# Chapitre 4 Décodage et Cryptage des codes linéaires.

## 4.1 Introduction.

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 codes, 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.

## 4.2 Code orthogonal et Matrice de contrôle.

### 4.2.1 Code orthogonal ou dual.

**Définition 4.2.1.**

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.

$C^{\perp} = \{x \in \mathbb{K}^n, pour\ tout\ y \in C : < x, y >= 0\}$.

**Propposition 4.2.1.** Le code dual de C(*n, k*) est un code linéaire de longueur *n* et de dimension $k' = n - k$.

**Preuve.** $\mathrm{C}^{\perp}$ est un code linéaire car c'est un sous-espace vectoriel de $\mathbb{K}^n$ et comme $\mathrm{C}^{\perp} \subset \mathbb{K}^n$ alors sa longueur est *n* et sa dimension est: $\dim \mathrm{C}^{\perp} = \dim \mathbb{K}^n - \dim \mathrm{C} = n-k$. □

### 4.2.2 Matrice de contrôle ou de parité.

**Définition 4.2.2.**

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 $\mathrm{C}^{\perp}$.

**Exemple 4.2.1.** Soit le code linéaire C(5,3) sur $\mathbb{F}_2$, de matrice génératrice $G=\begin{pmatrix}1&1&1&0&0\\1&1&0&1&0\\1&1&0&0&1\end{pmatrix}$.

C=<$L_1$=(1,1,1,0,0), $L_2$=(1,1,0,1,0), $L_3$=(1,1,0,0,1)> le code orthogonal du code C est:

$C^{\perp} = \{x = (x_1, x_2, x_3, x_4, x_5) \in {\mathbb{F}_2}^5 : \forall y = \alpha L_1 + \beta L_2 + \gamma L_3 \in C \ < x, y >= 0\}$

$= \{x = (x_1, x_2, x_3, x_4, x_5) \in {\mathbb{F}_2}^5 : \forall\ \alpha, \beta, \gamma \in \mathbb{F}_2 : \alpha < x, l_1 > + \beta < x, l_2 > + \gamma < x, l_3 > = 0\}$

$= \{x = (x_1, x_2, x_3, x_4, x_5) \in {\mathbb{F}_2}^5 : < x, L_1 >= 0, < x, L_2 >= 0, < x, L_3 >= 0\}$,

Ces trois égalités nous amène au système d'équations suivant:

$$\begin{cases} x_1 + x_2 + x_3 = 0, \\ x_1 + x_2 + x_4 = 0, \\ x_1 + x_2 + x_5 = 0. \end{cases}$$

Donc $x = (x_1, x_2, x_3, x_4, x_5) = (x_1, x_2, x_1 + x_2, x_1 + x_2, x_1 + x_2) = x_1(1,0,1,1,1) + x_2(0,1,1,1,1)$.

D'où: $C^{\perp}$ = <$v_1$=(1,0,1,1,1), $v_2$=(0,1,1,1,1)>. $v_1$ et $v_2$ sont linéairement indépendants et donc C admet comme matrice de contrôle la matrice: $H=\begin{pmatrix}1&0&1&1&1\\0&1&1&1&1\end{pmatrix}$.

**Remarque 4.2.1.**

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

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

**Théorème 4.2.1.**

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

c∈C si, et seulement, si c.$^t$*H*=0.

**Preuve.** Soit $(L_i)_{i \in \{1,\ldots,n-k\}}$, les lignes de *H*, alors:

c ∈C ⇔ c ∈$(C^{\perp})^{\perp}$

⇔ (∀ *y* ∈ $C^{\perp}$ : <c, *y*>= 0) ⇔ (∀ *i* ∈{1,…, *n*-*k*}: <c, $L_i$>=0)

⇔ (<c, $L_1$>, <c, .$L_2$>, …, <c, $L_{n-k}$>= (0,0,…, 0)

⇔ c.$^t$*H*=0. □

**Exemple 4.2.2.** Dans l'exemple ci-dessus le mot $m_1$=00110 ∈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∉C car

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

**Théorème 4.2.2.**

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

**Preuve**.

$H$ est une matrice de contrôle de $C \Leftrightarrow$ pour tout c ∈C : c.${}^tH$=0

$\Leftrightarrow x.(G.{}^tH)=0$, Pour $x \in K^k$

$\Leftrightarrow G.{}^tH=0$

Inversement Si G. $H^t$=0, en multipliant par $x \in K^k$, on aura $x.( G.{}^tH)=(x.G).{}^tH=0$, en posant c= x.$G$∈C on aura c.${}^tH$=0 avec c∈C et donc $H$ est une matrice de contrôle de $C$. □

### 4.2.3 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 vise versa.

**Théorème 4.2.3.**

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

ii) 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 la matrice génératrice normalisée de $C$ est donnée par: $G = (I_k \mid -{}^tA)$.

