Mathématiques des codes correcteurs d'erreurs

A. A. Pantchichkine*

Institut Fourier, B.P.74, 38402 St.–Martin d'Hères, FRANCE
e-mail : 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.

*Un cours dans le cadre du DESS "Cryptologie, Sécurité et Codage d'Information" (Institut Fourier), 2003-2004, Module C1A, préparé avec la participation de J.-B.Werner

Table des matières

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, qui sont présentées dans la deuxième partie du cours.

Les applications des codes correcteurs d'erreurs dans l'industrie concernent le 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.

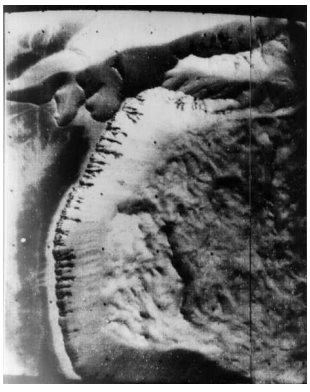
5. Codes de Reed-Solomon et codes BCH. Codage et décodage.

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 procedure 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 énentuelles 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 represente chaque message par une suite de 32 bits :

$$u = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6),$$

toutes les combinaisons linéaires mod 2 des lignes A_i de la matrice

$$\begin{matrix}\alpha_1\\ \alpha_2\\ \alpha_3\\ \alpha_4\\ \alpha_5\\ \alpha_6\end{matrix}\begin{pmatrix}1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\\ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\\ 0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1\\ 0,0,0,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1\\ 0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1\\ 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1\end{pmatrix} \qquad (*)$$

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

Cette transmission a la *redondance* = 32-6=26, le *coefficient de redondance*= 32/6.

Pour décrire les mots du code $C=\mathrm{Im}(E)$, on indexe les colonnes $A^{(j)}$ de la matrice (*) par les points $x=(x_1,x_2,x_3,x_5)$ 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

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

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

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

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

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

(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

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

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

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

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

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

$$\begin{aligned}
&A_2\longleftrightarrow f\equiv 1, f(x_1,x_2,x_3,x_5)=x_1 \text{ "le premier chiffre" de } j-1 \bmod 2\in\mathbb{F}_2,\\
&A_3\longleftrightarrow f\equiv 1, f(x_1,x_2,x_3,x_5)=x_2 \text{ "le second chiffre" de } j-1 \bmod 2\in\mathbb{F}_2,\\
&A_4\longleftrightarrow f\equiv 1, f(x_1,x_2,x_3,x_5)=x_3 \text{ "le troisième chiffre" de } j-1 \bmod 2\in\mathbb{F}_2,\\
&A_5\longleftrightarrow f\equiv 1, f(x_1,x_2,x_3,x_5)=x_4 \text{ "le quatrième chiffre" de } j-1 \bmod 2\in\mathbb{F}_2,\\
&A_6\longleftrightarrow f\equiv 1, f(x_1,x_2,x_3,x_5)=x_5 \text{ "le dernier chiffre" de } j-1 \bmod 2\in\mathbb{F}_2.
\end{aligned}$$

Ceci implique que la combinaison linéaire représant 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

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

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 ecartés : 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 ≤ 14 position.

Décodage d'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 ,

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

alors

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

Dans ce cas on vérifie si l'identité (0.2) est satisfaite en tous les autres 26 points (26 = 32-6 conditions à vérifier !) Sinon, on considère toutes les 64 combinaisons linéaires

$$y(x_1, x_2, x_3, x_4, x_5) - (\beta_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5), \tag{0.2}$$

et on choisit parmi eux un mot $e = e(x_1, x_2, x_3, x_4, x_5)$ avec le nombre minimum des valeurs non nulles. C'est "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 d'un même code auront la même longueur (le nombre de symboles d'un alphabet fini F). De tels mots sont appelés "codes en blocs" opposés aux "codes de longueur variable" ou "codes convolutionnels" dont nous ne parlerons pas.

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

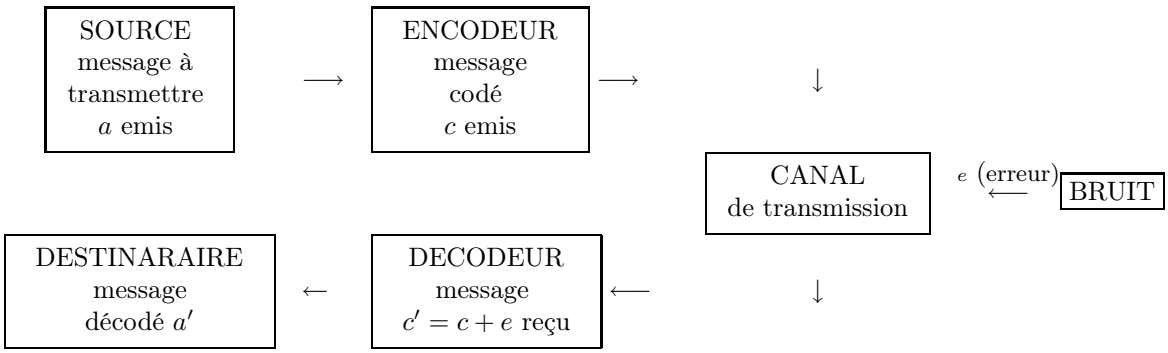
Codes de répétition pure.

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 déscription 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 assoocie 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* $\mathrm{Card}\,(C) = M$.

(b) *Un code $C \subset F^n$ est dit q-aire si $C = \mathrm{Im}E$ 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 $\mathrm{Card}\,(C) = q^k$

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

$$(a, b) \mapsto \mathrm{Card}\,\{i \in \{0, 1, \ldots, n\} \mid a_i \neq b_i\}$$

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

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

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

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 :*

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

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

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

Définition 1.4 *Soit $C \subset F^n$ un code* (a) *On appelle écart ou "distance" de C le nombre*

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

Soit $C = \mathrm{Im}E$ un code q-aire, $E : F^k \to F^n$ 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 $x = (x_1, \ldots, x_n)$ le mot transmis par le canal. Le *mot reçu* $y = (y_1, y_2, \ldots, y_n)$, eventuellement 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*

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

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

(b) *On dit que D est standard ("de vraisemblence 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

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

où

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

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

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

avec les hypothèses 1.2. :

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

Il est possible de diminuer considerablement la proportion d'erreures 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 ε, 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 ε et de n on peut supposer aussi que $p \leq \frac{\delta}{2} - \varepsilon - \frac{1}{n}$. Dans ce cas

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

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

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

Donc le succès du decodage des mots $\tilde{E}_N(a) \in F^{nN}$ dépend du fait si δ est assez grand $p \leq \frac{\delta}{2}$ mais aussi depend 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 1.9 *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*

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

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 erreures d'impression pendant la lecture on utilise un bon vocabulaire où les mots sont écartés : il n'y a qu'un seul mot qui resemble au mot imprimé (erroné).

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

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

Une autre méthode est celle de répétition :

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

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 : $\dfrac{k_m}{n_m} = \dfrac{1}{m} \to 0$.

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 $n \le km_0$, $\frac{k}{n} \ge \frac{1}{m_0}$. On voit donc que la répétition est inutile à cause de la restriction sur la longuer

$$k = 1, \frac{k}{n} \ge \frac{1}{m_0} \Rightarrow n \le m_0.$$

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

(a) Montrer que

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

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

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

1.5. Théorème de Shannon (1948)

dit qu'il existe des bonnes familles des codes. Shannon a considéré un canal binaire bruité ($q = 2$).

THÉORÈME 1.10 (SHANNON) *Soit* $0 < R < 1 + p\log_2 p + (1-p)\log_2(1-p) < 1$ *et on pose* $M_n := 2^{[Rn]}$. *Soit* $P^*(M_n, n, p)$= *la valeur minimale de la probabilité de decodage incorrecte par un mot sur tous les codes* $C \subset \mathbb{F}_2^n$ *du cardinal* $\mathrm{Card}\,(C) = M_n$. *Alors* $P^*(M_n, n, p) \to 0$ *si* $n \to \infty$.

(voir [vLi], p.27) pour la preuve détaillée.

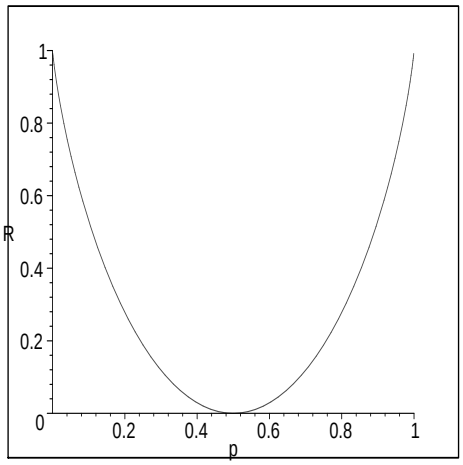
Représentation graphique : fonction d'entropie

```
> 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)';
```

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

$$\mathrm{H}(p) = -\frac{p\log(p)}{\log(2)} - \frac{(1-p)\log(1-p)}{\log(2)}$$

```
> 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

$$\mathrm{Card}\,(C_n) = M_n = 2^{[Rn]} \to \infty$$

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

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

Remarque 1.12 *Dans le cas d'un canal q-aire on utilise la* borne de Gilbert -Varshamov $R > 1 - H_q(\delta)$ *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 $C = \mathrm{Im}E$ un $[n,k,d]_q$-code, $E : F^k \to F^n$. Rapellons 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* $t = \left[\frac{d-1}{2}\right]$ *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 $\mathrm{Card}\, B(x,r)$ ne dépend pas de $x \in C$: $\mathrm{Card}\, B(x,r) = |B(x,r)| =: V_q(n,r)$ ("le volume de la boule de Hamming").

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

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

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

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

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

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

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 $\mathrm{Card}\,(F) = q$) de distance $d = d(C)$, la capacité de correction $t = \left[\frac{d-1}{2}\right]$ et de cardinal $M = \mathrm{Card}\,(C)$. Alors*

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

PREUVE : Dans F^n , il y a $\binom{n}{m}(q-1)^m$ vecteurs de poids de Hamming m. Le code corrige t erreurs, donc les boules centrées sur les mots codes et de rayon t sont 2 à 2 disjointes. Chacune des M boules contient $1 + \binom{n}{1}(q-1) + \cdots + \binom{n}{t}(q-1)^t$ vecteurs de F_q^n, qui en contient en tout q^n.

COROLLAIRE 2.4 *Soit $C \subset F^n$ un code sur l'alphabet F de $\mathrm{Card}\,(F) = q$ de distance $d = d(C)$ et la capacité de correction $t = \left[\frac{d-1}{2}\right]$. Alors*

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

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

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

OPTIMISATION DES PARAMÈTRES : on considère les codes $C \subset F^n$ sur l'alphabet F de $\mathrm{Card}\,(F) = q$ de cardinal $\mathrm{Card}\,(C) = M$, 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 $C = \mathrm{Im}E$ un $[n,k,d]_q$-code, $E : F^k \to F^n$. Alors*

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

où $R = k/n$, $\delta = d/n$.

PREUVE. On pose

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

alors l'application $E' : F^k \to F^{n-d+1}$ obtenue de $E : F^k \to F^n$ 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 δ assez grands dans [0, 1]. Rapellons (voir Definition1.9) qu'une famille $\{C_i\}$ des $[n_i, k_i, d_i]_q$-codes est dite bonne s'il existe et positives les limites

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

Toutefois, ces paramètres δ 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*

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

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 $i \to \infty$, 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

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

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

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

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 $g'(z)$:

> `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

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

Ceci implique

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

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 :

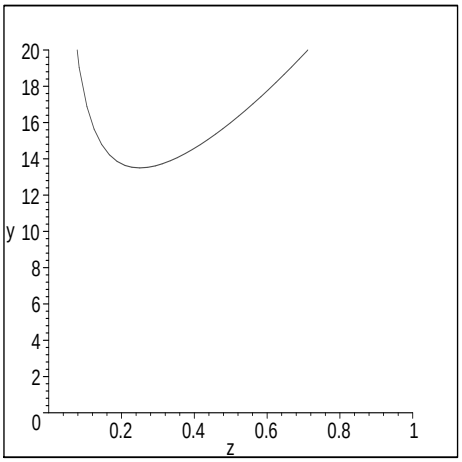
```
> restart;q:=3;r:=1;n:=3;
> g(z):=z^(-r)*(1+(q-1)*z)^n;
> gprime(z):=diff(z^(-r)*(1+(q-1)*z)^n,z);
```

$$q := 3$$
$$r := 1$$
$$n := 3$$
$$\mathrm{g}(z) := \frac{(1+2\,z)^3}{z}$$
$$\mathrm{gprime}(z) := -\frac{(1+2\,z)^3}{z^2} + \frac{6\,(1+2\,z)^2}{z}$$

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

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

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



$$y = \mathrm{g}(z)$$

D'autre part

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

d'où l'inégalité supérieure.

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

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

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 $A_r\tilde{z}^r = \binom{n}{i}(q-1)^r\tilde{z}^r$ soit maximal parmi $A_i\tilde{z}^i$:

$$\frac{A_{i-1}\tilde{z}^{i-1}}{A_i\tilde{z}^i} = \frac{1}{\tilde{z}}\binom{n}{i-1}\binom{n}{i}^{-1}(q-1)^{-1} = \tag{2.3}$$
$$\frac{1}{\tilde{z}}\cdot\frac{n!}{(i-1)!(n-i+1)!}\cdot\frac{i!(n-i)!}{n!}\frac{1}{\tilde{z}}\frac{A_{i-1}}{A_i} = \frac{1}{\tilde{z}}\frac{i}{(n-i+1)(q-1)}.$$

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

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

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

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

Ceci implique

$$A_{i-1}\tilde{z}^{i-1} \le A_i\tilde{z}^i \le \cdots \le A_r\tilde{z}^r, A_r\tilde{z}^r \ge A_{r+1}\tilde{z}^{r+1} \ge \cdots \ge A_j\tilde{z}^j \ge \ldots.$$

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

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

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 :

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

Corollaire 2.10 *Lorsque $n\to\infty$, $\frac{r}{n}\to\delta$, on a*

$$H_q(\delta) = \lim_{n\to\infty}\frac{\log_q V_q(n,[\delta n])}{n} = \lim_{n\to\infty}\log_q\left(\frac{V_q(n,[\delta n])}{\mathrm{Card}\,(F^n)}\right)$$

(la "proportion logarithmique de la boule de Hamming de rayon relative" δ dans F^n).

Preuve. On prend $\log_q$ de l'inégalité (2.1) :

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

et si $\frac{r}{n}\to\delta, n\to\infty$ alors $\frac{\log_q(n+1)}{n}\to 0$ 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*

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

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 $t_i = \left[\frac{d_i-1}{2}\right]$. Alors

$$\frac{d_i-2}{2} \leq t_i = \left[\frac{d_i-1}{2}\right] \leq \frac{d_i-1}{2} = \frac{n_i\delta_i - 1}{2} \Rightarrow \lim_{i\to\infty}\frac{t_i}{n_i} = \lim_{i\to\infty}\frac{n_i\delta_i-1}{2n_i} = \frac{\delta}{2}$$

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

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

Il vient que

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

```
> 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);
```

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

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

```
> f:=Hq(x);
```

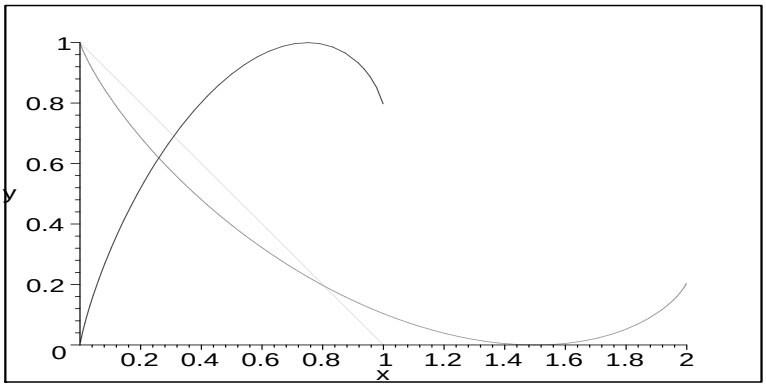
$$f := \frac{x\ln(3)}{\ln(4)} - \frac{x\ln(x)}{\ln(4)} - \frac{(1-x)\ln(1-x)}{\ln(4)}$$

```
> g1:=x/2;g:=algsubs(x=g1,f);
```

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

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

```
> plot([f,1-g(x),1-x], x=0..2,y=0..1);'y=Hq(x),1-Hq(x/2),1-x';
```



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

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 comme l'écriture matricielle.

En général, pour un alphabet fini F, étant donné une énumération de F^k, la donnée d'un code $C = \mathrm{Im}(E)$, $E : F^k \to F^n$ 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 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 $\mathbb{F}_q$.

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 $E : F^k \to F^n$ linéaire injective). Les vecteurs lignes $a = (a_1, \cdots, a_k) \in F^k$ sont les* mots d'information*, et les vecteurs lignes $c = E(a) = (c_1, \cdots, c_n) \in F^n$ sont les* mots de code*. La* matrice génératrice *G du code E est la matrice attachée à l'application linéaire $E : F^k \to F^n$ (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 $S : F^n \to F^{n-k}$. 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 :

$$C = \Big\{ c = (c_1, \cdots, c_n) \;\Big|\; Hc^t = 0 \Big\}.$$

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 a $E : \mathbb{F}_2^4 \to \mathbb{F}_2^7$

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

La matrice H est appelée *matrice de contrôle* de C.

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 systématique. Ici $A \in \mathrm{Mat}_{n-k,k}(F)$*

De plus, on a

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

c'est-à-dire,

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

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*

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

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 $d(y, c') \leq t$ 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 déja 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

$$E : \quad \mathbb{F}_2^4 \to \mathbb{F}_2^7 \tag{3.1}$$

$$(a_1,a_2,a_3,a_4) \mapsto (a_1,a_2,a_3,a_4,a_1+a_3+a_4,a_1+a_2+a_4,a_1+a_2+a_3) \tag{3.2}$$

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 $w(c) \geq s$ et donc $d \geq s$.

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

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

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

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

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	$\begin{smallmatrix}0\\1\end{smallmatrix}$
		⏟ Leaders de classe				⏟ 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^{me} colonne de H, et soit $y = x + e$ le message reçu, $x \in C$. Alors, $S(y) = He^t$. Si

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

alors

$$S(y) = h_{i1} + h_{i2} + \cdots$$

Si toutes les colonnes de H sont différentes, une erreur simple en i^{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 $1, 2, \cdots, 2^m - 1$.*

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^{me} 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 finis $\mathbb{F}_q$. 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 representant les coordonnées homogènes de points différents de l'espace projectif $\mathbb{P}^{m-1}_{\mathbb{F}_q}$*

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

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

donc

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

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

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

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

3.4. Codes cycliques

DÉFINITION 3.17 *Un code linéaire $C \subset \mathbb{F}_q^n$ est dit cyclique si*

$$(a_0, \cdots, a_{n-1}) \in C \Longleftrightarrow (a_{n-1}, a_0, \ldots, a_{n-2}) \in C$$

Pour la suite, nous supposons que 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 par exemple

$$(a_0, \cdots, a_{n-1}) \leftrightarrow a_0 + a_1 x + a_2 x^2 + ... + a_{n-1}x^{n-1}$$

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 $f \in \mathbb{F}_q[x]/(x^n-1)$, et $g_1, g_2 \in \mathbb{F}_q[x]$, alors $g_1 g_2 = f \Longleftrightarrow g_1 g_2 \equiv f \bmod (x^n-1)$.

3.5. Constructions

3.5.1. 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

$$(a_0, \cdots, a_{n-1}) \longleftrightarrow f(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

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

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

de l'anneau quotient

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

THÉORÈME 3.18 *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 $(a_0, \cdots, a_{n-1}) \in C$, alors

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

Inversement, si C est cyclique, pour tout $a(x) \in C$, $xa(x) \in C$, $x^2 a(x) \in C$ 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 :

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

3.5.2. Construction par polynôme correcteur

Si $g(x) = g_0 + g_1 x + \cdots + g_{n-k}x^{n-k}$, une *matrice génératrice* du code C est

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}$$

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

PROPOSITION 3.19 *Si $h(x) = (x^n - 1)/g(x) = h_0 + \cdots + h_k x^k$, alors*

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

est une matrice de contrôle de C.

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

$$\begin{aligned} h(x) \quad &= (x^n - 1)/g(x) \\ &= h_0 + h_1 x + h_2 x^2 + \ldots + h_k x^k = \sum_{j=0}^{k} h_j x^j = \sum_{j=0}^{n-1} h_j x^j \text{ où } h_j = 0 \text{ pour } j \geq k+1. \end{aligned} \tag{3.3}$$

Alors une *condition nécessaire* pour que

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

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

$$f(x) = g(x) \cdot u(x) \Rightarrow h(x) \cdot f(x) = h(x) g(x) \cdot u(x) \equiv 0 \bmod (x^n - 1).$$

On calcule donc le produit

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

De plus,

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

On pose $l = i + j$, alors

$$\begin{aligned} h(x) \cdot f(x) \quad &= \sum_{j=0}^{k} \sum_{i=0}^{n-1} h_j a_i x^{i+j} \\ &= \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} h_{l-i} a_i x^l + \sum_{l=n}^{2n-1} \sum_{i=0}^{n-1} h_{l-i} a_i x^l \\ &\equiv \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} h_{l-i} a_i x^l + \sum_{l'=0}^{n-1} \sum_{i=0}^{n-1} h_{l'+n-i} a_i x^{l'} \bmod (x^n - 1) \\ &= \sum_{l=0}^{n-1} (\sum_{i=0}^{n-1} h_{l-i} a_i + \sum_{i=0}^{n-1} h_{l+n-i} a_i) x^l \end{aligned} \tag{3.4}$$

(on a utilisé la notation $l' = l - n$ pour $l \geq n$). Puis, on observe que la somme sur l' dans (3.4) est nulle dès que $l' \geq k$ puisque $l' + n - i \geq k + 1$ et donc $h_{l'+n-i} = 0$ par (3.3).

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

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

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

$$\sum_{i=0}^{n-1} h_{l-i} a_i = 0, (l = k, k+1, \cdots, n-1) \tag{3.5}$$

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

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

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 vecoriel de C de dimension $n - (n - k) = k$. Mais la dimension du code C est égale à k donc la condition de contrôle (3.5) est *nécessaire et suffisante*.

DÉFINITION 3.20 *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 $v(x)h(x) \equiv 0 \bmod (x^n - 1)$. 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 vraisemblence maximale") d'un mot $v(x)$ on choisit parmi les polynômes

$$v(x) - \tilde{a}(x)g(x), \tilde{a}(x) = a'_0 + a'_1 x + \cdots a'_{k-1} x^{k-1}$$

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

$$\tilde{a}(x) = a'_0 + a'_1 x + \cdots a'_{k-1} x^{k-1}$$

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 $x^j = a_j(x)g(x) + r_j(x)$. Ainsi, $x^j - r_j(x) \in C$, ainsi que $g_j(x) = x^k(x^j - r_j(x))$ 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 $r_{n-k-1+i}(x)$:

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

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

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

```
> 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

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

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 $\alpha_1, \cdots, \alpha_s$ des éléments d'une extension $\mathbb{F}_{q^m}$ de $\mathbb{F}_q$, et $p_i(x)$ le polynôme minimal de α_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 $g(x) = ppcm(p_1(x), \cdots, p_s(x))$. Dans ces conditions, $g(x)$ divise $x^n - 1$ et si C est le code de polynôme générateur g, on a

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

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 $\alpha_1, \cdots, \alpha_{n-k}$. Alors, $f \in \mathbb{F}_q[x]/(x^n-1)$ est un mot du code si et seulement si le vecteur des coefficients de $f(f_0, \cdots, f_{n-1})$ est dans le noyau de*

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

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 e 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 α est un élément primitif de $\mathbb{F}_{2^m}$:

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

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

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

et

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

est une base de $\mathbb{F}_{2^m}$ sur $\mathbb{F}_2$. Soit alors H la matrice dont la $j^{\text{ème}}$ colonne est $(c_0, \cdots, c_{m-1})^t$, avec $\alpha^{j-1} = c_0 + c_1\alpha + \cdots + c_{m-1}\alpha^{m-1}$, et les c_i dans $\mathbb{F}_2$. Soit alors $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathbb{F}_2[x]$ avec $n = 2^m - 1$. On a $Ha^t = a(\alpha)$, exprimé dans la base $\{1, \alpha, \cdots, \alpha^{m-1}\}$. 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 α est un élément primitif de $\mathbb{F}_{16}$:

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

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

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

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

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

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

où

$$g_0(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1 = \prod_{i \in I}(x - \alpha^i), I = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$$

$$g_1(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1 = \prod_{j \in J}(x - \alpha^j), J = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\}$$

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 :

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

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

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

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;
```

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

```
> 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;
```

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

```
> 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 ;
```

$$i = 1,\ \alpha = \alpha,\ \mathrm{g}(\alpha) = 0$$
$$i = 2,\ \alpha^2 = \alpha^2,\ \mathrm{g}(\alpha^2) = 0$$
$$i = 3,\ \alpha^3 = \alpha^3,\ \mathrm{g}(\alpha^3) = 0$$
$$i = 4,\ \alpha^4 = \alpha^4,\ \mathrm{g}(\alpha^4) = 0$$
$$i = 6,\ \alpha^6 = \alpha^6,\ \mathrm{g}(\alpha^6) = 0$$
$$i = 8,\ \alpha^8 = \alpha^8,\ \mathrm{g}(\alpha^8) = 0$$
$$i = 9,\ \alpha^9 = \alpha^9,\ \mathrm{g}(\alpha^9) = 0$$
$$i = 12,\ \alpha^{12} = \alpha^{10}+\alpha^7+\alpha^4+\alpha^3+\alpha^2+\alpha+1,\ \mathrm{g}(\alpha^{12}) = 0$$
$$i = 13,\ \alpha^{13} = \alpha^{10}+\alpha^8+\alpha^6+\alpha^3+\alpha+1,\ \mathrm{g}(\alpha^{13}) = 0$$
$$i = 16,\ \alpha^{16} = \alpha^9+\alpha^8+\alpha^6+\alpha^5+\alpha^2+\alpha,\ \mathrm{g}(\alpha^{16}) = 0$$
$$i = 18,\ \alpha^{18} = \alpha^8+\alpha^7+\alpha^6+\alpha^5+\alpha^3+\alpha^2+1,\ \mathrm{g}(\alpha^{18}) = 0$$

```
> 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 ;
```

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

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

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

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

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

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

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

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

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

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

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

```
> g[1]:=x^11+x^9+x^7+x^6+x^5+x+1;Factor(g[1],alpha) mod 2;
```

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

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

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],
> [0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0],
> [0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0],
> [0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0],
> [0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0],
> [0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0],
> [0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0],
> [0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0],
> [0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0],
> [0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0],
> [0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0],
> [0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1]]) ;
```

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

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

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

```
> (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);
```

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

```
> '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;
```

$$1+x^{23}$$

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

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

```
> 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 $\mathrm{GF}(2^{11}) = F_2[\alpha]$. 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 $\alpha^{j-1} (j = 1, 2, ..., 23)$ dans la base

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

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

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

$$1 = 1$$
$$\alpha = \alpha$$
$$\alpha^2 = \alpha^2$$
$$\alpha^3 = \alpha^3$$
$$\alpha^4 = \alpha^4$$
$$\alpha^5 = \alpha^5$$
$$\alpha^6 = \alpha^6$$
$$\alpha^7 = \alpha^7$$
$$\alpha^8 = \alpha^8$$
$$\alpha^9 = \alpha^9$$
$$\alpha^{10} = \alpha^{10}$$
$$\alpha^{11} = \alpha^{10} + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1$$
$$\alpha^{12} = \alpha^{10} + \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$$
$$\alpha^{13} = \alpha^{10} + \alpha^8 + \alpha^6 + \alpha^3 + \alpha + 1$$
$$\alpha^{14} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1$$
$$\alpha^{15} = \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1$$
$$\alpha^{16} = \alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha$$
$$\alpha^{17} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^3 + \alpha^2$$
$$\alpha^{18} = \alpha^8 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + 1$$
$$\alpha^{19} = \alpha^9 + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha$$
$$\alpha^{20} = \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha^2$$
$$\alpha^{21} = \alpha^9 + \alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1$$
$$\alpha^{22} = \alpha^{10} + \alpha^9 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha$$

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

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

```
> 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 $\mathrm{GF}(2^{11}) = F_2[\alpha]$. 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-correctur ; 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

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

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

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

où

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

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

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 :

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

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

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

et avec le polynome de contrôle

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

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^11-1) mod 3;
```

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

```
> g:=x^5+2*x^3+x^2+2*x+2;irreduc(g) mod 3;
```

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

$$true$$

```
> 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^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 ;
```

$$i = 1,\ \alpha = \alpha,\ \mathrm{g}(\alpha) = 0$$

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

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

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

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

```
> 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 ;
```

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

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

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

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

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

```
> (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);
```

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

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

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

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

$g := x^5 + 2\,x^3 + x^2 + 2\,x + 2$
et avec le polynome de contrôle $h = (x+2)\,(x^5 + x^4 + 2\,x^3 + x^2 + 2)$,
$h := x^6 + x^4 + 2\,x^3 + 2\,x^2 + 2\,x + 1$.

C'est un [12,6, 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;
```

