

# Codes correcteur d'erreur

## Une Introduction

***Les transmissions et le stockage ne sont pas sûrs à 100%***

**Le bruit :**

poussière, électromagnétisme, chaleur, qualité des supports...

**Idée :**

Rajouter de la redondance à l'information  
pour **détecter/corriger** des erreurs

**Détection :** Ethernet, ...

**Correction :** stockage de données,  
communications interplanétaires,...

**Comment rajouter cette redondance ??**

# Applications des codes correcteurs

l'émergence des calculs, stockages et transmissions numériques a donné aux codes correcteurs d'erreurs une grande importance

*Beaucoup d'ordinateurs ont la capacité de corriger des erreurs.*

C'est moins cher de corriger des erreurs éventuelles plutôt que de construire des ordinateurs garantis sans erreurs

Les codes utilisés sont souvent des codes de **Hamming** pouvant corriger une erreur

*Stockage de hautes capacités*

On utilise aussi des codes pour stocker des données (sons, vidéo,...) car lorsque la densité d'information sur le disque augmente, le risque d'erreur augmente aussi.

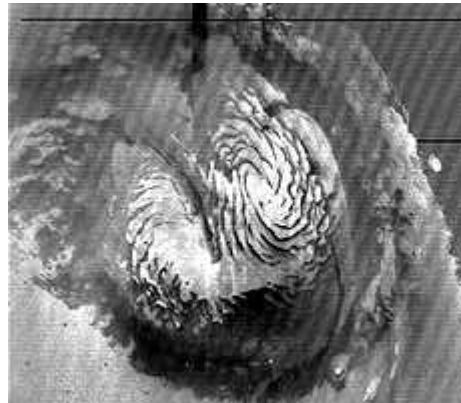
# ***1972 : la sonde Mariner transmet des images de Mars.***

**Le canal** est l'espace et l'atmosphère terrestre.

**Le bruit** vient de l'activité solaire et des conditions atmosphériques

**Le codage :**

64 dégradés de gris codés sur 6 bits  
l'encodage produit des mots de 32 bits  
le code de Reed et Muller est utilisé



# 1979 : La sonde Voyager transmet des images couleurs de Jupiter

4096 différents dégradés de couleurs

12 bits d'information

Les mots sont de longueur 24 bits

Le code utilisé : Le code de **Golay**

Il permet de corriger 3 erreurs et détecter 7 erreurs



## Audio numérique

L'information est stockée sur un film aluminium

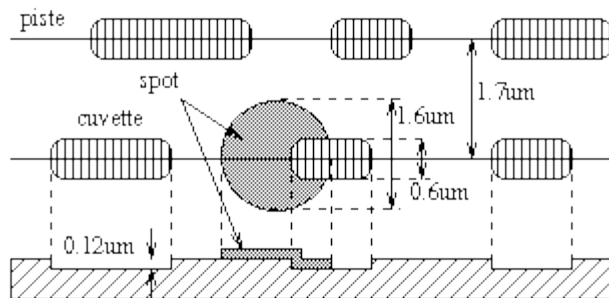
Des trous microscopiques représentent des 0 et des 1

Un faisceau laser permet de lire l'info

**Exemple** : les lecteurs CD de Philips et Sony utilisent des codes de **Reed-Solomon** entrelacés

# *Le disque laser :*

La lecture se fait à l'aide d'un spot laser



*Actuellement,  
toutes les communications sans fil  
utilisent des codes correcteurs d'erreurs*

# Concepts fondamentaux

Choisir un alphabet  $A$  de  $q$  symboles

**Exemples :**  $A = \{a,b,c,\dots,z\}$ ,  $A = \{0,1\}$  alphabet binaire.

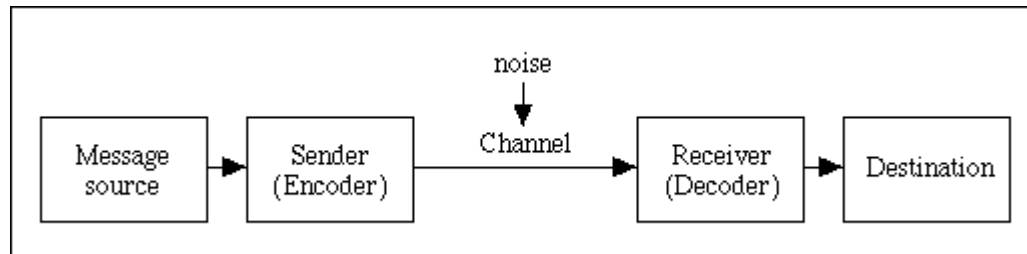
**Definition** Un code en block de longueur  $n$  contenant  $M$  mots Sur l'alphabet  $A$  est un ensemble de  $M$   $n$ -tuples

Ce code est appelé un  $(n,M)$ -code sur  $A$ .