**Preuve.**

i) Comme $G = (I_k \mid M)$, alors elle est de rang r= k. En utilisant la multiplication des matrices en bloc on a : $H .{}^tG =(-{}^tM \mid I_{n-k}). (I_k \mid M) = (-{}^tM \mid I_{n-k})\begin{pmatrix} I_k \\ -{}^tM \end{pmatrix} = -{}^tM. I_k + I_{n-k}. {}^tM = -{}^tM. + {}^tM = 0.$

ii) Comme $H = (A \mid I_{n-k})$, alors elle est de rang r= n-k. En utilisant la multiplication des matrices en bloc on a: $H .{}^tG =(A \mid I_{n-k}). {}^t(I_k \mid -{}^tA) = (A \mid I_{n-k})\begin{pmatrix} I_k \\ -{}^tA \end{pmatrix} = A. I_k - I_{n-k} . {}^tA = {}^tA - {}^tA = 0.$ □

**Exemple 4.2.3.** 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 systematique avec matrice generatrice $G_N=\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$. 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 4.2.4.** Soit C(6, 2) un code linéaire trinaire c.à.d. sur le corps fini $K=F_3$ 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 systematique 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 4.2.2.** Si $C_1, ...,C_n$ sont les colonnes de $H$ alors pour tout mot $x= (x_1, x_2, \ldots, x_n)\in K^n$ on a : $x.\,{}^tH=x_1\,C_1 + x_2\,C_2 +... + x_n\,C_n$

**Théorème 4.2.4.**

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

**Preuve.**

($\exists m=(m_1, m_2, ..., m_n) \in C$ tel que $w(m) = r$) $\Leftrightarrow$ ($\exists$ $i_1, i_2, \ldots, i_r$ tel que $m_{i_1}, m_{i_2}, \ldots, m_{i_r}$ non nuls et $m.\,{}^tH=0$.), donc d'après la Remarque 4.2.2. $\exists\, m_{i_1}, m_{i_2}, \ldots, m_{i_r}$ non nuls: $m_{i_1}\,C_{i_1} +... + m_{i_r}\,C_{i_r}= 0$. En posant $\alpha_{i_j} = m_{i_j} / j\in\{1,..,r\}$ alors $\alpha_{i_j}C_{i_1} + \alpha_{i_2}C_{i_2} +... + \alpha_{i_r}C_{i_r}= 0$. □

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

**Corollaire 4.2.1.**

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 4.2.3.**

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 4.2.5.** 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.

## 4.3 Les codes de Hamming et de Reed-Muller.

### 4.3.1 Les codes de Hamming.

Ces codes ont été inventés par *Richard Hamming*, en 1947 et ce sont les premiers codes corrigeant une erreur, ils sont utilisés dans le domaine des "*digital communications*" et des systèmes de sauvegarde de données pour détecter et corriger des erreurs se produisant à l'intérieur des calculateurs et ordinateurs.

**Définition 4.3.1.** On appelle code de *Hamming* (binaire) de longueur $2^m$-1 tout code linéaire binaire de longueur $2^m$-1 admettant comme matrice de contrôle $H$, une matrice dont les $2^m$-1 colonnes sont tous les vecteurs non nuls de $F_2^m$.

**Propriétés 4.3.1.** Un code d'un code de *Hamming* définit ci-dessus a pour longueur $n=2^m$-1, pour dimension $k=2^m$*-1-m*, pour distance $d$=3 et donc pour capacité $e$=1.

**Preuve.** Pour $n$ et $k$ découlent de la définition. Pour montrer $d$=3, il suffit de montrer le nombre minimum de colonnes d'une matrice de contrôle $H$ de C dont la somme est nulle est 3.
Comme les colonnes de H sont tous les vecteurs non nuls de $F_2^n$ donc d ≥ 2. Commes tous ces vecteurs sont distincts (i.e. la somme de deux vecteurs quelconques est non nul) alors $d \geq 3$.
Si $C_i$, $C_j$ sont deux colonnes distincts de $H$ et donc de $F_2^n$ et comme ce dernier est un espace vectoriel alors $C_i+C_j$ est un vecteur de $F_2^n$ et donc est une colonne $C_l$ de H et $C_i+C_j+C_l=0$ et donc $d$=3 et $e$=1. □

**Exemple 4.3.1.**

Le code binaire C suivant de longueur $n$=7 et de dimension $k$=4, définit par sa matrice génératrice normalisée suivante:

$G_N = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$ et une matrice de contrôle $H=\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$ dont les colonnes sont tous les vecteurs de $F_2^3$ est un code de Hamming de paramètres ($n$=7, $k$=4, $d$=3).

**Définition 4.3.2.** On appelle *code simplexe* (binaire) de longueur $2^m$-1 tout code linéaire binaire admettant comme matrice génératrice $G$, une matrice dont les $2^m$-1 colonnes sont tous les vecteurs non nuls de $F_2^m$.

**Exemple 4.3.2.**

Le code binaire C suivant de longueur $n$=7 et de dimension $k$=3, définit par:

sa matrice génératrice normalisée suivante: G = $\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$ et comme matrice de contrôle $H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$ est code simplexe de paramètres ($n$=7, $k$=3, $d$=4).

### 4.3.2 Les codes de Reed-Muller.

Les codes de *Reed-Muller* sont apparu en 1954 et ont étés utilisé par les sondes Mariner lancées par la NASA entre 1969 et 1973 pour assurer une transmission (numérique) correcte des photos de Mars.

Les codes de *Reed-Muller* (R.M) sont des extensions des codes de Hamming, précisément ces derniers sont des codes R.M du premier ordre.