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

```
> 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 $\mathrm{GF}(3^5) = F_3[\alpha]$. 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-correctur ; 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 emplilement 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. Polynômes 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 α une racine $n^{\text{ème}}$ de l'unité, primitive dans $\mathbb{F}_{q^m}$ et considérons le code généré par $g(x)$, le polynôme minimal de α sur $\mathbb{F}_q$. Soit par exemple

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

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 $i \neq j$, $1 \leq i, j \leq n$, α^{j-1} détermine la position de l'erreur.

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 plusieures racines $\alpha_1, \cdots, \alpha_r$ de g pour détérminer le polynôme d'erreur $e(x) = \sum_{i=1}^{t'} c_i x^{a_i}$, ou les a_i sont tous différentes dans $0, 1, \cdots, n-1$, 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 $n^{\text{ème}}$ de l'unité α , primitive dans $\mathbb{F}_{q^m}$, disons $\alpha = \alpha_1$, et considérons le code engendré par $g(x)$, le polynôme minimal de α sur $\mathbb{F}_q$. On utilise les notations

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

Posons alors

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

et

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

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

$$\eta_i = \alpha^{a_i}$$

pour la racine α fixée, et on introduit le *polynôme locateur d'erreur*

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

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

Soit alors l'identité polynomiale

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

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

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

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

$$(-1)^{t'} \sigma_{t'} S_b + (-1)^{r-1} \sigma_{r-1} S_{b+1} + \cdots - \sigma_1 S_{b+t'-1} + S_{b+t'} = 0 \tag{4.5}$$

pour tout b.

Résumé de cette méthode : (on note $\tau_i = (-1)^i \sigma_i$).

- $1^{\text{ème}}$ étape : Déterminer le syndrome

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

Soit

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

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

- $3^{\text{ème}}$ étape : Chercher les racines de $s(x)$ en testant les différentes puissances de α pour déterminer les locateurs d'erreurs η_i

- $4^{\text{ème}}$ étape : Remplacer les η_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 $w = v - e$.

4.4.1. Exemples de decodage des codes cycliques

Soit α un élément primitif de $\mathbb{F}_{16}$ racine du polynôme $x^4 + x + 1$ sur F_2, $m^{(i)}(x)$ le polynôme minimal unitaire de α^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 $x^{i-1} g(x)$, $1 \leq 1 \leq k$. 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 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

$$S_1 = \eta_1 + \eta_2, \quad S_3 = \eta_1^3 + \eta_2^3,$$

ceci implique

$$S_3 = S_1^3 + S_1^2\eta_1 + S_1\eta_1^2,$$

donc

$$1 + S_1\eta_1^{-1} + (S_1^2 + S_3S_1^{-1})\eta_1^{-2} = 0.$$
$$1 + S_1\eta_2^{-1} + (S_1^2 + S_3S_1^{-1})\eta_2^{-2} = 0.$$

S'il y a deux erreurs, η_1^{-1} et η_2^{-1} sont des racines différentes de

$$s(X) = 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2.$$

S'il n'y a qu'une seul 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.

Soit par exemple le mot reçu a la forme

$$v = (100111000000000).$$

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

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

Le polynôme $s(X)$ a la forme suivante :

$$s(X) = 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2 =$$
$$1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^2 + \alpha^3 + (1 + \alpha^2)(\alpha^2 + \alpha^3)^{-1})X^2$$
$$= 1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^3)X^2.$$

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 connait alors les erreurs : elles sont dans les positions correspondantes aux X^8 et X^{14}, c'est à direla 9^e et la 15^e 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

$$(1001011)$$

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 τ_i provenant des relations (4.4) :

$$\begin{aligned} S_2\tau_1 + S_1\tau_2 &= S_3 \\ S_3\tau_1 + S_2\tau_2 &= S_4 \end{aligned}$$

qui s'écrit également

$$\begin{aligned} \tau_1 + \tau_2 &= \alpha^4 \\ \alpha^4\tau_1 + \tau_2 &= 1 \end{aligned}$$

et dont la matrice est régulière. Il vient

$$\begin{aligned} \tau_1 &= 1 \\ \tau_2 &= \alpha \end{aligned}$$

$\tau_0 = 1$ par définition, et on a donc $s(x) = 1 + x + \alpha x^2$.

En testant les différentes puissances de α, on trouve

$$\eta_1^{-1} = \alpha^8, \eta_2^{-1} = \alpha^6,$$

et on a donc

$$\begin{aligned} \eta_1 &= \alpha^7 \\ \eta_2 &= \alpha^9 \end{aligned}$$

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

$$a(x) = w(x)/g(x) = 1 + x^3 + x^4$$

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 $\alpha_1, \cdots, \alpha_s$ des éléments d'une extension de $\mathbb{F}_q$ et $p_i(x)$ le polynôme minimal de α_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 $g(x) = ppcm(p_1(x), \cdots, p_s(x))$. Dans ces conditions, $g(x)$ divise $x^n - 1$ et si C est le code du polynôme générateur g, on a

$$v(x) \in C \Longleftrightarrow v(\alpha_i) = 0, \quad i = 1, \cdots, s.$$

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 $n^{èmes}$ de l'unité (le corps de décomposition de $x^n - 1$ sur $\mathbb{F}_q$). Soit b un entier, et β une racine primitive $n^{èmes}$. Si $g(x)$ possède, parmi les racines dans une extension L de $F = \mathbb{F}_q$, les puissances de β dont les exposants sont $d' - 1$ entiers consécutifs, soit*

$$\beta^b, \beta^{b+1}, \ldots, \beta^{b+d'-2},$$

alors le poids du code C est supériereur 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 $\beta^b, \ldots, \beta^{b+d'-2}$. Dans le cas $b = 1$ on dit que c'est un code BCH au sens strict.*

EXEMPLE. Soit β 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 β, β^2, β^4, β^8 sont les racines de $m^{(1)}(x)$, et β^3 est une racine de $m^{(3)}(x) = x^4 + x^3 + x^2 + x + 1 = (x^5 - 1)/(x - 1)$ $(\beta^3)^5 = 1$, 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, β et β^2. Le poids minimum du code est ≥ 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 ;
```

$$i = 1,\ \mathrm{g}(\beta) = 0$$
$$i = 2,\ \mathrm{g}(\beta^2) = 0$$
$$i = 4,\ \mathrm{g}(\beta^4) = 0$$
$$i = 5,\ \mathrm{g}(\beta^5) = 0$$
$$i = 7,\ \mathrm{g}(\beta^7) = 0$$
$$i = 8,\ \mathrm{g}(\beta^8) = 0$$

```
> 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 β, $\beta^2, \ldots, \beta^{d'-1}$.

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 serons 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 ≥ 4 on peut partir des racines $\beta, \beta^2, \beta^3 \in \mathbb{F}_{64}$, donc $d' - 1 = 3$,

$$g(x) = (x^6 + x^3 + 1)\,(x^2 + x + 1) = x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^2 + x + 1.$$

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 THEORÈME 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 β une racine primitive $n^{\text{èmes}}$. Le polynôme $g(x)$ est un polynôme de degré minimal qui possède, parmi les racines dans une extesion L de $F = \mathbb{F}_q$, les puissances de β 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

$$\begin{cases} a_0 + a_1(\beta^b) + a_2(\beta^b)^2 + \cdots + a_{n-1}(\beta^b)^{n-1} = 0 \\ a_0 + a_1(\beta^{b+1}) + a_2(\beta^{b+1})^2 + \cdots + a_{n-1}(\beta^{b+1})^{n-1} = 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_0 + a_1(\beta^{b+d'-2}) + a_2(\beta^{b+d'-2})^2 + \cdots + a_{n-1}(\beta^{b+d'-2})^{n-1} = 0 \end{cases} \tag{5.1}$$

Soit $V_i = {}^t((\beta^b)^i, (\beta^{b+1})^i, \dots, (\beta^{b+d'-2})^i)$, alors le système (5.1) $\iff a_0 V_0 + \cdots + a_{n-1} V_{n-1} = 0$. On souhaite montrer que toutes les $d' - 1$ colonnes $V_{i_1}, V_{i_2}, \dots, V_{i_{d'-1}}$ sont linéairement indépendentes, 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 $u_j = \beta^{i_j}, j = 1, \cdots, d' - 1$. Donc le déterminant (de type de Van der Monde) est non nul :

$$\begin{vmatrix} \beta^{bi_1} & \beta^{bi_2} & \dots & \beta^{bi_{d'-1}} \\ \beta^{(b+1)i_1} & \beta^{(b+1)i_2} & \dots & \beta^{(b+1)i_{d'-1}} \\ \dots & \dots & \dots & \dots \\ \beta^{(b+d'-2)i_1} & \beta^{(b+d'-2)i_2} & \dots & \beta^{(b+d'-2)i_{d'-1}} \end{vmatrix} = \beta^{b(i_1+i_2+\cdots i_{d'-1})} \prod_{1 \leq k < j \leq d'-1} (\beta^{i_j} - \beta^{i_k}) \neq 0,$$

car $u_1 = \beta^{i_1}, \dots, u_{d'-1} = \beta^{i_{d'-1}}$ sont *distincts*.

En conséquance, 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 $n^{ème}$ 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 $\beta^b, \beta^{b+1}, \cdots, \beta^{b+d'-2}$.*

Si $m^{(i)}(x)$ est le polynôme minimal de β^i sur $\mathbb{F}_q$, le polynôme générateur du code C est

$$g(x) = \text{ppcm}(m^{(b)}(x), m^{(b+1)}(x), \cdots, m^{(b+d'-2)}(x))$$

(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 α élément primitif de $\mathbb{F}_{16}$. On construit alors une matrice de correction H dont la $i^{\text{ème}}$ colonne est α^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 cinqième 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 $\beta^b, \beta^{b+1}, \cdots, \beta^{b+d'-2} \in \mathbb{F}_q^*$.

On utilise souvent $q = 2^m$, et on va écrire α au lieu de β :

$$x^{2^m-1} - 1 = \prod_{u \in \mathbb{F}_{2^m}^*} (x - u) = (x - 1)(x - \alpha) \cdot \ldots \cdot (x - \alpha^{2^m-2})$$

donc

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

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'.

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 $\mathbb{F}_{256}^* = \langle\alpha\rangle$, $g = \prod_{12}^{43}(x - \alpha^{11j})$, $\alpha^8 = \alpha^7 + \alpha^2 + \alpha + 1$. Alors $n = 255, k = 223, d = 33$ (un code utilsé par NASA).

5.3. Deuxième déscription des codes de Reed-Solomon

THÉORÈME 5.5 (voir [Pa-Wo], p.139) *Soit $\mathbb{F}_{2^m} = \langle\alpha\rangle$, et soit $p(x) \in \mathbb{F}_{2^m}[x]$ pacourt 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*

$$(p(1), p(\alpha), \cdots, p(\alpha^{2^m-2}))$$

est un code de Reed-Solomon de longueur $n = 2^m - 1$, de polynôme

$$g(x) = (x-\alpha)(x-\alpha^2)\cdot\cdots\cdot(x-\alpha^r)$$

avec $r = 2^m - 1 - k$.

REMARQUE IMPORTANTE. Cette discription montre que les mots de code sont certaines *fonctions polynômiales* sur l'ensemble des racines d'un polynôme.

PREUVE. On remarque que l'application

$$\Phi : p \mapsto (p(1), p(\alpha), \ldots, p(\alpha^{2^m-2})) \in \mathbb{F}_q^n$$

est injective puisque $\mathrm{Ker}(\Phi) = 0$ par l'interpolation de Lagrange. On pose $C_1 = \mathrm{Im}(\Phi) = \Phi(\mathcal{P}_k)$.

Une base convenable de C_1 :

$$c_i = \Phi(p_i), p_i = x^i \quad (i = 0, 1, \cdots, k-1) : c_i = (1, \alpha^i, \alpha^{2i}, \cdots, \alpha^{(2^m-2)i}),$$

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

$$S(v) = Hv^t = (S_b, S_{b+1}, \cdots, S_{b+d'-2})^t$$

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 $0, 1, \cdots, n-1$. 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,

$$S_j = e(\beta^j) = \sum_{i=1}^{r} c_i \eta_i^j \quad (j = b, b+1, \cdots, b+d'-2).$$

Nous voyons donc que nous pourrons décoder dès lors que nous aurons obtenu les couples (c_i, η_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,*

$$S_j^q = (\sum_{i=1}^{r} c_i \eta_i^j)^q = \sum_{i=1}^{r} c_i^q \eta_i^{jq} = \sum_{i=1}^{r} c_i \eta_i^{jq} = S_{jq}$$

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 σ_i sont donc $\sigma_0 = 1$ et pour $1 \leq i \leq r$, les σ_i sont les polynômes élémentaires symétriques en $\eta_1, \cdots, \eta_r$. En remplaçant x par η_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

$$S_j = e(\beta^j) = \sum_{i=1}^{r} c_i \eta_i^j \quad (j = b, b+1, \cdots, b+d'-2).)$$

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 :

$$\begin{pmatrix} S_b & S_{b+1} & \cdots & S_{b+r-1} \\ S_{b+1} & S_{b+2} & \cdots & S_{b+r} \\ S_{b+r-1} & S_{b+r} & \cdots & S_{b+2r-2} \end{pmatrix} = VDV^t$$

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 η_i sont tous distincts et D est diagonale, donc régulière si et seulement si les c_i et les η_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 η_i^{-1}, c'est à dire les β^{-a_i}. Il suffit donc ensuite d'évaluer le polynôme $s(x)$ pour les différentes puissances de β pour localiser les erreurs.

APRÈS AVOIR TROUVÉ LES RACINES η_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 η_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

$$S(v) = (S_b, S_{b+1}, \cdots, S_{b+d'-2})^t.$$

(On rappelle que

$$S_j = e(\beta^j) = \sum_{i=1}^{r} c_i \eta_i^j \quad (j = b, b+1, \cdots, b+d'-2).)$$

- 2ème étape : Déterminer l'entier r maximal tel que le système d'équations

$$S_{j+r} + S_{j+r-1}\tau_1 + \cdots + S_j \tau_r = 0, \quad b \leq j \leq b+r-1$$

des inconnues τ_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 β pour déterminer les locateurs d'erreurs η_i .

- 4ème étape : Remplacer les η_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* $2^{ème}$. *Pour parvenir à déterminer les coefficients* τ_i, *une méthode possible est d'utiliser l'algorithme de Berlekamp-Massey (voir [Li-Ni], p.235-239, et [vLi-vdG], p.20).*

5.5. Algorithme de Berlekamp-Massey

(voir [vLi-vdG], p.20).

Soit C un $[n, k, d,]_q$-code BCH de zéroes $\beta, \beta^2, \ldots, \beta^{d'-1}$, ou $\mathbb{F}_q^* = \langle\beta\rangle$, $\beta^n = 1$. Supposons que v, w et e soient le message reçu, le message émis et le vecteur erreur, $v = w + e$, considères comme des polynômes : $v(x), w(x)$ et $e(x) \in \mathbb{F}_q[x]$.

On pose

$$v(x) = v_0 + \cdots + v_{n-1}x^{n-1}, w(x) = v_0 + \cdots + w_{n-1}x^{n-1}, e(x) := v(x) - w(x) = e_0 + \cdots + e_{n-1}x^{n-1},$$

et soit $I := \{i \mid e_i \neq 0\}$, et on considère le polynôme locateur d'erreurs

$$s(x) = \prod_{i \in I}(1 - \beta^i x),$$

et le polynôme évaluateur d'erreurs

$$u(x) = \sum_{i \in I} e_i \beta^i \prod_{j \in I \setminus \{i\}} (1 - \beta^j x).$$

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

$$\boxed{s'(\beta^{-i}) = -\beta^i \prod_{j\in I\setminus\{i\}} (1-\beta^{j-i}), \quad u(\beta^{-i}) = e_i\beta^i \prod_{j\in I\setminus\{i\}} (1-\beta^{j-i}).}$$

De plus $d^\circ(s(x)) \leq t = \left[\frac{d-1}{2}\right], d^\circ(u(x)) < t$. Il reste à trouver les polynômes $s(x)$, $u(x)$. On utilise l'identité formelle

$$\frac{u(x)}{s(x)} = \sum_{i\in I} \frac{e_i\beta^i}{1-\beta^i x} = \sum_{i\in I} e_i x^{-1} \sum_{l=1}^{\infty} (\beta^i x)^l = \sum_{l=1}^{\infty} x^{l-1} e(\beta^l), \tag{5.2}$$

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

$$S(x) = \sum_{l=1}^{d'-1} e(\beta^l) x^{l-1}$$

et on cherche $u(x), s(x)$ à partir de l'égalité (5.2) comme une solution de la congruence :

$$u(x) \equiv s(x)S(x) (\mathrm{mod} x^{d'-1}) \tag{5.3}$$

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'au $\mathrm{pgcd}(x^{d'-1}, S(x))$ est atteint, à partir de

$$0 \cdot x^{d'-1} + 1 \cdot S(x) = S(x), 1 \cdot x^{d'-1} + 0 \cdot S(x) = x^{d'-1}$$

de telle façon que

$$(s_0(x), t_0(x), u_0(x)) = (0, 1, S(x)); (s_1(x), t_1(x), u_1(x)) = (1, 0, x^{d'-1})$$

Il est claire que le couple $(u(x), s(x)) = (u_n(x), s_n(x))$ est un unique couple vérifiant (5.3).

EXERCICE. Ecrire un algorithme pour trouver $s(x)$ et $u(x)$.

Exemple : l'identité de Bezout et de l'algorithme de Berlekamp-Massey

```
> 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)) ;
```

$$irr34 := x^4 + 2\,x^3 + 2$$

- Le polynôme irréductible obtenu par ce procédé n'est pas forcément le même d'une session à l'autre.

- On aliasse α à une racine de ce polynôme dans une extension de F_3, de sorte que $\mathrm{GF}(3^4) = F_3[\,\alpha]$.

```
>  alias(alpha = RootOf(irr34) mod 3) ;
```

$$\alpha$$

- La procédure **rnd3** génére 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 GF(3^4).

```
>  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 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
```

$$P,\, U,\, V := X^2 + (1 + \alpha + 2\,\alpha^3)\,X + 1 + \alpha^2 + 2\,\alpha^3,$$
$$-\frac{-4\,\alpha^4 + 8\,\alpha^5 - 3\,\alpha^7 - 20\,\alpha^8 + 17\,\alpha^6 + 2\,\alpha^3 + 3\,\alpha^2 - 1 + 16\,\alpha^{10}}{(1 + 16\,\alpha^3 + 10\,\alpha^2 + 4\,\alpha + 16\,\alpha^5 + 10\,\alpha^6 + 19\,\alpha^4 + 4\,\alpha^7 + \alpha^8)\,\alpha^2}$$
$$-\frac{(\alpha^3 + 6\,\alpha^4 - 8\,\alpha^6 - 2\,\alpha^7 + 1 + 2\,\alpha - \alpha^2 + 2\,\alpha^5 + 8\,\alpha^8)\,X}{(1 + 16\,\alpha^3 + 10\,\alpha^2 + 4\,\alpha + 16\,\alpha^5 + 10\,\alpha^6 + 19\,\alpha^4 + 4\,\alpha^7 + \alpha^8)\,\alpha^2},$$
$$\frac{2\,\alpha^3 + 1 + \alpha^2}{\alpha\,(\alpha^2 + 1 + \alpha)} + \frac{(2\,\alpha^3 - 2\,\alpha^2 - \alpha + 1)\,\alpha\,X}{(\alpha^2 + 1 + \alpha)^2}$$
$$-\frac{(4\,\alpha^7 - 2\,\alpha^6 - 3\,\alpha^5 + 2\,\alpha^4 + \alpha^3 + 1)\,X^2}{(\alpha^2 + 1 + \alpha)\,(2\,\alpha + 3\,\alpha^2 + \alpha^4 + 2\,\alpha^3 + 1)\,\alpha^2}$$
$$+\frac{(\alpha^3 + 6\,\alpha^4 - 8\,\alpha^6 - 2\,\alpha^7 + 1 + 2\,\alpha - \alpha^2 + 2\,\alpha^5 + 8\,\alpha^8)\,X^3}{(2\,\alpha + 3\,\alpha^2 + \alpha^4 + 2\,\alpha^3 + 1)^2\,\alpha^2}$$

```
> 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;
```

$$tn := \alpha\,X + 2\,\alpha + 2$$

```
> sn:=sort(Expand(sn),[alpha,X]) mod 3;
```

$$sn := 2\,\alpha^3\,X^2 + \alpha^2\,X^2 + 2\,\alpha\,X^3 + \alpha^3 + \alpha^2\,X + \alpha\,X^2 + \alpha\,X + X^2 + \alpha + 2\,X + 1$$

