaire, on définit le code étendu $C'(n', k', d')$ comme suit :

$$C' = \{ (x_1, \ldots, x_n, x_{n+1}) \in F_2^{n+1} \text{ tels que } (x_1, \ldots, x_n) \in C \text{ et } \sum_{i=1}^{i=n+1} x_i = 0 \}.$$

1- Déterminer les paramètres $n', k'$ respectivement en fonction des paramètres $n, k$.

2- Donner selon la polarité de $d$, la distance $d'$ en fonction de $d$.

**Exercice 12.**

Soit $C(n, k, d)$ un code linéaire sur un corps fini $\mathbb{K}$

1. Montrer que $C$ peut détecter $d$-1 erreurs et peut corriger [($d$-1)/2] erreurs.
2. Si $d = 2t+1$ (impaire) et $u, v$ deux mots de $\mathbb{K}^n$ tel que $w(u) \leq t$ et $w(v) \leq t$, montrer qu'ils ont des syndromes différents, et que $\sum_{i=1}^{i=t} C_n^i \leq 2^{n-k}$.

**Exercice 13.**

Soit $C(n, k, d)$ le code binaire de matrice de contrôle $H$ définie par :

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

1. Déterminer les paramètres $n$ et $k$. Montrer que $C$ est systématique.
2. Détermine sa matrice génératrice normalisée $G_N$. Construire le code $C$.
3. Calculer la distance $d$.
4. Calculer la capacité de correction $e$ et corriger si possible les mots suivants :

$$y_1 = 1011101,\ y_2 = 1110111,\ y_3 = 0010111.$$

**Exercice 14.** On considère un code de Hamming $C(7,4)$.

1. Coder le message suivant : $x$=010110010111
2. Décoder le message suivant : $y$=010001110010101101001

**Exercice 15.** On considère le code linéaire en blocs défini par une matrice de contrôle

$$H = \begin{pmatrix} 1\,0\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,0\,1\,0\,0 \\ 1\,1\,0\,1\,0\,0\,1\,0 \\ 1\,1\,1\,1\,1\,1\,1\,1 \end{pmatrix}$$

obtenue en rajoutant à la matrice de contrôle du code C(7,4,3), une colonne de zéros puis une ligne de uns.

1. Quelles sont la longueur *n* et la dimension *k* de ce code ?
2. A quoi correspond pratiquement la modification du code de Hamming ?
3. Mettre la matrice *H* sous forme systématique.
4. Trouver une matrice génératrice *G* de ce code.
5. Montrer que ce code détecte toutes les mots de deux erreurs et corrige toutes les configurations d'une erreur.

**Exercice 16.** Taille de paquets et taux de transfert (rendement)

L'objet de cet exercice est de comparer les taux de transmission et la fiabilité d'un code par répétition et un code de Hamming. Le but est de démontrer que dans le cas d'un canal bruité, émettre des paquets longs est plus efficace qu'émettre des paquets courts. On désire transmettre un message de 10000 bits à travers un canal bruité. On considère une probabilité d'erreur p= 0,01.

Codage par répétition : Chaque bit est émis trois fois. Le décodage se fait par un vote à la majorité.

1. Quel est le taux de transmission ?
2. Quelle est la probabilité que le décodage soit incorrect ?
3. Combien des 10000 bits du message ne sont pas correctement transmis ?

Paquets de 9 bits : On considère un code Hamming(9,3). Le message est envoyé sous forme de paquets de 9 bits, de la forme ($s_1$, $s_2$, $s_3$, $t_1$, $t_2$, $t_3$, $t_4$, $t_5$, $t_6$). Les trois premiers bits $s_1$,$s_2$, $s_3$ constituent le message original, les six suivants $t_1$, ... , $t_6$ sont les bits de contrôle.

4. Quel est le taux de transmission ?
5. Combien y a-t-il de configuration possible de 0 , 1 , ou 2 erreurs dans un tel paquet de 9 bits

6. Supposons qu'il existe un codage tel que les 6 bits de contrôle puissent localiser toutes les configurations jusqu'à deux erreurs. Quel est alors la probabilité qu'un tel paquet de 9 bits ne soit pas décodé correctement ?
7. Combien des 10000 bits du message ne sont pas transmis correctement ?