Soit $H_r$ la matrice de contrôle du code binaire de *Hamming* $C_H[2^r - 1, 2^r - 1 - r]$.

Soit $B_r = [H_r \mid 0]$ la matrice $H_r$ au quelle nous avons ajouté une colonne de zéros. Soient $v_1, v_2, \ldots, v_r$ les lignes de $B_r$ et soit enfin $v_0$=**1** le vecteur ligne de longueur $2^r$ dont toutes les composantes sont égales à 1. Nous pouvons alors définir les codes de *Reed-Muller* du premier ordre de la façon suivante:

**Définition 4.3.3.** Le code de *Reed-Muller* du premier ordre, noté RM(1, $r$), est le sous-espace vectoriel engendré par les vecteurs $v_0, v_1, v_2, \ldots, v_r$. La matrice génératrice de RM(1, $r$) est alors

$G = \left[\frac{v_0}{B_r}\right] = \left[\frac{v_0}{H_r\ 0}\right]$.

**Exemple 4.3.3.** Construisons la matrice génératrice de RM(1, 3).

Comme $H_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$, nous avons alors,

$B_3 = [H_3|0] = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$ Nous en déduisons donc que,

$G = \begin{bmatrix} v_0 \\ B_r \end{bmatrix} = \begin{bmatrix} v_0 \\ H_r\ 0 \end{bmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$. On peut vérifier que RM(1, 3) est un code (8, 4, 4).

**Théorème 4.3.1.** Un code $RM(1, r)$ est un code linéaire binaire de longueur $n=2^r$ de dimension $k = r+1$ et de distance $d = 2^{r-1}$.

**Preuve.** Comme $G$ admet $2^r$ colonnes alors la longueur est $n=2^r$. Comme les vecteurs $v_0, v_1, v_2, \ldots, v_r$ sont linéairement indépendants, on en déduit que la dimension est égale à r + 1. Il nous reste alors à montrer que la distance d du code est égale à $2^{r-1}$. Pour cela, nous montrerons que tous les mots du code (sauf 0 et 1) ont un poids égal à $2^{r-1}$. Tout mot $c \neq 0$ du code s'écrit par:
$c = r_{i1} + r_{i2} + \cdots + r_{ih}$, où $r_{ij}$ est la $i_j$-iéme ligne de $G$ , $1 \leq h \leq r+1$. Supposons qu'aucun $r_{ij}$ n'est égal à $v_0$ et considérons alors la matrice $A = \begin{bmatrix} r_{i_1} \\ r_{i_2} \\ \vdots \\ r_{i_h} \end{bmatrix}$. La composante $t$ de $c$ sera nulle si la $t$-iéme colonne de $A$ possède un nombre pair de 1, sinon elle sera égale à 1. Remarquons que pour chaque colonne distincte $u'$ de $A$, le nombre de colonnes $u$ de $B_r$ où la $i_j$-ième composante, $1 \leq j \leq h$ est la même que la $j$ de u' est $2^{r-h}$ car chacune des $r$-$h$ composantes restantes de $u$ est libre de prendre les valeurs 1 ou 0 (par définition de $B_r$ ). On obtient donc que chaque colonne distincte de $A$ apparait $2^{r-h}$ fois dans $A$. Mais comme chaque $h$-uplet binaire et distinct apparait comme colonne de $A$ et que le nombre des $h$-uplets de poids pairs est égal à celui de poids impairs, on obtient que exactement la moitié des composantes de $c$ sont égales à 1 et ainsi que $w(c)=2^{r-1}$.
Si par contre nous avons que $r_{ij} = 1$, alors il suffit de considérer $c$ -1. Ainsi le même argument peut être appliqué (il s'applique à tout mot différent de 0) et on obtient alors $wt(c - 1)=2^{r-1}$ (pour $c \neq 1$). On obtient alors que $wt(c)=2^{r-1}$ simplement en changeant les 1 et les 0 dans $c - 1$.□

On va donner maintenant une définition générale d'un code de Reed-Muller d'ordre $m$.

**Définition 4.3.4.** Le code de *Reed-Muller* d'ordre $m$ et de longueur $n=2^r$ où $0 \leq m \leq r$ , noté $RM(m, r)$, est le code linéaire $RM(2^r , \sum_{i=0}^{i=m} C_r^i, 2^{r-m})$ défini récursivement par:

$RM(0, r)= \{00\ldots0, 11\ldots1\}$, $RM(r, r) = F_2^n$,

$RM(m, r)= \{(x, x+y) / x \in RM(m, r\text{-}1), y \in RM(m\text{-}1, r\text{-}1)\}$, $0 < m < r$.

Notons par: $G(m, r)$ la matrice génératrice du code $RM(m, r)$, défini récursivement par:

$G(m, r)=\begin{bmatrix} G(m,r-1) & G(m,r-1) \\ 0 & G(m-1,r-1) \end{bmatrix}$ avec, $G(0, r)= [11\ldots1]$ et $G(r, r)=\begin{bmatrix} G(r-1,r) \\ 0\ldots01 \end{bmatrix}$

**Exemple 4.3.4.** Construisons le code de *Reed-Muller* $RM(1, 3)$. Soit $G(1, 3)$ sa matrice génératrice.

$G(1, 3)=\begin{bmatrix} G(1,2) & G(1,2) \\ 0 & G(0,2) \end{bmatrix}$ avec, $G(1, 2)=\begin{bmatrix} G(1,1) & G(1,1) \\ 0 & G(0,1) \end{bmatrix}$, $G(1, 1)=\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $G(0, 2)=[1\ 1\ 1\ 1]$.

