# Chapitre 6: Codes détecteurs et correcteurs d'erreurs de transmission

# Introduction

- Un code correcteur est une technique de codage basée sur la redondance. Elle est destinée à corriger les erreurs de transmission d'une information (plus souvent appelée message) sur une voie de communication peu fiable.
- La théorie des codes correcteurs ne se limite pas qu'aux communications classiques (radio, câble coaxial, fibre optique, etc.) mais également aux supports pour le stockage comme les disques compacts, la mémoire RAM et d'autres applications où l'intégrité des données est importante.

# Introduction

- La problématique des codes correcteurs d'erreurs est la suivante :

  un expéditeur A envoie un message m à B ; durant la transmission de ce message, des erreurs se produisent éventuellement, et B reçoit un message m' qui comporte peut-être des erreurs. Il s'agit de trouver comment faire pour que B:

  1- Détecte l'existence d'erreurs,

  2- Si les erreurs ne sont pas trop nombreuses, savoir les corriger.

- Dans certains cas, lorsqu'il est rapide de réexpédier le message, 1) suffit. Néanmoins, dans d'autres cas, 2) s'avère indispensable, ex: *Transmission satellitaire*.

# Introduction

- Dans ce chapitre on essayera de voire l'essentiel sur les codes détecteurs correcteurs d'erreurs. Différents catégories de codes existes; on va étudier essentiellement :

1. Principe de la Représentation vectorielle
2. Codes linéaires
3. Codes de Hamming

# Partie I: Représentation vectorielle

# Codes et codages binaires en blocs

$$x \longrightarrow \boxed{\text{CODEUR}} \longrightarrow y$$

- $X = \{0,1\}^m \quad Y = \{0,1\}^n \quad C = f(x) \subseteq Y$ est un **code**
- Mot à coder $\quad x = x_1\, x_2 \ldots x_m \quad x \in X$
  - « m » est appelé dimension du code
- Mot du code $\quad y = y_1\, y_2 \ldots y_n \quad y \in Y$
  - « n » est appelé longueur du code $n > m$
- On défini la redondance par $\quad \boxed{R = \dfrac{n-m}{m}}$
- Et l'efficacité

**de détection** $\quad \boxed{E_d = \dfrac{probabilit\acute{e}\ \ d'un\ \ message\ \ d\acute{e}tect\acute{e}\ \ faux}{probabilit\acute{e}\ \ d'un\ \ message\ \ faux}}$

**de correction** $\quad \boxed{E_c = \dfrac{probabilit\acute{e}\ \ d'un\ \ message\ \ faux\ \ corrig\acute{e}}{probabilit\acute{e}\ \ d'un\ \ message\ \ faux}}$

# Codes et codages binaires en blocs

- Taxonomie des codes détecteurs et correcteurs:

**Codes détecteurs et correcteurs d'erreurs**

Codes de blocs

Codes convolutifs
(continus, récurrents)

linéaires

non linéaires

polynomiaux

non polynomiaux

cycliques

non cycliques

# Codes systématiques

- Définition

$$x = \boxed{x_1 x_2 \ldots x_m}$$
$$y = \boxed{y_1\, y_2 \ldots y_m}\boxed{y_{m+1} \cdots y_n}$$

$\forall\, i \in [1..m]$   $y_i = x_i$ :partie utile; $y_{m+1} \ldots y_n$ partie redondante

Intérêt :

- En absence d'erreur la partie utile est égale au mot émis.
- Seule la partie redondante est à calculer.

- Code de parité simple (n=m+1):

  - Parité paire $\quad y_n = \sum_{i=1}^{m} y_i \Rightarrow \sum_{i=1}^{n} y_i = 0$
  - Parité impaire $\quad y_n = \sum_{i=1}^{m} y_i + 1 \Rightarrow \sum_{i=1}^{n} y_i = 1$

L'addition est module 2.

# Codes systématiques

- Propriétés du code de parité simple :
  - Redondance $\boxed{R = \frac{1}{m}}$
  - Détection d'un nombre impair d'erreurs
  - Erreurs en nombre pair non détectables
  - Aucune correction d'erreur
  - Efficacités

$$\boxed{E_d = \frac{\sum_{m=1,3,5,\ldots}^{n} p^m (1-p)^{n-m} C_n^m}{1-(1-p)^n}} \qquad \boxed{E_c = 0}$$

# Codes systématiques

- Inconvénient du code de parité simple :

Pour tracer une erreur il faut calculer S :

Parité paire

$$S = \sum_{i=1}^{n} y_i$$

Parité impaire

$$S = \sum_{i=1}^{n} y_i + 1$$

$S \in \{0,1\}$

S'il n'ya pas d'erreurs: S=0 mais la réciproque est fausse

Si S=1 alors il y'a détection d'une erreur.

Le code ne permet de détecté qu'un nombre impaire des erreurs. Si le nombre d'erreurs est paire S=0 alors qu'il y'a des erreurs.

La valeur de S est appelé Syndrome.

