

Mathématiques des codes correcteurs d'erreurs  
(Codage et tolérance aux pannes)  
(Master-2 de mathématiques (M2P),  
"Cryptologie, Sécurité et Codage d'Information",  
2014-2015)

A. A. Pantchichkine

Institut Fourier, B.P.74, 38402 St.-Martin d'Hères, FRANCE  
e-mail : [panchish@mozart.ujf-grenoble.fr](mailto:panchish@mozart.ujf-grenoble.fr), FAX : 33 (0) 4 76 51 44 78

## Résumé

L'objectif du présent cours n'est pas de proposer un exposé exhaustif de la théorie des codes correcteurs d'erreurs. Tout d'abord parce que la tâche est bien trop complexe, ensuite car la jeunesse de la théorie et son important développement actuel la rendent en perpétuel mouvement. Le but est ici d'introduire les concepts, principes et méthodes de base de l'étude des codes correcteurs d'erreurs, ainsi que de faire entrevoir certaines pistes permettant l'élaboration de codes performants.

La théorie des codes correcteurs d'erreurs se base pour l'essentiel sur l'étude des corps finis, certains rappels concernant ces derniers sont donnés dans Annexe A.

On donnera les définitions des principaux objets et grandeurs liés aux codes correcteurs d'erreurs. Les parties suivantes présenteront des classes de codes particulières, leurs propriétés et des procédures de décodage.

Ce cours introduit les outils utilisés pour assurer la transmission d'informations correctes sur des supports introduisant des erreurs. Les fondements mathématiques permettant la construction de codes avec un rendement garanti sont présentés, en particulier les codes cycliques et les codes géométriques de Goppa. Les applications dans l'industrie concernent le disque compact, le Minitel, la transmission d'images par satellite, sont mentionnées.

# Table des matières

	<b>1</b>	<b>Transmission d'information, codage et decodage optimal</b>	<b>30</b>
Mercredi 07/01/2015	1.1	Principe de transmission d'information . . . . .	30
	1.2	Hypothèses sur un canal bruité . . . . .	32
	1.3	Généralités sur les codes . . . . .	33
	1.4	Codage et decodage optimal sur un canal bruité . . . . .	38
	1.5	Théorème de Shannon (1948) . . . . .	45
	<b>2</b>	<b>Distance de Hamming, rendement et vitesse de transmission</b>	<b>50</b>
	2.1	Capacité de correction et rayon de recouvrement . . . . .	50
Mercredi 07/01/2015 et Vendredi 09/01/2015	2.2	Borne de Hamming et borne de Singleton . . . . .	53
	2.3	Bonnes familles des codes et problèmes asymptotiques. . . . .	56
	2.4	Borne de Hamming et borne de Singleton asymptotiques . . . . .	57
	<b>3</b>	<b>Codes linéaires et codes cycliques. Matrice génératrice et syndrome</b>	<b>71</b>
Mercredi 14/01/2015	3.1	Codes linéaires . . . . .	71
Mercredi 04/02/2015	3.2	Détection et correction d'erreurs, decodage . . . . .	77
	3.3	Classe des codes de Hamming . . . . .	86
	3.4	Énumération de poids et l'identité de MacWilliams . . . . .	89
	3.5	Codes cycliques . . . . .	95
	3.6	Constructions . . . . .	96

3.6.1.	Construction par polynôme générateur . . . . .	96
3.6.2.	Construction par polynôme correcteur . . . . .	98

## 4 Polynômes locateurs d'erreurs. Application au décodage 108

4.1	Construction de codes cycliques à partir des racines. . . . .	108
4.2	Exemples : codes de Golay . . . . .	112
4.2.1.	Code $G_{23}$ . . . . .	112
4.2.2.	Code $G_{24}$ . . . . .	132
4.2.3.	Code $G_{11}$ . . . . .	133
4.2.4.	Code $G_{12}$ . . . . .	141
4.3	Locateurs d'erreurs . . . . .	142
4.4	Décodage des codes cycliques . . . . .	143
4.4.1.	Exemples de decodage des codes cycliques . . . . .	147

## 5 Codes BCH et codes de Reed-Solomon. Codage et décodage 154

5.1	Classe des codes BCH (Bose, Ray-Chaudhuri et Hocquenghem) . . . . .	154
5.2	Codes de Reed-Solomon . . . . .	167
5.3	Deuxième description des codes de Reed-Solomon . . . . .	169
5.4	Problèmes de décodage. . . . .	171
5.4.1.	Décodage BCH . . . . .	178
5.5	Algorithme de Berlekamp-Massey . . . . .	180

<b>6</b>	<b>Bornes de Plotkin et de Gilbert-Varshamov</b>	<b>204</b>
6.1	Rendement, taux de correction et domaine de codes . . . . .	204
6.2	Borne de Plotkin . . . . .	210
6.3	Borne de Gilbert-Varshamov . . . . .	219
6.4	Borne de la géométrie algébrique (sans démonstration) . . . . .	226
<b>7</b>	<b>Codes géométriques de Goppa. Systèmes affines et courbes algébriques</b>	<b>234</b>
7.1	Systèmes affines et systèmes projectifs . . . . .	234
7.1.1.	Systèmes affines. . . . .	234
7.1.2.	Systèmes projectifs. . . . .	237
7.2	Codes géométriques . . . . .	239
7.3	Codes de Goppa rationnels . . . . .	242
7.3.1.	Construction des bonnes familles . . . . .	248
7.4	Décodage des codes de Goppa rationnels . . . . .	252
7.5	Espace projectif $\mathbb{P}^n$ , variétés algébriques . . . . .	255
7.6	Codes provenant des courbes algébriques . . . . .	264
7.6.1.	$\mathcal{P}$ -construction . . . . .	265
7.6.2.	Borne de la géométrie algébrique . . . . .	267
7.7	Généralités sur les diviseurs . . . . .	270
7.7.1.	Différentielles et calcul de $l(A)$ . . . . .	272
7.8	Courbes sur les corps finis . . . . .	274

Notions des codes  
géométriques et  
leurs familles



7.9	Théorème de Riemann-Roch sur un corps $K$ , genre . . . . .	274
7.10	Codes géométriques de Goppa : $L$ -construction. . . . .	277
7.11	Codes géométriques de Goppa : $\Omega$ -construction. . . . .	286
<b>8</b>	<b>Exercices de préparation à l'examen</b>	<b>288</b>
8.1	Estimation des sommes binomiales. . . . .	288
8.2	Encadrement de la probabilité du decodage erroné. . . . .	288
8.3	Codes de Golay et empilement de sphères. . . . .	289
8.4	Description géométrique des codes de Reed-Solomon . . . . .	289
8.5	Codes de Reed-Muller d'ordre 1. . . . .	291
8.6	Exemples de decodage des codes cycliques. . . . .	293
8.7	Codes de Reed-Solomon . . . . .	297
8.8	Décodage des codes de Goppa rationnels. . . . .	298
8.9	Codes de Hermite. . . . .	301
8.10	Examen du 13 janvier 2003, 10h-13h, Salle 014 . . . . .	303
8.11	Examen du 17 décembre 2003, 16h30-18h30, Salle 014 . . . . .	306
8.12	Contrôle continu du jeudi 2 décembre 2004, 8h00–10h00, AMPHI . . . .	308
<b>A</b>	<b>Annexe : Rappels sur les corps finis</b>	<b>309</b>
A.1	Structure . . . . .	310
A.2	Polynômes sur les corps finis . . . . .	311

# Cours N°1


(disponible sur : <http://www-fourier.ujf-grenoble.fr/~panchish/SCCI>).

## Introduction

Ce cours introduit les outils utilisés pour assurer la transmission d'informations correctes sur des supports introduisant des erreurs. Dans une première partie, les fondements mathématiques permettant la construction de codes avec un rendement garanti sont présentés, en particulier les codes cycliques. Dans les applications pratiques, notamment en informatique et télécommunications, des variantes de ces codes sont utilisées.

Les applications des codes correcteurs d'erreurs dans l'industrie concernent  disque compact, le  Minitel, la transmission d'images par satellite, ... (voir Chapitre XV de [Pa-Wo]).

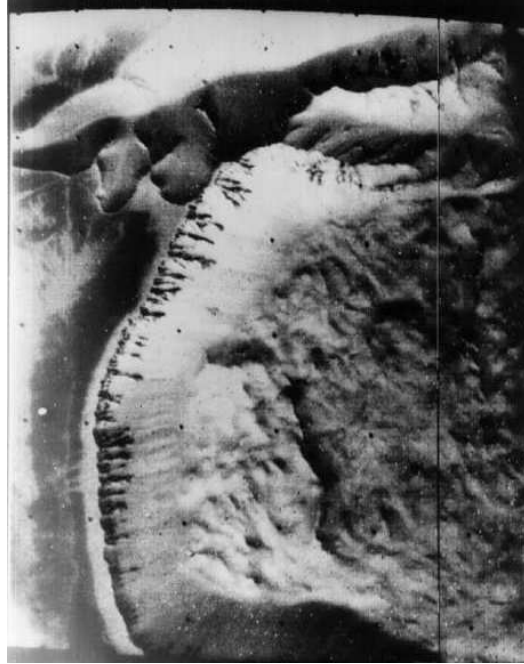
## BASES MATHÉMATIQUES :

1. Transmission d'information, codage et decodage optimal sur un canal bruité. Codes de répétition pure.
2. Distance de Hamming, rendement et vitesse de transmission, distance relative, borne de Hamming. Codes de Hamming.
3. Codes linéaires et codes cycliques. Matrice génératrice et calcul du syndrome d'erreur.
4. Polynômes locateurs d'erreurs. Application au décodage.
5. Codes de Reed-Solomon et codes BCH. Codage et décodage.
6. Bornes de Plotkin et de Gilbert-Varshamov.
7. Codes géométriques de Goppa et courbes algebriques sur les corps finis. 



A l'ère de l'information, un défi important à relever est celui de faire voyager celle-ci dans de bonnes conditions, c'est à dire de faire en sorte que le transport de l'information n'en altère pas le contenu. Aucun canal de transmission n'étant parfait, il va donc falloir «protéger» l'information pour qu'elle demeure exploitable. Les outils pour y parvenir sont les codes correcteurs d'erreurs, théorie récente de par la modernité de ses motivations.

Pour montrer de quoi il s'agit on commence par un exemple de réalisation effective d'une procédure de codage correcteur d'erreur, voir [Pa-Wo], Chapitre X. Le 19.01.1972 la sonde spatiale "MARINER-9" transmettait une photo du "Grand canyon" de la planète Mars. La très grande qualité de cette photo avait été obtenue en protégeant la transmission contre les erreurs éventuelles au moyen du "Code correcteur de Reed-Muller d'ordre 1 et de longueur 32".



On "discrétise" le problème : la photo est découpée en petits rectangles chacun d'entre eux étant assimilé à un point muni d'un "niveau d'énergie". Il existe en tout 64 niveaux d'énergie, on a donc besoin de 64 messages à transmettre, chacun représenté par une succession de 6 "bits" (symboles 0 et 1). Pour pouvoir corriger les erreurs de transmission on représente chaque message  $u$  de 6 bits par une suite plus longue  $E(u)$  de 32 bits :

$$u = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) \mapsto E(u), \mathbb{F}_2^6 \rightarrow \mathbb{F}_2^{32},$$

obtenue des combinaisons linéaires mod 2 des lignes  $A_i$  de la matrice

[illegible]

$$\mathbb{F}_2^6 \ni u = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) \xrightarrow{E} E(u) := A_1\alpha_1 + A_2\alpha_2 + A_3\alpha_3 + A_4\alpha_4 + A_5\alpha_5 + A_6\alpha_6 \in \mathbb{F}_2^{32}$$

On obtient une application

$$E : \mathbb{F}_2^6 \rightarrow \mathbb{F}_2^{32},$$

et on transmet le message  $E(u)$ . Cette transmission a la *redondance*  $= 32 - 6=26$ , le coefficient de redondance  $= 32/6$ .

Pour décrire les mots du code  $C = \text{Im}(E)$ , on indexe les colonnes  $A^{(j)}$  de la matrice (\*) par les points  $x = (x_1, x_2, x_3, x_4, x_5)$  de l'espace affine  $\mathbb{F}_2^5$  (il'y en a 32) :

$$j = 1 \leftrightarrow (0, 0, 0, 0, 0), j = 2 \leftrightarrow (0, 0, 0, 0, 1), \dots, j = 32 \leftrightarrow (1, 1, 1, 1, 1)$$

(on considère l'écriture binaire du nombre  $j - 1$ ) :

$$j - 1 = x_5 + 2x_4 + 4x_3 + 8x_2 + 16x_1, x_i = 0 \text{ ou } 1,$$

alors

$$j \in \{1, 2, \dots, 32\} \longleftrightarrow (x_1, x_2, x_3, x_4, x_5) \bmod 2 \in \mathbb{F}_2^5.$$

Ensuite, on considère chaque ligne  $A_i$  de la matrice (\*) comme la fonction indicatrice d'une partie de  $\mathbb{F}_2^5$  :

$$(1, 1) \leftrightarrow \{(*, *, *, *, *) \in \mathbb{F}_2^5\} = \mathbb{F}_2^5$$

(l'espace  $\mathbb{F}_2^5$  tout entier) ,

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1) \leftrightarrow \{(1, *, *, *, *) \in \mathbb{F}_2^5\}$$

(l'hyperplan  $\{x_1 = 1\}$ , car la partie correspondant de  $\mathbb{F}_2^5$  commence par le numero  $j = 17 \leftrightarrow (1, 0, 0, 0, 0)$ . Puis

$$(0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1) \leftrightarrow \{(*, 1, *, *, *) \in \mathbb{F}_2^5\}$$

(l'hyperplan  $\{x_2 = 1\}$ , les deux morceaux de la partie correspondant de  $\mathbb{F}_2^5$  commencent par le numero  $j = 9 \leftrightarrow (0, 1, 0, 0, 0)$  et le numero  $j = 25 \leftrightarrow (1, 1, 0, 0, 0)$ ) etc.

On vérifie que les parties de  $\mathbb{F}_2^5$  obtenues de toutes les combinaisons linéaires de  $A$  sont : l'espace  $\mathbb{F}_2^5$  tout entier,  $\emptyset$ , tous les hyperplans affines linéaires

$$f(x_1, x_2, x_3, x_4, x_5) = 1 \text{ de l'espace affine } \mathbb{F}_2^5$$

$$(d^\circ(f) = 1).$$

En effet la première ligne  $A_1$  représente la fonction constante  $f : \mathbb{F}_2^5 \rightarrow \mathbb{F}_2$  :

$$A_1 \longleftrightarrow f \equiv 1, f : \mathbb{F}_2^5 \rightarrow \mathbb{F}_2,$$

la ligne  $A_2$  représente la fonction  $f(x_1, x_2, x_3, x_4, x_5) = x_1$

$$A_2 \longleftrightarrow f(x_1, x_2, x_3, x_4, x_5) = x_1 \text{ "le premier chiffre" de } j - 1 \bmod 2 \in \mathbb{F}_2,$$

$$A_3 \longleftrightarrow f(x_1, x_2, x_3, x_4, x_5) = x_2 \text{ "le second chiffre" de } j - 1 \bmod 2 \in \mathbb{F}_2,$$

$$A_4 \longleftrightarrow f(x_1, x_2, x_3, x_4, x_5) = x_3 \text{ "le troisième chiffre" de } j - 1 \bmod 2 \in \mathbb{F}_2,$$

$$A_5 \longleftrightarrow f(x_1, x_2, x_3, x_4, x_5) = x_4 \text{ "le quatrième chiffre" de } j - 1 \bmod 2 \in \mathbb{F}_2,$$

$$A_6 \longleftrightarrow f(x_1, x_2, x_3, x_4, x_5) = x_5 \text{ "le dernier chiffre" de } j - 1 \bmod 2 \in \mathbb{F}_2.$$


Ceci implique que la combinaison linéaire repréasant la fonction

$$\alpha_1 A_1 + \alpha_2 A_2 + \alpha_3 A_3 + \alpha_4 A_4 + \alpha_5 A_5 + \alpha_6 A_6 \longleftrightarrow f = \alpha_1 + \alpha_2 x_1 + \alpha_3 x_2 + \alpha_4 x_3 + \alpha_5 x_4 + \alpha_6 x_5.$$

(la fonction indicatrice de la partie

$$f(x_1, x_2, x_3, x_4, x_5) = 1 \text{ de l'espace affine } \mathbb{F}_2^5).$$

On sait que l'intersection de deux hyperplans différents dans  $\mathbb{F}_2^5$  contient au plus 8 éléments, donc les mots du code sont bien écartés : pour obtenir un mot à partir d'un autre, il faut changer au moins 16 symboles (soit 16 soit tous les 32). C'est-à-dire, que le code corrige jusqu'à 7 erreurs de transmission : dans ce cas le résultat  $c'_1$  de la transmission d'un mot  $c_1$  ne peut pas être obtenu d'un autre mot  $c_2$  (sinon on pourrait obtenir  $c_2$  à partir de  $c_1$  en changeant  $\leq 14$  position.

DÉCODAGE  un mot reçu  $y \in \mathbb{F}_2^{32}$  vu comme une fonction  $y = y(x_1, x_2, x_3, x_4, x_5)$  sur  $\mathbb{F}_2^5$  : si  $y$  était un mot de code, on pourrait le chercher sous la forme suivante :

$$\tilde{y}(x_1, x_2, x_3, x_4, x_5) = \alpha_1 + \alpha_2 x_1 + \alpha_3 x_2 + \alpha_4 x_3 + \alpha_5 x_4 + \alpha_6 x_5, \quad (0.1)$$

alors

$$\left\{ \begin{array}{l} \alpha_1 = f(0, 0, 0, 0, 0) = y_1, \\ \alpha_1 + \alpha_2 = f(1, 0, 0, 0, 0) = y_{17}, \\ \alpha_1 + \alpha_3 = f(0, 1, 0, 0, 0) = y_9, \\ \alpha_1 + \alpha_4 = f(0, 0, 1, 0, 0) = y_5, \\ \alpha_1 + \alpha_5 = f(0, 0, 0, 1, 0) = y_3, \\ \alpha_1 + \alpha_6 = f(0, 0, 0, 0, 1) = y_2. \end{array} \right.$$

Dans ce cas on vérifie si le mot de code  $\tilde{y}(x_1, x_2, x_3, x_4, x_5)$  coïncide avec  $y$ , c'est à dire, si l'identité (0.2) soit satisfaite en tous les autres 26 points ( $26 = 32 - 6$  conditions à vérifier!) Sinon, on considère toutes les 64 combinaisons linéaires

$$y'(x_1, x_2, x_3, x_4, x_5) = \beta_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5, \quad (0.2)$$

et on choisit parmi telle que la différence  $e = e(x_1, x_2, x_3, x_4, x_5) = y - y'$  soit un mot avec le nombre minimum des valeurs non nulles. Ce mot représente "l'erreur de transmission". Alors

$$y(x_1, x_2, x_3, x_4, x_5) = \beta_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5 + e(x_1, x_2, x_3, x_4, x_5)$$

et on obtient le décodage  $u'$  du mot d'information  $u$  :

$$u' = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6).$$

Dans toute la suite les mots  $c$  d'un même code  $C$  auront la même longueur  $n$  :

$$c \in C \subset F^n.$$

De tels mots sont appelés "codes en blocs" opposés aux "codes de longueur variable" ou "codes convolutionnels" dont nous ne parlerons pas.

## Trois programmes de calcul en Maple pour trouver un decodage d'un mot reçu $y$

disponibles à l'adresse cachée : <http://www-fourier.ujf-grenoble.fr/~panchish/04mag-maple>, dans les fichiers 4mag-03linsolv.mws, 4mag-01chiff.mws, 4mag-02chiff.mws).

Pour travailler avec ces fichiers, on peut :

- 1) les ouvrir avec Netscape,
- 2) "enregistrer sous ..." comme un fichier .mws,
- 3) taper "xmaple" pour l'ouvrir avec une version de Maple.

## Résolution d'un système linéaire mod 2(4mag-03linsolv.mws)

```
> restart :with(LinearAlgebra):  
> y:=[1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1];  
> alpha:=vector(6):  
> print('alpha=[alpha[1],alpha[2],alpha[3],alpha[4],alpha[5],alpha[6]]'):  
  
y := [1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]  
 $\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6]$ 
```



$$\left\{ \begin{array}{l} \alpha_1 = f(0,0,0,0,0) = y_1, \\ \alpha_1 + \alpha_2 = f(1,0,0,0,0) = y_{17}, \\ \alpha_1 + \alpha_3 = f(0,1,0,0,0) = y_9, \\ \alpha_1 + \alpha_4 = f(0,0,1,0,0) = y_5, \\ \alpha_1 + \alpha_5 = f(0,0,0,1,0) = y_3, \\ \alpha_1 + \alpha_6 = f(0,0,0,0,1) = y_2. \end{array} \right.$$

```
> C := matrix([[1,0,0,0,0,0],
> [1,1,0,0,0,0],[1,0,1,0,0,0],[1,0,0,1,0,0],[1,0,0,0,1,0],[1,0,0,0,0,1]]
mod 2) ;
```

$$C := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

• Idem pour un vecteur second membre.

```
> b := vector([y[1],y[17],y[9],y[5],y[3],y[2]]mod 2) ;
```

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

- `Linsolve(...)` mod 2 permet de résoudre mod 2 :

> `alpha := Linsolve(C,b) mod 2;`

$$\alpha := [1, 0, 1, 1, 1, 0]$$

**Vérification. Calcul de  $C\alpha - b$ .**

> `zerov := evalm(C &* alpha - b) ;`

$$zerov := [0, 0, 2, 2, 2, 0]$$

- Les termes du vecteur obtenu ne sont pas « réduits » à leur forme canonique mod 2. Pour obtenir la réduction.

> `map(item -> Expand(item) mod 2, zerov) ;`

$$[0, 0, 0, 0, 0, 0]$$

**Application du codage :**  $y = E(\alpha) = \alpha * A = \alpha[1] * A[1] + \alpha[2] * A[2] + \alpha[3] * A[3] + \alpha[4] * A[4] + \alpha[5] * A[5] + \alpha[6] * A[6] :$

```
> A[1]:=
> vector([
> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1
> ]);
> A[2]:=vector([
> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1]);
> A[3]:=vector([
> 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 1, 1, 1, 1, 1]);
> A[4]:=vector([
> 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,
1, 0, 0, 0, 0, 1, 1, 1,
> 1]);
> A[5]:=vector([
> 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1,
1, 0, 0, 1, 1, 0, 0, 1, 1]);
> A[6]:=vector([
> 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
1, 0, 1, 0, 1, 0, 1, 0, 1
> ]);
```

```

A1 := [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
A2 := [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
A3 := [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]
A4 := [0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1]
A5 := [0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1]
A6 := [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]

```

```

> yt:=evalm(sum(alpha[i]*A[i], i=1..6)):
> yt:=map(item -> Expand(item) mod 2,yt);
  yt := [1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0]
> et:=evalm(y-yt):et:=map(item -> Expand(item) mod 2, et) ;
  et := [0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1]
> et[18];

```

0

```

> wt:=0:
> for j from 1 to 32 do
> #print('j'=j,'et[j]'=evalm(et[j]));
> if yt[j]-y[j] mod 2<>0 then wt:=wt+1 fi; od;
> print('wt'=wt);

```

$wt = 14$

## Utilisation de la multiplication des matrices : $\alpha$ par $A$

```
> A:=array(1..6,1..32,[
> [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1],
> [
> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1],
> [
> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 1, 1, 1, 1],
> [
> 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,
1, 0, 0, 0, 0, 1, 1, 1,
> 1],
> [
> 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1,
1, 0, 0, 1, 1, 0, 0, 1, 1],
> [
> 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
1, 0, 1, 0, 1, 0, 1, 0, 1
> ]
> ]);
```

$$A \coloneqq$$

```
> ys:=evalm(alpha&*A):
> ys:=map(item -> Expand(item) mod 2,ys);
```

## Vérification :

```
> yzero:=evalm(ys-yt):
> map(item -> Expand(item) mod 2,yzero);
```

Comparaison des deux décodages  $y' = yp$  et  $\tilde{y} = yt$ , et calcul du poids d'erreur (4mag-01chiff.mws) 

```
> restart;
> Chiffres:= proc(  d::nonnegint,l::nonnegint,n::nonnegint  )
>   local i,m, v;
>   v:=vector(l);
>   m:=n;
>   for i from 0 to l-1 do
>     v[l-i]:=modp(m,d);m:=floor(m/d); od;
>   return v;
>   end proc;
> evalm(Chiffres(2,12, 4095));
               [1,1,1,1,1,1,1,1,1,1,1,1]
> evalm(Chiffres(7,5,700));
               [0,2,0,2,0]
```

```

> y:=[1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1];
> yp:=vector(32):yt:=vector(32):
> print ( 'j=x[1]+2*x[2]+4*x[3]+8*x[4]+16*x[5]',
> 'x=[x[1],x[2],x[3],x[4],x[5]]',
> 'yp[j]=x[1] mod 2',
> ',
> yt[j]=1+x[2]+x[3]+x[4] mod 2') :
> wp:=0:wt:=0:
> for j from 1 to 32 do
> x:=evalm(Chiffres(2,5, j-1)):
> yt[j]:=1+x[2]+x[3]+x[4]
> mod 2:
> yp[j]:=x[1]
> mod 2:
> if
> yt[j]-y[j] mod 2<>0 then
> wt:=wt+1 fi;
> if yp[j]-y[j] mod 2<>0 then wp:=wp+1 fi;
> print ('j'=j,x=evalm(Chiffres(2,5, j-1)), 'y[j]'=y[j], 'yp[j]'=yp[j], 'yt[j]'=yt[j],
'wp'=wp, 'wt'=wt ):
> od:
> print ( 'y'=y):print ( 'yp'=evalm(yp)):print ( 'yt'=evalm(yt)):

```



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

$$j = 1, x = [0, 0, 0, 0, 0], y_j = 1, yp_j = 0, yt_j = 1, wp = 1, wt = 0$$

$$j = 2, x = [0, 0, 0, 0, 1], y_j = 1, yp_j = 0, yt_j = 1, wp = 2, wt = 0$$

$$j = 3, x = [0, 0, 0, 1, 0], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 0$$

$$j = 4, x = [0, 0, 0, 1, 1], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 0$$

$$j = 5, x = [0, 0, 1, 0, 0], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 0$$

$$j = 6, x = [0, 0, 1, 0, 1], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 0$$

$$j = 7, x = [0, 0, 1, 1, 0], y_j = 0, yp_j = 0, yt_j = 1, wp = 2, wt = 1$$

$$j = 8, x = [0, 0, 1, 1, 1], y_j = 0, yp_j = 0, yt_j = 1, wp = 2, wt = 2$$

$$j = 9, x = [0, 1, 0, 0, 0], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 2$$

$$j = 10, x = [0, 1, 0, 0, 1], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 2$$

$$j = 11, x = [0, 1, 0, 1, 0], y_j = 0, yp_j = 0, yt_j = 1, wp = 2, wt = 3$$

$$j = 12, x = [0, 1, 0, 1, 1], y_j = 0, yp_j = 0, yt_j = 1, wp = 2, wt = 4$$

$$j = 13, x = [0, 1, 1, 0, 0], y_j = 0, yp_j = 0, yt_j = 1, wp = 2, wt = 5$$

$$j = 14, x = [0, 1, 1, 0, 1], y_j = 0, yp_j = 0, yt_j = 1, wp = 2, wt = 6$$

$$j = 15, x = [0, 1, 1, 1, 0], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 6$$

$$j = 16, x = [0, 1, 1, 1, 1], y_j = 0, yp_j = 0, yt_j = 0, wp = 2, wt = 6$$

$$j = 17, x = [1, 0, 0, 0, 0], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 6$$

$$j = 18, x = [1, 0, 0, 0, 1], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 6$$

$$j = 19, x = [1, 0, 0, 1, 0], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 7$$

$$\begin{aligned}
j = 20, x &= [1, 0, 0, 1, 1], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 8 \\
j = 21, x &= [1, 0, 1, 0, 0], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 9 \\
j = 22, x &= [1, 0, 1, 0, 1], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 10 \\
j = 23, x &= [1, 0, 1, 1, 0], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 10 \\
j = 24, x &= [1, 0, 1, 1, 1], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 10 \\
j = 25, x &= [1, 1, 0, 0, 0], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 11 \\
j = 26, x &= [1, 1, 0, 0, 1], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 12 \\
j = 27, x &= [1, 1, 0, 1, 0], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 12 \\
j = 28, x &= [1, 1, 0, 1, 1], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 12 \\
j = 29, x &= [1, 1, 1, 0, 0], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 12 \\
j = 30, x &= [1, 1, 1, 0, 1], y_j = 1, yp_j = 1, yt_j = 1, wp = 2, wt = 12 \\
j = 31, x &= [1, 1, 1, 1, 0], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 13 \\
j = 32, x &= [1, 1, 1, 1, 1], y_j = 1, yp_j = 1, yt_j = 0, wp = 2, wt = 14 \\
y &= [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1] \\
yp &= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1] \\
yt &= [1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0]
\end{aligned}$$

## Un programme pour un decodage exhaustif de $y$ (4mag-02chiff.mws)

```
> restart;
> Chiffres:= proc(  d::nonnegint,l::nonnegint,n::nonnegint  )
>   local i,m, v;
>   v:=vector(l);
>   m:=n;
>   for i from 0 to l-1 do
>     v[l-i]:=modp(m,d);m:=floor(m/d); od;
>   return v;
>   end proc;
> evalm(Chiffres(2,12, 4095));
```

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

```
> evalm(Chiffres(7,5,700));
```

[0, 2, 0, 2, 0]

vérification :  $2*7+2*7^3=700$  ?

```
> 2*7+2*7^3=700;
```

$700 = 700$

```

> y:=[1,1,1,1,0,1,0,0,1,1,1,1,0,0,1,0,1,1,1,1,0,0,1,0,1,1,1,1,0,0,1,0];
> yp:=vector(32):be:=vector(6):wp:=vector(64):
> print('j=x[1]+2*x[2]+4*x[3]+8*x[4]+16*x[5]',
> 'b=be[1]+2*be[2]+4*be[3]+8*be[4]+16*be[5]+32*be[6]'):
> print('x=[x[1],x[2],x[3],x[4],x[5]]',
> 'be=[be[1],be[2],be[3],be[4],be[5],be[6]]',
> 'yp[j]=be[1]+be[2]*x[1]+be[3]*x[2]+be[4]*x[3]+be[5]*x[4]+be[6]*x[5] mod
2'):
> for b from 1 to 64 do
> be:=evalm(Chiffres(2,6, b-1));
> wp[b]:=0:
> for j from 1 to 32 do
> x:=evalm(Chiffres(2,5, j-1));
> yp[j]:=be[1]+be[2]*x[1]+be[3]*x[2]+be[4]*x[3]+be[5]*x[4]+be[6]*x[5] mod
2:
> if yp[j]-y[j] mod 2<>0 then wp[b]:=wp[b]+1 fi; od;
> #print ('yp'=evalm(yp));
> #'y[j]'=y[j], 'yp[j]'=yp[j], 'wp[b]'=wp[b]):
> if b=1 then minw:=wp[1];
> print('b'=b);
> print ('wp[b]'=evalm(wp[b]), 'be'=evalm(Chiffres(2,6, b-1)), 'yp'=evalm(yp));
> fi;
> if minw>wp[b] then minw:=wp[b];
> print('b'=b);
> print ('wp[b]'=evalm(wp[b]), 28'be'=evalm(Chiffres(2,6, b-1)), 'yp'=evalm(yp));
> od:

```

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

$$j = x_1 + 2x_2 + 4x_3 + 8x_4 + 16x_5, b = be_1 + 2be_2 + 4be_3 + 8be_4 + 16be_5 + 32be_6$$

$$b = 1$$

$$wp_b = 20, be = [0, 0, 0, 0, 0, 0], yp = [0, 0]$$

$$b = 2$$

$$wp_b = 18, be = [0, 0, 0, 0, 0, 1], yp = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$$

$$b = 3$$

$$wp_b = 14, be = [0, 0, 0, 0, 1, 0], yp = [0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1]$$

$$b = 4$$

$$wp_b = 12, be = [0, 0, 0, 0, 1, 1], yp = [0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0]$$

$$b = 37$$

$$wp_b = 4, be = [1, 0, 0, 1, 0, 0], yp = [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0]$$

## Résultat :

le vecteur  $u' = be = [1, 0, 0, 1, 0, 0]$ , obtenu à partir du mot

$$y' = [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0],$$

est le décodage cherché du mot reçu  $y$ . Il correspond à  $b = 37$ , et le poids d'erreur  $y - y'$  est égal à 4.

# 1 Transmission d'information, codage et decodage optimal sur un canal bruité

Codes de répétition pure.

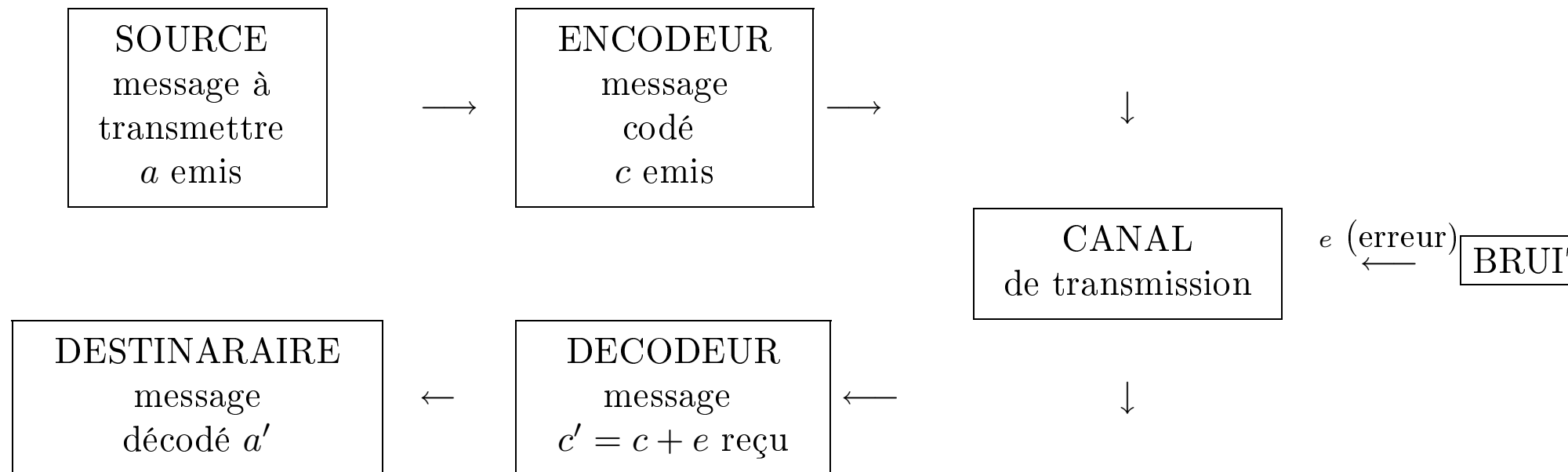
## 1.1 Principe de transmission d'information

On souhaite transmettre des informations via un canal de transmission. Celui-ci ne pouvant être parfait, l'information reçue par le destinataire peut être inexploitable ou erronée. Pour réduire au maximum la probabilité d'erreur, on construit **une procédure de codage-décodage** de l'information à transmettre qui, au prix d'éléments transmis supplémentaires, va permettre de détecter puis de corriger les altérations du message dues à

l'imperfection du canal. On se base pour ceci essentiellement sur l'étude des corps finis et des polynômes sur ceux-ci.

L'objet de la théorie de l'information est la description et l'étude des systèmes de communications, où l'information est considérée comme une grandeur mathématique, à partir du travail de Claude Shannon (1948) "The mathematical theory of communication".

Le modèle général d'un système de communication comportant une protection contre les erreurs de transmission est le suivant :



Par

- "source" on nomme tout organe ou dispositif émettant un message,

- "encodeur" un organe qui associe au message un signal de forme convenable
- "canal" le milieu utilisé pour transmettre le signal
- "decodeur" l'organe de restitution du message à partir du signal reçu
- "destinataire" la personne ou appareil qui exploite ce message

## 1.2 Hypothèses sur un canal bruité

Le message à transmettre est un bloc de symboles tous issus d'un même alphabet. On formule deux hypothèses fondamentales :

- Les perturbations de symboles au cours de la transmission sont indépendantes deux à deux et de même probabilité  $p$ .
- Le bruit du canal peut substituer un symbole à un autre, mais ne peut en adjoindre ou en supprimer. Le message reçu est ainsi de même longueur que le message émis.



### 1.3 Généralités sur les codes

DÉFINITION 1.1 (a) Soit  $F$  un ensemble de cardinal  $q$ ,  $M, n$  deux entiers strictement positifs. On appelle code  $C$  sur l'alphabet  $F$  de longueur  $n$  bloc-par-bloc *toute partie*  $C \subset F^n$  de cardinal  $\text{Card}(C) = M$ .

(b) Un code  $C \subset F^n$  est dit  $q$ -aire si  $C = \text{Im}E$  pour une application injective

$$E : F^k \longrightarrow F^n$$

L'élément  $E(u)$ , pour un  $u$  de  $F^k$  est appelé un mot code,  $k$  est dit la dimension du code,  $n$  est sa longueur. Dans ce cas  $\text{Card}(C) = q^k$

DÉFINITION 1.2 (a) Soit  $F$  un ensemble fini non vide et  $n$  entier strictement positif. L'application  $d : F^n \times F^n \longrightarrow \mathbb{N}$

$$(a, b) \mapsto \text{Card} \{i \in \{1, \dots, n\} \mid a_i \neq b_i\}$$

avec  $a = (a_1, \dots, a_n)$  et  $b = (b_1, \dots, b_n)$  est la distance de Hamming sur  $F^n$ .

(b) Soit  $F$  un corps fini. L'application  $w : F^n \rightarrow \mathbb{N}$

$$a \mapsto d(a, \underline{0}) = \text{Card} \{i \in \{1, \dots, n\} \mid a_i \neq 0\}$$

est le poids de Hamming.

REMARQUE 1.3 : *La distance de Hamming sur  $F^n$  est bien une distance sur  $F^n$ . En effet, on a :*

$$d(a, b) = 0 \iff (a_i = b_i; 1 \leq i \leq n) \iff a = b$$

$$d(a, b) = d(b, a) \text{ pour tous } a, b \text{ pour tous } E^n$$

$$d(a, c) \leq d(a, b) + d(b, c) \text{ pour tous } a, b, c \text{ pour tous } E^n.$$

DÉFINITION 1.4 Soit  $C = \text{Im}E$  un code  $q$ -aire, i.e., l'image d'une application injective  $E : F^k \rightarrow F^n$ , où  $\#F = q$ .

(a) On appelle écart ou "distance" de  $C$  le nombre

$$d = d(C) = \min_{\substack{x, y \in F^k \\ x \neq y}} d(E(x), E(y))$$

Soit  $C = \text{Im}E$  un code  $q$ -aire,  $E : F^k \rightarrow F^n$  de distance  $d$ . Dans ce cas on dit que  $C$  est un  $[n, k, d]_q$ -code.

(b) On appelle vitesse de transmission (le "rendement" ou "information rate" en anglais) de  $C$  le rapport  $R = k/n$

(c)  $1/R$  est le coefficient de redondance de  $C$ .

(d)  $\delta = d/n$  est la distance relative (ou le "taux de correction") de  $C$

Soit  $x = (x_1, \dots, x_n)$  le mot transmis par le canal. Le mot reçu  $y = (y_1, y_2, \dots, y_n)$ , éventuellement entaché d'erreurs, diffère de  $x$  en  $d(x, y)$  positions.

Lorsque on suppose qu'il n'y a pas plus de  $t$  erreurs commises,  $d(x, y) \leq t$ , on pourra retrouver  $x$  à la condition que chaque mot erroné reçu ne puisse provenir que d'un seul mot du code.

DÉFINITION 1.5 *Un code de longueur  $n$  sur l'alphabet  $F$  vérifie la condition de decodage d'ordre  $t$  si pour tout  $y \in F^n$  il existe au plus un mot  $x \in C \subset F^n$  tel que  $d(x, y) \leq t$ . Dans ce cas les boules*

$$B(x, t) = \{y \in F^n \mid d(x, y) \leq t\} \subset F^n$$

*(pour la distance de Hamming) sont deux-à-deux disjointes.*

THÉORÈME 1.6 *Un code  $C$  peut corriger  $t$  erreurs si son écart  $d$  est tel que  $d \geq 2t + 1$ .*

*Preuve* Si  $c$  est envoyé et  $y$  reçu, tels que  $d(y, c) \leq t$ , tout mot code  $c'$  de  $C$  est tel que  $d(c, c') \geq 2t + 1$ . Or,  $d$  est une distance, donc

$$d(y, c') \geq d(c, c') - d(c, y)$$

$$d(y, c') \geq t + 1$$

$C$  peut donc corriger  $t$  erreurs. Les boules  $B(c, t)$  sont donc deux-à-deux disjointes.

DÉFINITION 1.7 (a) *Un décodage de  $E$  est une application*

$$D : F^n \rightarrow F^k$$

*telle que  $D \circ E = Id_{F^k}$ .*

(b) *On dit que  $D$  est standard ("de vraisemblance maximale") si*

$$\forall y \in F^n \forall u \in F^k, \quad d(E(u), y) \geq d(E(D(y)), y)$$

*c'est-à-dire que  $E(D(y))$  se trouve parmi les mots de codes  $E(u)$  les plus proches de  $y$ .*

REMARQUE 1.8 *L'existence d'un décodage est garantie par l'injectivité de  $E$ .*

## 1.4 Codage et decodage optimal sur un canal bruité

On considère les problèmes mathématiques de transmission d'information (vue comme une longue suite

$$F^{kN} \ni a = (a_1, \dots, a_k, a_{k+1}, \dots, a_{kN}),$$

où

$$u_1 = (a_1, \dots, a_k), u_2 = (a_{k+1}, \dots, a_{2k}), \dots, u_N = (a_{k(N-1)+1}, \dots, a_{kN-1}, a_{kN}),$$

avec les *blocs d'information*  $u_1, \dots, u_N$ . On transmet l'information bloc-par-bloc à l'aide du code

$$a \mapsto E_N(a) = (E(u_1), \dots, E(u_N)) \in F^{nN}$$

avec les hypothèses 1.2 :

$$E_N(a) \mapsto \tilde{E}_N(a) \in F^{nN}$$

Il est possible de diminuer considérablement la proportion d'erreurs de transmission d'information avec des bons codes.

**Le nombre total d'erreurs** dans chaque bloc  $\tilde{E}(u_i)$  est d'espérance d'une distribution polynomiale, donc ce nombre est  $\leq (p + \varepsilon)n$  pour tout  $\varepsilon$ , avec  $n$  assez grand, et avec une grande probabilité souhaitée  $\geq 1 - \varepsilon_1$  (très proche à 1). Une bonne mesure de qualité de code est la *distance relative* (où "*taux de correction*") :

$$\delta = \frac{d}{n}, \text{ et on suppose que } p \text{ est assez petit : } p < \frac{\delta}{2}.$$

Donc pour un choix de  $\varepsilon$  et de  $n$  on peut supposer aussi que  $p \leq \frac{\delta}{2} - \varepsilon - \frac{1}{n}$ . Dans ce cas

$$(p + \varepsilon)n \leq \left(\frac{\delta}{2} - \frac{1}{n}\right)n = \frac{d}{2} - 1 \leq t = \left\lfloor \frac{d-1}{2} \right\rfloor,$$

et toutes les erreurs de transmission seront corrigées par le code  $C$ .

C'est-à-dire,  $D_N \circ E_N(a) = a$ , si  $p < \frac{\delta}{2}$  et  $n$  assez grand, avec **une probabilité souhaitée  $\geq 1 - \varepsilon_1$  très proche à 1.**

Donc le succès du decodage des mots  $\tilde{E}_N(a) \in F^{nN}$  dépend du fait si  $\delta$  est assez grand  $p \leq \frac{\delta}{2}$  mais aussi dépend du fait que de la *vitesse de transmission*  $R = \frac{kN}{nN}$  (le rendement) est suffisante  $R > R_0$  pour transmettre un mot de code de longueur  $n$  avant que le mot suivant soit fabriqué.

C'est pourquoi on est obligé d'utiliser des *codes longs*, avec  $n \approx 300 \sim 1000$ .

DÉFINITION 1.4 Soit  $C = \text{Im}E$  un code  $q$ -aire, i.e., l'image d'une application injective  $E : F^k \rightarrow F^n$ , où  $\#F = q$ . On appelle écart ou "distance" de  $C$  le nombre

$$d = d(C) = \min_{\substack{x, y \in F^k \\ x \neq y}} d(E(x), E(y))$$

Dans ce cas on dit que  $C$  est un  $[n, k, d]_q$ -code.

DÉFINITION 1.9 Une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes est dite bonne s'il existe et si sont positives toutes les deux limites

$$\lim_{i \rightarrow \infty} \frac{k_i}{n_i} = R > 0, \lim_{i \rightarrow \infty} \frac{d_i}{n_i} = \delta > 0,$$

Avec toute bonne famille on peut faire tendre vers 1 la probabilité d'une transmission correcte ayant en même temps la vitesse de transmission suffisante  $R > R_0$  pour pouvoir transmettre un mot de code de longueur  $n$  avant qu'un mots suivant soit fabriqué. En réalité, on utilise  $R_0 \approx 0,1 \sim 0,95$ .

### Illustration de l'idée :

pour pouvoir reconnaître les erreurs d'impression pendant la lecture on utilise un bon vocabulaire où les mots sont écartés : il n'y a qu'un seul mot qui ressemble au mot imprimé



(erroné).

Un procédé courant du "bit de parité", permet dans le cas d'une erreur de la "détecter" (sans la corriger!) Pour cela on attribue à un mot d'information binaire  $u = (u_1, \dots, u_k)$  la somme des coordonnées :  $E : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^{k+1}$ ,

$$x = E(u) = (u_1, \dots, u_k, x_{k+1}), \text{ où } x_{k+1} = u_1 + \dots + u_k \in \mathbb{F}_2.$$

Une autre méthode est celle de répétition :

$$E_m : F^k \rightarrow F^{km},$$

elle consiste à envoyer  $m$  fois le même mot :  $n = mk$ . La suite de codes  $\{C_m = E_m\}$  est mauvaise : le rendement tend vers zéro :  $\frac{k_m}{n_m} = \frac{1}{m} \rightarrow 0$ .

En réalité, si une source produit  $r$  symboles par minute, il y a une restriction sur la vitesse : un canal peut transmettre seulement  $m_0 r$  symboles par minute,  $m_0 \approx 1, 1 \sim 10$  donc il faut que  $n \leq km_0$ ,  $\frac{k}{n} \geq \frac{1}{m_0}$ . On voit donc que la répétition est inutile à cause de la restriction sur la longueur

$$k = 1, \frac{k}{n} \geq \frac{1}{m_0} \Rightarrow n \leq m_0.$$

DÉFINITION 1.9 (RAPPEL) *Une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes est dite bonne s'il existe et si sont positives toutes les deux limites*

$$\lim_{i \rightarrow \infty} \frac{k_i}{n_i} = R > 0, \lim_{i \rightarrow \infty} \frac{d_i}{n_i} = \delta > 0,$$

Avec toute bonne famille on peut faire tendre vers 1 la probabilité d'une transmission correcte ayant en même temps la vitesse de transmission suffisante  $R > R_0$ .

Le succès du decodage des mots  $\tilde{E}(u) \in F^n$  dépend du fait que  $\delta = \frac{d}{n}$  est assez grand :  $\frac{\delta}{2} > p$ , où  $p = p_s$  est la probabilité des perturbations de symboles au cours de la transmission.

En même temps, on demande que de la *vitesse de transmission*  $R = \frac{k}{n}$  (le rendement) soit suffisante  $R > R_0$  pour transmettre un mot de code de longueur  $n$  avant que le mot suivant soit fabriqué (on utilise  $R_0 \approx 0,1 \sim 0,95$ ).

Enfin, on est obligé d'utiliser des *codes longs*, avec  $n \approx 300 \sim 1000$ , pour que le nombre total d'erreurs dans chaque bloc  $\tilde{E}(u)$  soit  $\leq (p + \varepsilon)n$  avec  $n$  assez grand et  $\varepsilon < \frac{\delta}{2} - p - \frac{1}{n}$ .