**Exercice 17.** On considère l'ensemble $C$ définie par :

$$C=\{(2x_1 + x_2 + x_3,\ x_1 + 2x_2 + 2x_3,\ 2x_1+x_2 + x_3, x_1 + 2x_2 + 2x_3, x_2,\ 2x_1 + x_2 + x_3,\ 2x_2)/\ x_i \in \mathrm{F}_3\ \}$$

1. Montrer que $C$ est un code linéaire dont on détermine une base, sa longueur $n$ et sa dimension $k$.
2. Donner une matrice génératrice $G$ et déduire que $C$ n'est pas systématique.
3. Soit $C_s$ l'image de $C$ par la transposition $\tau_{23}$. Montrer que $C_s$ est un code systématique dont on détermine sa matrice génératrice normalisée $G_N$ et une matrice de contrôle $H_N$.
4. Construire le code $C_s$ et déduire sa distance $d$ et sa capacité de correction $e$.
5. Décoder si possible les mots $y_1$=0012102, $y_2$=121211.
6. Bob a envoyé un message $m=m_1m_2$ à Alice qui possède comme clés secrètes les matrices $G_N$, $S=\begin{pmatrix}2 & 1\\ 1 & 1\end{pmatrix}$ et $P=P_\sigma$ tel que la permutation $\sigma = \begin{bmatrix}1 & 2 & 3 & 4 & 5 & 6 & 7\\ 1 & 2 & 4 & 5 & 3 & 6 & 7\end{bmatrix}$.
   a. Déduire la clé publique $(G', e)$ que Bob a utilisé pour chiffrer $m$. Quel est le mot $c$ envoyé à Alice ?
   b. Soit $c$=022211 un message reçu par Alice. Quel est le message clair $m$ que Bob a envoyé à Alice ?

**Exercice 18.** Construire le code de Reed-Muller $RM(2, 3)$ de matrice génératrice $G(2, 3)$.

**Exercice 19.** Si $q$=5, construire le code de Hamming 5-aire pour m=2.

**Les exercices suivants sont sur les codes cycliques**

**Exercice 20**

Soit $C(n, k)$ un code cyclique sur le corps $\mathbb{K}=F_{2^r}$ des racines nièmes de l'unité, de générateur

$$g(X) = \sum_{i=0}^{i=t} g_i X^i.$$

1. Montrer que $C$ est un code systématique, et que le mot code associé au mot $a(X) = \sum_{i=0}^{i=k} a_i X^i$ est le mot $c(X) = X^t a(X)$+S($X^t a(X)$) où S représente le syndrome.

2. Pour $n$=7, $g(X) = X^3 + X^2 + 1$. Calculer en utilisant un registre à décalage circulaire, le syndrome S($X^3a(X)$).

**Exercice 21**

Soit C($n$=$2^r$-1, $k$) un code cyclique sur le corps $\mathbb{K}=F_{2^r}$ des racines nièmes de l'unité, de générateur $g(X)=(X-\alpha)(X-\alpha^2)(X-\alpha^4)\ldots(X-\alpha^{2^{r-1}})$, $\alpha$ une racine primitive de $\mathbb{K}$.

1- Montrer que $C$ est un code BCH primitif au sens strict dont on détermine sa distance construite δ.

2- Pour $r$=3,

a- Montrer que $g(X)$ est irréductible sur $F_2$.

b- $C$ est-il un code de Reed-Solomon? Est-il un code de Hamming ? justifier.

**Exercice 23**

Soient $\mathbb{K}=F_{2^r}$, le corps des racines 7èmes de l'unité, $\alpha$ une racine primitive de $\mathbb{K}$.

I) Soit $C$($n$=7, $k$) un code de Reed-Solomon au sens strict sur $\mathbb{K}$, 2-correcteur.