En pratique, c'est souvent l'alphabet binaire qui est utilisé.

**Remarque :**

Le mot reçu par le récepteur n'est pas toujours le mot envoyé





## Exemple 1

L'information à transmettre est un des symboles {N, E, O, S}. Pour des raisons pratiques on revient au binaire :

N -> 00
E -> 01
O -> 10
S -> 11

On rajoute ensuite de la redondance

N -> 00 000
E -> 01 110
O -> 10 011
S -> 11 101

Et on construit un (5,4) code

N ->	00	000
E ->	01	110
O ->	10	011
S ->	11	101

Les mots de codes sont de longueur 5 et il existe 4 mots différents

L'**information** est codée sur  $k=2$  bits

La **redondance**  $r$  est de 3 bits pour obtenir des mots de longueur  $n=5$

**Définition** : la valeur  $R=k/n$  est appelée le taux du code

Dans notre exemple  $R=2/5$

**Définition:** La **distance de Hamming**  $d(x,y)$  entre deux mots  $x$  et  $y$  est le nombre de positions de coordonnées qui diffèrent entre  $x$  et  $y$

**Exemple** :  $A = \{0,1\}$ ,  $x=(10110)$   $y=(11011)$   $d(x,y) = 3$ .

Le **poids** de  $x$ ,  $w(x)$ , est le nombre de coordonnées non nulles dans  $x$

**Exemple1**  $x=(10110)$   $w(x)=3$

**Exemple2** : Soit  $C = \{c_0 = 00000, c_1 = 01110, c_2 = 10011, c_3 = 11101\}$

Le poids des mots est respectivement 0, 3, 3 et 4

$$d(c_1, c_2) = 4$$

$$d(c_1, c_3) = 3$$

$$d(c_2, c_3) = 3$$

Donc la distance la plus petite entre deux mot du code est 3

## Exemple dans un autre alphabet :

$A = (0,1,2)$ ,  $u = (21002)$ ,  $v = (12001)$  ;  $d(u,v) = 3$ .

**Définition** : Soit  $C$  un  $(n,M)$  code, la **distance de Hamming** du code est

$d = \min \{d(x,y) : x,y \text{ appartenant à } C, x \neq y\}$ .

C'est la distance minimale entre deux mots distincts du code

## Exemple

$C = \{c_0, c_1, c_2, c_3\}$

avec  $c_0=(000000)$   $c_1=(101101)$   $c_2=(011110)$   $c_3=(111010)$

Calculer la distance de Hamming  $d$  du code  $C$

**Remarque** : si le nombre de mots est grand,  
 $d$  est difficile à calculer

Soit un  $(n, M)$  code de distance  $d$   
Soit  $y$  le mot reçu. Comment décoder ??