En remplaçant dans $G(1, 3)$ on trouve $G(1, 3)=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$. Qui est bien la matrice $G$ de l'exemple précédent (à équivalence prés). Le code $RM(1, 3)$ est le suivant;

$RM(1,3)=\{00000000, 11111111, 00011110, 01100110, 10101010, 11100001, 10011001, 01010101, 01111000, 10110100, 11001100, 10000111, 01001011, 00110011, 11010010, 00101101\}$

**Exercice 1.** Construire le code de *Reed-Muller* $RM(2, 3)$ de matrice génératrice $G(2, 3)$.

### 4.3.3 Les q-aires codes de Hamming

Considérons le corps fini $K=F_q$ de caractéristique un entier premier $p$ et de cardinal $q=p^r$. Les q-aires codes de *Hamming* sont une généralisation des codes de *Hamming* binaires ($q$=2). Soit la relation R définit sur $K^m$-{0} par :

$x, y \in K^m$-{0}: $x$ R $y$ si, et seulement si, $\exists\ \lambda \in K$-{0}: $y = \lambda x$. R ainsi définit est une relation d'équivalence dans $K^m$-{0} et la classe de $x \in K^m$-{0} est donné par: $\bar{x}=\{\lambda x / \exists\ \lambda \in K\text{-}\{0\}\}$. On remarque que chaque classe $\bar{x}$ est en bijection avec $K$-{0} donc card($\bar{x}$)=card($K$-{0})=$q$-1. Le nombre de classes d'équivalences est: $n= \frac{card(K^m-\{0\})}{card(K-\{0\})} = \frac{q^m-1}{q-1}$.

On considère la matrice $H$ à m lignes et n colonnes, où les colonnes de $H$ sont les représentants de chaque classe d'équivalence qui sont les éléments de l'ensemble quotient $K^m$-{0}/R. Le rang de $H$ est rg($H$) = $m$. En effet considérons la base canonique de $K^m$ :

$e_1 = (1, 0, \ldots, 0)$, $e_2 = (0, 1, \ldots, 0)$, …, $e_m = (0, 0, \ldots, 1)$, qui sont des colonnes de H dont le déterminant est 1 non nul donc rg($H$) = $m$.

**Définition 4.3.5.** Le code C de longueur $n= \frac{q^m-1}{q-1}$ de matrice de contrôle la matrice $H$ défini ci-dessus est dit *q-aire code* de *Hamming*.

**Proposition 4.3.1.** Soit C le *q-aire code de Hamming* sur le corps de Galois $K=F_q$ / $q=p^r$, Alors C est de longueur $n=\frac{q^m-1}{q-1}$, de dimension $k=n-m$ et de distance minimale $d=3$.

**Preuve.** En effet. Pour la longueur et la définition on les déduit de la définition de la matrice de contrôle *H*. Montrons que $d=3$. On sait que si *x* est un mot du code C défini par la matrice de contrôle *H*, alors :

$$x \cdot H^T = 0 .$$

Supposons que le poids minimum est strictement inférieur à 3, prenons le cas où il serait égal à 2.

Il existerait alors un mot $x = (x_0, x_1, ..., x_{n-1})$ de poids égal à 2, c'est-à-dire que les $x_i$ seraient tous nuls sauf aux deux positions *i* et *j* où on aurait $x_i = \alpha,\ x_j = \beta$ avec $\alpha, \beta \in K-\{0\}$.

On aurait alors : $x \cdot H^T = 0 \quad \Leftrightarrow \quad \alpha \cdot C_i + \beta \cdot C_j = 0 \Leftrightarrow \quad C_i = -\frac{\beta}{\alpha} \cdot C_j$

où $C_i$ et $C_j$ sont les colonnes de $H^T$ se situant aux positions *i* et *j* et qui sont linéairement indépendants et donc appartiennent à la même classe d'équivalence et ceci remet en cause la définition de la matrice, car celle-ci a été construite de telle sorte que ses colonnes soient les représentants des classes d'équivalence. Il y a donc une contradiction.

On en déduit que le poids minimum d'un mot doit être de 3, et donc la distance $d = 3$.