1- Déterminer son générateur $g(X)$ et une matrice de contrôle $H$.

2- Soit $y(X) = \sum_{i=0}^{i=6} y_i X^i$ le mot reçu. Montrer que son syndrome polynomial est :

$S(y(X)) = \sum_{j=1}^{j=4}\left(\sum_{i=0}^{i=6} y_i \alpha^{ji}\right)X^{j-1}$.

3- Décoder par la méthode algébrique le mot $y(X) = \alpha X^5 + \alpha^6 X^6$, sachant que le poids de l'erreur $w$(ε(X))=2.

II) Supposons maintenant le code Reed-Solomon de longueur $n$=7, de générateur le polynôme $g(X) = X^2 + \alpha^4 X + \alpha^3$.

Décoder par la méthode de T.F.D (Transformée de Fourier Discrète) le mot (polynôme) reçu $y(X) = \alpha^3 + X^2$.

**Exercice 24**

Soit $\mathbb{K}=\mathbb{F}_{2^r}$ un corps fini / $r \in \mathbb{N}^*$, α une racine primitive de $\mathbb{K}$..

1. Donner la définition d'un code $C(n, k, d)$ de Reed-Solomon au sens strict de générateur un polynôme

$g(X)$ de degré $t$.

2. Donner les paramètres du code $C(n, k, d, e)$. Est-il M.D.S ?

3. Pour *r*=3. *t*= 3. Calculer le générateur $g(X)$ et le polynôme de control $h(X)$ et déduire une matrice génératrice *G* et une matrice de contrôle *H*.
4. Décoder par la méthode algébrique, le mot $y(X) = \alpha^6 + \alpha X + \alpha^6 X^2$.

**Exercice 25**

Soit *C*(*n*=10, *k*) un code cyclique binaire sur le corps $\mathbb{K} = \mathbb{F}_{2^r}$ des racines 10eme de l'unité sur $\mathbb{F}_2$, de racine primitive α et de générateur $g(X)$.

1. Décrire le corps $\mathbb{K}=\mathbb{F}_{2^r}$ des racines 10eme de l'unité sur $\mathbb{F}_2$ .
2. Déterminer le groupe $G_{10}(\mathbb{K})$ des racines 10eme de l'unité en donnant son générateur $\beta$.
3. Décomposer le polynôme $X^{10}$-1 en produit de polynômes irréductibles sur $\mathbb{F}_2$.
4. Soit le polynôme $g(X) = X^5 - 1$. Montrer que le polynôme de control $h(X) = g(X)$ est un générateur d'un code cyclique C' dont on détermine sa dimension k', une matrice génératrice G' et une matrice de contrôle H'.
5. Soit $g(X) = X^5 + X^3 + X^2 + 1$, le générateur de *C*(12, k). Montrer que le mot code associé à un mot a(X)= $\sum_{i=0}^{i=k-1} a_i X^i$ est $c(X) = X^5 a(X) + S(X^5 a(X))$. S signifie syndrome.
6. Calculer par un registre à décalage circulaire le syndrome de $X^5(X^3+X+1)$ et le mot code associé au mot $X^3+X+1$.

**Exercice 26**

Montrer que le dual d'un code cyclique *C*(*n, k*) est un code cyclique *C'*(*n', k'*) qu'on détermine ces paramètres.

**Exercice 27**

Soit *C*(*n, k, d*) un code cycliques non trivial sur le corps $\mathbb{K}=\mathbb{F}_{2^r}$ des racines *n*ièmes de l'unité sur $\mathbb{F}_2$, de racine primitive α et de générateur $g(X)$.

I- Supposons que la matrice *H* ci-dessous est une matrice de contrôle de *C*:

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} \end{pmatrix}$$

1- Déterminer *n, r* et le polynôme primitif $M_\alpha(X)$ de $\mathbb{K}$ et décrire ce corps.

2- Calculer les classes cyclotomiques de α et $\alpha^3$ et déduire les racines de $M_\alpha(X) et\ M_{\alpha^3}(X)$.