# Codes systématiques

- Codes de parités croisées :

Ce sont des code systématiques de longueur de mots $m=p.q$ avec une redondance de $p+q/p.q$ $(n=p.q+p+q)$.

$$y=\begin{pmatrix} y_{11} & \cdots\cdots & y_{1q} \\ \cdots & \cdots\cdots & \cdots \\ y_{p1} & \cdots\cdots & y_{pq} \end{pmatrix}\begin{pmatrix} y_{1,q+1} \\ \cdots \\ y_{p,q+1} \end{pmatrix}$$
$$\begin{pmatrix} y_{p+1,1} & \cdots & y_{p+1,q} \end{pmatrix}\begin{pmatrix} \cancel{x_{p+1,q+1}} \end{pmatrix}$$

pour $i \leq p$, $j \leq q$

$y_{i,q+1}$ bit de parité des $y_{i,j}$

$y_{p+1,j}$ bit de parité des $y_{i,j}$

La valeur $(y_{p+1,q+1})$ est inutilisé

Les parités peuvent être paires ou impaires

# Codes systématiques

- **Propriétés des codes de parités croisées**

- La redondance est donnée par :

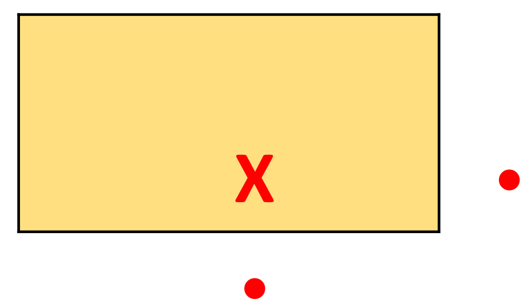
$$\boxed{R = \frac{p+q}{p \times q} = \frac{1}{p} + \frac{1}{q}}$$

- La détection d'une erreur implique la possibilité de corrigé cette erreur mais pas toujours.

- Le cas d'une seul erreur :

  Dans ce cas l'erreur est détectable est corrigeable.

  - syndrome =1 X erreur

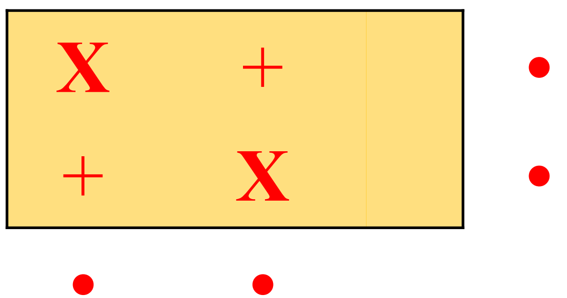


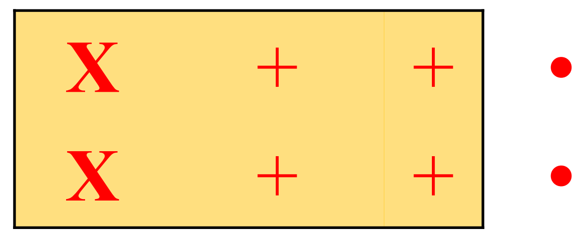
# Codes systématiques

- Le cas de deux erreurs :

Dans ce cas l’erreur est détectable mais non corrigeable (ambiguïté de position).

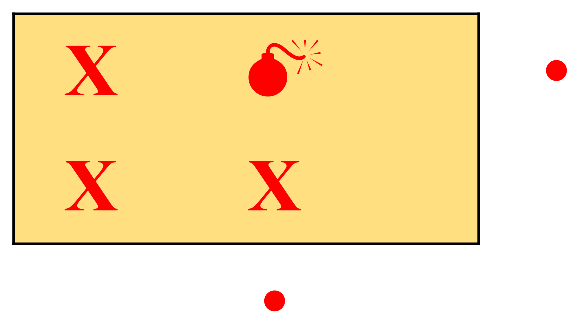
• syndrome =1 X erreur + autres possibilités





- Le cas de trois erreurs :

Deux syndromes non nulles donc la détection est possible. mais avec la possibilité d’une fausse correction.



# Codes systématiques

- Codes à répétition :

Chaque $x = x_1\ x_2 \ldots x_m$ est codé en

$$y = \underbrace{\boxed{y_{11} \cdots y_{1m}}\ \boxed{y_{21} \cdots y_{2m}} \cdots \boxed{y_{p1} \cdots y_{pm}}}_{x \text{ répété } p \text{ fois}}$$

avec $y_{ij} = x_j$.

- La redondance est donnée par : $R = p-1$
- Syndrome : $S = 0$ si est seulement si $\forall i,j,k \quad y_{ik} = y_{jk}$
- Correction possible si p impair: la correction est faite par vote majoritaire

- Inconvénient : redondance élevé (en fonction de p)

# Distances et erreurs

- Représentation géométrique :définitions

Soit $y_1, y_2, \ldots, y_n$ les coordonnées d'un vecteur (ou d'un point) y dans l'espace $\{0, 1\}^n$.