```
> Expand(G - tn*A - sn*B) mod 3;
```

$$0$$

Pour résoudre la congruence

$$u(x) \equiv s(x)S(x)(\mathrm{mod}\, x^{d'-1})$$

on a utilisé la division euclidienne et l'identité de Bezout : on a calculé trois suites $s_n(x), t_n(x), u_n(x)$ avec la propriété

$$\boxed{t_n(x)x^{d'-1} + s_n(x)S(x) = u_n(x)}$$

Exemple. Soit β un élément primitif de $\mathbb{F}_{16}$ racine du polynôme $x^4 + x + 1$ sur 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 $x^{i-1}g(x)$, $1 \leq 1 \leq k$. 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 τ_i est de la forme

$$\begin{array}{l} S_2\tau_1 + S_1\tau_2 = S_3 \\ S_3\tau_1 + S_2\tau_2 = S_4 \end{array} \Longleftrightarrow \begin{array}{l} S_1\tau_2 + S_2\tau_1 = S_3 \\ S_2\tau_2 + S_3\tau_1 = S_4 \end{array}$$

qui s'écrit également

$$\begin{aligned} \tau_1 + \tau_2 &= \beta^4 \\ \beta^4\tau_1 + \tau_2 &= 1 \end{aligned}$$

et dont la matrice est régulière. Il vient

$$\begin{aligned} \tau_1 &= 1 \\ \tau_2 &= \beta \end{aligned}$$

$\tau_0 = 1$ par définition, et on a donc $s(x) = 1 + x + \beta x^2$.

En testant les différentes puissances de β, 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

$$a(x) = w(x)/g(x) = 1 + x^3 + x^4 \Rightarrow a = (1001100).$$

6. Bornes de Plotkin et de Gilbert-Varshamov

6.1. Rendement, taux de correction et domaine de codes

Soit $C \subset F^n$ un code de cardinal $\mathrm{Card}\,(C) = M$ sur l'alphabet F de $\mathrm{Card}\,(F) = q$ de distance $d = d(C)$ et la capacité de correction $t = \left[\frac{d-1}{2}\right]$. 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 δ assez grands dans $[0,\ 1]$. Rapellons (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

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

On a vu que avec toute bonne famille on peut faire tendre vers 1 la probabilité de transmission correcte. En réalité, on utilise $R \approx 0,1 \sim 0,95$.

Toutefois, ces paramètres δ 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 $C = \mathrm{Im}E$ un $[n, k, d]_q$-code, $E : F^k \to F^n$. Alors

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

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

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

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 $\mathrm{Card}\,(F) = q$ de distance $d = d(C)$ et la capacité de correction $t = \left[\frac{d-1}{2}\right]$. Alors

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

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

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

alors

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

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

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

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 $V_q \backslash U_q$ ont dits isolés.

On pose

$$A_q(n,d) = \max\left\{ \text{Card}\,(C) = M \;\middle|\; \text{il existe un code } C \subset \mathbb{F}_q^n \text{ de distance } d = d(C) \right\},$$

$$\alpha_q(\delta) := \limsup_{n\to\infty} n^{-1} \log_q A_q(n,d).$$

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 particuler :

- 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 δ dans l'intervale $]0, (q-1)/q[$ il existe une famille $\{C_i\}$ des $[n_i, k_i, d_i]_q$-codes, telle que

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

et

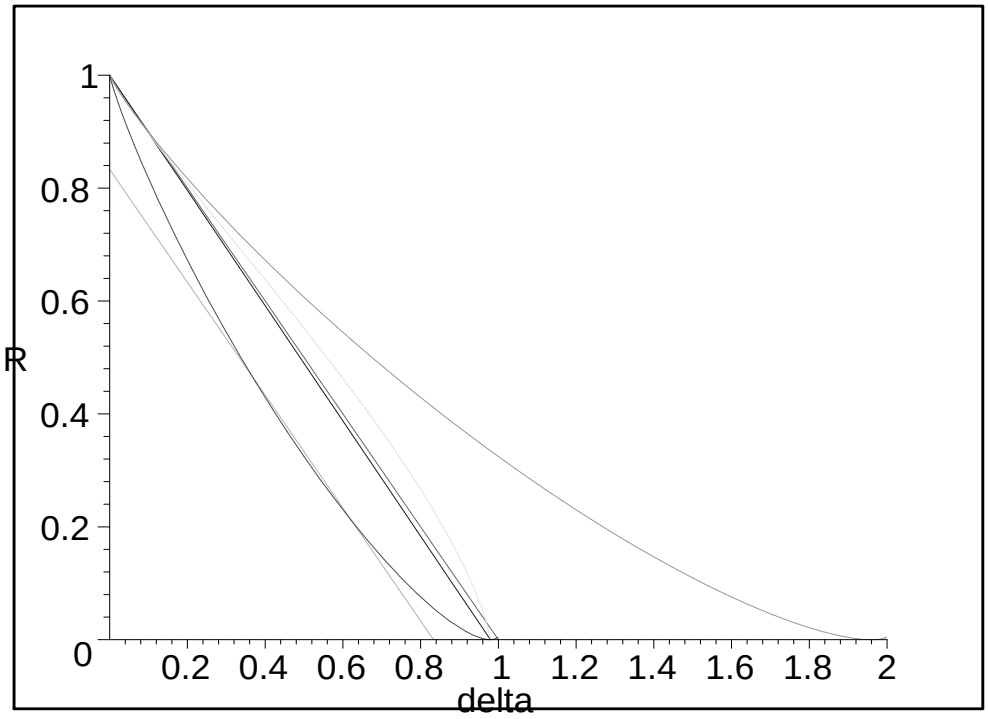
$$\alpha_q(\delta) \geq R \geq 1 - H_q(\delta),$$

c'est à direon 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*

$$d \leq \frac{nq^{k-1}(q-1)}{q^k-1} = n\frac{q-1}{q}\left(1+\frac{1}{q^k-1}\right). \qquad (*)$$

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 $M = \operatorname{Card} C = q^k$, alors il est commode d'écrire l'inégalité (*) sous la forme : si $\frac{d}{n} \neq \frac{q-1}{q}$, alors

$$\boxed{M \leq \left(\frac{d}{n}\frac{q}{q-1}-1\right)^{-1}+1}$$

.

THÉORÈME 6.2 (BORNE DE PLOTKIN ASYMPTOTIQUE). *Soit $\{C_i\}$ une famille des $[n_i,k_i,d_i]_q$-codes, on pose*

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

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

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

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

$$q^{k_i} \leq \left(\frac{d_i}{n_i}\frac{q}{q-1} - 1\right)^{-1} + 1 \to \frac{1}{\delta\frac{q}{q-1} - 1} + 1 > 0 (\text{une constante indépendent de } n), \tag{6.1}$$

puisque $\delta > \frac{q-1}{q}$.

Donc k_i sont uniformement bornés, cela signifie que

$$R = \limsup_{i\to\infty} \frac{k_i}{n_i} = 0.$$

2) DEUXIÈME CAS : $0 < \delta < \frac{q-1}{q}$. Dans ce cas on peut supposer : pour n_i assez grand

$$\frac{d_i}{n_i} = \delta_i \to \delta < \frac{q-1}{q} \Rightarrow \frac{d_i}{n_i} - \frac{1}{n_i} < \frac{q-1}{q} \Rightarrow d_i - 1 < n_i \cdot \frac{q-1}{q}.$$

On pose

$$n_i' = \left[\frac{q(d_i-1)}{q-1}\right] \leq \frac{q(d_i-1)}{q-1} \left(\Rightarrow n_i' \frac{q-1}{q} \leq d_i - 1\right).$$

Ensuite, $d_i = \delta_i n_i \to \infty$ donc

$$0 < n_i' < n_i$$

puisque

$$n_i' = \left[\frac{q(d_i-1)}{q-1}\right] \geq (d_i-1)\cdot\frac{q}{q-1} - 1 > 0,$$
$$n_i' = \left[\frac{q(d_i-1)}{q-1}\right] \leq (d_i-1)\cdot\frac{q}{q-1} < n_i$$

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

$$C_{n'} = \left\{ c' - c'_0 \middle| \ c' \in C' \right\} = \left\{ c = (c_1, \cdots, c_{n'}, 0, \cdots, 0) \middle| \ c' \in C' \right\} \subset C,$$

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 :

$$M \cdot q^{n'-n} \leq M' < \left(\tfrac{d'}{n'} \tfrac{q}{q-1} - 1\right)^{-1} + 1 \tag{6.2}$$

$$= \frac{n'(q-1)}{d'q - n'(q-1)} + 1 \leq \frac{n'(q-1)}{dq - n'(q-1)} + 1 \leq \frac{n'(q-1)}{q} + 1 \leq d \leq n$$

puisque $d \leq n$, $d' \geq d$ et

$$dq - n'(q-1) = dq - \left[\frac{q(d-1)}{q-1}\right](q-1) \geq dq - q(d-1) = q$$

(ne pas confondre la notation d' avec la distance construite des codes BCH !).

Pour conclure, on prend $\log_q$ de (6.2) : $M = q^k = \mathrm{Card}\,(C)$, 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 $n \to \infty$:

$$k \leq n - n' + \log_q n \leq n - \frac{q(d-1)}{q-1} + \log_q n \Rightarrow \boxed{\frac{k}{n} \leq 1 - \frac{q}{q-1} \cdot \frac{d}{n} + \frac{\log_q n + 1}{n}}$$

$$\Rightarrow \boxed{R \leq 1 - \frac{q}{q-1}\delta}$$

3) Troisième cas (en exercice) : $\delta = \frac{q-1}{q}$.

Pour montrer que $R = 0$ on raisonne par l'absurde. On suppose que

$$R = \limsup_{i \to \infty} \frac{k_i}{n_i} > 0,$$

et on choisit une sous-suite des codes C_i avec $k_i/n_i \geq R_0 > 0$. On pose

$$n'_i = \left[\frac{q(d_i - 1)}{q-1}\right] \leq \frac{q(d_i-1)}{q-1}.$$

Ensuite, $d_i = \delta_i n_i \to \infty$ donc

$$0 < n'_i < n_i$$

puisque

$$n'_i = \left[\frac{q(d_i-1)}{q-1}\right] \geq (d_i - 1) \cdot \frac{q}{q-1} - 1 > 0,$$

et pour voir que $n_i' < n_i$ on remarque que par Théorème 6.1

$$n_i' = \left[\frac{q(d_i-1)}{q-1}\right] \leq (d_i-1)\cdot\frac{q}{q-1} \leq \left(\frac{q-1}{q}\cdot n_i\cdot\left(1+\frac{1}{q^{k_i}-1}\right)-1\right)\cdot\frac{q}{q-1}$$
$$n_i + \frac{n_i}{q^{k_i}-1} - \frac{q}{q-1} < n_i, \quad (\frac{n_i}{q^{k_i}-1}\to 0)$$

puisque

$$d_i \leq \frac{q-1}{q}\cdot\left(1+\frac{1}{q^{k_i}-1}\right), \text{ et } q^{k_i} > q^{R_0 n_i} \Rightarrow \frac{n_i}{q^{k_i}-1}\to 0.$$

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

$$C_{n'} = \left\{c' - c_0' \,\middle|\, c' \in C'\right\} = \left\{c = (c_1,\cdots,c_{n'},0,\cdots,0) \,\middle|\, c' \in C'\right\} \subset C,$$

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 :

$$M\cdot q^{n'-n} \leq M' < \left(\tfrac{d'}{n'}\tfrac{q}{q-1}-1\right)^{-1}+1 \tag{6.3}$$
$$= \frac{n'(q-1)}{d'q-n'(q-1)}+1 \leq \frac{n'(q-1)}{dq-n'(q-1)} \leq \frac{n'(q-1)}{q} \leq n$$

puisque $d' \geq d$ et

$$dq - n'(q-1) = dq - \left[\frac{q(d-1)}{q-1}\right](q-1) \geq dq - q(d-1) = q$$

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 $n\to\infty$:

$$k \leq n - n' + \log_q n \leq n - \frac{q(d-1)}{q-1} + \log_q n \Rightarrow \frac{k}{n} \leq 1 - \frac{q}{q-1}\cdot\frac{d}{n} + \frac{\log_q n + 1}{n}$$

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 légérement 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

$$q^n > V_q(n, d-1)M, \quad V_q(n, d-1) = \sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i M.$$

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

$$q^n \leq V_q(n, d-1)M, \quad V_q(n, d-1) = \sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i M.$$

(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-nécessarement 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

$$M = \left[\frac{q^n}{V_q(n,d-1)}\right] - 1 \Rightarrow M < \frac{q^n}{V_q(n,d-1)} \leq M+2.$$

Borne de Gilbert-Varshamov asymptotique (voir Théorème 6.3).

Théorème 6.4 (Borne de Gilbert-Varshamov asymptotique) *Pour tout δ dans l'intervale $]0,(q-1)/q[$ il existe une famille $\{C_i\}$ des $[n_i,k_i,d_i]_q$-codes, telle que*

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

et

$$\alpha_q(\delta) \geq R \geq 1 - H_q(\delta),$$

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

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

Preuve : Soit $n_i \to \infty$ une suite arbitraire. On pose $d_i = [\delta n_i]$, et

$$k_i = \left[n_i - \log_q V_q(n_i - 1, d_i - 2)\right] - 1$$

un nombre vérifiant l'inégalité du Théorème 6.3 :

$$q^{n_i-k_i} > \sum_{i=0}^{d_i-2} \binom{n_i-1}{i}(q-1)^i = V_q(n_i-1,d_i-2)$$

puisque

$$-\frac{k_i}{n_i} \leq \frac{\log_q V_q(n_i-1,d_i-2)}{n_i} - 1 < \frac{1-k_i}{n_i}, \quad \frac{k_i-1}{n_i} < 1 - \frac{\log_q V_q(n_i-1,d_i-2)}{n_i} \leq \frac{k_i}{n_i}$$

Selon Théorème 6.3, il existe un $[n_i,k_i,d_i]_q$-code $\{C_i\}$. Il reste à passer à la limite $n_i \to \infty$:

$$\frac{\log_q V_q(n_i-1,d_i-2)}{n_i} = \frac{\log_q V_q(n_i-1,[\delta\cdot n_i]-2)}{n_i} \to H_q(\delta), \quad \frac{k_i}{n_i} \to R \Rightarrow R \geq 1 - H_q(\delta).$$

Rappelons le corollaire 2.10 : Lorsque $n\to\infty$, $\frac{r}{n} \to \delta$, on a

$$H_q(\delta) = \lim_{n\to\infty} \frac{\log_q V_q(n,[\delta n])}{n} = \lim_{n\to\infty} \log_q\left(\frac{V_q(n,[\delta n])}{\mathrm{Card}\,(F^n)}\right).$$

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*

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

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*

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

1)

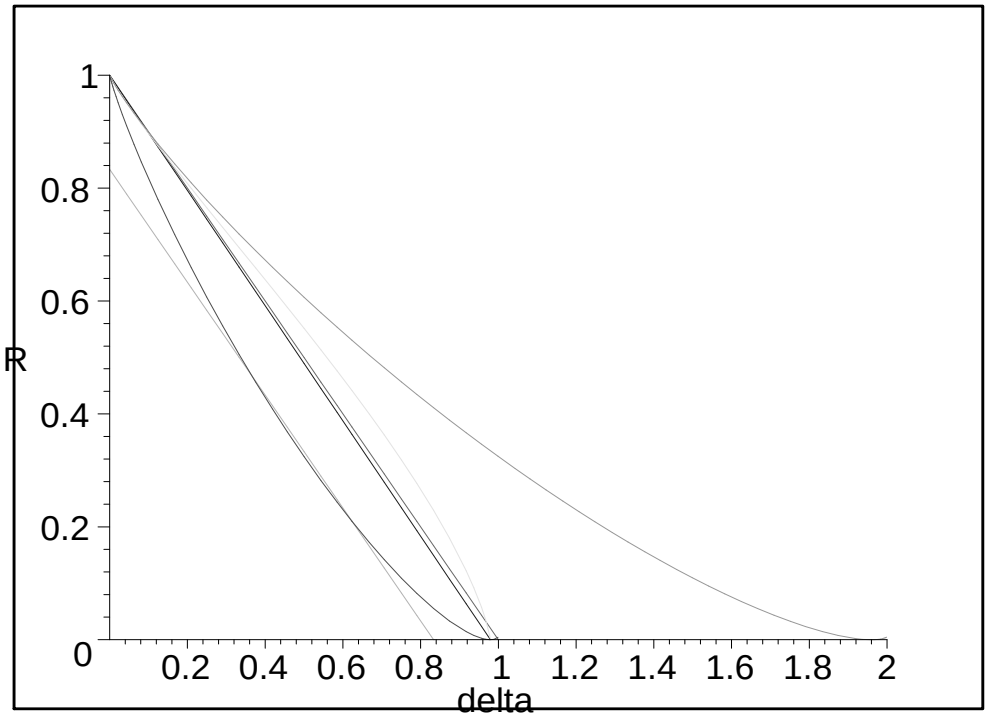
$$\alpha_q(\delta) \geq R \geq R_{GA} = 1 - \delta - \frac{1}{\sqrt{q}-1}. \tag{6.4}$$

2) Si q *est un carré, la borne (7.11) est atteinte*

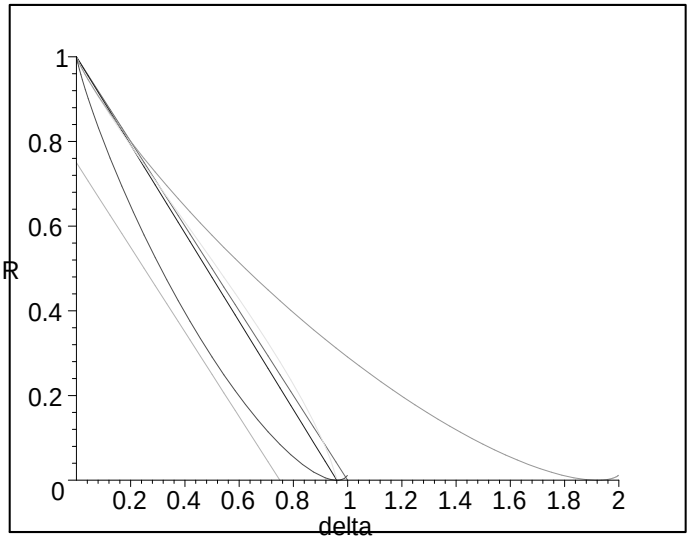
$$\limsup_{i\to\infty} \frac{d_i}{n_i} = \frac{1}{\sqrt{q}-1}.$$

REMARQUE 6.8 *Si* q *est un carré, alors la borne inférieure (7.11) est meilleure que la borne de Gilbert-Varshamov asymptotique (voir Théorème 6.4, et la représentation graphique des bornes ci-dessou).*

REPRÉSENTATION GRAPHIQUE DES BORNES : 1) Pour $q = 49$:


1
0.8
0.6
ℝ
0.4
0.2
0
0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
delta

2) Pour $q = 25$:


1
0.8
0.6
ℝ
0.4
0.2
0
0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
delta

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

$$R = \lim_{i\to\infty} \frac{k_i}{n_i}, \delta = \lim_{i\to\infty} \frac{d_i}{n_i}.$$

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*

$$R = \lim_{i\to\infty} \frac{k_i}{n_i} > 0, \delta = \lim_{i\to\infty} \frac{d_i}{n_i} \in]0, (q-1)/q[,$$

et

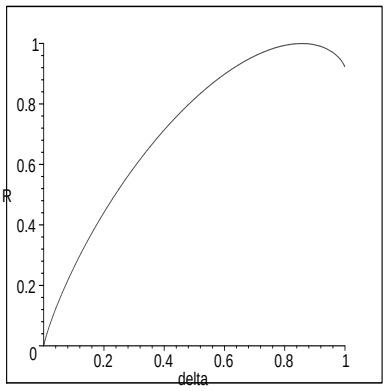
$$\boxed{R \geq 1 - H_q(\delta)}$$

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

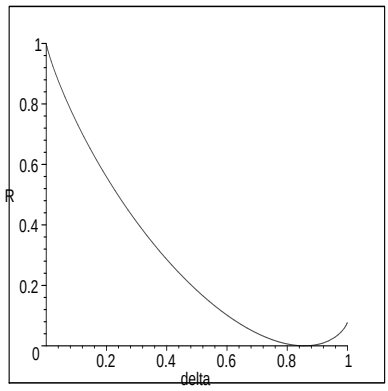
$$H_q(0) = 0,$$
$$H_q(\delta) = \delta \log_q(q-1) - \delta \log_q(\delta) - (1-\delta)\log_q(1-\delta) \text{ pour } 0 \leq \delta \leq (q-1)/q.$$

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*

$$q^{n-k} > V_q(n-1, d-2) = \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i. \qquad (*)$$

(Rappelons qu'on a montré l'existence d'un tel code en construisant sa matrice de contrôle $H \in \mathrm{M}_{n-k,n}(\mathbb{F}_q)$ 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

THÈORÈME 6.4 (Borne de Gilbert-Varshamov asymptotique) Pour tout δ dans l'intervale $]0,(q-1)/q[$ il existe une famille $\{C_i\}$ des $[n_i,k_i,d_i]_q$-codes, telle que

$$R=\limsup_{i\to\infty}\frac{k_i}{n_i},\delta=\limsup_{i\to\infty}\frac{d_i}{n_i},\text{ et }\boxed{R=1-H_q(\delta)},$$

PREUVE : Soit $n_i\to\infty$ une suite arbitraire. On pose $d_i=[\delta n_i]$, et

$$k_i=\left[n_i-\log_q V_q(n_i-1,d_i-2)\right]-1$$

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 $n_i\to\infty$ 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 $n\to\infty$, $\frac{r}{n}\to\delta$, on a

$$\lim_{n\to\infty}\frac{\log_q V_q(n,r)}{n}=H_q(\delta)\Rightarrow\frac{\log_q V_q(n_i-1,d_i-2)}{n_i}=\frac{\log_q V_q(n_i-1,[\delta\cdot n_i]-2)}{n_i}\to H_q(\delta),$$

$$\frac{k_i}{n_i}\to R\Rightarrow 1-R\leq H_q(\delta)\leq 1-R\Rightarrow\boxed{R=1-H_q(\delta)}.$$

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*

$$\mathcal{P}=\{P_1,\cdots,P_n\}\subset V$$

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

$$E_{\mathcal{P}} : V^* \to \mathbb{F}_q^n, \quad E_{\mathcal{P}}(l) = (l(P_1), \ldots, l(P_n))$$

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 $c = (l_0(P_1), \cdots, l_0(P_n))$ un mot de code C avec $l_0 \in V^*, l_0 \neq 0$, alors

$$\text{Card}\,\{i \in \{1,2,\cdots,n\} \mid l_0(P_i) \neq 0\} = n - \text{Card}\,(\{\mathcal{P} \cap H_0 \mid H_0 \text{ hyperplan } l_0 = 0\}$$

donc

$$d = \min_l \text{Card}\,\{i \in \{1,2,\cdots,n\} \mid l(P_i) \neq 0\}.$$

REMARQUE Pratiquement une matrice-génératrice de C est donnée par les coordonnées des points $P_1, \cdots, P_n$ dans une base de V : si $l_1, \cdots, l_k \in V^*$ les fonctions des coordonnées (la base duale d'une base de V), alors

$$G = \begin{pmatrix} l_1(P_1) & \ldots & \ldots & l_1(P_n) \\ \ldots & \ldots & \ldots & \ldots \\ l_k(P_1) & \ldots & \ldots & l_k(P_n) \end{pmatrix} \text{ (ne pas confondre avec une matrice de contrôle !)}$$

THÉORÈME 7.2 *Soit $Syst_q(n,k,d)$ l'ensemble des systèmes affines ci-dessus. Le groupe $\text{GL}_{\mathbb{F}_q}(V)$ agit sur $Syst_q(n,k,d)$, et il y a une bijection*

$$\varphi : Syst_q(n,k,d)/\text{GL}_{\mathbb{F}_q}(V) \xrightarrow{\sim} Codes([n,k,d]_q)$$

PREUVE. Pour $g \in \text{GL}_{\mathbb{F}_q}(V)$ on pose

$$g \cdot \mathcal{P} = \{g \cdot P_1, \cdots, g \cdot P_n\} \subset V$$

alors

$$P_i \in H \Longleftrightarrow g \cdot P_i \in g \cdot H \text{ un autre hyperplan .}$$

D'autre part,

$$\text{Ker}(E_{\mathcal{P}}) = \{l \in V^* \mid (l(P_1), \ldots, l(P_n))\} = \{0\} \Rightarrow E_{\mathcal{P}} \text{ est injective}$$

$$E_{g\cdot\mathcal{P}}(l) = (l(g \cdot P_1), \ldots, l(g \cdot P_n)) = (g^*l(P_1), \ldots, g^*l(P_n))E_{\mathcal{P}}(g^*l)$$

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 φ :

$$\varphi^{-1} : C \mapsto \varphi^{-1}(C) = \{P_1, \cdots, P_n\} \subset V = C^*, \quad P_i(c) = c_i, \ i^{\text{ème}} \text{ coordonnée de } c = (c_1, \cdots, c_n) \in C$$

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 $(x_0, \cdots, x_n)$, $x_i \in F$ pour $0 \leq i \leq n$; sous la relation $(x_0, \cdots, x_n) \sim (\lambda_0 x_0, \cdots, \lambda_n x_n)$ pour tout $\lambda \in F^*$. Une classe d'équivalence x est un point de $\mathbb{P}^n$.

DÉFINITION 7.3 *Un système projectif*

$$\mathcal{P} = \{P_1, \cdots, P_n\} \subset \mathbb{P}^{k-1}_{\mathbb{F}_q}$$

est un sous-ensemble dans $\mathbb{P}^{k-1}_{\mathbb{F}_q} = \mathbb{P}(V)$ *tel que* $\mathcal{P}$ *n'est pas contenu dans aucun hyperplan projectif* $H \subset \mathbb{P}^{k-1}_{\mathbb{F}_q}$.

Soit

$$\mathcal{P}^\circ = \{P_1^\circ, \cdots, P_n^\circ\} \subset V$$

un système affine dans V representant $\mathcal{P}$. Tout autre système

$$\mathcal{P}^\circ_\lambda = \{\lambda_1 P_1^\circ, \cdots, \lambda_n P_n^\circ\} \subset V$$

défini un code $C_\lambda = \mathrm{Im}(E_{\mathcal{P}^\circ_\lambda}) = \varphi(\mathcal{P}^\circ_\lambda)$ équivalent à $C = \mathrm{Im}(E_{\mathcal{P}^\circ}) = \varphi(\mathcal{P}^\circ)$:

$$C = \varphi(\mathcal{P}^\circ) = E_{\mathcal{P}^\circ}(V^*) = \{(l(P_1^\circ), \ldots, l(P_n^\circ)) \mid l \in V^*\}$$
$$C_\lambda = \varphi(\mathcal{P}^\circ_\lambda) = E_{\mathcal{P}^\circ}(V^*) = \{(\lambda_1 l(P_1^\circ), \ldots, \lambda_n l(P_n)^\circ) \mid l \in V^* \ , \lambda_i \in \mathbb{F}_q^*\}.$$

C'est-à-dire, $C_\lambda = \sigma(C)$ pour la bijection linéaire diagonale donnée par la matrice $\sigma = \mathrm{diag}\{\lambda_1, \cdots, \lambda_n\}$.

THÉORÈME 7.4 *Soit* $\overline{Syst}_q(n,k,d)$ *l'ensemble des systèmes projectifs ci-dessus. Le groupe* $\mathrm{PGL}_{\mathbb{F}_q}(V)$ *agit sur* $\overline{Syst}_q(n,k,d)$*, et il y a une bijection*

$$\overline{\varphi} : \overline{Syst}_q(n,k,d)/\mathrm{PGL}_{\mathbb{F}_q}(V) \xrightarrow{\sim} \{Codes([n,k,d]_q)\}/(\mathrm{mod}\sim)$$

où $\sim$ *désigne l'équivalence linéaire diagolale des* $[n,k,d]_q$ *codes.*

PREUVE. : en exercice.

7.2. Codes géométriques

On défini les codes géométriques à partir des sous-ensembles de

$$X = \mathcal{P} = \{P_1, \cdots, P_n\} \subset \mathbb{P}^{k-1}_{\mathbb{F}_q}, \text{ et } \{P_1, \cdots, P_n\} \subset \mathbb{F}_q^k.$$

Une matrice génératrice du code $C = \overline{\varphi}(\mathcal{P})$ est donnée par des coordonnées des points P_i :

$$P = (a_{ij}), \ P_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \cdots \\ a_{kj} \end{pmatrix}, j = 1, \cdots, n.$$

On pose

$$d = d(C) = d(X) = n - \max_H\{\mathrm{Card}\,(X \cap H) \mid H \text{ hyperplan } \} > 0$$

REMARQUE 7.5 *Le nombre* $\mathrm{Card}\,(X \cap H)$ *est majoré par le "degré" de* X*. Pour une courbe planaire* $X = X(\mathbb{F}_q) \subset \mathbb{P}^2_{\mathbb{F}_q}$ *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 $L_m = \mathbb{F}_q[T_1, \cdots, T_m]_{d^\circ \leq 1}$ le $\mathbb{F}_q$-espace vectoriel de tous les polynômes de degré ≤ 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

$$\mathcal{P} = \{P_1, \cdots, P_n\} \subset \mathbb{F}_q^m$$

tel que

$$\forall l \in L_m (\forall j = 1, \cdots, n, l(P_j) = 0 \Rightarrow l \equiv 0).$$

DÉFINITION 7.6

$$RM_q(1,n,m) = \mathrm{Im}Ev, \quad Ev : L_m \to \mathbb{F}_q^n, \quad f \mapsto ((f(P_1), \cdots, f(P_n))$$

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 ≤ 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 $\mathbb{F}_q$-space 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 $\mathbb{F}_{q^m}$, $L = \{\gamma_0, \gamma_1, \cdots, \gamma_{n-1}\} \subset \mathbb{F}_{q^m}$ avec $g(\gamma_i) \neq 0$ pour tout $i = 0, 1, 2, \cdots, n-1$.

DÉFINITION 7.7 *Le code de Goppa rationnel* $\Gamma(L,g)$ *est*

$$\Gamma(L,g) = \left\{ w = (w_0, w_1, \cdots, w_{n-1}) \subset \mathbb{F}_q^n \Big| \sum_{i=0}^{n-1} \frac{w_i}{x-\gamma_i} \equiv 0 \bmod g(x) \right\} \tag{7.1}$$

(c'est-à-dire, la fraction à droite est de la forme d'une fraction irréductible $\frac{a(x)}{b(x)}$ *avec* $g(x)|a(x)$.

Soient $v = (v_0, v_1, \cdots, v_{n-1}) = w + e$, un mot recu, $w = (w_0, w_1, \cdots, w_{n-1}) \in \Gamma(L,g)$ un mot de code, $e = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{F}_q^n$ le mot d'erreur, alors on considère les fractions suivantes

$$v(x) = \sum_{i=0}^{n-1} \frac{v_i}{x-\gamma_i}, \quad w(x) = \sum_{i=0}^{n-1} \frac{w_i}{x-\gamma_i}, \quad e(x) = \sum_{i=0}^{n-1} \frac{e_i}{x-\gamma_i}.$$

Pour écrire une matrice de contrôle du code $\Gamma(L,g)$ on utilise pour tout $i = 0, \cdots, n-1$ un unique polynôme $f_i(x) \bmod g(x)$ dans $\mathbb{F}_{q^m}[x]/(g)$ tel que

$$f_i(x)(x-\gamma_i) \equiv 1 \bmod g(x) \Longleftrightarrow f_i(x) := -\frac{1}{g(\gamma_i)} \left[\frac{g(x)-g(\gamma_i)}{x-\gamma_i}\right]. \tag{7.2}$$

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

$$\frac{g(x) - g(y)}{x - y} = \sum_{k+j \leq t-1} g_{k+j+1} y^j x^k \Rightarrow \sum_{i=0}^{n-1} c_i h_i \sum_{k+j \leq t-1} g_{k+j+1} \cdot (\gamma_i)^j x^k \equiv 0 \bmod g$$

(puisque

$$\frac{x^i - y^i}{x - y} = \sum_{k+j=i-1} y^j x^k).$$

Ceci implique

$$f_i(x) \equiv \sum_{k+j \leq t-1} g_{k+j+1} y^j x^k \tag{7.3}$$

Pour pouvoir travailler avec les polynômes, on fait correspondre (de façon unique) au mot $v(x)$ un polynôme $f_v(x)$ modulo g

$$v(x) \mapsto f_v(x) = \sum_{i=0}^{n-1} v_i f_i(x),$$

alors

$$w(x) \mapsto f_w(x) = \sum_{i=0}^{n-1} w_i f_i(x) \equiv 0 \bmod g.$$

En ettet,

$$f_i(x)(x - \gamma_i) \equiv 1 \bmod g(x) \Rightarrow f_i(x)(x - \gamma_i) = 1 + u_i(x) g(x), \quad \frac{1}{x - \gamma_i} = \frac{f_i(x)}{1 + u_i(x) g(x)}$$

donc

$$\sum_{i=0}^{n-1} \frac{w_i}{x - \gamma_i} = \sum_{i=0}^{n-1} \frac{w_i f_i(x)}{1 + u_i(x) g(x)} = \frac{\sum_{i=0}^{n-1} w_i f_i(x) \prod_{j \neq i} (1 + u_j(x) g(x))}{\prod_{j=0}^{n-1} (1 + u_j(x) g(x))}$$

Le dénominateur est congru à 1 mod g, et le numérateur est congru à $\sum_{i=0}^{n-1} w_i f_i(x)$ mod g. D'autre part, le numérateur est congru à 0 mod g par Définition 7.9.

Si $v = (v_0, v_1, \cdots, v_{n-1}) = w + e$, un mot recu, $w = (w_0, w_1, \cdots, w_{n-1}) \in \Gamma(L, g)$, $e = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{F}_q^n$, alors

$$f_v(x) = S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \equiv \sum_{i=0}^{n-1} v_i f_i(x) \bmod g$$

puisque $\sum_{i=0}^{n-1} w_i f_i(x) \equiv 0 \bmod g$. On obtient maintenant une matrice de contrôle de la forme explicite (7.3) :

$$\sum_{i=0}^{n-1} w_i f_i(x) \equiv 0 \bmod g \Rightarrow \sum_{i=0}^{n-1} w_i h_i \sum_{k+j \leq t-1} g_{k+j+1} \cdot (\gamma_i)^j x^k \equiv 0 \bmod g$$

ceci dit, pour tous les $k = 0, \cdots, t-1$ le coefficient de x^k à gauche est nul, donc une matrice de contrôle du code $\Gamma(L, g)$ est

$$\begin{pmatrix} h_0 g_t & h_1 g_t & \cdots & h_{n-1} g_t \\ h_0(g_{t-1} + g_t\gamma_0) & h_1(g_{t-1} + g_t\gamma_1) & \cdots & h_{n-1}(g_{t-1} + g_t\gamma_{n-1}) \\ \cdots & \cdots & \cdots & \cdots \\ h_0(g_1 + g_2\gamma_0 + \cdots + g_t\gamma_0^{t-1}) & h_1(g_1 + g_2\gamma_1 + \cdots + g_t\gamma_1^{t-1}) & \cdots & h_{n-1}(g_1 + g_2\gamma_{n-1} + \cdots + g_t\gamma_{n-1}^{t-1}) \end{pmatrix}$$

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 utilse comme syndrôme 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 syndrôme* $\bmod g$

$$f_e(x) \equiv S(x) \equiv f_v(x) \equiv \sum_{i=0}^{n-1} e_i f_i(x) \bmod g.$$

THÉORÈME 7.8 *Le code* $\Gamma(L, g)$ *a :*
(a) *la dimension* $k(\Gamma(L, g)) \geq n - mt$,
(b) *la distance* $d(\Gamma(L, g)) \geq t + 1$, $f = \deg(g)$

PREUVE. L'assértion (a) est immédiate de la taille de la matrice G vue comme une $\mathrm{Mat}_{mt,n}(\mathbb{F}_q)$-matrice après un choix d'une base de $\mathbb{F}_{q^m}$ sur $\mathbb{F}_q$.

L'assértion (b) est impliquée par l'unicité de décomposition d'une fraction rationnelle en éléments simples : pour avoir un élément non- nul

$$\sum_{i=0}^{n-1} \frac{c_i}{x - \gamma_i} = \frac{a(x)}{b(x)} \tag{7.4}$$

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 avoit 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

THÉORÈME 7.9 *Pour tout* $\delta \in]0, (q-1)/q[$ *on construit une famille* $\{C_i\}$ *des* $[n_i, k_i, d_i]_q$*-codes, telle que*

$$R = \lim_{i\to\infty} \frac{k_i}{n_i}, \delta = \lim_{i\to\infty} \frac{d_i}{n_i},$$

et

$$R = 1 - H_q(\delta)$$

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

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

Rapellons

DÉFINITION 6.9. *Une bonne famille $\{C_i\}$ des $[n_i, k_i, d_i]_q$-codes est dite excellente s'il existe*

$$R = \lim_{i\to\infty} \frac{k_i}{n_i}, \delta = \lim_{i\to\infty} \frac{d_i}{n_i},$$

et

$$R \geq 1 - H_q(\delta)$$

REMARQUE 7.10 *Théorème 7.9 donne une construction explicite d'une famille excellente $\{C_i\}$ des $[n_i, k_i, d_i]_q$-codes de Goppa rationnelles.*

PREUVE du Théorème 7.9. On choisit $n_i = q^{m_i}$, $m = m_i$, $L = \mathcal{P} = \mathbb{F}_{q^m}$, $d = d_i = [\delta n_i]$, 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 $c = (c_0, c_1, \cdots, c_{n-1})$ 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 $\left[\frac{j-1}{t}\right]$. Il nous faut donc exclure de considération au plus

$$\sum_{j=1}^{d-1} \binom{n}{j} (q-1)^j \left[\frac{j-1}{t}\right] \leq \frac{d}{t} V_q(n, d)$$

puisque le nombre des numérateurs non-nuls de la fraction (7.4) est majoré par le nombre de toutes le combinaisons linéaires non-triviales (sur $\mathbb{F}_q$) de d élements parmi n éléments.

D'autre part on sait que (voir Corollaire 2.10) que

$$\lim_{n\to\infty} \frac{\log_q V_q(n, [\delta n])}{n} = H_q(\delta),$$

et que le nombre total des polynômes irréductibles de degré t sur $\mathbb{F}_{q^m}[x]$ est (voir Théorème A17) :

$$\frac{1}{t} q^{mt}(1 + o(1)) \text{ lorsque } m \to \infty.$$

On obtient donc une condition suffisante pour l'existance d'un polynôme voulu g de degré t

$$\frac{d}{t} V_q(n, d) < \frac{1}{t} q^{mt}(1 + o(1)) \Longleftrightarrow d \cdot V_q(n, d) < q^{mt}(1 + o(1)) \ (m \to \infty), \tag{7.5}$$

que signifie :

$$H_q(\delta) < \frac{mt}{n} + o(1), \text{ lorsque } m \to \infty.$$

Plus précisement, pour satisfaire les inégalités (7.5) et $t < d = d_i$, on prend $\log_q$ de (7.5) :

$$\log_q(d \cdot V_q(n, d)) = \log_q([\delta n]) + \log_q(V_q(n, [\delta n])) = n(H_q(\delta) + \log_q(n)) \leq mt(1 + o(1)) \ (m \to \infty),$$

Pour choisir g et t on pose

$$t = \left[\frac{n}{m} H_q(\delta)(1 + o(1))\right] - 1 \tag{7.6}$$

pour satisfaire (7.5), et pour avoir $t < d$, 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, Théorème 7.8 résulte que, avec $n = n_i$, $m = m_i$, $k = k_i$, $d = d_i$ ce-dessus, et $t = t_i$ choisit comme dans l'égalité (7.6),

$$R_i = \frac{k_i}{n_i} \geq \frac{n_i - m_i t_i}{n_i} \geq 1 - H_q(\delta)(1+o(1)) \to 1 - H_q(\delta) \Rightarrow R \geq 1 - H_q(\delta)$$

donc la famille construite est *excellente*.

7.4. Décodage des codes de Goppa rationnels

On utilse comme syndrôme la fraction rationnelle

$$\sum_{i=0}^{n-1} \frac{e_i}{x - \gamma_i}$$

(le vecteur d'erreurs), représenté comme un polynôme $\bmod g$

$$S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \bmod g$$

(le polynôme de syndrôme). Remarquer que si $v = (v_0, v_1, \cdots, v_{n-1}) = w + e$, un mot recu, $w = (w_0, w_1, \cdots, w_{n-1}) \in \Gamma(L, g)$, $e = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{F}_q^n$, 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

$$\sum_{i=0}^{n-1} e_i f_i(x) \equiv 0$$

Ensuite, on utise une version de l'algorithme de Berlekamp-Massey, voir 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 la définition (8.1) comme une solution de la congruence :

$$u(x) \equiv s(x)S(x) (\mathrm{mod} g(x)) \tag{7.7}$$

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'au $\mathrm{pgcd}(g(x), S(x))$ est atteint, 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 $u(x)$.

7.5. Codes provenants des courbes algébriques : $\mathcal{P}$-construction

Tous les autres exemples considérés dans le reste de la Section proviennent d'une courbe algébrique X sur $\mathbb{F}_q$ muni d'un plongement projectif

$$\varphi : X \hookrightarrow \mathbb{P}^N_{\mathbb{F}_q}.$$

On utilise des sous-espaces vectoriels $\mathcal{L}(D) \subset \mathbb{F}_q(X)$ de dimenion finie (voir définition 7.21) associés à un *diviseur* D (voir Section 7.7.)

Il existe plusieures bonnes constructions des codes (L-construction, Ω-construction, $\mathcal{P}$-construction, voir [Ste], pp.243-248). Dans ce qui suit on donnera seulement quelques exemples.

7.5.1. $\mathcal{P}$-construction

Dans ce cas $\mathcal{P} = X$, $n = \mathrm{Card}\,(X(\mathbb{F}_q))$,

$$k = \dim l(D) = \dim_{\mathbb{F}_q} \mathcal{L}(D),$$

où $\mathcal{L}(D) \subset \mathbb{F}_q(X)$ est un sous-espace de dimension finie dans le corps

$$\mathbb{F}_q(X) = \left\{ \frac{u(T_0, \cdots, T_N)}{v(T_0, \cdots, T_N)} \right\} / \mathrm{mod} \sim$$

des fonctions rationnelles sur X.

On utilise un plongement canonique

$$\varphi_D : X \hookrightarrow \mathbb{P}^{l(D)-1}_{\mathbb{F}_q}$$

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 7.27.

On pose $a = \deg D$ alors

$$d = n - \max_H \{\mathrm{Card}\,(X \cap H) \mid H \text{ hyperplan } \}$$

puisque $\deg D \leq \max_H \{\mathrm{Card}\,(X \cap H) \mid H \text{ hyperplan } \}$: selon la construction du plongement φ_D ([Ste], p.92) avec $N = l(D) - 1$, il existe un hyperplan $H \subset \mathbb{P}^{l(D)-1}_{\mathbb{F}_q}$ tel que $D = \varphi_D^{-1}(H)$, donc $\deg D \leq \max_H \{\mathrm{Card}\,(X \cap H) \mid H \text{ hyperplan } \}$ (il se peut que les coordonnées des points d'intersection appartiennent à une extenstion du corps de base $\mathbb{F}_q$).

7.5.2. Borne de la géométrie algébrique

On utilise le théorème 7.28 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 7.27. Ceci impliqie :

$$\begin{cases} k = l(D) \geq \deg(D) - g + 1, \text{ où } a = \deg(D) \\ d \geq n - a, (d := n - \max_H \{\text{Card}\,(X \cap H) \mid H \text{ hyperplan }\}) \end{cases} \tag{7.8}$$

donc l'addition directe dans 7.8 donne

$$\boxed{k + d \geq n - g + 1} \tag{7.9}$$

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 :

$$R + \delta \geq 1 - \frac{g}{n} + \frac{1}{n} \tag{7.10}$$

THÉORÈME 7.11 (IHARA-TSFASMAN-ZINK-DRINFELD-VLADUT, sans démonstration). *Soit $g = g(X)$ est le genre de X, $n = \text{Card}\, X(\mathbb{F}_q)$). Alors*

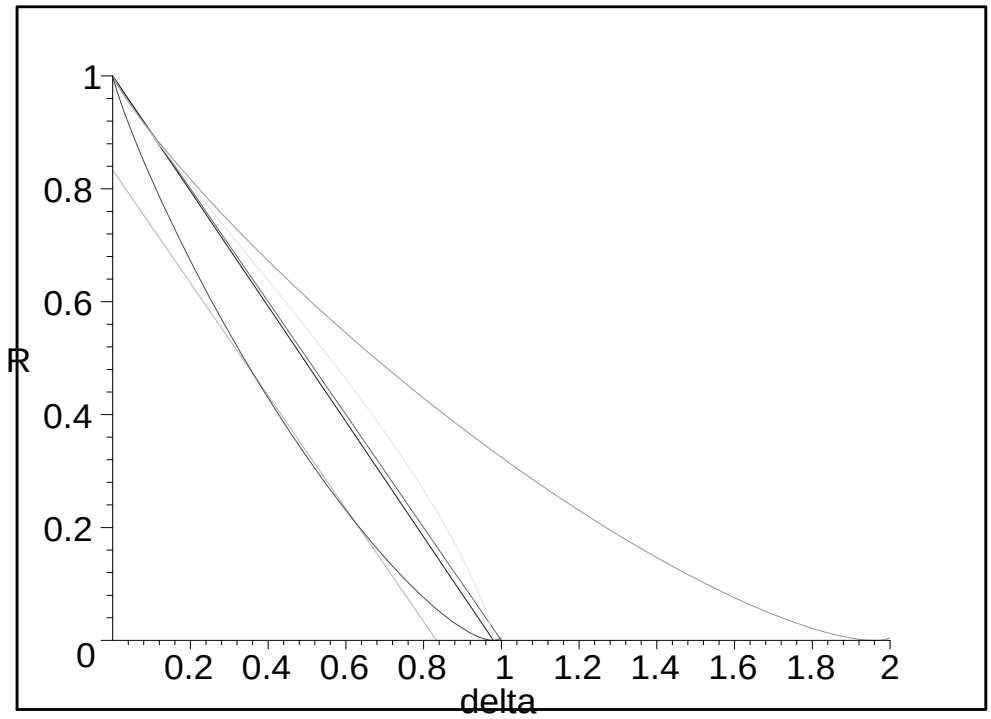
(a)

$$\liminf_X \frac{g}{n} \geq \frac{1}{\sqrt{q} - 1}. \tag{7.11}$$

(b) *Si q est un carré, la borne (7.11) est atteinte*

$$\liminf_X \frac{g}{n} = \frac{1}{\sqrt{q} - 1}.$$

Pour $q = 49$:



7.6. Espace projectif $\mathbb{P}^n$, variétés algébriques

Soit F un corps algébriquement clos et $n \geq 1$ un entier. On considère espace projectif de dimension n sur F et on note $\mathbb{P}^n_k$ ou simplement $\mathbb{P}^n$ l'ensemble des classes d'équivalence de $(n+1)$-uplets $(x_0, \cdots, x_n)$, $x_i \in F$ pour $0 \leq i \leq n$; sous la relation $(x_0, \cdots, x_n) \sim (\lambda_0 x_0, \cdots, \lambda_n x_n)$ pour tout $\lambda \in F^*$. Une classe d'équivalence x est un point de $\mathbb{P}^n$. Soit $F[T] = F[T_0, \cdots, T_n]$. On interprète les éléments de $F[T]$ comme des fonctions régulières de l'espace affine $\mathbb{A}^{n+1}$, et on interprète les éléments de

$$F\left(\frac{T_1}{T_0}, \frac{T_2}{T_0}, \cdots, \frac{T_n}{T_0}\right)$$

comme des fonctions rationnelles de $\mathbb{P}^n$

Définition 7.12 *: Une partie $X \subset \mathbb{P}^n$ est dit une variété algébrique projective si*

$$X = V(P) = \{x \in \mathbb{P}^n \mid \forall F \in P, F(x) = 0\}$$

où P est un idéal premier homogène de $F[T_0, \cdots, T_n]$, c'est à dire premier engendré par des polynômes homogènes. Tout sous-espace ouvert de X est une variété algébrique quasi-projective.

Définition 7.13 *Soit U un voisinage ouvert d'un point x d'une variété projective X. Une application $f : U \to F$ est une fonction régulière (ou lisse) au point x s'ils existent $F, G \in F[T_0, \cdots, T_n]$ des polynômes homogènes de même degré tels que $G(x) \in 0$ et $f = F/G$ dans un voisinage ouvert de x ; f est régulière sur U si elle est régulière en tout x de U.*

Définition 7.14 *Soit X une variété projective. L'idéal de X est l'ensemble*

$$I(X) = \{F \in F[T_0, \cdots, T_n] | F(x) = 0, \forall x \in X\}.$$

On définit également le corps des fonctions $F(X)$ de X comme le corps des fractions de la F-algèbre de type fini $F[T]/I(X)$.

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 :

$$(U, f) \sim (U', f') \Longleftrightarrow f = f' \text{ sur } U \cap U'$$

dont les classes d'équivalences forment un anneau.

Définition 7.15 *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*

$$\mathfrak{m}_x = \{ \text{ classes d'équivalence de } (U, f) \text{ avec } f(x) = 0\}$$

Remarque 7.16 *Tout idéal propre non nul de $\mathcal{O}_x$ est de la forme $\mathfrak{m}_x^n$ pour un entier $n > 0$ (voir [Ste], p.78).*

Définition 7.17 *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 7.18 *Une variété projective X est dite non-singulière au point x si $\mathcal{O}_x$ est un anneau local régulier. X est singulière au point x sinon. X est dite non-singulière ou lisse si elle est non singulière en tout point. L'ensemble des points non-singuliers de X est un ouvert non vide de X dense dans X.*

Définition 7.19 *: On appelle courbe projective lisse toute variété algébrique projective lisse de dimension 1. La dimension de X est le degré de transcendance de $F(X)$ sur F.*

7.7. Diviseurs

DÉFINITION 7.20 *Un diviseur sur une courbe projective lisse X est une combinaison linéaire*

$$D = \sum a_x x,$$

la somme étant sur tous les points de X, et les a_x entiers tous nuls sauf un ensemble fini appelé le support de D. Si les a_x sont tous positifs ou nuls, on note $D \geq 0$. Si les a_x sont de plus tous non nuls, D est dit positif, noté $D > 0$. On définit le degré de D : $\deg(D) = \sum a_x$.

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 $f \in m_x^{n+1}$. Si $h = f/g \in F(X)$, 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 $v_x(h-1)$. $(f) = \sum v_x(f) \cdot x$ est le diviseur appelé diviseur de f. Les diviseurs de la forme (f) sont appelés principaux. On définit alors la relation d'équivalence

$$D \sim D' \iff \text{il existe} f \text{ de } \mathfrak{m}_x \text{ telle que } D - D' = (f)$$

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 F-espace vectoriel*

$$\mathcal{L}(D) = \{f \in F(X)^* | (f) + D \geq 0\},$$

C'est à dire que si $D = -\sum a_x x + \sum a_{x'} x'$, avec les a_x et les $a_{x'}$ tous positifs ou nuls, $\mathcal{L}(D)$ est l'ensemble des éléments non nuls de $F(X)$ ayant des zéros d'ordre au moins a_x aux points x et des pôles d'ordre au plus $a_{x'}$ aux points x'. On note $l(D)$ la dimension sur F de $\mathcal{L}(D)$.

THÉORÈME 7.22 *On a*

$$l(D) = 0 \text{ si } \deg D < 0$$
$$l(D) \leq \deg D + 1 \text{ sinon.}$$

PREUVE : voir [Ste], p.83.

Soit ω 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 $F(X)$-module.

DÉFINITION 7.23 *Soit ω une forme différentielle rationnelle sur X courbe lisse. Au voisinage de x point de X, ω s'écrit $\omega = f dt$, avec $f = f_x$ une fonction rationnelle et $t = t_x$ un paramètre local. On définit le diviseur de ω 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 représentatant de W est appelé diviseur canonique noté K.

REMARQUE 7.24 *: On a $\mathcal{L}(K) \cong \Omega[X]$ (l'espace des formes différentielles régulières sur X) par l'isomorphisme de F-espaces vectoriels $f \to f\omega$*

7.8. Courbes sur les corps finis

DÉFINITION 7.25 *: Soit F' un sous-corps de F. Un point $x = (x_0, \cdots, x_n)$ de $\mathbb{P}^n_F$ est dit F'-rationnel si*

$$x_i \neq 0 \Longrightarrow x_j/x_i \in F' \text{ pour tout } 0 \leq j \leq n$$

DÉFINITION 7.26 *Soit X une courbe projective lisse sur F et σ l'automorphisme de Frobenius sur X. Un diviseur D est dit F-rationnel si*

$$\sigma(D) = D$$

7.9. Théorème de Riemann-Roch, genre

DÉFINITION 7.27 *Soit X une courbe projective lisse définie sur F. On appelle genre de X*

$$g = g(X) = l(K) = \dim_F \mathcal{L}(K)$$

THÉORÈME 7.28 (de Riemann-Roch). *Soit D un diviseur sur une courbe projective et lisse X de genre g et soit K un diviseur canonique. Alors*

$$l(D) - l(K - D) = \deg D - g + 1$$

PREUVE : Voir [Ste], p.91.

PROPOSITION 7.29 (Formule du genre de Plücker). *Soit X une courbe plane 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 7.30 (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 7.31 *: Soit X une courbe projective lisse définie sur $\mathbb{F}_q$. Soient $x_1, \cdots, x_n$ 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*

$$\begin{aligned} Ev : \mathcal{L}(D) &\to \mathbb{F}_{q^n} \\ f &\mapsto (f(x_1), \ldots, f(x_n)) \end{aligned}$$

Un tel code est appelé un code géométrique de Goppa.

Pour construire un code de Goppa sur 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 7.32 *Soit X une courbe projective lisse de genre g et $x_1, \cdots, x_n$ des points de X F_q-rationnels. Alors le code de Goppa $C(D_0, D)$ est de dimension k, de longueur n et d'écart d satisfaisant*

$$\begin{aligned} k = l(D) - l(D - D_0) = \quad & \deg D - g + 1 + l(K - D) - l(D_0 - D) \\ & d \geq n - \deg D \end{aligned}$$

COROLLAIRE 7.33 *: 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 $\{f_1, \cdots, f_k\}$ 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 $k = k_C$.

DÉFINITION 7.34 *: Soit X une courbe projective lisse définie sur $\mathbb{F}_q$. Soient $x_1, \cdots, x_n$ 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*

$$\begin{aligned} & Ev : \mathcal{L}(D) \to \mathbb{F}_{q^n} \\ & f \mapsto (f(x_1), \ldots, f(x_n)) \end{aligned}$$

Un tel code est appelé un code géométrique de Goppa.

Pour construire un code de Goppa sur 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 7.35 *Soit X une courbe projective lisse de genre g et $x_1, \cdots, x_n$ des points de X F_q-rationnels. Alors le code de Goppa $C(D_0, D)$ est de dimension k, de longueur n et de distance d satisfaisant*

$$\begin{aligned} k = l(D) - l(D - D_0) = \quad & \deg D - g + 1 + l(K - D) - l(D_0 - D) \\ & d \geq n - \deg D \end{aligned}$$

COROLLAIRE 7.36 *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 $\{f_1, \cdots, f_k\}$ 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 l'écart 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 7.37 *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 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 $X \subset \mathbb{P}^2_{\mathbb{F}_q}$

$$X : u^{q+1} + v^{q+1} + w^{q+1} = 0$$

de genre $g(X) = \dfrac{q(q-1)}{2}$. 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 $n = q^3$

$$D_0 = \{x_1, \cdots, x_n\}, x_i \in X(\mathbb{F}_{q^2})\backslash x_\infty,\ (x_\infty = (0, 1, 1)),\ D_0 = m \cdot x_\infty.$$

Soit $C = C(D_0, D) = \mathcal{L}(D_0 - D)$ le code de Goppa correspondant, c'est un

$$[q^3, m - g + 1, d]_q - \text{ code , avec } d \geq n - m, \text{ avec le choix de } m : q^2 - q < m < q^3.$$

Si $q = 2$, $g(X) = 1$, X est une courbe elliptique, 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;
```