3- Soit c = $(c_0, c_1, \ldots, c_7)$.

Montrer que $c \in C$ si, et seulement si $c(X)$ est un multiple de $M_{\alpha}(X)M_{\alpha^3}(X)$

4- Déduire que $g(X) = M_{\alpha}(X)M_{\alpha^3}(X)$. De quel type de code s'agit-il. Déterminer *k* et *d*.

II- Supposons que $g(X) = M_{\alpha^3}(X)$.

1- Montrer que $C(n, k, d)$ est un code BCH, déterminer k et montrer que $d$=3.

2- Pour se communiquer entre eux ALICE et BOB utilisent la cryptographie de McEliece.

BOB choisit le code $C(n, k, d)$ et choisit comme clés secrètes, la matrice génératrice normalisée $G_N$ de $C$, la matrice inversible $S= \tau_{12}(I_4)$ et la matrice de permutation $P= \tau_{23}(I_7)$.

ALICE et BOB se mettent d'accord sur le chiffrement des lettre de *A* à *P* comme suit :

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000 | 1000 | 0100 | 0010 | 0001 | 1100 | 1010 | 1001 | 0110 | 0101 | 0011 | 1110 | 1101 | 1011 | 0111 | 1111 |

a) Déterminer la clé publique ($G'$, $e$) ou e est la capacité de correction de $C$.

b) BOB chiffre le mot **m=NON** et l'envoya à ALICE. Quel est le message chiffré **c** reçu par ALICE.

c) Supposons que ALICE a reçu le message **c=**0010110 **0001010** 0010110**.** Utiliser la méthode de piégeage d'erreurs, pour déterminer le message **m** que BOB a envoyé à ALICE?

**Exercice 28**

Soit $C(n=2^r-1, k, d)$ un code cyclique sur le corps $\mathbb{K}=\mathbb{F}_{2^r}$ des racines *n*iemes de l'unité sur $\mathbb{F}_2$, de racine primitive α et de générateur $g(X)$.

1- Montrer que si $g(X) = (X-\alpha)(X-\alpha^2)\ ...\ (X-\alpha^t)$, tel que $t$ >1. Alors C admet la matrice $H$ suivante comme matrice de contrôle :

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^t & \alpha^{2t} & \cdots & \alpha^{t(n-1)} \end{pmatrix}$$

2- De quel type de code s'agit-il ? Montrer que $d=t+1$.

3- Pour $r$=3 et $t$=4. Donner les paramètres du code $C$ et développer le polynôme $g(X)$.

4- Soit $y(X) = X^3 + \alpha X^2 + \alpha^3 X$, le mot reçu ayant $v$=2 erreurs.

Décoder le mot $y(X)$ en utilisant la méthode algébrique.

**Exercice 29**

1. Soit $\mathbb{K}$ le corps des racines 7$^{eme}$ de l'unité sur $\mathbb{F}_2$, de racine primitive $\alpha$ et de polynôme primitif $M_\alpha(X)$ et soit $C(n=7, k, d)$ le code linéaire sur $\mathbb{K}$ dont son code orthogonal (dual) $C^\perp$ est engendré par la matrice $H$ donnée par:

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

a. Décrire le corps $\mathbb{K}$ et décomposer $X^7 - 1$ en produit de polynômes irréductibles sur $\mathbb{F}_2$.

b. Montrer que $C$ est un code cyclique et déterminer son polynôme de control $h(X)$ et son polynôme générateur $g(X)$.

2. Déterminer les racines du polynôme $g(X)$ dans $\mathbb{K}$ et déduire que C est un code BCH dont on détermine sa distance d et sa capacité de correction e.

3. Soit le mot reçu $y(X) = X^6 + X^5 + X^4 + X + 1$.

a. En utilisant un circuit à décalage circulaire, calculer le syndrome de $y(X)$?

b. Décoder le mot $y(X)$ par la méthode de Meggitt.