**Proposition 4.3.2.** Le *q-aire code de Hamming* sur le corps $K=F_q$, est un code parfait.

**Preuve.** On rappelle que la dimension du code C est $k=n-m$ et la longueur *n*, il suffit de montrer que ce code atteint la borne de Hamming c.à.d.

$$\sum_{i=0}^{e} \binom{n}{i} (q-1)^i = q^{n-k} = q^m \quad avec\ e = \left\lfloor \frac{d-1}{2} \right\rfloor = 1\ et\ k = n-m .$$

On a:

$$\sum_{i=0}^{e} \binom{n}{i} (q-1)^i = \sum_{i=0}^{1} \binom{n}{i} (q-1)^i = \left[1 + n(q-1)\right] = 1 + n(q-1)$$

$$= 1 + \frac{q^m - 1}{q-1} q - 1 = q^m$$

**Exemple 4.3.5.** Si $q=3$, le corps $K=F_3$. On obtient le code de *Hamming trinaire ou ternaire* de longueur $n=(3^m-1)/2$ et de dimension $k= (3^m-1)/2-m$. Pour $m=2$. La longueur est $n=4$, la dimension $k=2$. Construirons C ( donc la matrice *H*)

$K^2-\{(0, 0)\}=\{ (1, 0), (0, 1), (2, 0), (0, 2), (1, 2), (2, 1), (1, 1), (2, 2)\}$ les classes d'équivalences sont: $\overline{(x, y)}=\{\lambda(x, y) / \lambda \in \{1, 2\}\}=\{(x, y), (2x, 2y)$ tel que $(x, y) \neq (0, 0)\}$

$\overline{(1, 0)}=\{(1, 0), (2, 0)\}$, $\overline{(0, 1)}=\{(0, 1), (0, 2)\}$, $\overline{(1, 1)}=\{(1, 1), (2, 2)\}$, $\overline{(1, 2)}=\{(1, 2), (2, 1)\}$.

Les représentants des classes d'équivalences sont respectivement (1, 0), (0, 1), (1, 1) et (1, 2)

Le code ternaire de Hamming admet comme matrice de contrôle la matrice $H=\begin{pmatrix}1 & 0 & 1 & 1\\ 0 & 1 & 1 & 2\end{pmatrix}$

**Exemple 4.3.6.** Si $q$=3, construire le code de *Hamming ternaire* pour $m$=3.

Les classes d'équivalences sont:

$\overline{(x, y, z)}=\{\lambda(x, y, z) / \lambda\in\{1, 2\}\}=\{(x, y, z), (2x, 2y, 2z)$ tel que $(x, y, z)\neq(0, 0, 0)\}$

Le *code ternaire de Hamming* admet comme matrice de contrôle

$H=\begin{pmatrix}1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 2 & 1 & 1\\ 0 & 1 & 0 & 1 & 2 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 1\\ 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 2 & 1 & 1 & 1 & 2\end{pmatrix}$ et comme distance $d$=3.

## 4.4 Décodage des codes linéaires.

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

#### 4.4.1.1 Classes latérales.

**Définition 4.4.1.** Soient C($n, k, d$) un code linéaire sur un corps fini $K$ de cardinal $q$.

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

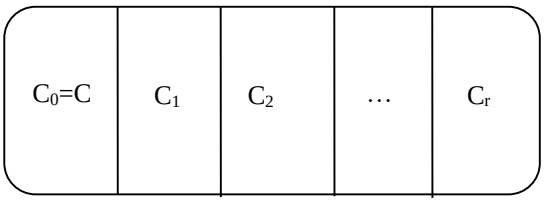


Figure 4.0 Classes latérales.

**Proposition 4.4.1.** Il y $q^{n-k}$ classes latérales de $K^n$ dont chaque classe contient $q^k$ mots.

**Preuve**. Soit $r$ le nombre des classes latérales. Montrons que $r= q^{n-k}$. Toute classe latérale $C_i$ tel que $0\leq i \leq r$-1 est en bijection avec le code C par l'application f de C dans $C_i$ définie par: pour tout c$\in$C: $f$(c) = c+$u_i$ tel que $u_i$ est le représentant de la classe $C_i$ et donc toutes les classes latérales ont le même cardinal $q^k$ et comme ces classes forment une partition de $K^n$, alors le nombre de ces classes est $r$ = card($K^n$)/card(C)= $q^n/q^k= q^{n-k}$.

**Définition 4.4.2.** On appelle tableau standard d'un code linéaire C($n, k, d$) sur un corps fini $K$ de cardinal q, le tableau ci-dessus 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 4.1 Forme d'un tableau standard.

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

#### 4.4.1.2 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 poind minimal et ainsi de suite jusqu'à épuisement de tous les mots de $K^n$. On pose: $s=q^k-1$ et $r= q^{n-k}-1$.

| Mots erreurs (Chefs de classes). | Classes latérales. |
|---|---|
| $u_0 = 00…00 = 0$ | $c_0=0$ $c_1$ $c_2$ . . . $c_S$ |
| $u_1 = r_1 0…00$ | $u_1$ $u_{1+} c_1$ $u_{1+}c_2$ . . . $u_{1+}c_S$ |
| $u_2 = 0\ r_2…00$ | $u_2$ $u_{2+} c_1$ $u_{2+}c_2$ . . . $u_{2+}c_S$ |
| … | … |
| $u_r = r_1\ r_2..\ r_e..00$ | $u_r$ $u_{r+} c_1$ $u_{r+}c_2$ . . . $u_{r+}c_S$ |

Tableau 4.2 Tableau standard.