On définie un code C(n,m) par l'ensemble de $2^m$ vecteurs de dimension n dans $\{0, 1\}^n$.

Soit D la distance euclidienne entre 2 points x et y, on définie la distance de Hamming d par :

$$d(x,y) = D^2(x,y) = \sum_{i=1}^{n} (y_i - x_i)^2$$

La somme est définie sur N, et le résultat est dans N.

Exemple : x =1010 y =1100 d(x , y)=2

# Distances et erreurs

- On définie le poids d'un vecteur x par :

$$\boxed{poids\,(x) = \sum_{i=1}^{n} x_i}$$

La valeur du poids est entière .

- Théorème :

$$\boxed{d\,(x,y) = poids\,(x+y)}$$

Avec x + y est un vecteur.

Preuve

$$\boxed{d\,(x,y) = \sum_{i=1}^{n} (y_i - x_i)^2 = \sum_{i=1}^{n} (y_i - x_i) = \sum_{i=1}^{n} (y_i + x_i) = poids\,(x+y)}$$

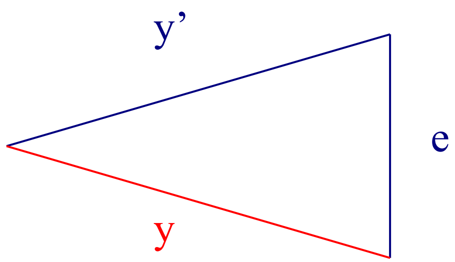
Exemple :

x = 1 0 1 0 ; y = 1 1 0 0 ; x + y = 0 1 1 0

d (x, y) = poids (x + y) = 2

# Distances et erreurs

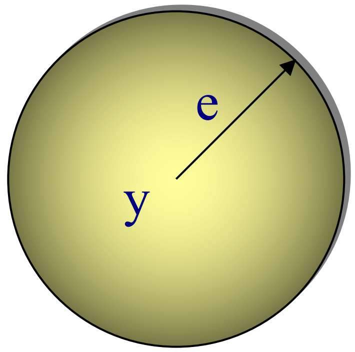
- Détection des erreurs :

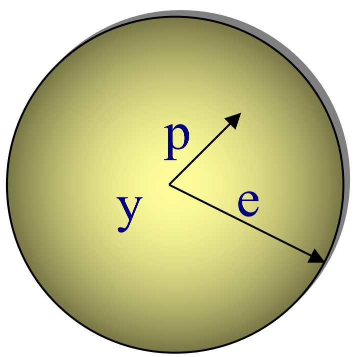

Soit y le vecteur émis et soit y′ le vecteur reçu.
L′erreur entre les deux e = y + y′ donc y′ = y + e et y = y′ + e. (somme mod 2)

d (y, y′) = poids (e) = nombre de bits « faux ». Chaque bit = 1 dans le vecteur e représente une erreur.

# Distances et erreurs

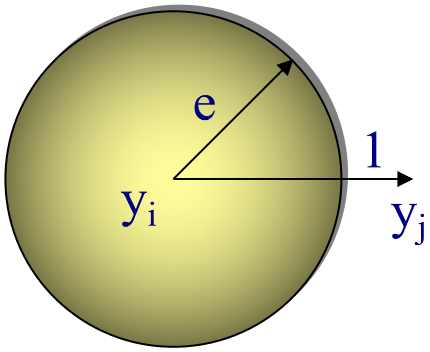



L'extrémité du vecteur d'une erreur
de poids e sur le symbole y émis
est située sur la surface de l'hypersphère
de rayon e et de centre y

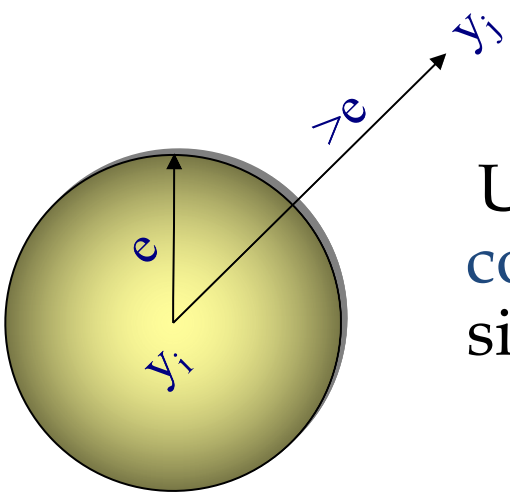



L'extrémité du vecteur d'une erreur
de poids $p < e$ sur le symbole y émis
est située à l'intérieur de l'hypersphère
de rayon e et de centre y

# Distances et erreurs




Une erreur de poids ≤ e sur $y_i$ est détectable s'il n'existe aucun mot du code $y_j$ situé à l'intérieur de l'hypersphère de rayon e et de centre $y_i$




Une erreur de poids ≤ e sur $y_i$ est corrigeable si $y_i$ est le seul mot du code situé à une distance d ≤ e