**Exercice 30**

Soit $C(n, k)$ un code cycliques binaire sur le corps $\mathbb{K}=\mathbb{F}_{2^r}$ des racines nièmes de l'unité sur $\mathbb{F}_2$, de racine primitive $\alpha$ et de générateur $g(X)$.

I- supposons que la matrice de contrôle $H$ est définie par :

$$H = (1 \quad \alpha \quad \alpha^2 \quad \alpha^3 \quad \alpha^4 \quad \alpha^5 \quad \alpha^6)$$

1. Déterminer $n$ et $r$ et le polynôme primitif $M_\alpha(X)$ et déduire le corps $\mathbb{K}$.
2. Déterminer les classes cyclotomiques de $\mathbb{K}^*= \mathbb{K}-\{0\}$ et déduire la décomposition du polynôme $X^n - 1$ sur $\mathbb{F}_2$ en polynômes minimaux (irréductibles).
3. Soit c = $(c_0, c_1, \ldots, c_6)$.

Montrer que $c \in C$ si et seulement si $c(X)$ est un multiple de $M_\alpha(X)$

4. Déduire que g(X) = $M_\alpha(X)$ et que $C$ est un code BCH primitif au sens strict dont on détermine sa dimension $k$ et sa distance construite $\delta$.
5. Supposons $y(X) = X^5 + X^4 + X^3 + X^2 + X + 1$ le mot reçu.

a- Calculer par un registre à décalage circulaire S($y(X)$) le syndrome de $y(X)$.

b- Donner l'algorithme de Meggitt et décoder le mot $y(X)$ par cette méthode.

II- supposons que $g(X) = M_\alpha(X)\, M_{\alpha^3}(X)$.

1. Déterminer tous les racines de $g(X)$, et déduire que *C* est un code BCH et déterminer sa dimension k et montrer dans ce cas que d=7.
2. Soit $y(X) = \alpha^2 X^6 + \alpha X^5$ le mot reçu contenant *v*=2 erreurs. Calculer le polynôme localisateur et ces racines.
3. Donner l'algorithme de décodage algébrique du code et décoder le mot y(X) par cette méthode.

**Exercice 31**

Soit *C*(7, *k*) un code cycliques binaire sur le corps $\mathbb{K}=\mathbb{F}_{2^r}$ des racines 7^iemes^ de l'unité sur $\mathbb{F}_2$, de distance minimale *d*, de racine primitive α et de générateur $g(X)$.

1. Montrer que si $g(\alpha) = g(\alpha^2) = g(\alpha^3) = 0$, alors la matrice *H* suivante est une matrice de contrôle

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} \end{pmatrix}$$

2. En déduire qu'il existe un entier δ tel que $d \geq \delta$.
3. Si $\delta=n$ déduire dans ce cas que *C* est de distance $d=n$.

**Exercice 32**

I. **Code cyclique**

1. Soit *C*(*n, k, d*) le code cyclique sur le corps $\mathbb{K}$ des racines 7^eme^ de l'unité sur $\mathbb{F}_2$, de racine primitive α, engendré par $g(X) = X^4 + X^3 + X^2 + 1$. Décrire le corps $\mathbb{K}$ et donner la longueur *n*, la dimension *k*.
2. Déterminer une matrice génératrice de C, de la forme $G_S=(I_k/A)$ et déduire une matrice de contrôle *H*.
3. Déterminer la distance *d* et la capacité de correction e de *C*.
4. Soit le mot reçu $y(X) = X^6 + X^5 + X^3 + X^2 + 1$.

a. En utilisant un circuit à décalage circulaire, calculer le syndrome de *y*(*X*).
b. Donner l'algorithme de la méthode de piégeage d'erreurs et décoder le mot *y*(*X*) par cette méthode**.**

**II- Application à la cryptographie de McEliece**

Pour se communiquer entre eux, **ALICE** et **BOB** utilisent la cryptographie de **McEliece**.

**BOB** choisit le code $C(n, k, d)$ dans la partie **(I)** et choisit comme clés secrètes, la matrice génératrice $G_S$ de $C$, la matrice inversible $S=\begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{bmatrix}$ et la matrice de permutation $P=P\tau_{23}(I_n)$.