**Remarque 4.4.1.** Pour un code *e*-correcteur binaire $K=F_2$. On a le tableau suivant:

Posons $s=2^k-1$ et $r= 2^{n-k}-1$.

| Nb erreurs | Mots erreurs. | Classes latérales. |
|---|---|---|
| | 100….00 | $u_1$ $u_{1+} c_1$ $u_{1+} c_2$ . . . $u_{1+}c_S$ |

| | | |
|---|---|---|
| 1 | 010…00<br>…<br>000….01 | $v_1$ $v_{1+}c_1$ $v_{1+}c_2$ . . . $v_{1+}c_S$<br>…<br>$w_1$ $w_{1+}c_1$ $w_{1+}c_2$ . . . $w_{1+}c_S$ |
| 2 | 110…00<br>101…00<br>…<br>000…11 | $u_2$ $u_{2+}c_1$ $u_{2+}c_2$ . . . $u_{2+}c_S$<br>$v_2$ $v_{2+}c_1$ $v_{2+}c_2$ . . . $v_{2+}c_S$<br>…<br>$w_2$ $w_{2+}c_1$ $w_{2+}c_2$ . . . $w_{2+}c_S$ |
| … | … | … |
| E | 111..1..00<br>011..1..00<br>…<br>000..1..11 | $u_e$ $u_{e+}c_1$ $u_{e+}c_2$ . . . $u_{e+}c_S$<br>$v_e$ $v_{e+}c_1$ $v_{e+}c_2$ . . . $v_{e+}c_S$<br>…<br>$w_e$ $w_{e+}c_1$ $w_{e+}c_2$ . . . $w_{e+}c_S$ |

Tableau 4.3 Tableau standard binaire ($K=F_2$).

#### 4.4.1.3 Principe de décodage par tableau standard.

Soient $y \in K^n$ le mot reçu , $c \in C$ le mot envoyé qu'on cherche et $\varepsilon \in K^n$ 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 erreurs et classes latérales.

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

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

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 les 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 standards assure un décodage au plus proche voisin c'est ce qu'on appelle "*principe de maximum de vraisemblance voisin*".

**Exemple 4.4.1.** Soit le code binaire C(5, 2) de matrice contrôle $H=\begin{pmatrix}1&0&1&0&0\\1&1&0&1&0\\0&1&0&0&1\end{pmatrix}$ et 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 4.4 Tableau de déchiffrement.

1. Si le mot reçu $y_1$=11101, c'est un mot de la premiere ligne, donc il n'y a pas d'erreurs et $x$=$y$=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 $e$.

#### 4.4.1.4 Inconvénients de la méthode du tableau standard.

La méthode de décodage par tableau standard présente plusieurs inconvénients:

1. Le tableau standard est long à construire si $n$ est grand, $n \geq 30$.
2. La recherche du mot reçu $y$ est trop lente quand $n$ est grand, $n \geq 30$.
3. L'algorithme de décodage occupe un espace mémoire considérable.

### 4.4.2 Décodage des codes linéaires par syndrome.

#### 4.4.2.1 Application syndrome.

**Définition 4.4.3.**Soient C($n, k, d$) un code linéaire sur un corps fini $K$ 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 $K^n$ dans $K^{n-k}$ dont $H$ est sa matrice dans les bases canoniques de $K^n$ et $K^{n-k}$.

$h$: $K^n \to K^{n-k}$, $x \mapsto h(x) = x.{}^tH$. Le mot $h(x) \in K^{n-k}$ est appelé *syndrome* de $x \in K^n$.

**Exemple 4.4.2.** 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: $F_2{}^5 \to F_2{}^3$,

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

| Les mots de $F_2{}^5$ | | | | 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 4.5 Syndromes des mots de $F_2{}^5$.

**Prposition 4.4.2.** Soient C($n, k, d$) un code linéaire sur un corps fini $K$ de cardinal $q$, $h$ l'application de $K^n$ dans $K^{n-k}$. alors $Kerh$=C et $K^n$ /C est isomorphe à $Imh$.

**Preuve.** $Kerh=\{x\in K^n: h(x)=0\}=\{x\in K^n: x.{}^tH=0\}=\{x\in K^n: x\in C\}=C$ et d'après le premier théorème d'isomorphisme d'éspaces vectoriels on a: $K^n/Kerh$ est isomorphe à $Imh$ ce qui est équivalent à $K^n/C$ est isomorphe à $Imh$, tel que $Imh=\{h(x)/x\in K^n\}$ ensembles des syndrome des éléments de $K^n$.

**Remarque 4.4.2.**

- Le quotient $K^n/C$ 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 4.4.3.** Soit C($n$, $k$, $d$) un code linéaire e-correcteur sur un corps fini $K$ et $y_1, y_2\in K^n$
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 est distance minimale alors $y_1-y_2=0$ d'où $y_1=y_2$.

### 4.4.2.2 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.

| Mots erreurs (Chefs de classe). | Classes latérale. | Syndrome |
|---|---|---|
| $u_0=00\ldots00=0$ | $c_0=0$ $c_1$ $c_2$ . . . $c_S$ | $h(0)$ |
| $u_1=r_10\ldots00$ | $u_1$ $u_{1+}c_1$ $u_{1+}c_2$ . . . $u_{1+}c_S$ | $h(u_1)$ |
| $u_2=0\ r_2\ldots00$ | $u_2$ $u_{2+}c_1$ $u_{2+}c_2$ . . . $u_{2+}c_S$ | $h(u_2)$ |
| . . | . . | .. |
| $u_r=r_1\ r_2..\ r_e00$ | $u_r$ $u_{r+}c_1$ $u_{r+}c_2$ . . . $u_{r+}c_S$ | $h(u_r)$ |

Tableau 4.6 Tableau de déchiffrement par syndrome.