## Codes de répétition pure

Soit  $p = p_s$  la probabilité des perturbations de symboles au cours de la transmission. On considère  $F = \{0, 1\}$  et l'application  $E : F \rightarrow F^n$  du codage de répétition pure. Calculer la probabilité  $P$  du decodage erroné.

(a) Montrer que

$$P = \sum_{0 \leq l < n/2} \binom{n}{l} (1-p)^l p^{n-l} = \mathcal{O}(p^{n/2}) \text{ lorsque } p \rightarrow 0.$$

En déduire, qu'on ne peut pas obtenir une très petite probabilité du decodage erroné à cause de la restriction  $n \leq m_0$ .

(b) Pour tout  $p < \frac{1}{8}$ , encadrer  $P$  (voir [vLi], p.24, et la section suivante).

## 1.5 Théorème de Shannon (1948)

Ce résultat affirme l'existence de bonne transmission d'information à l'aide d'une famille convenable des codes longs. Shannon a considéré un canal binaire bruité ( $q = 2$ ), voir [Sha].

Soit  $P_C = M_n^{-1} \sum_{i=1}^{M_n} P_i$  la probabilité de decodage incorrecte par un mot du code  $C \subset \mathbb{F}_2^n$

où  $P_i$  est la probabilité de decodage erroné d'un mot  $E(u_i) \in C$ .

NOTATION : on pose  $P^*(M_n, n, p) =$  la valeur minimale de la probabilité de decodage incorrecte par un mot sur tous les codes  $C \subset \mathbb{F}_2^n$  du cardinal  $\text{Card}(C) = M_n$  (c'est à dire, que  $k = [Rn]$ ),

$$P^*(M_n, n, p) = \min_{\substack{C \\ \text{Card}(C)=M_n}} (P_C = M_n^{-1} \sum_{i=1}^{M_n} P_i).$$

Pour formuler le théorème de Shannon, on utilise la fonction d'entropie  $H(p)$  définie pour  $0 < p < 1$  par l'égalité :  $0 < H(p) = -p \log_2 p - (1 - p) \log_2(1 - p) < 1$ .

THÉORÈME 1.10 (SHANNON) *On suppose que  $p < 1/2$ , et soit  $R$  un nombre strictement positif tel que*

$$0 < R < 1 + p \log_2 p + (1 - p) \log_2(1 - p) < 1,$$

*et on pose  $M_n := 2^{\lfloor Rn \rfloor}$ . Soit  $P^*(M_n, n, p)$  la valeur minimale de la probabilité de décodage incorrecte par un mot sur tous les codes  $C \subset \mathbb{F}_2^n$  du cardinal  $\text{Card}(C) = M_n$  :*

$$P^*(M_n, n, p) = \min_{\substack{C \\ \text{Card}(C) = M_n}} (P_C = M_n^{-1} \sum_{i=1}^{M_n} P_i)$$

Alors  $P^*(M_n, n, p) \rightarrow 0$  si  $n \rightarrow \infty$  .

(voir [vLi], p.27 pour une preuve détaillée, basée sur la théorie de probabilité). 

Le théorème de Shannon implique l'existence des bonne familles des codes binaires  $\{C_n\}_n$  ( $[n, k_n, d_n]_2$ -codes) de longueur  $n \rightarrow \infty$ , avec  $k_n = [Rn]$ .

On expliquera dans la section suivante la **signification géométrique de l'entropie** :  $H(p)$  est la proportion logarithmique du volume  $V(n, d)$  de la boule de Hamming de rayon relatif  $p$  :

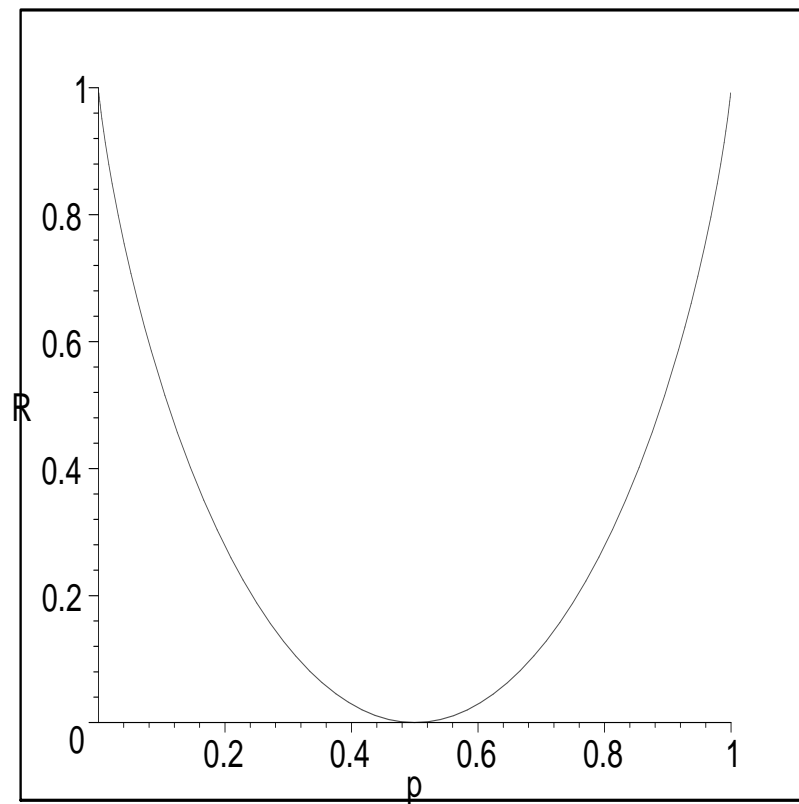
$$H(p) = \lim_{n \rightarrow \infty} \frac{\log_2(V_2(n, r))}{n}, \quad \text{si } \frac{r}{n} \rightarrow p.$$

### Représentation graphique : fonction d'entropie

```
> H(p):=- p*log(p)/log(2)-(1-p)*log(1-p)/log(2);
> 'H(p)=- p*log(p)/log(2)-(1-p)*log(1-p)/log(2)';
```

$$H(p) := -\frac{p \ln(p)}{\ln(2)} - \frac{(1-p) \ln(1-p)}{\ln(2)}$$

```
> plot([1-H(p)], p = 0..1, R = 0..1,discont=true);
> 'R=1+p*log(p)/log(2)+(1-p)*log(1-p)/log(2)';
```



$$R = 1 + \frac{p \log(p)}{\log(2)} + \frac{(1-p) \log(1-p)}{\log(2)}$$



REMARQUE 1.11


$$P^*(M_n, n, p) = \min_C (P_C = M_n^{-1} \sum_{i=1}^{M_n} P_i)$$

où  $P_i$  est la probabilité de decodage erroné d'un mot  $E(u_i) \in C$ . Le théorème signifie qu'il existe une famille  $\{C_n\} \subset \mathbb{F}_2^n$  des codes binaires de cardinal

$$\text{Card}(C_n) = M_n = 2^{[Rn]} \rightarrow \infty$$

telle que la valeur minimale de la probabilité de decodage incorrecte par un mot du code  $C_n \subset \mathbb{F}_2^n$

$$P^*(M_n, n, p) = P_{C_n} = M_n^{-1} \sum_{i=1}^{M_n} P_i \rightarrow 0.$$

REMARQUE 1.12 Dans le cas d'un canal  $q$ -aire on utilise la **borne de Gilbert -Varshamov**  $R \geq 1 - H_q(\delta)$  pour l'existence des bons codes, où  $H_q(\delta)$  est la fonction d'entropie  $q$ -aire : pour  $\delta \in [0, \frac{q-1}{q}]$  

$$H_q(0) = 0, H_q(\delta) = \frac{\delta \log(q-1)}{\log(q)} - \frac{\delta \log(\delta)}{\log(q)} - \frac{(1-\delta) \log(1-\delta)}{\log(q)}.$$



## 2 Distance de Hamming, rendement et vitesse de transmission, distance relative

Borne de Hamming. Borne de Singleton.

### 2.1 Capacité de correction et rayon de recouvrement

Soit  $C = \text{Im}E$  un  $[n, k, d]_q$ -code,  $E : F^k \rightarrow F^n$ . Rappelons que  $C$  vérifie la condition de decodage d'ordre  $t$  (voir Définition 1.5) si pour tout  $y \in F^n$  il existe au plus un mot  $x \in C \subset F^n$  tel que  $d(x, y) \leq t$ . Dans ce cas les boules

$$B(x, t) = \{y \in F^n \mid d(x, y) \leq t\} \subset F^n$$

pour la distance de Hamming soient deux-à-deux disjointes, et le code  $C$  peut corriger  $t$  erreurs si son écart  $d$  est tel que  $d \geq 2t + 1$  (voir Théorème 1.6).

DÉFINITION 2.1 (a) Le nombre  $t = \left\lfloor \frac{d-1}{2} \right\rfloor$  est dit **la capacité de correction**

(b) On appelle **rayon de recouvrement**  $\rho(C)$  du code  $C$  le plus petit rayon  $r$  tel que l'ensemble des boules  $B(x, r)$  de rayon  $r$  centrées en chaque mot de code  $x \in C$ , forme un recouvrement de  $F^n$ . On a  $t \leq \rho(C)$  et en cas d'égalité le code  $C$  est dit **parfait**.

QUESTIONS :

1) Quel est le nombre maximum de mots que peut contenir un code  $C$  de capacité de correction  $t$  ? 2) Quel est le nombre minimum de mots que doit contenir un code  $C$  pour avoir un rayon de recouvrement  $\rho(C)$  fixé ?

Pour répondre à ces deux questions on montre tout d'abord que  $\text{Card } B(x, r)$  ne dépend pas de  $x \in C$  :  $\text{Card } B(x, r) = |B(x, r)| =: V_q(n, r)$  ("le volume de la boule de Hamming").

PROPOSITION 2.2 *Pour tout entier  $r \leq n$ , et tout  $x \in F^n$*

$$\text{Card } B(x, r) = V_q(n, r) = \sum_{i=0}^r (q-1)^i \binom{n}{i}$$

*Preuve.* On peut représenter  $B(x, r)$  comme la réunion des sphères  $S(x, i)$  centrées en  $x$  et de rayon  $i \leq r$ .

Calculons  $\text{Card}(S(x, i))$  pour  $x$  et  $i$  fixés

$$\text{Card}(S(x, i)) = \text{Card} \{y \in F^n \mid d(x, y) = i\}$$

l'ensemble des mots dont le nombre de composantes distinctes de celles de  $x$  est  $i$ .

Il y a donc  $\binom{n}{i}$  ensembles d'indices à  $i$  éléments possibles. Chacune des  $i$  composantes distinctes peut être de  $q-1$  façons, c'est à dire

$$\text{Card}(S(x, i)) = (q-1)^i \binom{n}{i} \quad \blacksquare$$

## 2.2 Borne de Hamming et borne de Singleton

THÉORÈME 2.3 (BORNE DE HAMMING) *Soit  $C \subset F^n$  un code (sur l'alphabet  $F$  de  $\text{Card}(F) = q$ ) de distance  $d = d(C)$ , la capacité de correction  $t = \left\lfloor \frac{d-1}{2} \right\rfloor$  et de cardinal  $M = \text{Card}(C)$ . Alors*

$$M \sum_{i=0}^t (q-1)^i \binom{n}{i} \leq q^n, \text{ donc } V_q(n, t) \leq q^n / M \iff M \leq q^n / V_q(n, t).$$

*Preuve* est impliquée par le fait que les boules  $B(x, t)$  centrées en  $x \in C \subset F^n$ , sont disjointes, car  $2t + 1 \leq d$ .

COROLLAIRE 2.4 Soit  $C \subset F^n$  un code sur l'alphabet  $F$  de  $\text{Card}(F) = q$  de distance  $d = d(C)$  et la capacité de correction  $t = \lfloor \frac{d-1}{2} \rfloor$ . Alors

$$\max_C (\text{Card}(C)) \leq q^n / V_q(n, t)$$

REMARQUE 2.5 Soit  $C \subset F^n$  un code (sur l'alphabet  $F$  de  $\text{Card}(F) = q$ ) ayant le rayon de recouvrement  $\rho(C)$ . Alors

$$\min_C (\text{Card}(C)) \geq q^n / V_q(n, \rho(C))$$

OPTIMISATION DES PARAMÈTRES : on considère les codes  $C \subset F^n$  sur l'alphabet  $F$  de  $\text{Card}(F) = q$  de cardinal  $\text{Card}(C) = M$ , de distance  $d = d(C)$ . Deux des paramètres  $(n, M, d)$  sont fixés, on cherche à déterminer *la plus grande valeur possible* pour le troisième.

THÉORÈME 2.6 (BORNE DE SINGLETON) *Soit  $C = \text{Im}E$  un  $[n, k, d]_q$ -code,  $E : F^k \rightarrow F^n$ . Alors*

$$k \leq n - d + 1 \iff R \leq 1 - \delta + 1/n,$$

où  $R = k/n$ ,  $\delta = d/n$ .

*Preuve* On pose

$$x = (x_1, \dots, x_n) \in F^n, x' = (x_1, \dots, x_{n-d+1}) \in F^{n-d+1},$$

alors l'application  $E' : F^k \rightarrow F^{n-d+1}$  obtenue de  $E : F^k \rightarrow F^n$  par  $E'(u) := E(u)'$ , reste injective car le poids de  $C$  est  $d$ , d'où  $k \leq n - d + 1$ .

DÉFINITION 2.7 *Un code atteignant la borne de Singleton est dit M.D.S. ("maximum distance separable" en anglais).*

Le codes M.D.S. triviaux sont de type :  $[n, 1, n], [n, n - 1, 2], [n, n, 1]$ .

## 2.3 Bonnes familles des codes et problèmes asymptotiques.

Un bon code est un code corrigeant de nombreuses erreurs compte tenu de sa longueur tout en ayant une vitesse de transmission la plus élevée possible, ce qui correspond à des paramètres  $R$  et  $\delta$  assez grands dans  $[0, 1]$ . Rappelons (voir Definition 1.9) qu'une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes est dite bonne s'il existe et positives les limites

$$\lim_{i \rightarrow \infty} \frac{k_i}{n_i} = R > 0, \lim_{i \rightarrow \infty} \frac{d_i}{n_i} = \delta > 0,$$

Toutefois, ces paramètres  $\delta$  et  $R$  sont liés par des relations dites *asymptotiques*. Nous allons établir quelques relations.



## 2.4 Borne de Hamming et borne de Singleton asymptotiques

THÉORÈME 2.8 (BORNE DE SINGLETON ASYMPTOTIQUE) *Soit  $\{C_i\}$  une famille des  $[n_i, k_i, d_i]_q$ -codes, on pose*

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

*alors*

$$R \leq 1 - \delta$$

PREUVE : Pour toute  $i$ , on pose  $R_i = k_i/n_i$ ,  $\delta_i = d_i/n_i$ , alors  $R_i \leq 1 - \delta_i + 1/n_i$ . Lorsque  $i \rightarrow \infty$ , on obtient le résultat :  $R \leq 1 - \delta$ .

BORNE DE HAMMING ASYMPTOTIQUE utilise la *fonction d'entropie*  $q$ -aire : pour  $\delta \in [0, \frac{q-1}{q}]$

$$H_q(0) = 0, H_q(\delta) = \frac{\delta \log(q-1)}{\log(q)} - \frac{\delta \log(\delta)}{\log(q)} - \frac{(1-\delta) \log(1-\delta)}{\log(q)}$$

PROPOSITION 2.9 

$$\frac{1}{n+1} q^{nH_q\left(\frac{r}{n}\right)} \leq V_q(n, r) = \sum_{i=0}^r (q-1)^i \binom{n}{i} \leq q^{nH_q\left(\frac{r}{n}\right)} \quad (2.1)$$

*Preuve* On remarque tout d'abord, que pour tout  $z \in ]0, 1]$ , et pour tout  $i \leq r$ ,

$$z^{i-r} \geq 1 \implies \sum_{i=0}^r (q-1)^i \binom{n}{i} \leq \sum_{i=0}^r (q-1)^i \binom{n}{i} z^{i-r}.$$

Puis, on utilise l'inégalité : pour tout  $z \in ]0, 1]$ ,

$$\sum_{i=0}^r (q-1)^i \binom{n}{i} z^{i-r} \leq \sum_{i=0}^n (q-1)^i \binom{n}{i} z^{i-r} = z^{-r} (1 + (q-1)z)^n =: g(z).$$

Pour trouver le minimum de  $g(z)$  on calcule la dérivée  $g'(z)$  :

$$> \text{diff}(z^{(-r)} * (1 + (q-1) * z)^n, z);$$

$$-\frac{z^{(-r)} r (1 + (q-1) z)^n}{z} + \frac{z^{(-r)} (1 + (q-1) z)^n n (q-1)}{1 + (q-1) z}$$

Ceci dit,  $g'(z)$  est égale à

$$-r z^{-1-r} (1 + (q-1) z)^n + n (q-1) z^{-r} (1 + (q-1) z)^{n-1} = z^{-1-r} (1 + (q-1) z)^{n-1} ((q-1)(n-r)z - r).$$

Ceci implique

$$g'(z) = 0 \iff -z^{-1-r} (1 + (q-1) z)^{n-1} ((q-1)(n-r)z - r) = 0.$$

donc

$$z_0 = \frac{r}{(q-1)(n-r)} = \frac{r/n}{(q-1) \left(1 - \frac{r}{n}\right)},$$

$$g(z_0) = z_0^{-r} (1 + (q-1) z_0)^n, 1 + (q-1) z_0 = 1 + \frac{r}{n-r} = \frac{n}{n-r} = \frac{1}{1 - \frac{r}{n}},$$

et le minimum est

$$g(z_0) = (q-1)^r \left(\frac{r}{n}\right)^{-r} \left(1 - \frac{r}{n}\right)^{n-r}.$$

## Un exemple numérique :

```
> restart;q:=3;r:=1;n:=3;  
> g(z):=z^(-r)*(1+(q-1)*z)^n;  
> gprime(z):=diff(z^(-r)*(1+(q-1)*z)^n,z);
```

$$q := 3$$

$$r := 1$$

$$n := 3$$

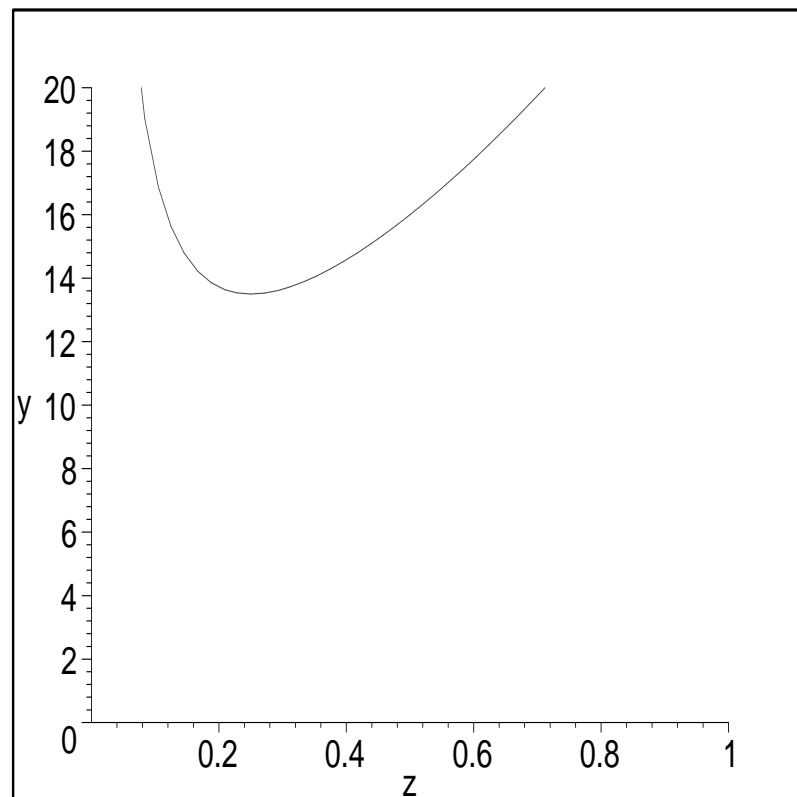
$$g(z) := \frac{(1 + 2z)^3}{z}$$

$$gprime(z) := -\frac{(1 + 2z)^3}{z^2} + \frac{6(1 + 2z)^2}{z}$$

```
> z0:=  
> r /((q-1)*(n-r));
```

$$z0 := \frac{1}{4}$$

```
> plot([g(z)], z = 0 .. 1, y = 0..20, discontinuity=true);  
> 'y=g(z)';
```



$$y = g(z)$$

D'autre part

$${}_q n H_q \left( \frac{r}{n} \right) = q^n \frac{\frac{r}{n} \log(q-1)}{\log(q)} - \frac{\frac{r}{n} \log(\frac{r}{n})}{\log(q)} - \frac{(1 - \frac{r}{n}) \log(1 - \frac{r}{n})}{\log(q)} =$$

$$(q-1)^r \left( \frac{r}{n} \right)^{-r} \left( 1 - \frac{r}{n} \right)^{n-r} = g(z_0),$$

d'où **l'inégalité supérieure**.

L'INÉGALITÉ INFÉRIEURE : on montre qu'il existe  $\tilde{z} \in ]0, 1]$  tel que

$$\sum_{i=0}^r (q-1)^i \binom{n}{i} > (q-1)^r \binom{n}{r} \geq \frac{1}{n+1} \tilde{z}^{-r} (1 + (q-1)\tilde{z})^n \geq \frac{1}{n+1} g(z_0). \quad (2.2)$$

En effet, le développement  $(1 + (q-1)\tilde{z})^n = \sum_{i=0}^n A_i \tilde{z}^i$  contient  $n+1$  termes,  $A_i \tilde{z}^i = \binom{n}{i} (q-1)^i \tilde{z}^i$ , et on peut choisir  $\tilde{z}$  de telle façon que le terme  $A_i \tilde{z}^i = \binom{n}{i} (q-1)^i \tilde{z}^i$  soit maximal parmi  $A_i \tilde{z}^i$  :

$$\frac{A_{i-1} \tilde{z}^{i-1}}{A_i \tilde{z}^i} = \frac{1}{\tilde{z}} \binom{n}{i-1} \binom{n}{i}^{-1} (q-1)^{-1} =$$

$$\frac{1}{\tilde{z}} \cdot \frac{n!}{(i-1)!(n-i+1)!} \cdot \frac{i!(n-i)!}{n!} \frac{1}{\tilde{z}} \frac{A_{i-1}}{A_i} = \frac{1}{\tilde{z}} \frac{i}{(n-i+1)(q-1)}.$$

On voit donc que

$$\frac{A_{i-1}\tilde{z}^{i-1}}{A_i\tilde{z}^i} = \frac{1}{\tilde{z}} \frac{i}{(n-i+1)(q-1)}$$

est une **suite croissante** pour  $0 \leq i \leq n$ . On choisira  $\tilde{z}$  de telle façon que cette suite dépasse 1 en  $r$ -ème terme. Plus précisément pour tout  $\tilde{z}$  dans le segment

$$\tilde{z} \in \left[ \frac{r}{(n-r+1)(q-1)}, \frac{r+1}{(n-r)(q-1)} \right] \iff 1 \in \left[ \frac{A_{r-1}\tilde{z}^{r-1}}{A_r\tilde{z}^r}, \frac{A_r\tilde{z}^r}{A_{r+1}\tilde{z}^{r+1}} \right],$$

on a (pour tous les  $i, j$  avec  $0 \leq i < r \leq j \leq n$ ) :

$$\frac{A_{i-1}\tilde{z}^{i-1}}{A_i\tilde{z}^i} \leq \frac{A_i\tilde{z}^i}{A_{i+1}\tilde{z}^{i+1}} \leq \dots \leq \frac{A_{r-1}\tilde{z}^{r-1}}{A_r\tilde{z}^r} \leq 1 \leq \frac{A_r\tilde{z}^r}{A_{r+1}\tilde{z}^{r+1}} \leq \dots \leq \frac{A_{j-1}\tilde{z}^{j-1}}{A_j\tilde{z}^j}.$$

Ceci implique

$$A_{i-1}\tilde{z}^{i-1} \leq A_i\tilde{z}^i \leq \dots \leq A_r\tilde{z}^r, A_r\tilde{z}^r \geq A_{r+1}\tilde{z}^{r+1} \geq \dots \geq A_j\tilde{z}^j \geq \dots$$

Mais par notre choix le terme  $A_r\tilde{z}^r = \binom{n}{i}(q-1)^r\tilde{z}^r$  est maximal parmi  $A_i\tilde{z}^i$ , donc

$$\tilde{z}^{-r}(1+(q-1)\tilde{z})^n \leq (n+1)A_r \Rightarrow g(z_0) \leq (n+1) \sum_{i=0}^r (q-1)^i \binom{n}{i}.$$

Il reste à rapeller que

$$g(z_0) = q^{nH_q\left(\frac{r}{n}\right)} = (q-1)^r \left(\frac{r}{n}\right)^{-r} \left(1 - \frac{r}{n}\right)^{n-r},$$

ceci implique la borne inférieure :

$$\frac{1}{n+1} q^{nH_q\left(\frac{r}{n}\right)} \leq V_q(n, r) = \sum_{i=0}^r (q-1)^i \binom{n}{i}.$$



RAPPELS :

PROPOSITION 2.9



$$\frac{1}{n+1} q^{nH_q\left(\frac{r}{n}\right)} \leq V_q(n, r) = \sum_{i=0}^r (q-1)^i \binom{n}{i} \leq q^{nH_q\left(\frac{r}{n}\right)} \quad (2.4)$$

COROLLAIRE 2.10 Lorsque  $n \rightarrow \infty$ ,  $\frac{r}{n} \rightarrow \delta$ , on a

$$H_q(\delta) = \lim_{n \rightarrow \infty} \frac{\log_q V_q(n, r)}{n} = \lim_{n \rightarrow \infty} \frac{\log_q(V_q(n, r))}{\log_q(\text{Card}(F^n))}$$

(la "proportion logarithmique de la boule de Hamming de rayon relatif"  $\delta$  dans  $F^n$ ).

*Preuve* On prend  $\log_q$  de l'inégalité (2.4) :

$$\begin{aligned} \frac{1}{n+1} q^{nH_q\left(\frac{r}{n}\right)} \leq V_q(n, r) = \sum_{i=0}^r (q-1)^i \binom{n}{i} \leq q^{nH_q\left(\frac{r}{n}\right)} \\ \Rightarrow -\frac{\log_q(n+1)}{n} + H_q\left(\frac{r}{n}\right) \leq \frac{\log_q V_q(n, r)}{n} \leq H_q\left(\frac{r}{n}\right). \end{aligned}$$

et si  $\frac{r}{n} \rightarrow \delta, n \rightarrow \infty$  alors  $\frac{\log_q(n+1)}{n} \rightarrow 0$  et on obtient le resultat.

THÉORÈME 2.11 (BORNE DE HAMMING ASYMPTOTIQUE) Soit  $\{C_i\}$  une famille des  $[n_i, k_i, d_i]_q$ -codes, on pose

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

alors

$$R \leq 1 - H_q(\delta/2),$$

où  $H_q(x)$  est la fonction entropie  $q$ -aire définie sur  $[0, (q-1)/q]$  par

$$H_q(0) = 0,$$

$$H_q(x) = x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x) \text{ pour } 0 \leq x \leq (q-1)/q.$$

*Preuve* Soit  $C_i \subset F^n$  un code de la famille  $\{C_i\}$ , on pose  $t_i = \left\lceil \frac{d_i-1}{2} \right\rceil$ . Alors

$$\frac{d_i-2}{2} \leq t_i = \left\lceil \frac{d_i-1}{2} \right\rceil \leq \frac{d_i-1}{2} = \frac{n_i \delta_i - 1}{2} \Rightarrow \lim_{i \rightarrow \infty} \frac{t_i}{n_i} = \lim_{i \rightarrow \infty} \frac{n_i \delta_i - 1}{2n_i} = \frac{\delta}{2}$$

Selon Théorème 2.3 (la borne de Hamming) on a

$$V_q(n_i, t_i) q^{k_i} \leq q^{n_i} \Rightarrow \log_q(V_q(n_i, t_i)) + k_i \leq n_i$$

Lorsque  $n_i \rightarrow \infty$ ,  $\frac{t_i}{n_i} \rightarrow \frac{\delta}{2}$  donc

$$\frac{\log_q(V_q(n_i, t_i))}{n_i} \rightarrow H_q(\delta/2).$$

Il vient que

$$\begin{aligned} V_q(n_i, t_i)q^{k_i} &\leq q^{n_i} \Rightarrow \log_q(V_q(n_i, t_i)) + k_i \leq n_i \\ &\Rightarrow H_q(\delta/2) + R \leq 1 \Rightarrow R \leq 1 - H_q(\delta/2). \end{aligned}$$

> restart;

FONCTION D'ENTROPIE 

> q:=4:

> Hq(x):=x\*log(q-1)/log(q)- x\*log(x)/log(q)-(1-x)\*log(1-x)/log(q);

$$Hq(x) := \frac{x \ln(3)}{\ln(4)} - \frac{x \ln(x)}{\ln(4)} - \frac{(1-x) \ln(1-x)}{\ln(4)}$$

# BORNES DE SINGLETON ET DE HAMMING (D'EMPILEMENT DE SPHÈRES) ASYMPTOTIQUES

>  $f := H_q(x);$

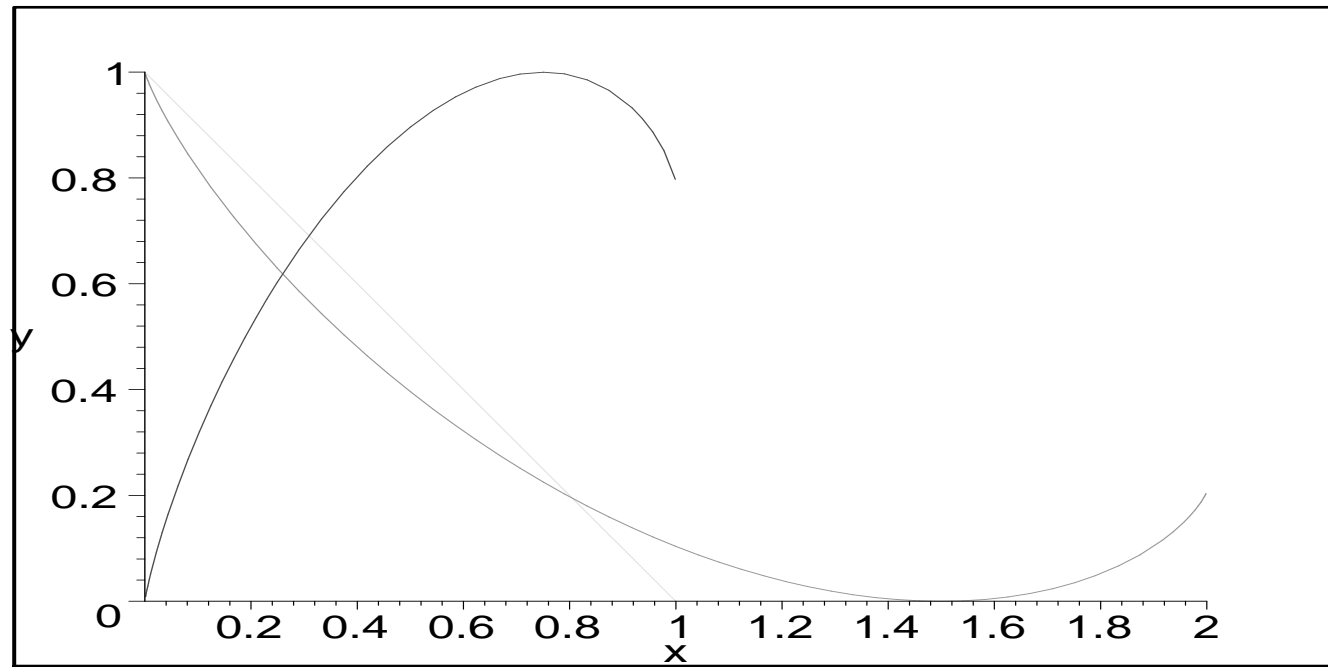
$$f := \frac{x \ln(3)}{\ln(4)} - \frac{x \ln(x)}{\ln(4)} - \frac{(1-x) \ln(1-x)}{\ln(4)}$$

>  $g1 := x/2; g := \text{algsubs}(x=g1, f);$

$$g1 := \frac{1}{2} x$$

$$g := -\frac{\ln(1 - \frac{1}{2} x) (1 - \frac{1}{2} x)}{\ln(4)} + \frac{\frac{1}{2} x (-\ln(\frac{1}{2} x) + \ln(3))}{\ln(4)}$$

>  $\text{plot}([f, 1-g(x), 1-x], x=0..2, y=0..1); 'y=H_q(x), 1-H_q(x/2), 1-x';$



$$y = \text{Hq}(x), 1 - \text{Hq}(\frac{1}{2}x), 1 - x$$

## 3 Codes linéaires et codes cycliques. Matrice génératrice et calcul du syndrome

d'erreur

### 3.1 Codes linéaires

Une classe de codes très importante est celle des codes linéaires, notamment en raison des outils dont nous disposons pour manipuler et représenter les applications linéaires à l'aide de l'écriture matricielle.

En général, pour un alphabet fini  $F$ , étant donné une énumération de  $F^k$ , la donnée d'un code  $C = \text{Im}(E)$ ,  $E : F^k \rightarrow F^n$  est la donnée de  $n \times q^k$  éléments de  $F$ , ce qui représente un gros volume d'information.

Si l'on munit  $F^k$  et  $F^n$  de structures supplémentaires et si l'on prend une application  $E$  qui respecte ces structures, on peut économiser sur le volume d'information représentant  $E$  au prix de calculs des valeurs non mémorisées de  $E$ .

En ce sens le plus simple est de prendre pour  $q$  un nombre primaire,  $q = p^r$ , pour  $F$  le corps  $\mathbb{F}_q$  et pour  $E$  une application linéaire injective de  $\mathbb{F}_q^k$  dans  $\mathbb{F}_q^n$ . Rapportant  $C$  aux bases canoniques adéquates, on caractérise cette application par  $n \times k$  éléments de  $\mathbb{F}_q$ .

Rapellons

DÉFINITION 1.2 (a) Soit  $F$  un ensemble fini non vide et  $n$  entier strictement positif. L'application  $d : F^n \times F^n \longrightarrow \mathbb{N}$

$$(a, b) \mapsto \text{Card} \{i \in \{1, \dots, n\} \mid a_i \neq b_i\}$$

avec  $a = (a_1, \dots, a_n)$  et  $b = (b_1, \dots, b_n)$  est la *distance de Hamming* sur  $F^n$ .

(b) Soit  $F$  un corps fini. L'application  $w : F^n \rightarrow \mathbb{N}$

$$a \mapsto d(a, \underline{0}) = \text{Card} \{i \in \{1, \dots, n\} \mid a_i \neq 0\}$$

est le *poids de Hamming*.



DÉFINITION 3.1 Soit  $F = \mathbb{F}_q$  un corps fini. Un code linéaire  $C$  est un sous-espace vectoriel de dimension  $k$  de l'espace vectoriel  $F^n$  (vu comme l'image d'une application  $E : F^k \rightarrow F^n$  linéaire injective). Les vecteurs lignes  $a = (a_1, \dots, a_k) \in F^k$  sont les **mots d'information**, et les vecteurs lignes  $c = E(a) = (c_1, \dots, c_n) \in F^n$  sont les **mots de code**. La **matrice génératrice**  $G$  du code  $E$  est la matrice attachée à l'application linéaire  $E : F^k \rightarrow F^n$  (dans les bases standards de  $F^k$  et  $F^n$ ), de telle façon que



$$c = E(a) = aG.$$



REMARQUE 3.2 Un code linéaire  $C$  est l'image d'une application linéaire injective, donc on peut considérer  $C$  comme un sous-espace vectoriel de dimension  $k$  de l'espace vectoriel  $F^n$ . On peut ainsi caractériser les codes linéaires à partir de matrices à coefficients dans  $F$  comme **noyau** d'une autre application linéaire  $S : F^n \rightarrow F^{n-k}$ . La matrice  $H$  de  $S$  est appelée **matrice de contrôle** de  $C$  :

$$S(c) = Hc^t$$



**EXEMPLE.** Soit  $H$  une matrice  $(n-k) \times n$  à coefficients dans  $\mathbb{F}_q$  de rang  $n-k$ . Le noyau de l'application représentée par  $H$  est un sous-espace vectoriel de  $\mathbb{F}_q^n$ . On peut donc définir un code linéaire par un système d'équations linéaires :

$$C = \left\{ c = (c_1, \dots, c_n) \mid Hc^t = 0 \right\}.$$

Posons

$$H = (A, I_{n-k}) = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

et soit  $q = 2$  ( $C$  code binaire) On désire transmettre le message

$$a = (a_1 a_2 a_3 a_4).$$

On le code en  $c = (a_1 a_2 a_3 a_4 c_5 c_6 c_7)$ , avec  $c_5, c_6, c_7$  tels que  $Hc^t = 0$ . Or,

$$Hc^t = 0 \iff \begin{array}{l} a_1 + a_3 + a_4 + c_5 = 0 \\ a_1 + a_2 + a_4 + c_6 = 0 \\ a_1 + a_2 + a_3 + c_7 = 0 \end{array} \iff \begin{array}{l} c_5 = a_1 + a_3 + a_4 \\ c_6 = a_1 + a_2 + a_4 \\ c_7 = a_1 + a_2 + a_3 \end{array}$$

On obtient l'application linéaire injective  $E : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^7$

$$(a_1, a_2, a_3, a_4) \longmapsto (a_1, a_2, a_3, a_4, a_1 + a_3 + a_4, a_1 + a_2 + a_4, a_1 + a_2 + a_3)$$

La matrice  $H$  est appelée *matrice de contrôle* de  $C$ . On a  $Hc^t = 0$  pour tous les mots de code  $c \in C$ .

REMARQUE 3.3 Si  $H = (A, I_{n-k})$ , alors un message  $a = a_1 \cdots a_k$  est codé en  $c = a_1 \cdots a_k c_{k+1} \cdots c_n$  le code est alors dit *sysématique*. Ici  $A \in \text{Mat}_{n-k,k}(F)$

De plus, on a

$$\{Hc^t = 0\} \implies c^t = \begin{pmatrix} I_k \\ -A \end{pmatrix} a^t = [a(I_k, -A^t)]^t$$

*c'est-à-dire, que*

$$\begin{pmatrix} c_{k+1} \\ c_{k+2} \\ \cdots \\ c_n \end{pmatrix} = -A \begin{pmatrix} a_1 \\ \cdots \\ a_k \end{pmatrix}, \quad \begin{pmatrix} c_{k+1} \\ c_{k+2} \\ \cdots \\ c_n \end{pmatrix} + A \begin{pmatrix} a_1 \\ \cdots \\ a_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}.$$

Il vient la définition suivante :

**DÉFINITION 3.4**  $G = (I_k, -A^t)$  est la matrice génératrice canonique du code linéaire  $C$  de matrice de contrôle  $H = (A, I_{n-k})$ . D'une manière plus générale, toute matrice  $G$  engendrant un code  $C$  est une matrice génératrice de  $C$ .

**REMARQUE 3.5** Pour tout mot code  $c$ , on a  $Hc^t = 0$  et  $c = aG$ . Donc

$$GH^t = 0 \in \text{Mat}_{k,n-k}(F), \quad HG^t = 0 \in \text{Mat}_{n-k,k}(F),$$

puisque  $Hc^t = HG^t a^t = 0$  pour tous les  $a \in F^k$ .

## 3.2 Détection et correction d'erreurs, décodage

Dans ce qui suit, nous noterons  $c$  un mot code émis,  $y$  le message reçu, et  $e = y - c$  le vecteur erreur.

La distance de Hamming  $d(y, c)$  est alors le nombre d'erreurs survenues au cours de la transmission. Pour décoder  $y$  reçu, on peut supposer que le nombre d'erreurs est minimal, c'est à dire que l'on va chercher le mot code  $c$  le plus proche de  $y$  au sens de la distance de Hamming. C'est la règle du décodage *par plus proche voisin*.



**DÉFINITION 3.6** Soit  $t$  un entier naturel.  $C$  un code linéaire de dimension  $r$  et de longueur  $n$  est dit  $t$ -correcteur d'erreurs si

$$\forall y \in \mathbb{F}_q^n, |\{c \in C : d(y, c) \leq t\}| \leq 1$$

Si alors  $c \in C$  est transmis et qu'au plus  $t$  erreurs surviennent, on a  $d(y, c) \leq t$  et  $d(y, c') > t$  pour tout autre élément de  $C$ . Ainsi, la méthode du décodage par plus proche voisin donne le bon résultat.

Il apparaît qu'un des objectifs de la théorie du codage consiste à élaborer des codes dont les mots sont très éloignés les uns des autres au sens de la distance de Hamming. Toutefois, un autre est de transmettre un maximum d'information et donc de garder des vitesses de transmission acceptables, et réunir les deux est épineux.

**THÉORÈME 3.7** Un code  $C$  peut corriger  $t$  erreurs si son écart  $d$  est tel que  $d \geq 2t + 1$

PREUVE : On a déjà vu ce resultat (Théorème 1.6) Si  $c$  est envoyé et  $y$  reçu, tels que  $d(y, c) \leq t$ , tout mot code  $c'$  de  $C$  est tel que  $d(c, c') \geq 2t + 1$ . Or,  $d$  est une distance, donc

$$d(y, c') \geq d(c, c') - d(c, y)$$

$$d(y, c') \geq t + 1$$

$C$  peut donc corriger  $t$  erreurs.

EXEMPLE. Reprenons le code déjà vu plus haut

$$\begin{aligned} E : \quad \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^7 \\ (a_1, a_2, a_3, a_4) &\mapsto (a_1, a_2, a_3, a_4, a_1 + a_3 + a_4, a_1 + a_2 + a_4, a_1 + a_2 + a_3) \end{aligned} \quad (3.1)$$

Soient  $a, b$  des éléments de  $\mathbb{F}_2^4$

Si  $d(a, b) = 1$ , alors  $d(E(a), E(b)) = 3$  ou  $4$ .



Si  $d(a, b) = 2$ , alors  $d(E(a), E(b)) = 3$  ou  $4$ .

Si  $d(a, b) = 3$ , alors  $d(E(a), E(b)) = 3$  ou  $4$ .

Si  $d(a, b) = 4$ , alors  $d(E(a), E(b)) = 7$ .

Ceci dit, l'application  $E$  écarte vraiment les mots d'information.

En effet, on peut toujours supposer  $b = (0, 0, 0, 0)$ , et on utilise directement la formule (3.1) pour le mot  $E(a)$ .

Donc  $d = 3$ , et le code corrige 1 erreur.

LEMME 3.8 *Soit un code linéaire  $C$  de matrice de correction  $H$  et d'écart  $d$ . Alors  $d \geq s+1$  si et seulement si  $s$  colonnes de  $H$  sont linéairement indépendantes.*

PREUVE : Supposons que  $s$  colonnes de  $H$  soient linéairement dépendantes. Alors il existe  $c \in C$  non nul tel que  $Hc^t = 0$  et  $w(c) \leq s$ . Ainsi,  $d \leq s$ . Inversement, si  $s$  colonnes de  $H$  sont toujours indépendantes,  $c \in C$  non nul est toujours tel que  $w(c) \geq s + 1$  et donc  $d \geq s + 1$ .

Ce qui suit est un algorithme simple de décodage des codes linéaires : le décodage par *leader de classe*.

Soit  $C$  un code linéaire de longueur  $n$  et de dimension  $k$  sur  $\mathbb{F}_q$ . L'espace vectoriel  $\mathbb{F}_q^n/C$  est formé de toutes les classes  $a + C$ ,  $a \in \mathbb{F}_q^n$ . Pour tout  $a$ ,  $|a + C| = q^k$ , et

$$\mathbb{F}_q^n = (a^{(0)} + C) \cup \dots \cup (a^{(s)} + C),$$

avec  $a^{(0)} = 0$ ,  $s = q^{n-k} - 1$ .

Alors, quel que soit le message  $y$  reçu, il existe  $i$  tel que  $y \in a^{(i)} + C$ , et si  $c$  est le message envoyé,  $e = y - c = a^{(i)} + z \in a^{(i)} + C$ . On peut ainsi construire une méthode de décodage des codes linéaires. En effet, quel que soit  $y \in a^{(i)} + C$  reçu, tous les vecteurs erreur possibles pour  $y$  sont également dans  $a^{(i)} + C$ . La règle de décodage par plus proche voisin nous conduit à choisir pour vecteur erreur le vecteur  $e \in a^{(i)} + C$  de poids de Hamming minimum, et on décode  $y$  en  $x = y - e$ . Nous allons voir maintenant l'algorithme, à proprement parler, plus en détail.



DÉFINITION 3.9 *Dans les conditions décrites précédemment, un élément de poids minimum dans  $a + C$  est appelé un leader de classe.*

Soient  $a^{(1)}, \dots, a^{(s)}$  les leaders des classes  $a + C$ ,  $a \neq 0$ , et soient  $c^{(1)} = 0, c^{(2)}, \dots, c^{(q^k)}$  tous les mots du code  $C$  et soit le tableau suivant :

$$\begin{array}{cccc}
 c^{(1)} & c^{(2)} & \dots & c^{(q^k)} \\
 a^{(1)} + c^{(1)} & a^{(1)} + c^{(2)} & \dots & a^{(1)} + c^{(q^k)} \\
 \dots & \dots & \dots & \dots \\
 a^{(s)} + c^{(1)} & a^{(s)} + c^{(2)} & \dots & a^{(s)} + c^{(q^k)}
 \end{array}$$

Si on reçoit le mot  $y = a^{(i)} + c^{(j)}$ , le vecteur erreur est  $e = a^{(i)}$  et on décode  $y$  en  $x = y - e = c^{(j)}$ , c'est à dire le mot du code (donc un terme de plus petit poids) de la colonne où est situé  $y$ . On peut déterminer la classe de  $y$  en évaluant ce que l'on appelle le syndrome de  $y$ .

DÉFINITION 3.10 Soit  $H$  la matrice de correction d'un code linéaire  $C$  de longueur  $n$  et de dimension  $k$ . Alors le vecteur  $S(y) = Hy^t$  de longueur  $n - k$  est appelé **le syndrome** de  $y$ .

THÉORÈME 3.11 Pour  $y, z$  éléments de  $\mathbb{F}_q^n$ , on a

- (i)  $S(y) = 0$  si et seulement si  $y \in C$
- (ii)  $S(y) = S(z)$  si et seulement si  $y + C = z + C$

PREUVE :  $S(y) = Hy^t$ , et  $C = \{y \in F_q^n : Hy^t = 0\}$ , d'où le (i). De plus,  $S(y) = S(z) \iff Hy^t = Hz^t \iff H(y - z)^t = 0 \iff y - z \in C \iff y + C = z + C$ , d'où le (ii).

Si le message  $c$  est envoyé et  $y$  reçu,  $e = y - c$ , alors

$$S(y) = S(c + e) = S(c) + S(e) = S(e),$$

$y$  et  $e$  sont dans la même classe, et le leader de cette classe a également le même syndrome.

Ceci nous permet d'améliorer l'algorithme précédent. Celui-ci consistait à rechercher le message reçu  $y$  dans le tableau construit précédemment, et à le décoder en remontant au premier terme de la colonne du tableau où il se situe. Or, de ce qui précède, tous les éléments d'une même ligne du tableau ont le même syndrome. Ainsi, pour ne pas perdre du temps de chercher  $y$  dans le tableau, il suffit d'y rajouter une colonne, celle des syndromes. Le décodage se fait alors comme suit :

- (i) On calcule  $S(y) = Hy^t$ .
- (ii) On cherche  $S(y)$  dans la colonne des syndromes.
- (iii) Le vecteur erreur  $e$  est le leader de cette classe, donc premier terme de la ligne où figure  $S(y)$ .
- (iv) On décode  $y$  en  $x = c(j) = y - e$ , qui est également un terme de plus petit poids de la colonne de  $y$ .

EXEMPLE. Soit  $C$  un code linéaire de matrice génératrice  $G$  et de matrice de contrôle  $H$ .