## Trois cas possibles

Zéro erreur :  **$y$  est un mot du code**

Quelques erreurs : **corriger  $y$  en un mot du code**

Quelques erreurs : **correction impossible**

## Le décodeur peut se tromper

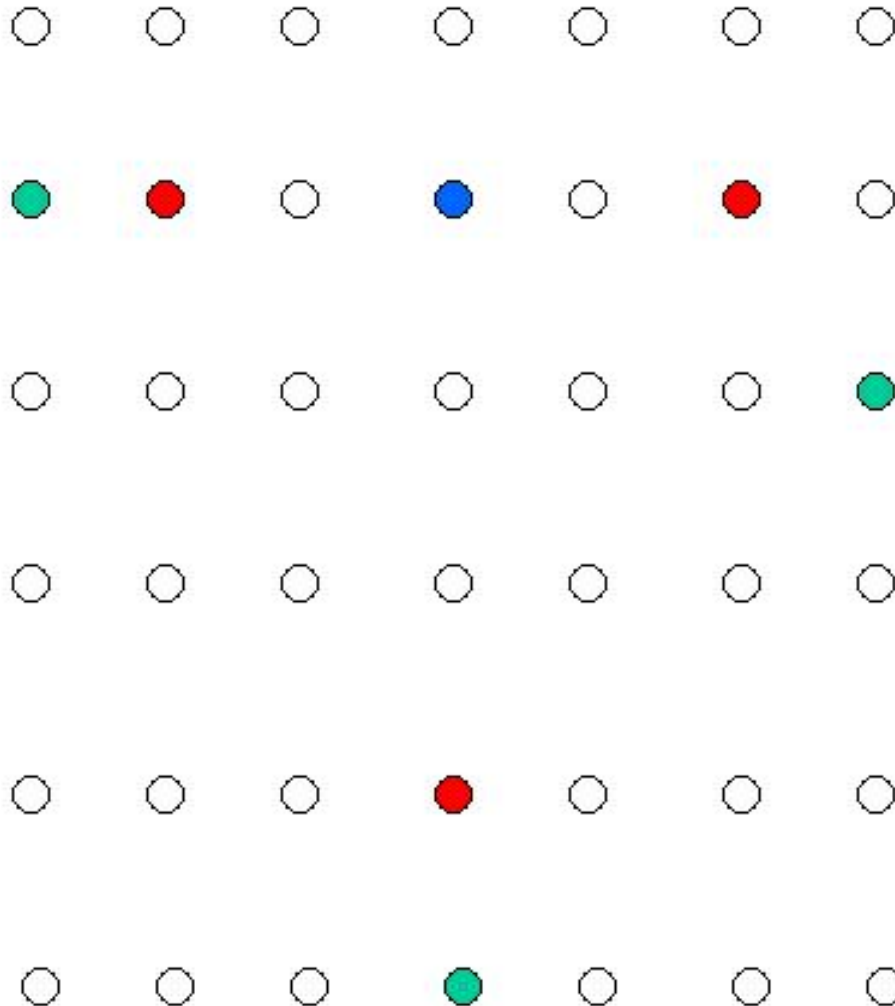
Il peut choisir un mot de code à la place d'un autre

Il choisit le mot qui a la probabilité la plus grande d'être correcte

## On suppose que

La probabilité qu'une coordonnée soit erronée est indépendante de sa position dans le mot

La stratégie adoptée est : **correction au plus proche voisin**



**Mots du code**

**Mots recus**

**Mot non corrigeable**

**Théorème** : Un code de paramètres  $(n, M, 2e+1)$   
peut corriger  $e$  erreurs  
peut aussi détecter  $2e$  erreurs

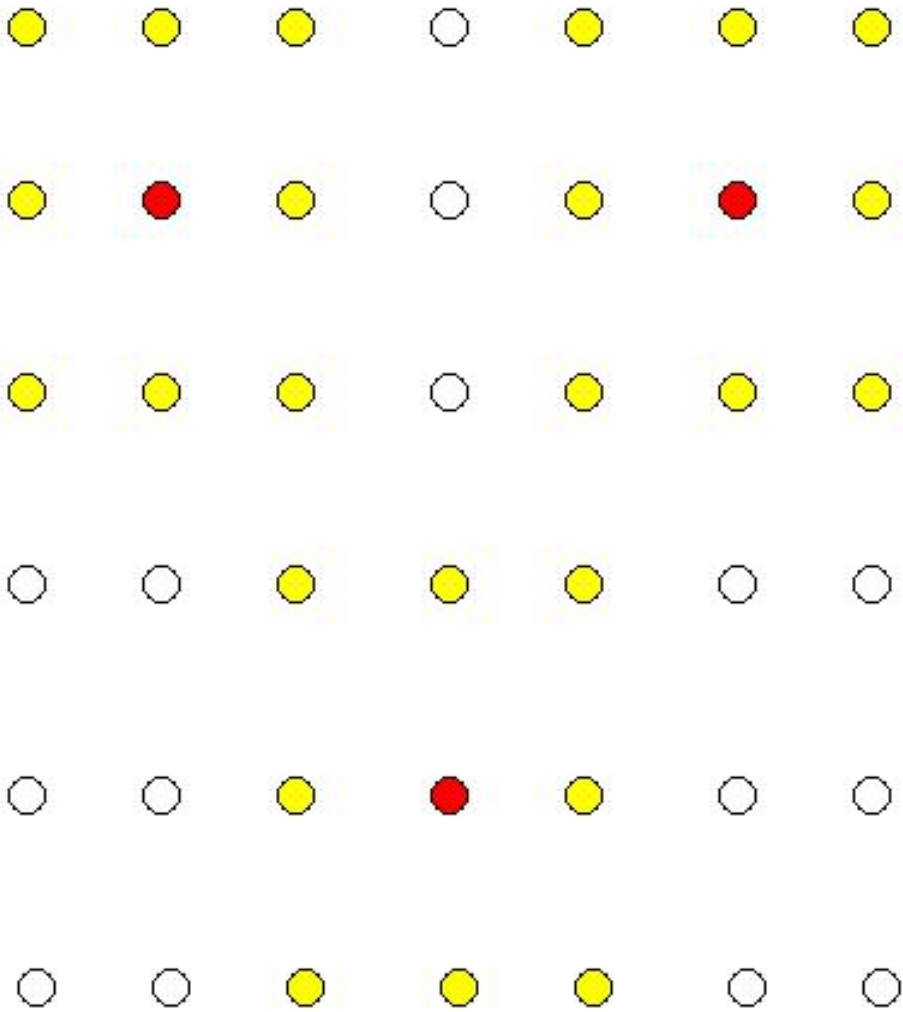
Soit  $S$  l'ensemble de tous les vecteurs de longueur  $n$  sur  $A$