**Remarque 4.4.3.** 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 est le suivant:

| 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 | $u_2$ $u_{2+}c_1$ $u_{2+}c_2$ . . . $u_{2+}c_S$<br>$v_2$ $v_{2+}c_1$ $v_{2+}c_2$ . . . $v_{2+}c_S$<br>…<br>$w_2$ $w_{2+}c_1$ $w_{2+}c_2$ . . . $w_{2+}c_S$ | $h(u_2)= C_{1+}C_2$<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 | $u_e$ $u_{e+}c_1$ $u_{e+}c_2$ . . . $u_{e+}c_S$<br>$v_e$ $v_{e+}c_1$ $v_{e+}c_2$ . . . $v_{e+}c_S$<br>…<br>$w_e$ $w_{e+}c_1$ $w_{e+}c_2$ . . . $w_{e+}c_S$ | $h(u_e)$<br>$h(v_e)$<br>…<br>$h(w_e)$ | Somme e à e des colonnes de la matrice H. |

Tableau 4.7 Tableau de déchiffrement binaire par syndrome.

Le tableau 4.7 peut être réduit au tableau suivant:

| Nb erreurs | Mots erreurs. | Syndromes |
|---|---|---|
| 0 | 000…00 | $h(0)= 0$ |
| 1 | 100….00<br>010…00<br>…<br>000….01 | Les colonnes de la matrice H. |
| 2 | 110…00<br>101…00 | Somme 2 à 2 des colonnes de la matrice H. |

| | | |
|---|---|---|
| | …<br>000…11 | |
| I | 1..10…00<br>01..1…00<br>…<br>00…1..1 | Somme i à i des colonnes de la matrice H. |
| e | 111..1..00<br>011..1..00<br>…<br>000..1..11 | Somme e à e des colonnes de la matrice H. |

Tableau 4.8 Tableau de déchiffrement binaire par syndrome réduit.

1. Si le code C est un code *e*-correcteur *non binaire* ($K \neq F_2$), alors les mots erreurs sont tous les mots erreurs dans le cas binaire (Tableau ci-dessus) multipliés par les scalaires non nuls du corps K et pour les syndromes, l'expression "*Somme i à i des colonnes de la matrice H*" est remplacé par "*combinaison linéaires de i colonnes de H*"
2. Si le nombre d'erreurs du mot reçu y dépasse *e*, il y a en général plusieurs cas possibles (selon le nombre d'erreurs) dans la ligne du syndrome de *y*.

#### 4.4.2.3 Principe du décodage par syndrome.

Soient $y \in K^n$ le mot reçu , $c \in C$ le mot envoyé qu'on cherche et $\varepsilon \in K^n$ 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. $h(y)=y.{}^{t}H$.
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 $\varepsilon$ de poids $w(\varepsilon) \leq e$.
6. Calculer le mot envoyé $x=y-\varepsilon$.

**Exemple 4.4.3.** 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 4.9 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 4.4.4.** Soit le code ternaire C(5, 2) de matrice contrôle $H=\begin{pmatrix}2&0&1&0&0\\1&2&0&1&0\\0&1&0&0&1\end{pmatrix}$ et

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 | $2C_1=120$ | | 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 4.10 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 $h(y_1)=C_2+ C_4+C_5=002=2C_5=h$(00002). Le mot erreur est donc $\varepsilon_1$=00002 et le mot envoyé est $x_1= y_1-\varepsilon=y+2\varepsilon_1=01011+00001=01012$.

**Exemple 4.4.5.** 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

Et de matrice génératrice normalisée $G_N=(\ I_{10}\ /\ M)$ 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:

$\varepsilon$=0 0 2 0 0 0 0 0 0 0 0 0 0 et le mot envoyé est: $x = y-\varepsilon = y+2\varepsilon$ = 0 1 0 0 0 0 0 0 0 0 12.

### 4.4.3 Codage et Décodage des codes de Hamming.

#### 4.4.3.1 Codage de Hamming (7,4)

Il permet de **détecter et corriger une erreur** sur un bit dans un bloc de 7 bits.

- **4 bits** sont des bits de données ($d_1$ $d_2$ $d_3$ $d_4$).
- **3 bits** sont des bits de parité ($p_1$ $p_2$ $p_3$), calculés à partir des données.

Les bits de parité ($p_1$ $p_2$ $p_3$) sont placés aux positions puissances de 2 (1, 2, 4) c.à.d. $p_1$ en position 1, $p_2$ en position 2 et $p_3$ en position 4 et $d_1$ $d_2$ $d_3$ $d_4$ en position 3, 5, 6, 7 respectivement.

Le codage du mot **$d = d_1\ d_2\ d_3\ d_4$** sera le mot ccode **$c=p_1\ p_2\ d_1\ p_3\ d_2\ d_3\ d_4$** tel que $p_1$, $p_2$, $p_3$ sont calculer comme suit :

$p_1$ se calcule tel que la parité des bits en position 1,3,5,7 soit paire c.a.d. $p_1=d_1+d_2+d_4$.