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

Mots d'information	{	00	10	01	11	
Mots codes	{	0000	1010	0111	1101	$\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}$
		1000	0010	1111	0101	$\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}$
		0100	1110	0011	1001	$\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}$
		0001	1011	0110	1100	$\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}$
		$\underbrace{\hspace{10em}}$ Leaders de classe				$\underbrace{\hspace{10em}}$ Syndromes

Si  $y = (1110)$  est le message reçu, plutôt que de chercher  $y$  dans le tableau, ce qui serait coûteux pour de grands tableaux, on calcule son *syndrome* :

$$S(y) = Hy^t = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Il vient ensuite immédiatement que le vecteur erreur est le leader de la classe correspondante, ayant le même syndrome, donc  $e = (0100)$ , et on décode  $y$  en  $x = y - e = (1010)$

Cette méthode est toutefois limitée, car pour de très grands codes, il devient impossible de trouver les leaders de classe. Un code binaire de longueur 50 et de dimension 20 possède  $\approx 10^9$  classes. Pour surmonter ces difficultés, il va falloir construire des codes particuliers.

### 3.3 Classe des codes de Hamming

THÉORÈME 3.12 *Soit  $C$  un code linéaire binaire de matrice de correction  $H$ . Alors, le syndrome d'un vecteur reçu est égal à la somme des colonnes de  $H$  correspondant aux positions des erreurs.*

PREUVE : On note  $h_j$  la  $j^{\text{me}}$  colonne de  $H$ , et soit  $y = x + e$  le message reçu,  $x \in C$ . Alors,  $S(y) = He^t$ . Si

$$e = (0, \dots, 0, \underset{i_1}{1}, 0, \dots, 0, \underset{i_2}{1}, 0, \dots),$$

alors

$$S(y) = h_{i_1} + h_{i_2} + \dots$$

Si toutes les colonnes de  $H$  sont différentes, une erreur simple en  $i^{\text{me}}$  position entraîne  $S(y) = h_j$ , donc une erreur peut être corrigée. Dans le cas de codes visant à corriger une erreur, la classe des codes de Hamming simplifie le problème de la localisation de l'erreur.

DÉFINITION 3.13 *Un code binaire  $C_m$ , de longueur  $n = 2^m - 1$ ,  $m \geq 2$ , de matrice de correction  $m \times (2^m - 1)$   $H$  est appelé code de Hamming binaire si les colonnes de  $H$  sont les écritures binaires de  $1, 2, \dots, 2^m - 1$ .*

LEMME 3.14  *$C_m$  est de dimension  $2^m - m - 1$  et corrige 1 erreur.*

PREUVE : Par construction,  $H$  est de rang  $m$ , et deux colonnes distinctes de  $H$  sont toujours linéairement indépendantes. En revanche, comme  $H$  contient avec deux colonnes distinctes également leur somme, on a l'écart de  $C$   $d = 3$ , donc  $C$  corrige une erreur.

EXEMPLE Soit  $C_3$  le code de Hamming binaire de longueur 7 et de dimension 4. Alors sa matrice de correction est

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

Le message reçu  $y = (1101111)$  a pour syndrome  $S(y) = (011)^t$ , alors nous pouvons affirmer qu'une erreur s'est produite en 3<sup>ème</sup> position, puisque 011 est l'écriture de 3 en base 2, donc le mot de code corrigé est  $x = (1111111)$ .

REMARQUE 3.15 *On peut définir les codes de Hamming  $q$ -aires  $C(m, q)$  pour tout corps fini  $\mathbb{F}_q$ . Ils sont de type  $[(q^m - 1)/(q - 1), (q^m - 1)/(q - 1) - m, 3]$ , et la matrice de contrôle est formée par les colonnes représentant les coordonnées homogènes de points différents de l'espace projectif  $\mathbb{P}_{\mathbb{F}_q}^{m-1}$*

REMARQUE 3.16 *Les codes de Hamming  $C(m, q)$  sont parfaits : ils réalisent un empilement de sphères de rayon 1 dans  $\mathbb{F}_q^n$ .*

Rapellons que

$$\text{Card } B(x, r) = V_q(n, r) = \sum_{i=0}^r (q - 1)^i \binom{n}{i}$$

donc

$$V_q(n, 1) = 1 + n(q - 1).$$

La borne d'empilement de sphères de rayon 1 a la forme :

$$q^n \leq q^k V_q(n, 1) \iff q^{n-k} \leq 1 + n(q - 1),$$

et cette borne est atteinte puisque  $n - k = m$ ,  $q^m = 1 + n(q - 1)$ .





### 3.4 Énumération de poids et l'identité de MacWilliams

(voir [vLi], §3.5). Pour obtenir une information plus détaillée sur les distances entre les mots d'un code linéaire  $C$ , on utilise le **polynôme-énumérateur** de poids  $A(z) = \sum_{i=0}^n A_i z^i$ .

DÉFINITION 3.17 (a) Soit  $C$  un code linéaire, Le polynôme-énumérateur de poids est donné par

$$A(z) = \sum_{i=0}^n A_i z^i,$$

où

$$A_i := \#\{c \in C \mid w(c) = i = d(c, \underline{0})\}$$

est le nombre de mots de  $C$  de poids  $i \leq n$ ,  $A_0 = 1$ .

(b) La suite  $(A_i)_{i=0}^n$  est dit la distribution des poids.

On utilise la notation  $C^\perp$  pour le **code dual** d'un code linéaire  $C$  sur  $\mathbb{F}_q$ ,

$$C^\perp = \{(x_1, \dots, x_n) \in \mathbb{F}_q^n \mid \forall c = (c_1, \dots, c_n) \in C, x_1 c_1 + \dots + c_n x_n = 0\}.$$

EXEMPLE. Si  $C = \mathbb{F}_q^n$ , alors  $C^\perp = (\mathbb{F}_q^n)^\perp = \{0\}$ , puis

$$A_i = \binom{n}{i} (q-1)^i \Rightarrow A(z) = \sum_{i=0}^n \binom{n}{i} (q-1)^i z^i = (1 + (q-1)z)^n,$$

et  $B(z) = 1$ .

THÉORÈME 3.18 Soit  $C$  un  $[n, k, d]_q$ -code linéaire sur  $\mathbb{F}_q$  avec le polynôme-énumérateur  $A(z) = \sum_{i=0}^n A_i z^i$ , et soit  $B(z)$  le polynôme-énumérateur du code dual  $C^\perp$ . Alors

$$B(z) = q^{-k} (1 + (q-1)z)^n A\left(\frac{1-z}{1+(q-1)z}\right).$$

*Preuve* On fixe un morphisme de groupes  $\chi : \mathbb{F}_q \rightarrow \mathbb{C}^*$  non-triviale, par exemple  $\chi(x) = \exp(2i\pi \text{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(x)/p)$ , où  $q = p^r$

$$\text{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(x) = x + x^p + \cdots + x^{p^{r-1}} \in \mathbb{F}_p.$$

Puis, on considère la fonction génératrice

$$g(u, z) := \sum_{v \in \mathbb{F}_q^n} \chi(\langle u, v \rangle) z^{w(v)},$$

où

$$\langle u, v \rangle = u_1 v_1 + \cdots + u_n v_n, \quad u = u_1 + \cdots + u_n, \quad v = v_1 + \cdots + v_n \in \mathbb{F}_q^n.$$

On pose  $w(\nu) = 1$  pour tout  $\nu \in \mathbb{F}_q^*$ ,  $w(0) = 0$ , alors

$$w(v_1, \dots, v_n) = w(v_1) + \dots + w(v_n) \in \mathbb{N}.$$

Puis on montre que

$$\sum_{u \in C} g(u, z) = |C|B(z), \tag{3.2}$$

parce que

$$\sum_{u \in C} g(u, z) = \sum_{u \in C} \sum_{v \in \mathbb{F}_q^n} \chi(\langle u, v \rangle) z^{w(v)} = \sum_{v \in \mathbb{F}_q^n} z^{w(v)} \sum_{u \in C} \chi(\langle u, v \rangle).$$

Ici la somme intérieure est  $|C|$  pour  $v \in C^\perp$ , et nulle sinon, puisque  $\langle u, v \rangle$  prends alors toute valeur de  $\mathbb{F}_q$  le même nombre de fois.

D'autre part,

$$\begin{aligned}
g(u, z) &= \sum_{v=(v_1, \dots, v_n) \in \mathbb{F}_q^n} \chi(u_1 v_1) z^{w(v_1)} \cdot \dots \cdot \chi(u_n v_n) z^{w(v_n)} \\
&= \prod_{i=1}^n \sum_{v_i \in \mathbb{F}_q} \chi(u_i v_i) z^{w(v_i)} = (1 - z)^{w(u)} (1 + (q - 1)z)^{n-w(u)} \\
&= (1 + (q - 1)z)^n \left( \frac{1 - z}{1 + (q - 1)z} \right)^{w(u)} = (1 + (q - 1)z)^{n-w(u)} (1 - z)^{w(u)},
\end{aligned} \tag{3.3}$$

parce que

$$\sum_{v_i \in \mathbb{F}_q} \chi(u_i v_i) z^{w(v_i)} = \begin{cases} \sum_{v \in \mathbb{F}_q} \chi(0) z^{w(v)} = (1 + (q - 1)z), & \text{si } u_i = 0, \\ 1 + z \sum_{\nu \in \mathbb{F}_q^*} \chi(\nu) = 1 - z, & \text{si } u_i \neq 0. \end{cases}$$

La sommation des égalités (3.3) substituées dans  $\sum_{u \in C} g(u, z) = |C|B(z)$ , montre directement l'identité du théorème 3.18, puisque  $|C| = q^k$  :

$$B(z) = q^{-k} (1 + (q - 1)z)^n \sum_{i=0}^n A_i \left( \frac{1 - z}{1 + (q - 1)z} \right)^i.$$

## EXERCICES

- 3.1 Soit  $A_i := \#\{c \in C \mid w(c) = i = d(c, \underline{0})\}$  est le nombre de mots d'un code  $C$  de poids  $i \leq n$ ,  $A_0 = 1$ . Montrer que pour le code de Hamming  $C = H_m$  binaire il y a la relation suivante :

$$iA_i = \binom{n}{i-1} - A_{i-1} - (n-i+2)A_{i-2}$$

- 3.2 En déduire :

$$A'(z) = (1+z)^n - A(z) - nzA(z) + z^2A'(z), \quad A(0) = 1.$$

- 3.3 Calculer le polynôme-énumérateur de poids  $A(z) = \sum_{i=0}^n A_i z^i$  pour le code de Hamming binaire  $C = H_m$ , où  $A_i := \#\{c \in C \mid w(c) = i = d(c, \underline{0})\}$  est le nombre de mots de  $C$  de poids  $i \leq n$ ,  $A_0 = 1$ . (*Réponse* :  $A(z) = ((1+z)^n + n(1+z)^{(n-1)/2}(1-z)^{(n+1)/2})/(n+1)$ ).
- 3.4 Calculer le polynôme-énumérateur de poids  $A(z) = \sum_{i=0}^n A_i z^i$  pour le code  $C$  de répétition pûr, et pour le code d'un seul contrôle de parité. Vérifier directement l'identité de MacWilliams dans ce cas.



### 3.5 Codes cycliques

DÉFINITION 3.19 *Un code linéaire  $C \subset \mathbb{F}_q^n$  est dit cyclique si*

$$(a_0, \dots, a_{n-1}) \in C \iff (a_{n-1}, a_0, \dots, a_{n-2}) \in C$$

Pour la suite, nous supposons que  $\text{pgcd}(n, q) = 1$  et on notera  $(x^n - 1)$  l'idéal de  $\mathbb{F}_q[x]$  engendré par  $x^n - 1$ . Alors, tout élément de  $\mathbb{F}_q[x]/(x^n - 1)$  peut être représenté par des polynômes de degré inférieur à  $n$  (ou le polynôme nul), et cet anneau est ainsi isomorphe à  $\mathbb{F}_q^n$  comme  $\mathbb{F}_q$ -espace vectoriel. Un isomorphisme est donné par

$$(a_0, \dots, a_{n-1}) \longleftrightarrow a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$$

Cet isomorphisme permet de considérer les éléments de  $\mathbb{F}_q[x]/(x^n - 1)$  comme des vecteurs de  $\mathbb{F}_q^n$  ou comme des polynômes de degré  $< n$  modulo  $x^n - 1$ . La multiplication des polynômes modulo  $x^n - 1$  est introduite de manière usuelle, c'est à dire, que si  $g_1, g_2 \in \mathbb{F}_q[x]$  alors  $(g_1 \bmod (x^n - 1))(g_2 \bmod (x^n - 1)) = (g_1g_2) \bmod (x^n - 1)$ ,  
 $(g_1 \bmod (x^n - 1)) + (g_2 \bmod (x^n - 1)) = (g_1 + g_2) \bmod (x^n - 1)$ .

## 3.6 Constructions

### 3.6.1. Construction par polynôme générateur

Pour obtenir un code cyclique de dimension  $k$  et de longueur  $n$ , on peut coder les messages à transmettre (identifiés à des polynômes de degré  $\leq k - 1$ ) en les multipliant par un polynôme  $g$  donné de degré  $n - k$  diviseur de  $x^n - 1$ . La correspondance

$$(a_0, \dots, a_{n-1}) \longleftrightarrow f(x) = a_{n-1}x^{n-1} + \dots + a_1x + a_0$$

entre les vecteurs et les polynômes permet d'interpréter  $C$  comme le sous-espace suivant :

$$C = \langle 1 \cdot g(x), x \cdot g(x), x^2 \cdot g(x), \dots, x^{k-1} \cdot g(x) \rangle \subset \mathbb{F}_q[x]/(x^n - 1)$$

de l'anneau quotient

$$\mathbb{F}_q[x]/(x^n - 1).$$



THÉORÈME 3.20 *Le code linéaire  $C$  est cyclique si et seulement si  $C$  est un idéal de  $\mathbb{F}_q[x]/(x^n - 1)$ .*

PREUVE : Si  $C$  est un idéal de  $\mathbb{F}_q[x]/(x^n - 1)$ , et  $(a_0, \dots, a_{n-1}) \in C$ , alors

$$x \cdot (a_0, \dots, a_{n-1}) = (a_{n-1}, a_0, \dots, a_{n-2}) \in C.$$

Inversement, si  $C$  est cyclique, pour tout  $a(x) \in C$ ,  $xa(x) \in C$ ,  $x^2a(x) \in C$  et ainsi de suite, donc  $b(x)a(x) \in C$  et  $C$  est un idéal.

L'anneau  $\mathbb{F}_q[x]$  est principal, donc tous les idéaux de l'anneau  $\mathbb{F}_q[x]/(x^n - 1)$  sont principaux. En particulier, tout idéal non nul est engendré par un polynôme  $g(x)$  de plus bas degré qu'il contient, et  $g(x)$  divise  $x^n - 1$  :

$$C = \langle 1 \cdot g(x), x \cdot g(x), x^2 \cdot g(x), \dots, x^{k-1} \cdot g(x) \rangle,$$

### 3.6.2. Construction par polynôme correcteur

Si  $g(x) = g_0 + g_1x + \cdots + g_{n-k}x^{n-k}$ , une *matrice génératrice* du code  $C$  est

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix} \in \mathbb{M}_{k,n}(\mathbb{F}_q)$$

Les lignes de  $G$  sont, de manière évidente, linéairement indépendantes et  $rg(G) = k$ , la *dimension du code*.

PROPOSITION 3.21 Si  $h(x) = (x^n - 1)/g(x) = h_0 + \cdots + h_kx^k$ , alors

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

est une *matrice de contrôle* de  $C$ .

*Preuve* (voir [MW-S], p. 194) : En effet, soit

$$h(x) = (x^n - 1)/g(x) \tag{3.4}$$

$$= h_0 + h_1x + h_2x^2 + \dots + h_kx^k = \sum_{j=0}^k h_jx^j = \sum_{j=0}^{n-1} h_jx^j \text{ où } h_j = 0 \text{ pour } j \geq k+1.$$

Alors une *condition nécessaire* pour que

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$$

appartienne au code  $C = (g)$ , est donnée par la *congruence* suivante :

$$f(x) = g(x) \cdot u(x) \Rightarrow h(x) \cdot f(x) = h(x)g(x) \cdot u(x) \equiv 0 \text{ mod } (x^n - 1).$$

On calcule donc le produit

$$h(x) \cdot f(x) = \sum_{j=0}^k h_jx^j \sum_{i=0}^{n-1} a_ix^i = \sum_{j=0}^k \sum_{i=0}^{n-1} h_ja_ix^{i+j}, \text{ où } i+j \leq k+n-1 \leq 2n-1.$$

De plus,

$$x^{i+j} \equiv x^{i+j-n} \text{ mod } (x^n - 1), \text{ si } i+j \geq n.$$

On pose  $l = i + j$ , alors

$$\begin{aligned}
h(x) \cdot f(x) &= \sum_{j=0}^k \sum_{i=0}^{n-1} h_j a_i x^{i+j} \\
&= \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} h_{l-i} a_i x^l + \sum_{l=n}^{2n-1} \sum_{i=0}^{n-1} h_{l-i} a_i x^l \\
&\equiv \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} h_{l-i} a_i x^l + \sum_{l'=0}^{n-1} \sum_{i=0}^{n-1} h_{l'+n-i} a_i x^{l'} \pmod{x^n - 1} \\
&= \sum_{l=0}^{n-1} \left( \sum_{i=0}^{n-1} h_{l-i} a_i + \sum_{i=0}^{n-1} h_{l+n-i} a_i \right) x^l
\end{aligned} \tag{3.5}$$

(on utilise la notation  $l' = l - n$  pour  $l \geq n$ ). Puis, on observe que la seconde somme sur  $i$  dans la dernière ligne de (3.5) est nulle dès que  $l \geq k$  puisque  $l + n - i \geq k + 1$  et donc  $h_{l+n-i} = 0$  puisque le degré du polynôme  $h$  est égal à  $k$  (voir (3.4)).

Donc, pour tous les  $l = k, k + 1, \dots, n - 1$  il n'y a qu'une seule somme, et on a la condition suivante : pour tous les  $l = k, k + 1, \dots, n - 1$

$$\sum_{i=0}^{n-1} h_{l-i} a_i = 0.$$

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

$$\sum_{i=0}^{n-1} h_{l-i} a_i = 0, (l = k, k+1, \dots, n-1) \quad (3.6)$$

de plus,  $h_{l-i} = 0$  si  $l - i \geq k + 1 \iff h_{l-i} = 0$  si  $i \leq l - k - 1$ . Le système devient donc

$$\left\{ \begin{array}{ll} (l = n - 1) & 0 \cdot a_0 + \cdots + h_k \cdot a_{n-k-1} + \cdots + h_1 \cdot a_{n-2} + \cdots + h_0 \cdot a_{n-1} \\ (l = n - 2) & 0 \cdot a_0 + \cdots + h_k \cdot a_{n-k-2} + h_{k-1} \cdot a_{n-k-1} + \cdots + h_0 \cdot a_{n-2} + 0 \cdot a_{n-1} \\ \dots & \dots\dots\dots \\ (l = k) & h_k \cdot a_0 + \cdots + h_0 \cdot a_k + 0 \cdot a_{k+1} + \cdots + h_0 \cdot a_{n-2} + \cdots + 0 \cdot a_{n-1} \end{array} \right. = 0,$$

c'est à dire, la matrice du système est la matrice suivante (de rang  $n - k$ , puisque  $h_0 \neq 0$ )

$$H = \begin{pmatrix} 0 & 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 \\ 0 & 0 & \cdots & h_k & h_{k-1} & h_{k-2} & \cdots & h_0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

Comme la condition est *nécessaire*, ce système donne un sous-espace vectoriel  $C'$  de  $C$  de dimension  $n - (n - k) = k$ . Mais la dimension du code  $C$  est égale à  $k$  donc  $C' = C$ , et la condition de contrôle (3.6) est *nécessaire et suffisante*.

DÉFINITION 3.22 Soit  $C = (g(x))$  un code cyclique. Alors  $g(x)$  est appelé polynôme générateur de  $C$ , et  $h(x) = (x^n - 1)/g(x)$  est appelé polynôme correcteur de  $C$ .

Soit  $x^n - 1 = f_1(x)f_2(x) \cdots f_m(x)$  la décomposition de  $x^n - 1$  en facteurs irréductibles sur  $\mathbb{F}_q$ . Nous supposons dans cette partie que  $\text{pgcd}(n, q) = 1$ , ce qui élimine l'éventualité de facteurs multiples. Si  $f_i(x)$  est irréductible sur  $\mathbb{F}_q$ , alors  $(f_i(x))$  est un idéal maximal et  $C$  est un code cyclique maximal. On engendre tous les codes cycliques de longueur  $n$  sur  $\mathbb{F}_q$  grâce à la factorisation ci-dessus, en choisissant n'importe quel diviseur de  $x^n - 1$  (parmi les  $2^m$  diviseurs distincts de  $x^n - 1$ ) comme polynôme générateur. On a de plus les mêmes propriétés que pour les codes construits à partir de matrices. En effet, si  $C$  est cyclique,  $g(x)$  et  $h(x)$  sont des polynômes respectivement générateur et correcteur de  $C$ , alors  $v(x) \in \mathbb{F}_q[x]/(x^n - 1)$  est un mot du code  $C$  si et seulement si  $v(x)h(x) \equiv 0 \pmod{(x^n - 1)}$ . Un message  $a(x)$  est codé en  $w(x) = a(x)g(x)$ . Si on divise le message reçu  $v(x)$  par  $g(x)$  et que le reste est non nul, on sait que des erreurs sont survenues :

$$v(x) = b(x)g(x) + r(x).$$

Ceci dit, le *syndrôme*  $S(v) = r(x)$  est le *reste* de la division euclidienne. Un decodage possible de  $v(x)$  est donc  $b(x)$ .

Pour avoir un décodage standard ("de vraisemblance maximale") d'un mot  $v(x)$  on choisit parmi les polynômes

$$v(x) - \tilde{a}(x)g(x), \tilde{a}(x) = a'_0 + a'_1x + \cdots + a'_{k-1}x^{k-1}$$

un polynôme de nombre minimum des coefficients non nuls. Avec ce choix, on déclare

$$\tilde{a}(x) = a'_0 + a'_1x + \cdots + a'_{k-1}x^{k-1}$$

un mot *décodé*.



On peut obtenir la matrice génératrice *canonique* de  $C$  de la manière suivante (voir [Li-Ni], Ch. IX, §2). Soit  $\deg(g(x)) = n - k$ . Alors il existe avec unicité  $a_j(x)$  et  $r_j(x)$  avec  $\deg(r_j(x)) < n - k$  et tels que

$$x^j = a_j(x)g(x) + r_j(x) \quad \text{pour tout } j$$

Ainsi,  $x^j - r_j(x) \in C$ , ainsi que

$$g_j(x) = x^k(x^j - r_j(x)) \quad (\text{pour } n - k \leq j \leq n - 1)$$

considéré modulo  $x^n - 1$ . Les polynômes  $g_j(x)$ , pour  $n - k \leq j \leq n - 1$ , sont linéairement indépendants et forment la matrice génératrice canonique de  $C : (I_k, -R)$ , où la  $i^{\text{ème}}$  ligne de  $R$  est le vecteur des coefficients de  $r_{n-k-1+i}(x)$  :

$$\begin{aligned} g_{n-k} &= x^k(x^{n-k} - r_{n-k}(x)) \equiv 1 - x^k r_{n-k}(x) \mod (x^n - 1), \deg x^k r_{n-k}(x) < n \\ g_{n-k+1} &= x^k(x^{n-k+1} - r_{n-k}(x)) \equiv x - x^k r_{n-k+1}(x) \mod (x^n - 1), \deg x^k r_{n-k+1}(x) < n \\ &\dots\dots\dots \\ g_{n-1} &= x^k(x^{n-1} - r_{n-1}(x)) \equiv x^{k-1} - x^k r_{n-1}(x) \mod (x^n - 1), \deg x^k r_{n-1}(x) < n. \end{aligned}$$



EXEMPLE Soit  $n = 7, q = 2$ . Alors

$$x^7 - 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$$

et  $g(x) = x^3 + x^2 + 1$  engendre un code cyclique de longueur 7 et de dimension 4 de polynôme correcteur

$$h(x) = (x^7 - 1)/g(x) = (x + 1)(x^3 + x + 1) = x^4 + x^3 + x^2 + x + 1.$$

(cet exemple est disponible à l'adresse cachée :

`http ://www-fourier.ujf-grenoble.fr/~panchish/04mag-maple` dans le fichier  
`4mag-6rem.mws`)



> restart;g:=x^3+x^2+1 mod 2;

$$g := x^3 + x^2 + 1$$

> r[3]:=rem(x^3, g,x)mod 2;

$$r_3 := 1 + x^2$$

> r[4]:=rem(x^4, g,x)mod 2;

$$r_4 := x + 1 + x^2$$

> r[5]:=rem(x^5, g,x)mod 2;

$$r_5 := x + 1$$

> r[6]:=rem(x^6, g,x)mod 2;

$$r_6 := x^2 + x$$

Les matrices canoniques (voir Définition 3.4) génératrice et correctrice correspondantes sont

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

EXEMPLES 3.23 (CODES DE REED-SOLOMON) Soit  $\mathbb{F}_q$  un corps fini et  $\mathbb{F}_q^* = \langle \alpha \rangle$ . Un code de Reed-Solomon est un code sur  $\mathbb{F}_q$  de longueur  $q - 1$ , défini par le polynôme générateur  $g(x)$  des racines  $\alpha^b, \alpha^{b+1}, \dots, \alpha^{b+d'-2} \in \mathbb{F}_q^*$  :

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdot \dots \cdot (x - \alpha^{b+d'-2}), \quad d^\circ(g) = d' - 1$$

avec les facteurs linéaires dans  $\mathbb{F}_q[x]$ .

On montrera que,  $k = n - d^\circ(g) = q - 1 - d' + 1 = q - d'$ ,  $d \geq d'$ . D'autre part, par la borne de Singleton (le théorème 2.6),  $d \leq n - k + 1 = q - 1 - (q - d') + 1 = d'$ , donc le poids de  $C$  est exactement  $d'$ .

EXEMPLE. 1) Soit  $\mathbb{F}_8^* = \langle \alpha \rangle$ ,  $\alpha^3 = \alpha + 1$ ,  $g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3) = \alpha^6 + \alpha x + \alpha^6 x^2 + x^3$ . Alors  $n = 7, k = 4, d = 4$ ,

$$G = \begin{pmatrix} \alpha^6 & \alpha & \alpha^6 & 1 & 0 & 0 & 0 \\ 0 & \alpha^6 & \alpha & \alpha^6 & 1 & 0 & 0 \\ 0 & 0 & \alpha^6 & \alpha & \alpha^6 & 1 & 0 \\ 0 & 0 & 0 & \alpha^6 & \alpha & \alpha^6 & 1 \end{pmatrix}.$$

2) Soit  $\mathbb{F}_{256}^* = \langle \alpha \rangle$ ,  $g = \prod_{j=1}^{43} (x - \alpha^{11j})$ ,  $\alpha^8 = \alpha^7 + \alpha^2 + \alpha + 1$ . Alors  $n = 255, k = 223, d = 33$  (un code utilisé par la NASA).

## 4 Polynômes locateurs d'erreurs. Application au décodage

### 4.1 Construction de codes cycliques à partir des racines.

Lorsque l'on définit un code cyclique par un polynôme générateur  $g$ , tous les mots du code sont multiples de ce polynôme, et s'annulent donc sur l'ensemble des racines de  $g$ . De plus, on peut trouver une extension de  $\mathbb{F}_q$  contenant ces racines. Soient donc  $\alpha_1, \dots, \alpha_s$  des éléments d'une extension  $\mathbb{F}_{q^m}$  de  $\mathbb{F}_q$ , et  $p_i(x)$  le polynôme minimal de  $\alpha_i$  sur  $\mathbb{F}_q$ ,  $1 \leq i \leq s$ . Soit  $n \in \mathbb{N}$  tel que  $\alpha_i^n = 1$ ,  $1 \leq i \leq s$ , et soit  $g(x) = \text{ppcm}(p_1(x), \dots, p_s(x))$ . Dans ces conditions,  $g(x)$  divise  $x^n - 1$  et si  $C$  est le code de polynôme générateur  $g$ , on a


$$v(x) \in C \iff v(\alpha_i) = 0, \quad i = 1, \dots, s$$

Dans ce qui suit, nous allons étudier l'intérêt de cette méthode de construction de codes cycliques, par rapport à la construction directe par polynôme générateur.

THÉORÈME 4.1 : Soit  $C \subset \mathbb{F}_q[x]/(x^n - 1)$  un code cyclique de polynôme générateur  $g$  dont les racines sont  $\alpha_1, \dots, \alpha_{n-k}$ . Alors,  $f \in \mathbb{F}_q[x]/(x^n - 1)$  est un mot du code si et seulement si le vecteur des coefficients de  $f(f_0, \dots, f_{n-1})$  est dans le noyau de

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

PREUVE :  $f \in C$  si et seulement si  $f(\alpha_i) = 0$  pour  $1 \leq i \leq n - k$ , ce qui équivaut bien à l'assertion du théorème.

 THÉORÈME 4.2 Le code cyclique binaire  $C$  de longueur  $n = 2^m - 1$  dont le polynôme générateur est le polynôme minimal sur  $\mathbb{F}_2$  d'un élément primitif de  $\mathbb{F}_{2^m}$  est équivalent au code de Hamming binaire  $(n, n - m)$ .

PREUVE : Si  $\alpha$  est un élément primitif de  $\mathbb{F}_{2^m}$  :

$$\mathbb{F}_{2^m}^* = \langle \alpha \rangle = \{1, \alpha, \alpha^2, \dots, \alpha^{2^m-2}\}$$

alors son polynôme minimal sur  $\mathbb{F}_2$  est

$$p(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4) \cdots (x - \alpha^{2^{m-1}}) (\deg(p) = m)$$

et

$$\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$$

est une base de  $\mathbb{F}_{2^m}$  sur  $\mathbb{F}_2$ . Soit alors  $H$  la matrice dont la  $j^{\text{ème}}$  colonne est  $(c_0, \dots, c_{m-1})^t$ , avec  $\alpha^{j-1} = c_0 + c_1\alpha + \dots + c_{m-1}\alpha^{m-1}$ , et les  $c_i$  dans  $\mathbb{F}_2$ . Soit alors  $a(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1} \in \mathbb{F}_2[x]$  avec  $n = 2^m - 1$ . On a  $Ha^t = a(\alpha)$ , exprimé dans la base  $\{1, \alpha, \dots, \alpha^{m-1}\}$ . La matrice  $H$  est donc une matrice de contrôle du code engendré par  $p(x)$ , et les colonnes de  $H$  sont une permutation des représentations binaires des  $2^m - 1$  premiers entiers, qui forment une matrice de correction du code de Hamming binaire  $(n, n - m)$ , et les deux codes sont donc équivalents.

EXEMPLE. Considérons le polynôme  $p(x) = x^4 + x + 1$ . Il est primitif sur  $\mathbb{F}_2$  et une de ses racines  $\alpha$  est un élément **primitif** de  $\mathbb{F}_{16}$  :

$$\begin{aligned} \alpha^4 &= 1 + \alpha, \alpha^5 = \alpha + \alpha^2, \alpha^6 = \alpha^2 + \alpha^3, \alpha^7 = 1 + \alpha + \alpha^3, \alpha^8 = 1 + \alpha^2, \\ \alpha^9 &= \alpha + \alpha^3, \alpha^{10} = 1 + \alpha + \alpha^2, \alpha^{11} = \alpha + \alpha^2 + \alpha^3, \alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3, \\ \alpha^{13} &= 1 + \alpha^2 + \alpha^3, \alpha^{14} = 1 + \alpha^3, \alpha^{15} = 1. \end{aligned}$$

Ecrivons  $H$  dont la  $j^{\text{ème}}$  colonne est  $\alpha^{j-1}$  exprimé dans la base  $\{1, \alpha, \alpha^2, \alpha^3\}$ ,  $0 \leq j \leq 14$ .  
Il vient

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

Si alors  $a(x) = a_0 + \cdots + a_{10}x^{10}$  est le message à transmettre, il sera codé en  $w(x) = a(x)(x^4 + x + 1)$ . Supposons qu'une erreur survienne au cours de la transmission, le message reçu est alors  $v(x) = w(x) + x^{e-1}$ . Son syndrome est  $S(v) = v(\alpha) = w(\alpha) + \alpha^{e-1} = \alpha^{e-1}$ , et on sait qu'une erreur est survenue en  $e^{\text{ème}}$  position.

## 4.2 Exemples : codes de Golay



Les calculs suivants sont disponibles à l'adresse cachée :  
`http ://www-fourier.ujf-grenoble.fr/~panchish/04mag-maple` dans le fichier  
`4mag-7cycl-gol.mws` :

### 4.2.1. Code $G_{23}$

On considère le groupe cyclique  $\mathbb{F}_{2^{11}}^*$  d'ordre  $2^{11} - 1 = 23 \cdot 89$ . Soit  $\alpha \in \mathbb{F}_{2^{11}}^*$  une racine primitive de degré 23. On pose

$$G_{23} = \left\{ x = (x_0, \dots, x_{22}) \in \mathbb{F}_2^{23} \mid \sum_{i=0}^{22} x_i \alpha^i = 0 \right\} \subset \mathbb{F}_2[x]/(x^{23} - 1).$$

On a  $n=23$ ,  $q = 2$ ,

$$x^{23} - 1 = x^{23} + 1 = (x + 1)g_0(x)g_1(x) = \\ (x + 1)(x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)$$



où

$$\begin{aligned}
g_0(x) &= x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1 \\
&= \prod_{i \in I} (x - \alpha^i), I = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\} \\
g_1(x) &= x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1 \\
&= \prod_{j \in J} (x - \alpha^j), J = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\}
\end{aligned}$$

REMARQUE. L'ensemble

$$I = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$$

coïncide avec l'ensemble des **résidues quadratiques** modulo 23, et l'ensemble complémentaire

$$J = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\}$$

coïncide avec l'ensemble des **non-résidues quadratiques** modulo 23.

**L'application de Frobenius**  $\alpha^k \mapsto \alpha^{2k}$  laisse  $I$  et  $J$  stable puisque  $\left(\frac{2}{23}\right) = 1$ , et l'application  $\alpha^k \mapsto \alpha^{-k}$  échange les ensembles  $I$  et  $J$  puisque  $\left(\frac{-1}{23}\right) = -1$ , grâce à la loi de

réciprocité quadratique de Gauss : pour les nombres premiers positifs impairs  $p, q$  on a

$$\left(\frac{p}{q}\right) \left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)}{2} \frac{(q-1)}{2}},$$

et on a les deux compléments suivants de cette loi :

$$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}, \quad \left(\frac{-1}{p}\right) = (-1)^{(p-1)/2},$$

DÉFINITION 4.3 *Code de Golay  $G_{23}$  est un sous-espace vectoriel ( $g_0$ ) de dimension 12 dans le quotient*

$$\mathbb{F}_2[x]/(x^{23} - 1)$$

*vu comme un espace vectoriel de dimension 23 sur  $\mathbb{F}_2$  avec le polynôme générateur*

$$g_0(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

*et avec le polynome de contrôle*

$$h(x) = (x + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1) = (x + 1)g_1(x)$$

*C'est un  $[23, 12, 7]_2$ -code.*

```
> restart ;
```

```
> with(linalg) :
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

```
> Factor(x^23+1) mod 2;
```

$$(x + 1)(x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)$$

```
> g:=x^11+x^10+x^6+x^5+x^4+x^2+1;irreduc(g) mod 2;
```

$$g := x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

*true*

```
> alias(alpha = RootOf(g)) ;
```

$\alpha$

```
> Factor(g,alpha) mod 2;
```

$$(x + \alpha^9)(x + \alpha^6)(x + \alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha)(x + \alpha^4)$$
$$(x + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + 1)(x + \alpha^{10} + \alpha^8 + \alpha^6 + \alpha^3 + \alpha + 1)$$
$$(x + \alpha^{10} + \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1)(x + \alpha)(x + \alpha^2)(x + \alpha^8)(x + \alpha^3)$$

```
> for i from 0 to 23 do
```

```
> if Eval(g, x=alpha^i) mod 2 = 0 then Expand (alpha^i) mod 2;
```

```
> print('i'=i, alpha^i=Expand (alpha^i) mod 2, 'g'(alpha^i)=0) fi
```

```
> od ;
```

$$i = 1, \alpha = \alpha, g(\alpha) = 0$$
$$i = 2, \alpha^2 = \alpha^2, g(\alpha^2) = 0$$
$$i = 3, \alpha^3 = \alpha^3, g(\alpha^3) = 0$$
$$i = 4, \alpha^4 = \alpha^4, g(\alpha^4) = 0$$
$$i = 6, \alpha^6 = \alpha^6, g(\alpha^6) = 0$$
$$i = 8, \alpha^8 = \alpha^8, g(\alpha^8) = 0$$

$$i = 9, \alpha^9 = \alpha^9, g(\alpha^9) = 0$$

$$i = 12, \alpha^{12} = \alpha^{10} + \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1, g(\alpha^{12}) = 0$$

$$i = 13, \alpha^{13} = \alpha^{10} + \alpha^8 + \alpha^6 + \alpha^3 + \alpha + 1, g(\alpha^{13}) = 0$$

$$i = 16, \alpha^{16} = \alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha, g(\alpha^{16}) = 0$$

$$i = 18, \alpha^{18} = \alpha^8 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + 1, g(\alpha^{18}) = 0$$

```
> for i from 0 to 23 do
>   if Eval(g, x=alpha^(-i)) mod 2 = 0 then Expand (alpha^(-i)) mod 2;
>   print('i'=i, alpha^i)=Expand (alpha^i) mod 2, 'g'(alpha^(-i))=0) fi
> od ;
```

$$i = 5, \alpha^5 = \alpha^5, g\left(\frac{1}{\alpha^5}\right) = 0$$

$$i = 7, \alpha^7 = \alpha^7, g\left(\frac{1}{\alpha^7}\right) = 0$$

$$i = 10, \alpha^{10} = \alpha^{10}, g\left(\frac{1}{\alpha^{10}}\right) = 0$$

$$i = 11, \alpha^{11} = \alpha^{10} + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1, g\left(\frac{1}{\alpha^{11}}\right) = 0$$

$$i = 14, \alpha^{14} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1, g\left(\frac{1}{\alpha^{14}}\right) = 0$$

$$i = 15, \alpha^{15} = \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1, g\left(\frac{1}{\alpha^{15}}\right) = 0$$

$$i = 17, \alpha^{17} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^3 + \alpha^2, g\left(\frac{1}{\alpha^{17}}\right) = 0$$

$$i = 19, \alpha^{19} = \alpha^9 + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha, g\left(\frac{1}{\alpha^{19}}\right) = 0$$

$$i = 20, \alpha^{20} = \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha^2, g\left(\frac{1}{\alpha^{20}}\right) = 0$$

$$i = 21, \alpha^{21} = \alpha^9 + \alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1, g\left(\frac{1}{\alpha^{21}}\right) = 0$$

$$i = 22, \alpha^{22} = \alpha^{10} + \alpha^9 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha, g\left(\frac{1}{\alpha^{22}}\right) = 0$$

$$> \text{g}[1] := x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1; \text{Factor}(\text{g}[1], \alpha) \bmod 2;$$

$$g_1 := x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

$$\begin{aligned}
& (x + \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^3 + \alpha^2) (x + \alpha^{10} + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1) (x + \alpha^7) \\
& (x + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1) (x + \alpha^{10}) (x + \alpha^5) (x + \alpha^{10} + \alpha^9 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha) \\
& (x + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha) (x + \alpha^9 + \alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1) \\
& (x + \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1) (x + \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha^2)
\end{aligned}$$

**Code de Golay**  $G_{23} = (g)$  est un sous-espace vectoriel de dimension 12 dans le quotient

$$\mathbb{F}_2[x]/(x^{23} - 1)$$

(vu comme un espace vectoriel de dimension 23 sur  $\mathbb{F}_2$ ) avec le polynôme générateur

$$g := x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

et avec le polynome de contrôle  $h = (x + 1) (x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)$  ,  
 $h = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1$ .

C'est un  $[23, 12, 7]_2$ -code.

```

> for i from 0 to 23 do
>   if Eval(g[1], x=alpha^i) mod 2 = 0 then Expand (alpha^i) mod 2;
>   print('i'=i, alpha^i=Expand (alpha^i) mod 2, 'g[1]'(alpha^i)=0) fi
> od ;

```

$$i = 5, \alpha^5 = \alpha^5, g_1(\alpha^5) = 0$$

$$i = 7, \alpha^7 = \alpha^7, g_1(\alpha^7) = 0$$

$$i = 10, \alpha^{10} = \alpha^{10}, g_1(\alpha^{10}) = 0$$

$$i = 11, \alpha^{11} = \alpha^{10} + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1, g_1(\alpha^{11}) = 0$$

$$i = 14, \alpha^{14} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1, g_1(\alpha^{14}) = 0$$

$$i = 15, \alpha^{15} = \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1, g_1(\alpha^{15}) = 0$$

$$i = 17, \alpha^{17} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^3 + \alpha^2, g_1(\alpha^{17}) = 0$$

$$i = 19, \alpha^{19} = \alpha^9 + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha, g_1(\alpha^{19}) = 0$$

$$i = 20, \alpha^{20} = \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha^2, g_1(\alpha^{20}) = 0$$

$$i = 21, \alpha^{21} = \alpha^9 + \alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1, g_1(\alpha^{21}) = 0$$

$$i = 22, \alpha^{22} = \alpha^{10} + \alpha^9 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha, g_1(\alpha^{22}) = 0$$

$$> (x^{11}+x^{10}+x^6+x^5+x^4+x^2+1)*(x+1)*(x^{11}+x^9+x^7+x^6+x^5+x+1); g:=x^1$$

$$> 1+x^{10}+x^6+x^5+x^4+x^2+1;$$

$$(x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1)(x + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)$$

$$g := x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$



```

> G:= matrix(12, 23,
> [[1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0],
> [0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0],
> [0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0],
> [0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0],
> [0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0],
> [0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0],
> [0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0],
> [0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0],
> [0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0],
> [0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0],
> [0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1]]) ;

```

```
> transpose(G);
```

123

```

> (x^11+x^10+x^6+x^5+x^4+x^2+1)*(x+1)*(x^11+x^9+x^7+x^6+x^5+x+1);
(x11 + x10 + x6 + x5 + x4 + x2 + 1)(x + 1)(x11 + x9 + x7 + x6 + x5 + x + 1)
> 'h'=(x+1)*(x^11+x^9+x^7+x^6+x^5+x+1);
> h:=Expand((x+1)*(x^11+x^9+x^7+x^6+x^5+x+1)) mod 2;

```

$$h = (x + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)$$

$$h := x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1$$

```

> 'h'=(x+1)*(x^11+x^9+x^7+x^6+x^5+x+1); 'h'=sort(h,x);

```

$$h = (x + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)$$

$$h = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1$$

```

> Expand(h*g) mod 2;

```

$$1 + x^{23}$$

[illegible]

```
> K:=multiply(H,transpose(G)) ;
```

$$K := \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 2 \\ 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 2 & 2 \\ 2 & 2 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 2 & 2 & 2 \\ 2 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 0 \\ 4 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 2 \\ 4 & 4 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 2 & 0 \end{bmatrix}$$

Les termes de la matrice obtenue ne sont pas « réduits » à leur forme canonique dans  $\text{GF}(2^{11}) = F_2[\alpha]$ . Pour obtenir la réduction, on utilise :

```
> map(item -> Expand(item) mod 2, K) ;
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\%1 := [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

**Une autre matrice de contrôle :**

on considère la matrice dont la  $j$ -me colonne est formée par les coordonnées de  $\alpha^{j-1}$  ( $j = 1, 2, \dots, 23$ ) dans la base

$$\langle 1, \alpha, \alpha^2, \dots, \alpha^{10} \rangle$$

$$\text{GF}(2^{11}) = \mathbb{F}_2[\alpha].$$

```
> for j from 1 to 23 do print(alpha^(j-1)=Expand(alpha^(j-1)) mod 2) ;
> od;
```

$$1 = 1$$

$$\alpha = \alpha$$

$$\alpha^2 = \alpha^2$$

$$\alpha^3 = \alpha^3$$

$$\alpha^4 = \alpha^4$$

$$\alpha^5 = \alpha^5$$

$$\alpha^6 = \alpha^6$$

$$\alpha^7 = \alpha^7$$

$$\alpha^8 = \alpha^8$$

$$\alpha^9 = \alpha^9$$

$$\alpha^{10} = \alpha^{10}$$

$$\alpha^{11} = \alpha^{10} + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1$$

$$\alpha^{12} = \alpha^{10} + \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$$

$$\alpha^{13} = \alpha^{10} + \alpha^8 + \alpha^6 + \alpha^3 + \alpha + 1$$

$$\alpha^{14} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1$$

$$\alpha^{15} = \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1$$



$$\begin{aligned}
\alpha^{16} &= \alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha \\
\alpha^{17} &= \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^3 + \alpha^2 \\
\alpha^{18} &= \alpha^8 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + 1 \\
\alpha^{19} &= \alpha^9 + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha \\
\alpha^{20} &= \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha^2 \\
\alpha^{21} &= \alpha^9 + \alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1 \\
\alpha^{22} &= \alpha^{10} + \alpha^9 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha
\end{aligned}$$

```

> H1:= matrix(11, 23,
> [[1,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,1,0,0,1,0],
> [0,1,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,1,0,0,1],
> [0,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,1,1,1,0,1,1,0],
> [0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,1,1,0,1,1],
> [0,0,0,0,1,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,1,1,1],
> [0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,1,1,1,0,1,0,1,0,1],
> [0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,1,0,1,1,1,1,0,0,0],
> [0,0,0,0,0,0,0,1,0,0,0,0,1,0,1,1,0,1,1,1,1,0,0],
> [0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,1,1,0,1,1,1,1,0],
> [0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,1,1,0,1,1,1,1],
> [0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,1,0,0,1,0,1]]) ;

```

$$H1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

> K1:=multiply(H1,transpose(G)) ;

$$K1 := \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 \\ 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 \\ 2 & 2 & 2 & 0 & 0 & 2 & 4 & 4 & 2 & 2 & 4 & 4 \\ 0 & 2 & 2 & 2 & 0 & 0 & 2 & 4 & 4 & 2 & 2 & 4 \\ 2 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 \\ 2 & 2 & 0 & 2 & 2 & 4 & 2 & 2 & 2 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 2 & 4 \\ 0 & 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 2 \\ 0 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Les termes de la matrice obtenue ne sont pas « réduits » à leur forme canonique dans  $\text{GF}(2^{11}) = F_2[\alpha]$ . Pour obtenir la réduction.

```
> map(item -> Expand(item) mod 2, K1) ;
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\%1 := [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

#### 4.2.2. Code $G_{24}$

de type  $[24, 12, 8]_2$  est un 3-correcteur ; il est obtenu en ajoutant un contrôle total de parité à la matrice  $H$  de code  $G_{23}$ .

Ce code est bien adapté à la transmission de 4096 nuances de couleur.

### 4.2.3. Code $G_{11}$

On considère le groupe cyclique  $\mathbb{F}_{3^5}^*$  d'ordre  $3^5 - 1 = 11 \cdot 22$ . Soit  $\alpha \in \mathbb{F}_{3^5}^*$  une racine primitive de degré 11. On pose

$$G_{11} = \left\{ x = (x_0, \dots, x_{10}) \in \mathbb{F}_3^{11} \mid \sum_{i=0}^{10} x_i \alpha^i = 0 \right\} \subset \mathbb{F}_3[x]/(x^{11} - 1).$$

On a  $n=11$ ,  $q = 3$ ,

$$X^{11} - 1 = X^{11} + 2 = (x+2)g_0(x)g_1(x) = (x+2)(x^5 + 2x^3 + x^2 + 2x + 2)(x^5 + x^4 + 2x^3 + x^2 + 2)$$

où

$$g_0(x) = x^5 + 2x^3 + x^2 + 2x + 2 = \prod_{i \in I} (x - \alpha^i), I = \{1, 3, 4, 5, 9\}$$

$$g_1(x) = x^5 + x^4 + 2x^3 + x^2 + 2 = \prod_{j \in J} (x - \alpha^j), J = \{2, 6, 7, 8, 10\}$$

REMARQUE. L'ensemble

$$I = \{1, 3, 4, 5, 9\}$$

coïncide avec l'ensemble des **résidues quadratiques** modulo 11, et l'ensemble complémentaire

$$J = \{2, 6, 7, 8, 10\}$$

coïncide avec l'ensemble des **non-résidues quadratiques** modulo 11.

**L'application de Frobenius**  $\alpha^k \mapsto \alpha^{3k}$  laisse  $I$  et  $J$  stable puisque  $\left(\frac{3}{11}\right) = 1$ , et l'application  $\alpha^k \mapsto \alpha^{-k}$  échange les ensembles  $I$  et  $J$  puisque  $\left(\frac{-1}{11}\right) = -1$ , grâce à la loi de réciprocité quadratique de Gauss : pour les nombres premiers positifs impairs  $p, q$  on a

$$\left(\frac{p}{q}\right) \left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)}{2} \frac{(q-1)}{2}},$$

et on a les deux compléments suivants de cette loi :

$$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}, \quad \left(\frac{-1}{p}\right) = (-1)^{(p-1)/2},$$

DÉFINITION 4.4 *Code de Golay  $G_{11}$  est un sous-espace vectoriel ( $g_0$ ) de dimension 6 dans le quotient*

$$\mathbb{F}_3[x]/(x^{11} - 1)$$

*vu comme un espace vectoriel de dimension 11 sur  $\mathbb{F}_3$  avec le polynôme générateur*

$$g_0(x) = g_0(x) = x^5 + 2x^3 + x^2 + 2x + 2$$

*et avec le polynome de contrôle*

$$h(x) = (x + 2)(x^5 + x^4 + 2x^3 + x^2 + 2) = (x + 1)g_1(x)$$

*C'est un  $[11, 6, 5]_3$ -code.*

```
> q:=3;
```

```
q := 3
```

```
> restart ;
```

```
> with(linalg) :
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

> Factor(x<sup>11</sup>-1) mod 3;

$$(x + 2)(x^5 + 2x^3 + x^2 + 2x + 2)(x^5 + x^4 + 2x^3 + x^2 + 2)$$

> g:=x<sup>5</sup>+2\*x<sup>3</sup>+x<sup>2</sup>+2\*x+2;irreduc(g) mod 3;

$$g := x^5 + 2x^3 + x^2 + 2x + 2$$

*true*

> alias(alpha = RootOf(g)) ;

$\alpha$

> Factor(g,alpha) mod 3;

$$(x + 2\alpha)(x + 2\alpha^3 + \alpha^2 + 2\alpha + 2)(x + 2\alpha^4)(x + \alpha^4 + 2\alpha^3 + 2\alpha^2 + 2\alpha + 1)(x + 2\alpha^3)$$

> for i from 0 to 11 do

> if Eval(g, x=alpha<sup>i</sup>) mod 3 = 0 then Expand (alpha<sup>i</sup>) mod 3;

> print('i'=i, alpha<sup>i</sup>=Expand (alpha<sup>i</sup>) mod 3, 'g'(alpha<sup>i</sup>)=0) fi

> od ;

$$i = 1, \alpha = \alpha, g(\alpha) = 0$$



$$i = 3, \alpha^3 = \alpha^3, g(\alpha^3) = 0$$

$$i = 4, \alpha^4 = \alpha^4, g(\alpha^4) = 0$$

$$i = 5, \alpha^5 = \alpha^3 + 2\alpha^2 + \alpha + 1, g(\alpha^5) = 0$$

$$i = 9, \alpha^9 = 2\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 2, g(\alpha^9) = 0$$

```
> for i from 0 to 11 do
> if Eval(g, x=alpha^(-i)) mod 3 = 0 then Expand (alpha^(-i)) mod 3;
> print('i'=i, alpha^i)=Expand (alpha^i) mod 3, 'g'(alpha^(-i))=0) fi
> od ;
```

$$i = 2, \alpha^2 = \alpha^2, g\left(\frac{1}{\alpha^2}\right) = 0$$

$$i = 6, \alpha^6 = \alpha^4 + 2\alpha^3 + \alpha^2 + \alpha, g\left(\frac{1}{\alpha^6}\right) = 0$$

$$i = 7, \alpha^7 = 2\alpha^4 + 2\alpha^3 + \alpha + 1, g\left(\frac{1}{\alpha^7}\right) = 0$$

$$i = 8, \alpha^8 = 2\alpha^4 + 2\alpha^3 + 2\alpha^2 + 2, g\left(\frac{1}{\alpha^8}\right) = 0$$

$$i = 10, \alpha^{10} = \alpha^4 + 2\alpha^2 + \alpha + 2, g\left(\frac{1}{\alpha^{10}}\right) = 0$$

```

> (x+2)*(x^5+2*x^3+x^2+2*x+2)*(x^5+x^4+2*x^3+x^2+2);
> 'h'=(x+2)*(x^5+x^4+2*x^3+x^2+2);
> h:= sort(Expand((x+2)*(x^5+x^4+2*x^3+x^2+2)) mod 3,x);

```

$$(x+2)(x^5+2x^3+x^2+2x+2)(x^5+x^4+2x^3+x^2+2)$$

$$h = (x+2)(x^5+x^4+2x^3+x^2+2)$$

$$h := x^6 + x^4 + 2x^3 + 2x^2 + 2x + 1$$

Code de Golay  $C_{11}$  = (g) est un sous-espace vectoriel de dimension 6 dans le quotient  $Z_3[X]/(x^{11}-1)$  vu comme un espace vectoriel de dimension 11 sur  $Z_3$  avec le polynôme générateur

$$g := x^5 + 2x^3 + x^2 + 2x + 2$$

et avec le polynome de contrôle  $h = (x+2)(x^5+x^4+2x^3+x^2+2)$  ,

$$h := x^6 + x^4 + 2x^3 + 2x^2 + 2x + 1.$$

C'est un  $[11, 6, 5]_3$  -code.