Le mot du code émis le plus probable est alors $y_i$

# Théorème fondamental

- **Définition : Distance d'un code**

  On appelle distance d'un code C, la valeur d définie par:
  $d(C) = Min(d(y_i, y_j)) \quad \forall\, y_i, y_j \in C\,,\ \ y_i \neq y_j$

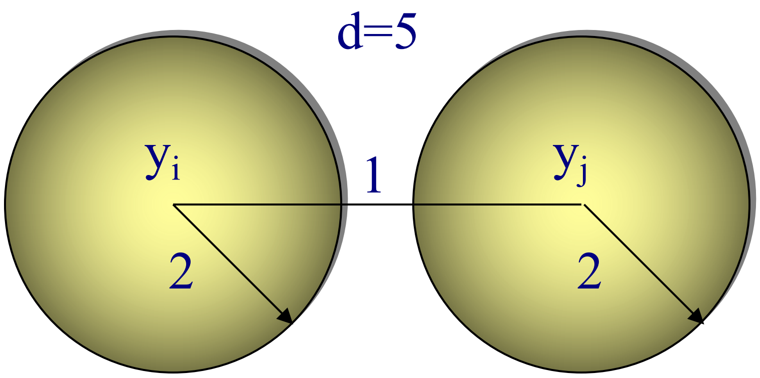
- **Théorème :**

Tout code C de distance d:

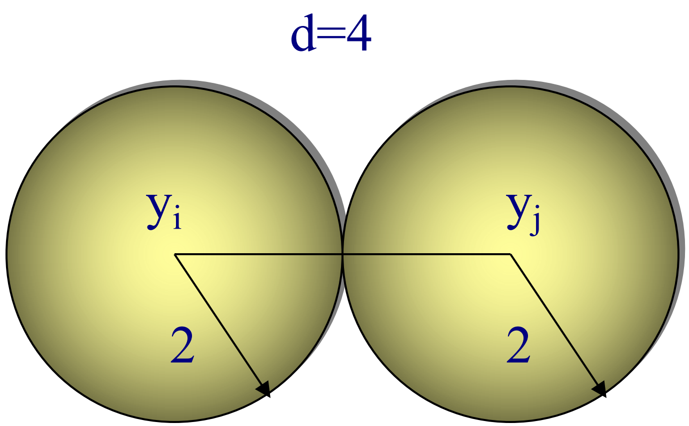
  - Détecte $p = d - 1$ erreurs : peut détecter au moins toutes les erreurs de poids $\leq p$
  - Corrige $q = int((d - 1) / 2)$ erreurs : peut corriger au moins toutes les erreurs de poids $\leq q$

# Théorème fondamental

- Illustration du théorème :




d = 5
4 erreurs détectables
2 erreurs corrigeables




d = 4
3 erreurs détectables
1 erreur corrigeable

# Construction de codes

- Problématique :

  Trouver un code de dimension m et de longueur n corrigeant toutes les erreurs de poids au plus égal à r , revient à placer $2^m$ points (les mots à codés) dans un espace de $2^n$ points, chaque point étant le centre d'une hypersphère de rayon r, ces sphères devant être disjointes.

  m, n et r sont donc dépendants

# Partie II: Codes Linéaires et codes de Hamming

# Généralisation des codes de parité (paire)

- Exemple:

  \- On construit un code C(7,4):m= 4 et n = 7 (on ajoute 3 bits de parité paire).

  \- Soit x = 1101 le message à envoyer. On ajoute un bit sur $x_1x_2x_4$, un bit sur $x_1x_3x_4$ et un bit sur $x_2x_3x_4$. On obtient comme code y = 1101 100.

  \- Soit l'erreur e = 0100 000 avec poids (e) = 1. Le mot reçu y' = 1001 100 avec y' = y+e d(y,y') =1.

  \- Les bits de parité sur $x_1x_2x_4$ et $x_2x_3x_4$ sont faux. Le bit de parité sur $x_1x_3x_4$ est juste.

  S'il n'y a qu'une erreur elle ne peut être qu'en $x_2$.

# Propriétés des codes de parité (paire)

- $y_j$ pour $m < j \leq n$ est un bit de parité paire pour certains bits $y_i$ pour $1 \leq i \leq m$ $\Leftrightarrow$ $y_j$ est une combinaison linéaire des $y_i$

$$y_j = \sum_{i=1}^{m} \lambda_i . y_i \ , \ \lambda_i \in \{0,1\}$$

- Les bits $y_i$ contrôlés par $y_j$ sont tels que $\lambda_i = 1$, la parité est selon la relation:

$$y_j + \sum_{i=1}^{m} \lambda_i y_i = 0$$

# Espaces vectoriels

- On considère les vecteurs y de codage appartenant à un espace vectoriel E=<$F_2$,$F_2^n$,.,0> avec $F_2$=<{0,1},+,.,0,1> un corps et $F_2^n$= <{0,1},+,0> un groupe.
- Une forme linéaire sur E est définie par :

$$\sum_{i=1}^{m} \lambda_i.Y_i \ , \ \lambda_i \in \{0,1\} \ Y_i \in Y$$

- On dit que {$Y_1$,…$Y_n$} est une base de E ssi:

$$\forall y \in Y \ , \exists \lambda_1,...\lambda_n : y = \sum_{i=1}^{n} \lambda_i.Y_i$$