$$f := (x,\, y,\, z) \to x^3 + y^3 + z^3$$

```
> 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;
```

$$c := 0$$

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

$$2,\, [0,\, 1,\, 1]$$

$$3,\ [0,\ 1,\ 1+\alpha]$$
$$4,\ [1,\ \alpha,\ 0]$$
$$5,\ [1,\ 1,\ 0]$$
$$6,\ [1,\ 1+\alpha,\ 0]$$
$$7,\ [1,\ 0,\ \alpha]$$
$$8,\ [1,\ 0,\ 1]$$
$$9,\ [1,\ 0,\ 1+\alpha]$$

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

$$\langle 1, \frac{u}{v+w} \rangle$$

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 mathrice de contrôle de C.

Exercice. 2) Construire un code analogue pour $m = 2$.

Remarque 7.38 *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 (Théoème 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 : Ω-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

$$\Omega(D_0 - D) = \{\omega \in \Omega(X)^* | (\omega) + D_0 - D \geq 0\} \cup \{0\}$$

c'est à dire l'espace des formes différentielles ayant des zéros de multiplicités appropriées dans Supp$D$ et des pôles au plus simples dans SuppD_0. Alors l'application

$$\begin{aligned} &\mathrm{Res} : \Omega(D_0 - D) \to F_{q^n} \\ &\omega \mapsto (\mathrm{Res}_{x_1}(\omega), \cdots, \mathrm{Res}_{x_n}(\omega)) \end{aligned}$$

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 ω_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'exemen

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 $E : F \to F^n$ du codage de répétition pure. Calculer la probabilité P du decodage erroné.

(a) Montrer que

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

(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 emplilement de sphères parfait.

8.4. Déscription géométrique des codes de Reed-Solomon

(voir [Pa-Wo], p.139).

Soit $\mathbb{F}_{2^m} = \langle \alpha \rangle$, et soit $p(x) \in \mathbb{F}_{2^m}[x]$ pacourt le sous-espace linéaire $\mathcal{P}_k \subset \mathbb{F}_{2^m}[x]$ avec $d^\circ(p) \le k-1$ ou $p \equiv 0$. Alors l'ensemble des mots de la forme

$$(p(1), p(\alpha), \cdots, p(\alpha^{2^m-2}))$$

est un code de Reed-Solomon de longueur $n = 2^m - 1$, de polynôme

$$g(x) = (x-\alpha)(x-\alpha^2)\cdot\cdots\cdot(x-\alpha^r)$$

avec $r = 2^m - 1 - k$.

Solution. On remarque que l'application

$$\Phi : p \mapsto (p(1), p(\alpha), \ldots, p(\alpha^{2^m-2})) \in \mathbb{F}_q^n$$

est injective puisque $\mathrm{Ker}(\Phi) = 0$ par l'interpolation de Lagrange. On pose $C_1 = \mathrm{Im}(\Phi) = \Phi(\mathcal{P}_k)$. Une base convenable de C_1 :

$$c_i = \Phi(p_i), p_i = x^i \quad (i = 0, 1, \cdots, k-1) : c_i = (1, \alpha^i, \alpha^{2i}, \cdots, \alpha^{(2^m-2)i}),$$

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

$$\dim_{\mathbb{F}_{2^m}} C = \dim_{\mathbb{F}_{2^m}} C_1 = k$$

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 $L_m = \mathbb{F}_q[T_1, \cdots, T_m]_{d^\circ \leq 1}$ le $\mathbb{F}_q$-space 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

$$\mathcal{P} = \{P_1, \cdots, P_n\} \subset \mathbb{F}_q^m$$

tel que

$$\forall l \in L_m (\forall j = 1, \cdots, n, l(P_j) = 0 \Rightarrow l \equiv 0).$$

$$RM_q(1, n, m) = \mathrm{Im} Ev, \quad Ev : L_m \to \mathbb{F}_q^n, \quad f \mapsto ((f(P_1), \cdots, f(P_n))$$

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 $\mathbb{F}_q$-space 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

$$\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{array} \begin{pmatrix} 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 \\ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 \\ 0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1 \\ 0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 \\ 0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1 \\ 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1 \end{pmatrix} \quad (*)$$

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

8.6. Exemples de decodage des codes cycliques.

Soit α 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 α^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 $x^{i-1}g(x)$, $1 \le 1 \le k$.

(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 $0 \le a_1, a_2 \le 14$, $a_1 \ne a_2$. 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, η_1^{-1} et η_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 seul 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

$$v = (100111000000000).$$

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

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

Le polynôme $s(X)$ a la forme suivante :

$$s(X) = 1 + S_1X + (S_1^2 + S_3S_1^{-1})X^2 =$$
$$1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^2 + \alpha^3 + (1 + \alpha^2)(\alpha^2 + \alpha^3)^{-1})X^2$$
$$= 1 + (\alpha^2 + \alpha^3)X + (1 + \alpha + \alpha^3)X^2.$$

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 connait alors les erreurs : elles sont dans les positions correspondantes aux X^8 et X^{14}, c'est à direla 9^e et la 15^e 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 $\mathbb{F}_{256}^* = \langle\alpha\rangle$, $g = \prod_{12}^{43}(x - \alpha^{11j})$, $\alpha^8 = \alpha^7 + \alpha^2 + \alpha + 1$. Alors $n = 255, k = 223$ (un code de Reed-Solomon utilsé par NASA). Montrer que $d = 33$.

8.8. Décodage des codes de Goppa rationnels.

Soit $g(x)$ un polynôme unitaire irréductible sur $\mathbb{F}_{q^m}$, $L = \{\gamma_0, \gamma_1, \cdots, \gamma_{n-1}\} \subset \mathbb{F}_{q^m}$ avec $g(\gamma_i) \neq 0$ pour tout $i = 1, 2, \cdots, n-1$.

Le code de Goppa rationnel $\Gamma(L, g)$ est

$$\Gamma(L,g) = \left\{ (c_0, c_1, \cdots, c_{n-1}) \subset \mathbb{F}_q^n \Big| \sum_{i=0}^{n-1} \frac{c_i}{x - \gamma_i} \equiv 0 \bmod g(x) \right\}$$

(c'est-à-dire, la fraction à droite est de la forme d'une fraction irréductible $\frac{a(x)}{b(x)}$ avec $g(x)|a(x)$.

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 utilse comme syndrôme la fraction rationnelle

$$\sum_{i=0}^{n-1} \frac{e_i}{x - \gamma_i}$$

(le vecteur d'erreurs), représenté comme un polynôme $\bmod g$

$$S(x) = \sum_{i=0}^{n-1} e_i f_i(x) \bmod g$$

(le polynôme de syndrôme).

(a) Montrer que si $v = (v_0, v_1, \cdots, v_{n-1}) = w + e$, un mot recu, $w = (w_0, w_1, \cdots, w_{n-1}) \in \Gamma(L, g)$, $e = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{F}_q^n$, 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 :

$$u(x) \equiv s(x)S(x)(\mathrm{mod} g(x)) \tag{8.1}$$

(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'au $\mathrm{pgcd}(g(x), S(x))$ est atteint, 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 $X \subset \mathbb{P}^2_{\mathbb{F}_q}$

$$X : u^{q+1} + v^{q+1} + w^{q+1} = 0$$

de genre $g(X) = \dfrac{q(q-1)}{2}$. 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

$$\mathrm{Card}\,(X(\mathbb{F}_{q^2})) = 3(q+1) + (q-2)(q+1)^2 = q^3 + 1 = q^2 + 1 + 2g(X)q.$$

(a) Montrer que le nombre des points $X(\mathbb{F}_{q^2})$ de la courbe X est maximal possible.
On pose $n = q^3$

$$D_0 = \{x_1, \cdots, x_n\}, x_i \in X(\mathbb{F}_{q^2}) \backslash x_\infty,\ (x_\infty = (0, 1, 1)),\ D_0 = m \cdot x_\infty.$$

Soit $C = C(D_0, D) = \mathcal{L}(D_0 - D)$ le code de Goppa correspondant, c'est un

$$[q^3, m - g + 1, d]_q - \text{ code , avec } d \geq n - m, \text{ avec le choix de } m : q^2 - q \leq m < q^3.$$

Si $q = 2$, $g(X) = 1$, X est une courbe elliptique, $\mathrm{Card}\,(X(\mathbb{F}_4)) = 9$
(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

$$\langle 1, \frac{u}{v+w} \rangle$$

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 mathrice de contrôle de C.
(c) Construire un code analogue pour $m = 3$.

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 $V_q(n,r) = \mathrm{Card}\,(B_q(x,r))$.

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 $E : F^2 \to F^{2n}$ du codage de répitition 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 $q \geq 3$ éléménts, 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 $\mathbb{F}_q$-space 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

$$\mathcal{P} = \{P_1, \cdots, P_n\} \subset \mathbb{F}_q^{m+1} \backslash \{0\}$$

un sous-ensemble tel que l'espace vectoriel engendré par les $\{P_1, \cdots, P_n\}$ coïncide avec $\mathbb{F}_q^{m+1}$, et les points P_i ne sont pas proportionnels.

a) Montrer qu'obtient un $\left[n, m+1, n - \dfrac{q^m-1}{q-1}\right]_q$-code pour tout $n \geq \dfrac{q^m-1}{q-1}$.

b) Montrer que pour

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

un tel code atteinte 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

$$\begin{pmatrix}
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\\
0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1\\
0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1\\
0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1\\
0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
\end{pmatrix}$$

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].

A1. Structure

PROPOSITION-DÉFINITION A1 *: 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 A2 *: 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 A3 *: Soit $q = p^n$. Tout sous-corps de $\mathbb{F}_q$ est d'ordre p^m, où m est un diviseur de n.*

A2. Polynômes sur les corps finis

THÉORÈME A4 *: 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 A5 *Si on a un tel polynôme unitaire et r une racine de P dans $\mathbb{F}_q$, la famille $\{1, r, \cdots, r^{d-1}\}$ est une base du $\mathbb{F}^p$-espace vectoriel $\mathbb{F}_q$.*

THÉORÈME A6 *: 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 A7 *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 A8 *: 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 $r, r^q, \cdots, r^{q^{m-1}}$, éléments distincts de $\mathbb{F}_{q^m}$.*

COROLLAIRE A9 *: Le corps de décomposition d'un polynôme P de degré m irréductible sur $\mathbb{F}_q$ est $\mathbb{F}_{q^m}$.*

DÉFINITION A10 *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,* $\mathrm{ord}(P) = \mathrm{ord}(Q)$.

REMARQUE A11 *Si P est irréductible de degré m sur $\mathbb{F}_q$, alors l'ordre de P divise $q^m - 1$.*

THÉORÈME A12 *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 A13 *: Le degré de P irréductible sur $\mathbb{F}_q$ d'ordre e est l'ordre multiplicatif de q modulo e.*

DÉFINITION A14 *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 A15 *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 A16 *On appelle fonction de Moebius la fonction définie sur $\mathbb{N}$ par :*

$$\mu(n) = \begin{cases} 1 & , si\ n = 1 \\ (-1)^k & , si\ n \text{ est le produit de } k \text{ nombres premiers distincts,} \\ 0 & , si\ n \text{ est divisible par le carré d'un nombre premier.} \end{cases}$$

THÉORÈME A17 *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 A18 *Soit α un élément de $\mathbb{F}_{q^m}$ extension de $\mathbb{F}_q$. Soit d le degré de α sur $\mathbb{F}_q$ et P le polynôme minimal de α sur $\mathbb{F}_q$. Alors,*

(i) P est irréductible sur 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 $x^{q^d} - 1$ et $x^{q^m} - 1$.
(v) Les racines de P sont $\alpha, \alpha^q, \cdots, \alpha^{q^{d-1}}$ 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 α dans le groupe multiplicatif $\mathbb{F}^*_{q^m}$.
(vii) P est un polynôme primitif sur $\mathbb{F}_q$ si et seulement si α est d'ordre $q^d - 1$ dans $\mathbb{F}^*_{q^m}$.

THÉORÈME A19 (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 $d - 1$.*

B Annexe : Restes chinois, suite : Factorisation des Polynômes (F. SERGERAERT)

- **Référence : Don Knuth, The Art of Computer Programming, vol. 2, pp. 381-398.**

```
> restart ;
> with(linalg) :
```

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

- La factorisation des polynômes n'est pas un sujet vraiment facile, pas plus que la factorisation des entiers ! On présente ici le principal outil, la *méthode de Berlekamp*, qui concerne en fait la factorisation dans le cas du *corps de base fini*, puis on explique dans les cas les plus simples comment elle peut être utillisée pour obtenir la factorisation des polynômes à coefficients rationnels.

B1. Rappels sur les corps finis.

- On rappelle que p est premier si et seulement si tous les coefficients du binôme C(p, k) sont divisibles par p pour $0 < k < p$:

```
> seq(evalb(binomial(13,k) mod 13 = 0), k = 0..13) ;
> seq(evalb(binomial(9,k) mod 9 = 0), k = 0..9) ;
```

false, *true*, *true*, *true*, *true*, *true*, *true*, *true*, *true*, *true*, *true*, *true*, *false*

false, *true*, *true*, *false*, *true*, *true*, *false*, *true*, *true*, *false*

- Il en résulte que si on travaille dans $Z/\ p$, la formule $(a+b)^p = a^p + b^p$ est valide, en particulier $(a+1)^p = a^p + 1$ et il en résulte par récurrence, partant de $1^p = 1$, que $a^p = a$ dans $Z/\ p$.

- Un corps K fini est de caractéristique bien définie p : c'est le plus petit entier positif vérifiant p x 1 = (déf.) 1 + 1 + ... + 1 (p fois) = 0. Nécessairement, p est *premier*, sinon on aurait dans K des diviseurs de 0. Le corps K contient donc en particulier le sous-corps {0, 1, 2, ..., $p-1$} = $Z/\ p$ qu'on notera simplement Z_p et K est donc un espace vectoriel sur Z_p de dimension d ; il a alors p^d éléments. La formule démontrant que le *morphisme de Frobenius* $a \to a^p$ est *Z-linéaire* (et donc aussi Z_p-linéaire) : $(a+b)^p = a^p + b^p$ reste valable, mais il est maintenant *faux*, sauf si d=1, que $a^p = a$ pour tout a de K. En effet, puisque $X^p - X$ est de degré p, il ne peut avoir que p racines dans K, à savoir les éléments de Z_p dans K. On verra que cette remarque est la clé de la *méthode de Berlekamp* pour factoriser un polynôme à coefficients dans Z_p.

- Le groupe *multiplicatif* des éléments non nuls de K a pour cardinal $p^d - 1$, et il en résulte que pour tous ces éléments, la relation $a^{(p^d-1)} = 1$ est satisfaite ; le polynôme $X^{(p^d)} - X$ a donc pour racines tous ces éléments, et de plus l'élément nul. Les éléments de K sont donc tous racines de ce polynôme,et il en résulte que $X^{(p^d)} - X = \prod_a (X - a)$ où a parcourt exactement tous les éléments de K. Donc K est un (donc *le*) corps de décomposition de ce polynôme et il en résulte qu'il n'existe qu'un seul corps de cardinal p^d.

- Soit P dans $Z_p[\ X]$ un polynôme *irréductible* de degré d. Alors le quotient $Z_p[\ X]/\ P$ est un corps à p^d éléments. Il résulte de ce qui précède que la *classe d'isomorphisme* de ce corps est indépendante de P. Le polynôme P divise nécessairement le polynôme $X^{(p^d)}$- X ; en effet, si x est la classe de X

dans $Z_p\,[\,X]/\,P$, le polynôme minimal de x ne peut être que P, mais d'après ce qui est dit ci-dessus, x est aussi racine de $X^{(p^d)}$ - X et donc ce dernier polynôme est divisble par P. Il en résulte aussi que P est *entièrement* scindé dans K, donc une seule extension suffit toujours pour obtenir le corps de décomposition d'un polynôme irréductible : toute extension de Z_p est *galoisienne*.

Illustration.

```
> rnd := rand(0..2) :
```

- Contrairement au cas des coefficients entiers banals, il faut «tâtonner» un peu pour trouver un polynôme irréductible à coeffcients dans Z_3.

```
> _seed := 1639 :
> P := sort(X^3 + add(rnd()*X^i, i=0..2)) ;
```

$$P := X^3 + 2\,X^2 + 1$$

```
> Irreduc(P) mod 3 ;
```

true

- Il en résulte que le même polynôme est Q-irréductible :

```
> irreduc(P) ;
```

true

- Mais la réciproque est *fausse* : il arrive souvent qu'un polynôme soit Q-irréductible, mais pas Z_p-irréductible ; il arrive même que ceci se produise *quel que soit p* pour le même polynôme Q-irréductible, c'est le cas de $X^4 + 1$.

```
> irreduc(X^4+1) ;
```

true

```
> Irreduc(X^4+1) mod 67 ;
```

false

```
> Factor(X^4+1) mod 67 ;
```

$$(X^2 + 47\,X + 66)\,(X^2 + 20\,X + 66)$$

```
> Factor(X^4+1) mod nextprime(10^6) ;
```

$$(X^2 + 410588\,X + 1000002)\,(X^2 + 589415\,X + 1000002)$$

- Pour travailler *sous Maple* dans des extensions de Z_p, on procède comme pour les extensions de Q, mais il n'y a *aucune différence* dans l'usage initial de **RootOf**, c'est seulement *en fin de calcul* qu'on précise qu'on veut travailler dans Z_p et ses extensions, en *suffixant* par «**mod p**».

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

$$\alpha$$

- Le quotient $Z_3[\,X]/\,P$ est un espace vectoriel de degré 3 sur Z_3, dont les éléments sont tous de la forme $i + j\,\alpha + k\,\alpha^2$ pour i, j et k parcourant Z_3.

- Calcul du polynôme ayant *exactement* tous les éléments de $Z_3[\,X]/\,P$ comme racine

```
> mul(mul(mul(X-i-j*alpha-k*alpha^2,
> k=0..2),
> j=0..2),
> i=0..2) ;
```

$$X\,(X-\alpha^2)\,(X-2\,\alpha^2)\,(X-\alpha)\,(X-\alpha-\alpha^2)\,(X-\alpha-2\,\alpha^2)\,(X-2\,\alpha)\,(X-2\,\alpha-\alpha^2)$$
$$(X-2\,\alpha-2\,\alpha^2)\,(X-1)\,(X-1-\alpha^2)\,(X-1-2\,\alpha^2)\,(X-1-\alpha)\,(X-1-\alpha-\alpha^2)$$
$$(X-1-\alpha-2\,\alpha^2)\,(X-1-2\,\alpha)\,(X-1-2\,\alpha-\alpha^2)\,(X-1-2\,\alpha-2\,\alpha^2)\,(X-2)$$
$$(X-2-\alpha^2)\,(X-2-2\,\alpha^2)\,(X-2-\alpha)\,(X-2-\alpha-\alpha^2)\,(X-2-\alpha-2\,\alpha^2)$$
$$(X-2-2\,\alpha)\,(X-2-2\,\alpha-\alpha^2)\,(X-2-2\,\alpha-2\,\alpha^2)$$

- Développement du produit.

```
> Expand(%) mod 3 ;
```

$$2\,X+X^{27}$$

- ... autrement dit $X^{27}-X$. A comparer avec :

```
> collect(evala(expand(%%)), [X, alpha]) :
```

- Vérification de la propriété de divisibilité.

```
> Divide(X^27-X, P) mod 3 ;
```

true

- Et pour cause :

```
> Factor(P, alpha) mod 3 ;
```

$$(X+\alpha^2+\alpha+1)\,(X+2\,\alpha^2+1)\,(X+2\,\alpha)$$

- Car, comme expliqué plus haut, *une seule extension* suffit toujours à décomposer *complètement* le polynôme initial. Propriété en général fausse dans le cas rationnel :

```
> factor(P, alpha) ;
```

$$(X-\alpha)\,(X^2+2\,X+X\,\alpha+2\,\alpha+\alpha^2)$$

B2. Bases de la méthode de Berlekamp.

- On travaille dans l'anneau de polynômes $Z_p[\,X]$, pour un premier p, et sauf indication contraire, $Z_7[\,X]$ dans les exemples

- Soit P dans $Z_p[\,X]$ et $P=P_1...\,P_r$ sa décomposition en facteurs irréductibles. On suppose d'abord que P est sans facteur multiple, sinon ceci est détecté facilement par le PGCD du polynôme et du polynôme dérivé. Construisons un exemple de cette sorte.

```
> rnd := rand(0..6) :
> rndP := proc(n)
> RETURN(sort(X^n + add(rnd()*X^i, i=0..(n-1))))
> end :
> _seed := 1730 :
> P1, P2 := rndP(3), rndP(3) ;
```

$$P1,\ P2 := X^3+X^2+2\,X+4,\ X^3+4\,X^2+4\,X+2$$

- Le polynôme qui suit va certainement avoir un facteur multiple, mais on *fait semblant* de ne rien savoir à ce propos.

```
> P := sort(Expand(P1^2 * P2) mod 7) ;
```

$$P := X^9+6\,X^8+3\,X^7+3\,X^4+5\,X^3+5\,X^2+5\,X+4$$

- On détecte un facteur multiple éventuel par l'examen du PGCD entre le polynôme et son dérivé.

```
> Gcd(P, diff(P, X) mod 7) mod 7 ;
```

$$X^3 + X^2 + 2\,X + 4$$

- Et on commencerait par factoriser $X^3 + X^2 + 2\,X + 4$. Presque toujours (pas toujours, pourquoi ?) le polynôme initial est divisible par le carré de ce terme.

```
> Rem(P, %^2, X) mod 7 ;
```

$$0$$

```
> Quo(P, %%^2, X) mod 7 ;
```

$$X^3 + 4\,X^2 + 4\,X + 2$$

- ce qui redonne comme par hasard nos polynômes initiaux.

- On suppose donc désormais qu'il n'y a aucun facteur multiple dans P.

- Si $P = P_1 ... \ P_r$ est la décomposition de P en facteurs irréductibles, le *théorème du reste chinois* donne un isomorphisme canonique :

Z_p

$[\ X]/\ P = Z_p[\ X]/\ P_1 + ... \ Z_p[\ X]/\ P_r$. Les facteurs du second membre sont tous des corps, le premier membre n'est un corps que si P est irréductible, autrement dit si $r = 1$.

- Il en résulte, c'est l'astuce de Berlekamp, un test permettant, *sans connaître* r, de «deviner» sa valeur. Considérons en effet l'équation où l'inconnue V est un élément de $Z_p[\ X]/\ P$:

$V^P - V = 0$

- Si on traduit cette équation vers le second membre (qu'on ne connaît pas !), l'inconnue V devient un r-uplet $(\ V_1, ..., \ V_r)$, et comme l'isomorphisme utilisé *est un isomorphisme d'anneaux*, l'équation se transforme en r équations $V_i\ ^p = V_i$. Comme l'inconnue V_i est cette fois dans le *corps* $Z_p[\ X]/\ P_i$, on sait qu'il y a *exactement* les p racines $0, ..., \ p-1$ dans $Z_p[\ X]/\ P_i$. On en déduit que le cardinal des solutions est *exactement* p^r. Ainsi le cardinal de l'ensemble des solutions va nous donner le nombre fatidique r. D'une façon très approximative, on peut dire que Berlekamp observe que *plus* P est réductible, *«moins»* $Z_p[\ X]/\ P$ est un corps, et plus l'ensemble des solutions de notre équation va être vaste.

- Le deuxième élément clé de la méthode de Berlekamp consiste à remarquer que puisque l'application $V \to V^p$ est *linéaire*, l'équation $V^p - V = 0$ est, malgré les apparences, une *équation linéaire*, et on dispose donc de tous les outils linéaires classiques pour la traiter.

- **Exemple.** Soit à étudier si notre polynôme **P1** est irréductible dans Z_7. Il faut construire la matrice de l'application linéaire $V \to V^p - V$ dans $Z_p[\ X]/\ P_1$. On prend la base canonique $1, X, X^2$ (ou plus précisément leurs classes modulo P_1). L'image de X^j est donc la classe de $X^{(p\,j)} - X^j$, et les éléments de colonne correspondants sont les coefficients appropriés. On construit à part la procédure **BerlTerm** permettant le calcul du terme d'indices $(\ i, \ j)$ de la *matrice de Berlekamp* du polynôme P par rapport à Z_p.

```
> BerlTerm := proc(p::posint, P::polynom(integer, X),
> i::posint, j::posint)
> RETURN(coeff(Rem(X^(p*(j-1))-X^(j-1), P, X) mod p,
> X, i-1))
> end :
> BerlMatrix1 := matrix(3, 3, (i,j) -> BerlTerm(7, P1, i,j)) ;
```

$$BerlMatrix1 := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 0 & 6 & 1 \end{bmatrix}$$

- La *dimension du noyau* nous donne la dimension de l'espace des solutions, c'est-à-dire le nombre des facteurs irréductibles. On voit que le rang est 2, l'espace des solutions est donc de dimension 1, ce ne peut être que Z_7, solutions «inévitables», et notre polynôme est donc irréductible. Pour obtenir le rang de cette matrice dans le cas général, il faut utiliser la procédure **Nullspace** combinée avec **mod.**

```
> Nullspace(BerlMatrix1) mod 7 ;
```

$$\{[1, 0, 0]\}$$

- Vérification.

```
> Irreduc(P1) mod 7 ;
```

$$true$$

- Mëme travail avec P_2.

```
> BerlMatrix2 := matrix(3, 3, (i,j) -> BerlTerm(7, P2, i,j)) ;
```

$$BerlMatrix2 := \begin{bmatrix} 0 & 4 & 0 \\ 0 & 5 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

```
> Nullspace(BerlMatrix2) mod 7 ;
```

$$\{[1, 0, 0], [0, 0, 1]\}$$

```
> Irreduc(P2) mod 7 ;
```

$$false$$

- On voit qu'ici notre équation $V^p - V$ a 49 solutions, $49 = 7^2$, et notre polynôme a donc *deux* facteurs irréductibles. Il reste à les déterminer. Dans un cas si simple, c'est très facile, il suffit de chercher l'élément a de Z_7 nécessairement racine de P_2 et le quotient par $X - a$ donnera l'autre facteur irréductible. Mais on veut expliquer comment il faut procéder dans le cas général. C'est le sujet de la section suivante.

```
> for i from 0 to 6 do
> if Eval(P2, X=i) mod 7 = 0 then print(i) fi
> od ;
```

$$3$$

```
> Quo(P2, X-3, X) mod 7 ;
```

$$X^2 + 4$$

```
> Irreduc(%) mod 7 ;
```

$$true$$

```
> Factor(P2) mod 7 ;
```

$$(X^2 + 4)\,(X + 4)$$

B3. Trouver les facteurs irréductibles.