%% %%%%%%%%%

```

> G:= matrix(6, 11,
> [[2,2,1,2,0,1,0,0,0,0,0],
> [0,2,2,1,2,0,1,0,0,0,0],
> [0,0,2,2,1,2,0,1,0,0,0],
> [0,0,0,2,2,1,2,0,1,0,0],
> [0,0,0,0,2,2,1,2,0,1,0],
> [0,0,0,0,0,2,2,1,2,0,1]]);

```

$$G := \begin{bmatrix} 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 \end{bmatrix}$$

```

> 'h' = x^6+x^4+2*x^3+2*x^2+2*x+1;

```

$$h = x^6 + x^4 + 2x^3 + 2x^2 + 2x + 1$$

```

> H:= matrix(5, 11,
> [[0,0,0,0,1,0,1,2,2,2,1],
> [0,0,0,1,0,1,2,2,2,1,0],
> [0,0,1,0,1,2,2,2,1,0,0],
> [0,1,0,1,2,2,2,1,0,0,0],
> [1,0,1,2,2,2,1,0,0,0,0]]);

```

$$H := \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 2 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 2 & 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 2 & 2 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```

> sort(Expand(h*g) mod 3,x);

```

$$x^{11} + 2$$

```

> K:=multiply(H,transpose(G)) ;

```

$$K := \begin{bmatrix} 0 & 3 & 3 & 6 & 9 & 9 \\ 3 & 3 & 6 & 9 & 9 & 12 \\ 3 & 6 & 9 & 9 & 12 & 12 \\ 6 & 9 & 9 & 12 & 12 & 9 \\ 9 & 9 & 12 & 12 & 9 & 6 \end{bmatrix}$$

Les termes de la matrice obtenue ne sont pas « réduits » à leur forme canonique dans  $\text{GF}(3^5) = F_3[\alpha]$ . Pour obtenir la réduction.



```
> map(item -> Expand(item) mod 3, K) ;
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



#### 4.2.4. Code $G_{12}$



de type  $[12, 6, 6]_3$  est un 2-correcteur ; il est obtenu en ajoutant une ligne de contrôle total à la matrice  $H$  de code  $G_{11}$ .

EXERCICE. (a) Calculer le volume de la boule de Hamming  $V_q(n, t)$ ,  $t = 1, 2, 3$ .

(b) Montrer que la borne de Hamming est atteinte pour les codes  $G_{23}$  et  $G_{11}$  donc on a un empiement parfait de sphères.

REMARQUE 4.5 *On peut montrer que les codes  $G_{23}$  ,  $G_{11}$  et  $C(m, q)$  (voir Section 3.16) sont tous les codes parfaits.*



### 4.3 Locateurs d'erreurs

Pour détecter et corriger les erreurs, nous avons vu qu'il fallait déterminer le **syndrome** du message reçu. Dans le cas de certains codes cycliques, ce vecteur de longueur  $n - k$  peut être remplacé par un objet plus léger ayant les mêmes possibilités. En effet, soit  $\alpha$  une racine primitive  $n^{\text{ème}}$  de l'unité, contenue dans  $\mathbb{F}_{2^m}$ , <sup>\*)</sup> et considérons le code généré par  $g(x)$ , le polynôme minimal de  $\alpha$  sur  $\mathbb{F}_2$ . Soit par exemple

Ici  $q = 2$

$$H = (1 \ \alpha \ \alpha^2 \ \dots \ \alpha^{n-1})$$

et

$$S(v) = Hv^t = v(\alpha).$$

Soit  $w$  le message émis, posons  $e^{(j)}(x) = x^{j-1}$ ,  $1 \leq j \leq n$ , et plaçons-nous dans le cas d'une erreur simple. Il existe donc  $j$ ,  $1 \leq j \leq n$ , tel que  $v = w + e^{(j)}$  donc

$$S(v) = v(\alpha) = w(\alpha) + e^{(j)}(\alpha) = \alpha^{j-1}$$

$e^{(j)}(\alpha)$  est appelé locateur d'erreur. En effet, comme on a  $e^{(j)}(\alpha) \neq e^{(i)}(\alpha)$  pour  $i \neq j$ ,  $1 \leq i, j \leq n$ ,  $\alpha^{j-1}$  détermine la position de l'erreur. <sup>\*)</sup>

Faire en e  
le cas q

## 4.4 Décodage des codes cycliques

Soit  $C$  un code cyclique de polynôme générateur  $g$  et soient  $w$  le message émis et  $v$  le message reçu. Supposons qu'au plus  $t$  erreurs se produisent, et on note  $t'$  le nombre exacte d'erreurs,  $t' \leq t$ . On utilise maintenant plusieurs racines  $\alpha_1, \dots, \alpha_r$  de  $g$  pour déterminer le polynôme d'erreur  $e(x) = \sum_{i=1}^{t'} c_i x^{a_i}$ , ou les  $a_i$  sont tous différentes dans  $0, 1, \dots, n-1$ , et  $c_i$  sont les valeurs d'erreurs. On suppose dans cette section en simplifiant que parmi les racines de  $g$  il y a une racine primitive  $n^{\text{ème}}$  de l'unité  $\alpha$ , contenue dans  $\mathbb{F}_{q^m}$ , disons  $\alpha = \alpha_1$ , et considérons le code engendré par  $g(x)$ , le polynôme minimal de  $\alpha$  sur  $\mathbb{F}_q$ . On utilise les notations

$$\alpha_j = \alpha^{b_j}, \quad S_{b_j} = e(\alpha^{b_j}) = \sum_{i=1}^{t'} c_i \alpha^{b_j a_i}, \quad (4.1)$$

Posons alors

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

et

$$S(v) = H v^t = (S_{b_1}, S_{b_2}, \dots, S_{b_r})^t.$$

Soit  $w$  le message émis,  $v = w + e$  donc

$$S_{b_j} = v(\alpha_j) = w(\alpha^{b_j}) + e(\alpha^{b_j}) = e(\alpha^{b_j}).$$

On pose

$$\eta_i = \alpha^{a_i}, \text{ (où } e(x) = \sum_{i=1}^{t'} c_i x^{a_i} \text{)}$$

pour la racine  $\alpha$  fixée, et on introduit le *polynôme locateur d'erreur*

$$s(x) = \prod_{i=1}^{t'} (1 - \eta_i x) = \sum_{i=0}^{t'} \tau_{t'-i} x^i, \text{ avec } \eta_i = \alpha^{a_i} \quad (4.2)$$

Les racines de  $s(x)$  sont les  $\eta_i^{-1}$ , c'est à dire les  $\alpha^{-a_i}$ .

Soit alors l'identité polynomiale

$$\prod_{i=1}^{t'} (\eta_j - x) = \sum_{i=0}^{t'} (-1)^i \sigma_{t'-i} x^i = \sigma_{t'} - \sigma_{t'-1} x + \cdots + (-1)^{t'} \sigma_0 x^{t'}. \quad (4.3)$$

Les coefficients  $\sigma_i$  sont donc  $\sigma_0 = 1$  et pour  $1 \leq i \leq t'$ , les  $\sigma_i$  sont les polynômes élémentaires symétriques en  $\eta_1, \dots, \eta_{t'}$ , et  $\tau_i = (-1)^i \sigma_i$ . En remplaçant  $x$  par  $\eta_i$  dans (4.3), il vient

$$(-1)^{t'} \sigma_{t'} + (-1)^{t'-1} \sigma_{t'-1} \eta_i + \cdots - \sigma_1 \eta_i^{t'-1} + \eta_i^{t'} = 0 \quad (4.4)$$



pour tout  $1 \leq j \leq t'$ . En multipliant par  $c_i \eta_i^b$  et en sommant pour  $1 \leq i \leq t'$ , il vient

$$(-1)^{t'} \sigma_{t'} S_b + (-1)^{t'-1} \sigma_{t'-1} S_{b+1} + \cdots - \sigma_1 S_{b+t'-1} + S_{b+t'} = 0 \text{ (pour tout } b). \quad (4.5)$$

RÉSUMÉ DE CETTE MÉTHODE : (on note  $\tau_i = (-1)^i \sigma_i$ ).

- 1<sup>ère</sup> étape : Déterminer le syndrome

$$S(v) = (S_{b_1}, S_{b_2}, \dots, S_{b_r})^t.$$

Soit

$$S_{b_j} = \sum_{i=0}^{t'} c_i \alpha^{a_i b_j} = \sum_{i=0}^{t'} c_i \eta_i^{b_j} = \sum_{i=0}^{n-1} v_i \alpha_i^{i b_j} \quad (4.6)$$

- 2<sup>ème</sup> étape : Trouver les coefficients  $\tau_{t'-i}$  du polynôme locateur d'erreurs (4.2). Ici on utilise des relations entre  $S_{b_j}$ ,  $\eta_i$  et les syndrômes  $S_{b_j}$ , par exemple (4.4), (4.5) (voir Section 5).

- 3<sup>ème</sup> étape : Chercher les racines de  $s(x)$  en testant les différentes puissances de  $\alpha$  pour déterminer les locateurs d'erreurs  $\eta_i$

- 4<sup>ème</sup> étape : Remplacer les  $\eta_i$  par leurs valeurs dans l'expression des  $S_j$  pour déterminer les valeurs des erreurs  $c_i$  (cas non binaire seulement). On obtient ainsi le vecteur erreur  $e$  et on peut décoder  $w = v - e$ ,  $u = (v - e)/g$  (le mot d'information).

#### 4.4.1. Exemples de decodage des codes cycliques

Soit  $\alpha$  un élément primitif de  $\mathbb{F}_{16}$  racine du polynôme  $x^4 + x + 1$  sur  $\mathbb{F}_2$ ,  $m^{(i)}(x)$  le polynôme minimal unitaire de  $\alpha^i$ . Alors

$$m^{(1)}(x) = m^{(2)}(x) = m^{(4)}(x) = m^{(8)}(x) = x^4 + x + 1$$

$$m^{(3)}(x) = m^{(6)}(x) = m^{(12)}(x) = x^4 + x^3 + x^2 + x + 1$$

Ainsi, un code de polynôme générateur

$$g(x) = m^{(1)}(x)m^{(3)}(x) = 1 + x^4 + x^6 + x^7 + x^8$$

est un code cyclique dont les paramètres sont  $d = 5, q = 2, n = 15$ . Sa dimension est  $k = 7$  et un polynôme de correction est

$$h(x) = (x^{15} - 1)/g(x) = 1 + x^4 + x^6 + x^7$$

On construit une matrice génératrice  $G$  dont la  $i^{\text{ème}}$  ligne est le vecteur  $x^{i-1}g(x)$ ,  $1 \leq i \leq k$ . On obtient

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

La distance du code est 5 donc ce code corrige deux erreurs. Pour cela on considère les composantes

$$S_1 = \sum_{i=0}^{14} v_i \alpha^i, \quad S_3 = \sum_{i=0}^{14} v_i \alpha^{3i}.$$

du syndrome  $S(v) = Hv^t$ . Alors  $v \in C$  si et seulement si  $S(v) = Hv^t = 0$ . Supposons que le vecteur reçu  $v = (v_0, \dots, v_{14})$  contient deux erreurs. Par exemple  $e(X) = X^{a_1} + X^{a_2}$ , où  $0 \leq a_1, a_2 \leq 14$ ,  $a_1 \neq a_2$ . Alors

$$S_1 = \alpha^{a_1} + \alpha^{a_2}, \quad S_3 = \alpha^{3a_1} + \alpha^{3a_2}.$$

Soit  $\eta_1 = \alpha^{a_1}$ ,  $\eta_2 = \alpha^{a_2}$  les locateurs des erreurs, alors

$$S_1 = \eta_1 + \eta_2, \quad S_3 = \eta_1^3 + \eta_2^3 \implies S_1^3 = (\eta_1 + \eta_2)^3 = \eta_1^3 + \eta_2^3 + 3\eta_1\eta_2(\eta_1 + \eta_2).$$

Ceci implique

$$(X - \eta_1)(X - \eta_1) = X^2 - S_1X + \eta_1\eta_2 \implies S_1(X - \eta_1)(X - \eta_1) = S_1X^2 - S_1^2X + S_1^3 + S_3$$

et la substitution  $X = \eta_1$  donne  $S_3 = S_1^3 + S_1^2\eta_1 + S_1\eta_1^2$ ,

$$1 + S_1\eta_1^{-1} + (S_1^2 + S_3S_1^{-1})\eta_1^{-2} = 0.$$

$$1 + S_1\eta_2^{-1} + (S_1^2 + S_3S_1^{-1})\eta_2^{-2} = 0.$$

S'il y a deux erreurs,  $\eta_1^{-1}$  et  $\eta_2^{-1}$  sont des racines différentes du polynôme locateur d'erreurs :

$$s(X) = 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2.$$

S'il n'y a qu'une seule erreur,  $S_1 = \eta_1$ ,  $S_3 = \eta_1^3$  donc  $S_1^3 + S_3 = 0$ , et on a

$$s(X) = 1 + S_1X.$$

S'il n'y a pas d'erreurs,  $S_1 = S_3 = 0$ , et on a reçu message correct  $w$ . Si  $S_1 \neq 0$  et  $S_1^3 + S_3 = 0$ , le polynôme  $s(X)$  a une seule racine dans  $\mathbb{F}_{16}$ .

**L'exemple suivant est disponible à l'adresse cachée :**

`http ://www-fourier.ujf-grenoble.fr/~panchish/04mag-maple`

dans le fichier 4mag-08cycl.mws :

Soit par exemple le mot reçu a la forme

$$v = (1001110000000000) \implies v(x) = 1 + x^3 + x^4 + x^5.$$

Alors  $S(v) = (S_1(v), S_3(v))$  est donné par

$$S_1 = 1 + \alpha^3 + \alpha^4 + \alpha^5 = \alpha^2 + \alpha^3,$$

$$S_3 = 1 + \alpha^9 + \alpha^{12} + \alpha^{15} = 1 + \alpha^2.$$

(rappelons que

$$\begin{aligned} \alpha^4 &= 1 + \alpha, \alpha^5 = \alpha + \alpha^2, \alpha^6 = \alpha^2 + \alpha^3, \alpha^7 = 1 + \alpha + \alpha^3, \alpha^8 = 1 + \alpha^2, \\ \alpha^9 &= \alpha + \alpha^3, \alpha^{10} = 1 + \alpha + \alpha^2, \alpha^{11} = \alpha + \alpha^2 + \alpha^3, \alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3, \\ \alpha^{13} &= 1 + \alpha^2 + \alpha^3, \alpha^{14} = 1 + \alpha^3, \alpha^{15} = 1). \end{aligned}$$

Le polynôme  $s(X)$  a la forme suivante :

$$\begin{aligned} s(X) &= 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2 = \\ &= 1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^2 + \alpha^3 + (1 + \alpha^2)(\alpha^2 + \alpha^3)^{-1})X^2 \\ &= 1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^3)X^2. \end{aligned}$$

On trouve les racines de ce polynôme :  $X = \alpha$  et  $X = \alpha^7$ . Alors  $\eta_1^{-1} = \alpha$  et  $\eta_2^{-1} = \alpha^7$ , c'est à dire,  $\eta_1 = \alpha^{14}$ ,  $\eta_2 = \alpha^8$ . On connaît alors les erreurs : elles sont dans les positions correspondantes aux  $X^8$  et  $X^{14}$ , c'est-à-dire, la 9<sup>e</sup> et la 15<sup>e</sup> composantes de  $v$ . Alors le mot transmis était

$$w = (100111001000001).$$

On décode ce mot par la division du polynôme correspondant par le polynôme générateur  $g(X)$ . On obtient le polynôme  $1 + X^3 + X^5 + X^6$  et le reste nul. Alors le message initial était

$$(1001011) \longleftrightarrow u(x) = x^6 + x^5 + x^3 + 1 = (v(x) - e(x))/g$$

EXEMPLE.

Si le message reçu est  $v = (100100110000100)$ , soit encore  $v(x) = 1 + x^3 + x^6 + x^7 + x^{12}$ . On calcule alors les composantes du syndrome :

$$S_1 = v(\alpha) = 1 = S_2 = S_4, S_3 = v(\alpha^3) = \alpha^4$$

Cette fois on utilise le système linéaire d'équations des inconnues  $\tau_i$  provenant des relations (4.5) (avec  $b = 1, 2, t' = 2$ ) :  $\tau_{t'}S_b + \tau_{t'-1}S_{b+1} + \cdots + \tau_1S_{b+t'-1} + S_{b+t'} = 0$  :

$$S_2\tau_1 + S_1\tau_2 = S_3$$

$$S_3\tau_1 + S_2\tau_2 = S_4$$

qui s'écrit également

$$\tau_1 + \tau_2 = \alpha^4$$

$$\alpha^4\tau_1 + \tau_2 = 1$$

et dont la matrice est régulière. Il vient

$$\begin{array}{l} \tau_1 = 1 \\ \tau_2 = \alpha \\ \tau_0 = 1 \end{array} \implies s(x) = 1 + x + \alpha x^2.$$

En testant les différentes puissances de  $\alpha$ , on trouve



$$\begin{array}{lcl} \eta_1^{-1} = \alpha^8 & & \eta_1 = \alpha^7 \\ \eta_2^{-1} = \alpha^6 & \implies & \eta_2 = \alpha^9, \end{array}$$

ainsi, le polynôme erreur est

$$e(x) = x^7 + x^9,$$

et on peut décoder

$$\begin{aligned} w(x) &= v(x) - e(x) \\ &= (1 + x^3 + x^6 + x^7 + x^{12}) - (x^7 + x^9) \\ &= 1 + x^3 + x^6 + x^9 + x^{12} \end{aligned}$$

Le message émis était  $w = (100100100100100)$ .

Pour retrouver le message original, il suffit de diviser  $w(x)$  par  $g(x)$ . Il vient

$$a(x) = w(x)/g(x) = 1 + x^3 + x^4$$

et enfin

$$a = (1001100).$$



## 5 Codes BCH et codes de Reed-Solomon. Codage et décodage



Ce sont des codes cycliques particuliers qui permettent de prévoir la distance minimum avant la construction.

### 5.1 Classe des codes BCH (Bose, Ray-Chaudhuri et Hocquenghem)

Lorsque l'on définit un code cyclique par polynôme générateur  $g$ , tous les mots du code sont multiples de ce polynôme, et s'annulent donc sur l'ensemble des racines de  $g$ . De plus, on peut trouver une extension de  $\mathbb{F}_q$  contenant ces racines. Soient donc  $\alpha_1, \dots, \alpha_s$  des éléments d'une extension de  $\mathbb{F}_q$  et  $p_i(x)$  le polynôme minimal de  $\alpha_i$  sur  $\mathbb{F}_q$ ,  $1 \leq i \leq s$ . Soit  $n \in \mathbb{N}$  tel que  $\alpha_i^n = 1$ ,  $1 \leq i \leq s$ , et soit  $g(x) = \text{ppcm}(p_1(x), \dots, p_s(x))$ . Dans ces conditions,  $g(x)$  divise  $x^n - 1$  et si  $C$  est le code du polynôme générateur  $g$ , on a

$$v(x) \in C \iff v(\alpha_i) = 0, \quad i = 1, \dots, s.$$

Dans ce qui suit, nous allons étudier l'intérêt de cette méthode de construction de codes cycliques, par rapport à la construction directe par polynôme générateur.

THÉORÈME 5.1 Soit  $C$  un code cyclique sur  $F$  de longueur  $n$  avec  $n$  premier avec  $q$ , et de polynôme générateur  $g(x)$ . Soit  $L$  le corps des racines  $n^{\text{èmes}}$  de l'unité (le corps de décomposition de  $x^n - 1$  sur  $\mathbb{F}_q$ ). Soit  $b$  un entier, et  $\beta$  une racine primitive  $n^{\text{ème}}$ . Si  $g(x)$  possède, parmi les racines dans une extension  $L$  de  $F = \mathbb{F}_q$ , les puissances de  $\beta$  dont les exposants sont  $d' - 1$  entiers consécutifs, soit

$$\beta^b, \beta^{b+1}, \dots, \beta^{b+d'-2},$$

alors le poids du code  $C$  est supérieur ou égal à  $d'$  :  $d \geq d'$ .

DÉFINITION 5.2 Un code BCH de **distance construite**  $d'$  est un code cyclique dont le générateur est le produit (sans répétition de facteurs) de polynômes minimaux de  $\beta^b, \dots, \beta^{b+d'-2}$ . Dans le cas  $b = 1$  on dit que c'est un **code BCH au sens strict**.

EXEMPLE. Soit  $\beta$  un élément primitif de  $\mathbb{F}_{16}$  racine du polynôme  $x^4 + x + 1$  sur  $\mathbb{F}_2$ . Alors

$$m^{(1)}(x) = m^{(2)}(x) = m^{(4)}(x) = m^{(8)}(x) = x^4 + x + 1$$

$$m^{(3)}(x) = m^{(6)}(x) = m^{(9)}(x) = m^{(12)}(x) = x^4 + x^3 + x^2 + x + 1$$

Ainsi, un code de polynôme générateur

$$g(x) = m^{(1)}(x)m^{(3)}(x) = 1 + x^4 + x^6 + x^7 + x^8$$

est un code BCH dont les paramètres sont  $b = 1, d' = 5 = d, q = 2, n = 15$  puisque  $\beta, \beta^2, \beta^4, \beta^8$  sont les racines de  $m^{(1)}(x)$ , et  $\beta^3$  est une racine de  $m^{(3)}(x) = x^4 + x^3 + x^2 + x + 1 = (x^5 - 1)/(x - 1)$  ( $\beta^3$ )<sup>5</sup> = 1, et

$$\beta, \beta^2, \beta^3, \beta^4,$$

sont des racines d'exposants consécutifs de  $g(x)$ . La dimension du code est  $k = 7$ , et un polynôme de contrôle est

$$h(x) = (x^{15} - 1)/g(x) = 1 + x^4 + x^6 + x^7$$

EXEMPLE. On considère le polynôme  $g(x) = x^6 + x^3 + 1$ , un diviseur de

$$(x^9 - 1) = (x^2 + x + 1)(x^6 + x^3 + 1)(x + 1) \in \mathbb{F}_2[x]$$

On voit que les racines de  $g$  sont  $\beta, \beta^2, \beta^4, \beta^5, \beta^7$ , et que  $\alpha := \beta^5 + \beta^3$  est un générateur du groupe cyclique  $\mathbb{F}_{64}^*$ ,  $\alpha^7 = \beta$ ,  $\beta^9 = 1$ . Il y a donc deux racines d'exposants consécutifs,  $\beta$  et  $\beta^2$ . Le poids minimum du code est  $\geq 3$ , et le mot  $g$  est lui-même de poids 3. Donc le code  $C$  est un 1-correcteur.

```
> restart ;
```

```
> with(linalg) :
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

```
> Factor(x^9+1) mod 2;
```

$$(x^6 + x^3 + 1)(x^2 + x + 1)(x + 1)$$

```
> g:=x^6+x^3+1;alias(beta = RootOf(g)) ;
```

$$g := x^6 + x^3 + 1$$

$$\beta$$

```
> for i from 0 to 8 do
```

```
> if Eval(g, x=beta^i) mod 2 = 0 then print('i'=i, 'g'(beta^i)=0) fi
```

```
> od ;
```

$$i = 1, g(\beta) = 0$$

$$i = 2, g(\beta^2) = 0$$

$$i = 4, g(\beta^4) = 0$$

$$i = 5, g(\beta^5) = 0$$

$$i = 7, g(\beta^7) = 0$$

$$i = 8, g(\beta^8) = 0$$

```
> Expand( beta^7) mod 2;
```

$$\beta^4 + \beta$$

```
> Factor(x^7-beta, beta) mod 2;
```

$$(x + \beta^5 + \beta^4 + \beta^3)(x + \beta^5 + \beta^4 + \beta^2 + 1)(x + \beta^5 + \beta^2 + 1)(x + \beta^4 + \beta^3 + \beta^2 + 1)$$

$$(x + \beta^3 + \beta^2 + 1)(x + \beta^5 + \beta^3)(x + \beta^4)$$

```
> alpha:=beta^5+beta^3;
```

```
> for i from 0 to 63 do
```

```
>   if Eval(x-1, x=alpha^i) mod 2 = 0 then print('i'=i, alpha^i=1) fi
```

```
>   od ;
```

$$\alpha := \beta^5 + \beta^3$$

$$i = 0, 1 = 1$$

$$i = 63, (\beta^5 + \beta^3)^{63} = 1$$

Le théorème permet de trouver un code corrigeant  $t$  erreurs pour tout entier  $t$ . Il suffit de trouver  $n$  tel qu'un diviseur de  $x^n - 1$  satisfasse aux conditions du Théorème 5.1, par exemple le produit  $g(x)$  des diviseurs de  $x^n - 1$ , de racines  $\beta, \beta^2, \dots, \beta^{d'-1}$ .

REMARQUE 5.3 *On peut poser  $d' = 2t + 1$  mais dans ce cas,  $n$  et  $k$  ne peuvent pas être choisies arbitrairement. Les entiers  $n-1$  et  $0$  seront considérés comme consécutifs puisque  $\beta^n = \beta^0 = 1$ .*



Rappelons qu'un code BCH de **distance construite**  $d'$  est un code cyclique dont le générateur est le produit (sans répétition de facteurs) de polynômes minimaux de  $\beta^b, \dots, \beta^{b+d'-2}$ . Dans le cas  $b = 1$  on dit que c'est un **code BCH au sens strict**.

EXEMPLE. Pour obtenir BCH de longueur 9 sur  $\mathbb{F}_2$  et de distance  $\geq 4$  on peut partir des racines  $\beta, \beta^2, \beta^3 \in \mathbb{F}_{64}$ , donc  $d' - 1 = 3$ ,

$$g(x) = (x^6 + x^3 + 1)(x^2 + x + 1) = x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$$

Malheureusement, on obtient le **code de répétition pure** (de poids 9). C'est à dire,  $k = 1$ ,  $n = d = 9$ ; on a vu qu'un tel code n'est pas efficace. Dans ce cas on a choisi  $d' = 4$ , mais en réalité on a obtenu  $d = 9$ . Remarquez que, a priori, on n'a, en général que  $d \geq d'$ .

PREUVE DU THEOREME 5.1 Soit  $C$  un code cyclique sur  $F$  de longueur  $n$  avec  $n$  premier avec  $q$ , et de polynôme générateur  $g(x)$ . Soit  $L$  le corps des racines  $n^{\text{èmes}}$  de l'unité (le corps de décomposition de  $x^n - 1$  sur  $\mathbb{F}_q$ ). Soit  $b$  un entier, et  $\beta$  une racine primitive  $n^{\text{ème}}$ . Le polynôme  $g(x)$  est un polynôme de degré minimal qui possède, parmi les racines dans une extension  $L$  de  $F = \mathbb{F}_q$ , les puissances de  $\beta$  suivantes

$$\beta^b, \beta^{b+1}, \dots, \beta^{b+d'-2}.$$

Soit  $a = (a_0, \dots, a_{n-1})$  un mot du code,  $a \in C \iff g(x)|a(x) \iff a(x)$  s'annule sur les racines de  $g$ , car  $g$  ne possède pas de racines multiples, d'où

[illegible]

Soit  $V_i = {}^t((\beta^b)^i, (\beta^{b+1})^i, \dots, (\beta^{b+d'-2})^i)$ , alors le système (5.1)  $\iff a_0 V_0 + \dots + a_{n-1} V_{n-1} = 0$ . On souhaite montrer que **toutes les  $d' - 1$  colonnes  $V_{i_1}, V_{i_2}, \dots, V_{i_{d'-1}}$  sont linéairement indépendantes**, voir le lemme 3.8.

Mais

$$((\beta^b)^i, (\beta^{b+1})^i, \dots, (\beta^{b+d'-2})^i) = ((\beta^i)^b, (\beta^i)^{b+1}, \dots, (\beta^i)^{b+d'-2}),$$

donc la matrice du système avec les colonnes  $V_{i_1}, V_{i_2}, \dots, V_{i_{d'-1}}$  est

$$\begin{pmatrix} u_1^b & u_2^b & \dots & u_{d'-1}^b \\ u_1^{b+1} & u_2^{b+1} & \dots & u_{d'-1}^{b+1} \\ \dots & \dots & \dots & \dots \\ u_1^{b+d'-2} & u_2^{b+d'-2} & \dots & u_{d'-1}^{b+d'-2} \end{pmatrix}$$

avec  $u_j = \beta^{i_j}, j = 1, \dots, d' - 1$ . Donc le déterminant (de type de Van der Monde) est non nul :

$$\begin{vmatrix} \beta^{bi_1} & \beta^{bi_2} & \dots & \beta^{bi_{d'-1}} \\ \beta^{(b+1)i_1} & \beta^{(b+1)i_2} & \dots & \beta^{(b+1)i_{d'-1}} \\ \dots & \dots & \dots & \dots \\ \beta^{(b+d'-2)i_1} & \beta^{(b+d'-2)i_2} & \dots & \beta^{(b+d'-2)i_{d'-1}} \end{vmatrix} = \beta^{b(i_1+i_2+\dots+i_{d'-1})} \prod_{1 \leq k < j \leq d'-1} (\beta^{i_j} - \beta^{i_k}) \neq 0,$$

car  $u_1 = \beta^{i_1}, \dots, u_{d'-1} = \beta^{i_{d'-1}}$  sont *distincts*.

En conséquence, le système (5.1) ne peut pas être satisfait avec moins de  $d'$  coefficients non-nuls, et donc le poids de  $a$  est au moins  $d'$ .

DÉFINITION 5.4 (a) Soit  $b \in \mathbb{N}$  et soit  $\beta \in \mathbb{F}_{q^m}$  une racine  $n^{\text{ème}}$  primitive de l'unité, où  $m$  est l'ordre multiplicatif de  $q$  modulo  $n$ . Soit  $C$  le code BCH sur  $\mathbb{F}_q$  de longueur  $n$  et de **distance construite**  $d'$ ,  $2 \leq d' \leq n$ , défini par les racines  $\beta^b, \beta^{b+1}, \dots, \beta^{b+d'-2}$ .

Si  $m^{(i)}(x)$  est le polynôme minimal de  $\beta^i$  sur  $\mathbb{F}_q$ , le polynôme générateur du code  $C$  est

$$g(x) = \text{ppcm}(m^{(b)}(x), m^{(b+1)}(x), \dots, m^{(b+d'-2)}(x))$$

(b) De plus, si  $n = q^m - 1$ , les codes BCH correspondants sont appelés **primitifs**.

(c) Si  $n = q - 1$ , le code est dit de **Reed-Solomon**.

EXEMPLE : Soit, avec les notations de la définition,  $m^{(1)}(x) = x^4 + x + 1$  le polynôme minimal sur  $\mathbb{F}_2$  de  $\alpha$  élément primitif de  $\mathbb{F}_{16}$ . On construit alors une matrice de correction  $H$  dont la  $i^{\text{ème}}$  colonne est  $\alpha^i$  exprimé comme combinaison linéaire de  $1, \alpha, \alpha^2, \alpha^3$ . La matrice  $H$  obtenue définit un code équivalent au code de Hamming de dimension 11 et de longueur 15 :

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

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

De plus, on a également  $m^{(1)}(\alpha^2) = 0$ . Le code ainsi construit est donc un code BCH  $\mathbb{F}_2$  avec  $b = 1$  et  $d = 3$ , et corrige donc une seule erreur.

Pour décoder un message reçu  $v \in \mathbb{F}_2^{15}$ , il faut calculer son syndrome  $Hv^t = v(\alpha)$  exprimé dans la base  $\{1, \alpha, \alpha^2, \alpha^3\}$ . Il suffit pour cela de diviser  $v(x)$  par  $m^{(1)}(x)$ . Si  $r(x)$  est le reste de la division, alors  $v(\alpha) = r(\alpha)$ .



Si par exemple

$$v = (010110001011101), v(x) = x + x^3 + x^4 + x^8 + x^{10} + x^{11} + x^{12} + x^{14},$$

alors  $r(x) = 1 + x$  et donc  $Hv^t = 1 + \alpha$ . Or, on a d'après la matrice  $H : \alpha^4 = 1 + \alpha$ , ce qui correspond donc à une erreur en cinquième position. Le message émis était donc  $w = (010100001011101)$ ,

$$w(x) = x + x^3 + x^4 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{14} = (x^{10} + x^8 - x^5 + x^2 + x) \cdot (1 + x + x^4).$$

VÉRIFICATION :