- {$Y_1$,…$Y_n$} sont linéairement indépendants :

$$\sum_{i=1}^{n} \lambda_i.Y_i = 0 \Rightarrow \forall i, \lambda_i = 0$$

# Vecteurs binaires :propriétés

- Un vecteur binaire n'est pas orienté

  **Preuve**: $y + y = 0 \Rightarrow y = -y$

- Un vecteur binaire peut être orthogonal à lui-même:

  $y \perp y \Leftrightarrow y \,.\, y = 0$

  Exemple: y=1010 : $y.y = (1\ 0\ 1\ 0) \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = 0$

- **Théorème :**

  $y \perp y$ (y est orthogonal à y) $\Leftrightarrow$ poids (y) pair

  **Preuve :**

$$y.y = \sum_{i=1}^{n} (y_i . y_i) = \sum_{i=1}^{n} y_i = 0 \Rightarrow \text{Le nombre de 1 est paire}$$

$$\Rightarrow \text{Poid(y) est paire}$$

# Matrice génératrice de codage

- Le codage linéaire f : X → Y peut être représenté par une matrice dite génératrice:

$$(x_1,\ldots\ldots,x_m)\bullet\begin{array}{cc} & \begin{array}{cccccc} 1 & \ldots & \ldots & \ldots & \ldots & n \end{array} \\ \begin{array}{c} 1 \\ \vdots \\ \vdots \\ \vdots \\ m \end{array} & \begin{pmatrix} g_{11} & \ldots & \ldots & \ldots & g_{1n} \\ \vdots & \ddots & & & \vdots \\ \vdots & & g_{ij} & & \vdots \\ \vdots & & & \ddots & \vdots \\ g_{m1} & \ldots & \ldots & \ldots & g_{mn} \end{pmatrix} \end{array} = (y_1,\ldots\ldots,y_n)$$

$$X \quad \bullet \quad G \quad = \quad Y$$

$$\forall j, 1 \leq j \leq n : y_j = \sum_{i=1}^{m} x_i . g_{ij}$$

# Matrice génératrice de codage

- Les lignes de G sont des mots du code C images des vecteurs x de poids 1 (base canonique de X)
- L'application G doit être injective donc:

Les lignes de G doivent être linéairement indépendantes alors:

- L'espace vectoriel d'arrivée est un sous-espace vectoriel de dimension m.
- Les lignes de G constituent une base de C.

- G est diagonale-gauche car $\forall\ 1 \leq i \leq m : y_i = x_i$ (généralement unitaire-gauche).

$$\begin{array}{cc} & \begin{array}{ccc} 1 & \quad\quad m & \quad n \end{array} \\ \begin{array}{c} 1 \\ \vdots \\ \vdots \\ \vdots \\ m \end{array} & \left(\begin{array}{c|c} \quad I \quad & \quad A \quad \end{array}\right) \end{array}$$

A(m,n-m) est appelé matrice de contrôle.

# Matrice génératrice de codage

- Exemples :

Codes de parité paire

$m = 3 \;\; n = 4$

$$G = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right)$$

$y_1 = x_1 \;\; y_2 = x_2 \;\; y_3 = x_3 \quad y_4 = x_1 + x_2 + x_3$

Code à répétitions

$m = 1 \;\; n = 4$

$$G = \left(\begin{array}{c|ccc} 1 & 1 & 1 & 1 \end{array}\right)$$

$y_1 = y_2 = y_3 = y_4 = x_1$

# Systématisation d'un code linéaire

- À tout code linéaire C correspond au moins un code linéaire systématique équivalent.

Preuve : Algorithmique (soit G la matrice génératrice de C)

**Pour** i = 1 à m **faire**
{ **si** $g_{ii}$=0 *// $\exists m > i, g_{im} = 1$*
**alors** Permute($G_{.i}$ , $G_{.m}$ ) *//permuter les colonnes i et m*
**Pour** tout j≠i **faire**
{**si** $g_{ji}$ = 1
**alors** $G_j$:=$G_{j.}$+$G_{i.}$ *// ligne j = somme des lignes j et i*
}
}

Cet algorithme permet de transformé G en matrice de code systématique (unitaire-gauche)

# Systématisation d'un code linéaire

- Exemples :

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

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

$$\Rightarrow\begin{pmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{pmatrix}\Rightarrow\left(\begin{array}{c|ccc} & 0&1&1\\ I & 1&0&1\\ & 1&1&0\\ & 1&1&1\end{array}\right)$$

# Propriétés des codes linéaires

- C linéaire $\Rightarrow \{d(y_1,y_2): y_1,y_2 \in C\} = \{\text{poids } (y_3) : y_3 \in C\}$

  L'ensemble de tout les valeurs des distances entre les $y_i$ coïncide avec l'ensemble des valeurs des poids des vecteurs $y_i$.