- Les solutions de l'équation $V^p - V = 0$ ne donnent pas seulement le *nombre* de facteurs irréductibles, chaque solution donne aussi une décomposition, en général *partielle*, mais toujours *non triviale*, du polynôme proposé en facteurs de degrés plus petits. Ceci est dû au fait que dans Z_p, nous avons la décomposition :

$V^p - V = V\,(V-1)$
... $(\,V - p - 1)$.

- Dire que V est une solution de $V^p - V$ dans $Z_p[\,X]/\,P$ revient à dire que P divise $V^p - V$, mais la factorisation ci-dessus de $V^p - V$ va justement nous permettre de «découper en tranches» le polynôme P.

- D'abord si P est irréductible, les seules solutions de $V^p - V = 0$ sont les éléments de Z_p, auquel cas le «polynôme» $V^p - V$ est non seulement divisible par P, il est même nul ! et aucune «information» ne peut être obtenue, heureusement.

- Par contre si P est factorisable, on va avoir des solutions différentes. Ces solutions vont être de «vrais» polynômes (non constants), et un tel polynôme V va être de degré forcément $< d = \text{degree}(P)$. On a alors le résultat suivant :

$P = \prod_{i=0}^{p-1} \text{PGCD}(P,\, V - i)$
.

- En effet $V^p - V$ est divisible par P, et donc tout facteur irréductible de P va diviser $V^p - V$ et se retrouver dans l'un des PGCD. Donc P divise le produit. Inversement, comme les $V - i$ sont *premiers deux à deux* (pourquoi ?), le même facteur de P ne peut pas se retrouver deux fois à droite. Compte tenu par ailleurs du fait que $\text{degree}(V - i) < d$, on voit donc qu'on a ainsi, quel que soit V solution «non constante» de $V^p - V = 0$, une factorisation non triviale de P.

- Essayons ce mécanisme avec notre polynôme P_2. Un V non trivial est à trouver dans le noyau de la matrice de Berlekamp :

```
> eval(BerlMatrix2) ;
```

$$\begin{bmatrix} 0 & 4 & 0 \\ 0 & 5 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

```
> Nullspace(BerlMatrix2) mod 7 ;
```

$$\{[1,\, 0,\, 0],\, [0,\, 0,\, 1]\}$$

- Le générateur [1,0,0] du noyau correspond aux solutions triviales de $V^p - V = 0$, mais l'autre, [0,0,1], expression dans notre base du polynôme X^2, est une solution non triviale.

```
> Divide(X^14-X^2, P2) mod 7 ;
```

$$\textit{true}$$

```
> seq(Gcd(P2, X^2-i) mod 7, i=0..6) ;
```

$$1,\, 1,\, X + 4,\, X^2 + 4,\, 1,\, 1,\, 1$$

- Et on a bien notre décomposition.

```
> evalb(P2 = Expand(mul(gcdi, gcdi=[%])) mod 7) ;
```

true

- Dans le cas général, il n'y a pas de raison que la décomposition *complète* de P soit ainsi obtenue à l'aide d'une seule solution non triviale V. C'est forcément ce qui arrive si le nombre de facteurs de P est plus grand que p. Construisons P de sorte qu'il ait au moins 8 facteurs.

```
> _seed := 1054 :
> for i from 1 to 8 do
> P||i := rndP(2)
> end do ;
```

$$P1 := X^2 + 4\,X + 2$$
$$P2 := X^2 + 2\,X + 3$$
$$P3 := X^2 + 3\,X + 2$$
$$P4 := X^2 + 2\,X$$
$$P5 := X^2 + 5\,X + 5$$
$$P6 := X^2 + 2\,X + 5$$
$$P7 := X^2 + 6\,X$$
$$P8 := X^2 + 5\,X + 1$$

```
> P := sort(Expand(mul(P||i, i=1..8)) mod 7) ;
```

$$P := X^{16} + X^{15} + 6\,X^{14} + 4\,X^{13} + 3\,X^{12} + 6\,X^{11} + 6\,X^{10} + 5\,X^9 + 3\,X^8 + 3\,X^7 + 5\,X^6 + X^5 + 2\,X^4 + X^3 + 2\,X^2$$

- Mais on peut avoir des facteurs multiples, qu'il faut éliminer pour que notre exemple soit correct.

```
> Gcd(P, diff(P,X) mod 7) mod 7 ;
```

$$X^5 + 6\,X^4 + 4\,X^3 + 5\,X^2 + 5\,X$$

```
> P := Quo(P, %, X) mod 7 ;
```

$$P := X^{11} + 2\,X^{10} + 4\,X^9 + 2\,X^8 + 2\,X^7 + 5\,X^6 + X^5 + X^4 + 4\,X^2 + 6\,X$$

```
> Gcd(P, diff(P,X) mod 7) mod 7 ;
```

$$1$$

- Donc plus de facteurs multiples.

```
> BerlMatrix := matrix(11,11, (i,j) -> BerlTerm(7,P,i,j)) :
> Kernel := Nullspace(%) mod 7 ;
```

$$Kernel := \{[0, 5, 3, 5, 0, 0, 1, 0, 0, 0, 0], [0, 5, 2, 5, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 3, 0, 1, 0, 0, 0, 0, 0, 1, 0], [0, 1, 0, 3, 0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0], [0, 4, 6, 4, 1, 0, 0, 0, 0, 0, 0]\}$$

- Les éléments du noyau sont des *vecteurs*, ce qui est techniquement désagréable pour la suite, on les transforme tous en *listes.*

```
> Kernel := map(convert, Kernel, list) ;
```

$$Kernel := \{[0, 5, 3, 5, 0, 0, 1, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 5, 2, 5, 0, 0, 0, 0, 0, 0, 1], [0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 3, 0, 1, 0, 0, 0, 0, 0, 1, 0], [0, 1, 0, 3, 0, 1, 0, 0, 0, 0, 0], [0, 4, 6, 4, 1, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]\}$$

```
> r := nops(Kernel) ;
```

$$r := 8$$

- Donc 8 facteurs irréductibles. On retire l'élément de noyau correspondant aux solutions triviales.

```
> Kernel := Kernel minus {[1, 0$10]} ;
```

$$\begin{aligned} \mathit{Kernel} := \{&[0, 5, 3, 5, 0, 0, 1, 0, 0, 0, 0], [0, 5, 2, 5, 0, 0, 0, 0, 0, 0, 1], \\ &[0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 3, 0, 1, 0, 0, 0, 0, 0, 1, 0], \\ &[0, 1, 0, 3, 0, 1, 0, 0, 0, 0, 0], [0, 4, 6, 4, 1, 0, 0, 0, 0, 0, 0], \\ &[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]\} \end{aligned}$$

```
> Vvector1 := Kernel[1] ;
```

$$\mathit{Vvector1} := [0, 5, 3, 5, 0, 0, 1, 0, 0, 0, 0]$$

```
> V1 := sort(add(Vvector1[i]*X^(i-1), i=1..11)) ;
```

$$\mathit{V1} := X^6 + 5\,X^3 + 3\,X^2 + 5\,X$$

```
> factors1 := {seq(Gcd(V1-i, P) mod 7, i=0..6)} minus{1} ;
```

$$\mathit{factors1} := \{X^2 + 2\,X + 5,\ X^2 + 5\,X + 5,\ X + 1,\ X^3 + 4\,X^2 + 6,\ X^3 + 4\,X^2 + 2\,X\}$$

```
> nops(factors1) ;
```

$$5$$

```
> evalb(P = Expand(mul(gcdi, gcdi=factors1)) mod 7) ;
```

$$\mathit{true}$$

- On voit qu'une décomposition non triviale de P est bien obtenue, mais ce n'est évidemment pas là la décomposition *complète* en facteurs irréductibles, puisqu'on a construit P comme un produit de facteurs de degré 2. Comme seulement 7 cases sont disponibles dans le résultat, certainement certains facteurs ainsi obtenus sont encore réductibles.

- Le point suivant consiste à dire qu'en essayant au besoin les autres éléments «non triviaux» du noyau de la matrice de Berlekamp, on va réussir, en *recoupant* les résultats, à obtenir la factorisation complète. Expliquons ce qu'il faut entendre par *recouper*.

- Prenons un autre vecteur de notre noyau.

```
> Vvector2 := Kernel[2] ;
```

$$\mathit{Vvector2} := [0, 5, 2, 5, 0, 0, 0, 0, 0, 0, 1]$$

```
> V2 := sort(add(Vvector2[i]*X^(i-1), i=1..11)) ;
```

$$\mathit{V2} := X^{10} + 5\,X^3 + 2\,X^2 + 5\,X$$

```
> factors2 := {seq(Gcd(V2-i, P) mod 7, i=0..6)} minus {1} ;
```

$$\mathit{factors2} := \{X^2 + X,\ X + 2,\ X + 5,\ X^4 + 4\,X^3 + 5\,X^2 + 2\,X + 1,\ X^3 + 4\,X^2 + 2\}$$

```
> nops(factors2) ;
```

$$5$$

- On *voit* que la factorisation n'est pas la même que celle précédemment obtenue. On va démontrer juste après que c'est toujours le cas. On obtient donc une «meilleure» factorisation en prenant l'*intersection*, à coups de PGCD, des deux factorisations.

```
> factors12 := {seq(seq(Gcd(f1,f2) mod 7,
> f2 = factors2),
> f1 = factors1)}
> minus {1} ;
```

$$factors12 := \{X^2 + 2\,X + 3,\ X^2 + 2\,X + 5,\ X^2 + 5\,X + 5,\ X,\ X + 1,\ X + 2,\ X + 5,\ X + 6\}$$

```
> nops(factors12) ;
```

8

- La factorisation est donc complète. D'autres fois, il faut encore continuer.

- Expliquons pourquoi la méthode des *recoupements* aboutit. Pour mieux faire comprendre, on se contente du cas $r = 3$. Donc $P = P_1\,P_2\,P_3$ et :

$K\,[X]$
/ $P = K\,[X]/\ P_1\ +\ K\,[X]/\ P_2\ +\ K\,[X]/\ P_3$.

- Toute solution de notre équation $V^p - V = 0$ a deux interprétations. Du côté «gauche», c'est un polynôme «modulo P» ; mais du côté droit, comme pour chaque facteur les seules solutions sont les polynômes «constants», ces solutions sont essentiellement des triplets (α_1, α_2, α_3) d'entiers modulo p. La correspondance de la droite vers la gauche n'est rien d'autre que le *théorème des restes chinois.* Le relèvement de (1,0,0) est le produit $B_1\,P_2\,P_3$ de la relation de Bezout $A_1\,P_1 + B_1\,P_2\,P_3 = 1$, car il s'agit d'avoir un polynôme divisible par P_2 et P_3, mais égal à 1 modulo P_1. De la même façon, le relèvement de (0,1,0) (resp. (0,0,1)) est le produit $B_2\,P_1\,P_3$ (resp. $B_3\,P_1\,P_2$) avec des interprétations analogues. Une solution «triviale» est de la forme (α, α, α). Donc une solution non triviale vérifie par exemple $\alpha_1 <> \alpha_2$. Soit V l'interprétation polynôme de cette solution. Alors $V - \alpha_1 = (\alpha_2 - \alpha_1)\,B_2\,P_1\,P_3 + (\alpha_3 - \alpha_1)\,B_3\,P_1\,P_2$, alors que $V - \alpha_2 = (\alpha_1 - \alpha_2)\,B_1\,P_2\,P_3 + (\alpha_3 - \alpha_2)\,B_3\,P_1\,P_2$. Il en résulte que $V - \alpha_1$ est divisible par P_1 et $V - \alpha_2$ est divisible par P_2. Donc le «traitement» de V va forcément «séparer» les facteurs P_1 et P_2. Maintenant l'ensemble des solutions à gauche correspond à l'ensemble des solutions à droite, et il y a donc forcément une solution à gauche correspondant (sans qu'on le «voie») à un cas où $\alpha_1 <> \alpha_2$. Il en est forcément de même pour l'un des vecteurs de la base du noyau, sinon on aurait $\alpha_1 = \alpha_2$ pour *tous* les éléments du noyau, ce qui est exclu. Le même travail peut être fait en général pour toutes les paires d'indices, d'où le fait que les éléments de la base du noyau suffisent à complètement factoriser. CQFD.

- On est prêt maintenant pour une procédure générale de factorisation. Elle va utiliser notre procédure **BerlTerm** déjà construite.

```
> Berl := proc(p::posint, P::polynom(integer, X))
> local d, r, i,
> BerlMatrix, Kernel,
> Vvector, V,
> result, new_factors ;
> d := degree(P, X) ;
> # Erreur si facteur multiple.
> if Gcd(P, diff(P, X)) mod p <> 1 then
> ERROR(`The polynom is not squarefree.`)
> fi ;
> BerlMatrix := matrix(d,d, (i,j) -> BerlTerm(p,P,i,j)) ;
> Kernel := Nullspace(BerlMatrix) mod p ;
> # r = nombre de facteurs irréductibles.
> r := nops(Kernel) ;
> # Si r = 1, le polynôme est irréductible.
> if r = 1 then RETURN([P]) fi ;
> Kernel := map(convert, Kernel, list) ;
> # Suppression de la solution triviale.
> Kernel := Kernel minus {[1, 0$(d-1)]} ;
> # Pour signaler le début de l'algorithme.
> result := {} ;
> # Il faut parcourir les solutions non triviales et # arrêter quand le nombre de facteurs requis est atteint.
> for Vvector in Kernel while nops(result) < r do
> # Expression polynômiale du nouveau vecteur solution considéré.
> V := add(Vvector[i]*X^(i-1), i=1..d) ;
> # Découpage en tranches.
> new_factors := {seq(Gcd(P, V-i) mod p, i=0..(p-1))}
> minus {1} ;
> # Recoupement (éventuel) avec ce qui a été précédemment fait.
> if nops(result) > 1 then
> result := {seq(seq(Gcd(f1,f2) mod p,
> f2 = new_factors),
> f1 = result)}
> minus {1}
> else result := new_factors
> fi
> od ;
> result := convert(result, list) ;
> result := map(sort, result) ;
> result := sort(result,
> (P1,P2)->evalb(degree(P1)<degree(P2))) ;
> RETURN(map(sort,result))
> end :
> fs := Berl(7, P) ;
```

$$fs := [X+6,\ X+5,\ X+2,\ X+1,\ X,\ X^2+5\,X+5,\ X^2+2\,X+5,\ X^2+2\,X+3]$$

```
> nops(fs) ;
```

$$8$$

```
> P - Expand(mul(f, f=fs)) mod 7 ;
```

$$0$$

```
> Berl(73, X^4+1) ;
```

$$[X+51,\ X+10,\ X+63,\ X+22]$$

- Des polynômes complexes peuvent ainsi être factorisés.

```
> _seed := 1535 :
> P := rndP(50) ;
```

$$P := X^{50} + 2\,X^{49} + 5\,X^{48} + 6\,X^{47} + 3\,X^{46} + 2\,X^{45} + X^{43} + X^{42} + 4\,X^{41} + 2\,X^{40} + 5\,X^{39} + 5\,X^{38} + 6\,X^{37} + 3\,X^{36} + 3\,X^{35} + 3\,X^{31} + 4\,X^{29} + 2\,X^{28} + 3\,X^{27} + 4\,X^{26} + 3\,X^{25} + X^{23} + 3\,X^{22} + 6\,X^{20} + X^{19} + X^{18} + 4\,X^{17} + 3\,X^{16} + 5\,X^{15} + 6\,X^{13} + X^{12} + 6\,X^{11} + 3\,X^{10} + 3\,X^9 + 6\,X^8 + X^7 + 3\,X^6 + 5\,X^5 + 3\,X^4 + 2\,X^3 + 5\,X + 6$$

```
> fs := Berl(7, P) :
> map(print, fs) :
```

$$X^2 + 5\,X + 2$$

$$X^{11} + X^{10} + X^9 + 6\,X^7 + 6\,X^6 + 5\,X^5 + 5\,X^4 + 4\,X^3 + 2\,X^2 + 4\,X + 4$$

$$X^{37} + 3\,X^{36} + 3\,X^{34} + 5\,X^{33} + X^{31} + 5\,X^{30} + 5\,X^{28} + 3\,X^{27} + 3\,X^{26} + 5\,X^{25} + 3\,X^{24} + 2\,X^{23} + 3\,X^{22} + 3\,X^{21} + 3\,X^{20} + X^{19} + 6\,X^{17} + 3\,X^{16} + 6\,X^{15} + 6\,X^{14} + 6\,X^{13} + 4\,X^{12} + 3\,X^{10} + 6\,X^9 + X^8 + 6\,X^7 + X^6 + 3\,X^5 + 2\,X^4 + 3\,X^3 + 5\,X + 6$$

```
> nops(fs) ;
```

$$3$$

```
> map(item -> Irreduc(item) mod 7, fs) ;
```

$$[true,\ true,\ true]$$

```
> P - Expand(mul(item, item=fs)) mod 7 ;
```

$$0$$

- Un cas irréductible.

```
> Berl(7, fs[1]) ;
```

$$[X^2 + 5\,X + 2]$$

B4. Factorisation des polynômes à coefficients entiers.

- Les méthodes efficaces de factorisation des polynômes à coefficients entiers commencent toutes par $\mathbb{Z}_p$-factoriser pour un p premier, ou puissance d'un nombre premier, avec p assez grand. Par examen de la taille des coefficients pour une $\mathbb{Z}$-factorisation éventuelle, on finit par en déduire la $\mathbb{Z}$-factorisation cherchée. Le point clé dans cette direction consiste à majorer les racines (en général complexes) du polynôme à factoriser.

- Soit donc $P = X^n + a_{n-1}\,X^{(n-1)} + ... + a_0$ un polynôme *unitaire* à coefficients complexes. Alors toute racine de P est *strictement* majorée par : $A = 2\max(|a_i|^{(\frac{1}{n-i})},\ i = 0..n-1)$. En effet on peut écrire $P = X^n\left(1 + \left(\sum_{i=1}^{n} \frac{a_{n-i}}{X^i}\right)\right)$ où $|a_{n-i}| \leq (\frac{A}{2})^i$. La somme de la dernière expression devient une progression géométrique *strictement* majorée par $\frac{A}{2\,X}\ \frac{1}{1-\frac{A}{2X}} = \frac{A}{2\,X-A} \quad \leq 1$ si $A \leq |X|$. Donc la dernière inégalité implique $\mathrm{P}(X)$ non nul. Le raisonnement est en défaut si $A = 0$, mais ce cas est sans intérêt, car alors toutes les racines sont nulles. Procédure conséquente.

```
> RootsSup := proc(P::polynom(rational, X))
> local dgr ;
> dgr := degree(P, X) ;
> if coeff(P, X, dgr) <> 1 then
> ERROR(sprintf("Polynôme %a non unitaire.", P))
> fi ;
> RETURN(2 * max(seq(evalf(abs(coeff(P, X, i))^(1/(dgr-i))),
> i=0..(dgr-1))))
> end :
```

- Soit donc P un Z-polynôme *unitaire* (il est facile de se ramener à ce cas par «changement de variable») où les coefficients sont majorés par A. Il est élémentaire d'en déduire que les racines (complexes) sont aussi *strictement* majorées en module par $A+1$; il existe des inégalités sensiblement plus fines à ce sujet, mais pour simplifier, on se contentera ici de celle-ci. Puisqu'un facteur potentiel de P est un produit de $(X-\alpha)$ où α parcourt certaines racines de P, on en déduit des majorations pour les coefficients d'une Z-factorisation éventuelle. On peut alors conclure en examinant la Z_p-factorisation de P pour p assez grand.

- Examinons par exemple le cas de X^4+1. Ici $A=2$ mais on sait bien que les quatre racines sont de module 1. Tentons la Z_3-factorisation.

```
> RootsSup(X^4+1) ;
> Berl(3, X^4+1) ;
```

$$2.$$

$$[X^2+2\,X+2,\ X^2+X+2]$$

- Il en résulte qu'une Z-factorisation a au plus deux facteurs de degré 2, où les termes constants sont de la forme $3\,n+2$, mais ce pourrait être -1, et on ne peut conclure. On augmente p.

```
> Berl(5, X^4+1) ;
```

$$[X^2+3,\ X^2+2]$$

- Cette fois on a gagné, parce que l'un des facteurs a un terme constant de la forme $5\,n+2$, incompatible avec les modules connus des racines de X^4+1. Donc X^4+1 est Z-irréductible et donc (voir la démonstration du théorème de Gauss sur la factorialité de $Z[\,X]$) Q-irréductible.

```
> irreduc(X^4+1) ;
```

true

- Avec un polynôme moins trivial.

```
> _seed := 921 :
> P := rndP(10) ;
> RootsSup(P) ;
```

$$P := X^{10}+2\,X^9+3\,X^8+5\,X^7+2\,X^6+4\,X^5+4\,X^4+2\,X^3+5\,X^2+X+6$$

$$4.$$

- Toute racine est majorée par 4, et si une factorisation non triviale est possible, elle aura un facteur de degré au plus 5 où le coefficient du terme après le terme de plus haut degré sera donc majoré par 20 d'où l'idée d'utiliser 41.

```
> Berl(41,P) ;
```

$$[X^{10}+2\,X^9+3\,X^8+5\,X^7+2\,X^6+4\,X^5+4\,X^4+2\,X^3+5\,X^2+X+6]$$

- On est chanceux, le polynôme est donc irréductible.

```
> _seed := 1529 :
> P := rndP(10) ;
> RootsSup(P) ;
```

$$P := X^{10}+6\,X^9+3\,X^8+6\,X^7+X^6+6\,X^5+2\,X^4+2\,X^3+5\,X^2+2\,X+3$$

$$12.$$

- On essaie un nombre premier $> 10\ .\ 12 = 120$, par exemple 127

```
> Berl(127, P) ;
```

$$[X+45,\ X+6,\ X^3+86\,X^2+108\,X+28,\ X^5+123\,X^4+105\,X^3+51\,X^2+13\,X+19]$$

- Le facteur $X+45$ seul ne peut pas provenir d'une Z-factorisation, ni le facteur de degré 3. Dans une telle situation il faut essayer si -6 est racine :

```
> subs(X=-6, P) ;
```

$$3361563$$

- Le produit $(X+6)\,(X+45)$ va commencer par $X^2+51\,X$ et est aussi exclu. Essayons un autre nombre premier, pour voir.

```
> Berl(131, P) ;
```

$$[X^2+104\,X+55,\ X^8+33\,X^7+53\,X^6+15\,X^5+111\,X^4+82\,X^3+41\,X^2+5\,X+112]$$

- Mais le facteur de degré 2 est impossible et le polynôme P est donc Q-irréductible. Vérification.

```
> irreduc(P) ;
```

$$true$$

- Il se trouve qu'on aurait pu essayer dans ce cas un entier un peu plus... petit. Ce polynôme est en effet déjà irréductible modulo 5 !

```
> Berl(5,P) ;
```

$$[X^{10}+6\,X^9+3\,X^8+6\,X^7+X^6+6\,X^5+2\,X^4+2\,X^3+5\,X^2+2\,X+3]$$

- Un polynôme aléatoire est presque toujours irréductible. Forçons le choix d'un polynôme exercice certainement *réductible*.

```
> _seed := 940 :
> P1, P2 := rndP(4), rndP(6) :
> P := sort(expand(P1 * P2)) ;
```

$$P := X^{10}+4\,X^9+7\,X^8+16\,X^7+30\,X^6+34\,X^5+49\,X^4+51\,X^3+28\,X^2+30\,X+20$$

```
> RootsSup(P) ;
```

$$8.$$

```
> nextprime(2*5*8) ;
```

$$83$$

```
> Berl(83, P) ;
```

$$[X+9,\ X+74,\ X+68,\ X+2,\ X+1,\ X^2+37\,X+64,\ X^3+62\,X^2+40\,X+67]$$

- On doit donc examiner si -1 et -2 sont racines.

```
> eval(P, X=-1) ;
```

$$0$$

```
> eval(P, X=-2) ;
```

$$0$$

- Le reste est un peu confus. Divisons et réexaminons la question.

```
> P2 := quo(P, (X+1)*(X+2), X) ;
```

$$P2 := X^8+X^7+2\,X^6+8\,X^5+2\,X^4+12\,X^3+9\,X^2+10$$

```
> RootsSup(P2) ;
```

$$4.000000000$$

```
> Berl(37, P2) ;
```

$$[X^2+2,\, X^6+X^5+6\,X^3+2\,X^2+5]$$

- Essai.

```
> rem(P2, X^2+2, X) ;
```

$$0$$

- D'où la Q-factorisation définitive. Vérification.

```
> factor(P) ;
```

$$(X+1)\,(X+2)\,(X^6+X^5+6\,X^3+2\,X^2+5)\,(X^2+2)$$

- En «bricolant» de la sorte, on arrive en général à factoriser les polynômes pas trop compliqués, mais *programmer* une méthode *générale* est autrement complexe. Il serait confortable de savoir faire la factorisation modulo un *grand* nombre premier, mais la méthode de Berlekamp, telle qu'elle a été programmée précédemment, échoue alors, parce que l'équation de Berlekamp $V^p - V = 0$ devient trop difficile à résoudre, à cause de la taille de p.

```
> # Ne pas effectuer sous Maple 6.
> # Berl(nextprime(10^6), P) ;
```

- Beaucoup d'améliorations peuvent être intégrées à la procédure **Berl**, mais elle n'ira jamais très loin pour une factorisation par rapport à de *grands nombres premiers*. Le corps «fini» Z_p est bien trop grand pour mener les calculs bien loin. Penser en particulier à la factorisation qu'il faut utiliser $V^p - V = V\,(V-1)\,...\,V-p-1$, où le nombre de facteurs est justement p ! ! Une autre solution devient alors beaucoup plus intéressante, basée sur le *lemme de Hensel*.

B5. Lemme de Hensel.

- Le *lemme de Hensel* est une méthode largement utilisée en algèbre commutative consistant à résoudre un problème d'abord «approximativement» modulo un idéal m puis à *affiner* en travaillant modulo les puissances de cet idéal, puissances *de plus en plus petites*. Le problème de la factorisation des polynômes est justement un cadre parfait pour comprendre le mécanisme.

- L'outil essentiel pour démontrer le lemme de Hensel consiste à utiliser judicieusement une relation *à la Bezout* pour exprimer un élément quelconque, et pas seulement 1, en fonction de deux éléments u et v premiers entre eux dans un anneau principal.

- Commençons pour comprendre le principe par le cas entier.
  Proposition B1 *Si u et v sont deux entiers positifs premiers entre eux, alors tout entier x ε [0, uv[s'exprime d'une façon et d'une seule sous la forme :*

$$x = \alpha\,u + \beta\,v \bmod (u\,v)$$

  avec $\alpha \in [0, v[$ et $\beta \in [0, u[$.

  Démonstration. Soit $1 = a\,u + b\,v$ une relation de Bezout entre u et v. Par multiplication par x, on obtient $x = x\,a\,u + x\,b\,v$, mais $x\,a$ et $x\,b$ sont en général trop grands ; on les divise donc respectivement par v et u, pour obtenir $x = \alpha\,u + \beta\,v + \gamma\,u\,v$ où on peut choisir α et β dans les intervalles requis. Si un autre choix était possible, on trouverait par différences une relation

$\alpha\, u + \beta\, v = 0 \bmod (u\, v)$ avec α non nul et de module $< v$; mais ceci contredit la divisibilité de α par v.

- Programme conséquent.

```
>  Bezout2 := proc(x::nonnegint, u::posint, v::posint)
>  local a, b, gcd ;
>  gcd := igcdex(u, v, 'a', 'b') ;
>  if gcd <> 1 then ERROR(
>  sprintf("les nombres %a et %a
>  ne sont pas premiers entre eux.",
>  u, v))
>  fi ;
>  RETURN(x*a mod v, x*b mod u)
>  end :
>  Bezout2(4, 6, 15) ;
```

```
Error, (in Bezout2) les nombres 6 et 15
            ne sont pas premiers entre eux.
```

```
>  Bezout2(4, 3, 5) ;
```

$$3,\, 2$$

```
>  evalb(4 = 3 * 3 + 2 * 5 mod (3*5)) ;
```

$$\mathit{true}$$

- Le même résultat est valide pour les polynômes à coefficients dans un corps, sous une forme encore plus confortable.
  PROPOSITION B2 *Si U et V sont deux polynômes de degrés respectifs m et n, premiers entre eux, alors pour tout polynôme P de degré $< m+n$, il existe un unique polynôme A (resp. B) de degré $< n$ (resp. $< m$) tel que $P = AU + BV$.*

- La démonstration est la même mais un examen de degrés montre qu'on peut même se dispenser de l'imprécision «modulo UV». Le programme conséquent suit.

```
>  Bezout3 := proc(P::polynom(rational, X),
>  U::polynom(rational, X),
>  V::polynom(rational, X))
>  local gcd, A, B ;
>  gcd := gcdex(U, V, X, 'A', 'B') ;
>  if gcd <> 1 then ERROR(
>  sprintf("les polynômes %a et %a
>  ne sont pas premiers entre eux.",
>  U, V))
>  fi ;
>  RETURN(rem(P*A, V, X), rem(P*B, U, X))
>  end :
>  _seed := 1925 :
>  P, U, V := rndP(5), rndP(3), rndP(3) ;
```

$$P,\, U,\, V := X^5 + 5\,X^4 + 6\,X^3 + 2\,X^2 + 6\,X + 2,\; X^3 + X^2 + X + 6,\; X^3 + 2\,X^2 + 3$$

```
>  A, B := Bezout3(P, U, V) ;
```

$$A,\, B := \frac{1}{6} - \frac{7}{18}\,X^2 - \frac{2}{3}\,X,\; \frac{25}{18}\,X^2 + \frac{59}{18}\,X + \frac{1}{3}$$

```
>  expand(P - A*U - B*V) ;
```

$$0$$

- Ceci est valable quel que soit le corps, par exemple Z_7, mais il faut adapter la procédure.

```
> `type/Z7` := proc(obj::anything)
> RETURN(type(obj, And(integer, Range(-1, 7))))
> end :
> type(3, Z7), type(-3, Z7) ;
```

true, *false*

```
> Bezout4 := proc(P::polynom(Z7, X),
> U::polynom(Z7, X),
> V::polynom(Z7, X))
> local gcd, A, B ;
> gcd := Gcdex(U, V, X, 'A', 'B') mod 7 ;
> if gcd <> 1 then ERROR(
> ### WARNING: %x or %X format should be %y or %Y if used with
> floating point arguments
> ### WARNING: incomplete string;  use " to end the string
> sprintf("les polynômes %a et %a
> ne sont pas premiers entre eux.",
> U, V))
> fi ;
> RETURN(Rem(P*A, V, X) mod 7, Rem(P*B, U, X) mod 7)
> end :
```

- On prend les mêmes polynômes, mais l'interprétation est différente.

```
> A, B := Bezout4(P, U, V) ;
```

$$A,\, B := 4\,X + 6,\, X^2 + 6\,X + 5$$

```
> Expand(P - A*U - B*V) mod 7 ;
```

$$0$$

- Un énoncé équivalent est obtenu pour des polynômes à coefficients entiers, à condition de faire intervenir une égalité «modulo p» pour un premier p. Cet énoncé va pouvoir être généralisé au cas p^k où le quotient $Z/\,p^k$ n'est plus un corps.
  PROPOSITION B3 *: soient U et V des polynômes unitaires à coefficients entiers dans [0, p[, p-premiers entre eux, de degrés respectifs m et $n > 0$. Alors pour tout polynôme P de degré $< m + n$, il existe des polynômes uniques A et B , à coefficients entiers dans [0, p[, de degrés respectifs $< n$ et $< m$, et un polynôme R de degré $< m + n$, tels que $P = A\,U + B\,V + p\,R$.*
  Il suffit en effet d'appliquer le résultat précédent, mais quand on finit le calcul, il n'est exact que modulo p. Il n'y a rien à changer à la procédure **Bezout4**, seule la fin de la vérification est différente.

```
> R := expand(P - A*U - B*V) / 7 ;
```

$$R := -X^4 - 3\,X^3 - 3\,X^2 - 6\,X - 7$$

- On énonce maintenant un résultat analogue modulo p^k. Cette fois l'anneau des polynômes à coefficients modulo p^k n'est plus principal, car il n'est même pas intègre. On se place donc dans une situation *où on suppose donnée* une relation de Bezout entre U et V.
  PROPOSITION B4 *Soient U et V deux polynômes unitaires à coefficients entiers dans [0, p^k[, de degrés respectifs m et n. On suppose donnée une relation de Bezout : $1 = A\,U + B\,V + p^k\,C$ où le degré de A (resp. B, C) est $< n$ (resp. $< m$, $< m + n$). Alors, pour tout polynôme P à coefficients entiers de degré $< m + n$, il existe des polynômes uniques E (resp. F, G), de degré $< n$ (resp. $< m$, $< m + n$), à coefficients entiers dans [0, p^k[, (resp. dans [0, p^k[, entiers), tels que $P = E\,U + F\,V + p^k\,G$.*
  DÉMONSTRATION. Analogue avec les adaptations évidentes. Prendre $E = P\,A$ et $F = P\,B$ est tentant, mais les degrés sont trop grands. Comme V est unitaire, on peut diviser $P\,A$ par V pour obtenir un reste à coefficients entiers E qu'on réduit modulo p^k : $P\,A = Q\,V + E + p^k\,R$, et de même on di-

vise $P\,B$ par U pour obtenir $P\,B = S\,U + F + p^k\,T$. En reportant dans $P = P\,A\,U + P\,B\,V + p^k\,P\,C$, il vient $P = E\,U + F\,V + (Q+S)\,U\,V + p^k\,(P\,C + R\,U + T\,V)$. La réduction modulo p^k de tous les termes laisse un polynôme $U\,V$ *unitaire* de degré $m+n$. Comme $P - E\,U - F\,V$ est de degré $< m+n$, il en résulte que le polynôme $Q+S$ est nul modulo p^k ; on peut donc «glisser» le terme $(Q+S)\,U\,V$ dans le facteur de p^k et finalement le degré résultant est nécessairement $< m+n$. Si on avait deux solutions différentes, on déduirait une relation $E\,U + F\,V = p^k\,R$; en multipliant par A, il vient $E\,A\,U + F\,A\,V = p^k\,A\,R$ ou, compte tenu de la relation presque-Bezout, $E - E\,B\,V + F\,A\,V = p^k\,(A\,R + E\,C)$. Mais si on réduit modulo p^k, E devient divisible par V, ce qui est incompatible avec V *unitaire* et degré de $E <$ degré de V , à moins que E soit nul. CQFD.

```
> Bezout5 := proc
> (p::posint, k::posint, P::polynom(integer, X),
> U::polynom(integer, X), V::polynom(integer, X),
> A::polynom(integer, X), B::polynom(integer, X))
> RETURN(Rem(P*A, V, X) mod p^k, Rem(P*B, U, X) mod p^k)
> end :
```

- On redéfinit **rndP** pour que l'entier modulaire soit libre.

```
> rndP := proc(p::posint, d::posint)
> local rnd ;
> rnd := rand(0..(p-1)) ;
> RETURN(sort(X^d + add(rnd()*X^i, i=0..(d-1))))
> end :
> _seed := 2210 :
> p, k := 7,2 ; U,V := seq(rndP(49,3), i=1..2) ;
```

$$p,\, k := 7,\, 2$$

$$U,\, V := X^3 + 43\,X^2 + 28\,X + 48,\; X^3 + 18\,X^2 + 12\,X + 24$$

```
> Gcdex(U, V, X, 'A', 'B') mod 49 ;
```

$$1$$

```
> sort(expand(1-A*U-B*V)/49) ;
```

$$-X^5 - 38\,X^4 - 63\,X^3 - 85\,X^2 - 56\,X - 23$$

```
> P := rndP(49, 5) ;
```

$$P := X^5 + 22\,X^4 + 26\,X^3 + 11\,X^2 + 39\,X + 4$$

```
> E, F := Bezout5(7, 2, P, U, V, A, B) ;
```

$$E,\, F := 43\,X^2 + 22\,X + 19,\; 7\,X^2 + 34\,X + 3$$

```
> expand(P-E*U-F*V)/49 ;
```

$$-X^5 - 41\,X^4 - 58\,X^3 - 84\,X^2 - 49\,X - 20$$

- Nous pouvons maintenant énoncer le *lemme de Hensel* pour les polynômes modulo une puissance de nombre premier.