**ALICE** et **BOB** se mettent d'accord sur le chiffrement des lettres suivantes :

| A | B | C | D | E | F | M | N |
|---|---|---|---|---|---|---|---|
| 000 | 100 | 010 | 001 | 110 | 101 | 011 | 111 |

1. Déterminer la clé publique $(G', e)$ ou e est la capacité de correction de $C$.
2. **ALICE** chiffre le mot **m="MFD"** et l'envoya à **BOB**. Quel est le message chiffré **c** reçu par **BOB**.
3. Supposons que **BOB** a reçu le message **c=001010**10101100**0010101**.
   Déchiffrer **c** pour déterminer le message **m** (en lettres) qu'**ALICE** a envoyé à **BOB**?

**Exercice 33** Soit le corps de Galois $\mathbb{K}=\mathbb{F}_8$, $\alpha$ une racine primitive de $\mathbb{K}$.

1. Soit $C(n, k, d)$ un code BCH primitif sur $\mathbb{K}$ de générateur le polynôme $g(X) = (X-\alpha)(X-\alpha^2)(X-\alpha^3)(X-\alpha^4)$. De quel code s'agit-il ? Déterminer les paramètres $n$, $k$ et $d$.
2. Soit le mot reçu $y(X)=\alpha^3X+\alpha X^2+X^5$ contenant deux erreurs.

a- Calculer le syndrome $S=(s_j)$ / $1\leq j\leq d-1$ du mot $y(X)$.

b- Déterminer le polynôme localisateur $\sigma(X)$ et ses racines dans $\mathbb{K}$.

c- Déduire les localisateurs $X_i$.

d- Corriger en utilisant **la méthode Algébrique** le mot $y(X)$.

**Exercice 34**

Montrer que le dual d'un code cyclique $C(n, k)$ est un code cyclique $C'(n', k')$ qu'on détermine ses paramètres.

**Exercice 35**

**I- Code cyclique**

1. Soit $C(n, k, d)$ le code cyclique sur le corps $\mathbb{K}=\mathbb{F}_{2^r}$ des racines $7^{eme}$ de l'unité sur $\mathbb{F}_2$, de racine primitive $\alpha$, engendré par $g(X) = (X-1)(X^3 + X + 1)$. Décrire le corps $\mathbb{K}$.
2. Déterminer la longueur $n$, la dimension $k$ et montrer que la matrice $G_S=(I_k/A)$ tel que:

$A=\begin{bmatrix}1 & 0 & 1 & 1\\ 1 & 1 & 1 & 0\\ 0 & 1 & 1 & 1\end{bmatrix}$ est une matrice génératrice de *C* et déduire une matrice de contrôle *H*.

a. Déterminer la distance d et la capacité de correction e de C.

b. Soit le mot reçu $y(X) = X^6+X^5+ X^3 + X^2+1$

c. En utilisant un circuit à décalage circulaire, calculer le syndrome de $y(X)$?

d. En utilisant la méthode de Meggitt, décoder le mot $y(X)$**.**

**II- Application à la cryptographie de McEliece**

1. Pour se communiquer entre eux, **ALICE** et **BOB** utilisent la cryptographie de **McEliece**.

**BOB** choisit le code $C(n, k, d)$ dans la partie **(I)** et choisit comme clés secrètes, la matrice génératrice $G_S$ de *C*, la matrice inversible $S=\begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{bmatrix}$ et la matrice de permutation $P=P\tau_{23}(I_n)$.

**ALICE** et **BOB** se mettent d'accord sur le chiffrement des lettres suivantes:

| A | B | C | D | E | F | M | N |
|---|---|---|---|---|---|---|---|
| 000 | 100 | 010 | 001 | 110 | 101 | 011 | 111 |

Déterminer la clé publique ($G'$, $e$) ou e est la capacité de correction de *C*.