$S(c) = \{x \text{ appartenant à } S : d(x, c) \leq e\}.$

$S(c)$  est la boule de centre  $c$  et de rayon  $e$

**Pour pouvoir corriger  $e$  erreurs, il faut que les boules ne s'intersectent pas**

Si le mot reçu n'appartient pas à une boule il ne peut pas être corrigé





## Définition

Si les boules sont disjointes et recouvre l'espace,  
le code est dit parfait

Tous les mots peuvent alors être corrigés

Problèmes :

1. Comment trouver des codes parfaits?

2.  $n, M$  donnés

Trouver des codes dont la distance  $d$  est optimale?

3.  $n$  et  $d$  donnés

Trouver des codes qui ont le plus de mots possibles?

**Remarque** : Trouver la distance d'un code peut prendre plusieurs siècles lorsque le code est grand et n'admet aucune structure.

**Dans la pratique**, on utilise des **codes linéaires** ce qui permet d'utiliser les outils de l'algèbre linéaire

Parmi les codes linéaires, les codes cycliques représentent la famille la plus intéressante et la plus utilisée :

Hamming, BCH, RS, Kerdock, RQ, ...

# Les codes linéaires

- Soit le code

$$C = \{00000000, 11110000, 00001111, 11111111\}$$

On peut le représenter par une matrice

$$G = \begin{pmatrix} 11110000 \\ 00001111 \end{pmatrix} \quad \text{appelée } \mathbf{matrice\ g\acute{e}n\acute{e}ratrice}$$

Le code est un sous espace vectoriel de  $F_2^8$  de dimension  $k=2$

C'est un **code linéaire** de paramètres  $[n=8, k=2, d=4]$

**Remarque** : pour calculer la distance minimale de  $C$ , il suffit de calculer le poids des mots et de prendre le plus petit non nul.

- Soit  $u$  un message de longueur  $k$  à coder et  $G$  une matrice génératrice du code  $C$  de longueur  $n$

Le mot de code est  $c = uG$

Exemple précédent, et  $u = (11)$ , on obtient  
 $C = uG = (11111111)$

On vérifie que  $c$  est un mot du code  $C$

Deux matrices génératrices qui se déduisent l'une de l'autre par permutation de colonnes donnent des codes équivalents

- Tout code linéaire est équivalent à un code C possédant un **codage systématique**, pour lequel

$$G = \left( I_k, P \right)$$

Exemple : le code de matrice génératrice

$$G = \begin{pmatrix} 10111000 \\ 01000111 \end{pmatrix}$$

Est équivalent au code de l'exemple précédent

**Il a les mêmes propriétés**

- Après un codage systématique, les  $k$  premiers symboles du mot de code sont les **symboles d'information** et les  $n-k$  restants sont les **symboles de contrôle**.
- Les mots sont donc de la forme
- $(c_0, c_1, \dots, c_{k-1}, c_k, \dots, c_{n-1})$ 