PROPOSITION B5 (lemme de Hensel) *: On suppose qu'on dispose de deux relations :*

$$P = U\,V + p^k\,R;\quad 1 = A\,U + B\,V + p^k\,S;$$

où les degrés de U (resp. V, P, R, A, B, S) sont m (resp. n, $m+n$, $< m+n$, $< n$, $< m$, $< m+n$). Par ailleurs P, U et V sont supposés unitaires, et les coefficients de A, U, B et V sont des entiers dans $[0,\, p^k[$. Alors il existe U_0, V_0, R_0, A_0, U_0, B_0, V_0 et S_0 vérifiant les mêmes conditions à ceci près que p^k doit être remplacé par $p^{(2\,k)}$. De plus U_0, V_0, A_0 et B_0 sont uniques. Il en résulte la même propriété pour R_0 et S_0.

Démonstration. On pose $U_0 = U + p^k U_1$ et de même pour V, A et B. En reportant dans les équations à satisfaire et en réduisant modulo $p^{(2\,k)}$, il vient les équations suivantes.

$R = (V_1 U + U_1 V) \bmod p^k$

;

$S - A U_1 - B V_1 = (A_1 U + B_1 V) \bmod p^k$

. On va donc trouver les correctifs d'indice 1 par application de **Bezout5**. CQFD.

```
> Hensel := proc
> (p::posint, k::posint, P::polynom(integer, X),
> U::polynom(integer, X), V::polynom(integer, X),
> A::polynom(integer, X), B::polynom(integer, X))
> local R, S, U1, V1, A1, B1 ;
> R := expand(P - U*V)/p^k ;
> S := expand(1 - A*U - B*V)/p^k ;
> if not type(R, polynom(integer, X)) then
> ERROR("P, U, V not coherent for Hensel.")
> fi ;
> if not type(S, polynom(integer, X)) then
> ERROR("A, U, B, V not coherent for Hensel.")
> fi ;
> V1, U1 := Bezout5(p, k, R, U, V, A, B) ;
> A1, B1 := Bezout5(p, k, S-A*U1-B*V1, U, V, A, B) ;
> RETURN(p, 2*k, P, sort(U+p^k*U1), sort(V+p^k*V1),
> sort(A+p^k*A1), sort(B+p^k*B1))
> end :
```

- On a maintenant l'outil ad hoc pour augmenter très vite l'entier modulaire par rapport auquel on effectue une factorisation de polynômes à coefficients entiers. On trouve ainsi assez vite une factorisation éventuelle pour un polynôme à coefficients entiers, ou au contraire son irréductibilité.

```
> _seed := 1713 ;
> P := rndP(10, 10) ;
> RootsSup(P) ;
```

$$\_seed := 1713$$

$$P := X^{10} + 4\,X^9 + 8\,X^8 + 2\,X^7 + 9\,X^6 + 3\,X^5 + 4\,X^4 + X^3 + 9\,X^2 + 5$$

8.

```
> fs := Berl(11, P) ;
```

$$fs := [X^2 + 4\,X + 2,\ X^8 + 6\,X^6 + 8\,X^4 + 4\,X^3 + 5\,X^2 + 6\,X + 8]$$

- Deux facteurs, c'est la situation idéale pour Hensel. Il faut préparer les données.

```
> U, V := op(fs) ;
```

$$U,\ V := X^2 + 4\,X + 2,\ X^8 + 6\,X^6 + 8\,X^4 + 4\,X^3 + 5\,X^2 + 6\,X + 8$$

```
> Gcdex(U, V, X, 'A', 'B') mod 11 ;
```

$$1$$

```
> p,k,P,U,V,A,B := Hensel(11,1,P,U,V,A,B) : U ; V ;
```

$$X^2 + 92\,X + 68$$

$$X^8 + 33\,X^7 + 50\,X^6 + 55\,X^5 + 19\,X^4 + 81\,X^3 + 93\,X^2 + 94\,X + 41$$

- Le facteur de degré 2 n'est pas possible, car il devrait être $X^2 - 29\,X + ...$ mais ceci est incompatible avec la majoration par 8 des racines. Vérification.

```
> irreduc(P) ;
```

$$true$$

- Considérons comme plus haut un cas certainement factorisable.

```
> _seed := 16 :
> P := sort(expand(rndP(5,4)*rndP(5,6))) ;
> RootsSup(P) ;
```

$$P := X^{10} + 2\,X^8 + 4\,X^7 + 2\,X^6 + 12\,X^5 + 4\,X^4 + 16\,X^3 + 8\,X^2 + 8\,X + 8$$

$$3.287503660$$

```
> fs := Berl(11, P) ;
```

$$fs := [X + 4,\ X^4 + 2\,X^2 + 2,\ X^5 + 7\,X^4 + 5\,X^3 + 6\,X^2 + 9\,X + 1]$$

- Une racine est majorée par 3.3 et $X + 4$ ne peut pas venir d'un Z-facteur. Regroupons les deux premiers facteurs.

```
> U := Expand(fs[1]*fs[2]) mod 11 ; V := fs[3] ;
```

$$U := X^5 + 2\,X^3 + 2\,X + 4\,X^4 + 8\,X^2 + 8$$

$$V := X^5 + 7\,X^4 + 5\,X^3 + 6\,X^2 + 9\,X + 1$$

```
> Gcdex(U, V, X, 'A', 'B') mod 11;
```

$$1$$

```
> p,k,P,U,V,A,B := Hensel(11,1,P,U,V,A,B) : U ; V ;
```

$$X^5 + 70\,X^4 + 2\,X^3 + 19\,X^2 + 2\,X + 19$$

$$X^5 + 51\,X^4 + 60\,X^3 + 39\,X^2 + 53\,X + 45$$

- Mais les coefficients de X^4 sont impossibles. On essaie l'autre combinaison.

```
> U := Expand(fs[1]*fs[3]) mod 11 ; V := fs[2] ;
```

$$U := X^6 + 4\,X^3 + 4\,X + 4$$

$$V := X^4 + 2\,X^2 + 2$$

```
> Gcdex(U, V, X, 'A', 'B') mod 11;
```

$$1$$

```
> p,k,P,U,V,A,B := Hensel(11,1,P,U,V,A,B) : U ; V ;
```

$$X^6 + 4\,X^3 + 4\,X + 4$$

$$X^4 + 2\,X^2 + 2$$

- Cette fois les facteurs potentiels restent curieusement constants. Continuons.

```
> p,k,P,U,V,A,B := Hensel(11,2,P,U,V,A,B) : U ; V ;
```

$$X^6 + 4\,X^3 + 4\,X + 4$$

$$X^4 + 2\,X^2 + 2$$

```
> p,k,P,U,V,A,B := Hensel(11,4,P,U,V,A,B) : U ; V ;
```

$$X^6 + 4\,X^3 + 4\,X + 4$$

$$X^4 + 2\,X^2 + 2$$

- Cette fois il s'agit d'une factorisation certaine modulo :

```
> 11^8 ;
```

$$214358881$$

- On est donc certain d'avoir une vraie factorisation entière, qu'on aurait pu essayer plus vite. Vérification.

```
> factor(P) ;
```

$$(X^4 + 2\,X^2 + 2)\,(X^6 + 4\,X^3 + 4\,X + 4)$$

- Jouons à trouver ainsi des factorisations élevées de $X^4 + 1$;

```
> P := X^4+1 ;
```

$$P := X^4 + 1$$

```
> Berl(3, P) ;
```

$$[X^2 + 2\,X + 2,\ X^2 + X + 2]$$

```
> U,V := op(%) ;
> Gcdex(U,V,X,'A','B') mod 3 ;
> p,k := 3, 1 ;
```

$$U,\ V := X^2 + 2\,X + 2,\ X^2 + X + 2$$

$$1$$

$$p,\ k := 3,\ 1$$

```
> for i from 1 to 6 do
> p,k,P,U,V,A,B := Hensel(p,k,P,U,V,A,B)
> od :
> k, p^k ; U ; V ;
```

$$64,\ 34336838202925124846578490892 81$$

$$X^2 + 135295558823394433955461041579 2\,X + 343368382029251248465784908928 0$$

$$X^2 + 208072823205856814510323867348 9\,X + 343368382029251248465784908928 0$$

C Annexe : Bornes sur codes

```
> restart;
```

FONCTION D'ENTROPIE

```
> q:=7;
```

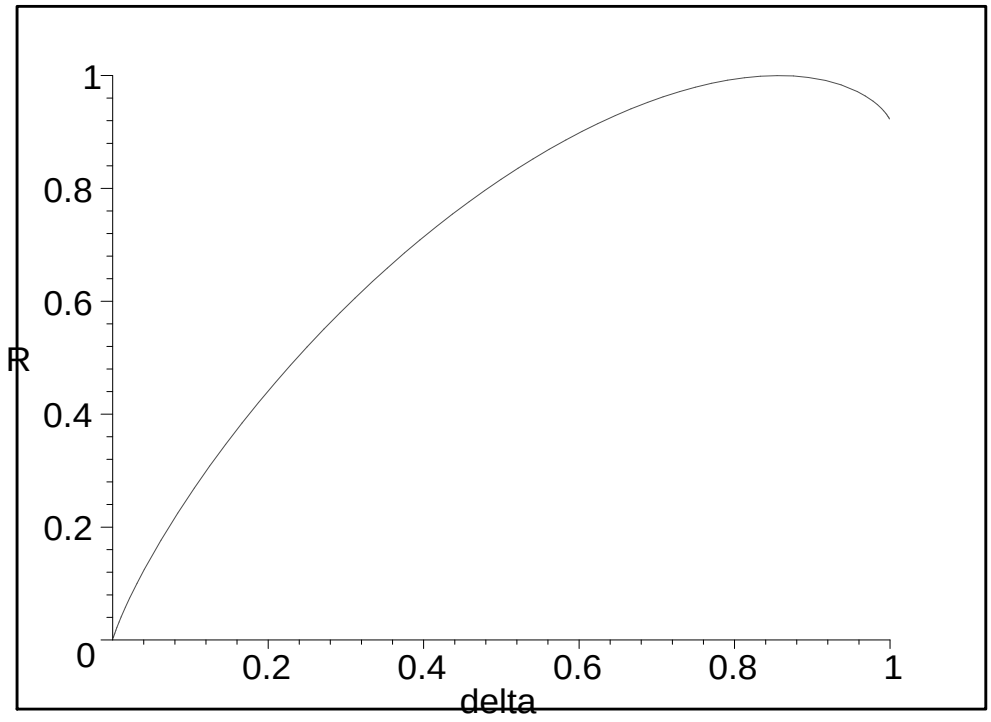
$$q := 7$$

```
> H[q](delta):=delta*log(q-1)/log(q)-
> delta*log(delta)/log(q)-(1-delta)*log(1-delta)/log(q);
> 'H[q](delta)=delta*log(q-1)/log(q)-
> delta*log(delta)/log(q)-(1-delta)*log(1-delta)/log(q)';
```

$$H_{25}(\delta) := \frac{\delta \ln(24)}{\ln(25)} - \frac{\delta \ln(\delta)}{\ln(25)} - \frac{(1-\delta)\ln(1-\delta)}{\ln(25)}$$

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

```
> plot([H[q](delta)], delta = 0..1, R = 0..1,discont=true);
> 'R=H[q](delta)';
```

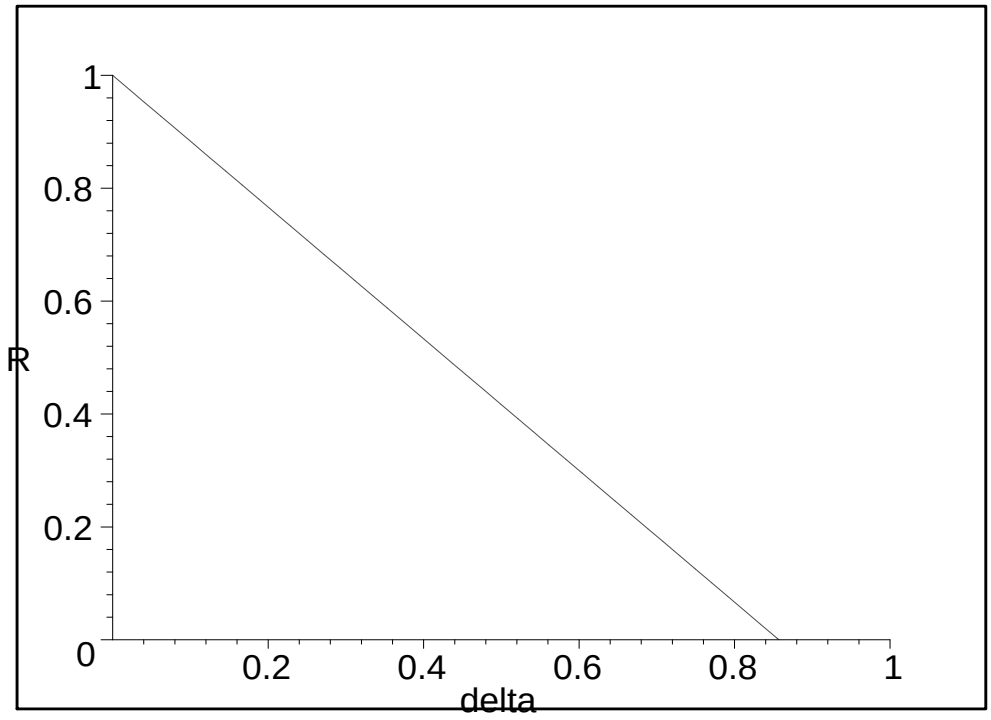


$$R = H_q(\delta)$$

BORNE DE PLOTKIN ASYMPTOTIQUE

```
> R=1-q*delta/(q-1);plot([1-q*delta/(q-1)],delta=0..1, R=0..1, discont
> = true);
```

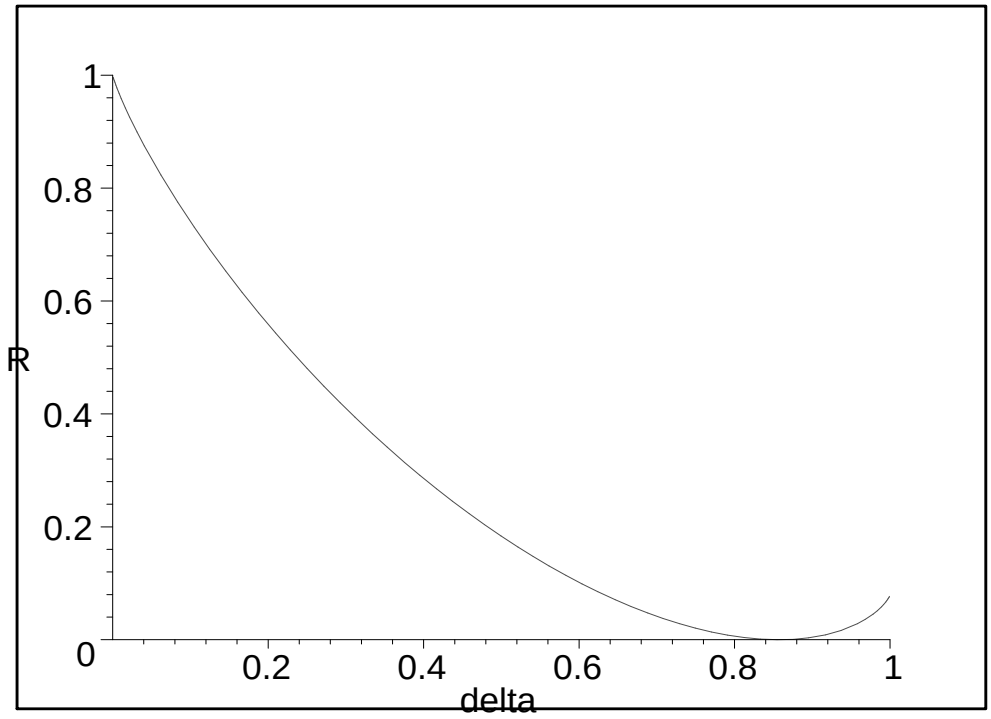
$$R = 1 - \frac{7}{6}\,\delta$$



BORNE DE GILBERT-VARSHAMOV

```
> R=1-H[q](delta);plot([1-H[q](delta)],delta=0..1, R=0..1);
```

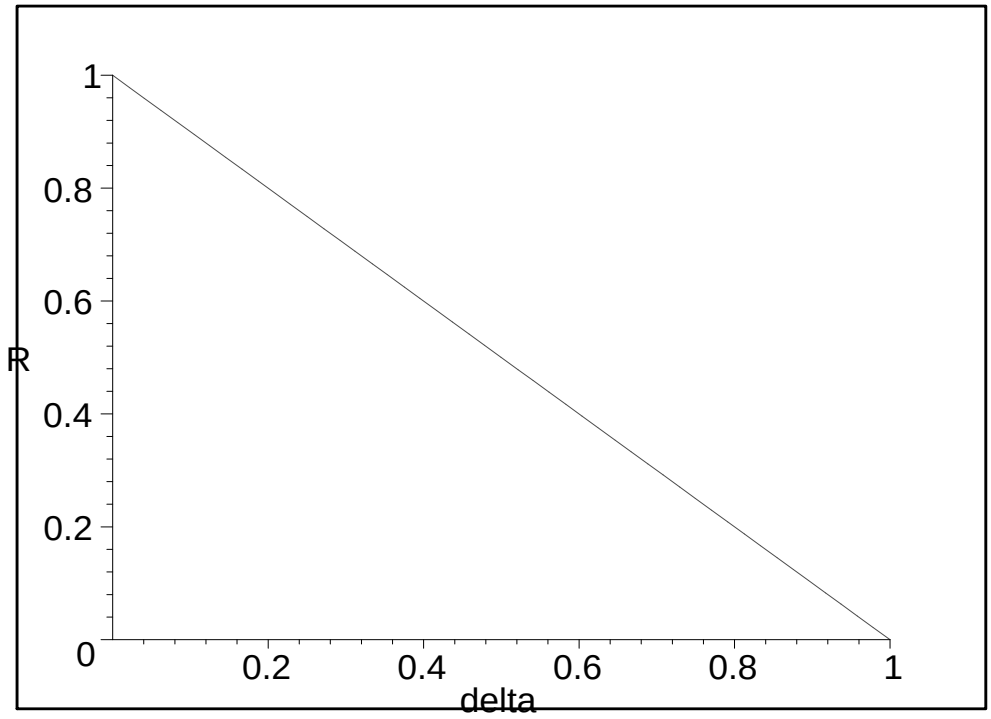
$$R = 1 - \frac{\delta \ln(6)}{\ln(7)} + \frac{\delta \ln(\delta)}{\ln(7)} + \frac{(1-\delta)\ln(1-\delta)}{\ln(7)}$$



BORNE DE SINGLETON ASYMPTOTIQUE

```
> R=1-delta;plot([1-delta],delta=0..1, R=0..1);
```

$$R = 1 - \delta$$



BORNE DE HAMMING (D'EMPILEMENT DE SPHÈRES) ASYMPTOTIQUE

```
> f:=H[q](delta);
```

$$f := \frac{\delta \ln(6)}{\ln(7)} - \frac{\delta \ln(\delta)}{\ln(7)} - \frac{(1-\delta)\ln(1-\delta)}{\ln(7)}$$

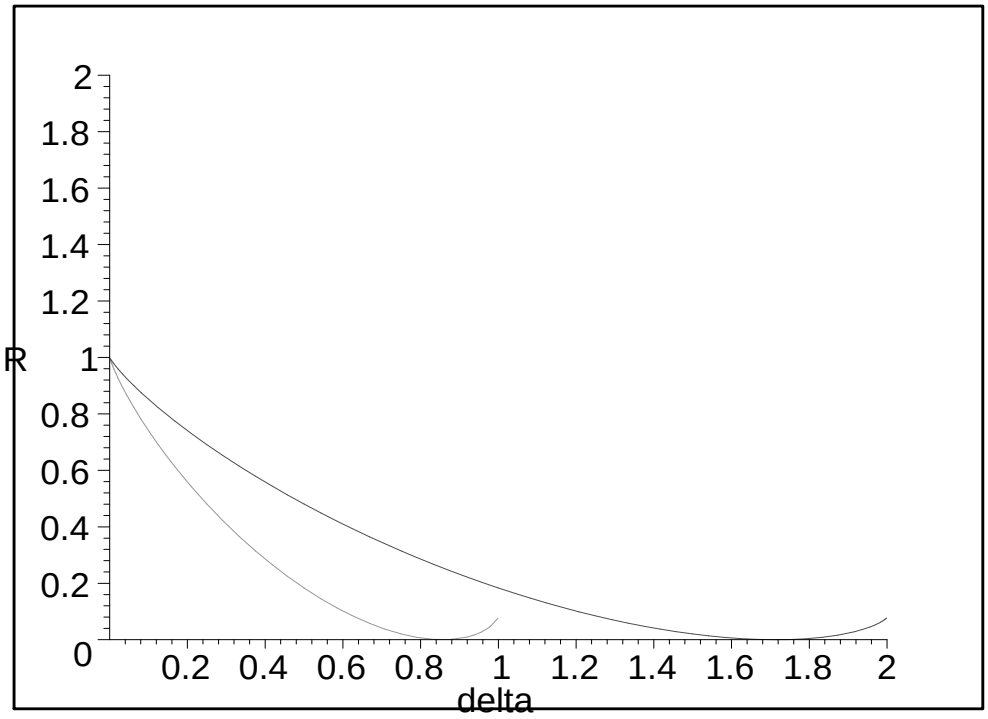
```
> g1:=delta/2;g:=algsubs(delta=g1,f);
> 'g=H[q](delta/2)';
```

$$g1 := \frac{1}{2}\,\delta$$

$$g := -\frac{\ln(1-\frac{1}{2}\,\delta)\,(1-\frac{1}{2}\,\delta)}{\ln(7)} + \frac{\frac{1}{2}\,\delta\,(-\ln(\frac{1}{2}\,\delta)+\ln(6))}{\ln(7)}$$

$$g = H_q(\frac{1}{2}\,\delta)$$

```
> plot([1-g(delta),1-f], delta=0..2,R=0..2);
> 'R=1-H[q](delta/2) ,1-H[q](delta)';
```



$$R = 1 - H_q(\frac{1}{2}\,\delta),\ 1 - H_q(\delta)$$

BORNE DE BASSALYGO-ELIAS ASYMPTOTIQUE

```
> h:=(q-1)/q-((q-1)/q)*(1-(q*delta/(q-1)))^(1/2);
> R[BE]:=1-algsubs(delta=h,f);
> 'R[BE]= 1-H[q]((q-1)/q-((q-1)/q)*((1-(q*delta/(q-1)))^(1/2)))';
```

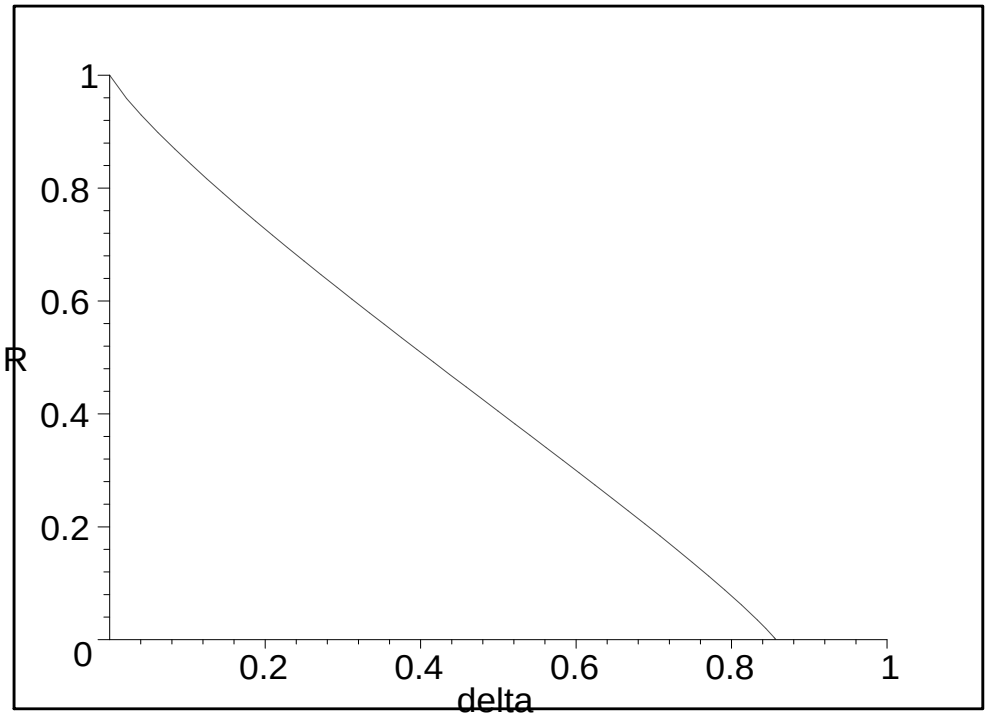
$$h := \frac{6}{7} - \frac{6}{7}\sqrt{1 - \frac{7}{6}\,\delta}$$

$$R_{BE} := 1 + \frac{\ln(\frac{1}{7} + \frac{1}{7}\,\%1)\,(\frac{1}{7} + \frac{1}{7}\,\%1)}{\ln(7)} + \frac{\frac{1}{7}\,(-6 + \%1)\,(-\ln(\frac{6}{7} - \frac{1}{7}\,\%1) + \ln(6))}{\ln(7)}$$

$$\%1 := \sqrt{36 - 42\,\delta}$$

$$R_{BE} = 1 - H_q\left(\frac{q-1}{q} - \frac{(q-1)\sqrt{1 - \frac{q\,\delta}{q-1}}}{q}\right)$$

```
> plot([R[BE]],delta=0..1,R=0..1);
> 'R=R[BE](delta)';
```



$$R = R_{BE}(\delta)$$

BORNE DE LA GEOMETRIE ALGEBRIQUE

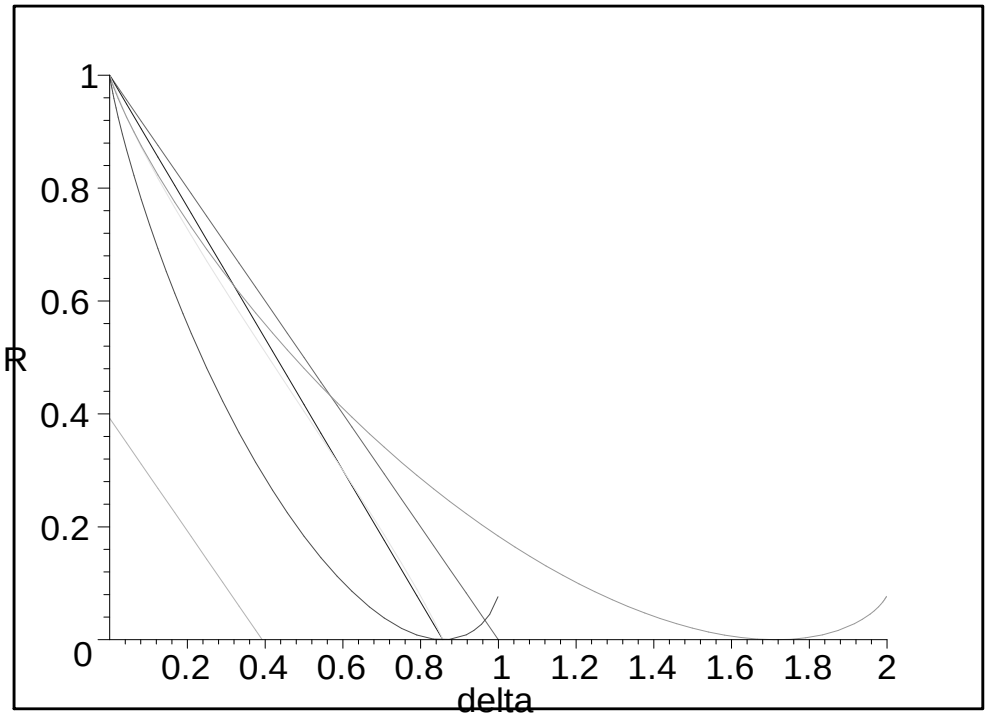
```
> R[GA]:=1-delta-(sqrt(q)-1)^(-1);
> 'R[GA]=1-delta-(sqrt(q)-1)^(-1)';
```

$$R_{GA} := 1 - \delta - \frac{1}{\sqrt{7} - 1}$$

$$R_{GA} = 1 - \delta - \frac{1}{\sqrt{q} - 1}$$

TOUTES LES BORNES

```
> plot([1-f,1-g,R[BE],1-q*delta/(q-1),
> 1-delta,R[GA]],delta=0..2,R=0..1);
> 'R=1-H[q](delta),1-H[q](delta/2),R[BE](delta),
> 1-q*delta/(q-1), 1-delta, R[GA](delta)';
```



$$R = 1 - H_q(\delta),\ 1 - H_q(\frac{1}{2}\,\delta),\ R_{BE}(\delta),\ 1 - \frac{q\,\delta}{q-1},\ 1 - \delta,\ R_{GA}(\delta)$$