```
> rem(x+x^3+x^8+x^(10)+x^(11)+x^(12)+x^(14), 1+x+x^4,x, 'q') mod 2;
```

0

```
> q;
```

$$x^{10} + x^8 - x^5 + x^2 + x$$



## 5.2 Codes de Reed-Solomon

Selon la définition 5.4, un code de Reed-Solomon est un code BCH sur  $\mathbb{F}_q$  de longueur  $q-1$  et de distance construite  $d'$ ,  $2 \leq d' \leq n$ , défini par les racines  $\beta^b, \beta^{b+1}, \dots, \beta^{b+d'-2} \in \mathbb{F}_q^*$ .

On utilise souvent  $q = 2^m$ , et on va écrire  $\alpha$  au lieu de  $\beta$  :

$$x^{2^m-1} - 1 = \prod_{u \in \mathbb{F}_{2^m}^*} (x - u) = (x - 1)(x - \alpha) \cdot \dots \cdot (x - \alpha^{2^m-2})$$

donc

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdot \dots \cdot (x - \alpha^{b+d'-2}), \quad d^\circ(g) = d' - 1$$

avec les facteur linéaires dans  $\mathbb{F}_{2^m}[x]$ .

D'une part, par Théorème 5.1,  $k = 2^m - 1 - d' + 1 = 2^m - d'$ ,  $d \geq d'$ . D'autre part, par la borne de Singleton, Théorème 2.6,  $d \leq n - k + 1 = 2^m - 1 - (2^m - d') + 1 = d'$ , donc le poids de  $C$  est *exactement*  $d'$ .

EXAMPLE. 1) Soit  $\mathbb{F}_8^* = \langle \alpha \rangle$ ,  $\alpha^3 = \alpha + 1$ ,  $g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3) = \alpha^6 + \alpha x + \alpha^6 x^2 + x^3$ . Alors  $n = 7, k = 4, d = 4$ ,

$$G = \begin{pmatrix} \alpha^6 & \alpha & \alpha^6 & 1 & 0 & 0 & 0 \\ 0 & \alpha^6 & \alpha & \alpha^6 & 1 & 0 & 0 \\ 0 & 0 & \alpha^6 & \alpha & \alpha^6 & 1 & 0 \\ 0 & 0 & 0 & \alpha^6 & \alpha & \alpha^6 & 1 \end{pmatrix}.$$

2) Soit  $\mathbb{F}_{256}^* = \langle \alpha \rangle$ ,  $g = \prod_{12}^{43}(x - \alpha^{11j})$ ,  $\alpha^8 = \alpha^7 + \alpha^2 + \alpha + 1$ . Alors  $n = 255, k = 223, d = 33$  (un code utilisé par le NASA).





### 5.3 Deuxième description des codes de Reed-Solomon

THÉORÈME 5.5 (voir [Pa-Wo], p.139) Soit  $\mathbb{F}_{2^m} = \langle \alpha \rangle$ , et soit  $p(x) \in \mathbb{F}_{2^m}[x]$  parcourt le sous-espace linéaire  $\mathcal{P}_k \subset \mathbb{F}_{2^m}[x]$  avec  $d^\circ(p) \leq k-1$  ou  $p \equiv 0$ . Alors l'ensemble des mots de la forme

$$(p(1), p(\alpha), \dots, p(\alpha^{2^m-2}))$$

est un code de Reed-Solomon de longueur  $n = 2^m - 1$ , de polynôme

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^r)$$

avec  $r = 2^m - 1 - k$ .

REMARQUE IMPORTANTE. Cette description montre que les mots de code sont certaines *fonctions polynômiales* sur l'ensemble des racines d'un polynôme.

*Preuve* On remarque que l'application

$$\Phi : p \mapsto (p(1), p(\alpha), \dots, p(\alpha^{2^m-2})) \in \mathbb{F}_q^n$$

est injective puisque  $\text{Ker}(\Phi) = 0$  par l'interpolation de Lagrange. On pose  $C_1 = \text{Im}(\Phi) = \Phi(\mathcal{P}_k)$ .

Une base convenable de  $C_1$  :

$$c_i = \Phi(p_i), p_i = x^i \quad (i = 0, 1, \dots, k-1) : c_i = (1, \alpha^i, \alpha^{2i}, \dots, \alpha^{(2^m-2)i}),$$

donc  $c_i(x) := \sum_{j=0}^{2^m-2} \alpha^{ji} x^j$ .

Ensuite, on calcule le syndrôme du polynôme  $c_i(x)$  :

$$c_i(\alpha^t) := \sum_{j=0}^{2^m-2} (\alpha^{i+t})^j = \sum_{j=0}^{2^m-2} \beta^j, \text{ avec } \beta = \alpha^{i+t}.$$

Puisque  $1 \leq t \leq r$ ,  $0 \leq i \leq k-1$ , alors  $1 \leq i+t \leq r+k-1 = 2^m-2$ , donc

$$\beta \neq 1, c_i(\alpha^t) = \frac{\beta^{2^m-2} - 1}{\beta - 1} = 0.$$

Ceci dit,  $C_1 \subset C$ , puisque tous les mots de  $C_1$  ont le syndrôme nul. Mais

$$\dim_{\mathbb{F}_{2^m}} C = \dim_{\mathbb{F}_{2^m}} C_1 = k \Rightarrow C = C_1.$$

EXERCICE. Montrer un résultat analogue pour un corps fini  $\mathbb{F}_q$  arbitraire.



## 5.4 Problèmes de décodage.

Nous avons vu que la classe des codes BCH permet, sous seule condition d'augmenter  $n$  et donc  $m$ , de construire des codes de poids  $\geq d'$  pour tout entier positif  $d'$ . Nous allons, dans ce qui suit, construire un algorithme général de décodage des codes BCH, puis l'appliquer dans un exemple.

Considérons un code BCH de distance construite  $d' \geq 2t + 1$ . Supposons que  $v, w$  et  $e$  soient le message reçu, le message émis et le vecteur erreur,  $v = w + e$ . Il faut dans un premier temps calculer le syndrome de  $v$

$$S(v) = Hv^t = (S_b, S_{b+1}, \dots, S_{b+d'-2})^t$$

avec  $S_j = v(\beta^j) = e(\beta^j)$  pour  $b \leq j \leq b + d' - 2$ .

Si  $r$  erreurs se produisent, avec  $r \leq t$ , alors  $2r + 1 \leq d'$  et

$$e(x) = \sum_{i=1}^r c_i x^{a_i},$$

où les  $a_i$  sont tous différents dans  $0, 1, \dots, n - 1$ . Les  $\eta_i = \beta^{a_i}$  sont les **locateurs d'erreurs**, et les  $c_i$ , qui sont des éléments de  $\mathbb{F}_q^*$ , sont les **valeurs d'erreurs**. Or,

$$S_j = e(\beta^j) = \sum_{i=1}^r c_i \eta_i^j \quad (j = b, b + 1, \dots, b + d' - 2).$$

Nous voyons donc que nous pourrions décoder dès lors que nous aurons obtenu les couples  $(c_i, \eta_i)$ . Dans le cas binaire, les  $c_i$  sont de plus tous égaux à 1.

REMARQUE 5.6 On a  $S_j^q = S_{jq}$ . En effet,

$$S_j^q = \left( \sum_{i=1}^r c_i \eta_i^j \right)^q = \sum_{i=1}^r c_i^q \eta_i^{jq} = \sum_{i=1}^r c_i \eta_i^{jq} = S_{jq}$$

Soit alors l'identité polynomiale

$$\prod_{i=1}^r (\eta_i - x) = \sum_{i=0}^r (-1)^i \sigma_{r-i} x^i = \sigma_r - \sigma_{r-1} x + \cdots + (-1)^r \sigma_0 x^r$$

Les coefficients  $\sigma_i$  sont donc  $\sigma_0 = 1$  et pour  $1 \leq i \leq r$ , les  $\sigma_i$  sont les polynômes élémentaires symétriques en  $\eta_1, \dots, \eta_r$ . En remplaçant  $x$  par  $\eta_i$ , il vient

$$(-1)^r \sigma_r + (-1)^{r-1} \sigma_{r-1} \eta_i + \cdots - \sigma_1 \eta_i^{r-1} + \eta_i^r = 0$$

pour tout  $1 \leq i \leq r$ .

En multipliant par  $c_i \eta_i^j$  et en sommant pour  $1 \leq i \leq r$ , il vient

$$\sum_{i=1}^r ((-1)^r \sigma_r c_i \eta_i^j + (-1)^{r-1} \sigma_{r-1} c_i \eta_i^{j+1} + \cdots - \sigma_1 c_i \eta_i^{j+r-1} + c_i \eta_i^{j+r}) = 0$$

donc

$$(-1)^r \sigma_r S_j + (-1)^{r-1} \sigma_{r-1} S_{j+1} + \cdots - \sigma_1 S_{j+r-1} + S_{j+r} = 0$$

pour tout  $b \leq j \leq b+r-1$ , où  $j+r \leq b+2r-1 \leq b+d'-2$  puisque  $2r+1 \leq d'$ . (On rappelle que

$$S_j = e(\beta^j) = \sum_{i=1}^r c_i \eta_i^j \quad (j = b, b+1, \dots, b+d'-2).)$$

LEMME 5.7 *Le système d'équations*

$$(-1)^r \sigma_r S_j + (-1)^{r-1} \sigma_{r-1} S_{j+1} + \cdots - \sigma_1 S_{j+r-1} + S_{j+r} = 0,$$

$b \leq j \leq b + r - 1$ , des inconnues  $(-1)^i \sigma_i$  est résoluble avec une seule solution si et seulement si  $r$  erreurs se sont produites.

PREUVE : La matrice du système est décomposable en produit suivant :

$$\begin{pmatrix} S_b & S_{b+1} & \cdots & S_{b+r-1} \\ S_{b+1} & S_{b+2} & \cdots & S_{b+r} \\ \cdots & \cdots & \cdots & \cdots \\ S_{b+r-1} & S_{b+r} & \cdots & S_{b+2r-2} \end{pmatrix} = V D V^t$$

avec

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \eta_1 & \eta_2 & \cdots & \eta_r \\ \cdots & \cdots & \cdots & \cdots \\ \eta_1^{r-1} & \eta_2^{r-1} & \cdots & \eta_r^{r-1} \end{pmatrix}, D = \begin{pmatrix} c_1 \eta_1^b & 0 & \cdots & 0 \\ 0 & c_2 \eta_2^b & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & c_r \eta_r^b \end{pmatrix}$$

$V$  est une matrice de Van der Monde régulière dès lors que les  $\eta_i$  sont tous distincts et  $D$  est diagonale, donc régulière si et seulement si les  $c_i$  et les  $\eta_i$  sont tous non nuls. Ces deux conditions sont remplies si et seulement si  $r$  erreurs se produisent.

En effet, la multiplication des matrices montre

$$\begin{aligned}
VDV^t &= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \eta_1 & \eta_2 & \cdots & \eta_r \\ \cdots & \cdots & \cdots & \cdots \\ \eta_1^{r-1} & \eta_2^{r-1} & \cdots & \eta_r^{r-1} \end{pmatrix} \begin{pmatrix} c_1\eta_1^b & 0 & \cdots & 0 \\ 0 & c_2\eta_2^b & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & c_r\eta_r^b \end{pmatrix} \begin{pmatrix} 1 & \eta_1 & \cdots & \eta_r^{r-1} \\ 1 & \eta_2 & \cdots & \eta_r \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \eta_1^{r-1} & \cdots & \eta_r^{r-1} \end{pmatrix} \\
&= \begin{pmatrix} c_1\eta_1^b & c_2\eta_2^b & \cdots & c_r\eta_r^b \\ c_1\eta_1^{b+1} & c_2\eta_2^{b+1} & \cdots & c_r\eta_r^{b+1} \\ \cdots & \cdots & \cdots & \cdots \\ c_1\eta_1^{b+r-1} & c_2\eta_2^{b+r-1} & \cdots & c_r\eta_r^{b+r-1} \end{pmatrix} \begin{pmatrix} 1 & \eta_1 & \cdots & \eta_r^{r-1} \\ 1 & \eta_2 & \cdots & \eta_r \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \eta_1^{r-1} & \cdots & \eta_r^{r-1} \end{pmatrix} \\
&= \begin{pmatrix} S_b & S_{b+1} & \cdots & S_{b+r-1} \\ S_{b+1} & S_{b+2} & \cdots & S_{b+r} \\ S_{b+r-1} & S_{b+r} & \cdots & S_{b+2r-2} \end{pmatrix}
\end{aligned}$$

DÉFINITION 5.8 : On appelle polynôme locateur d'erreur le polynôme

$$s(x) = \prod_{i=1}^r (1 - \eta_i x) = \sum_{i=0}^n (-1)^i \sigma_i x^i$$

avec les notations utilisées précédemment. Les racines de  $s(x)$  sont les  $\eta_i^{-1}$ , c'est à dire les  $\beta^{-a_i}$ . Il suffit donc ensuite d'évaluer le polynôme  $s(x)$  pour les différentes puissances de  $\beta$  pour localiser les erreurs.



APRÈS AVOIR TROUVÉ LES RACINES  $\eta_i$ , on utilise le lemme suivant :

LEMME 5.9 *Le système  $S_j = \sum c_i \eta_i^j$ ,  $b \leq j \leq b+r-1$  des inconnues  $c_i$  est résoluble si les coefficients  $\eta_i$  sont des éléments distincts de  $\mathbb{F}_{q^m}^*$ .*

PREUVE : Le déterminant du système est alors

$$\begin{vmatrix} \eta_1^b & \eta_2^b & \cdots & \eta_r^b \\ \eta_1^{b+1} & \eta_2^{b+1} & \cdots & \eta_r^{b+1} \\ \cdots & \cdots & \cdots & \cdots \\ \eta_1^{b+r-1} & \eta_2^{b+r-1} & \cdots & \eta_r^{b+r-1} \end{vmatrix} = \eta_1^b \eta_2^b \cdots \eta_r^b \prod_{1 \leq i < j \leq r} (\eta_i - \eta_j) \neq 0$$

Nous allons résumer cet algorithme en notant  $\tau_i = (-1)^i \sigma_i$ .

### 5.4.1. Décodage BCH

Soit  $C$  un code BCH de distance construite  $d' \geq 2t + 1$ ,  $w$  le message émis et  $v$  le message reçu. Supposons qu'au plus  $t$  erreurs se produisent.

1ère étape : Déterminer le syndrome

$$S(v) = (S_b, S_{b+1}, \dots, S_{b+d'-2})^t.$$

(On rappelle que

$$S_j = e(\beta^j) = \sum_{i=1}^r c_i \eta_i^j \quad (j = b, b+1, \dots, b+d'-2).)$$

2ème étape : Déterminer l'entier  $r$  maximal tel que le système d'équations

$$S_{j+r} + S_{j+r-1}\tau_1 + \dots + S_j\tau_r = 0, \quad b \leq j \leq b+r-1$$

des inconnues  $\tau_i$  soit résoluble avec *une seule solution* pour ainsi déterminer le nombre d'erreurs  $r$ . Ceci permet de former le polynôme *locateur d'erreur*

$$s(x) = \prod_{i=1}^r (1 - \eta_i x) = \sum_{i=0}^r \tau_i x^i$$

dont les coefficients se déduisent des  $S_j$ .

3ème étape : Chercher les racines de  $s(x)$  en testant les différentes puissances de  $\beta$  pour déterminer les locateurs d'erreurs  $\eta_i$  .

4ème étape : Remplacer les  $\eta_i$  par leurs valeurs dans l'expression des  $S_j$  pour déterminer les valeurs des erreurs  $c_i$  (cas non binaire seulement). On obtient ainsi le vecteur erreur  $e$  et on peut décoder  $w = v - e$ .

REMARQUE 5.10 : *L'étape difficile est la 2<sup>ème</sup>. Pour parvenir à déterminer les coefficients  $\tau_i$ , une méthode possible est d'utiliser l'algorithme de Berlekamp-Massey (voir [Li-Ni], p.235-239, [vLi-vdG], p.20) et la section 5.5.*



## Cours N°5

(disponible sur : <http://www-fourier.ujf-grenoble.fr/~panchish/SCCI>).

### 5.5 Algorithme de Berlekamp-Massey

(voir [vLi-vdG], p.20, [vLi], p.98, et le mémoire de TER de Insa FRÖHLICH, 2004).

Soit  $C$  un  $[n, k, d]_q$ -code BCH de zéros  $\beta, \beta^2, \dots, \beta^{d'-1}$ , ou  $\mathbb{F}_{q^m}^* \supset \langle \beta \rangle_n$ ,  $\beta^n = 1$ , une racine primitive de degré  $n$  dans  $\mathbb{F}_{q^m}$ . Soient  $v, w$  et  $e$  le message reçu, le message émis et le vecteur erreur,  $v = w + e$ , considérés comme des **polynômes** :  $v(x), w(x)$  et  $e(x) \in \mathbb{F}_q[x]$ .

On pose

$$v(x) = v_0 + \dots + v_{n-1}x^{n-1}, w(x) = \cancel{v}_0 + \dots + w_{n-1}x^{n-1},$$

$$e(x) := v(x) - w(x) = e_0 + \dots + e_{n-1}x^{n-1}$$

et soit  $I := \{i \mid e_i \neq 0\}$  (comparer avec la notation de la section 5.4 :  $e(x) = \sum_{i=1}^r c_i x^{a_i}$ , donc  $c_i = e_{a_i}$ ).

DÉFINITION 5.11 (a) Le polynôme **locateur d'erreurs** est

$$s(x) = \prod_{i \in I} (1 - \beta^i x),$$

Le degré de  $s(x)$  est égale au nombre d'erreurs commises, donc il est inférieur ou égale à  $t$ , où  $t = \left\lfloor \frac{d' - 1}{2} \right\rfloor$ . (b) Le polynôme évaluateur d'erreurs est

$$u(x) = \sum_{i \in I} e_i \beta^i \prod_{j \in I \setminus \{i\}} (1 - \beta^j x).$$

Le degré de  $u(x)$  est strictement plus petit que  $t$ .

On constate que

$$e_i = -\frac{u(\beta^{-i})}{s'(\beta^{-i})}, \quad s'(x) = \sum_{i \in I} -\beta^i \prod_{j \in I \setminus \{i\}} (1 - \beta^j x)$$

puisque

$$s'(\beta^{-i}) = -\beta^i \prod_{j \in I \setminus \{i\}} (1 - \beta^{j-i}), \quad u(\beta^{-i}) = e_i \beta^i \prod_{j \in I \setminus \{i\}} (1 - \beta^{j-i}).$$

De plus  $d^\circ(s(x)) \leq t = \left\lceil \frac{d'-1}{2} \right\rceil$ ,  $d^\circ(u(x)) < t$ . Il reste à trouver les polynômes  $s(x)$ ,  $u(x)$ .  
On utilise l'identité formelle

$$\frac{u(x)}{s(x)} = \sum_{i \in I} \frac{e_i \beta^i}{1 - \beta^i x} = \sum_{i \in I} e_i x^{-1} \sum_{l=1}^{\infty} (\beta^i x)^l = \sum_{l=1}^{\infty} x^{l-1} e(\beta^l), \quad (5.2)$$

puisque

$$u(x) = \sum_{i \in I} e_i \beta^i \prod_{j \in I \setminus \{i\}} (1 - \beta^j x) \text{ et } \boxed{e(x) = v(x) - w(x) = e_0 + \cdots + e_{n-1} x^{n-1}.}$$

Les  $d' - 1$  premiers coefficients de la série à droite dans (5.2) sont connus :

$$e(\beta^l) = v(\beta^l) \quad (1 \leq l \leq d' - 1).$$

On pose

$$S(x) = \sum_{l=1}^{d'-1} e(\beta^l) x^{l-1} \text{ le polynôme de syndrome}$$

et on cherche  $u(x), s(x)$  à partir de l'égalité (5.2) comme une solution de la congruence :

$$u(x) \equiv s(x)S(x) \pmod{x^{d'-1}} \quad (5.3)$$

Pour résoudre la congruence (5.3) on utilise la division euclidienne et l'identité de Bezout : on calcule trois suites  $s_j(x), r_j(x), u_j(x)$  avec la propriété

$$r_j(x)x^{d'-1} + s_j(x)S(x) = u_j(x)$$

où le degré de  $u_j(x)$  décroît jusqu'à ce que  $\deg(u_{j+1}) < t$  et  $\deg(u_j) \geq t$ , à partir de

$$1 \cdot x^{d'-1} + 0 \cdot S(x) = x^{d'-1}, 0 \cdot x^{d'-1} + 1 \cdot S(x) = S(x),$$

où  $t = \left\lfloor \frac{d' - 1}{2} \right\rfloor$ , de telle façon que

$$(s_0(x), t_0(x), u_0(x)) = (1, 0, x^{d'-1}); \quad (s_1(x), t_1(x), u_1(x)) = (0, 1, S(x)).$$

On va montrer que le couple  $(u(x), s(x)) = (u_{j+1}(x), s_{j+1}(x))$  est un unique couple vérifiant (5.3).

Le problème, qui reste en appliquant l'algorithme de décodage, est qu'on ne connaît pas le nombre exact  $r$  d'erreurs commises. Cependant, on suppose toujours que le nombre maximal d'erreurs est borné par  $t := \left\lfloor \frac{d' - 1}{2} \right\rfloor$ . En général, on a donc un polynôme d'er-

reur  $e(x) = \sum_{i=0}^{n-1} e_i x^i$  qui possède beaucoup de coefficients  $e_i = 0$ . Rappelons qu'on note par

$I = \{i \mid e_i \neq 0\}$  l'ensemble d'indices des coefficients non nuls. Le polynôme locateur d'erreurs s'écrit donc  $s(x) = \prod_{i \in I} (1 - \beta^i x)$  avec  $\deg(s) \leq t$ . Pour un code BCH le système linéaire de

la deuxième étape de l'algorithme est de la forme

$$\begin{pmatrix} e(\beta^b) & e(\beta^{b+1}) & \dots & e(\beta^{b+t'-1}) \\ e(\beta^{b+1}) & e(\beta^{b+2}) & \dots & e(\beta^{b+t'}) \\ \dots & \dots & \dots & \dots \\ e(\beta^{b+t'-1}) & e(\beta^{b+t'}) & \dots & e(\beta^{b+2t'-2}) \end{pmatrix} \tau = \begin{pmatrix} e(\beta^{b+t'}) & e(\beta^{b+t'+1}) & \vdots & e(\beta^{b+2t'-1}) \end{pmatrix}$$

avec  $\tau = ((-1)^{t'} \sigma_{t'}, (-1)^{t'-1} \sigma_{t'-1}, \dots, -\sigma_1)$ .

On a vu dans la section 5.4 que ce système a une unique solution si et seulement si  $t' = r$  coïncide avec le nombre d'erreurs. Pour ne pas considérer plusieurs systèmes, on utilise l'algorithme suivant de Berlekamp-Massey, qui détermine directement le polynôme locateur d'erreurs  $s(x)$  du bon degré. On utilise pour ceci un code BCH au sens strict, c'est-à-dire  $b = 1$ .

Puisque le code est engendré par  $\beta, \dots, \beta^{d'-1}$ , les premiers  $d' - 1$  coefficients de la série sont connus (première étape de l'algorithme de décodage) et on écrit la somme partielle  $S(x) := \sum_{j=1}^{d'-1} e(\beta^j) x^{j-1}$ . Donc  $u(x) \equiv s(x)S(x) \pmod{x^{d'-1}}$ . L'algorithme de Berlekamp-Massey calcule trois suites de polynômes,  $u_i(x)$ ,  $s_i(x)$  et  $r_i(x)$ , qui vérifient l'égalité :

$$u_i(x) = s_i(x)S(x) + r_i(x)x^{d'-1} \text{ pour } i \geq 0 : \quad (5.4)$$

On pose  $u_0 = x^{d'-1}$ ,  $u_1 = S(x)$ ,  $s_0 = r_1 = 0$ ,  $s_1 = r_0 = 1$  (pour  $i = 0, 1$ ).

On calcule  $u_{i+1} = u_{i-1} - u_i q_i$  par la division euclidienne



et puis  $s_{i+1} = s_{i-1} - s_i q_i$  et  $r_{i+1} = r_{i-1} - r_i q_i$  (l'algorithme d'Euclide étendu), jusqu'on obtient  $\deg(u_{i+1}) < t$  et  $\deg(u_i) \geq t$ .

Résultat :  $s(x) = s_{i+1}(x)$  est le polynôme locateur d'erreurs, et  $u(x) = u_{i+1}(x)$  est le polynôme évaluateur d'erreurs.

L'ensemble  $I$  contient comme dans ce qui précède tous les  $i$  pour lesquels  $\beta^{-i}$  est une racine de  $s(x)$ . Puis, les  $e_i$  se calculent facilement avec la formule ci-dessus.

**THÉORÈME 5.12** *Soit  $d' = 2t + 1$ . L'algorithme de Berlekamp-Massey donne (à facteur constant près) l'unique couple  $s(x), u(x)$  avec  $\deg(s) \leq t$ ,  $\deg(u) < t$  vérifiant la congruence  $u(x) \equiv s(x)S(x) \pmod{x^{d'} - 1}$ .*

*Preuve* : **Existence** : Il est clair que chaque couple  $u_i, s_i$  vérifie l'égalité (5.4). Il faut donc contrôler le degré de  $u_i$  et  $s_i$ . La division euclidienne montre que  $u_{i-1} = u_i q_i + u_{i+1}$  avec  $\deg(u_{i+1}) < \deg(u_i)$ . Donc la suite  $(\deg(u_i))$  est strictement décroissante pour  $i \geq 1$  et il existe un  $j \in \mathbb{N}$  avec  $\deg(u_j) \geq t$  et  $\deg(u_{j+1}) < t$ . En plus, on obtient de la première équation  $\deg(q_i) = \deg(u_{i-1}) - \deg(u_i) \geq 1$  pour  $i \geq 2$ . On utilise cette propriété pour estimer le degré des  $s_i$ . On a  $s_2 = -s_1 q_1$ , donc  $\deg(s_2) \geq \deg(s_1)$ . Pour  $i \geq 2$  la suite  $(\deg(s_i))$  est strictement croissante : Pour  $s_3 = s_1 - s_2 q_2$ , le cas précédent donne  $\deg(s_3) = \deg(s_2) + \deg(q_2) > \deg(s_2)$ . Si on a cette propriété jusqu'à  $i$ , on obtient pour  $s_{i+1} = s_{i-1} - s_i q_i$ ,  $\deg(s_{i+1}) = \deg(s_i) + \deg(q_i) > \deg(s_i)$ . Le degré de  $s_{j+1}$  est donc

$$\begin{aligned}
& \deg(s_{j+1}) = \deg(s_j) + \deg(q_j) \\
\text{donné par une somme télescopique :} & = \deg(s_1) + \deg(q_1) + \dots + \deg(q_j) \\
& = \deg(u_0) - \deg(u_j) \leq t, \\
\text{car } \deg(q_i) = \deg(u_{i-1}) - \deg(u_i) \text{ pour } i \geq 2, \deg(u_0) = d' - 1 = 2t, \deg(u_j) \geq t.
\end{aligned}$$

**Unicité** : On suppose, qu'il y a  $s(x)$ ,  $u(x)$  et  $s^*(x)$ ,  $u^*(x)$  avec

$$u(x) \equiv s(x)S(x) \pmod{x^{d'-1}} \text{ et } u^*(x) \equiv s^*(x)S(x) \pmod{x^{d'-1}}.$$

En multipliant la congruence par  $s^*(x)$ , respectivement par  $s(x)$ , on obtient

$$s^*(x)u(x) \equiv s(x)u^*(x) \pmod{x^{d'-1}}.$$

Puisque  $\deg(s), \deg(s^*) \leq t$  et  $\deg(u), \deg(u^*) < t$ , les deux côtés de la congruence sont de degré inférieur à  $d' - 1$ . On a donc égalité, d'où  $u = \gamma u^*$ ,  $s = \gamma s^*$ . Comme  $u(x)$  et  $s(x)$  sont premiers entre eux,  $\gamma$  doit être une constante non nulle.

## Exemple de décodage

Soit  $\alpha$  une racine du polynôme  $x^2 + x + 2$  irréductible sur  $\mathbb{F}_3$ .  $\alpha$  engendre  $\mathbb{F}_9^*$  et les puissances de  $\alpha$  sont :

$$\begin{aligned} \alpha^1 &= \alpha, & \alpha^2 &= 2\alpha + 1, & \alpha^3 &= 2\alpha + 2, & \alpha^4 &= 2, \\ \alpha^5 &= 2\alpha, & \alpha^6 &= \alpha + 2, & \alpha^7 &= \alpha + 1, & \alpha^8 &= 1. \end{aligned}$$

On considère le polynôme générateur

$$g(x) = (x^2 + 1)(x^2 + x + 2)(x + 1) = (x - \alpha)(x - \alpha^2)(x - \alpha^3)(x - \alpha^4)(x - \alpha^6),$$

divisible par le polynôme minimal de  $\alpha$  et possède les racines  $\alpha, \alpha^2, \alpha^3, \alpha^4$  et  $\alpha^6$ . La matrice de contrôle  $H$ , qui est engendrée par leurs puissances, s'écrit alors :

$$H = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^1 & \alpha^4 & \alpha^7 & \alpha^2 & \alpha^5 \\ 1 & \alpha^4 & 1 & \alpha^4 & 1 & \alpha^4 & 1 & \alpha^4 \\ 1 & \alpha^6 & \alpha^4 & \alpha^2 & 1 & \alpha^6 & \alpha^4 & \alpha^2 \end{pmatrix}.$$

Soit le mot reçu  $c' = (0, 2, 1, 0, 0, 1, 1, 2)$  et on suppose, qu'au plus deux erreurs ont été commises. Le polynôme d'erreurs est alors  $e(x) = e_{a_1}x^{a_1} + e_{a_2}x^{a_2}$ . En calculant  $Hc'^t$ , on obtient

$$(e(\alpha), e(\alpha^2), e(\alpha^3), e(\alpha^4), e(\alpha^6)) = (\alpha^4, \alpha^3, \alpha^4, 0, \alpha).$$

Avec ces valeurs, on peut déterminer les coefficients du polynôme locateur d'erreurs  $s(x) = 1 - \sigma_1(\alpha^{a_1}, \alpha^{a_2})x + \sigma_2(\alpha^{a_1}, \alpha^{a_2})x^2$ . Puisqu'on a quatre puissances consécutives de  $\alpha$  comme racines de  $g(x)$ , il faut résoudre le système suivant :

$$\begin{vmatrix} 0 & = & \sigma_2 e(\alpha) + 2\sigma_1 e(\alpha^2) + e(\alpha^3) \\ 0 & = & \sigma_2 e(\alpha^2) + 2\sigma_1 e(\alpha^3) + e(\alpha^4) \end{vmatrix} \iff \begin{vmatrix} \sigma_1 & = & 1 \\ \sigma_2 & = & \alpha \end{vmatrix}$$

Donc  $s(x) = 1 + 2x + \alpha x^2$ . En évaluant  $s(x)$  en les puissances de  $\alpha$ , on voit que  $s(\alpha^2) = 0$  et  $s(\alpha^5) = 0$ . Donc  $a_1 = 8 - 5 = 3$ ,  $a_2 = 8 - 2 = 6$ , d'où le polynôme d'erreurs  $e(x) = e_3 x^3 + e_6 x^6$ . En utilisant les valeurs de  $e(\alpha)$  et de  $e(\alpha^4)$ , on obtient le système

$$\begin{vmatrix} \alpha^4 & = & e_3 \alpha^3 + e_6 \alpha^6 \\ 0 & = & e_3 \alpha^4 + e_6 \end{vmatrix} \Leftrightarrow \begin{vmatrix} e_3 & = & 2 \\ e_6 & = & e_3 \end{vmatrix}.$$



Ces valeurs de  $e_3$  et  $e_6$  sont aussi des solutions pour les trois autres égalités. Le polynôme d'erreurs est donc  $e(x) = 2x^3 + 2x^6$  et le mot envoyé était  $c = c' - e = (0, 2, 1, 1, 0, 1, 2, 2)$ . La division de  $c(x) = 2x^7 + 2x^6 + x^5 + x^3 + x^2 + 2x$  par  $g(x)$  donne bien le mot décodé  $m(x) = 2x^2 + x$ , qui correspond à  $m = (0, 1, 2)$ .

La méthode de décodage naïve (la division de  $c'(x)$  par  $g(x)$ ) donnerait dans ce cas  $m(x) = 2 + x^2$ , respectivement  $c = (1, 0, 0, 2, 0, 1, 1, 2)$ , où seulement trois lettres sont correctes.

Par la construction du polynôme générateur  $g$ , il s'agit d'un code BCH au sens strict avec  $d' = 5$  et donc  $t = 2$ . Si  $c' = (0, 2, 1, 0, 0, 1, 1, 2)$  est le mot reçu,  $(\alpha^4, \alpha^3, \alpha^4, 0, \alpha)$  est son syndrome. Le polynôme  $S(x)$  s'écrit  $S(x) = \alpha^4 + \alpha^3x + \alpha^4x^2 + 0x^3 = 2 + \alpha^3x + 2x^2$ . L'algorithme de Berlekamp-Massey calcule le tableau suivant :

$i$	0	1	2
$u_i$	$x^4$	$2x^2 + \alpha^3x + 2$	$2x + \alpha^3$
$q_i$		$2x^2 + \alpha^7x + \alpha^3$	
$r_i$	1	0	1
$s_i$	0	1	$x^2 + \alpha^3x + \alpha^7$

On a  $\deg(u_2) = 1 < 2$  et  $\deg(u_1) = 2 \geq 2$ . Donc  $u_2(x) = 2x + \alpha^3$  est le polynôme évaluateur d'erreurs et  $s_2(x) = x^2 + \alpha^3x + \alpha^7$  est le polynôme locateur d'erreurs. En essayant les différentes puissances de  $\alpha$ , on (re)trouve que  $s_2(\alpha^2) = 0$  et  $s_2(\alpha^5) = 0$ , d'où  $I = \{3, 6\}$ . Car  $s'_2(x) = 2x + \alpha^3 = u_2(x)$ , les coefficients du polynôme d'erreur sont  $e_3 = e_6 = -1 = 2$ . On a donc obtenu le polynôme d'erreur sans savoir à priori, qu'il y avait deux erreurs à corriger.

**Exemple : l'algorithme de Berlekamp-Massey, disponible à l'adresse cachée :**  
<http://www-fourier.ujf-grenoble.fr/~panchish/04mag-maple>   
**dans le fichier 4mag-09berl-ma.mws** 

On cherche  $u(x), s(x)$  à partir de l'égalité (5.2) comme une solution de la congruence (5.3) :

$$u(x) \equiv s(x)S(x) \pmod{x^{d'}-1}$$

Pour résoudre la congruence (5.3) on utilise la division euclidienne et l'identité de Bezout :  
on calcule trois suites  $s_n(x), t_n(x), u_n(x)$  avec la propriété

$$t_n(x)x^{d'-1} + s_n(x)S(x) = u_n(x)$$

où le degré de  $u_n(x)$  décroît jusqu'à ce que  $\deg(u_n) < t$  et  $\deg(u_{n-1}) \geq t$

```
> restart ;  
> with(linalg) :  
> alias(alpha = RootOf(X^2+X+2) mod 3) ;
```

$\alpha$

```

> BerlMa:=proc(a::anything,dp::posint)
> local S,B,s0,s1,s2,r0,r1,r2,u0,u1,u2,t,q,i;
> # après on va substituer  $S:=\alpha^4+\alpha^3X+\alpha^4X^2+0X^3$ ; dp:=5;
> t:=floor((dp-1)/2);
> b:=X^(dp-1);
> s0:=0:r0:=1:s1:=1:r1:=0:
> u0:=b; u1:=a; u2:=rem( u0,u1,X ); i:=0;
> while(degree(u1,X)>t-1) do
> print('i'=i,degree(u1,X)>t-1);
> #d1<>0) do
> i:=i+1;
> q:=quo( u0,u1,X ); u2:=rem( u0,u1,X );
> u0:=u1; u1:= u2;
> s2:=s0-q*s1;
> r2:=r0-q*r1;
> s0:=s1;s1:= s2;
> r0:=r1;r1:= r2;
> #printf("i=%d,%a*%a+%a*%a=%a\n"
> #,i,a,u0 ,b, v0, d0);
> print('i'=i,
> 'degree(u1,X)'=degree(u1,X), degree(u1,X)>t-1,
> 'S'=a,'s'[i]=s0, 'B'= b, 'r'[i]=r0, 'u'[i]=u0);
> od;
> printf("les valeurs de [u,r, s] sont :");
> u2:=sort(Expand(u2) ,[alpha,X]) mod 3;
> s2:=sort(Expand(s2) ,[alpha,X]) mod 3;
> r2:=sort(Expand(r2) ,[alpha,X]) mod 3;
> print('u'=u2);print('s'=s2);print('r'=r2);
> return [u2,s2, r2];
> end proc:

```

Warning, 'b' is implicitly declared local to procedure 'BerlMa'

> S:=alpha^4+alpha^3\*X+alpha^4\*X^2+0\*X^3;dp:=5;BerlMa(S,dp);

$$S := \alpha^4 + \alpha^3 X + \alpha^4 X^2$$

$$dp := 5$$

$$i = 0, 1 < 2$$

$i = 1, \text{degree}(u1, X) = 1, 1 < 1, S = \alpha^4 + \alpha^3 X + \alpha^4 X^2, s_1 = 1, B = X^4, r_1 = 0,$

$$u_1 = \alpha^4 + \alpha^3 X + \alpha^4 X^2$$

les valeurs de [u,r, s] sont :

$$u = 2\alpha + 2 + 2X$$

$$s = X^2 + 2X + 2X\alpha + \alpha + 1$$

$$r = 1$$

$$[2\alpha + 2 + 2X, X^2 + 2X + 2X\alpha + \alpha + 1, 1]$$

> Expand((2\*alpha-1)/alpha^3) mod 3;

$$1$$

> Expand((2\*X+alpha^3)) mod 3;

$$2\alpha + 2 + 2X$$

> Expand((X^2+alpha^3\*X+alpha^7)) mod 3;



$$X^2 + 2X + 2X\alpha + \alpha + 1$$

> X^2+2\*X+2\*X\*alpha+alpha+1;

$$X^2 + 2X + 2X\alpha + \alpha + 1$$

> Factor(X^2+alpha^3\*X+alpha^7,alpha) mod 3;

$$(X + \alpha + 2)(X + \alpha)$$

> f:=X^2+alpha^3\*X+alpha^7;

> for i from 1 to 8 do

> if Eval(f, X=alpha^(-i)) mod 3 = 0 then print(i) fi

> od ;

$$f := X^2 + \alpha^3 X + \alpha^7$$

3

6

**Exemple : l'identité de Bezout pour les polynomes pseudo-aléatoires sur  $\mathbb{F}_{3^4}$ , disponible à l'adresse cachée : <http://www-fourier.ujf-grenoble.fr/~panchish/04mag-maple> dans le fichier 4mag-11bezout-poly.mws**

```
> restart ;
```

```
> with(linalg) ;
```

Warning, the protected names norm and trace have been redefined and unprotected

```
> irr34 := op(1, select(has, Factor(x^81-x) mod 3, 4)) ;
```

$$irr34 := x^4 + x + 2$$

Le polynôme irréductible obtenu par ce procédé n'est pas forcément le même d'une session à l'autre.

On aliasse  $\alpha$  à une racine de ce polynôme dans une extension de  $F_3$ , de sorte que  $\text{GF}(3^4) = F_3[\alpha]$ .

```
> alias(alpha = RootOf(irr34) mod 3) ;
```

$\alpha$

La procédure **rnd3** génère un entier modulo 3 pseudo-aléatoire.

```
> rnd3 := rand(0..2) :
```

```
> seq(rnd3(), i = 1..5) ;
```

0, 2, 0, 2, 1

La procédure **rnd34** génère un élément pseudo-aléatoire de  $\text{GF}(2^4)$ .

```
> rnd34 := () -> add(rnd3()*alpha^i, i=0..3):
```

$$2 + 2\alpha + 2\alpha^2 + \alpha^3, 1 + \alpha, 2 + 2\alpha + \alpha^2, 2\alpha^2, \alpha^2$$

La procédure **rndP34**(n) génère un polynôme pseudo-aléatoire unitaire de  $\text{GF}(2^4)$  de degré n.

```
> rndP34 := proc(n)
> RETURN(sort(X^n + add(rnd34()*X^i, i=0..(n-1))))
> end :
> rndP34(5);
```

$$X^5 + \alpha^2 X^4 + 2\alpha^2 X^3 + (2 + 2\alpha + \alpha^2) X^2 + (1 + \alpha) X + 2 + 2\alpha + 2\alpha^2 + \alpha^3$$

```

> Bezout34 := proc( P::anything,
> A::anything,
> B::anything)
> local gcd, U, V ;
> gcd := gcdex(A, B, X, 'U', 'V') ; #A*U+B*V=gcd
> if gcd <> 1 then ERROR(
> sprintf("les polynômes %a et %a
> ne sont pas premiers entre eux.",
> A, B))
> fi ;
> RETURN(P, rem(P*U, B, X), rem(P*V, A, X)) #P, A*U+B*V=P
> end :
> _seed := 1925 :
> A, B, P:= X^(4), rndP34(2), rndP34(2); G := gcdex(A, B, X, 'U', 'V')
> ;#G=pgcd(A,B), A*U+B*V=P

```

$A, B, P := X^4, X^2 + (\alpha + 2\alpha^2)X + \alpha + \alpha^2 + \alpha^3, X^2 + (1 + \alpha + 2\alpha^3)X + 1 + \alpha^2 + 2\alpha^3$

$G := 1$

```

> P, U, V:= Bezout34(P, A, B);#A*U+B*V=P

```

$$\begin{aligned}
P, U, V &:= X^2 + (1 + \alpha + 2\alpha^3)X + 1 + \alpha^2 + 2\alpha^3, \\
&- \frac{(\alpha^3 + 6\alpha^4 - 8\alpha^6 - 2\alpha^7 + 1 + 2\alpha - \alpha^2 + 2\alpha^5 + 8\alpha^8)X}{(1 + 19\alpha^4 + 16\alpha^5 + 10\alpha^6 + 4\alpha^7 + \alpha^8 + 4\alpha + 10\alpha^2 + 16\alpha^3)\alpha^2} \\
&- \frac{-4\alpha^4 + 8\alpha^5 - 3\alpha^7 - 20\alpha^8 + 17\alpha^6 + 2\alpha^3 + 3\alpha^2 - 1 + 16\alpha^{10}}{(1 + 19\alpha^4 + 16\alpha^5 + 10\alpha^6 + 4\alpha^7 + \alpha^8 + 4\alpha + 10\alpha^2 + 16\alpha^3)\alpha^2}, \\
&\frac{(\alpha^3 + 6\alpha^4 - 8\alpha^6 - 2\alpha^7 + 1 + 2\alpha - \alpha^2 + 2\alpha^5 + 8\alpha^8)X^3}{(2\alpha + 3\alpha^2 + \alpha^4 + 2\alpha^3 + 1)^2\alpha^2} \\
&- \frac{(4\alpha^7 - 2\alpha^6 - 3\alpha^5 + 2\alpha^4 + \alpha^3 + 1)X^2}{(\alpha^2 + 1 + \alpha)(2\alpha + 3\alpha^2 + \alpha^4 + 2\alpha^3 + 1)\alpha^2} + \frac{(2\alpha^3 - 2\alpha^2 - \alpha + 1)\alpha X}{(\alpha^2 + 1 + \alpha)^2} \\
&+ \frac{1 + \alpha^2 + 2\alpha^3}{\alpha(\alpha^2 + 1 + \alpha)}
\end{aligned}$$

> Expand(P - U\*A - V\*B) mod 3;

0

> G, U, V:= Bezout34(G, A, B) :#A\*U+B\*V=gcd

> Expand(G - U\*A - V\*B) mod 3;

0

```
> un,tn,sn:=Bezout34(G, X^4,
> B):#A=X^(d'-1),B=S(X),un=G(X),tn=U(X),sn=V(X)
> un:=sort(Expand(un) ,[alpha,X]) mod 3;
```

$un := 1$

```
> tn:=sort(Expand(tn) ,[alpha,X]) mod 3;
```

$tn := 2X + 2X\alpha + X\alpha^2 + 2X\alpha^3 + 2\alpha^3 + \alpha^2$

```
> sn:=sort(Expand(sn) ,[alpha,X]) mod 3;
```

$sn := X^3 + X^3\alpha + 2X^3\alpha^2 + X^3\alpha^3 + X\alpha + 2X\alpha^2 + X\alpha^3 + X^2 + 2X^2\alpha + X^2\alpha^2 + 2\alpha^2 + \alpha + 1 + \alpha^3$