$p_2$ se calcule tel que la parité des bits en position 2,3,6,7 soit paire c.a.d. $p_2=d_1+d_3+d_4$.

$p_3$ se calcule tel que la parité des bits en position 1,3,5,7 soit paire c.a.d. $p_3=d_2+d_3+d_4$.

En résumé est comme suit :

Le codage du mot **$d = d_1\ d_2\ d_3\ d_4$** est le mot code **$c= d_1+d_2+d_4\ d_1+d_3+d_4\ d_1\ d_2+d_3+d_4\ d_2\ d_3\ d_4$.**

## Exemple

**Codage du message :**
Supposons que nos **4 bits de données** soient : d=1101.
On les place aux positions 3, 5, 6, 7 du vecteur de 7 bits.

```
Position : 1    2    3    4    5    6    7
Vecteur  : p1   p2   d1   p4   d2   d3   d4
           p1   p2   1    p4   1    1    1
```

On calcule les bits de parité (ici, parité paire) :

- $p_1+d_1+d_2+d_4 = p_1+1+1+1=0$, alors $p_1$ doit etre egale à 1
- $p_2+d_1+d_3+d_4 = p_2+1+0+1=0$, alors $p_2$ doit etre egale à 0.
- $p_3+d_2+d_3+d_4 = = p_3+1+0+1=0$, alors $p_2$ doit etre egale à 0.

Mot de code code (envoyé) est **c=1010101.**

## Algorithme de Décodage (Correction d'erreur)

Lorsqu'on reçoit un mot de 7 bits (r1,r2,r3,r4,r5,r6,r7 ), on suit ces étapes :

1. **Calcul des bits de syndrome (S1,S2,S3) :**
   On recalcule les contrôles de parité en fonction des bits reçus. Chaque syndrome vérifie un ensemble spécifique de bits.
   - S1 : Parité des bits de position **1, 3, 5, 7** (tous les bits où le 1er bit du code binaire de la position est à 1).
   - S2 : Parité des bits de position **2, 3, 6, 7**.
   - S3 : Parité des bits de position **4, 5, 6, 7**.

   Si le mot reçu est correct, tous les syndromes valent 0.

2. **Interprétation du syndrome :**
   Les trois bits de syndrome forment un **nombre binaire S3S2S1**.
   - **Si ce nombre = 0** : Aucune erreur détectée.
   - **Sinon** : La valeur de ce nombre donne **exactement la position du bit erroné** dans le mot de 7 bits.

3. **Correction de l'erreur :**
   On inverse (on "bascule") le bit à la position indiquée par le syndrome.
4. **Extraction des données :**
   Après correction, on extrait les bits de données (aux positions 3, 5, 6, 7) pour retrouver le message original de 4 bits.

---

## Exemple Complet

**1. Encodage du message :**
Supposons que nos **4 bits de données** soient : d=1101.
On les place aux positions 3, 5, 6, 7 du vecteur de 7 bits.

text
```
Position : 1    2    3    4    5    6    7
Vecteur  : ?    ?    1    ?    1    0    1
           p1   p2   d1   p4   d2   d3   d4
```

On calcule les bits de parité (ici, parité paire) :

- p1 (pos 1) : vérifie pos 1,3,5,7 → (p1,1,1,1). Pour que la parité soit paire : `p1=1` (car 1+1+1+p1 doit être pair).
- p2 (pos 2) : vérifie pos 2,3,6,7 → (p2,1,0,1). Parité paire → `p2=0`.
- p4 (pos 4) : vérifie pos 4,5,6,7 → (p4,1,0,1). Parité paire → `p4=0`.

**Mot de code envoyé :** `1 0 1 0 1 0 1`

**2. Transmission et erreur :**
Supposons qu'une erreur se produise sur le **6ème bit** pendant la transmission.
**Mot reçu :** `1 0 1 0 1 **1** 1` (le bit en position 6 est passé de 0 à 1).

---

**3. Décodage et correction :**

- **Calcul des syndromes :**
  - S1 = parité des bits (1,3,5,7) = (1,1,1,1) → $1 \oplus 1 \oplus 1 \oplus 1 = 0$
  - S2 = parité des bits (2,3,6,7) = (0,1,1,1) → $0 \oplus 1 \oplus 1 \oplus 1 = 1$
  - S3 = parité des bits (4,5,6,7) = (0,1,1,1) → $0 \oplus 1 \oplus 1 \oplus 1 = 1$
- **Interprétation :**
  Le syndrome est S3S2S1=**110**S3S2S1=**110**.
  C'est le nombre binaire **6** (car 1∗4+1∗2+0∗1=61∗4+1∗2+0∗1=6).
- **Correction :**
  L'erreur est donc à la **position 6**. On inverse le bit en position 6 du mot reçu :
  `1 0 1 0 1 **1** 1` → `1 0 1 0 1 **0** 1`

**Extraction des données :**

On récupère les bits de données aux positions 3,5,6,7 → `1 1 0 1`.

C'est bien le message original `1101`.

## 4.5 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.

$$uG := \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_1 + u_3 + u_4 & u_1 + u_2 + u_3 & u_1 + u_2 + u_4 \end{bmatrix}$$

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.

$$x := [1, 1, 0, 0, 1, 0, 0]$$