Preuve :

$\forall\, y_1, y_2 \in C, \exists\, y_3 \in C : d(y_1, y_2) = \text{poids } (y_3)$

$d(y_1, y_2) = \text{poids } (y_1 + y_2)$ et $y_3 = y_1 + y_2 \in C$

$\Rightarrow \boldsymbol{\{d(y_1, y_2) : y_1, y_2 \in C\} \subseteq \{\text{poids } (y_3) : y_3 \in C\}}$

$\forall\, y_3 \in C\ \exists\, y_1, y_2 \in C$ poids $(y_3) = d(y_1, y_2)$

On pose $y_1 = y_3$ $y_2 = 0$ on obtient $d(y_1,y_2)=\text{poids}(y_3)$

$\Rightarrow \boldsymbol{\{d(y_1, y_2) : y_1, y_2 \in C\} \supseteq \{\text{poids } (y_3) : y_3 \in C\}}$

*C.Q.F.D.*

# Propriétés des codes linéaires

- Définition :

Le poids d'un code linéaire C est définie par :

$$\text{Poids}(C)=\text{Min}(\text{Poids}(x))\ \forall\, x \in C,\ \ x \neq 0$$

- Théorème :

$$C \text{ linéaire} \Rightarrow d(C) = \text{poids}\ (C)$$

Preuve

$\{d\ (y_1, y_2) : y_1, y_2 \in C\} = \{\text{poids}\ (y_3) : y_3 \in C\}$

$\{d\ (y_1, x_2) : y_1, y_2 \in C\} -\{0\} = \{\text{poids}\ (x_3) : x_3 \in C\} -\{0\}$

$\min\ (\{d\ (x_1, x_2) : x_1, x_2 \in C\} -\{0\}) = \min\ (\{\text{poids}\ (x_3) : x_3 \in C\} -\{0\})$

*C.Q.F.D.*

# Propriétés des codes linéaires

- Borne de Singleton :

$$C\ (n,m) \text{ linéaire} \Rightarrow d\ (C) \leq n - m + 1$$

Preuve

Soit C′ le code systématique équivalent à C de matrice G′.

Les lignes de G′ : $G'_{i.}$ sont des mots du code (base).

On sait que $d\ (C) = d\ (C') = \text{poids}\ (C') \leq \max\ (\text{poids}\ (G'_{i.}))$

Puisque G′ est diagonale gauche $\Rightarrow$ au moins m-1 zéros dans $G'_{i.} \Rightarrow \max\ (\text{poids}\ (G'_{i.})) = n - (m - 1) = n - m + 1$

Donc $d\ (C) \leq n - m + 1$.

*C.Q.F.D.*

# Théorème fondamental

- C linéaire $\Rightarrow$ une erreur de vecteur e est détectable si et seulement si $e \notin C$

Preuve

$y$ émis $\in C$  $y'$ reçu  $y' = y + e$
Le théorème est de type A si et seulement si B, pour le prouver, on démontre (1) $\neg B \Rightarrow \neg A$ et (2) $\neg A \Rightarrow \neg B$

(1) $e \in C \Rightarrow y' = y + e \in C \Rightarrow$ e non détectable
(2) e non détectable $\Rightarrow y' \in C \Rightarrow e = y + y' \in C$

# Matrice de vérification

- Pour chaque matrice génératrice G d'un code C(n,m), il existe une matrice H(n-m,n) (matrice de vérification ou de contrôle) qui vérifie :

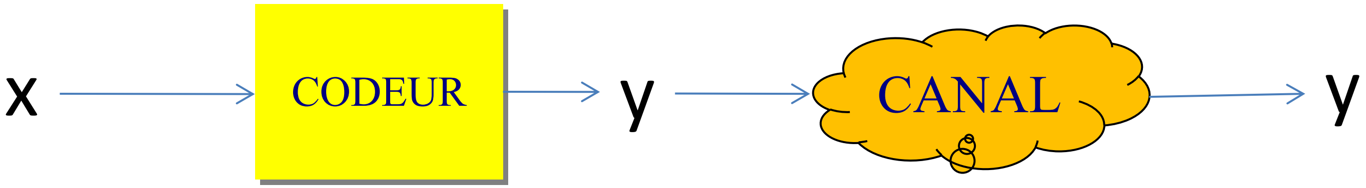
$$G \bullet H^t = 0 \quad \text{et} \quad H \bullet G^t = 0$$

- C'est-à-dire que :

$$\forall i, j \; 1 \leq i \leq m \text{ et } 1 \leq j \leq n-m \quad \sum_{k=1}^{n} G_{ik}.H^t_{kj} = 0$$

- Cette matrice est construite en se basent sur les dépendances linéaires entre les vecteurs colonnes de la matrice G. (les vecteurs colonnes de G sont linéairement dépendants, alors que les n-m vecteurs sont linéairement indépendants).