```
> Expand(G - tn*A - sn*B) mod 3;
```

0

```

> Bezout:=proc(a::anything,b::anything)
> local A,B,u0,u1,u2,v0,v1,v2,d0,d1,r,t,i;
> #if a<0 then a:=-a;
> #else fi;
> #if b<0 then b:=-b;
> #else fi;
> u0:=1+0*X:v0:=0*X: u1:=0*X:v1:=1+0*X:
> d0:=a; d1:=b;r:=b;i:=0;
> while(d1<>0) do
> i:=i+1;t:=quo( d0,d1,X );r:=rem( d0,d1,X );
> d0:=d1; d1:= r;
> u2:=u0-t*u1;
> v2:=v0-t*v1;
> u0:=u1;u1:= u2;
> v0:=v1;v1:= v2;
> #printf("i=%d, (%a)*(%a)+(%a)*(%a)=%a\n"
> #,i,a,u0 ,b, v0, d0);
> u0:=sort(Expand(u0) ,[alpha,X]) mod 3;
> v0:=sort(Expand(v0) ,[alpha,X]) mod 3;
> d0:=sort(Expand(d0) ,[alpha,X]) mod 3;
> print('i'=i,
> 'A'*'u'[i]+'B'*'v'[i]='d'[i],
> 'A'=a,'u'[i]=u0, 'B'= b, 'v'[i]=v0, 'd'[i]=d0);
> od;
> printf("les valeurs de [d,u, v] sont :");
> print('d'=d0);print('u'=u0);print('v'=v0);
> return [d0,u0, v0];
> end proc:

```

> A, B:= rndP34(2), rndP34(2);Bezout(A,B);

$$A, B := X^2 + (2 + \alpha + \alpha^3)X + 1 + \alpha + 2\alpha^3, X^2 + (2\alpha^3 + \alpha^2 + 2\alpha)X + 1 + 2\alpha^3$$

$$i = 1, Au_1 + Bv_1 = d_1, A = X^2 + (2 + \alpha + \alpha^3)X + 1 + \alpha + 2\alpha^3, u_1 = 0,$$

$$B = 2\alpha^3 + X^2 + (2\alpha^3 + \alpha^2 + 2\alpha)X + 1, v_1 = 1,$$

$$d_1 = 2\alpha^3 + X^2 + 2X\alpha^3 + X\alpha^2 + 2\alpha X + 1$$

$$i = 2, Au_2 + Bv_2 = d_2, A = X^2 + (2 + \alpha + \alpha^3)X + 1 + \alpha + 2\alpha^3, u_2 = 1,$$

$$B = 2\alpha^3 + X^2 + (2\alpha^3 + \alpha^2 + 2\alpha)X + 1, v_2 = 2,$$

$$d_2 = \alpha + 2X\alpha^3 + 2X\alpha^2 + 2\alpha X + 2X$$

$$i = 3, Au_3 + Bv_3 = d_3, A = X^2 + (2 + \alpha + \alpha^3)X + 1 + \alpha + 2\alpha^3, u_3 = 2\alpha^3 + \alpha^2 + 1 + X\alpha^3,$$

$$B = 2\alpha^3 + X^2 + (2\alpha^3 + \alpha^2 + 2\alpha)X + 1, v_3 = \alpha^3 + 2\alpha^2 + 2X\alpha^3, d_3 = 2\alpha$$

les valeurs de [d,u, v] sont :

$$d = 2\alpha$$

$$u = 2\alpha^3 + \alpha^2 + 1 + X\alpha^3$$

$$v = \alpha^3 + 2\alpha^2 + 2X\alpha^3$$

$$[2\alpha, 2\alpha^3 + \alpha^2 + 1 + X\alpha^3, \alpha^3 + 2\alpha^2 + 2X\alpha^3]$$



EXEMPLE. Soit  $\beta$  un élément primitif de  $\mathbb{F}_{16}$  racine du polynôme  $x^4 + x + 1$  sur  $\mathbb{F}_2$ . Alors

$$m^{(1)}(x) = m^{(2)}(x) = m^{(4)}(x) = m^{(8)}(x) = x^4 + x + 1$$

$$m^{(3)}(x) = m^{(6)}(x) = m^{(9)}(x) = m^{(12)}(x) = x^4 + x^3 + x^2 + x + 1$$

Ainsi, un code de polynôme générateur

$$g(x) = m^{(1)}(x)m^{(3)}(x) = 1 + x^4 + x^6 + x^7 + x^8$$

est un code BCH dont les paramètres sont  $b = 1, d' = 5 = d, q = 2, n = 15$ . Sa dimension est  $k = 7$  et un polynôme de contrôle est

$$h(x) = (x^{15} - 1)/g(x) = 1 + x^4 + x^6 + x^7$$

On construit une matrice génératrice  $G$  dont la  $i^{\text{ème}}$  ligne est le vecteur  $x^{i-1}g(x)$ ,  $1 \leq i \leq k$ . On obtient

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

Si le message reçu est  $v = (100100110000100)$ , soit encore  $v(x) = 1 + x^3 + x^6 + x^7 + x^{12}$ .  
On calcule alors les composantes du syndrome :

$$S_1 = S_2 = S_4 = v(\beta) = 1, S_3 = v(\beta^3) = \beta^4$$

Il apparaît alors que le plus grand système linéaire d'équations des inconnues  $\tau_i$  est de la forme

$$\begin{array}{l} S_2\tau_1 + S_1\tau_2 = S_3 \\ S_3\tau_1 + S_2\tau_2 = S_4 \end{array} \iff \begin{array}{l} S_1\tau_2 + S_2\tau_1 = S_3 \\ S_2\tau_2 + S_3\tau_1 = S_4 \end{array}$$

qui s'écrit également

$$\begin{array}{l} \tau_1 + \tau_2 = \beta^4 \\ \beta^4\tau_1 + \tau_2 = 1 \end{array}$$

et dont la matrice est régulière. Il vient

$$\begin{array}{l} \tau_1 = 1 \\ \tau_2 = \beta \end{array}$$

$\tau_0 = 1$  par définition, et on a donc  $s(x) = 1 + x + \beta x^2$ .

En testant les différentes puissances de  $\beta$ , on trouve

$$\eta_1^{-1} = \beta^8, \eta_2^{-1} = \beta^6,$$

et on a donc

$$\begin{aligned}\eta_1 &= \beta^7 \\ \eta_2 &= \beta^9\end{aligned}$$

ainsi, le polynôme erreur est

$$e(x) = x^7 + x^9$$

et on peut décoder

$$\begin{aligned}w(x) &= v(x) - e(x) \\ &= (1 + x^3 + x^6 + x^7 + x^{12}) - (x^7 + x^9) \\ &= 1 + x^3 + x^6 + x^9 + x^{12}\end{aligned}$$

Le message émis était  $w = (100100100100100)$ .

Pour retrouver le message original, il suffit de diviser  $w(x)$  par  $g(x)$ . Il vient

$$a(x) = w(x)/g(x) = 1 + x^3 + x^4 \Rightarrow a = (1001100)\text{.}\textcolor{blue}{\text{!}}$$

## 6.1 Rendement, taux de correction et domaine de codes

Soit  $C \subset F^n$  un code de cardinal  $\text{Card}(C) = M$  sur l'alphabet  $F$  de  $\text{Card}(F) = q$  de distance  $d = d(C)$  et la capacité de correction  $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ . Nous avons vu la définition 1.4 du rendement et du taux de correction : soit  $C$  un  $[n, k, d]_q$ -code, alors la **vitesse de transmission** (le "**rendement**" ou "information rate" en anglais) de  $C$  est le rapport  $R = k/n$ ,  $\delta = d/n$  est la **distance relative** (ou le "**taux de correction**") de  $C$ .

Un bon code est un code corrigeant de nombreuses erreurs compte tenu de sa longueur tout en ayant une vitesse de transmission la plus élevée possible, ce qui correspond à des paramètres  $R$  et  $\delta$  assez grands dans  $[0, 1]$ . Rappelons (voir Définition 1.9) qu'une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes est dite bonne s'il existe et positives les limites

$$\lim_{i \rightarrow \infty} \frac{k_i}{n_i} = R > 0, \lim_{i \rightarrow \infty} \frac{d_i}{n_i} = \delta > 0,$$

On a vu que avec toute bonne famille on peut faire tendre vers 1 la probabilité de transmission correcte. En réalité, on utilise  $R \approx 0,1 \sim 0,95$ ,  $n \approx 300 \sim 1000$ , et  $\delta > 2p$ .

Toutefois, ces paramètres  $\delta$  et  $R$  sont liés par des relations dites **asymptotiques**.

Nous avons déjà établi quelques relations dans Section 2 :

1) BORNE DE SINGLETON (voir Théorème 2.6). Soit  $C = \text{Im}E$  un  $[n, k, d]_q$ -code,  $E : F^k \rightarrow F^n$ . Alors

$$k \leq n - d + 1 \iff R \leq 1 - \delta + 1/n$$

1') BORNE DE SINGLETON ASYMPTOTIQUE (voir Théorème 2.8 ). Soit  $\{C_i\}$  une famille des  $[n_i, k_i, d_i]_q$ -codes, on pose

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

alors

$$R \leq 1 - \delta$$

2) BORNE DE HAMMING (voir Théorème 2.3). Soit  $C \subset F^n$  un code sur l'alphabet  $F$  de  $\text{Card}(F) = q$  de distance  $d = d(C)$  et la capacité de correction  $t = \lfloor \frac{d-1}{2} \rfloor$ . Alors

$$\text{Card}(C) \sum_{i=0}^t (q-1)^i \binom{n}{i} \leq q^n, \text{ donc } V_q(n, t) \leq q^n / \text{Card}(C).$$

2') BORNE DE HAMMING ASYMPTOTIQUE voir Théorème 2.11. Soit  $\{C_i\}$  une famille des  $[n_i, k_i, d_i]_q$ -codes, on pose

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

alors  $\boxed{R \leq 1 - H_q(\delta/2)}$  où  $H_q(\delta)$  est la fonction entropie  $q$ -aire définie sur  $[0, (q-1)/q]$  par

$$H_q(0) = 0, H_q(\delta) = \delta \log_q(q-1) - \delta \log_q(\delta) - (1-\delta) \log_q(1-\delta) \text{ pour } 0 \leq \delta \leq (q-1)/q$$

Rappelons la signification géométrique de l'entropie  $q$ -aire :  $H_q(\delta)$  est la proportion logarithmique du volume  $V_q(n, d)$  de la boule de Hamming de rayon relatif  $\delta$  :

$$H_q(\delta) = \lim_{n \rightarrow \infty} \frac{\log_q(V_q(n, d))}{n}, \quad \text{si } \frac{d}{n} \rightarrow \delta.$$

Maintenant on va obtenir **des meilleurs bornes** (supérieures et inférieures), pour décrire le **domaine de codes**, où se trouvent les **bons codes**.

Soit  $\mathcal{C}_q$  l'ensemble des codes sur  $\mathbb{F}_q$  et  $V_q$  l'image de l'application

$$\mathcal{F} : \mathcal{C}_q \longrightarrow [0, 1] \times [0, 1], \quad \mathcal{F}(C) = (\delta, R)$$

On note  $U_q$  l'ensemble des points d'accumulation de  $V_q$ . Les codes de  $V_q \setminus U_q$  sont dits **isolés**.

On pose

$$A_q(n, d) = \max \left\{ \text{Card}(C) = M \mid \text{il existe un code } C \subset \mathbb{F}_q^n \text{ de distance } d = d(C) \right\},$$

$$\alpha_q(\delta) := \limsup_{\substack{n \rightarrow \infty, \\ \frac{d}{n} \rightarrow \delta}} n^{-1} \log_q A_q(n, d),$$

Un code est dit **optimal** s'il atteint la borne  $\alpha_q(\delta)$ .

Un théorème de Yu.I.Manin (voir [Ts-V]) montre que  $\alpha_q(\delta)$  est **continue** avec  $\alpha_q(0) = 1$ ,  $\alpha_q(\delta) = 0$  pour  $(q-1)/q \leq \delta \leq 1$  et  $\alpha_q(\delta)$  est **décroissante** dans  $0 \leq \delta \leq (q-1)/q \leq 1$ .

On montrera en particulier :

– BORNE DE PLOTKIN ASYMPTOTIQUE

$$\alpha_q(\delta) \leq \max \left( 1 - \frac{q}{q-1} \delta, 0 \right)$$

– BORNE DE GILBERT-VARSHAMOV ASYMPTOTIQUE Pour tout  $\delta$  dans l'intervalle  $]0, (q-1)/q[$  il existe une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes, telle que

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

et

$$\alpha_q(\delta) \geq R \geq 1 - H_q(\delta),$$

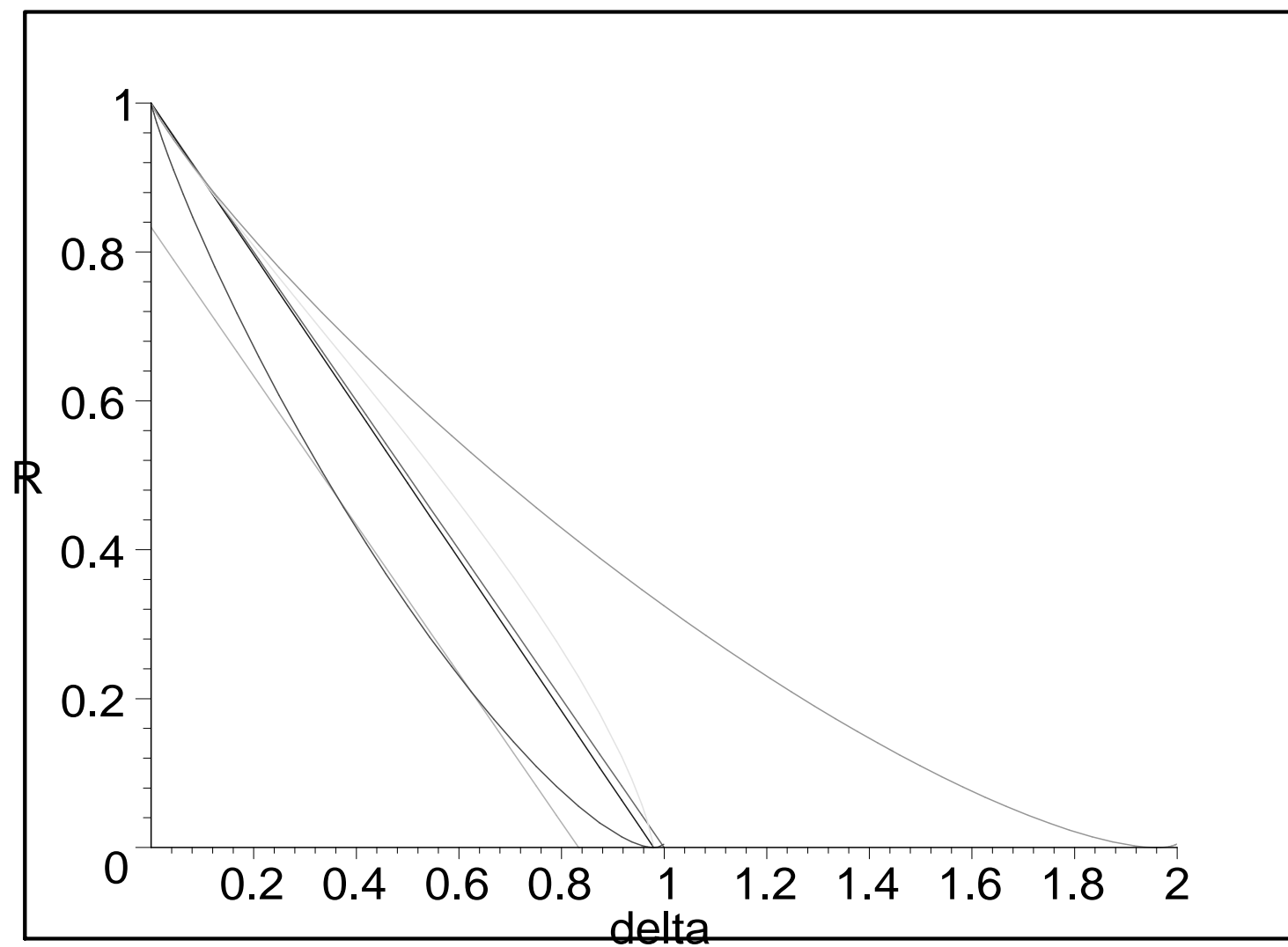
c'est à dire on a

$$\alpha_q(\delta) \geq 1 - H_q(\delta)$$

où  $H_q(\delta)$  est la fonction entropie  $q$ -aire



REPRÉSENTATION GRAPHIQUE DES BORNES : pour  $q = 49$  :



## 6.2 Borne de Plotkin

THÉORÈME 6.1 (Borne de Plotkin). *Soit  $C$  un code linéaire de longueur  $n$  et de dimension  $k$  sur  $\mathbb{F}_q$ , d'écart  $d$ . Alors*

$$d \leq \frac{nq^{k-1}(q-1)}{q^k-1} = n \frac{q-1}{q} \left(1 + \frac{1}{q^k-1}\right). \quad (*)$$

*Preuve :* Soit  $1 \leq i \leq n$  tel que  $C$  contienne un mot dont la  $i^{\text{ème}}$  coordonnée est non nulle. Soit  $D$  le sous-espace de  $C$  constitué des mots codes dont la  $i^{\text{ème}}$  coordonnée est nulle. On a  $|C/D| = q$ , donc, comme  $|C/D| = |C|/|D|$ ,  $|D| = q^{k-1}$ . Il vient alors que la somme des poids des mots codes est  $\leq nq^{k-1}(q-1)$ . Comme  $C$  comporte  $q^k - 1$  mots de poids non nul, on obtient la relation voulue :

$$d(q^k - 1) \leq nq^{k-1}(q-1).$$

REMARQUE. Soit  $M = \text{Card } C = q^k$ , alors il est commode d'écrire l'inégalité (\*) sous la forme : si  $\frac{d}{n} \neq \frac{q-1}{q}$ , alors

$$M \leq \left( \frac{d}{n} \frac{q}{q-1} - 1 \right)^{-1} + 1.$$

THÉORÈME 6.2 (BORNE DE PLOTKIN ASYMPTOTIQUE). Soit  $\{C_i\}$  une famille des  $[n_i, k_i, d_i]_q$ -codes, on pose

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

alors

$$R \leq \max \left( 1 - \frac{q}{q-1} \delta, 0 \right)$$

*Preuve* Soit  $\{C_i\}$  une famille des  $[n_i, k_i, d_i]_q$ -codes, on pose

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

alors il y a **trois possibilités** :

$$\boxed{\delta > \frac{q-1}{q}, \delta < \frac{q-1}{q}, \text{ ou } \delta = \frac{q-1}{q}.}$$

1) PREMIER CAS :  $\delta > \frac{q-1}{q}$ . On écrit l'inégalité du Théorème 6.1 :

$$d_i \leq \frac{n_i q^{k_i-1} (q-1)}{q^{k_i} - 1} \Rightarrow n_i \cdot \frac{q-1}{q} \cdot \left(1 + \frac{1}{q^{k_i} - 1}\right) \geq d_i.$$

Ceci implique

$$q^{k_i} \leq \left( \frac{d_i}{n_i} \frac{q}{q-1} - 1 \right)^{-1} + 1 \rightarrow \frac{1}{\delta \frac{q}{q-1} - 1} + 1 > 0 \text{ (une constante indépendant de } n), \quad (6.1)$$

puisque  $\delta > \frac{q-1}{q}$ .

Donc  $k_i$  sont **uniformement bornés**, cela signifie que

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i} = 0.$$

2) DEUXIÈME CAS :  $0 < \delta < \frac{q-1}{q}$ . Dans ce cas on peut supposer : pour  $n_i$  assez grand

$$\frac{d_i}{n_i} = \delta_i \rightarrow \delta < \frac{q-1}{q} \Rightarrow \frac{d_i}{n_i} - \frac{1}{n_i} < \frac{q-1}{q} \Rightarrow d_i - 1 < n_i \cdot \frac{q-1}{q}.$$

On pose

$$n'_i = \left\lfloor \frac{q(d_i - 1)}{q - 1} \right\rfloor \leq \frac{q(d_i - 1)}{q - 1} \left( \Rightarrow n'_i \frac{q-1}{q} \leq d_i - 1 \right).$$

Ensuite,  $d_i = \delta_i n_i \rightarrow \infty$  donc

$$0 < n'_i < n_i$$

puisque

$$\begin{aligned} n'_i &= \left\lfloor \frac{q(d_i - 1)}{q - 1} \right\rfloor \geq (d_i - 1) \cdot \frac{q}{q - 1} - 1 > 0, \\ n'_i &= \left\lfloor \frac{q(d_i - 1)}{q - 1} \right\rfloor \leq (d_i - 1) \cdot \frac{q}{q - 1} < n_i \end{aligned}$$

Pour tout  $n = n_i$  on considère les  $n - n'$  derniers symboles des mots du code. Il existe un sous-ensemble  $C' = C(\xi) \subset C$  formé par les mots avec les mêmes  $n - n'$  derniers symboles  $\xi \in \mathbb{F}_q^{n-n'}$ , et avec

$$M' := \text{Card}(C') \geq M \cdot q^{n'-n}.$$

Ainsi que  $C'$  n'est pas un sous-espace vectoriel de  $C$ , on peut fixer un  $c'_0 \in C'$ , alors l'ensemble de toutes les différences

$$C_{n'} = \left\{ c' - c'_0 \mid c' \in C' \right\} = \left\{ c = (c_1, \dots, c_{n'}, 0, \dots, 0) \mid c' \in C' \right\} \subset C,$$

est un sous-espace vectoriel de  $C$  ayant le même cardinal que  $C'$ . On utilise encore le théorème 6.1 pour le sous-espace vectoriel  $C'$  de  $C$  donc

$$d' \leq \frac{n'q^{k'-1}(q-1)}{q^{k'}-1} \Rightarrow n' \cdot \frac{q-1}{q} \cdot \left(1 + \frac{1}{q^{k'-1}}\right) \geq d'.$$

Ceci résulte :

$$\begin{aligned} M \cdot q^{n'-n} &\leq M' < \left( \frac{d'}{n'} \frac{q}{q-1} - 1 \right)^{-1} + 1 \\ &= \frac{n'(q-1)}{d'q - n'(q-1)} + 1 \leq \frac{n'(q-1)}{dq - n'(q-1)} + 1 \leq \frac{n'(q-1)}{q} + 1 \leq d \leq n \end{aligned} \tag{6.2}$$

puisque  $d \leq n$ ,  $d' \geq d$  et

$$dq - n'(q-1) = dq - \left\lfloor \frac{q(d-1)}{q-1} \right\rfloor (q-1) \geq dq - q(d-1) = q$$

(ne pas confondre la notation  $d'$  avec la distance construite des codes BCH!).

Pour conclure, on prend  $\log_q$  de (6.2) :  $M = q^k = \text{Card}(C)$ , et

$$M \cdot q^{n'-n} < n \Rightarrow \boxed{k + n' - n \leq \log_q n,} \quad \frac{q(d-1)}{q-1} - 1 < n' \leq \frac{q(d-1)}{q-1}.$$

Il reste à diviser par  $n = n_i$  et passer à la limite  $n \rightarrow \infty$  :

$$k \leq n - n' + \log_q n \leq n - \frac{q(d-1)}{q-1} + \log_q n \Rightarrow \boxed{\frac{k}{n} \leq 1 - \frac{q}{q-1} \cdot \frac{d}{n} + \frac{\log_q n + 1}{n}}$$

$$\Rightarrow \boxed{R \leq 1 - \frac{q}{q-1} \delta}$$

3) TROISIÈME CAS (EN EXERCICE) :  $\delta = \frac{q-1}{q}$ .

Pour montrer que  $R = 0$  on raisonne par l'absurde. On suppose que

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i} > 0,$$

et on choisit une sous-suite des codes  $C_i$  avec  $k_i/n_i \geq R_0 > 0$ . On pose

$$n'_i = \left\lceil \frac{q(d_i - 1)}{q - 1} \right\rceil \leq \frac{q(d_i - 1)}{q - 1}.$$

Ensuite,  $d_i = \delta_i n_i \rightarrow \infty$  donc

$$0 < n'_i < n_i$$

puisque

$$n'_i = \left\lceil \frac{q(d_i - 1)}{q - 1} \right\rceil \geq (d_i - 1) \cdot \frac{q}{q - 1} - 1 > 0,$$

et pour voir que  $n'_i < n_i$  on remarque que par Théorème 6.1

$$\begin{aligned} n'_i &= \left\lceil \frac{q(d_i - 1)}{q - 1} \right\rceil \leq (d_i - 1) \cdot \frac{q}{q - 1} \leq \left( \frac{q - 1}{q} \cdot n_i \cdot \left( 1 + \frac{1}{q^{k_i} - 1} \right) - 1 \right) \cdot \frac{q}{q - 1} \\ n_i + \frac{n_i}{q^{k_i} - 1} - \frac{q}{q - 1} &< n_i, \quad \left( \frac{n_i}{q^{k_i} - 1} \rightarrow 0 \right) \end{aligned}$$



puisque

$$d_i \leq \frac{q-1}{q} \cdot \left(1 + \frac{1}{q^{k_i}-1}\right), \text{ et } q^{k_i} > q^{R_0 n_i} \Rightarrow \frac{n_i}{q^{k_i}-1} \rightarrow 0.$$

Maintenant on répète le raisonnement du deuxième cas. Pour tout  $n = n_i$  et  $n' = n'_i$ , on considère les  $n - n'$  derniers symboles des mots du code. Il existe un sous-ensemble  $C' = C(\xi) \subset C$  formé par les mots avec les mêmes  $n - n'$  derniers symboles  $\xi \in \mathbb{F}_q^{n-n'}$ , et avec

$$M' := \text{Card}(C') \geq M \cdot q^{n'-n}$$

Ainsi que  $C'$  n'est pas un sous-espace vectoriel de  $C$ , on peut fixer un  $c'_0 \in C'$ , alors l'ensemble de toutes les différences

$$C_{n'} = \left\{ c' - c'_0 \mid c' \in C' \right\} = \left\{ c = (c_1, \dots, c_{n'}, 0, \dots, 0) \mid c' \in C' \right\} \subset C,$$

est un sous-espace vectoriel de  $C$  ayant le même cardinal que  $C'$ . On utilise encore Théorème 6.1 pour le sous-espace vectoriel  $C'$  de  $C$  donc

$$d' \leq \frac{n' q^{k'-1} (q-1)}{q^{k'}-1} \Rightarrow n' \cdot \frac{q-1}{q} \cdot \left(1 + \frac{1}{q^{k'}-1}\right) \geq d'.$$

Ceci résulte :

$$\begin{aligned} M \cdot q^{n'-n} &\leq M' < \left( \frac{d'}{n'} \frac{q}{q-1} - 1 \right)^{-1} + 1 \\ &= \frac{n'(q-1)}{d'q - n'(q-1)} + 1 \leq \frac{n'(q-1)}{dq - n'(q-1)} \leq \frac{n'(q-1)}{q} \leq n \end{aligned} \quad (6.3)$$

puisque  $d' \geq d$  et

$$dq - n'(q-1) = dq - \left\lfloor \frac{q(d-1)}{q-1} \right\rfloor (q-1) \geq dq - q(d-1) = q$$

Pour conclure dans ce cas, on prend de nouveaux  $\log_q$  de (6.3) :

$$k + n' - n \leq \log_q n, \quad \frac{q(d-1)}{q-1} \leq n' < \frac{q(d-1)}{q-1} + 1.$$

Il reste à diviser par  $n = n_i$  et passer à la limite  $n \rightarrow \infty$  :

$$k \leq n - n' + \log_q n \leq n - \frac{q(d-1)}{q-1} + \log_q n \Rightarrow \frac{k}{n} \leq 1 - \frac{q}{q-1} \cdot \frac{d}{n} + \frac{\log_q n + 1}{n}$$

Mais dans ce cas  $\delta = \frac{q-1}{q}$ ,

$$\Rightarrow R \leq 1 - \frac{q}{q-1} \delta = 0.$$

### 6.3 Borne de Gilbert-Varshamov

THÉORÈME 6.3 (Borne de Gilbert-Varshamov) **Il existe un code linéaire** de dimension  $k$ , de longueur  $n$  sur  $\mathbb{F}_q$  et d'écart  $\geq d$  dès lors que

$$q^{n-k} > \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i = V_q(n-1, d-2).$$

PREUVE DU THÉORÈME 6.3 : Soit  $C$  un code comme décrit ci-dessus. Construisons sa matrice de correction  $H$ . On choisit pour la première colonne n'importe quel vecteur de  $\mathbb{F}_q^{n-k}$  non nul, puis pour la seconde un vecteur non multiple scalaire de la première, et ainsi de suite jusqu'à obtenir  $j-1$  colonnes dont  $d-1$  sont toujours linéairement indépendantes. Par combinaison linéaire de  $d-2$  ou moins de ces  $j-1$  colonnes, on peut former au plus

$$\sum_{i=0}^{d-2} \binom{j-1}{i} (q-1)^i = V_q(j-1, d-2).$$

colonnes.

Comme par combinaison linéaire de  $d - 2$  ou moins de ces  $j - 1$  colonnes, on peut former au plus

$$\sum_{i=0}^{d-2} \binom{j-1}{i} (q-1)^i = V_q(j-1, d-2)$$

colonnes, on obtient **une condition suffisante** pour l'existence d'une  $j^{\text{ème}}$  colonne cherchée. Elle est donnée pour un choix de  $j - 1 \leq n - 1$  colonnes

$$q^{n-k} > \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i = V_q(n-1, d-2)$$

(on obtient le nombre maximal possible des combinaisons linéaires de  $d-2$  colonnes choisis parmi  $n - 1$  colonnes). Ainsi, si l'inégalité du théorème est vérifiée, on peut trouver une  $j^{\text{ème}}$  colonne linéairement indépendante de toutes  $d - 2$  des  $j - 1$  premières colonnes, et ce jusqu'à obtenir  $H$  de rang  $n - k$ , ce qui fournit un code de distance  $\geq d$ .

REMARQUE. La borne de Gilbert-Varshamov reste valable pour **tous les codes** sous une forme légèrement plus faible (voir [vLi], p. 56, [Ste], p.35, [MW-S], p.34) : Il existe un code  $C \subset \mathbb{F}_q^n$  de distance minimale  $\geq d$  et de cardinale  $M = \text{Card}(C)$  dès lors que

$$q^n > V_q(n, d-1)M, \quad V_q(n, d-1) = \sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i.$$

Pour un tel code on suppose qu'il n'est pas possible de trouver un mot  $c$  en dehors de  $C$  de distance  $\geq d$ . Alors

$$q^n \leq V_q(n, d-1)M, \quad V_q(n, d-1) = \sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i.$$

(c'est-à-dire, le rayon de recouvrement est  $\leq d-1$ ). **Dans le cas contraire** on trouve un tel mot  $c$  et on l'ajout au code  $C$  (non-nécessairement linéaire), à partir du code trivial comportant un seul mot (on raisonne par récurrence sur  $M$ ). Ceci dit, il existe un code du cardinal  $M$  pour le nombre

$$M = \left\lceil \frac{q^n}{V_q(n, d-1)} \right\rceil - 1 \Rightarrow M < \frac{q^n}{V_q(n, d-1)} \leq M + 2.$$

BORNE DE GILBERT-VARSHAMOV ASYMPTOTIQUE (voir Théorème 6.3).

Rappelons qu'on a défini le rendement maximal

$$\alpha_q(\delta) := \limsup_{\substack{n \rightarrow \infty, \\ \frac{d}{n} \rightarrow \delta}} n^{-1} \log_q A_q(n, d),$$

pour le taux de correction  $\delta$  donné, où on pose

$$A_q(n, d) = \max \left\{ \text{Card}(C) = M \mid \text{il existe un code } C \subset \mathbb{F}_q^n \text{ de distance } d = d(C) \right\}.$$

Rappelons la signification géométrique de l'entropie  $q$ -aire :  $H_q(\delta)$  est la proportion logarithmique du volume  $V_q(n, d)$  de la boule de Hamming de rayon relatif  $\delta$  :

$$H_q(\delta) = \lim_{n \rightarrow \infty} \frac{\log_q(V_q(n, d))}{n}, \quad \text{si } \frac{d}{n} \rightarrow \delta.$$

THÉORÈME 6.4 (BORNE DE GILBERT-VARSHAMOV ASYMPTOTIQUE) *Pour tout  $\delta$  dans l'intervalle  $]0, (q-1)/q[$  il existe une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes, telle que*

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

*et*

$$R \geq 1 - H_q(\delta),$$

*c'est à dire, qu'on a*

$$\alpha_q(\delta) \geq 1 - H_q(\delta)$$

*où  $H_q(\delta)$  est la fonction entropie  $q$ -aire définie sur  $[0, (q-1)/q]$  par  $H_q(0) = 0$ ,  $H_q(\delta) = \delta \log_q(q-1) - \delta \log_q(\delta) - (1-\delta) \log_q(1-\delta)$  pour  $0 \leq \delta \leq (q-1)/q$ .*

PREUVE : Soit  $n_i \rightarrow \infty$  une suite arbitraire. On pose  $d_i = [\delta n_i]$ , et

$$k_i = [n_i - \log_q V_q(n_i - 1, d_i - 2)] - 1$$

Alors on a l'encadrement :

$$n_i - \log_q V_q(n_i - 1, d_i - 2) > k_i \geq n_i - \log_q V_q(n_i - 1, d_i - 2) - 2$$

donc  $k_i$  vérifie l'inégalité du Théorème 6.3 :

$$q^{n_i - k_i} > \sum_{i=0}^{d_i - 2} \binom{n_i - 1}{i} (q - 1)^i = V_q(n_i - 1, d_i - 2) \geq q^{n_i - k_i - 2}.$$

donc il existe un  $[n_i, k_i, d_i]_q$ -code  $C_i$ . Il reste à passer à la limite  $n_i \rightarrow \infty$  :

$$\frac{\log_q V_q(n_i - 1, d_i - 2)}{n_i} = \frac{\log_q V_q(n_i - 1, [\delta \cdot n_i] - 2)}{n_i} \rightarrow H_q(\delta), \quad \frac{k_i}{n_i} \rightarrow R \Rightarrow R \geq 1 - H_q(\delta).$$

Rappelons le corollaire 2.10 : Lorsque  $n \rightarrow \infty$ ,  $\frac{r}{n} \rightarrow \delta$ , on a

$$H_q(\delta) = \lim_{n \rightarrow \infty} \frac{\log_q V_q(n, r)}{n} = \lim_{n \rightarrow \infty} \frac{\log_q(V_q(n, r))}{\log_q(\text{Card}(F^n))}.$$



THÉORÈME 6.5 (BORNE DE BASSALYGO-ELIAS, SANS DÉMONSTRATION) *Pour tout code de dimension  $k$ , de longueur  $n$  et d'écart  $d$  sur  $F_q$  et pour tout entier  $w$  tel que  $1 \leq w \leq n$  tel que*

$$d - 2w + qw^2/(q - 1)n > 0$$

*on a*

$$n - k \geq \log_q \binom{n}{w} + w \log_q(q - 1) - \log_q d + \log_q r$$

PREUVE : voir [Ste], p.32.

THÉORÈME 6.6 (BORNE DE BASSALYGO-ELIAS ASYMPTOTIQUE , sans démonstration. Soit  $\{C_i\}$  une famille des  $[n_i, k_i, d_i]_q$ -codes, on pose

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

*alors*

$$R \leq \alpha_q(\delta) \leq R_{BE}, \text{ où } R_{BE} = 1 - H_q \left( \frac{q-1}{q} - \frac{(q-1) \sqrt{1 - \frac{q\delta}{q-1}}}{q} \right).$$

## 6.4 Borne de la géométrie algébrique (sans démonstration)

THÉORÈME 6.7 (TSFASMAN-ZINK-DRINFELD-VLADUT). *Il existe une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes provenant des courbes algébriques sur  $\mathbb{F}_q$ , telle que si l'on pose*

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i},$$

$$1) \quad R \geq 1 - \delta - \frac{1}{\sqrt{q} - 1}, \text{ c'est à dire,}$$

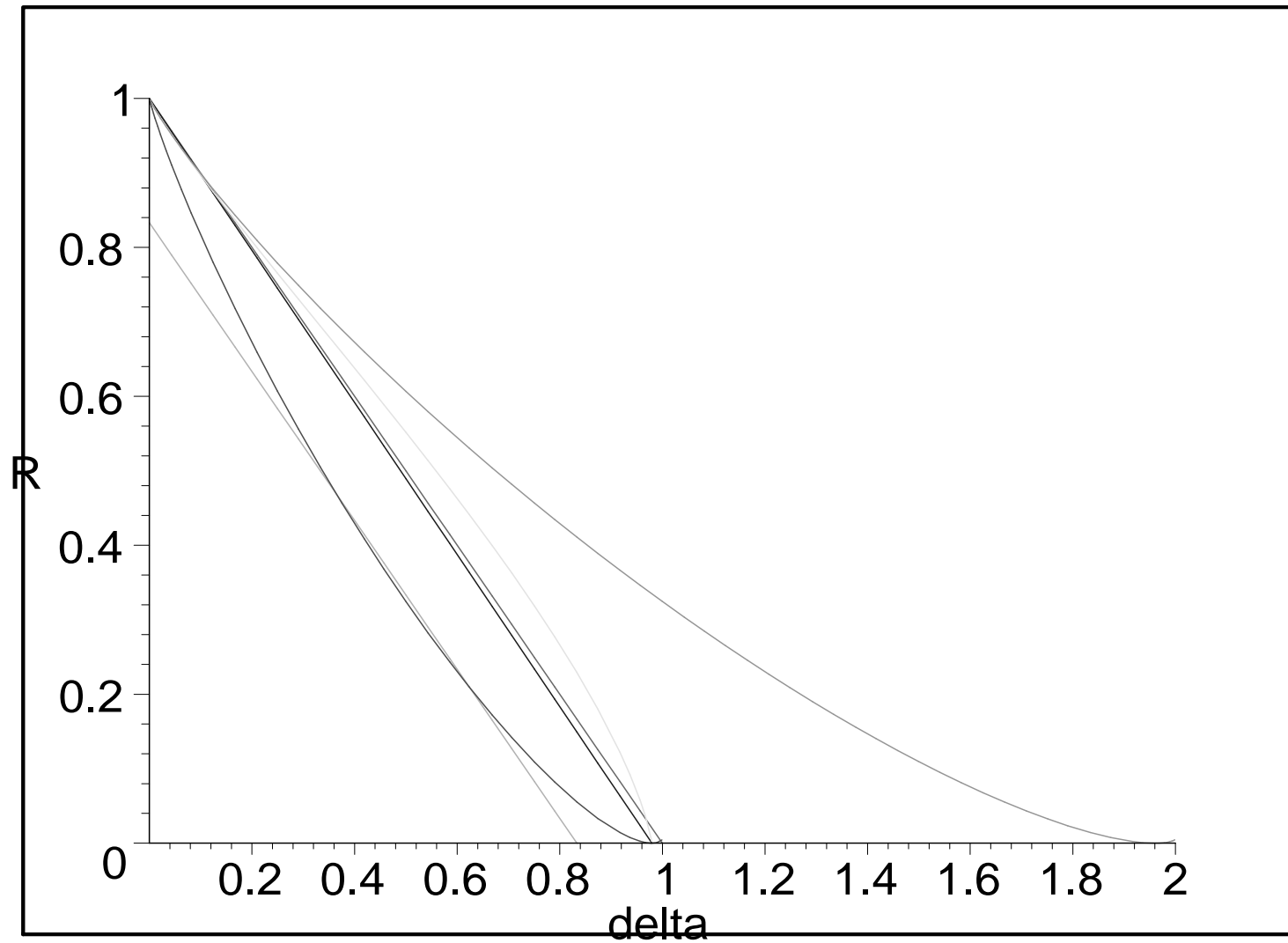
$$\alpha_q(\delta) \geq R \geq R_{GA} = 1 - \delta - \frac{1}{\sqrt{q} - 1}. \quad (6.4)$$

2) *Si  $q$  est un carré, la borne (6.4) est atteinte*

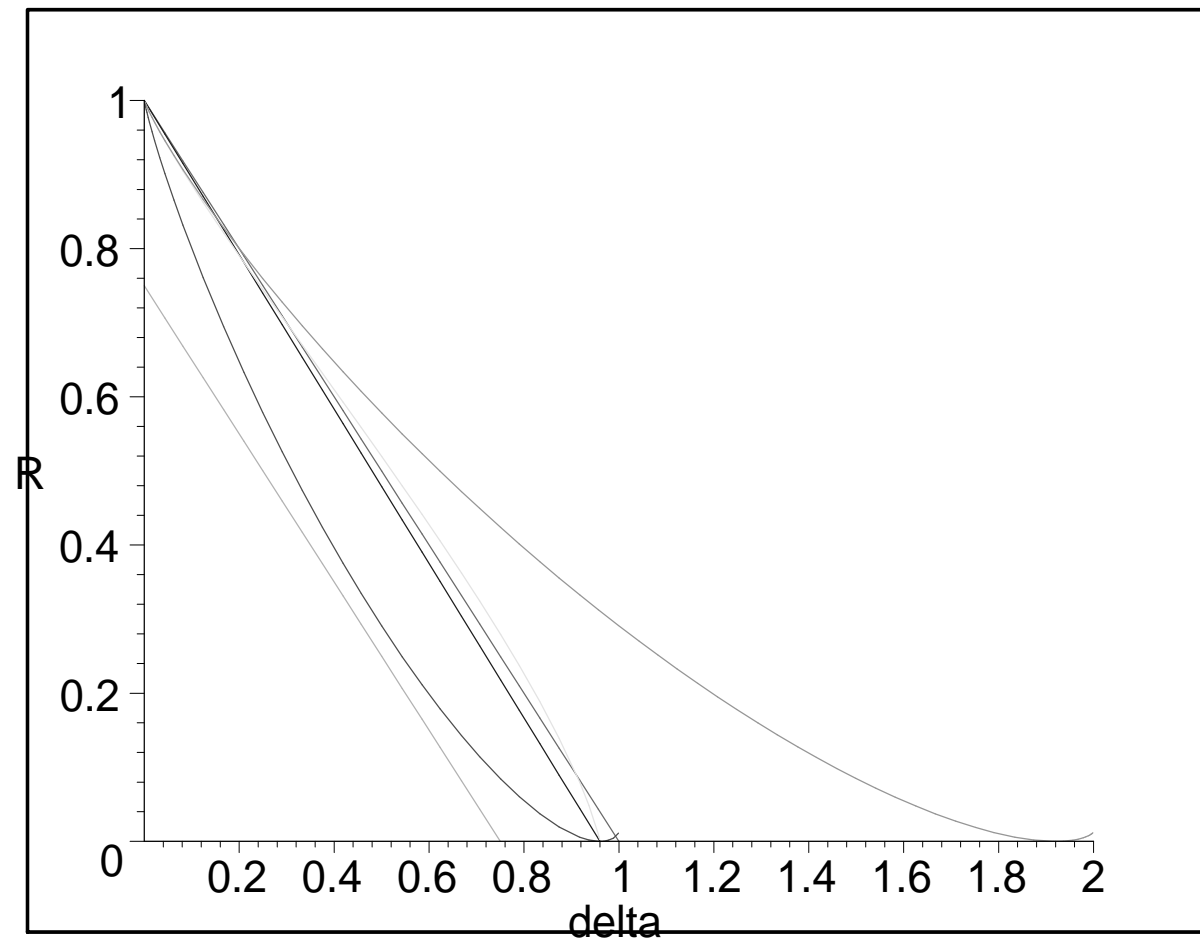
$$\limsup_{i \rightarrow \infty} \frac{d_i}{n_i} = \frac{1}{\sqrt{q} - 1}.$$

REMARQUE 6.8 *Si  $q$  est un carré, alors la borne inférieure (6.4) est meilleure que la borne de Gilbert-Varshamov asymptotique (voir le théorème 6.4, et la représentation graphique des bornes ci-dessous).*

REPRÉSENTATION GRAPHIQUE DES BORNES : 1) Pour  $q = 49$  :



2) Pour  $q = 25$  :



# Existence des bonnes familles et codes géométriques

## Rappel : bonne famille, voir Definition 1.9

Une famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes est dite **bonne** s'il existe et positive les limites

$$R = \lim_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \lim_{i \rightarrow \infty} \frac{d_i}{n_i}.$$

On montrera l'existence des bonnes familles, et même des familles excellentes :

DÉFINITION 6.9 Une bonne famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes est dite **excellente** s'il existe

$$R = \lim_{i \rightarrow \infty} \frac{k_i}{n_i} > 0, \delta = \lim_{i \rightarrow \infty} \frac{d_i}{n_i} \in ]0, (q-1)/q[,$$

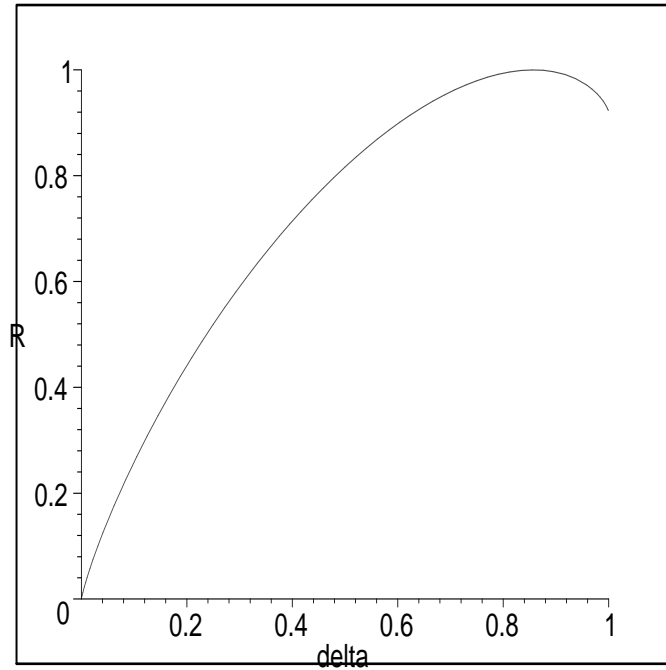
et

$$\boxed{R \geq 1 - H_q(\delta)}$$

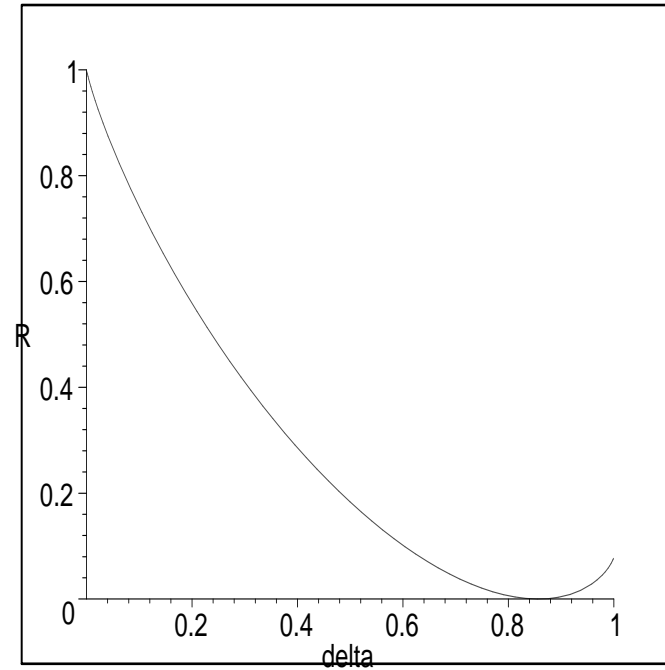
où  $H_q(\delta)$  est la fonction entropie  $q$ -aire définie sur  $[0, (q-1)/q]$  par  $H_q(0) = 0$ ,  
 $H_q(\delta) = \delta \log_q(q-1) - \delta \log_q(\delta) - (1-\delta) \log_q(1-\delta)$  pour  $0 \leq \delta \leq (q-1)/q$ .

## Représentation graphique : fonction d'entropie, $q = 7$

$$H_q(\delta) = \frac{\delta \log(q - 1)}{\log(q)} - \frac{\delta \log(\delta)}{\log(q)} - \frac{(1 - \delta) \log(1 - \delta)}{\log(q)}$$



$$R = H_q(\delta)$$



$$R = 1 - H_q(\delta)$$

Pour construire une telle famille on utilise

THÉORÈME 6.3 (Borne de Gilbert-Varshamov) **Il existe** un code **linéaire** de dimension  $k$ , de longueur  $n$  sur  $\mathbb{F}_q$  et d'écart  $\geq d$  dès lors que

$$q^{n-k} > V_q(n-1, d-2) = \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i. \quad (*)$$

(Rappelons qu'on a montré l'existence d'un tel code en construisant sa **matrice de contrôle**  $H \in M_{n-k,n}(\mathbb{F}_q)$  avec la propriété que toutes les  $d-1$  colonnes sont linéairement indépendantes. La partie gauche de (\*) est le nombre total des colonnes, et la partie droite de (\*) est le **nombre maximum** des combinaisons linéaires de  $d-2$  colonnes parmi les colonnes construites par récurrence; ceci dit que (\*) permet de choisir les colonnes d'une matrice  $H$  voulue)

## Passage à la limite

THÉORÈME 6.4 (Borne de Gilbert-Varshamov asymptotique) Pour tout  $\delta$  dans l'intervalle  $]0, (q-1)/q[$  **il existe une famille**  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes, telle que

$$R = \limsup_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \limsup_{i \rightarrow \infty} \frac{d_i}{n_i}, \text{ et } \boxed{R = 1 - H_q(\delta)},$$

PREUVE : Soit  $n_i \rightarrow \infty$  une suite arbitraire. On pose  $d_i = \lceil \delta n_i \rceil$ , et

$$k_i = \lceil n_i - \log_q V_q(n_i - 1, d_i - 2) \rceil - 1$$

Alors on a l'encadrement :

$$n_i - \log_q V_q(n_i - 1, d_i - 2) > k_i \geq n_i - \log_q V_q(n_i - 1, d_i - 2) - 2$$

donc  $k_i$  vérifie l'inégalité du Théorème 6.3 :

$$q^{n_i - k_i} > \sum_{i=0}^{d_i-2} \binom{n_i-1}{i} (q-1)^i = V_q(n_i-1, d_i-2) \geq q^{n_i - k_i - 2}.$$

donc il existe un  $[n_i, k_i, d_i]_q$ -code  $C_i$ .



Il reste à passer à la limite  $n_i \rightarrow \infty$  en utilisant l'encadrement ci-dessus :

$$\frac{n_i - k_i}{n_i} < \frac{\log_q V_q(n_i - 1, d_i - 2)}{n_i} \leq \frac{n_i - k_i - 2}{n_i}.$$

Rappelons le corollaire 2.10 : lorsque  $n \rightarrow \infty$ ,  $\frac{r}{n} \rightarrow \delta$ , on a

$$\lim_{n \rightarrow \infty} \frac{\log_q V_q(n, r)}{n} = H_q(\delta) \Rightarrow \frac{\log_q V_q(n_i - 1, d_i - 2)}{n_i} = \frac{\log_q V_q(n_i - 1, [\delta \cdot n_i] - 2)}{n_i} \rightarrow H_q(\delta),$$

$$\frac{k_i}{n_i} \rightarrow R \Rightarrow 1 - R \leq H_q(\delta) \leq 1 - R \Rightarrow \boxed{R = 1 - H_q(\delta)}.$$



## 7 Codes géométriques de Goppa. Systèmes affines et courbes algébriques

sur les corps finis

### 7.1 Systèmes affines et systèmes projectifs

#### 7.1.1. Systèmes affines.

Soit  $V$  un espace vectoriel sur  $\mathbb{F}_q$ ,  $\dim_{\mathbb{F}_q} V = k$ .

DÉFINITION 7.1 *Un système affine*

$$\mathcal{P} = \{P_1, \dots, P_n\} \subset V$$

*est un sous-ensemble dans  $V$  tel que  $\mathcal{P}$  n'est pas contenu dans aucun hyperplan  $H \subset V$ .*

On pose

$$d = d(\mathcal{P}) = n - \max_H \{ \text{Card}(\mathcal{P} \cap H) \mid H \text{ hyperplan} \} > 0$$

On considère l'application linéaire

$$E_{\mathcal{P}} : V^* \rightarrow \mathbb{F}_q^n, \quad E_{\mathcal{P}}(l) = (l(P_1), \dots, l(P_n))$$

alors on obtient le code

$$C = \text{Im}(E_{\mathcal{P}}) = E_{\mathcal{P}}(V^*) \subset \mathbb{F}_q^n$$

associé à  $\mathcal{P}$ . On voit que  $C$  est un  $[n, k, d]_q$ -code (voir la définition 1.4) : si  $c = (l_0(P_1), \dots, l_0(P_n))$  un mot de code  $C$  avec  $l_0 \in V^*, l_0 \neq 0$ , alors

$$\text{Card} \{i \in \{1, 2, \dots, n\} \mid l_0(P_i) \neq 0\} = n - \text{Card}(\{\mathcal{P} \cap H_0 \mid H_0 \text{ hyperplan } l_0 = 0\})$$

donc

$$d = \min_l \text{Card} \{i \in \{1, 2, \dots, n\} \mid l(P_i) \neq 0\}.$$

REMARQUE Pratiquement une matrice-génératrice de  $C$  est donnée par les coordonnées des points  $P_1, \dots, P_n$  dans une base de  $V$  : si  $l_1, \dots, l_k \in V^*$  les fonctions des coordonnées (la base duale d'une base de  $V$ ), alors

$$G = \begin{pmatrix} l_1(P_1) & \dots & \dots & l_1(P_n) \\ \dots & \dots & \dots & \dots \\ l_k(P_1) & \dots & \dots & l_k(P_n) \end{pmatrix} \quad (\text{ne pas confondre avec une matrice de contrôle !})$$

THÉORÈME 7.2 Soit  $Syst_q(n, k, d)$  l'ensemble des systèmes affines ci-dessus. Le groupe  $GL_{\mathbb{F}_q}(V)$  agit sur  $Syst_q(n, k, d)$ , et il y a une bijection

$$\varphi : Syst_q(n, k, d)/GL_{\mathbb{F}_q}(V) \xrightarrow{\sim} Codes([n, k, d]_q)$$

*Preuve* Pour  $g \in GL_{\mathbb{F}_q}(V)$  on pose

$$g \cdot \mathcal{P} = \{g \cdot P_1, \dots, g \cdot P_n\} \subset V$$

alors

$$P_i \in H \iff g \cdot P_i \in g \cdot H \text{ un autre hyperplan .}$$

D'autre part,

$$\text{Ker}(E_{\mathcal{P}}) = \{l \in V^* \mid (l(P_1), \dots, l(P_n))\} = \{0\} \Rightarrow E_{\mathcal{P}} \text{ est injective}$$

$$E_{g \cdot \mathcal{P}}(l) = (l(g \cdot P_1), \dots, l(g \cdot P_n)) = (g^*l(P_1), \dots, g^*l(P_n))E_{\mathcal{P}}(g^*l)$$

mais  $g^*$  est un  $\mathbb{F}_q$ -isomorphisme donc les systèmes  $\mathcal{P}$  et  $g \cdot \mathcal{P}$  donnent le même code  $\text{Im}(E_{\mathcal{P}})$ . L'application réciproque à  $\varphi$  :

$$\varphi^{-1} : C \mapsto \varphi^{-1}(C) = \{P_1, \dots, P_n\} \subset V = C^*, \quad P_i(c) = c_i, \quad i^{\text{ème}} \text{ coordonnée de } c = (c_1, \dots, c_n) \in C$$

### 7.1.2. Systèmes projectifs.

Soit  $n \geq 1$  un entier. On appelle **espace projectif** de dimension  $n$  sur un corps  $F$  et on note  $\mathbb{P}_F^n$  ou simplement  $\mathbb{P}^n$  l'ensemble des classes d'équivalence de  $(n+1)$ -uplets  $(x_0, \dots, x_n)$ ,  $x_i \in F$  pour  $0 \leq i \leq n$ ; sous la relation  $(x_0, \dots, x_n) \sim (\lambda_0 x_0, \dots, \lambda_n x_n)$  pour tout  $\lambda \in F^*$ . Une classe d'équivalence  $x$  est un point de  $\mathbb{P}^n$ .

DÉFINITION 7.3 *Un système projectif*

$$\mathcal{P} = \{P_1, \dots, P_n\} \subset \mathbb{P}_{\mathbb{F}_q}^{k-1}$$

est un sous-ensemble dans  $\mathbb{P}_{\mathbb{F}_q}^{k-1} = \mathbb{P}(V)$  tel que  $\mathcal{P}$  n'est pas contenu dans aucun hyperplan projectif  $H \subset \mathbb{P}_{\mathbb{F}_q}^{k-1}$ .

Soit

$$\mathcal{P}^\circ = \{P_1^\circ, \dots, P_n^\circ\} \subset V$$

un système affine dans  $V$  représentant  $\mathcal{P}$ . Tout autre système

$$\mathcal{P}_\lambda^\circ = \{\lambda_1 P_1^\circ, \dots, \lambda_n P_n^\circ\} \subset V$$

défini un code  $C_\lambda = \text{Im}(E_{\mathcal{P}_\lambda^\circ}) = \varphi(\mathcal{P}_\lambda^\circ)$  équivalent à  $C = \text{Im}(E_{\mathcal{P}^\circ}) = \varphi(\mathcal{P}^\circ)$  :

$$\begin{aligned} C &= \varphi(\mathcal{P}^\circ) = E_{\mathcal{P}^\circ}(V^*) = \{(l(P_1^\circ), \dots, l(P_n^\circ)) \mid l \in V^*\} \\ C_\lambda &= \varphi(\mathcal{P}_\lambda^\circ) = E_{\mathcal{P}^\circ}(V^*) = \{(\lambda_1 l(P_1^\circ), \dots, \lambda_n l(P_n^\circ)) \mid l \in V^*, \lambda_i \in \mathbb{F}_q^*\}. \end{aligned}$$

C'est-à-dire,  $C_\lambda = \sigma(C)$  pour la bijection linéaire diagonale donnée par la matrice  $\sigma = \text{diag}\{\lambda_1, \dots, \lambda_n\}$ .

THÉORÈME 7.4 *Soit  $\overline{Syst}_q(n, k, d)$  l'ensemble des systèmes projectifs ci-dessus. Le groupe  $\text{PGL}_{\mathbb{F}_q}(V)$  agit sur  $\overline{Syst}_q(n, k, d)$ , et il y a une bijection*

$$\overline{\varphi} : \overline{Syst}_q(n, k, d) / \text{PGL}_{\mathbb{F}_q}(V) \xrightarrow{\sim} \{Codes([n, k, d]_q)\} / (\text{mod } \sim)$$

où  $\sim$  désigne l'équivalence linéaire diagonale des  $[n, k, d]_q$  codes.

*Preuve* : en exercice.

## 7.2 Codes géométriques

On définit les **codes géométriques** à partir des sous-ensembles de

$$X = \mathcal{P} = \{P_1, \dots, P_n\} \subset \mathbb{P}_{\mathbb{F}_q}^{k-1}, \text{ et } \{P_1, \dots, P_n\} \subset \mathbb{F}_q^k.$$

Une matrice génératrice du code  $C = \overline{\varphi}(\mathcal{P})$  est donnée par des coordonnées des points  $P_i$  :

$$P = (a_{ij}), \quad P_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{kj} \end{pmatrix}, j = 1, \dots, n.$$

On pose

$$d = d(C) = d(X) = n - \max_H \{ \text{Card}(X \cap H) \mid H \text{ hyperplan} \} > 0$$

**REMARQUE 7.5** *Le nombre  $\text{Card}(X \cap H)$  est majoré par le "degré" de  $X$ . Pour une courbe plane  $X = X(\mathbb{F}_q) \subset \mathbb{P}_{\mathbb{F}_q}^2$  donnée comme l'ensemble des zéros d'un polynôme homogène de degré  $m$ ,  $H$  est une droite qui coupe  $X$  en maximum  $m$  points.*

EXEMPLE. CODES DE REED-MULLER D'ORDRE 1. Soit  $L_m = \mathbb{F}_q[T_1, \dots, T_m]_{d \leq 1}$  le  $\mathbb{F}_q$ -espace vectoriel de **tous les polynômes de degré  $\leq 1$  de  $m$  variables** sur  $\mathbb{F}_q$  (avec le polynôme nul), donc  $\dim_{\mathbb{F}_q} L_m = m + 1$ . On considère un sous ensemble

$$\mathcal{P} = \{P_1, \dots, P_n\} \subset \mathbb{F}_q^m$$

tel que

$$\forall l \in L_m (\forall j = 1, \dots, n, l(P_j) = 0 \Rightarrow l \equiv 0).$$

DÉFINITION 7.6

$$RM_q(1, n, m) = \text{Im} Ev, \quad Ev : L_m \rightarrow \mathbb{F}_q^n, \quad f \mapsto ((f(P_1), \dots, f(P_n)))$$

*est dit le **code de Reed-Muller d'ordre 1**.*

On vérifie que  $RM_q(1, n, m)$  est un  $[n, m + 1, n - q^{m-1}]$ -code : tout polynôme de degré  $\leq 1$  de  $m$  variables s'annule en  $\leq q^{m-1}$  points. Pour  $q = 2, n = 32, m = 5$  on obtient le code de l'Introduction.



EXERCICE. 1) Donner une version projective du code : on considère le  $\mathbb{F}_q$ -espace vectoriel  $L'_m$  de tous les polynômes homogènes de degré 1 de  $m + 1$  variables sur  $\mathbb{F}_q$  (avec le polynôme nul), donc  $\dim_{\mathbb{F}_q} L'_m = m + 1$ . On obtient un

$$\left[ n, m + 1, n - \frac{q^m - 1}{q - 1} \right]_q$$

-code pour tout

$$n \geq \frac{q^m - 1}{q - 1},$$

(voir [Ste], p. 46).

2) Montrer que pour

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

on obtient un code se trouvant sur la borne de Plotkin (voir Théorème 6.1) de paramètres

$$\left[ \frac{q^{m+1} - 1}{q - 1}, m + 1, q^m \right]_q.$$