2. **ALICE** chiffre le mot **m="MNM"** et l'envoya à **BOB**. Quel est le message chiffré **c** reçu par **BOB**.
3. Supposons que **BOB** a reçu le message **c=0010101**0101100**0010101**.

   Déchiffrer **c** pour déterminer le message **m** (en lettres) qu'**ALICE** a envoyé à **BOB**?

**Exercice 36**

1. Soit $\mathbb{K}= \mathbb{F}_4$ le corps des racines 4^eme de l'unité sur $\mathbb{F}_2$ de racine primitive $\alpha$. Décrire le corps $\mathbb{K}$.
2. Soit le code de Reed-Solomon C(3,1,3) sur $\mathbb{F}_4$ de polynôme générateur $g(X) = X^2 + X + 1$, ce polynôme a 2 racines : $\alpha$ et $\alpha^2$. Soit $y(X) = X + 1$, le mot reçu avec $v$=1 erreur. Corriger en utilisant **la méthode TFD,** le mot $y(X)$.

**Exercice 37**

Soit le corps de Galois $\mathbb{K}=\mathbb{F}_{2^r} / r\in\mathbb{N}^*$, $\alpha$ une racine primitive de $\mathbb{K}$.

1. Donner la définition d'un code $C(n, k, d)$ de Reed-Solomon au sens strict sur $\mathbb{K}$, de générateur un polynôme $g(X)$ de degré $t$.
2. Pour $r$=3, $t$=4. Déterminer le générateur $g(X)$, $n$, $k$ et $d$.
   Décoder par la méthode T.F.D, le mot reçu $y(X) = \alpha^3 + \alpha X^5 + X^6$, sachant qu'il contient deux erreurs.

**Exercice 38**

Soit $C(n, k)$ un code cyclique dont le polynôme générateur $g(X)$ est divisible par $X - 1$.

Montrer alors que tout mot reçu $y(X)$ de longueur n, comportant un nombre **impair** d'erreurs (c.à.d. $w(\varepsilon(X))$ impair) est détecté comme message erroné.

**Exercice 39**

1. Soit le polynôme $M(X)= X^3+ X^2+1$ de $\mathbb{F}_2[X]$, montrer que $M(X)$ est irréductible sur $\mathbb{F}_2$.
2. Soit $\mathbb{K}=\mathbb{F}_2[X]/< M(X) >$ posons $\alpha = \bar{X}$ la classe de $X$.
   Montrer que tout élément $x=\bar{P}$ de $\mathbb{K}$ s'écrit sous la forme $x=a_0+ a_1\alpha+ a_2\alpha^2 / a_i \in \mathbb{F}_2$.
3. Déterminer $\beta$ l'inverse de $\alpha$ dans $\mathbb{K}$.
4. Montrer que $M(X)$ a trois racines dans $\mathbb{K}$ et exprimer les en fonction de 1, $\alpha$, $\alpha^2$.

**Exercice 40**

I. Soit $C(n=12, k)$ un code cycliques trinaire sur le corps $\mathbb{K}= \mathbb{F}_{3^r}$ des racines $12^{eme}$ de l'unité sur $\mathbb{F}_3$, de racine primitive $\alpha$ et de générateur $g(X)$.

1. Décomposer par deux méthodes, le polynôme $X^{12}-1$ en produit de polynômes irréductibles sur $\mathbb{F}_3$.
2. Soit $g(X) = X^2 + 2X^2 + 2X + 1$. Montrer que $g(X)$ est un générateur de $C(12, k)$, déterminer $k$ et montrer que ce code est systématique.
3. Soit le mot $a(X) = X^3 + X + 1$ associé au mot $a$=(1,1,0,1,0,0,0,0,0) de $\mathbb{K}^9$

Calculer par un registre à décalage circulaire le syndrome de $X^3a(X)$ et déduire le mot code $c(X)$ associé à $a(X)$.