# Calcul des Syndromes




- Soit x le mot à envoyer et y le mot de code correspondant.
- y′ est le mot reçu avec y′=y+e (e :l′erreur de transmission)

Le syndrome de y′: S(y′)est définie par $H \bullet y'^t$ (S est linéaire).

$$H \bullet G^t = 0 \Rightarrow \forall x : \; H \bullet G^t \bullet {}^t x = 0$$

$$\Rightarrow \forall x : \; H \bullet {}^t(x \bullet G) = 0$$

$$\Rightarrow \forall y \;\; H \bullet {}^t y = 0$$

Donc S(y)=0 et S(y′)=S(e) car S(y′)=S(y+e)=S(y)+S(e)=S(e)

e=0 $\Rightarrow$ S(y′)=S(e)=0 (pas d′erreurs)

La réciproque est fausse (erreurs non détectés)

La matrice H est utilisé pour vérifié la transmission

# Matrice H d'un code Systématique

- Pour un code systématique équivalent C, la matrice G est de la forme :

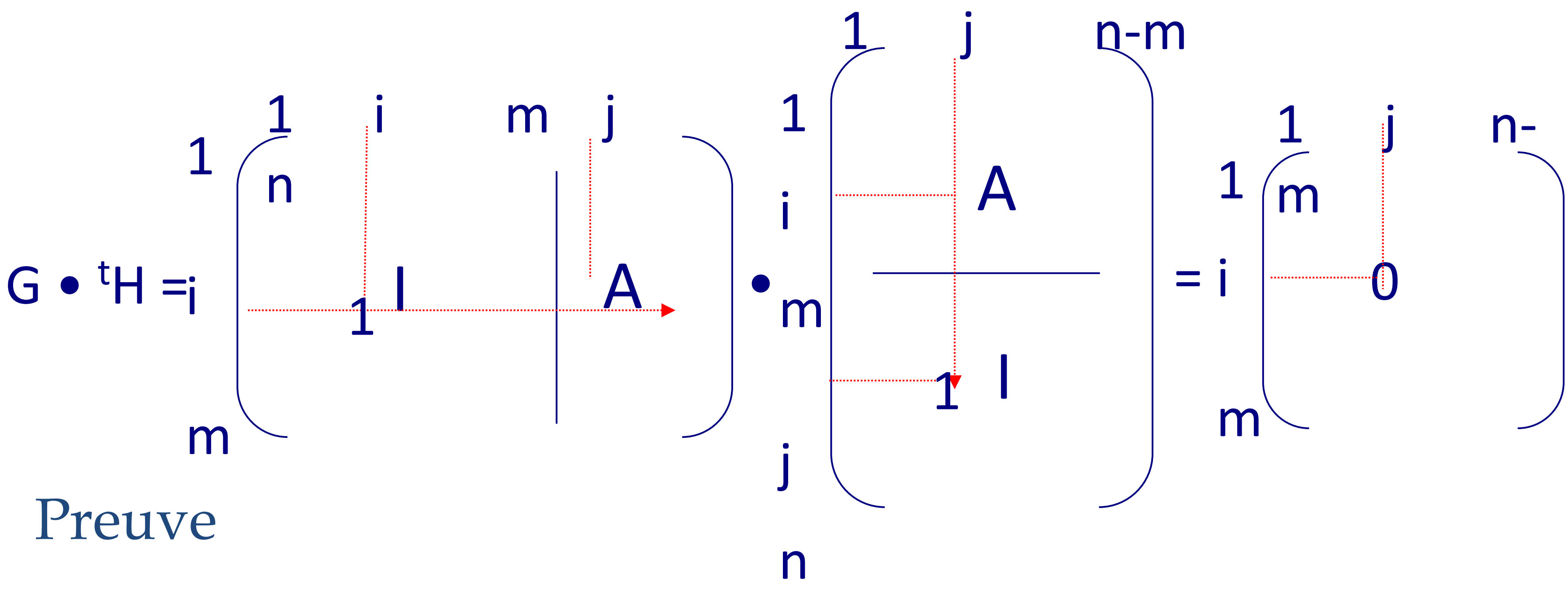
$$G = \begin{array}{cc} & \begin{array}{ccc} 1 & \quad m & \quad n \end{array} \\ \begin{array}{c} 1 \\ \\ m \end{array} & \left(\begin{array}{c|c} \\ I & A \\ \\ \end{array}\right) \end{array}$$

- Donc la matrice de contrôle H est de la forme

$$H = \begin{array}{cc} & \begin{array}{ccc} 1 & \quad m & \quad n \end{array} \\ \begin{array}{c} 1 \\ \\ n-m \end{array} & \left(\begin{array}{c|c} \\ {}^{t}A & I \\ \\ \end{array}\right) \end{array}$$

# Vérification d'un code systématique

- On prouve que cette forme de H vérifie bien $G \bullet {}^tH = 0$ :