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
> restart;
```

FONCTION D'ENTROPIE

```
> q:=49;
```

$$q := 49$$

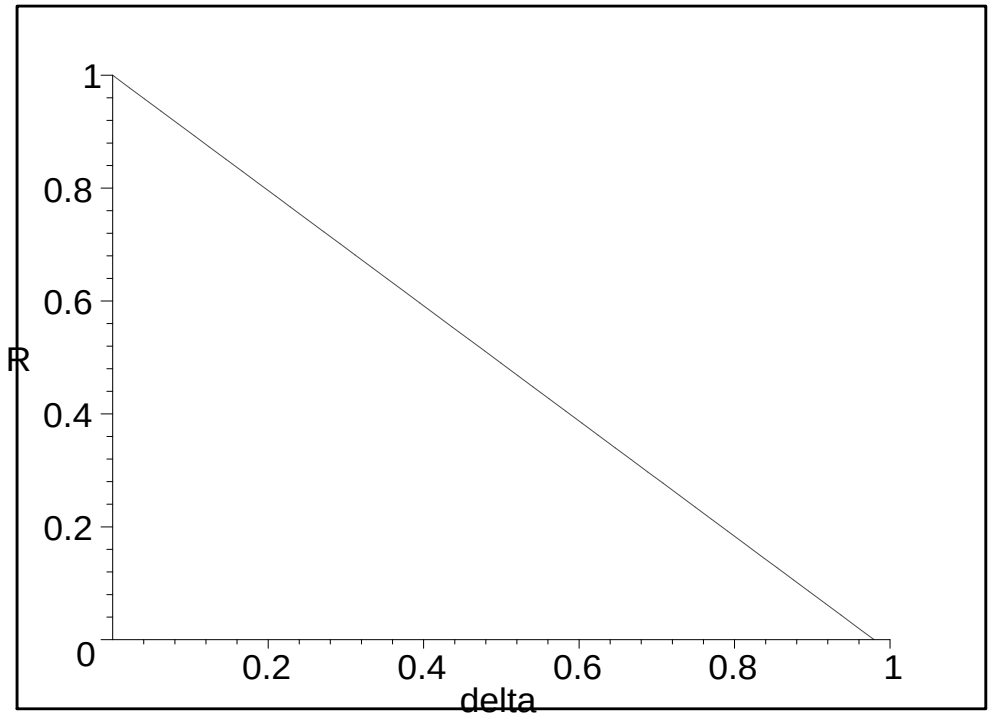
```
> H[q](delta):=delta*log(q-1)/log(q)-
> delta*log(delta)/log(q)-(1-delta)*log(1-delta)/log(q);
> 'H[q](delta)=delta*log(q-1)/log(q)-
> delta*log(delta)/log(q)-(1-delta)*log(1-delta)/log(q)';
```

$$H_{49}(\delta) := \frac{\delta \ln(48)}{\ln(49)} - \frac{\delta \ln(\delta)}{\ln(49)} - \frac{(1-\delta)\ln(1-\delta)}{\ln(49)}$$

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

BORNE DE PLOTKIN ASYMPTOTIQUE

```
> plot([1-q*delta/(q-1)],delta=0..1, R=0..1, discont = true);
> 'R=1-q*delta/(q-1)';
```

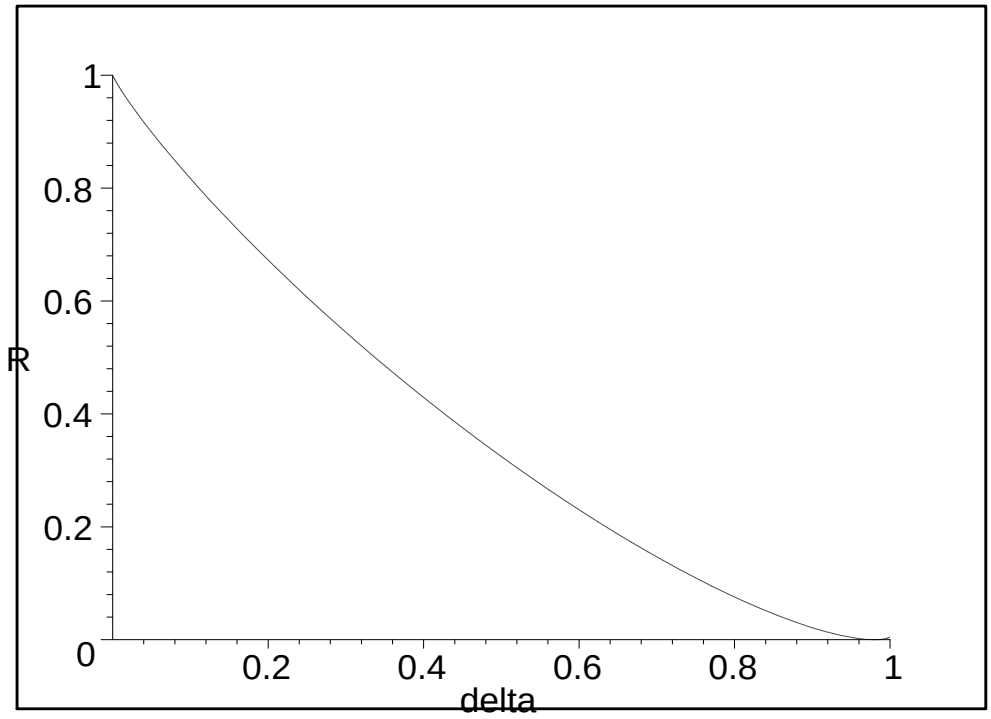


$$R = 1 - \frac{q\,\delta}{q-1}$$

BORNE DE GILBERT-VARSHAMOV

```
> 'R=1-H[q](delta)';
> plot([1-H[q](delta)],delta=0..1, R=0..1, discont = true);
```

$$R = 1 - H_q(\delta)$$




```
\mapleinline{active}{1d}{h:=(q-1)/q-((q-1)/q)*(1-(q*delta/(q-1)))^(1/2);}{%
}
```

$$h := \frac{48}{49} - \frac{48}{49}\sqrt{1 - \frac{49}{48}\,\delta}$$

BORNE DE HAMMING (D'EMPILEMENT DE SPHÈRES) ASYMPTOTIQUE

```
> f:=H[q](delta);
> 'f=H[q](delta)';
```

$$f := \frac{\delta \ln(48)}{\ln(49)} - \frac{\delta \ln(\delta)}{\ln(49)} - \frac{(1-\delta)\ln(1-\delta)}{\ln(49)}$$

$$f = H_q(\delta)$$

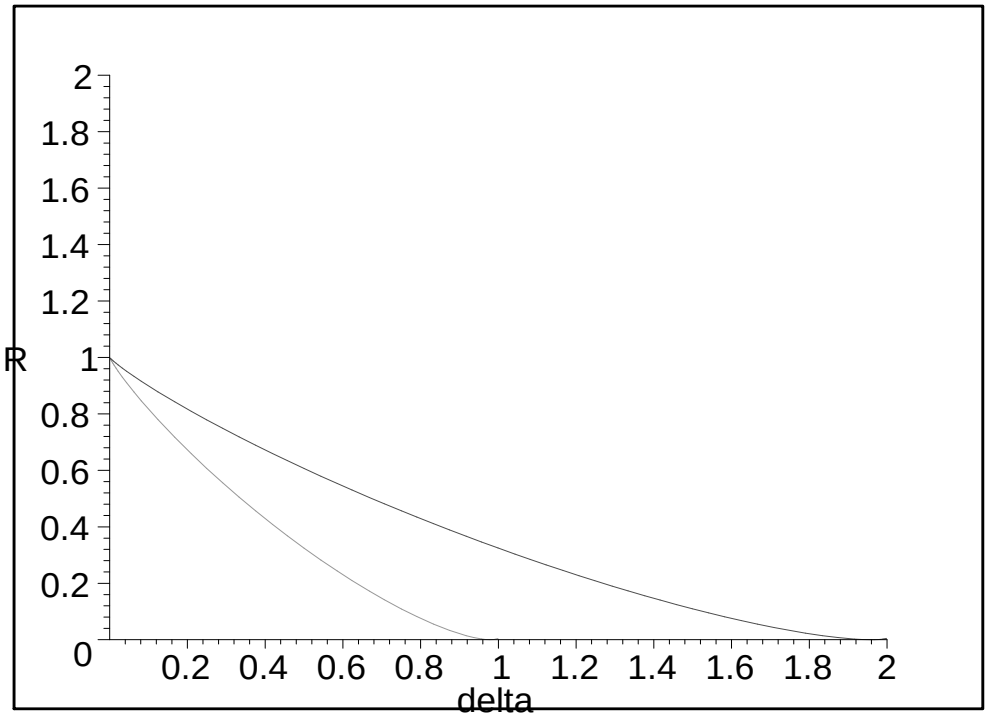
```
> g1:=delta/2;g:=algsubs(delta=g1,f);
> 'g=H[q](delta/2)';
```

$$g1 := \frac{1}{2}\,\delta$$

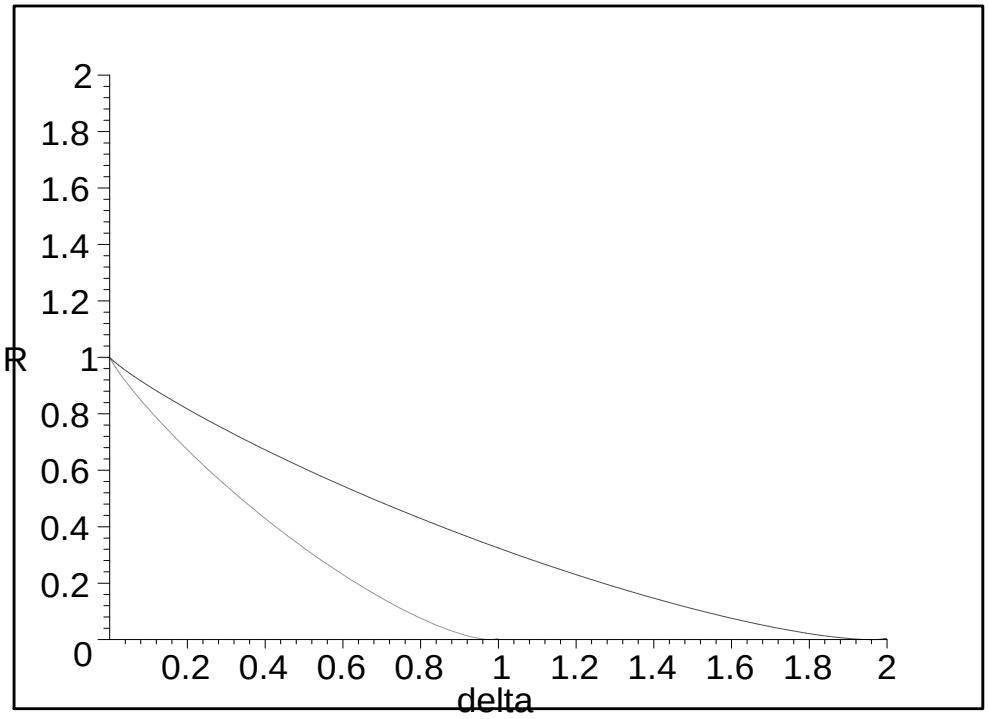
$$g := -\frac{\ln(1-\frac{1}{2}\,\delta)\,(1-\frac{1}{2}\,\delta)}{\ln(49)} - \frac{1}{2}\,\frac{\delta\,(\ln(\frac{1}{2}\,\delta) - \ln(48))}{\ln(49)}$$

$$g = H_q(\frac{1}{2}\,\delta)$$

```
> plot([1-g(delta),1-f], delta=0..2,R=0..2);
```




```
>  plot([1-g(delta),1-f], delta=0..2,R=0..2);
>  'R=1-H[q](delta/2) ,1-H[q](delta)';
```



$$R = 1 - H_q(\frac{1}{2}\,\delta),\ 1 - H_q(\delta)$$

```
> h ;R[BE]:=1-algsubs(delta=h,f);
> 'R[BE]= 1-H[q]((q-1)/q-((q-1)/q)*(1-(q*delta/(q-1)))^(1/2))';
```

$$\frac{48}{49} - \frac{48}{49}\sqrt{1 - \frac{49}{48}\,\delta}$$

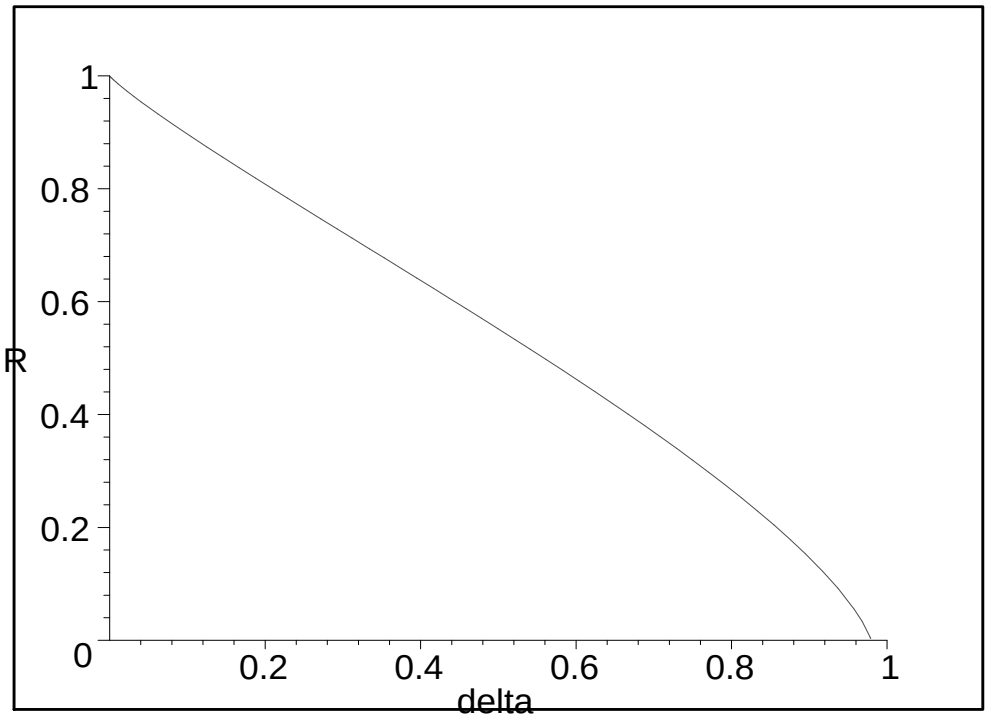
$$R_{BE} := 1 + \frac{\ln(\frac{1}{49} + \frac{4}{49}\,\%1)\,(\frac{1}{49} + \frac{4}{49}\,\%1)}{\ln(49)} + \frac{\frac{4}{49}\,(-12 + \%1)\,(-\ln(\frac{48}{49} - \frac{4}{49}\,\%1) + \ln(48))}{\ln(49)}$$

$$\%1 := \sqrt{144 - 147\,\delta}$$

$$R_{BE} = 1 - H_q\left(\frac{q-1}{q} - \frac{(q-1)\sqrt{1 - \frac{q\,\delta}{q-1}}}{q}\right)$$

BORNE DE BASSALYGO-ELIAS ASYMPTOTIQUE

```
> plot([R[BE](delta)],delta=0..1,R=0..1);
> 'R=R[BE](delta)';
```



$$R = R_{BE}(\delta)$$

BORNE DE LA GEOMETRIE ALGEBRIQUE

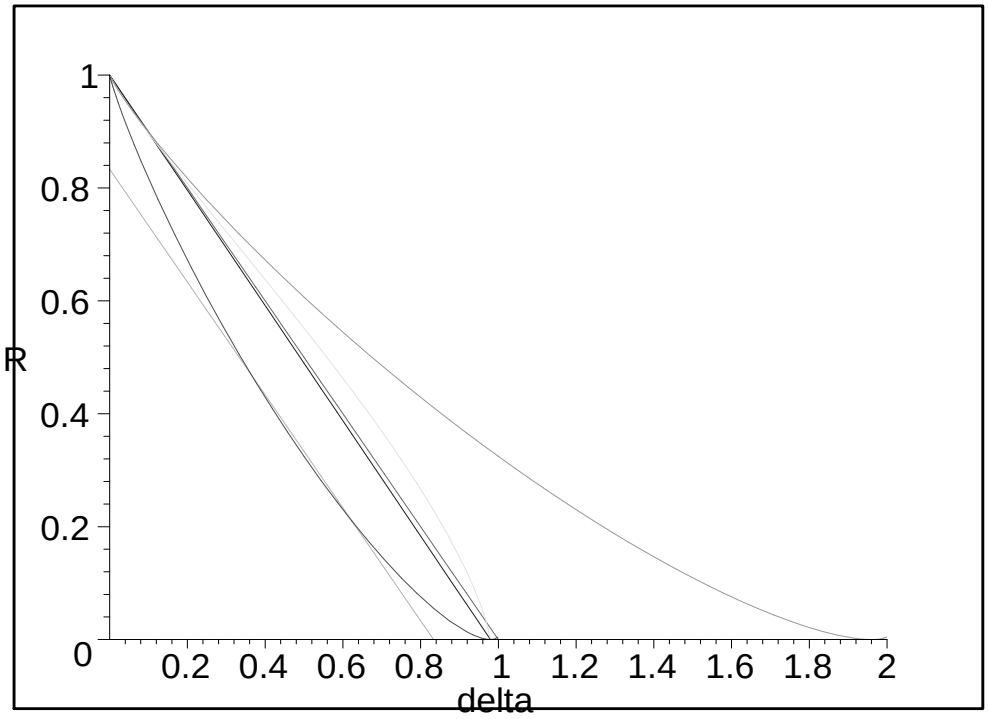
```
> R[GA]:=1-delta-(sqrt(q)-1)^(-1);
> 'R[GA]=1-delta-(sqrt(q)-1)^(-1)';
```

$$R_{GA} := \frac{5}{6} - \delta$$

$$R_{GA} = 1 - \delta - \frac{1}{\sqrt{q} - 1}$$

TOUTES LES BORNES

```
> plot([1-f,1-g,R[BE],1-q*delta/(q-1),
> 1-delta,R[GA]],delta=0..2,R=0..1);
```




```
> restart;
```

FONCTION D'ENTROPIE

```
> q:=25;
```

$$q := 25$$

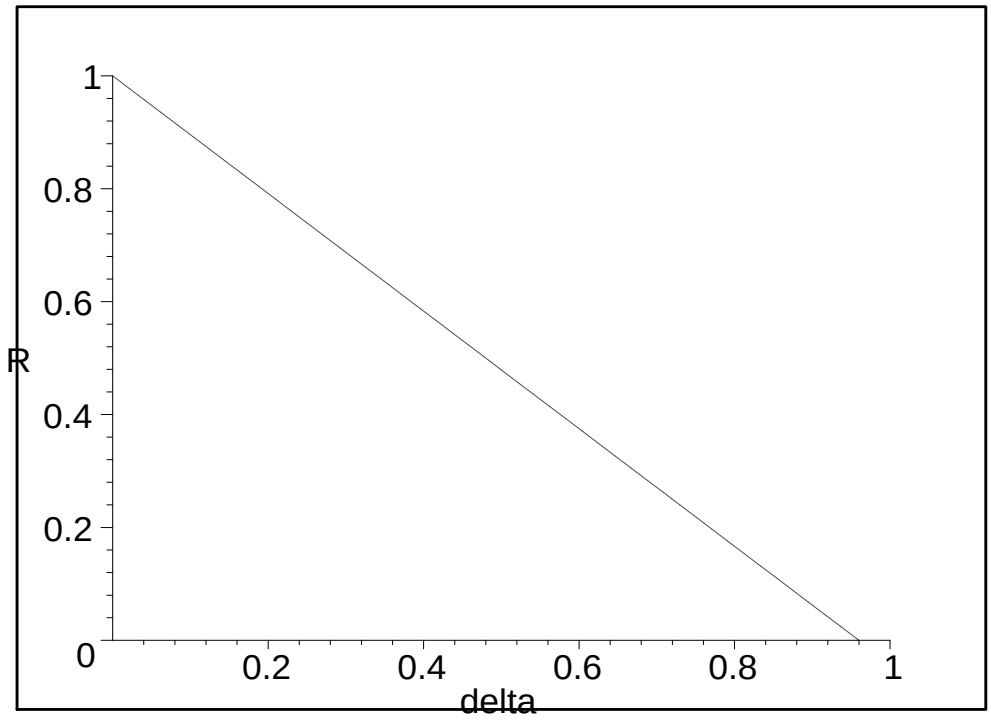
```
> H[q](delta):=delta*log(q-1)/log(q)-
> delta*log(delta)/log(q)-(1-delta)*log(1-delta)/log(q);
> 'H[q](delta)=delta*log(q-1)/log(q)-
> delta*log(delta)/log(q)-(1-delta)*log(1-delta)/log(q)';
```

$$H_{25}(\delta) := \frac{\delta \ln(24)}{\ln(25)} - \frac{\delta \ln(\delta)}{\ln(25)} - \frac{(1-\delta)\ln(1-\delta)}{\ln(25)}$$

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

BORNE DE PLOTKIN ASYMPTOTIQUE

```
> plot([1-q*delta/(q-1)],delta=0..1, R=0..1, discont = true);
> 'R=1-q*delta/(q-1)';
```

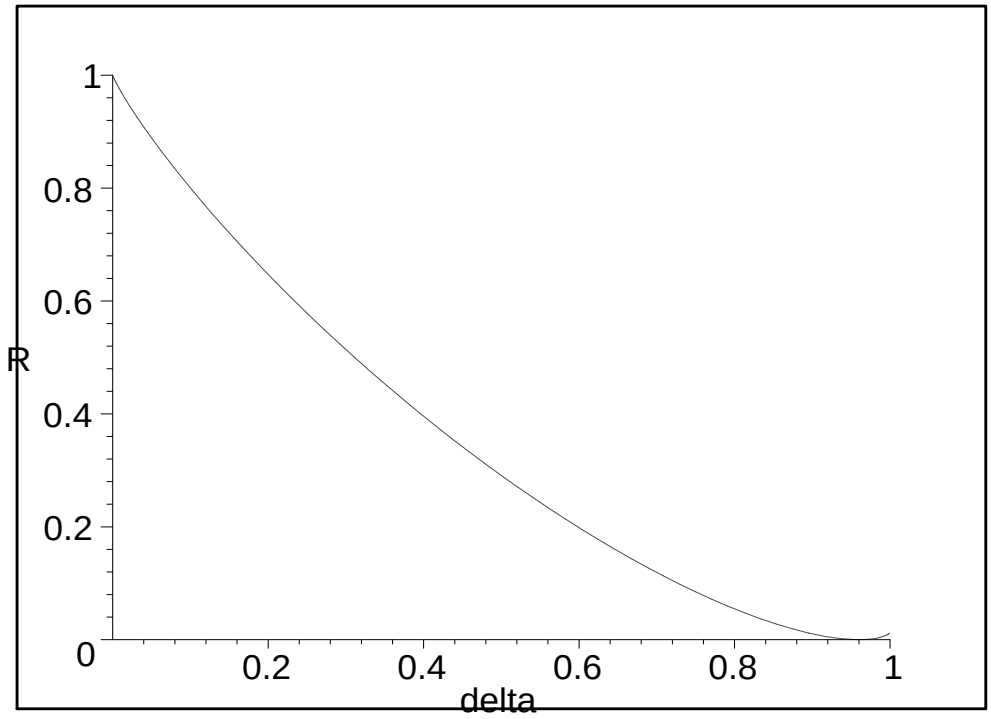


$$R = 1 - \frac{q\,\delta}{q-1}$$

BORNE DE GILBERT-VARSHAMOV

```
> 'R=1-H[q](delta)';
> plot([1-H[q](delta)],delta=0..1, R=0..1, discont = true);
```

$$R = 1 - H_q(\delta)$$




```
\mapleinline{active}{1d}{h:=(q-1)/q-((q-1)/q)*(1-(q*delta/(q-1)))^(1/2);}{%
}
```

$$h := \frac{24}{25} - \frac{24}{25}\sqrt{1 - \frac{25}{24}\,\delta}$$

BORNE DE HAMMING (D'EMPILEMENT DE SPHÈRES) ASYMPTOTIQUE

```
> f:=H[q](delta);
> 'f=H[q](delta)';
```

$$f := \frac{\delta \ln(24)}{\ln(25)} - \frac{\delta \ln(\delta)}{\ln(25)} - \frac{(1-\delta)\ln(1-\delta)}{\ln(25)}$$

$$f = H_q(\delta)$$

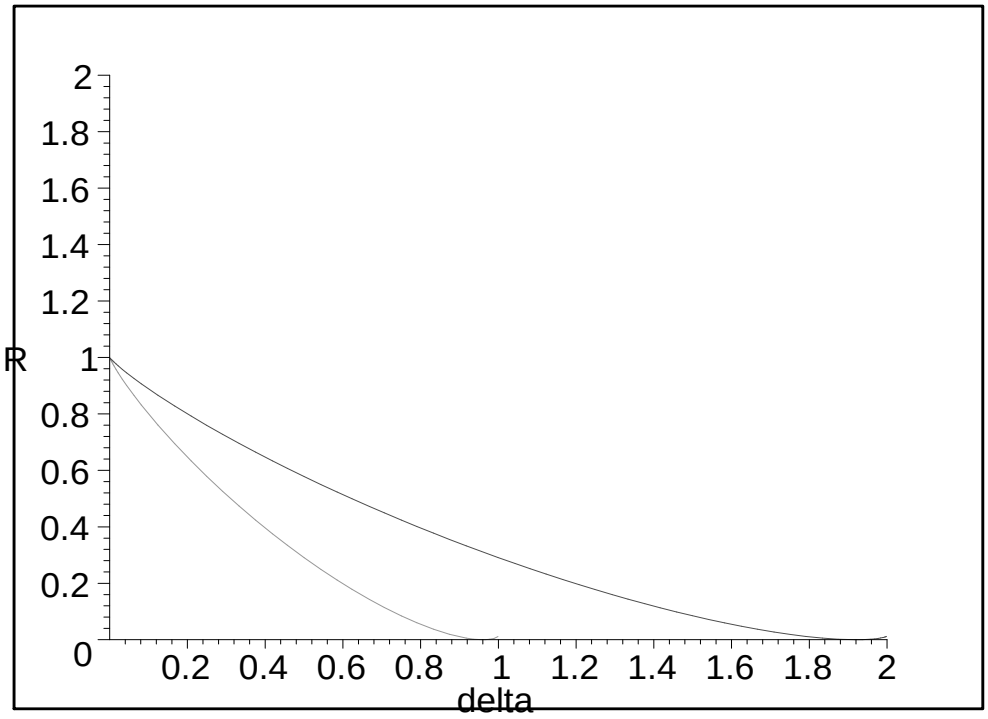
```
> g1:=delta/2;g:=algsubs(delta=g1,f);
> 'g=H[q](delta/2)';
```

$$g1 := \frac{1}{2}\,\delta$$

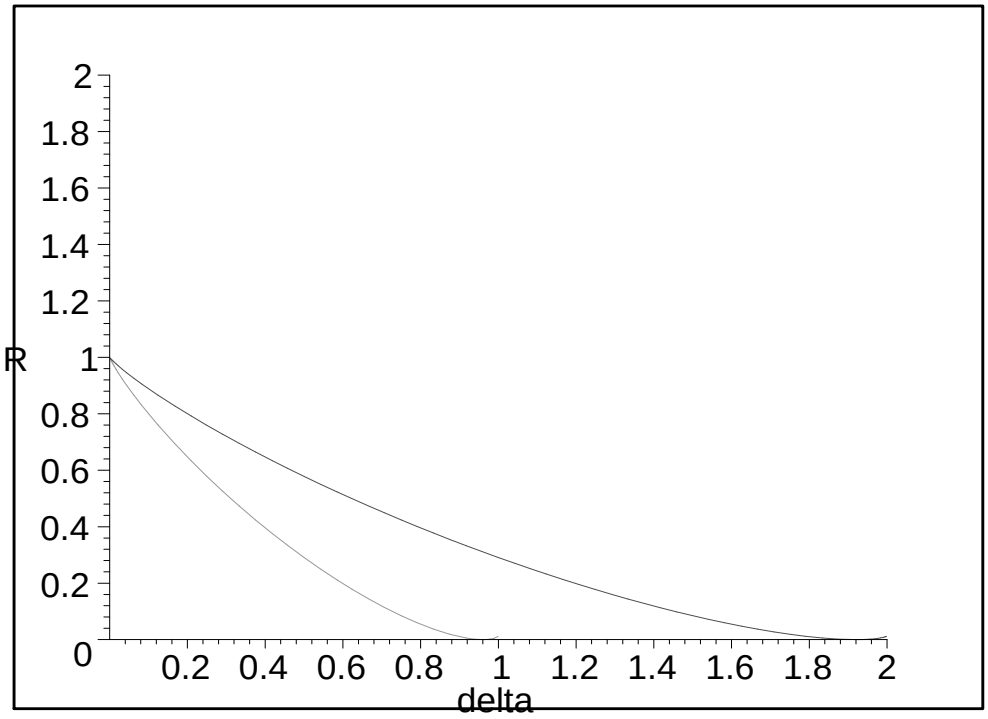
$$g := -\frac{\ln(1-\frac{1}{2}\,\delta)\,(1-\frac{1}{2}\,\delta)}{\ln(25)} - \frac{1}{2}\,\frac{\delta\,(\ln(\frac{1}{2}\,\delta) - \ln(24))}{\ln(25)}$$

$$g = H_q(\frac{1}{2}\,\delta)$$

```
> plot([1-g(delta),1-f], delta=0..2,R=0..2);
```




```
>  plot([1-g(delta),1-f], delta=0..2,R=0..2);
>  'R=1-H[q](delta/2) ,1-H[q](delta)';
```



$$R = 1 - H_q(\frac{1}{2}\,\delta),\ 1 - H_q(\delta)$$

```
> h ;R[BE]:=1-algsubs(delta=h,f);
> 'R[BE]= 1-H[q]((q-1)/q-((q-1)/q)*(1-(q*delta/(q-1)))^(1/2))';
```

$$\frac{24}{25} - \frac{24}{25}\sqrt{1 - \frac{25}{24}\,\delta}$$

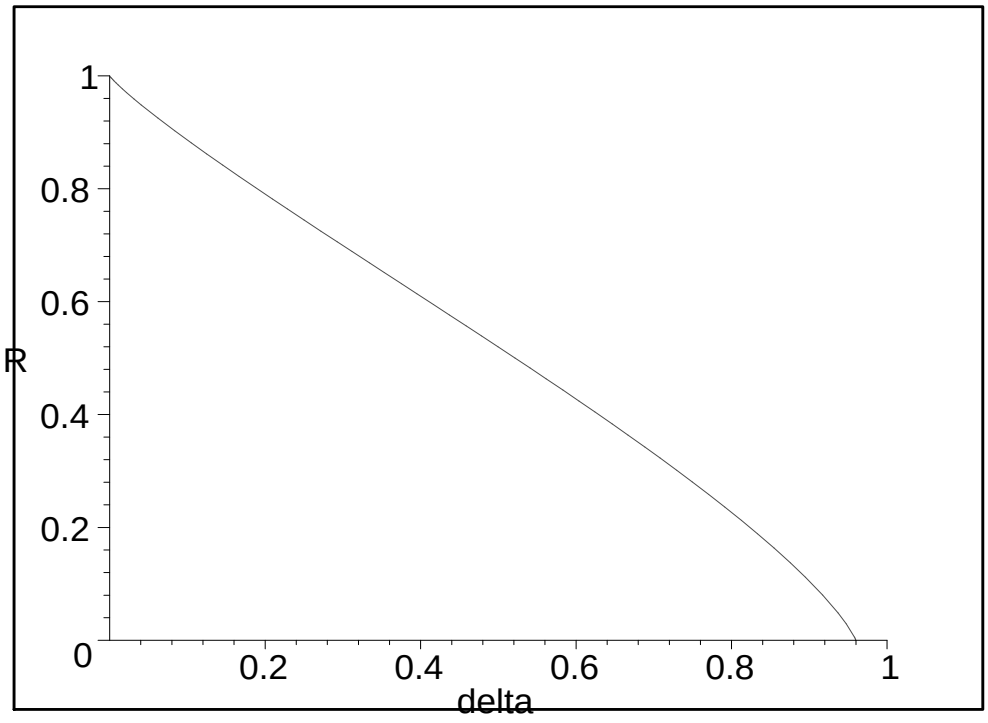
$$R_{BE} := 1 + \frac{\ln(\frac{1}{25} + \frac{2}{25}\,\%1)\,(\frac{1}{25} + \frac{2}{25}\,\%1)}{\ln(25)} + \frac{\frac{2}{25}\,(-12 + \%1)\,(-\ln(\frac{24}{25} - \frac{2}{25}\,\%1) + \ln(24))}{\ln(25)}$$

$$\%1 := \sqrt{144 - 150\,\delta}$$

$$R_{BE} = 1 - H_q\left(\frac{q-1}{q} - \frac{(q-1)\sqrt{1 - \frac{q\,\delta}{q-1}}}{q}\right)$$

BORNE DE BASSALYGO-ELIAS ASYMPTOTIQUE