### 7.3 Codes de Goppa rationnels

Ce sont des codes géométriques qui produisent une bonne famille des codes (dans le sens de la définition 1.9.)

Soit  $g(x)$  un polynôme *unitaire irréductible* sur  $\mathbb{F}_{q^m}$ ,  $L = \{\gamma_0, \gamma_1, \dots, \gamma_{n-1}\} \subset \mathbb{F}_{q^m}$  avec  $g(\gamma_i) \neq 0$  pour tout  $i = 0, 1, 2, \dots, n-1$ .

DÉFINITION 7.7 *Le code de Goppa rationnel  $\Gamma(L, g)$  est*

$$\Gamma(L, g) = \left\{ w = (w_0, w_1, \dots, w_{n-1}) \in \mathbb{F}_q^n \mid \sum_{i=0}^{n-1} \frac{w_i}{x - \gamma_i} \equiv 0 \pmod{g(x)} \right\} \quad (7.1)$$

(c'est-à-dire, la fraction à droite est de la forme d'une fraction irréductible  $\frac{a(x)}{b(x)}$  avec  $g(x) \nmid a(x)$ ).

L'IDÉE CLÉ : la distance  $d(\Gamma(L, g)) \geq t + 1$ ,  $t := \deg(g)$  puisque  $\sum_{i=0}^{n-1} \frac{w_i}{x - \gamma_i} = \frac{a(x)}{b(x)} \neq 0$  avec  $g(x) \nmid a(x) \Rightarrow$  il y a au moins  $s \geq t + 1$  termes  $w_i$  non-nuls car après avoir amener au dénominateur commun on aura le degré du polynôme  $a(x) \leq s - 1$ .

Soient  $v = (v_0, v_1, \dots, v_{n-1}) = w + e$ , un mot reçu,  $w = (w_0, w_1, \dots, w_{n-1}) \in \Gamma(L, g)$  un mot de code,  $e = (e_0, e_1, \dots, e_{n-1}) \in \mathbb{F}_q^n$  le mot d'erreur, alors on considère les fractions

suivantes

$$v(x) = \sum_{i=0}^{n-1} \frac{v_i}{x - \gamma_i}, \quad w(x) = \sum_{i=0}^{n-1} \frac{w_i}{x - \gamma_i}, \quad e(x) = \sum_{i=0}^{n-1} \frac{e_i}{x - \gamma_i}.$$

On écrira une matrice de contrôle  $G \in \text{Mat}_{mt,n}(\mathbb{F}_q)$  du code  $\Gamma(L, g)$  : on utilise pour tout  $i = 0, \dots, n-1$  un unique polynôme  $f_i(x) \bmod g(x)$  dans  $\mathbb{F}_{q^m}[x]/(g)$  tel que

$$f_i(x)(x - \gamma_i) \equiv 1 \bmod g(x) \Rightarrow f_i(x) := -\frac{1}{g(\gamma_i)} \left[ \frac{g(x) - g(\gamma_i)}{x - \gamma_i} \right]. \quad (7.2)$$

(c'est à dire,  $f_i(x)$  est l'inverse de  $(x - \gamma_i) \bmod g(x)$ ).

Soit

$$g(x) = \sum_{i=0}^t g_i x^i, \quad h_j = \frac{1}{g(\gamma_j)} \in \mathbb{F}_{q^m}^*$$

alors

$$\frac{g(x) - g(y)}{x - y} = \sum_{k+j \leq t-1} g_{k+j+1} y^j x^k \Rightarrow \sum_{i=0}^{n-1} w_i h_i \sum_{k+j \leq t-1} g_{k+j+1} \cdot (\gamma_i)^j x^k \equiv 0 \pmod{g}.$$

En effet

$$\frac{x^i - y^i}{x - y} = \sum_{k+j=i-1} y^j x^k;$$

ceci implique

$$f_i(x) = -\frac{1}{g(\gamma_i)} \left[ \frac{g(x) - g(\gamma_i)}{x - \gamma_i} \right] = -h_i \sum_{k+j \leq t-1} g_{k+j+1} \cdot (\gamma_i)^j x^k. \quad (7.3)$$

Pour pouvoir travailler avec les polynômes, on fait correspondre (de façon unique) au mot  $v$  un polynôme  $f_v(x)$  modulo  $g$ ,

$$v \mapsto f_v(x) = \sum_{i=0}^{n-1} v_i f_i(x) \Rightarrow w \mapsto f_w(x) = \sum_{i=0}^{n-1} w_i f_i(x) \equiv 0 \pmod{g}.$$

Si  $v = (v_0, v_1, \dots, v_{n-1}) = w + e$ , un mot reçu,  $w = (w_0, w_1, \dots, w_{n-1}) \in \Gamma(L, g)$ ,  $e = (e_0, e_1, \dots, e_{n-1}) \in \mathbb{F}_q^n$ , alors

$$f_v(x) = S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \equiv \sum_{i=0}^{n-1} v_i f_i(x) \pmod{g}$$

puisque  $\sum_{i=0}^{n-1} w_i f_i(x) \equiv 0 \pmod{g}$ . On obtient maintenant une matrice de contrôle de la forme explicite (7.3) :

$$\sum_{i=0}^{n-1} w_i f_i(x) \equiv 0 \pmod{g} \Rightarrow \sum_{i=0}^{n-1} w_i h_i \sum_{k+j \leq t-1} g_{k+j+1} \cdot (\gamma_i)^j x^k \equiv 0 \pmod{g}$$

ceci dit, pour tous les  $k = 0, \dots, t-1$  le coefficient de  $x^k$  à gauche est nul, donc une matrice de contrôle du code  $\Gamma(L, g)$  est

$$\begin{pmatrix} h_0 g_t & h_1 g_t & \dots & h_{n-1} g_t \\ h_0(g_{t-1} + g_t \gamma_0) & h_1(g_{t-1} + g_t \gamma_1) & \dots & h_{n-1}(g_{t-1} + g_t \gamma_{n-1}) \\ \dots & \dots & \dots & \dots \\ h_0(g_1 + g_2 \gamma_0 + \dots + g_t \gamma_0^{t-1}) & h_1(g_1 + g_2 \gamma_1 + \dots + g_t \gamma_1^{t-1}) & \dots & h_{n-1}(g_1 + g_2 \gamma_{n-1} + \dots + g_t \gamma_{n-1}^{t-1}) \end{pmatrix}$$

On obtient une matrice simplifiée en faisant des transformations élémentaires :

$$\begin{pmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_0\gamma_0 & h_1\gamma_1 & \cdots & h_{n-1}\gamma_{n-1} \\ \cdots & \cdots & \cdots & \cdots \\ h_0\gamma_0^{t-1} & h_1\gamma_1^{t-1} & \cdots & h_{n-1}\gamma_{n-1}^{t-1} \end{pmatrix}$$

Puis, on utilise comme syndrome la fraction rationnelle

$$e(x) = \sum_{i=0}^{n-1} \frac{e_i}{x - \gamma_i}$$

(le vecteur d'erreurs), représenté comme le *polynôme de syndrome* mod  $g$

$$f_e(x) \equiv S(x) \equiv f_v(x) \equiv \sum_{i=0}^{n-1} e_i f_i(x) \pmod{g}.$$

THÉORÈME 7.8 *Le code  $\Gamma(L, g)$  a :*

- (a) *la dimension  $k(\Gamma(L, g)) \geq n - mt$ ,*
- (b) *la distance  $d(\Gamma(L, g)) \geq t + 1$ ,  $t = \deg(g)$*

*Preuve* L'assertion (a) est immédiate de la taille de la matrice  $G$  vue comme une  $\text{Mat}_{mt, n}(\mathbb{F}_q)$ -matrice après un choix d'une base de  $\mathbb{F}_{q^m}$  sur  $\mathbb{F}_q$ .

L'assertion (b) est impliquée par l'unicité de décomposition d'une fraction rationnelle en éléments simples : pour avoir un élément non-nul

$$\sum_{i=0}^{n-1} \frac{c_i}{x - \gamma_i} = \frac{a(x)}{b(x)} \quad (7.4)$$

sous la forme d'une fraction irréductible  $\frac{a(x)}{b(x)}$  avec  $a(x)$  divisible par un polynôme irréductible  $g(x)$  de degré  $t$ , il faut avoir au moins  $s \geq t + 1$  termes non-nuls car après avoir amener au dénominateur commun on aura le degré du polynôme  $a(x) \leq s - 1$ .

### 7.3.1. Construction des bonnes familles

Rappelons

DÉFINITION 6.9. *Une bonne famille  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes est dite excellente s'il existe*

$$R = \lim_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \lim_{i \rightarrow \infty} \frac{d_i}{n_i}, \text{ tels que } R \geq 1 - H_q(\delta)$$

THÉORÈME 7.9 *Pour tout  $\delta \in ]0, (q-1)/q[$  il existe une famille excellente  $\{C_i\}$  des  $[n_i, k_i, d_i]_q$ -codes de Goppa rationnelles :*

$$R = \lim_{i \rightarrow \infty} \frac{k_i}{n_i}, \delta = \lim_{i \rightarrow \infty} \frac{d_i}{n_i},$$

et

$$R = 1 - H_q(\delta)$$

où  $H_q(\delta)$  est la fonction entropie  $q$ -aire définie sur  $[0, (q-1)/q]$  par

$$H_q(0) = 0,$$

$$H_q(\delta) = \delta \log_q(q-1) - \delta \log_q(\delta) - (1-\delta) \log_q(1-\delta) \text{ pour } 0 \leq \delta \leq (q-1)/q.$$



PREUVE du Théorème 7.9. On choisit  $n_i = q^{m_i}$ ,  $m = m_i$ ,  $d = d_i = [\delta n_i]$ , et on précisera le polynôme  $g(x) = g_i(x) \in \mathbb{F}_{q^m}[x]$  de degré  $t = t_i$  à partir de considération suivante : Soit  $c = (c_0, c_1, \dots, c_{n-1})$  un mot de code de poids  $j < d$ , alors le numérateur de la fraction (7.4) est de degré  $\leq j - 1$ , donc le nombre maximal de ces diviseurs irréductibles de degré  $t$  est inférieur ou égal à  $\left\lfloor \frac{j-1}{t} \right\rfloor$ . Il nous faut donc exclure de considération un petit ensemble de polynômes irréductibles, dont le nombre est majoré par

$$\sum_{j=1}^{d-1} \binom{n}{j} (q-1)^j \left\lfloor \frac{j-1}{t} \right\rfloor \leq \frac{d}{t} V_q(n, d)$$

puisque le nombre des numérateurs non-nuls de la fraction (7.4) est majoré par le nombre de toutes les combinaisons linéaires non-triviales (sur  $\mathbb{F}_q$ ) de  $d$  éléments parmi  $n$  éléments.

D'autre part on sait que (voir Corollaire 2.10) que

$$\lim_{n \rightarrow \infty} \frac{\log_q V_q(n, [\delta n])}{n} = H_q(\delta),$$

et que le nombre total des polynômes irréductibles de degré  $t$  sur  $\mathbb{F}_{q^m}[x]$  est donné par le théorème A.17 :

$$N_{q^m}(t) = \frac{1}{t} \sum_{d|t} \mu(d) q^{mt/d} \Rightarrow N_{q^m}(t) = \frac{1}{t} q^{mt} (1 + o(1)) \text{ lorsque } m \rightarrow \infty.$$

On obtient une condition suffisante pour l'existence d'un polynôme voulu  $g$  de degré  $t$

$$\frac{d}{t}V_q(n, d) < \frac{1}{t}q^{mt}(1 + o(1)) \iff d \cdot V_q(n, d) < q^{mt}(1 + o(1)) \quad (m \rightarrow \infty), \quad (7.5)$$

que signifie :

$$H_q(\delta) < \frac{mt}{n} + o(1), \text{ lorsque } m \rightarrow \infty.$$

Plus précisément, pour satisfaire les inégalités (7.5) et  $t < d = d_i$ , on prend  $\log_q$  de (7.5) :

$$\log_q(d \cdot V_q(n, d)) = \log_q([\delta n]) + \log_q(V_q(n, [\delta n])) \leq n(H_q(\delta) + \log_q(n)) \leq mt(1 + o(1)) \quad (m \rightarrow \infty),$$

Pour choisir  $g$  et  $t$  on pose

$$t = \left\lceil \frac{n}{m} H_q(\delta)(1 + o(1)) \right\rceil - 1 \quad (7.6)$$

pour satisfaire (7.5), et pour avoir  $t < d = [\delta n]$ , il suffit d'avoir

$$\frac{n}{m} H_q(\delta)(1 + o(1)) \leq \delta n - 1$$

que est vraie pour

$$m \geq \frac{H_q(\delta)(1 + o(1))}{\delta}.$$

Concernant le *rendement* de cette famille, on utilise le théorème 7.8 : avec  $n = n_i$ ,  $m = m_i$ ,  $k = k_i$ ,  $d = d_i$  ci-dessus, et  $t = t_i$  choisis comme dans l'égalité (7.6), on a

$$k_i \geq n_i - m_i t_i \text{ et } \frac{d_i}{n_i} \rightarrow \delta > 0.$$

Ceci résulte l'inégalité suivante :

$$R_i = \frac{k_i}{n_i} \geq \frac{n_i - m_i t_i}{n_i} \geq 1 - H_q(\delta)(1 + o(1)) \rightarrow 1 - H_q(\delta) \Rightarrow R \geq 1 - H_q(\delta)$$

donc la famille construite est **excellente**.

## 7.4 Décodage des codes de Goppa rationnels

On utilise comme syndrome la fraction rationnelle

$$\sum_{i=0}^{n-1} \frac{e_i}{x - \gamma_i}$$

(le vecteur d'erreurs), représenté comme un polynôme mod  $g$

$$S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \bmod g$$

(le polynôme de syndrome). Remarquer que si  $v = (v_0, v_1, \dots, v_{n-1}) = w + e$ , un mot reçu,  $w = (w_0, w_1, \dots, w_{n-1}) \in \Gamma(L, g)$ ,  $e = (e_0, e_1, \dots, e_{n-1}) \in \mathbb{F}_q^n$ , alors

$$S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \equiv \sum_{i=0}^{n-1} v_i f_i(x) \bmod g$$

puisque

$$\sum_{i=0}^{n-1} w_i f_i(x) \equiv 0.$$

Ensuite, on utilise une version de l'algorithme de Berlekamp-Massey, voir la section 5.5.

Soit  $I := \{i \mid e_i \neq 0\}$ , et on considère le polynôme locateur d'erreurs

$$s(x) = \prod_{i \in I} (x - \gamma_i),$$

et le polynôme évaluateur d'erreurs

$$u(x) = \sum_{i \in I} e_i \prod_{j \in I \setminus \{i\}} (x - \gamma_j).$$

On cherche  $u(x), s(x)$  à partir de sa définition comme une solution de la congruence :

$$u(x) \equiv s(x)S(x) \pmod{g(x)} \quad (7.7)$$

Pour résoudre la congruence (7.7) on utilise la division euclidienne et l'identité de Bezout : on calcule trois suites  $s_n(x), t_n(x), u_n(x)$  avec la propriété

$$t_n(x)g(x) + s_n(x)S(x) = u_n(x)$$

où le degré de  $u_n(x)$  décroît jusqu'à ce que  $\deg(u_{j+1}) < t$  et  $\deg(u_j) \geq t$  ( $t = \lceil (\deg g - 1)/2 \rceil$ ) à partir de

$$0 \cdot g(x) + 1 \cdot S(x) = S(x), 1 \cdot g(x) + 0 \cdot S(x) = g(x)$$

de telle façon que

$$(s_0(x), t_0(x), u_0(x)) = (0, 1, S(x)), (s_1(x), t_1(x), u_1(x)) = (1, 0, g(x)).$$

Il est clair que le couple  $(u(x), s(x)) = (u_n(x), s_n(x))$  est un unique couple satisfaisant (7.7) avec la propriété que le degré de  $s(x)$  est inférieur de degré de  $g(x)$ .

EXERCICE. Ecrire un algorithme pour trouver  $s(x)$  et  $u(x)$ .

# Constructions des codes géométriques à partir des courbes projectives

## 7.5 Espace projectif $\mathbb{P}^n$ , variétés algébriques

Soit  $K$  un corps et  $n \geq 1$  un entier. On considère l'espace projectif de dimension  $n$  sur  $K$  et on note  $\mathbb{P}_K^n$  ou simplement  $\mathbb{P}^n$  l'ensemble des classes d'équivalence de  $(n+1)$ -uplets

$$(x_0, \dots, x_n) \in K^{n+1} \setminus (0, \dots, 0),$$

sous la relation  $(x_0, \dots, x_n) \sim (\lambda x_0, \dots, \lambda x_n)$  pour tout  $\lambda \in K^*$ . Une classe d'équivalence  $x$ , notée  $(x_0 : \dots : x_n)$  est dit un point de  $\mathbb{P}^n$ .

Soit  $K[T] = K[T_0, \dots, T_n]$ . On interprète les éléments de  $K[T]$  comme des fonctions régulières de l'espace affine  $\mathbb{A}^{n+1}$ , et on interprète les éléments de

$$K \left( \frac{T_1}{T_0}, \frac{T_2}{T_0}, \dots, \frac{T_n}{T_0} \right),$$

comme des fonctions rationnelles de  $\mathbb{P}^n$ .

REMARQUE : Soit  $(x_0, \dots, x_n) \in K^{n+1}$ , et soit  $F = F_0 + F_1 + \dots + F_m \in K[T_0, \dots, T_n]$  un polynôme avec des composantes homogènes  $F_i$  de degré  $i$ , alors

$$F(\lambda x_0, \dots, \lambda x_n) = F_0(x_0, \dots, x_n) + \lambda F_1(x_0, \dots, x_n) + \dots + \lambda^m F_m(x_0, \dots, x_n).$$

Il vient que pour un polynôme homogène  $F$ , la condition  $F(x) = 0$  ne dépend que de la classe  $x = (x_0 : \dots : x_n) \in \mathbb{P}^n$ .

DÉFINITION 7.10 *Une partie  $X \subset \mathbb{P}^n$  est dite une variété algébrique projective irréductible s'il existe un idéal premier homogène  $P$  de  $K[T_0, \dots, T_n]$ , tel que*

$$X = V(P) = \{x \in \mathbb{P}^n \mid \forall F \in P, F(x) = 0\},$$

*c'est à dire,  $P$  est un idéal premier engendré par des polynômes homogènes, et  $X$  est l'ensemble des zéros d'un système homogène donné par des générateurs de  $P$ .*



Il y a une grande différence entre les fonctions rationnelles des coordonnées affines et des coordonnées projectives : une fraction

$$f(T_0, \dots, T_n) = \frac{R(T_0, \dots, T_n)}{S(T_0, \dots, T_n)}$$

ne définit pas une fonction d'un point  $x = (x_0 : \dots : x_n) \in \mathbb{P}^n$ , si  $S(x_0, \dots, x_n) \neq 0$ , car sa valeur dépend du choix des coordonnées homogènes de  $x$ .

Cependant, pour les polynômes homogènes  $R$  et  $S$  de même degré, tels que  $S(x_0, \dots, x_n) \neq 0$ , la valeur

$$f(x_0, \dots, x_n) = \frac{R(x_0, \dots, x_n)}{S(x_0, \dots, x_n)}$$

est bien définie.

DÉFINITION 7.11 *Soit  $X$  une variété projective. L'idéal de  $X$  est l'ensemble*

$$I(X) = \{F \in F[T_0, \dots, T_n] \mid F(x) = 0, \forall x \in X\}.$$

Pour définir le corps des fonctions  $K(X)$  d'une variété projective irréductible  $X$  on considère d'abord l'anneau

$$\mathcal{O}_X = \left\{ f(T_0, \dots, T_n) = \frac{R(T_0, \dots, T_n)}{S(T_0, \dots, T_n)} \right. \\ \left. \mid R, S \in F[T_0, \dots, T_n] \text{ polynômes homogènes de même degré, tels que } S \notin I(X) \right\}.$$

Alors l'idéal

$$M_X = \left\{ f(T_0, \dots, T_n) = \frac{R(T_0, \dots, T_n)}{S(T_0, \dots, T_n)} \mid R, S \in F[T_0, \dots, T_n], R \in I(X), S \notin I(X) \right\}$$

est un seul idéal maximal de l'anneau  $\mathcal{O}_X$ , et l'anneau quotient  $\mathcal{O}_X/M_X$  est un corps, dit le corps des fractions sur  $X$ .

DÉFINITION 7.12 (a) Soit  $U$  une partie d'une variété projective  $X$ . On dit que  $U$  est un voisinage ouvert contenant un point  $x \in X$ , s'il existe un polynôme homogène  $S(T_0, \dots, T_n) \in K[T_0, \dots, T_n]$  tel que  $U \ni x$ , et

$$U = X \cap \{t = (t_0 : \dots : t_n) \in \mathbb{P}^n \mid S(t_0, \dots, t_n) \neq 0\}.$$

(b) Une application  $f : U \rightarrow K$  est dite une fonction régulière au point  $x$  s'ils existe  $R, S \in K[T_0, \dots, T_n]$  des polynômes homogènes de même degré tels que  $S(x) \neq 0$  et  $f = R/S$  dans un voisinage de  $x$ ;  $f$  est régulière sur une partie  $Y$  de  $X$ , si elle est régulière en tout  $x$  de  $Y$ .

EXEMPLE. On considère la variété projective  $X \subset \mathbb{P}^2$  sur  $K = \mathbb{C}$ , donné par l'équation homogène irréductible  $u^3 = v^3 - w^3$ . Alors la fonction rationnelle

$$\frac{u}{v - w} = \frac{u(v^2 + vw + w^2)}{(v - w)(v^2 + vw + w^2)} = \frac{v^2 + vw + w^2}{u^2}$$

est régulière si  $v \neq w$ , même si  $u = 0$ , par exemple en  $(0 : 1 : j)$  et en  $(0 : 1 : j^2)$ .

Soit  $x$  un point de  $X$  et soient les couples  $(U, f)$  dans lesquels  $f$  est régulière sur  $U$  voisinage ouvert de  $x$ . On définit la relation d'équivalence suivante :

$$(U, f) \sim (U', f') \iff f = f' \text{ sur } U \cap U'$$

dont les classes d'équivalences forment un anneau.

**DÉFINITION 7.13** *L'anneau des classes d'équivalences de la relation ci-dessus est appelé l'anneau local de  $x$ , noté  $\mathcal{O}_x$ . Cet anneau est local au sens algébrique d'unique idéal maximal*

$$\mathfrak{m}_x = \{ \text{classes d'équivalence de } (U, f) \text{ avec } f(x) = 0 \}$$

DÉFINITION 7.14 *Un anneau noethérien local  $\mathcal{O}$  d'idéal maximal  $\mathfrak{m}$  et de corps résiduel  $k(\mathfrak{m}) = \mathcal{O}/\mathfrak{m}$  est dit régulier si le  $k(\mathfrak{m})$ -espace vectoriel  $\mathfrak{m}/\mathfrak{m}^2$  a la même dimension que la dimension de l'anneau  $\mathcal{O}$ .*

DÉFINITION 7.15 *Une variété projective  $X$  est dite non-singulière au point  $x$  si  $\mathcal{O}_x$  est un anneau local régulier, et  $X$  est dite singulière au point  $x$  sinon. Une variété  $X$  s'appelle non-singulière ou lisse si elle est non singulière en tout point dans toute extension de  $K$ . L'ensemble des points non-singuliers de  $X$  est un ouvert non vide de  $X$  dense dans  $X$ .*

DÉFINITION 7.16 : *On appelle courbe projective lisse toute variété algébrique projective lisse de dimension 1. La dimension de  $X$  est le degré de transcendance de  $K(X)$  sur  $K$ .*

## Exemple : courbes planes

Rappelons qu'une courbe projective plane  $\mathcal{C}$  sur un corps  $K$  est défini par une équation de type  $F(X : Y : Z) = 0$ , où  $F(X : Y : Z) \in K[X, Y, Z]$  est une forme homogène des variables projectives  $X, Y, Z$ .

L'équation de la tangente dans les coordonnées affines a la forme

$$f'_x(P)(x - \alpha) + f'_y(P)(y - \beta) = 0.$$

Par la construction,

$$f(x, y) = F(x, y, 1), \text{ où } F(X, Y, Z) = 0 \text{ l'équation homogène de la courbe.}$$

Ceci implique :  $f'_x = F'_X, f'_y = F'_Y$ , et selon le théorème connu de Euler (sur les fonctions homogènes) on a

$$XF'_X + YF'_Y + ZF'_Z = nF \text{ où } n \text{ est le degré de } F$$

Lorsque  $P = (\alpha : \beta : 1)$  se trouve sur la courbe alors

$$\alpha F'_X(P) + \beta F'_Y(P) + F'_Z(P) = nF,$$

donc l'équation de la tangente se transforme vers

$$xF'_X(P) + yF'_Y(P) + F'_Z(P) = 0 \iff XF'_X + YF'_Y + ZF'_Z = 0.$$

(c'est la forme projective de la droite tangente).

DÉFINITION 7.17

(a) Un **point singulier** sur une courbe projective plane  $\mathcal{C}$  sur  $K$  est toute solution du système

$$F = F'_X = F'_Y = F'_Z = 0$$

dans une extension de  $K$ .

(b) On dit qu'une courbe projective plane  $\mathcal{C}$  sur  $K$  est **lisse** si le système

$$F = F'_X = F'_Y = F'_Z = 0$$

n'a pas de solutions non-triviales dans toute extension de  $K$ .

## 7.6 Codes provenant des courbes algébriques

Tous les exemples considérés dans le reste de la section proviennent d'une courbe algébrique  $X$  sur  $\mathbb{F}_q$  muni d'un plongement projectif

$$\varphi : X \hookrightarrow \mathbb{P}_{\mathbb{F}_q}^N.$$

On utilise des sous-espaces vectoriels  $\mathcal{L}(D) \subset \mathbb{F}_q(X)$  de dimension finie (voir la définition 7.21) associés à un *diviseur*  $D$  (voir la section 7.7). **Un diviseur est une combinaison linéaire finie formelle sur  $\mathbb{Z}$**

$$D = \sum_i a_i P_i$$

des points  $P_i$  sur  $X$ . Un exemple typique de l'espace  $\mathcal{L}(D)$  est donné par les polynômes de degré  $\leq t$ , qui correspondent aux **fonctions rationnelles** sur la droite projective  $X = \mathbb{P}^1$  avec un pôle de degré  $\leq t$  en  $\infty$ ; dans ce cas  $D = t \cdot (\infty)$ .

Il existe plusieurs bonnes constructions des codes ( $L$ -construction,  $\Omega$ -construction,  $\mathcal{P}$ -construction, voir [Ste], pp.243-248). Dans ce qui suit on donnera seulement quelques exemples.



### 7.6.1. $\mathcal{P}$ -construction

Dans ce cas  $\mathcal{P} = X$ ,  $n = \text{Card}(X(\mathbb{F}_q))$ , On considère un diviseur  $D$  sur une courbe projective lisse  $X$ , c'est à dire, une combinaison linéaire

$$D = \sum a_P P,$$

la somme étant sur tous les points de  $X$ , et les  $a_P$  entiers tous nuls sauf un ensemble fini. Ensuite, on considère l'espace vectoriel  $\mathcal{L}(D)$  attaché au diviseur  $D$  :

$$\mathcal{L}(D) = \{f \equiv 0, \text{ ou } f \in K(X)^* | \text{div}(f) + D \geq 0\},$$

C'est à dire, que si  $D = -\sum a_P P + \sum a_{P'} P'$ , avec les  $a_P$  et les  $a_{P'}$  tous positifs ou nuls,  $\mathcal{L}(D)$  est l'ensemble des éléments non nuls de  $K(X)$  ayant des zéros d'ordre au moins  $a_P$  aux points  $P$  et des pôles d'ordre au plus  $a_{P'}$  aux points  $P'$ .

Puis on pose

$$k = l(D) = \dim_{\mathbb{F}_q} \mathcal{L}(D),$$

où  $\mathcal{L}(D) \subset \mathbb{F}_q(X)$  est un sous-espace de dimension finie dans le corps

$$\mathbb{F}_q(X) = \left\{ \frac{u(T_0, \dots, T_N)}{v(T_0, \dots, T_N)} \right\} / \text{mod } \sim$$

des fonctions rationnelles sur  $X$ .

On utilise un plongement canonique

$$\varphi_D : X \hookrightarrow \mathbb{P}_{\mathbb{F}_q}^{l(D)-1}$$

attaché à un diviseur de degré  $\deg(D) > 2g$ , où  $g = g(X)$  est le *genre* de  $X$ , voir [Ste], p.93 et la définition 7.29.

REMARQUE : On peut définir le **genre** comme le maximum  $\boxed{g(X) = \max_D (\deg D - l(D) + 1)}$ ,

pris sur tous les diviseurs  $D$  sur  $X$ . Cette définition simple à été utilisé par Riemann. Il existe des définitions équivalentes, par exemple,  $g(X) = l(K_X)$ , où  $K_X$  est le diviseur d'une différentielle non nulle. Pour les courbes complexes  $X$ ,  $g(X)$  coïncide avec le **nombre d'anses** de  $X$ , vue comme une **surface de Riemann compacte**.

On pose  $a = \deg D$ , et on suppose  $D$  positif :  $\forall P, a_P \geq 0$ . Alors

$$d = n - \max_H \{ \text{Card}(X \cap H) \mid H \text{ hyperplan} \}$$

puisque  $\deg D \leq \max_H \{ \text{Card}(X \cap H) \mid H \text{ hyperplan} \}$  : selon la construction du plongement  $\varphi_D$  ([Ste], p.92) avec  $N = l(D) - 1$ , il existe un hyperplan  $H \subset \mathbb{P}_{\mathbb{F}_q}^{l(D)-1}$  tel que  $D = \varphi_D^{-1}(H)$ , donc  $\deg D \leq \max_H \{ \text{Card}(X \cap H) \mid H \text{ hyperplan} \}$  (il se peut que les coordonnées des points d'intersection appartiennent à une extension du corps de base  $\mathbb{F}_q$ ).

### 7.6.2. Borne de la géométrie algébrique

On utilise le théorème 7.30 de Riemann-Roch (sous une forme faible) :

$$l(D) \geq \deg(D) - g + 1,$$

où  $g = g(X)$  est le *genre* de  $X$ , voir [Ste], p.93 et la définition 7.29.

Ceci implique :

$$\left\{ \begin{array}{l} k = l(D) \geq \deg(D) - g + 1, \text{ où } a = \deg(D) \\ d \geq n - a, (d := n - \max_H \{ \text{Card}(X \cap H) \mid H \text{ hyperplan} \}) \end{array} \right\} \quad (7.8)$$

donc l'addition directe dans 7.8 donne

$$\boxed{k + d \geq n - g + 1} \quad (7.9)$$

la *borne de l'existence de la géométrie algébrique*, comparer avec la borne de Singleton :  $k + d \leq n + 1$ .

Ensuite on divise (7.9) par  $n$  :

$$R + \delta \geq 1 - \frac{g}{n} + \frac{1}{n} \quad (7.10)$$

THÉORÈME 7.18 (IHARA-TSFASMAN-ZINK-DRINFELD-VLADUT, sans démonstration). *Soit  $g = g(X)$  est le genre de  $X$ ,  $n = \text{Card } X(\mathbb{F}_q)$ ). Alors*

(a)

$$\liminf_X \frac{g}{n} \geq \frac{1}{\sqrt{q} - 1}. \quad (7.11)$$

(b) *Si  $q$  est un carré, la borne (7.11) est atteinte*

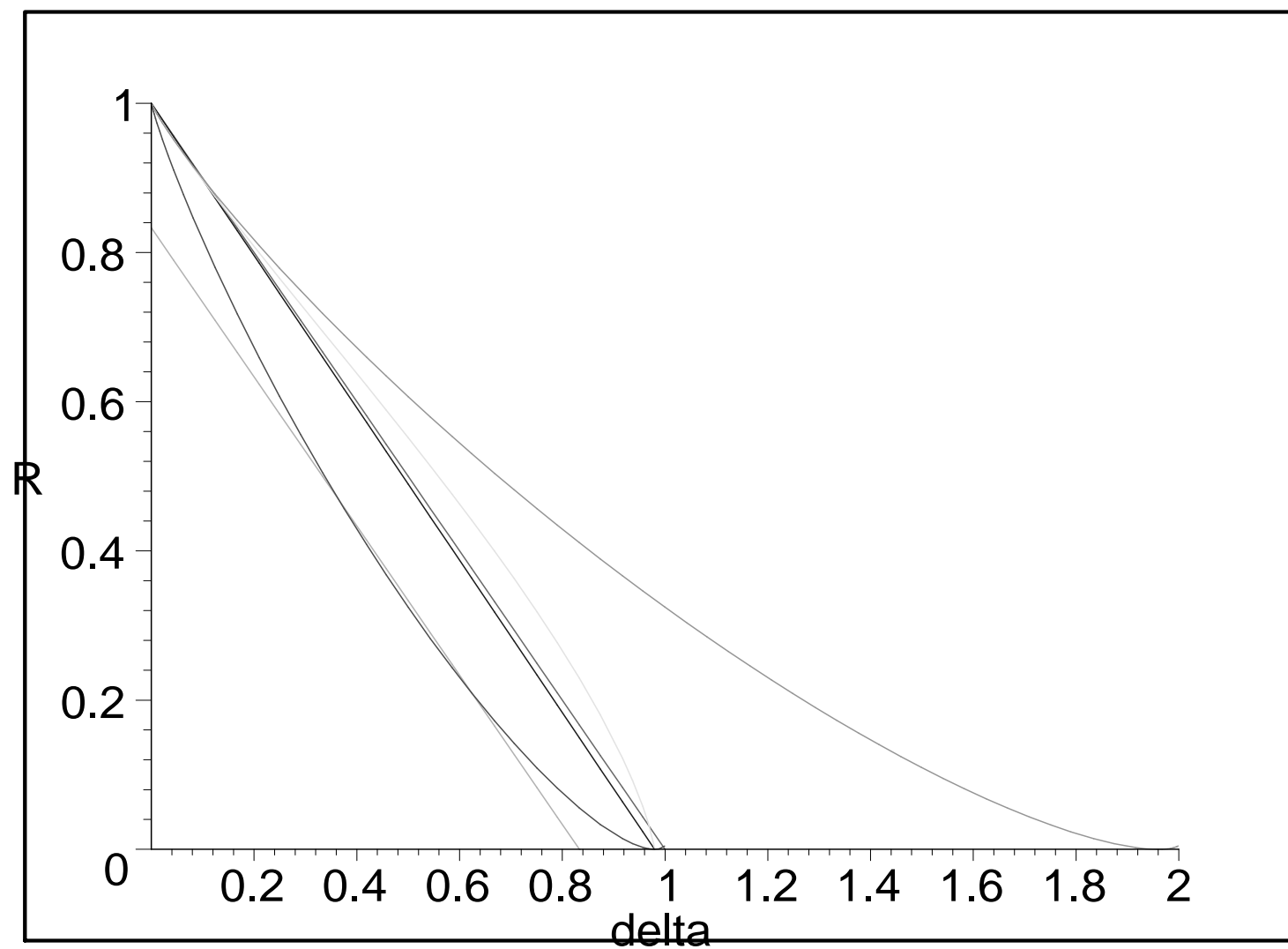
$$\liminf_X \frac{g}{n} = \frac{1}{\sqrt{q} - 1}.$$

COROLLAIRE 7.19 *Si  $q$  est un carré, alors*

$$\limsup R \geq 1 - \frac{1}{\sqrt{q} - 1} - \delta.$$

En effet, il suffit de substituer  $\liminf_X \frac{g}{n} = \frac{1}{\sqrt{q} - 1}$  dans l'inégalité  $R + \delta \geq 1 - \frac{g}{n} + \frac{1}{n}$ , et passer à la limite.

Pour  $q = 49$  :



## 7.7 Généralités sur les diviseurs

DÉFINITION 7.20 *Un diviseur sur une courbe projective lisse  $X$  est une combinaison linéaire*

$$D = \sum a_P P,$$

*la somme étant sur les points de  $X$ , et les  $a_P$  entiers tous nuls sauf un ensemble fini appelé le support de  $D$ . Si les  $a_P$  sont tous positifs ou nuls, on note  $D \geq 0$ . On définit le degré de  $D$  :  $\deg(D) = \sum a_P$ .*

Soit  $x$  point non singulier de  $X$ , et  $f \in \mathcal{O}_x$ . On pose  $v_x(f) = n$  si  $f \in m_x^n$  et  $f \notin m_x^{n+1}$ . Si  $h = f/g \in K(X)$ , on pose  $v_x(h) = v_x(f) - v_x(g)$ . Si  $h \in \mathfrak{m}_x$ , alors  $x$  est un zéro de  $h$  d'ordre  $v_x(h)$ . Si  $h^{-1} \in \mathfrak{m}_x$ , alors  $x$  est un pôle de  $h$  d'ordre  $v_x(h^{-1})$ . Ensuite, le diviseur  $(f) = \sum v_x(f) \cdot x$  s'appelle le diviseur de la fonction  $f$ . Les diviseurs de la forme  $(f)$  sont appelés principaux. On définit alors la relation d'équivalence

$$D \sim D' \iff \text{il existe } f \text{ de } K(X)^* \text{ telle que } D - D' = (f)$$

La classe d'équivalence de  $D$  sous cette relation est appelée classe de diviseurs de  $D$ . On dit que  $D$  et  $D'$  sont équivalents.

DÉFINITION 7.21 *Le  $K$ -espace vectoriel  $\mathcal{L}(D)$  attaché à un diviseur  $D$  est défini par :*

$$\mathcal{L}(D) = \{f \equiv 0, \text{ ou } f \in K(X)^* | \operatorname{div}(f) + D \geq 0\}.$$

*C'est à dire, que si  $D = -\sum a_P P + \sum a_{P'} P'$ , avec les  $a_P$  et les  $a_{P'}$  tous positifs ou nuls,  $\mathcal{L}(D)$  est l'ensemble des éléments non nuls de  $K(X)$  ayant des zéros d'ordre au moins  $a_P$  aux points  $P$  et des pôles d'ordre au plus  $a_{P'}$  aux points  $P'$ . On note  $l(D)$  la dimension sur  $K$  de  $\mathcal{L}(D)$ .*

THÉORÈME 7.22 *On a*

$$\begin{cases} l(D) = 0, & \text{si } \deg D < 0 \\ l(D) \leq \deg D + 1, & \text{sinon.} \end{cases}$$

PREUVE : voir [Ste], p.83.

EXEMPLE. Un exemple typique de l'espace  $\mathcal{L}(D)$  est donné par les polynomes de degré  $\leq t$ , qui correspondent aux fonctions rationnelles sur la droite projective  $X = \mathbb{P}^1$  avec un pôle de degré  $\leq t$  en  $\infty$ ; dans ce cas  $D = t \cdot (\infty)$ . Dans ce cas l'égalité est atteinte  $l(D) = \deg D + 1$ .

### 7.7.1. Différentielles et calcul de $l(A)$

Soit  $R$  un anneau sur un autre anneau  $\mathcal{O}$  (avec le morphisme de structure  $i : \mathcal{O} \rightarrow R$ ).

**DÉFINITION 7.23** *Le module de différentielles  $\Omega_{R/\mathcal{O}}$  est défini comme un  $R$ -module muni d'une application de  $R$ -modules  $d : R \rightarrow \Omega(R/\mathcal{O})$  satisfaisante la condition  $d(rs) = rd(s) + sd(r)$ , et qui est universel par rapport à cette condition. Une construction explicite : posons  $I = \text{Ker } \delta$ ,  $\delta : R \otimes_{\mathcal{O}} R \rightarrow R$  morphisme de la multiplication, alors  $\Omega(R/\mathcal{O}) = I/I^2$ , et  $d(r) = r \otimes 1 - 1 \otimes r$ .*

**PROPOSITION 7.24** *Soit  $R/\mathcal{O}$  une extension des corps  $R = K$  et  $k = \mathcal{O}$ . Alors  $\Omega(K/k)$  est un espace vectoriel sur  $K$  et  $df_1, \dots, df_n$  est une base de  $\Omega(K/k)$  sur  $K \iff K/k(f_1, \dots, f_n)$  est une extension séparable algébrique des corps.*

*Preuve* : voir [La]. En particulier, pour une courbe  $X$  sur  $k$  on pose  $\Omega_X = \Omega(k(X)/k)$  : alors  $\dim_K \Omega_X = 1$  car le degré de transcendance de  $K/k$  est égale à 1.

Pour un  $P \in X$  posons  $\Omega_P = \Omega(\mathcal{O}_P/k)$ , alors  $\Omega_P = \mathcal{O}_P dt$ , où  $t$  une uniformisante locale en  $P$ , c'est à dire,  $\mathfrak{m}_{\mathcal{O}} = (t)$ . De plus,  $dt$  est en même temps une base de  $\Omega_X$  sur  $K$ .

Soit  $\omega$  une forme différentielle régulière sur  $U$  ouvert non vide de  $X$ . Les classes de la relation d'équivalence

$$(U, \omega) \sim (U', \omega') \iff \omega = \omega' \text{ sur } U \cap U'$$

sont appelées formes différentielles rationnelles dont l'ensemble  $\Omega(X)$  est un  $K(X)$ -module.



DÉFINITION 7.25 Soit  $\omega$  une forme différentielle rationnelle sur  $X$  courbe lisse. Au voisinage de  $x$  point de  $X$ ,  $\omega$  s'écrit  $\omega = f dt$ , avec  $f = f_x$  une fonction rationnelle et  $t = t_x$  un paramètre local. On définit le diviseur de  $\omega$  par

$$(\omega) = \sum v_x(f_x) \cdot x$$

La classe de diviseurs d'une forme différentielle rationnelle sur  $X$  est appelée classe des diviseurs canoniques, notée  $W$ . Un représentant de  $W$  est appelé diviseur canonique noté  $K$ .

REMARQUE 7.26 : On a  $\mathcal{L}(K) \cong \Omega[X]$  (l'espace des formes différentielles régulières sur  $X$ ) par l'isomorphisme de  $K$ -espaces vectoriels  $f \rightarrow f\omega$

## 7.8 Courbes sur les corps finis

DÉFINITION 7.27 : Soit  $K'$  un sous-corps de  $K$ . Un point  $x = (x_0, \dots, x_n)$  de  $\mathbb{P}_K^n$  est dit  $K'$ -rationnel si

$$x_i \neq 0 \implies x_j/x_i \in K' \text{ pour tout } 0 \leq j \leq n$$

DÉFINITION 7.28 Soit  $X$  une courbe projective lisse sur  $K$  et  $\sigma$  l'automorphisme de Frobenius sur  $X$ . Un diviseur  $D$  est dit  $K$ -rationnel si

$$\sigma(D) = D$$

## 7.9 Théorème de Riemann-Roch sur un corps $K$ , genre

DÉFINITION 7.29 Soit  $X$  une courbe projective lisse définie sur  $K$ , et  $K_X$  la classe canonique de  $X$  (c'est à dire, la classe d'une différentielle non nulle). On appelle genre de  $X$

$$g = g(X) = l(K_X) = \dim_K \mathcal{L}(K_X)$$

REMARQUE : On peut définir aussi le genre comme le maximum  $\boxed{g(X) = \max_D (\deg D - l(D) + 1)}$ ,

pris sur tous les diviseurs  $D$  sur  $X$ . Cette définition simple à été utilisé par Riemann. Il existe des définitions équivalentes, par exemple,  $g(X) = l(K_X)$ , où  $K_X$  est le diviseur d'une différentielle non nulle. Pour les courbes complexes  $X$ ,  $g(X)$  coïncide avec le nombre d'anses de  $X$ , vue comme une surface de Riemann compacte.

THÉORÈME 7.30 (de Riemann-Roch). *Soit  $D$  un diviseur sur une courbe projective et lisse  $X$  de genre  $g$  et soit  $K_X$  la classe canonique de  $X$ . Alors*

$$l(D) - l(K_X - D) = \deg D - g + 1$$

PREUVE : Voir [Ste], p.91.

PROPOSITION 7.31 (Formule du genre de Plücker). *Soit  $X$  une courbe plane projective et lisse de degré  $m$  dans  $\mathbb{P}^2$ . Alors*

$$g = (m-1)(m-2)/2$$

PREUVE : voir [Ste], p.96.

Les codes de Goppa se basent sur une courbe algébrique lisse et un ensemble de points rationnels sur cette courbe. Pour construire des codes efficaces, il est utile de choisir des courbes avec un grand nombre de points rationnels.

THÉORÈME 7.32 (Borne de Hasse-Weil). *Soit  $N_{q^m}(X)$  le nombre de points  $\mathbb{F}_{q^m}$ -rationnels sur la courbe projective lisse  $X$ . Alors*

$$|N_{q^m}(X) - q^m - 1| \leq 2gq^{m/2}$$

*Il existe bien d'autres bornes concernant le nombre de points rationnels sur des courbes particulières, et le lecteur est renvoyé à [Ste], chap.6 pour une connaissance plus approfondie sur celles-ci.*

## 7.10 Codes géométriques de Goppa : $L$ -construction.

DÉFINITION 7.33 : Soit  $X$  une courbe projective lisse définie sur  $\mathbb{F}_q$ . Soient  $x_1, \dots, x_n$  des points de  $X$   $\mathbb{F}_q$ -rationnels et soit  $D_0 = \sum x_i$ . Soit enfin  $D$  un diviseur sur  $X$   $\mathbb{F}_q$ -rationnel tel que les supports de  $D_0$  et  $D$  soient disjoints. Le code linéaire  $C = C(D_0, D)$   $q$ -aire de longueur  $n$  associé au couple  $(D_0, D)$  est l'image de

$$\begin{aligned} Ev : \mathcal{L}(D) &\rightarrow \mathbb{F}_{q^n} \\ f &\mapsto (f(x_1), \dots, f(x_n)) \end{aligned}$$

*Un tel code est appelé un code géométrique de Goppa.*

Pour construire un code de Goppa (non rationnel) sur  $\mathbb{F}_q$ , nous devons choisir une courbe lisse  $X$ , et deux diviseurs particuliers. En effectuant pour ces éléments des choix judicieux, nous allons pouvoir construire des codes très efficaces. Afin de mieux guider notre choix, nous allons établir quelques relations entre les divers paramètres du code et certaines grandeurs liées aux objets choisis.

THÉORÈME 7.34 Soit  $X$  une courbe projective lisse de genre  $g$  et  $x_1, \dots, x_n$  des points de  $X$   $\mathbb{F}_q$ -rationnels. Alors le code de Goppa  $C(D_0, D)$  est de dimension  $k$ , de longueur  $n$  et d'écart  $d$  satisfaisant

$$k = l(D) - l(D - D_0) = \deg D - g + 1 + l(K - D) - l(D_0 - D)$$

$$d \geq n - \deg D$$

COROLLAIRE 7.35 : Si  $\deg D < n$ , alors

- (i)  $k \geq \deg D - g + 1$  et  $d \geq n - \deg D$
- (ii) Si de plus  $2g - 2 < \deg D < n$ , alors  $k = \deg D - g + 1$
- (iii) Si  $\{f_1, \dots, f_k\}$  est une base de  $\mathcal{L}(D)$ , alors

$$G = \begin{pmatrix} f_1(x_1) & f_1(x_2) & \cdots & f_1(x_n) \\ \vdots & \vdots & \vdots & \vdots \\ f_k(x_1) & f_k(x_2) & \cdots & f_k(x_n) \end{pmatrix}$$

est une matrice génératrice de  $C(D_0, D)$ .

Les paramètres  $k_C = \deg D - g + 1$  et  $d_C = n - \deg D$  sont appelés *dimension désignée* et *distance désignée* du code  $C(D_0, D)$ . D'après le théorème plus haut, on a la distance de  $C$   $d \geq d_C$ , et le corollaire ci-dessus nous indique que si  $n > \deg D > 2g - 2$ , alors  $k = k_C$ .

REMARQUE 7.36 Soit désormais  $C = C(D_0, D)$  un code sur une courbe lisse de genre zéro sur  $\mathbb{F}_q$ , tel que  $\deg D < n$ . Alors les paramètres de  $C$  satisfont

$$k + d \geq n + 1 - g$$

ou encore

$$R + \delta \geq 1 + (1 - g)/n$$

Or, d'après la borne de Singleton, on a  $k + d \leq n + 1$ . Donc si  $g = 0$ , on a

$$k + d = n + 1$$

et la borne est atteinte.

EXEMPLE : Soit  $X = \mathbb{P}^1(\mathbb{F}_q)$  la courbe lisse sur  $\mathbb{F}_q$  de genre zéro et les diviseurs  $D = m \cdot x$  et  $D_0$  tel que  $\text{Supp} D_0 \not\ni x$

Alors  $\mathcal{L}(D)$  est l'espace des polynômes sur  $\mathbb{F}_q$  de degré au plus  $m$ , et le code de Goppa généré est de longueur  $q$ , de dimension  $m + 1$  et d'écart  $q - m$  sur  $\mathbb{F}_q$  qui est un code de Reed-Solomon.

EXEMPLE : CODES DE HERMITE. On considère la courbe projective et lisse  $X \subset \mathbb{P}_{\mathbb{F}_q}^2$

$$X : u^{q+1} + v^{q+1} + w^{q+1} = 0$$

de genre  $g(X) = \frac{q(q-1)}{2}$ . Elle contient  $3(q+1)$  points sur  $\mathbb{F}_{q^2}$  avec  $uvw = 0$  :

$$(1, \zeta, 0), (0, 1, \zeta), (1, 0, \zeta) \in X(\mathbb{F}_{q^2}) \text{ avec } \zeta^{q+1} = 1, \zeta \in \mathbb{F}_{q^2}^*,$$

de plus, il y a  $(q-2)(q+1)^2$  point avec  $uvw \neq 0$  : dans ce cas on peut supposer  $u = 1$ ,  $v^{q+1} \neq -1$ , et le nombre de solution de l'équation  $w^{q+1} = a \in \mathbb{F}_q^*$  égal à  $q+1$ . Donc

$$\text{Card}(X(\mathbb{F}_{q^2})) = 3(q+1) + (q-2)(q+1)^2 = q^3 + 1 = q^2 + 1 + 2g(X)q.$$

C'est-à-dire, le nombre des points  $X(\mathbb{F}_{q^2})$  de la courbe  $X$  est maximal possible. On pose  $n = q^3$

$$D_0 = \{x_1, \dots, x_n\}, x_i \in X(\mathbb{F}_{q^2}) \setminus x_\infty, (x_\infty = (0, 1, 1)), D_0 = m \cdot x_\infty.$$

Soit  $C = C(D_0, D) = \mathcal{L}(D_0 - D)$  le code de Goppa correspondant, c'est un

$$[q^3, m - g + 1, d]_q - \text{code}, \text{ avec } d \geq n - m, \text{ avec le choix de } m : q^2 - q < m < q^3.$$

Si  $q = 2$ ,  $g(X) = 1$ ,  $X$  est une courbe elliptique,  $\text{Card}(X(\mathbb{F}_4)) = 9$  :



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

COURBE de FERMAT sur GF(4)