- Preuve

$$\forall i, j \; (\mathrm{G} \bullet {}^{\mathrm{t}}\,\mathrm{H})_{\mathrm{ij}} = \sum_{\mathrm{k}=1}^{\mathrm{n}} \mathrm{G}_{\mathrm{ik}} . {}^{\mathrm{t}}\mathrm{H}_{\mathrm{kj}}$$

$$= \sum_{k=1}^{m} I_{ik} . A_{kj} + \sum_{k=m}^{n} A_{ik} . I_{kj} = A_{ij} + A_{ij} = 0$$

# Exemples de vérification de code linéaires

- Code de parité paire:

$$m = 3 \quad n = 4$$

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

$$y_1 = x_1 \quad y_2 = x_2 \quad y_3 = x_3 \quad y_4 = x_1 + x_2 + x_3$$

$$H = (1\ 1\ 1\ 1)$$

$$y_1 + y_2 + y_3 + y_4 = 0$$

- Code à répétition

$$m = 1 \quad n = 4$$

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

$$y_1 = y_2 = y_3 = y_4 = x_1$$

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

$$y_1 + y_2 = 0$$

$$y_1 + y_3 = 0$$

$$y_1 + y_4 = 0$$

# Construction des codes de Hamming

- Hypothèse poids (e) = 1; e = (0 …1…0) avec 1 en position j ; $y' = y + e$ $S(y') = S(e) = H \bullet {}^te$

$$\begin{array}{cc} & \begin{array}{ccc} 1 & j & n \end{array} \\ \begin{array}{c} 1 \\ i \\ n-m \end{array} & \begin{pmatrix} & & \\ & \circ & \\ & & \end{pmatrix} \end{array} \bullet \begin{array}{c} 1 \\ \\ j \\ \\ n \end{array} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{array}{c} 1 \\ i \\ n-m \end{array} \begin{pmatrix} \\ \circ \\ \\ \end{pmatrix} = H_j$$

$$H \bullet {}^te = H_j$$

Le syndrome d'une erreur dans la position j est égale à la $j^{ième}$ colonne de la matrice H.

# Construction des codes de Hamming

- Le syndrome d'une erreur de poids 1 dont le $j^{ième}$ bit= 1 est égal à la colonne j de la matrice H :$S(e) = H_{.j}$
- Pour corriger une erreur, la matrice H doit avoir toutes ses colonnes non nulles et distinctes.
- La position de l'erreur est alors la position du syndrome dans une et une seule des colonnes de H.
- Un code ayant cette propriété est appelé

  Code De Hamming
- La distance d'un tel code est égale à 3.

# Construction des codes de Hamming

- Exemple :

n=7 et m=4 : code C(7,4)

$$G = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right) \Rightarrow H = \left(\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right)$$

$$X = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix} \Rightarrow Y = X \bullet G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

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

$$\Downarrow$$

$$Y = X + E = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \Rightarrow s(Y) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \Rightarrow$$

L'erreur est dans le 4$^{\text{ième}}$ bit

# Codes optimaux de Hamming

- Colonnes de H (n-m,n) non nulles et distinctes $\Leftrightarrow n \leq 2^{n-m} - 1 \Leftrightarrow 2^{n-m} \geq n + 1 \Leftrightarrow n - m \geq \log_2 (n + 1)$

$$\begin{array}{cc} & \begin{array}{ccc} 1 & \qquad\qquad & n \end{array} \\ \begin{array}{c} 1 \\ \\ \\ n\text{-}m \end{array} & \begin{pmatrix} \quad & \qquad\qquad & \quad \\ & & \\ & & \\ & & \end{pmatrix} \end{array}$$

- Un code de Hamming est optimal si sa redondance R =(n-m)/m et minimale c'est-à-dire n-m= $(\log_2 (n + 1))$ donc R= $(\log_2 (n + 1))/m$.

- Théorème : Les codes optimaux de Hamming sont des codes parfaits.

Preuve :

n-m= $\log_2(n + 1) \Rightarrow (n + 1).2^m = 2^n \Rightarrow$ le code est parfait

# Vérification du second théorème de Shannon

- Codes de Hamming optimaux

| $n$ | $n$-$k$ | $k$ | $R$ |
|---|---|---|---|
| 3 | 2 | 1 | 2 |
| 7 | 3 | 4 | 3/4 |
| 15 | 4 | 11 | 4/11 |
| 31 | 5 | 26 | 5/26 |
| … | … | … | … |
| $\propto$ | | | 0 |

# Conclusion

- Dans ce chapitre, on à vu seulement quelque illustrations simples des codes correcteurs et détecteurs d'erreurs, qui reposent essentiellement sur l'algèbre des corps.
- Plusieurs autres codes existent qui sont actuellement utilisé en pratique avec une grande efficacité:
- Codes polynomiaux,
- Les codes cycliques de longueur impaire, BCH (Bose, Ray-Chaudhuri, Hocquenghem),
- Les codes cycliques non-linéaires : Code de ReedSolomon
- Les codes de Reed-Muller,
- Les turbos codes ...