$\underbrace{\hspace{10em}}$   
 $U = \text{message}$

$\underbrace{\hspace{10em}}$   
redondance

Deux vecteurs  $x$  et  $y$  sont dits orthogonaux si  $x \cdot y = 0$

- Le code dual de  $C$  est l'ensemble des vecteurs de  $F^n$  dont le produit scalaire avec un mot du code est nul.
- La dimension du code dual est  $n-k$
- Le dual du dual de  $C$  est  $C$
- Soit  $H = \begin{pmatrix} -P^T & I_{n-k} \end{pmatrix}$  alors  $GH^T = 0$
- Tous les vecteurs engendrés par  $H$  sont duaux des mots de  $C$ . Donc si  $x$  est un mot de  $C$  alors  $xH^T = 0$
- $H$  est appelée **matrice de contrôle** de  $C$

- $c$  est dans  $C$  **SSI**  $cH^T=0$

C'est un test pour vérifier qu'un mot appartient au code.

- Soit  $y$  un mot reçu. Le syndrome est  $s(y) = yH^T$

$$y = c+e \text{ (e étant l'erreur éventuelle)}$$

or par linéarité

$$yH^T = (c+e)H^T = cH^T+eH^T = eH^T$$

**Le syndrome ne dépend que de l'erreur !**



# Comment utiliser le syndrome pour décoder?

Le code de Hamming H7 est défini par sa matrice de contrôle H, dont toutes les colonnes sont distinctes

$$H = \begin{pmatrix} 0001111 \\ 0110011 \\ 1010101 \end{pmatrix} \quad \text{soit } y = (1011000)$$

y appartient-il à H7?

$s = yH^T = (110)$ , l'erreur est donc à la sixième position :  $c = (1011010)$

- La distance minimale d'un code de Hamming est 3

Soit  $H=(h_1, \dots, h_n)$  et soit  $y = c+e$

$$S(y) = eH^T$$

Si l'erreur est de poids 1,  $e$  n'a qu'une composante non nulle en position  $i$  :  $e = (0000..1..000)$

Le syndrome vaut  **$eH^T = h_i^T$**

Les colonnes  $h_i$  étant toutes différentes, on peut localiser l'erreur

$$(00..1..00) \begin{pmatrix} h_1 \\ \cdot \\ \cdot \\ h_i \\ \cdot \\ \cdot \\ h_n \end{pmatrix} = h_i$$

Si l'erreur est de poids supérieur à 1, le code ne permet pas la correction. Par exemple, si il y a deux erreurs en position  $i$  et  $j$ , on a

$$eH^T = h_i^T + h_j^T$$

Or  $h_i + h_j$  est nécessairement égal à une autre colonne de  $H$ .

Il existe donc un mot de code à distance 1 du mot reçu  
La distance minimale est donc au moins 3

Dans  $H_7$ , il existe des mots de poids 3

Par exemple (1000101) exercice : en trouver d'autres

Donc  $d(H_7)=3$

- Plus généralement, les codes de Hamming ont une distance minimale égale à trois
- Ils peuvent corriger une erreur
- Il existe des codes qui permettent de corriger plus d'une erreur :
  - Reed et Muller
  - Codes de Résidus Quadratiques
  - Codes BCH
  - Codes de Reed-Solomon
  - Codes de Goppa
- Beaucoup de ces codes sont cycliques ou cycliques étendus

# Du codage à la crypto

## (Chiffrement de McEliece)

- Bob veut chiffrer  $m$  pour Alice
- Génération de clé : Alice
- Choisit  $[n,k]$ -code qui corrige  $t$  erreurs
- Choisit une matrice  $k \times k$  inversible :  $S$
- Choisit une matrice de permutation  $n \times n$  :  $P$
- Calcule  $R = SGP$
- Clé publique  $(R, t)$
- Clé privée  $(S, G, P)$

- Chiffrement : Bob chiffre  $m$  avec  $(R, t)$
  - $M$  est transformé en un string de  $k$  bits
  - Calcule  $c' = mR$
  - Génère aléatoirement  $z$  de poids  $t$  et longueur  $n$
  - $c = c' + z$  est le chiffré
- 
- Déchiffrement : Alice déchiffre  $c$  avec  $S, G, P$
  - Calcule  $P^{-1}$
  - Calcule  $cP^{-1}$  et décode en  $m'$
  - $M = m'S^{-1}$

# Tailles

- Pour une sécurité post quantum
- $n = 6960$  bits
- $k = 5413$  bits
- $t = 119$  bits
- Clé publique : 8 373 911 bits
- Trop long (pour l'instant) pour être pratique