II. Soit $\mathbb{K}=\mathbb{F}_{2^r}$ un corps fini / $r \in \mathbb{N}^*$, $\alpha$ une racine primitive de $\mathbb{K}$.

a. Soit $C(n=2^r-1, k, d)$ un code cyclique sur $\mathbb{K}$, de générateur le polynôme $g(X) = (X - \alpha)(X - \alpha^2) \dots (X - \alpha^{2^{r-1}})$. De quel code s'agit-il ?

b. Donner les paramètres du code $C(n, k, d, e)$. Est-il M.D.S ?

# Bibliographie


[1] E.R. Berlekamp, *Algebric Coding Theory*, McGraw Hill, 1968.

[2] J. A. Buchmann, *Introduction à la cryptographie*, Dunod, 2006.

[3] M. Chelgham, Codes correcteurs d'erreurs 1, codage, décodage et cryptage, polycopié de cours Master 1, Université de Jijel, 2019. http://elearning.univ-jijel.dz/mod/resource/view.php?id=14014

[4] G. Cohen, J.L Dornstetter & P. Godlweski, *Codes Correcteurs d'Erreurs*, Masson, CENT, 1992.

[5] B. Courteau, *Mathématiques d'hier et d'aujourd'hui : Les codes correcteurs d'erreurs*, Modulo,1999.

[6] P. Czillag, *Introduction aux Codes Correcteurs d'Erreurs*, Ellipse, 1992.

[7] M. Demazure, *Cours d'algèbre*, Cassini, Paris, 2008.

[8] G. Dubertret, *Initialisation à la cryptographie*, EMS S.A.S , Paris, Novembere 2012.

[9] J.B. Fraleigh, R.A. Beauregard, *Linear Algebra*, Addison-Wesley, Reading, 1995.

[10] W. J. Gilbert, *Moderne Algebra with Applications*, John Wiley and Sons, 1976.

[11] R. Hill, *A First Cours in Coding Theory*, Oxford University Press, 1986.

[12] J.H .V. Lint, *Introduction in Coding Theory*, Springer Verlag, 1999, Third edition.

[13] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, 1977.

[14] R.J. McEliece, *A puplic-key cryptosystem based on algebraic coding theory*, DSN Prog. Rep, Jet Prop. Lab, California Inst. Technol. Pasadena, CA, p 114-116, 1978.

[15] J.M. Monier, *Algébre – MP*, Dunod, 2005.

[16] J.M. Monier, *Algébre – MPSI*, Dunod, 2005.

[17] H. Niederreiter, *Knapsack-type Cryptosystems and Algebraic Coding Theory*, Problems of Control and Information Theory, 15, vol. 1, n° 6, p. 159-166, 1986.

[18] O. Papini, *Etude de techniques de codage par permutation*, Thèse 3$^{eme}$cycle, Université de Provence, 1984.

[19] O. Papini et J. Wolfmann, *Algébre discrète et codes correcteurs*, Springer-Verlag, 1995.

[20] A. Poli, L. Huguet, *Codes Correcteurs* : *Théorie et applications, Logique, Mathématiques, Informatique*, Masson, 1989.

# Bibliographie


[21] J. Vélu, *Méthodes mathématiques pour l'informatique*, *Cours et exercices corrigés*, Dunod, quatrième édition, 2005.

[22] R. Lidl, H. Niederreiter, *Finite Fields*, Cambridge University Press, 1996.

[23] W.W. Peterson, E.J. Weldon, *Error-Correcting Codes*, MIT Press, 1972.

[24] W.E. Ryan, *Modern Coding Theory*, Cambridge University Press, 2009.

[25] G. C. Jr. Clark, J. B. Cain, *Error-Correction Coding for Digital Communications*, Plenum Press, 1981.

[26] W.C. Huffman, V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, 2003.

[27] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing Company, 1977.

[28] S. Lin, D. Costello, *Error Control Coding*, Prentice-Hall, 2004.

[29] R.E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press, 2003.

[30] T.K. Moon, W.C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice-Hall, 2000.

[31] G. Solomon, *Polynomial Codes Over Certain Finite Fields*, Journal of the Society for Industrial and Applied Mathematics, 1961.