```
> restart ;
```

```
> with(linalg) :
```

Warning, the protected names norm and trace have been redefined and unprotected

```
> g:=x^2+x+1 mod 2;
```

$$g := x^2 + x + 1$$

```
> alias(alpha = RootOf(g)) ;
```

$$\alpha$$

```
> f:=(x,y,z)->x^3+y^3+z^3;
```

$$f := (x, y, z) \rightarrow x^3 + y^3 + z^3$$

```
> h:=(i,j)->x[i-1,j-1]: X:=Matrix(3,2,h);
```

$$X := \begin{bmatrix} x_{0,0} & x_{0,1} \\ x_{1,0} & x_{1,1} \\ x_{2,0} & x_{2,1} \end{bmatrix}$$

```
> u:=(i,j)-> alpha^(i-1):U:=Matrix(2,1,u);
```

$$U := \begin{bmatrix} 1 \\ \alpha \end{bmatrix}$$

```
> with(LinearAlgebra):
> Y:= Multiply(X, U);
```

Warning, the assigned name GramSchmidt now has a global binding

$$Y := \begin{bmatrix} x_{0,0} + x_{0,1} \alpha \\ x_{1,0} + x_{1,1} \alpha \\ x_{2,0} + x_{2,1} \alpha \end{bmatrix}$$

```
> v[1]:=f(0, 1, x[2]);
```

$$v_1 := 1 + x_2^3$$

```
> v[2]:=f(1, x[1], x[2]);
```

$$v_2 := 1 + x_1^3 + x_2^3$$

```
> vv[1]:=subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],v[1]) ;
```

$$vv_1 := 1 + (x_{2,0} + x_{2,1} \alpha)^3$$

```
> vv[2]:= subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],v[2]) ;
```

$$vv_2 := 1 + (x_{1,0} + x_{1,1} \alpha)^3 + (x_{2,0} + x_{2,1} \alpha)^3$$

```

> c:=0;
> if f(0,0,1) mod 2 =0 then c:=c+1;
> print (c, [0, 0, 1]) fi;
> for i from 0 to 1 do
>   for j from 0 to 1 do
>     if Eval(vv[1],{x[2,0]=i,x[2,1]=j}) mod 2 =0 then c:=c+1;
>     print (c,[0, 1, i+j*alpha]) fi ;od ;od;
>   for i2 from 0 to 1 do
>     for j2 from 0 to 1 do
>       for i1 from 0 to 1 do
>         for j1 from 0 to 1 do
>           if Eval(vv[2],{x[2,0]=i2,x[2,1]=j2, x[1,0]=i1,x[1,1]=j1}) mod 2
>           =0 then c:=c+1;
>           print (c,[1, i1+j1*alpha, i2+j2*alpha])
>           fi od; od ;od ;od;

```

$$\begin{aligned}
& c := 0 \\
& 1, [0, 1, \alpha] \\
& 2, [0, 1, 1] \\
& 3, [0, 1, 1 + \alpha] \\
& 4, [1, \alpha, 0] \\
& 5, [1, 1, 0] \\
& 6, [1, 1 + \alpha, 0] \\
& 7, [1, 0, \alpha] \\
& 8, [1, 0, 1] \\
& 9, [1, 0, 1 + \alpha]
\end{aligned}$$

Dans ce cas  $q = 2$ ,  $n = q^3 = 8$ ,  $m = 2$ ,  $k = 2$ ,  $d = 6$ , et on obtient un  $[8, 2, 6]_4$ -code  $C$  donné par l'espace vectoriel

$$\left\langle 1, \frac{u}{v+w} \right\rangle$$

puisque la fonction

$$\frac{u}{v+w} = \frac{u(v^2 + vw + w^2)}{(v+w)(v^2 + vw + w^2)} = \frac{(v^2 + vw + w^2)}{u^2}$$

est *finie* aux points

$$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

et appartient à  $\mathcal{L}(2x_\infty)$ .

EXERCICE. 1) Calculer une matrice génératrice et une matrice de contrôle de  $C$ .

EXERCICE. 2) Construire un code analogue pour  $m = 2$ .

REMARQUE 7.37 *Ainsi, la classe des codes de Goppa peut être vue comme une généralisation des codes BCH. On peut d'ailleurs étendre la méthode de construction des codes BCH vue dans Section 5 pour définir les codes de Goppa (comme dans Théorème 7.9. Cette classe possède plusieurs avantages sur la classe des codes BCH, et notamment celui de fournir de bons codes de grande longueur.*

Nous avons vu déjà (Théorème 7.9), qu'il existe une bonne famille de codes de Goppa sur  $\mathbb{F}_q$  dont le rendement tend vers la borne de Gilbert-Varshamov.

## 7.11 Codes géométriques de Goppa : $\Omega$ -construction.

A un couple  $(D_0, D)$  et une courbe lisse  $X$ , on peut également associer un autre code de Goppa. Considérons l'espace de formes différentielles

$$\Omega(D_0 - D) = \{\omega \in \Omega(X)^* \mid (\omega) + D_0 - D \geq 0\} \cup \{0\}$$

c'est à dire l'espace des formes différentielles ayant des zéros de multiplicités appropriées dans  $\text{Supp}D$  et des pôles au plus simples dans  $\text{Supp}D_0$ . Alors l'application

$$\begin{aligned} \text{Res} : \Omega(D_0 - D) &\rightarrow \mathbb{F}_{q^n} \\ \omega &\mapsto (\text{Res}_{x_1}(\omega), \dots, \text{Res}_{x_n}(\omega)) \end{aligned}$$

définit un code  $q$ -aire linéaire  $C^* = C^*(D_0, D)$  de longueur  $n$ . Or, nous avons vu que l'on pouvait identifier  $\Omega(D_0 - D)$  et  $\mathcal{L}(K + D_0 - D)$  par le morphisme

$$\omega \mapsto \omega/\omega_0 = f$$

où  $\omega_0$  est une forme différentielle rationnelle sur  $X$  fixée.

## Conclusion

Nous avons vu que la théorie des codes correcteurs d'erreurs se doit de relever plusieurs défis s'opposant parfois. En effet, les paramètres principaux d'un code de dimension donnée sont sa vitesse de transmission et sa distance relative. Or, afin d'augmenter la vitesse de transmission, il convient de limiter la longueur du code et donc sa distance relative, ce qui réduit le nombre d'erreurs que le code est capable de corriger. Ainsi, selon le besoin et les moyens, on pourra privilégier l'un ou l'autre de ces deux paramètres. D'autre part, nous avons vu dans la dernière partie de l'étude la classe des codes de Goppa, qui se construisent à partir d'une courbe projective lisse et de deux diviseurs. Un moyen d'optimiser ces codes et d'en construire de performants est d'étudier plus en profondeur ces objets (comme par exemple les courbes elliptiques, modulaires ...) ainsi que les bornes concernant les paramètres des codes construits sur ces objets.

## 8 Exercices de préparation à l'examen

### 8.1 Estimation des sommes binomiales.

Pour tout  $z \in ]0, 1]$ , soit

$$g(z) = \sum_{i=0}^n (q-1)^i \binom{n}{i} z^{i-r} = z^{-r} (1 + (q-1)z)^n.$$

Montrer que  $g(z)$  atteint le minimum en

$$z_0 = \frac{r}{(q-1)(n-r)} = \frac{r/n}{(q-1) \left(1 - \frac{r}{n}\right)},$$

et le minimum est

$$g(z_0) = (q-1)^r \left(\frac{r}{n}\right)^{-r} \left(1 - \frac{r}{n}\right)^{n-r}.$$

### 8.2 Encadrement de la probabilité du decodage erroné.

Soit  $p = p_s$  la probabilité des perturbations de symboles au cours de la transmission. On considère  $F = \{0, 1\}$  et l'application  $E : F \rightarrow F^n$  du codage de répétition pure. Calculer la probabilité  $P$  du decodage erroné.



(a) Montrer que

$$P = \sum_{0 \leq l < n/2} \binom{n}{l} (1-p)^l p^{n-l} = \mathcal{O}(p^{n/2}) \text{ lorsque } p \rightarrow 0$$

(b) Pour tout  $p < \frac{1}{8}$ , encadrer  $P$  (voir [vLi], p.24).

(a) Calculer le volume de la boule de Hamming  $V_q(n, t)$ ,  $t = 1, 2, 3$ .

### 8.3 Codes de Golay et empilement de sphères.

(a) Calculer le volume de la boule de Hamming  $V_q(n, t)$ ,  $t = 1, 2, 3$ .

(b) Montrer que la borne de Hamming est atteinte pour les codes  $G_{23}$  et  $G_{11}$  donc on a un empilement de sphères parfait.

### 8.4 Description géométrique des codes de Reed-Solomon

(voir [Pa-Wo], p.139).

Soit  $\mathbb{F}_{2^m} = \langle \alpha \rangle$ , et soit  $p(x) \in \mathbb{F}_{2^m}[x]$  parcourt le sous-espace linéaire  $\mathcal{P}_k \subset \mathbb{F}_{2^m}[x]$  avec  $d^\circ(p) \leq k-1$  ou  $p \equiv 0$ . Alors l'ensemble des mots de la forme

$$(p(1), p(\alpha), \dots, p(\alpha^{2^m-2}))$$

est un code de Reed-Solomon de longueur  $n = 2^m - 1$ , de polynôme

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^r)$$

avec  $r = 2^m - 1 - k$ .

*Solution.* On remarque que l'application

$$\Phi : p \mapsto (p(1), p(\alpha), \dots, p(\alpha^{2^m-2})) \in \mathbb{F}_q^n$$

est injective puisque  $\text{Ker}(\Phi) = 0$  par l'interpolation de Lagrange. On pose  $C_1 = \text{Im}(\Phi) = \Phi(\mathcal{P}_k)$ . Une base convenable de  $C_1$  :

$$c_i = \Phi(p_i), p_i = x^i \quad (i = 0, 1, \dots, k-1) : c_i = (1, \alpha^i, \alpha^{2i}, \dots, \alpha^{(2^m-2)i}),$$

donc  $c_i(x) := \sum_{j=0}^{2^m-2} \alpha^{ji} x^j$ .

Ensuite, on calcule le syndrome du polynôme  $c_i(x)$  :

$$c_i(\alpha^t) := \sum_{j=0}^{2^m-2} (\alpha^{i+t})^j = \sum_{j=0}^{2^m-2} \beta^j, \text{ avec } \beta = \alpha^{i+t}.$$

Puisque  $1 \leq t \leq r$ ,  $0 \leq i \leq k-1$ , alors  $1 \leq i+t \leq r+k-1 = 2^m-2$ , donc

$$\beta \neq 1, c_i(\alpha^t) = \frac{\beta^{2^m-2} - 1}{\beta - 1} = 0.$$

Ceci dit,  $C_1 \subset C$ , puisque tous les mots de  $C_1$  ont le syndrome nul. Mais

$$\dim_{\mathbb{F}_{2^m}} C = \dim_{\mathbb{F}_{2^m}} C_1 = k$$

donc  $C = C_1$ .

Montrer un résultat analogue pour un corps fini  $\mathbb{F}_q$  arbitraire.

## 8.5 Codes de Reed-Muller d'ordre 1.

Soit  $L_m = \mathbb{F}_q[T_1, \dots, T_m]_{d \leq 1}$  le  $\mathbb{F}_q$ -espace vectoriel de tous les polynômes de degré 1 ou 0 de  $m$  variables sur  $\mathbb{F}_q$  (avec le polynôme nul), donc  $\dim_{\mathbb{F}_q} L_m = m + 1$ . On considère un sous ensemble

$$\mathcal{P} = \{P_1, \dots, P_n\} \subset \mathbb{F}_q^m$$

tel que

$$\forall l \in L_m (\forall j = 1, \dots, n, l(P_j) = 0 \Rightarrow l \equiv 0).$$

$$RM_q(1, n, m) = \text{Im} Ev, \quad Ev : L_m \rightarrow \mathbb{F}_q^n, \quad f \mapsto ((f(P_1), \dots, f(P_n)))$$

est dit le code de Reed-Muller d'ordre 1.

Vérifier que  $RM_q(1, n, m)$  est un  $[n, m + 1, n - q^{m-1}]$ -code : tout polynôme de degré 1 de  $m$  variables s'annule en  $q^{m-1}$  points.

1) Donner une version projective du code : on considère le  $\mathbb{F}_q$ -espace vectoriel  $L'_m$  de tous les polynômes homogènes de degré 1 de  $m + 1$  variables sur  $\mathbb{F}_q$  (avec le polynôme nul), donc  $\dim_{\mathbb{F}_q} L'_m = m + 1$ . On obtient un

$$\left[ n, m + 1, n - \frac{q^m - 1}{q - 1} \right]_q$$

-code pour tout

$$n \geq \frac{q^m - 1}{q - 1}.$$

2) Montrer que pour

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

on obtient un code se trouvant sur la borne de Plotkin (voir Théorème 6.1) de paramètres

$$\left[ \frac{q^{m+1} - 1}{q - 1}, m + 1, q^m \right]_q.$$

3) *Code correcteur de Reed-Muller d'ordre 1 et de longueur 32.* Trouver une matrice de contrôle du code donné par toutes les combinaisons linéaires mod 2 des lignes  $A_i$  de

la matrice

$$\begin{pmatrix} \alpha_1 & 1, 1 \\ \alpha_2 & 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 \\ \alpha_3 & 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1 \\ \alpha_4 & 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1 \\ \alpha_5 & 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1 \\ \alpha_6 & 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 \end{pmatrix} \quad (*)$$

$$\mathbb{F}_2^6 \ni u = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) \xrightarrow{E} E(u) := A_1\alpha_1 + A_2\alpha_2 + A_3\alpha_3 + A_4\alpha_4 + A_5\alpha_5 + A_6\alpha_6 \in \mathbb{F}_2^{32}$$

## 8.6 Exemples de decodage des codes cycliques.

Soit  $\alpha$  un élément primitif de  $\mathbb{F}_{16}$  racine du polynôme  $x^4 + x + 1$  sur  $F_2$ ,  $m^{(i)}(x)$  le polynôme minimale unitaire de  $\alpha^i$ . Alors

$$m^{(1)}(x) = m^{(2)}(x) = m^{(4)}(x) = m^{(8)}(x) = x^4 + x + 1$$

$$m^{(3)}(x) = m^{(6)}(x) = m^{(12)}(x) = x^4 + x^3 + x^2 + x + 1$$

Ainsi, un code de polynôme générateur

$$g(x) = m^{(1)}(x)m^{(3)}(x) = 1 + x^4 + x^6 + x^7 + x^8$$

est un code cyclique dont les paramètres sont  $d = 5, q = 2, n = 15$ . Sa dimension est  $k = 7$  et un polynôme de correction est

$$h(x) = (x^{15} - 1)/g(x) = 1 + x^4 + x^6 + x^7$$

On construit une matrice génératrice  $G$  dont la  $i^{\text{ème}}$  ligne est le vecteur  $x^{i-1}g(x)$ ,  $1 \leq i \leq k$ .

(a) Montrer que

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

(b) Montrer que la distance du code est 5 donc ce code corrige deux erreurs.

(c) On considère les composantes

$$S_1 = \sum_{i=0}^{14} v_i \alpha^i, \quad S_3 = \sum_{i=0}^{14} v_i \alpha^{3i}.$$

du syndrome  $S(v) = Hv^t$ . Alors  $v \in C$  si et seulement si  $S(v) = Hv^t = 0$ . Supposons que le vecteur reçu  $v = (v_0, \dots, v_{14})$  contient au plus deux erreurs. Par exemple  $e(X) = X^{a_1} + X^{a_2}$ , où  $0 \leq a_1, a_2 \leq 14$ ,  $a_1 \neq a_2$ . Alors

$$S_1 = \alpha^{a_1} + \alpha^{a_2}, \quad S_3 = \alpha^{3a_1} + \alpha^{3a_2}.$$

Soit  $\eta_1 = \alpha^{a_1}$ ,  $\eta_2 = \alpha^{a_2}$  les locateurs des erreurs, alors

$$S_1 = \eta_1 + \eta_2, \quad S_3 = \eta_1^3 + \eta_2^3,$$

Montrer que

$$S_3 = S_1^3 + S_1^2\eta_1 + S_1\eta_1^2,$$

donc

$$1 + S_1\eta_1^{-1} + (S_1^2 + S_3S_1^{-1})\eta_1^{-2} = 0.$$

$$1 + S_1\eta_2^{-1} + (S_1^2 + S_3S_1^{-1})\eta_2^{-2} = 0.$$

(d) Montrer que s'il y a deux erreurs,  $\eta_1^{-1}$  et  $\eta_2^{-1}$  sont des racines différentes du polynôme

$$s(X) = 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2.$$

S'il n'y a qu'une seule erreur,  $S_1 = \eta_1$ ,  $S_3 = \eta_1^3$  donc  $S_1^3 + S_3 = 0$ , et on a

$$s(X) = 1 + S_1X.$$

S'il n'y a pas d'erreurs,  $S_1 = S_3 = 0$ , et on a reçu message correct  $w$ . Si  $S_1 \neq 0$  et  $S_1^3 + S_3 = 0$ , le polynôme  $s(X)$  a une seule racine dans  $\mathbb{F}_{16}$ . Si

$$s(X) = 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2$$

n'a pas de racines dans  $\mathbb{F}_{16}$ , le vecteur d'erreurs  $e(X)$  a plus que deux composantes non nuls, et il n'est pas possible de corriger les erreurs à l'aide de ce code.

(e) Soit par exemple le mot reçu a la forme

$$v = (1001110000000000).$$

Montrer que  $a_1 = 8$  et  $a_2 = 14$ . Corriger le mot  $v$ .

*Solution.* On a  $S(v) = (S_1(v), S_3(v))$  est donné par

$$S_1 = 1 + \alpha^3 + \alpha^4 + \alpha^5 = \alpha^2 + \alpha^3,$$

$$S_3 = 1 + \alpha^9 + \alpha^{12} + \alpha^{15} = 1 + \alpha^2.$$

(rappelons que

$$\begin{aligned} \alpha^4 &= 1 + \alpha, \alpha^5 = \alpha + \alpha^2, \alpha^6 = \alpha^2 + \alpha^3, \alpha^7 = 1 + \alpha + \alpha^3, \alpha^8 = 1 + \alpha^2, \\ \alpha^9 &= \alpha + \alpha^3, \alpha^{10} = 1 + \alpha + \alpha^2, \alpha^{11} = \alpha + \alpha^2 + \alpha^3, \alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3, \\ \alpha^{13} &= 1 + \alpha^2 + \alpha^3, \alpha^{14} = 1 + \alpha^3, \alpha^{15} = 1). \end{aligned}$$



Le polynôme  $s(X)$  a la forme suivante :

$$\begin{aligned} s(X) &= 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2 = \\ &= 1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^2 + \alpha^3 + (1 + \alpha^2)(\alpha^2 + \alpha^3)^{-1})X^2 \\ &= 1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^3)X^2. \end{aligned}$$

On trouve les racines de ce polynôme :  $X = \alpha$  et  $X = \alpha^7$ . Alors  $\eta_1^{-1} = \alpha$  et  $\eta_2^{-1} = \alpha^7$ , c'est à dire  $\eta_1 = \alpha^{14}$ ,  $\eta_2 = \alpha^8$ . On connaît alors les erreurs : elles sont dans les positions correspondantes aux  $X^8$  et  $X^{14}$ , c'est à dire la 9<sup>e</sup> et la 15<sup>e</sup> composantes de  $v$ . Alors le mot transmis était

$$w = (100111001000001).$$

On décode ce mot par la division du polynôme correspondant par le polynôme générateur  $g(X)$ . On obtient le polynôme  $1 + X^3 + X^5 + X^6$  et le reste nul. Alors le message initial était

$$(1001011)$$

## 8.7 Codes de Reed-Solomon

Soit  $\mathbb{F}_{256}^* = \langle \alpha \rangle$ ,  $g = \prod_{j=1}^{43} (x - \alpha^{11j})$ ,  $\alpha^8 = \alpha^7 + \alpha^2 + \alpha + 1$ . Alors  $n = 255$ ,  $k = 223$  (un code de Reed-Solomon utilisé par NASA). Montrer que  $d = 33$ .

## 8.8 Décodage des codes de Goppa rationnels.

Soit  $g(x)$  un polynôme unitaire irréductible sur  $\mathbb{F}_{q^m}$ ,  $L = \{\gamma_0, \gamma_1, \dots, \gamma_{n-1}\} \subset \mathbb{F}_{q^m}$  avec  $g(\gamma_i) \neq 0$  pour tout  $i = 1, 2, \dots, n-1$ .

Le code de Goppa rationnel  $\Gamma(L, g)$  est

$$\Gamma(L, g) = \left\{ (c_0, c_1, \dots, c_{n-1}) \in \mathbb{F}_q^n \mid \sum_{i=0}^{n-1} \frac{c_i}{x - \gamma_i} \equiv 0 \pmod{g(x)} \right\}$$

(c'est-à-dire, la fraction à droite est de la forme d'une fraction irréductible  $\frac{a(x)}{b(x)}$  avec  $g(x) \nmid a(x)$ ).

alors on considère les fractions suivantes

$$v(x) = \sum_{i=0}^{n-1} \frac{v_i}{x - \gamma_i}, \quad w(x) = \sum_{i=0}^{n-1} \frac{w_i}{x - \gamma_i}, \quad e(x) = \sum_{i=0}^{n-1} \frac{e_i}{x - \gamma_i}.$$

On utilise comme syndrome la fraction rationnelle

$$\sum_{i=0}^{n-1} \frac{e_i}{x - \gamma_i}$$

(le vecteur d'erreurs), représenté comme un polynôme mod  $g$

$$S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \bmod g$$

(le polynôme de syndrome).

(a) Montrer que si  $v = (v_0, v_1, \dots, v_{n-1}) = w + e$ , un mot reçu,  $w = (w_0, w_1, \dots, w_{n-1}) \in \Gamma(L, g)$ ,  $e = (e_0, e_1, \dots, e_{n-1}) \in \mathbb{F}_q^n$ , alors

$$S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \equiv \sum_{i=0}^{n-1} v_i f_i(x) \bmod g$$

puisque

$$\sum_{i=0}^{n-1} e_i f_i(x) \equiv 0$$

(le vecteur d'erreurs).

(b) Soit  $I := \{i \mid e_i \neq 0\}$ , et on considère le polynôme *locateur d'erreurs*

$$s(x) = \prod_{i \in I} (x - \gamma_i),$$

et le polynôme *évaluateur d'erreurs*

$$u(x) = \sum_{i \in I} e_i \prod_{j \in I \setminus \{i\}} (x - \gamma_j).$$

Montrer que  $u(x), s(x)$  satisfont la congruence :

$$u(x) \equiv s(x)S(x) \pmod{g(x)} \quad (8.1)$$

(d) *Une version de l'algorithme de Berlekamp-Massey.* Pour résoudre la congruence (8.1) on utilise la division euclidienne et l'identité de Bezout : on calcule trois suites  $s_n(x), t_n(x), u_n(x)$  avec la propriété

$$t_n(x)g(x) + s_n(x)S(x) = u_n(x)$$

où le degré de  $u_n(x)$  décroît jusqu'à ce que  $\deg(u_{j+1}) < t$  et  $\deg(u_j) \geq t$  ( $t = [(\deg g - 1)/2]$ ), à partir de

$$0 \cdot g(x) + 1 \cdot S(x) = S(x), 1 \cdot g(x) + 0 \cdot S(x) = g(x)$$

de telle façon que

$$(s_0(x), t_0(x), u_0(x)) = (0, 1, S(x)), (s_1(x), t_1(x), u_1(x)) = (1, 0, g(x)).$$

Décrire un algorithme pour trouver  $s(x)$  et  $u(x)$ .

## 8.9 Codes de Hermite.

On considère la courbe projective et lisse  $X \subset \mathbb{P}_{\mathbb{F}_q}^2$

$$X : u^{q+1} + v^{q+1} + w^{q+1} = 0$$

de genre  $g(X) = \frac{q(q-1)}{2}$ . Elle contient  $3(q+1)$  points sur  $\mathbb{F}_{q^2}$  avec  $uvw = 0$  :

$$(1, \zeta, 0), (0, 1, \zeta), (1, 0, \zeta) \in X(\mathbb{F}_{q^2}) \text{ avec } \zeta^{q+1} = 1, \zeta \in \mathbb{F}_{q^2}^*,$$

de plus, il y a  $(q-2)(q+1)^2$  point avec  $uvw \neq 0$  : dans ce cas on peut supposer  $u = 1$ ,  $v^{q+1} \neq -1$ , et le nombre de solution de l'équation  $w^{q+1} = a \in \mathbb{F}_q^*$  égale à  $q+1$ . Donc

$$\text{Card}(X(\mathbb{F}_{q^2})) = 3(q+1) + (q-2)(q+1)^2 = q^3 + 1 = q^2 + 1 + 2g(X)q.$$

(a) Montrer que le nombre des points  $X(\mathbb{F}_{q^2})$  de la courbe  $X$  est maximal possible.

On pose  $n = q^3$

$$D_0 = \{x_1, \dots, x_n\}, x_i \in X(\mathbb{F}_{q^2}) \setminus x_\infty, (x_\infty = (0, 1, 1)), D_0 = m \cdot x_\infty.$$

Soit  $C = C(D_0, D) = \mathcal{L}(D_0 - D)$  le code de Goppa correspondant, c'est un

$$[q^3, m - g + 1, d]_q - \text{code}, \text{ avec } d \geq n - m, \text{ avec le choix de } m : q^2 - q \leq m < q^3.$$

Si  $q = 2$ ,  $g(X) = 1$ ,  $X$  est une courbe elliptique,  $\text{Card}(X(\mathbb{F}_4)) = 9$

(b) Dans ce cas  $q = 2$ ,  $n = q^3 = 8$ ,  $m = 2$ ,  $k = 2$ ,  $d = 6$ , et on obtient un  $[8, 2, 6]_4$ -code  $C$  donné par l'espace vectoriel

$$\left\langle 1, \frac{u}{v+w} \right\rangle$$

puisque la fonction

$$\frac{u}{v+w} = \frac{u(v^2 + vw + w^2)}{(v+w)(v^2 + vw + w^2)} = \frac{(v^2 + vw + w^2)}{u^2}$$

est *finie* aux points

$$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

et appartient à  $\mathcal{L}(2x_\infty)$ . Calculer une matrice génératrice et une matrice de contrôle de  $C$ .

(c) Construire un code analogue pour  $m = 3$ .

## 8.10 Exemen du 13 janvier 2003, 10h-13h, Salle 014

1. a) Soient  $F$  un ensemble fini de cardinal  $q$ ,  $d(x, y)$  la distance de Hamming sur  $F^n$ , et

$$B_q(x, r) = \{y \in F^n \mid d(x, y) \leq r\} \subset F^n.$$

Trouver  $V_q(n, r) = \text{Card}(B_q(x, r))$ .

b) Encadrer  $V_5(3n, n)$ .

2. Soit  $p = p_s$  la probabilité des perturbations de symboles au cours de la transmission sur un canal bruité. On considère  $F = \{0, 1\}$  et l'application  $E : F^2 \rightarrow F^{2n}$  du codage de répétition pure.

(a) Calculer la probabilité  $P$  du decodage erroné.

(b) Pour tout  $p < \frac{1}{100}$ , encadrer  $P$

3. (a) Est-ce-que la borne de Singleton est atteinte pour les codes de Golay  $G_{23}$  et  $G_{11}$  ?

(b) Même question pour la borne de Plotkin.

(c) Montrer que la borne de Hamming est atteinte pour les codes  $G_{23}$  et  $G_{11}$ .

4. Soit  $\mathbb{F}_q$  un corps fini de  $q \geq 3$  éléments,  $k, d$  deux nombres entiers.

(a) Montrer que pour tout couple  $(k, d)$  avec  $k + d = q$  il existe un code  $C$  de type  $[q - 1, k, d]_q$ .

(b) Pour quelles valeurs  $k, d$  un tel code  $C$  soit parfait ?

(c) Pour  $q = 8$ ,  $k = 5$ ,  $d = 3$ , construire une matrice de contrôle de  $C$ .

5. On considère le  $\mathbb{F}_q$ -espace vectoriel  $V_m$  de tous les polynômes homogènes de degré 1 de  $m + 1$  variables sur  $\mathbb{F}_q$  (avec le polynôme nul), donc  $\dim_{\mathbb{F}_q} V_m = m + 1$  et soit

$$\mathcal{P} = \{P_1, \dots, P_n\} \subset \mathbb{F}_q^{m+1} \setminus \{0\}$$

un sous-ensemble tel que l'espace vectoriel engendré par les  $\{P_1, \dots, P_n\}$  coïncide avec  $\mathbb{F}_q^{m+1}$ , et les points  $P_i$  ne sont pas proportionnels.

a) Montrer qu'on obtient un  $\left[ n, m + 1, n - \frac{q^m - 1}{q - 1} \right]_q$ -code pour tout  $n \geq \frac{q^m - 1}{q - 1}$ .

b) Montrer que pour

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

un tel code atteint la borne de Plotkin.

6. CODE CORRECTEUR DE REED-MULLER D'ORDRE 1 ET DE LONGUEUR 32.

Décrire une matrice de contrôle du code donné par toutes les combinaisons linéaires



mod 2 des lignes  $A_i$  de la matrice

[illegible]

## 8.11 Exemen du 17 décembre 2003, 16h30-18h30, Salle 014

1. (a) Trouver une matrice génératrice et une matrice de contrôle du code cyclique

$$C = (g) \subset \mathbb{F}_2[x]/(x^{15} - 1)$$

de longueur 15, où  $g = x^4 + x + 1$  est un polynôme sur  $\mathbb{F}_2$ .

(b) Trouver la distance  $d$  du code  $C$ , et sa capacité de correction.

(c) Le code  $C$  est-il parfait ?

(d) On suppose qu'au plus une erreur s'est produite au cours de la transmission. Décoder le message

$$v = (101101000111001),$$

trouver le message émis et le mot d'information.

2. a) Soient  $F$  un ensemble fini de cardinal  $q$ ,  $d(x, y)$  la distance de Hamming sur  $F^n$ , et

$$B_q(x, r) = \{y \in F^n \mid d(x, y) \leq r\} \subset F^n.$$

Encadrer  $V_q(n, r) = \text{Card}(B_q(x, r))$ .

b) Calculer  $V_3(11, 2)$ .

3. Soit  $\mathbb{F}_q$  un corps fini de  $q \geq 3$  éléments,  $k, d$  deux nombres entiers.

- (a) Pour tout couple  $(k, d)$  avec  $k + d = q$  construire un code  $C$  de type  $[q - 1, k, d]_q$ .
- (b) Pour  $q = 4, k = 2, d = 2$ , construire une matrice de contrôle de  $C$ .

4. CODES DE REED-MULLER D'ORDRE 1. Soit  $L_m = \mathbb{F}_q[T_1, \dots, T_m]_{d \leq 1}$  le  $\mathbb{F}_q$ -espace vectoriel de tous les polynômes de degré  $\leq 1$  de  $m$  variables sur  $\mathbb{F}_q$  (avec le polynôme nul).

- (a) Montrer que  $\dim_{\mathbb{F}_q} L_m = m + 1$ .
- (b) On considère un sous ensemble du cardinal  $n$

$$\mathcal{P} = \{P_1, \dots, P_n\} \subset \mathbb{F}_q^m$$

tel que

$$\forall l \in L_m (\forall j = 1, \dots, n, l(P_j) = 0 \Rightarrow l \equiv 0).$$

Le code

$$RM_q(1, n, m) = \text{Im} Ev, \quad Ev : L_m \rightarrow \mathbb{F}_q^n, \quad f \mapsto ((f(P_1), \dots, f(P_n)))$$

est dit le **code de Reed-Muller d'ordre 1**.

Montrer que  $RM_q(1, n, m)$  est un  $[n, m + 1, n - q^{m-1}]$ -code.

- c) Pour quelles valeurs de  $n$  un tel code atteinte la borne de Plotkin ?

## 8.12 Contrôle continu du jeudi 2 décembre 2004, 8h00–10h00, AMPHI

1. a) Soient  $F$  un ensemble fini de cardinal  $q$ ,  $d(x, y)$  la distance de Hamming sur  $F^n$ , et

$$B_q(x, r) = \{y \in F^n \mid d(x, y) \leq r\} \subset F^n.$$

Trouver  $V_q(n, r) = \text{Card}(B_q(x, r))$ .

b) Encadrer  $V_5(3n, n)$  à l'aide de la fonction d'entropie  $H_5(x)$ , où  $H_q(\delta)$  est la fonction d'entropie  $q$ -aire : pour  $\delta \in [0, \frac{q-1}{q}]$

$$H_q(0) = 0, H_q(\delta) = \frac{\delta \log(q-1)}{\log(q)} - \frac{\delta \log(\delta)}{\log(q)} - \frac{(1-\delta) \log(1-\delta)}{\log(q)}.$$

2. (a) Est-ce-que la borne de Singleton est atteinte pour les codes de Golay  $G_{23}$  et  $G_{11}$  ?

(b) Montrer que la borne de Hamming (d'empilement de sphères) est atteinte pour les codes  $G_{23}$  et  $G_{11}$ .

(c) Même question pour les codes de Hamming

3. Soit  $\mathbb{F}_q$  un corps fini de  $q \geq 3$  éléments,  $k, d$  deux nombres entiers.

(a) Montrer que pour tout couple  $(k, d)$  avec  $k + d = q$  il existe un code  $C$  de type  $[q - 1, k, d]_q$ .

(b) Pour quelles valeurs  $k, d$  un tel code  $C$  soit parfait ?

(c) Pour  $q = 8, k = 5, d = 3$ , construire une matrice de contrôle de  $C$ .

4. Soit  $C_3$  le code de Hamming binaire de longueur 7 et de dimension 4. Alors sa matrice de correction est

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

(a) Pour le message reçu  $y = (1101101)$  trouver le syndrome  $S(y)$ ,

(b) trouver la position d'une erreur éventuelle d'une transmission avec  $C_3$ ,

(c) trouver le polynôme énumérateur des poids  $A(z)$  du code  $C_3$

## A Annexe : Rappels sur les corps finis

Cette partie est une somme d'éléments d'algèbre utiles dans la compréhension de la théorie des codes correcteurs d'erreurs, sans faire partie à proprement parler de la théorie en elle-même qui est le sujet de l'étude. C'est pourquoi rien de ce qui suit n'est démontré. Pour un plus large développement sur les corps finis et les démonstrations de ce qui est affirmé ci-dessous, le lecteur est renvoyé aux textes d'E.Peyre [Pey], ainsi que Lidl-Niederreiter, [Li-Ni].

## A.1 Structure

PROPOSITION-DÉFINITION A.1 : *Soit  $A$  un anneau commutatif de caractéristique  $p$  un nombre premier. L'application*

$$Fr_p : x \mapsto x^p, x \in A$$

*est un morphisme d'anneau appelé morphisme de Frobenius. Plus généralement, si  $A$  est un anneau commutatif de caractéristique  $p$  premier et si  $q$  est une puissance de  $p$ , on note  $Fr_q : x \mapsto x^q$ .*

THÉORÈME A.2 : *Soit  $K$  un corps fini. Alors  $K$  est de caractéristique  $p$  premier,  $K$  est de cardinal  $q = p^d$ , avec  $d = [K : \mathbb{F}_p]$ . Inversement, si  $p$  est premier,  $d$  est un entier, il existe à isomorphisme près un unique corps à  $q = p^d$  éléments, qui est le corps de décomposition de  $X^q - X$  sur  $\mathbb{F}_p$ . On note ce corps  $\mathbb{F}_q$ . De plus, on a l'existence de deux isomorphismes : un de  $(\mathbb{F}_q, +)$  sur  $((\mathbb{Z}/p\mathbb{Z})^d, +)$  et un du groupe multiplicatif  $\mathbb{F}_q^*$  sur  $\mathbb{Z}/(q-1)\mathbb{Z}$  ( $\mathbb{F}_q^*$  est un groupe cyclique d'ordre  $q-1$ ).*

THÉORÈME A.3 : *Soit  $q = p^n$ . Tout sous-corps de  $\mathbb{F}_q$  est d'ordre  $p^m$ , où  $m$  est un diviseur de  $n$ .*

## A.2 Polynômes sur les corps finis

THÉORÈME A.4 : Soit  $p$  premier et  $q$  une puissance de  $p$ . Pour tout entier  $d$  strictement positif, il existe un polynôme  $P$  irréductible de degré  $d$  sur  $\mathbb{F}_q$  et  $\mathbb{F}_{q^d}$  est isomorphe à  $\mathbb{F}_q[T]/(P)$ .

REMARQUE A.5 Si on a un tel polynôme unitaire et  $r$  une racine de  $P$  dans  $\mathbb{F}_q$ , la famille  $\{1, r, \dots, r^{d-1}\}$  est une base du  $\mathbb{F}^p$ -espace vectoriel  $\mathbb{F}_q$ .

THÉORÈME A.6 : Soit  $P$  un polynôme irréductible sur  $\mathbb{F}_q$  et  $r$  une racine de  $P$  dans une extension de  $\mathbb{F}_q$ . Alors, pour tout polynôme  $Q$  sur  $\mathbb{F}_q$ ,  $Q(r) = 0$  si et seulement si  $Q$  divise  $P$ .

LEMME A.7 Soit  $P$  un polynôme irréductible sur  $\mathbb{F}_q$  de degré  $m$ . Alors  $P$  divise  $x^{q^n} - x$  si et seulement si  $m$  divise  $n$ .

THÉORÈME A.8 : Soit  $P$  un polynôme irréductible sur  $\mathbb{F}_q$  de degré  $m$ . Alors  $P$  possède une racine  $r$  dans  $\mathbb{F}_{q^m}$ . De plus, toutes les racines de  $P$  sont simples et sont données par  $r, r^q, \dots, r^{q^{m-1}}$ , éléments distincts de  $\mathbb{F}_{q^m}$ .

COROLLAIRE A.9 : Le corps de décomposition d'un polynôme  $P$  de degré  $m$  irréductible sur  $\mathbb{F}_q$  est  $\mathbb{F}_{q^m}$ .

DÉFINITION A.10 Soit  $P$  un polynôme sur  $\mathbb{F}_q$  tel que  $P(0) \neq 0$ . L'ordre de  $P$  est le plus petit entier positif  $e$  tel que  $P$  divise  $x^e - 1$ . Si  $P(0)=0$ , alors il existe  $Q$  sur  $\mathbb{F}_q$  non nul en 0 et  $h$  un entier positif tels que  $P = x^h Q$ , et dans ce cas,  $\text{ord}(P) = \text{ord}(Q)$ .

REMARQUE A.11 Si  $P$  est irréductible de degré  $m$  sur  $\mathbb{F}_q$ , alors l'ordre de  $P$  divise  $q^m - 1$ .

THÉORÈME A.12 Le nombre de polynômes irréductibles unitaires sur  $\mathbb{F}_q$  de degré  $m$  et d'ordre  $e$  est

$$N_{q,m,e} = \varphi(e)/m \text{ si } e > 1.$$

où  $\varphi(e)$  est l'indicateur d'Euler de  $e$ .

REMARQUE A.13 : Le degré de  $P$  irréductible sur  $\mathbb{F}_q$  d'ordre  $e$  est l'ordre multiplicatif de  $q$  modulo  $e$ .



DÉFINITION A.14 *Un polynôme  $P$  de degré  $m$  sur  $\mathbb{F}_q$  est dit primitif sur  $\mathbb{F}_q$  s'il est le polynôme minimal sur  $\mathbb{F}_q$  d'un élément primitif de  $\mathbb{F}_{q^m}$ .*

THÉORÈME A.15  *$P$  est primitif sur  $\mathbb{F}_q$  si et seulement si  $P$  est unitaire, non nul en 0 et d'ordre  $q^m - 1$ .*

DÉFINITION A.16 *On appelle fonction de Moebius la fonction définie sur  $\mathbb{N}$  par :*

$$\mu(n) = \begin{cases} 1 & , \text{ si } n = 1 \\ (-1)^k & , \text{ si } n \text{ est le produit de } k \text{ nombres premiers distincts,} \\ 0 & , \text{ si } n \text{ est divisible par le carré d'un nombre premier.} \end{cases}$$

THÉORÈME A.17 *Le nombre de polynômes irréductibles unitaires de degré  $n$  sur  $\mathbb{F}_q$  est*

$$N_q(n) = \frac{1}{n} \sum_{d|n} \mu(n/d) q^d = \frac{1}{n} \sum_{d|n} \mu(d) q^{n/d}$$

*La somme se faisant sur tous les diviseurs positifs de  $n$ .*

THÉORÈME A.18 Soit  $\alpha$  un élément de  $\mathbb{F}_{q^m}$ , une extension de  $\mathbb{F}_q$ . Soit  $d$  le degré de  $\delta$  sur  $\mathbb{F}_q$  et  $P$  le polynôme minimal de  $\alpha$  sur  $\mathbb{F}_q$ . Alors,

- (i)  $P$  est irréductible sur  $\mathbb{F}_q$  et son degré  $d$  divise  $m$ .
- (ii)  $Q$  polynôme sur  $\mathbb{F}_q$  est tel que  $Q(\alpha) = 0$  si et seulement si  $P$  divise  $Q$ .
- (iii)  $Q$  polynôme irréductible unitaire sur  $\mathbb{F}_q$  et tel que  $Q(\alpha) = 0$  est tel que  $P = Q$ .
- (iv)  $P$  divise  $x^{q^d} - x$  et  $x^{q^m} - x$ .
- (v) Les racines de  $P$  sont  $\alpha, \alpha^q, \dots, \alpha^{q^{d-1}}$  et  $P$  est le polynôme minimal sur  $\mathbb{F}_q$  de toutes ces racines.
- (vi) Si  $P(\alpha) = 0$ , alors l'ordre de  $P$  est égal à celui de  $\alpha$  dans le groupe multiplicatif  $\mathbb{F}_{q^m}^*$ .
- (vii)  $P$  est un polynôme primitif sur  $\mathbb{F}_q$  si et seulement si  $\alpha$  est d'ordre  $q^d - 1$  dans  $\mathbb{F}_{q^m}^*$ .

THÉORÈME A.19 (CRITÈRE D'IRRÉDUCTIBILITÉ) Soit  $P$  un polynôme de degré  $d$  sur  $\mathbb{F}_q$ . Alors  $P$  est irréductible si et seulement si le rang de  $Fr_q - 1$  est égal à  $d - 1$ .

## Références

### Livres de base :

[Dem] DEMAZURE, MICHEL *Cours d'algèbre. Primalité. Divisibilité. Codes.*, Nou-

velle Bibliothèque Mathématique [New Mathematics Library], 1. Cassini, Paris, 1997. xviii+302 pp.

- [La] LANG, SERGE *Algebra*. Reading, Mass. : Addison–Wesley (1965).
- [vLi] J.H. VAN LINT, *Introduction to coding theory*. Springer-Verlag, 3<sup>e</sup> édition, 1999
- [vLi-vdG] J.H. VAN LINT, G. VAN DER GEER, *Introduction to coding theory and algebraic geometry*. Springer-Verlag, New-York
- [Li-Ni] RUDOLF LIDL et HARALD NIEDERREITER, *Introduction to finite fields and their applications*. Addison–Wesley : Reading, 1983
- [MW-S] MCWILLIAMS F.J., SLOANE N.J.A., *The theory of error–correcting codes*, North – Holland, Amsterdam, 1977
- [Pa-Wo] PAPINI, O., ET WOLFMAN, J., *Algèbre discrète et codes correcteurs*, Collection Math. et Applications, Springer-Verlag, 1995

### **Livres supplémentaires :**

- [Ebe02] W. EBELING, *Lattices and Codes. A cours Partially Based on Lectures by F. Hirzebruch*. Second Ed. Advanced Lectures in Mathematics. Vieweg. xi, 187 p., 2002
- [Se70] J.– P. SERRE, *Cours d’arithmétique*. Paris : Presses Univ. France, 1970.

- [Ste] S. A. STEPANOV, *Codes on algebraic curves*. Kluwer Academic Publishers. vii, 350 p., 1999
- [Ts-V] TSFASMAN, M.A. et VLADUT, S.G. , *Algebraic-geometric codes*. Kluwer Academic Publishers. xxiv, 667 p., 1991

### Articles et autres sources :

- [Aug] AUGOT, DANIEL *Les travaux de M.Sudan sur les codes correcteurs d'erreurs*, Gazette des mathématiciens, SMF, No 98, octobre 2003, 5-13
- [Pey] EMMANUEL PEYRE, *Corps finis et courbes elliptiques*. DESS Cryptologie, sécurité et codage d'information, Modules A1A et A1B, Grenoble, 2002, pp. 1-128
- [Pre] OLIVER PRETZEL, *Codes and Algebraic Curves* (Oxford Lecture Series in Mathematics and Its Applications, 8), 1998
- [Sha] SHANNON, C.E : *A mathematical theory of communication*, Bell Syst. Techn. J., SMF, **27**, 379-423, 623-656 (1948)