```
> plot([R[BE](delta)],delta=0..1,R=0..1);
> 'R=R[BE](delta)';
```



$$R = R_{BE}(\delta)$$

BORNE DE LA GEOMETRIE ALGEBRIQUE

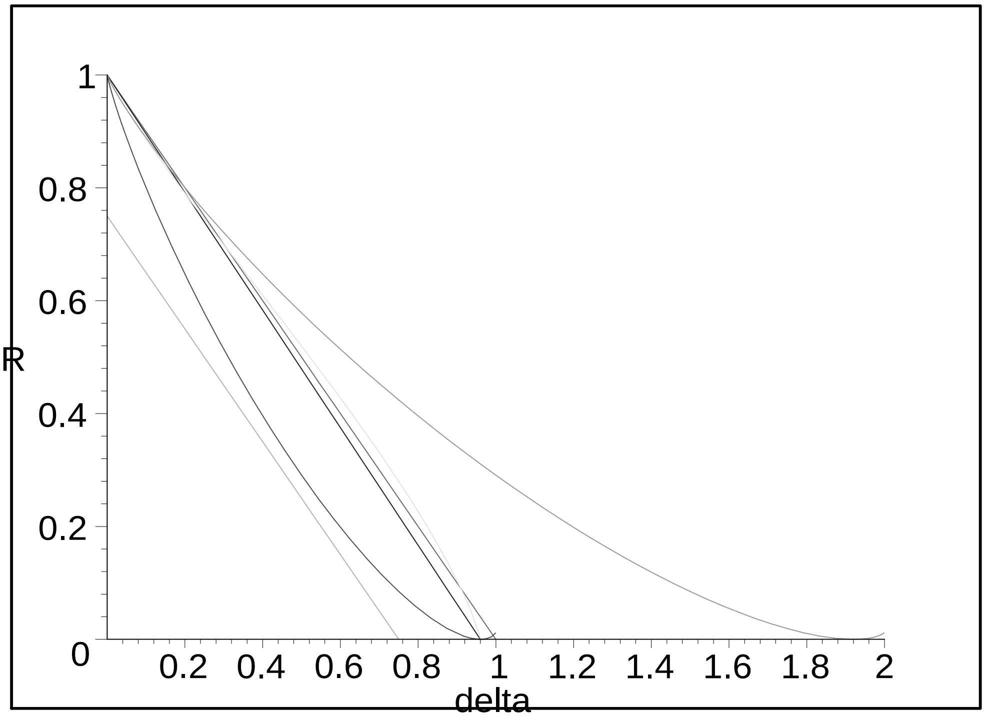
```
> R[GA]:=1-delta-(sqrt(q)-1)^(-1);
> 'R[GA]=1-delta-(sqrt(q)-1)^(-1)';
```

$$R_{GA} := \frac{3}{4} - \delta$$

$$R_{GA} = 1 - \delta - \frac{1}{\sqrt{q} - 1}$$

TOUTES LES BORNES

```
> plot([1-f,1-g,R[BE],1-q*delta/(q-1),
> 1-delta,R[GA]],delta=0..2,R=0..1);
```



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

D Annexe : Codes cycliques. Codes de Golay

D1. Exemple de correction de deux erreurs par un code cyclique

```
.
> restart;
> restart ;
> with(linalg) :
Warning, the protected names norm and trace have been redefined and
unprotected
> q:=2;
```

$$q := 2$$

```
> g1:= x^4+x^3+1;
```

$$g1 := x^4 + x^3 + 1$$

```
> g:= x^4+x+1;
```

$$g := x^4 + x + 1$$

```
> g2:=x^4+x^3+x^2+x+1;
```

$$g2 := x^4 + x^3 + x^2 + x + 1$$

- Pour travailler *sous Maple* dans des extensions de Z_p, on procède comme pour les extensions de Q, mais il n'y a ***aucune différence*** dans l'usage initial de **RootOf**, c'est seulement *en fin de calcul* qu'on précise qu'on veut travailler dans Z_p et ses extensions, en *suffixant* par «**mod p**».

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

$$\alpha$$

- Le quotient $Z_2[\,X]/(\mathrm{g})$ est un espace vectoriel de degré 4 sur Z_2, dont les éléments sont tous de la forme $i + j\,\alpha + k\,\alpha^2 + l\,\alpha^3$ pour i, j, k et l parcourant Z_2.

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

$$(x + \alpha^2)\,(x + \alpha + 1)\,(x + \alpha^2 + 1)\,(x + \alpha)$$

```
> s:=1+(alpha^2+alpha^3)*x+(1+alpha+alpha^3)*x^2;
```

$$s := 1 + (\alpha^2 + \alpha^3)\,x + (1 + \alpha + \alpha^3)\,x^2$$

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

$$(1 + \alpha + \alpha^3)\,(x + \alpha^3 + \alpha + 1)\,(x + \alpha)$$

```
> Expand(alpha^7) mod 2;
```

$$1 + \alpha + \alpha^3$$

```
> for i from 0 to 15 do
> if Eval(s, x=alpha^i) mod 2 = 0 then print(i) fi
> od ;
```

$$1$$
$$7$$

D2. Codes de Golay

```
> 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 ;
```

$$i=1,\ \alpha=\alpha,\ \mathrm{g}(\alpha)=0$$
$$i=2,\ \alpha^2=\alpha^2,\ \mathrm{g}(\alpha^2)=0$$
$$i=3,\ \alpha^3=\alpha^3,\ \mathrm{g}(\alpha^3)=0$$
$$i=4,\ \alpha^4=\alpha^4,\ \mathrm{g}(\alpha^4)=0$$
$$i=6,\ \alpha^6=\alpha^6,\ \mathrm{g}(\alpha^6)=0$$
$$i=8,\ \alpha^8=\alpha^8,\ \mathrm{g}(\alpha^8)=0$$
$$i=9,\ \alpha^9=\alpha^9,\ \mathrm{g}(\alpha^9)=0$$
$$i=12,\ \alpha^{12}=\alpha^{10}+\alpha^7+\alpha^4+\alpha^3+\alpha^2+\alpha+1,\ \mathrm{g}(\alpha^{12})=0$$
$$i=13,\ \alpha^{13}=\alpha^{10}+\alpha^8+\alpha^6+\alpha^3+\alpha+1,\ \mathrm{g}(\alpha^{13})=0$$
$$i=16,\ \alpha^{16}=\alpha^9+\alpha^8+\alpha^6+\alpha^5+\alpha^2+\alpha,\ \mathrm{g}(\alpha^{16})=0$$
$$i=18,\ \alpha^{18}=\alpha^8+\alpha^7+\alpha^6+\alpha^5+\alpha^3+\alpha^2+1,\ \mathrm{g}(\alpha^{18})=0$$

```
> 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 ;
```

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

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

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

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

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

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

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

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

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

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

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

```
> g[1]:=x^11+x^9+x^7+x^6+x^5+x+1;Factor(g[1],alpha) mod 2;
```

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

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

Code de Golay C_{23} = (g) est un sous-espace vectoriel de dimension 12 dans le quotient $Z_2[X]/(x^{23} - 1)$ vu comme un espace vectoriel de dimension 23 sur Z_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],
> [0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0],
> [0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0],
> [0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0],
> [0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0],
> [0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0,0],
> [0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0,0],
> [0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,0],
> [0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0],
> [0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0],
> [0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0],
> [0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,0,0,0,1,1]]) ;
```

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

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

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

```
> (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);
```

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

```
> '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;
```

$$1+x^{23}$$

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

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

```
> 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 $\mathrm{GF}(2^{11}) = F_2[\alpha]$. Pour obtenir la réduction.

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

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

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

<u>Une autre matrice de contrôle</u> :
on considère la matrice dont la j-me colonne est formée par les coordonnées de $\alpha^{j-1}(j = 1, 2, ..., 23)$ dans la base

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

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

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

$$1 = 1$$
$$\alpha = \alpha$$
$$\alpha^2 = \alpha^2$$
$$\alpha^3 = \alpha^3$$
$$\alpha^4 = \alpha^4$$
$$\alpha^5 = \alpha^5$$
$$\alpha^6 = \alpha^6$$
$$\alpha^7 = \alpha^7$$
$$\alpha^8 = \alpha^8$$
$$\alpha^9 = \alpha^9$$
$$\alpha^{10} = \alpha^{10}$$
$$\alpha^{11} = \alpha^{10} + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1$$
$$\alpha^{12} = \alpha^{10} + \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$$
$$\alpha^{13} = \alpha^{10} + \alpha^8 + \alpha^6 + \alpha^3 + \alpha + 1$$
$$\alpha^{14} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1$$
$$\alpha^{15} = \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1$$
$$\alpha^{16} = \alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha$$
$$\alpha^{17} = \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^6 + \alpha^3 + \alpha^2$$
$$\alpha^{18} = \alpha^8 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + 1$$
$$\alpha^{19} = \alpha^9 + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha$$
$$\alpha^{20} = \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha^2$$
$$\alpha^{21} = \alpha^9 + \alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1$$
$$\alpha^{22} = \alpha^{10} + \alpha^9 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha$$

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

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

```
> 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 $\mathrm{GF}(2^{11}) = F_2[\alpha]$. Pour obtenir la réduction.

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

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

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%

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

$$q := 3$$

```
> restart ;

> with(linalg) :

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

> Factor(x^11-1) mod 3;
```

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

```
> g:=x^5+2*x^3+x^2+2*x+2;irreduc(g) mod 3;
```

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

$$\mathit{true}$$

```
> 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^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 ;
```

$$i=1,\ \alpha=\alpha,\ \mathrm{g}(\alpha)=0$$

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

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

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

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

```
> 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 ;
```

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

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

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

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

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

```
> (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);
```

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

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

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

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

$g := x^5 + 2\,x^3 + x^2 + 2\,x + 2$
et avec le polynome de contrôle $h = (x + 2)\,(x^5 + x^4 + 2\,x^3 + x^2 + 2)$,
$h := x^6 + x^4 + 2\,x^3 + 2\,x^2 + 2\,x + 1$.

C'est un [12,6, 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;
```

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

```
> 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 $\mathrm{GF}(3^5) = F_3[\alpha]$. Pour obtenir la réduction.

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

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

E Annexe : Points des courbes algébriques sur les corps finis

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

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;
```

$$f := (x,\, y,\, z) \to x^3 + y^3 + z^3$$

```
> 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);
```

$$Y := \begin{bmatrix} x_{0,0} + x_{0,1}\,\alpha \\ x_{1,0} + x_{1,1}\,\alpha \\ x_{2,0} + x_{2,1}\,\alpha \end{bmatrix}$$

```
> 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)mod 2,
> (i2+j2*alpha)mod 2]) fi od; od ;od ;od;
```

$$c := 0$$
$$1,\ [0,\ 1,\ \alpha]$$
$$2,\ [0,\ 1,\ 1]$$
$$3,\ [0,\ 1,\ 1+\alpha]$$
$$4,\ [1,\ \alpha,\ 0]$$
$$5,\ [1,\ 1,\ 0]$$
$$6,\ [1,\ 1+\alpha,\ 0]$$
$$7,\ [1,\ 0,\ \alpha]$$
$$8,\ [1,\ 0,\ 1]$$
$$9,\ [1,\ 0,\ 1+\alpha]$$

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

COURBE de FERMAT sur GF(16)

```
> restart ;
> with(linalg) :

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

> g:=x^4+x+1 mod 2;
```

$$g := x^4 + x + 1$$

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

$$\alpha$$

```
> f:=(x,y,z)->x^3+y^3+z^3;
```

$$f := (x,\ y,\ z) \to x^3 + y^3 + z^3$$

```
> h:=(i,j)->x[i-1,j-1]: X:=Matrix(3,4,h);
```

$$X := \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} \end{bmatrix}$$

```
> u:=(i,j)-> alpha^(i-1):U:=Matrix(4,1,u);
```

$$U := \begin{bmatrix} 1 \\ \alpha \\ \alpha^2 \\ \alpha^3 \end{bmatrix}$$

```
> with(LinearAlgebra):
> Y:= Multiply(X, U);
```

```
Warning, the assigned name GramSchmidt now has a global binding
```

$$Y := \begin{bmatrix} x_{0,0} + x_{0,1}\,\alpha + x_{0,2}\,\alpha^2 + x_{0,3}\,\alpha^3 \\ x_{1,0} + x_{1,1}\,\alpha + x_{1,2}\,\alpha^2 + x_{1,3}\,\alpha^3 \\ x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2 + x_{2,3}\,\alpha^3 \end{bmatrix}$$

```
> 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]) ;
```

$$vv_1 := 1 + (x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2 + x_{2,3}\,\alpha^3)^3$$

```
> vv[2]:= subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],v[2]) ;
```

$$vv_2 := 1 + (x_{1,0} + x_{1,1}\,\alpha + x_{1,2}\,\alpha^2 + x_{1,3}\,\alpha^3)^3 + (x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2 + x_{2,3}\,\alpha^3)^3$$

```
> 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
> for k from 0 to 1 do
> for l from 0 to 1 do
> if Eval(vv[1],{x[2,0]=i,x[2,1]=j,x[2,2]=k,x[2,3]=l}) mod 2 =0
> then c:=c+1;
> print (c,[0, 1,
> i+j*alpha+k*alpha^2+l*alpha^3]) fi; od; od ;od ;od;
> for i2 from 0 to 1 do
> for j2 from 0 to 1 do
> for k2 from 0 to 1 do
> for l2 from 0 to 1 do
> for i1 from 0 to 1 do
> for j1 from 0 to 1 do
> for k1 from 0 to 1 do
> for l1 from 0 to 1 do
> if Eval(vv[2],{x[2,0]=i2,x[2,1]=j2,x[2,2]=k2,x[2,3]=l2,
> x[1,0]=i1,x[1,1]=j1,x[1,2]=k1,x[1,3]=l1}) mod 2 =0 then c:=c+1;
> print (c,[1,
> i1+j1*alpha+k1*alpha^2+l1*alpha^3, i2+j2*alpha+k2*alpha^2+l2*alpha^3])
> fi; od; od ;od ;od;odd;od;od;od; od;
```

$$c := 0$$

$$1,\ [0,\ 1,\ \alpha + \alpha^2]$$

$$2,\ [0,\ 1,\ 1]$$

$$3,\ [0,\ 1,\ 1 + \alpha + \alpha^2]$$

$$4,\ [1,\ \alpha + \alpha^2,\ 0]$$

$$5,\ [1,\ 1,\ 0]$$
$$6,\ [1,\ \alpha+\alpha^2+1,\ 0]$$
$$7,\ [1,\ 0,\ \alpha^2+\alpha]$$
$$8,\ [1,\ 0,\ 1]$$
$$9,\ [1,\ 0,\ \alpha+1+\alpha^2]$$

%%

COURBE de KLEIN sur GF(16)

```
> fk:=(x,y,z)->x^3*y+y^3*z+z^3*x;
```

$$fk := (x,\ y,\ z) \to x^3\,y + y^3\,z + z^3\,x$$

```
> vk[1]:=fk(0, 1, x[2]);
```

$$vk_1 := x_2$$

```
> vk[2]:=fk(1, x[1], x[2]);
```

$$vk_2 := x_1 + {x_1}^3\,x_2 + {x_2}^3$$

```
> vvk[1]:=subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],vk[1]) ;
```

$$vvk_1 := x_{2,\,0} + x_{2,\,1}\,\alpha + x_{2,\,2}\,\alpha^2 + x_{2,\,3}\,\alpha^3$$

```
> vvk[2]:= subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],vk[2]) ;
```

$$\begin{aligned} vvk_2 := {} & x_{1,\,0} + x_{1,\,1}\,\alpha + x_{1,\,2}\,\alpha^2 + x_{1,\,3}\,\alpha^3 \\ & + (x_{1,\,0} + x_{1,\,1}\,\alpha + x_{1,\,2}\,\alpha^2 + x_{1,\,3}\,\alpha^3)^3\,(x_{2,\,0} + x_{2,\,1}\,\alpha + x_{2,\,2}\,\alpha^2 + x_{2,\,3}\,\alpha^3) \\ & + (x_{2,\,0} + x_{2,\,1}\,\alpha + x_{2,\,2}\,\alpha^2 + x_{2,\,3}\,\alpha^3)^3 \end{aligned}$$

```
> c:=0;
> if fk(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
> for k from 0 to 1 do
> for l from 0 to 1 do
> if Eval(vvk[1],{x[2,0]=i,x[2,1]=j,x[2,2]=k,x[2,3]=l}) mod 2 =0
> then c:=c+1;
> print (c,[0, 1,
> i+j*alpha+k*alpha^2+l*alpha^3]) fi; od; od ;od ;od;
> for i2 from 0 to 1 do
> for j2 from 0 to 1 do
> for k2 from 0 to 1 do
> for l2 from 0 to 1 do
> for i1 from 0 to 1 do
> for j1 from 0 to 1 do
> for k1 from 0 to 1 do
> for l1 from 0 to 1 do
> if Eval(vvk[2],{x[2,0]=i2,x[2,1]=j2,x[2,2]=k2,x[2,3]=l2,
> x[1,0]=i1,x[1,1]=j1,x[1,2]=k1,x[1,3]=l1}) mod 2 =0 then c:=c+1;
> print (c,[1,
> i1+j1*alpha+k1*alpha^2+l1*alpha^3, i2+j2*alpha+k2*alpha^2+l2*alpha^3])
> fi; od; od ;od ;od;odd;od;od;od; od;
```

$$c := 0$$

$$c := 1$$
$$1, [0, 0, 1]$$
$$2, [0, 1, 0]$$
$$3, [1, 0, 0]$$
$$4, [1, 1+\alpha^2, \alpha^3]$$
$$5, [1, \alpha, \alpha^3+\alpha^2]$$
$$6, [1, \alpha+1, \alpha+\alpha^3]$$
$$7, [1, \alpha^3, \alpha+\alpha^2]$$
$$8, [1, 1+\alpha^2+\alpha, \alpha+\alpha^2]$$
$$9, [1, 1+\alpha+\alpha^2+\alpha^3, \alpha+\alpha^2]$$
$$10, [1, \alpha+\alpha^2, \alpha^3+\alpha+\alpha^2]$$
$$11, [1, \alpha^2+\alpha, \alpha^3+1]$$
$$12, [1, 1+\alpha+\alpha^2, \alpha^3+1+\alpha^2]$$
$$13, [1, \alpha+\alpha^2+1, 1+\alpha^3+\alpha]$$
$$14, [1, \alpha^3+\alpha^2, \alpha+\alpha^2+1]$$
$$15, [1, \alpha+\alpha^3, \alpha+\alpha^2+1]$$
$$16, [1, \alpha+\alpha^2, \alpha+\alpha^2+1]$$
$$17, [1, \alpha^2, 1+\alpha+\alpha^2+\alpha^3]$$

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%

COURBE

de KLEIN sur GF(8)

```
> restart ;
> with(linalg) :
```

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

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

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

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

$$\alpha$$

```
> fk:=(x,y,z)->x^3*y+y^3*z+z^3*x;
```

$$fk := (x, y, z) \to x^3\, y + y^3\, z + z^3\, x$$

```
> h:=(i,j)->x[i-1,j-1]: X:=Matrix(3,3,h);
```

$$X := \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix}$$

```
> u:=(i,j)-> alpha^(i-1):U:=Matrix(3,1,u);
```

$$U := \begin{bmatrix} 1 \\ \alpha \\ \alpha^2 \end{bmatrix}$$

```
> with(LinearAlgebra):
> Y:= Multiply(X, U);
```

```
Warning, the assigned name GramSchmidt now has a global binding
```

$$Y := \begin{bmatrix} x_{0,0} + x_{0,1}\,\alpha + x_{0,2}\,\alpha^2 \\ x_{1,0} + x_{1,1}\,\alpha + x_{1,2}\,\alpha^2 \\ x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2 \end{bmatrix}$$

```
> vk[1]:=fk(0, 1, x[2]);
```

$$vk_1 := x_2$$

```
> vk[2]:=fk(1, x[1], x[2]);
```

$$vk_2 := x_1 + {x_1}^3\,x_2 + {x_2}^3$$

```
> vvk[1]:=subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],vk[1]) ;
```

$$vvk_1 := x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2$$

```
> vvk[2]:= subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],vk[2]) ;
```

$$vvk_2 := x_{1,0} + x_{1,1}\,\alpha + x_{1,2}\,\alpha^2 + (x_{1,0} + x_{1,1}\,\alpha + x_{1,2}\,\alpha^2)^3\,(x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2) + (x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2)^3$$

```
> c:=0;
> if fk(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
> for k from 0 to 1 do
> if Eval(vvk[1],{x[2,0]=i,x[2,1]=j,x[2,2]=k}) mod 2 =0 then
> c:=c+1;
> print (c,  [0, 1, i+j*alpha+k*alpha^2])
> fi; od; od ;od ;
> for i2 from 0 to 1 do
> for j2 from 0 to 1 do
> for k2 from 0 to 1 do
> for i1 from 0 to 1 do
> for j1 from 0 to 1 do
> for k1 from 0 to 1 do
> if Eval(vvk[2],{x[2,0]=i2,x[2,1]=j2,x[2,2]=k2,
> x[1,0]=i1,x[1,1]=j1,x[1,2]=k1}) mod 2 =0 then c:=c+1;
> print (c, [1, i1+j1*alpha+k1*alpha^2,
> i2+j2*alpha+k2*alpha^2]) fi; od; od ;od ;od;od;od;
```

$$c := 0$$

$$c := 1$$

$$1,\ [0,\ 0,\ 1]$$

$$2,\ [0,\ 1,\ 0]$$

$$3,\ [1,\ 0,\ 0]$$

$$4,\ [1,\ \alpha,\ \alpha^2]$$

$$5,\ [1,\ 1,\ \alpha^2]$$

$$6,\ [1,\ 1+\alpha,\ \alpha^2]$$

$$7,\ [1,\ \alpha^2+\alpha,\ \alpha]$$
$$8,\ [1,\ 1,\ \alpha]$$
$$9,\ [1,\ 1+\alpha+\alpha^2,\ \alpha]$$
$$10,\ [1,\ \alpha^2,\ \alpha^2+\alpha]$$
$$11,\ [1,\ 1,\ \alpha^2+\alpha]$$
$$12,\ [1,\ 1+\alpha^2,\ \alpha^2+\alpha]$$
$$13,\ [1,\ \alpha^2,\ 1]$$
$$14,\ [1,\ \alpha,\ 1]$$
$$15,\ [1,\ \alpha^2+\alpha,\ 1]$$
$$16,\ [1,\ \alpha,\ 1+\alpha^2]$$
$$17,\ [1,\ 1+\alpha^2,\ 1+\alpha^2]$$
$$18,\ [1,\ 1+\alpha+\alpha^2,\ 1+\alpha^2]$$
$$19,\ [1,\ \alpha^2+\alpha,\ 1+\alpha]$$
$$20,\ [1,\ 1+\alpha^2,\ 1+\alpha]$$
$$21,\ [1,\ 1+\alpha,\ 1+\alpha]$$
$$22,\ [1,\ \alpha^2,\ 1+\alpha+\alpha^2]$$
$$23,\ [1,\ 1+\alpha,\ 1+\alpha+\alpha^2]$$
$$24,\ [1,\ 1+\alpha+\alpha^2,\ 1+\alpha+\alpha^2]$$

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%

COURBE de FERMAT sur GF(8)

```
> restart ;
> with(linalg) :
```

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

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

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

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

$$\alpha$$

```
> f:=(x,y,z)->x^3+y^3+z^3;
```

$$f := (x,\ y,\ z) \to x^3 + y^3 + z^3$$

```
> h:=(i,j)->x[i-1,j-1]: X:=Matrix(3,3,h);
```

$$X := \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix}$$

```
> u:=(i,j)-> alpha^(i-1):U:=Matrix(3,1,u);
```

$$U := \begin{bmatrix} 1 \\ \alpha \\ \alpha^2 \end{bmatrix}$$

```
> with(LinearAlgebra):
> Y:= Multiply(X, U);
```

```
Warning, the assigned name GramSchmidt now has a global binding
```

$$Y := \begin{bmatrix} x_{0,0} + x_{0,1}\,\alpha + x_{0,2}\,\alpha^2 \\ x_{1,0} + x_{1,1}\,\alpha + x_{1,2}\,\alpha^2 \\ x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2 \end{bmatrix}$$

```
> 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]) ;
```

$$vv_1 := 1 + (x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2)^3$$

```
> vv[2]:= subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],v[2]) ;
```

$$vv_2 := 1 + (x_{1,0} + x_{1,1}\,\alpha + x_{1,2}\,\alpha^2)^3 + (x_{2,0} + x_{2,1}\,\alpha + x_{2,2}\,\alpha^2)^3$$

```
> c:=0;
> for i from 0 to 1 do
> for j from 0 to 1 do
> for k from 0 to 1 do
> if Eval(vv[1],{x[2,0]=i,x[2,1]=j,x[2,2]=k}) mod 2 =0 then
> c:=c+1;
> print (c,[0, 1, i+j*alpha+k*alpha^2])
> fi; od; od ;od ;
> for i from 0 to 1 do
> for j from 0 to 1 do
> for k from 0 to 1 do
> for i1 from 0 to 1 do
> for j1 from 0 to 1 do
> for k1 from 0 to 1 do
> if Eval(vv[2],{x[2,0]=i,x[2,1]=j,x[2,2]=k,
> x[1,0]=i1,x[1,1]=j1,x[1,2]=k1}) mod 2 =0 then c:=c+1;
> print (c,[1, (i1+j1*alpha+k1*alpha^2)
> mod 2 ,(i+j*alpha+k*alpha^2) mod 2]) fi; od; od ;od ;od;odd;od;od;
```

$$c := 0$$

$$1,\ [0,\ 1,\ 1]$$

$$2,\ [1,\ 1,\ 0]$$

$$3,\ [1,\ 1+\alpha,\ \alpha^2]$$

$$4,\ [1,\ \alpha^2+\alpha+1,\ \alpha]$$

$$5,\ [1,\ \alpha^2+1,\ \alpha+\alpha^2]$$

$$6,\ [1,\ 0,\ 1]$$

$$7,\ [1,\ \alpha+\alpha^2,\ 1+\alpha^2]$$

$$8,\ [1,\ \alpha^2,\ \alpha+1]$$

$$9,\ [1,\ \alpha,\ 1+\alpha+\alpha^2]$$

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%

COURBE de HERMITE sur GF(9)

```
> restart ;
> with(linalg) :
> g:=x^2+1 mod 3;
```

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

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

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

$$\alpha$$

```
> f:=(x,y,z)->x^4+y^4+z^4;
```

$$f := (x,\, y,\, z) \to x^4 + y^4 + z^4$$

```
> 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]);
```

$$v_1 := 1 + {x_2}^4$$

```
> v[2]:=f(1, x[1], x[2]);
```

$$v_2 := 1 + {x_1}^4 + {x_2}^4$$

```
> vv[1]:=subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],v[1]) ;
```

$$vv_1 := 1 + (x_{2,0} + x_{2,1}\,\alpha)^4$$

```
> vv[2]:= subs(x[0]=Y[1,1],x[1]=Y[2,1], x[2]=Y[3,1],v[2]) ;
```

$$vv_2 := 1 + (x_{1,0} + x_{1,1}\,\alpha)^4 + (x_{2,0} + x_{2,1}\,\alpha)^4$$

```
> c:=0;
> for i from 0 to 2 do
> for j from 0 to 2 do
> if Eval(vv[1],{x[2,0]=i,x[2,1]=j}) mod 3 =0 then c:=c+1;
> print (c,[0, 1, i+j*alpha]) fi; od; od
> ;for i2 from 0 to 2 do
> for j2 from 0 to 2 do
> for i1 from 0 to 2 do
> for j1 from 0 to 2 do
> if Eval(vv[2],{x[2,0]=i2,x[2,1]=j2,x[1,0]=i1,x[1,1]=j1})
> mod 3 =0 then c:=c+1;
> print (c,[1, (i1+j1*alpha)mod 3
> ,(i2+j2*alpha) mod 3]) fi; od; od ;od ;od;
```

$$c := 0$$

$$1,\, [0,\, 1,\, 1 + \alpha]$$

2, $[0,\ 1,\ 1+2\,\alpha]$

3, $[0,\ 1,\ 2+\alpha]$

4, $[0,\ 1,\ 2+2\,\alpha]$

5, $[1,\ 1+\alpha,\ 0]$

6, $[1,\ 1+2\,\alpha,\ 0]$

7, $[1,\ 2+\alpha,\ 0]$

8, $[1,\ 2+2\,\alpha,\ 0]$

9, $[1,\ \alpha,\ \alpha]$

10, $[1,\ 2\,\alpha,\ \alpha]$

11, $[1,\ 1,\ \alpha]$

12, $[1,\ 2,\ \alpha]$

13, $[1,\ \alpha,\ 2\,\alpha]$

14, $[1,\ 2\,\alpha,\ 2\,\alpha]$

15, $[1,\ 1,\ 2\,\alpha]$

16, $[1,\ 2,\ 2\,\alpha]$

17, $[1,\ \alpha,\ 1]$

18, $[1,\ 2\,\alpha,\ 1]$

19, $[1,\ 1,\ 1]$

20, $[1,\ 2,\ 1]$

21, $[1,\ 0,\ 1+\alpha]$

22, $[1,\ 0,\ 1+2\,\alpha]$

23, $[1,\ \alpha,\ 2]$

24, $[1,\ 2\,\alpha,\ 2]$

25, $[1,\ 1,\ 2]$

26, $[1,\ 2,\ 2]$

27, $[1,\ 0,\ 2+\alpha]$

28, $[1,\ 0,\ 2+2\,\alpha]$

F Annexe : Systèmes linéaires dans GF(p^d) (F. SERGERAERT)

```
> restart :
```

- Exemple dans GF(3^4).

- On prend un polynôme irréductible de degré 4 dans F_3, affecté à **irr34**.

```
> irr34 := op(1, select(has, Factor(x^81-x) mod 3, 4)) ;
```

$$irr34 := x^4 + 2\,x^3 + 2\,x^2 + x + 2$$

- Le polynôme irréductible obtenu par ce procédé n'est pas forcément le même d'une session à l'autre.

- On aliasse α à une racine de ce polynôme dans une extension de F_3, de sorte que GF(3^4) = F_3[α].

```
> alias(alpha = RootOf(irr34) mod 3) ;
```

$$\alpha$$

- La procédure **rnd3** génére 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 GF(3^4).

```
> rnd34 := () -> add(rnd3()*alpha^i, i=0..3) :
> seq(rnd34(), i = 1..5) ;
```

$$2 + 2\,\alpha + 2\,\alpha^2 + \alpha^3,\ 1 + \alpha,\ 2 + 2\,\alpha + \alpha^2,\ 2\,\alpha^2,\ \alpha^2$$

- La matrice A est une matrice 3x3 pseudo-aléatoire à coefficients dans GF(3^4).

```
> A := matrix(3, 3, rnd34) ;
```

$$A := \begin{bmatrix} 1 + 2\,\alpha^2 + 2\,\alpha^3 & 2 + 2\,\alpha + 2\,\alpha^3 & 2 + 2\,\alpha^2 + 2\,\alpha^3 \\ 2\,\alpha^2 & 2 + \alpha + \alpha^2 & 1 + \alpha^3 \\ 2 + 2\,\alpha + \alpha^3 & 1 + \alpha^2 & \alpha + 2\,\alpha^2 \end{bmatrix}$$

- Idem pour un vecteur second membre.

```
> b := vector(3, rnd34) ;
```

$$b := \left[\alpha^3,\ \alpha^3,\ 2 + 2\,\alpha^2\right]$$

- Linsolve(...) mod 3 permet de résoudre dans GF(3^4).

```
> x := Linsolve(A,b) mod 3 ;
```

$$x := \left[\alpha^3 + 2\,\alpha^2 + 2,\ \alpha + \alpha^2,\ 2\right]$$

- Vérification. Calcul de $A\,x - b$.

```
> zerov := evalm(A &* x - b) ;
```

$$zerov := \Big[(1+2\,\alpha^2+2\,\alpha^3)\,(\alpha^3+2\,\alpha^2+2)+(2+2\,\alpha+2\,\alpha^3)\,(\alpha+\alpha^2)+4+4\,\alpha^2+3\,\alpha^3,$$
$$2\,\alpha^2\,(\alpha^3+2\,\alpha^2+2)+(2+\alpha+\alpha^2)\,(\alpha+\alpha^2)+\alpha^3+2,$$
$$(2+2\,\alpha+\alpha^3)\,(\alpha^3+2\,\alpha^2+2)+(1+\alpha^2)\,(\alpha+\alpha^2)+2\,\alpha+2\,\alpha^2-2\Big]$$

- Les termes du vecteur obtenu ne sont pas « réduits » à leur forme canonique dans $\mathrm{GF}(3^4) = F_3[\alpha]$. Pour obtenir la réduction.

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

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

Références

[Dem] Demazure, Michel *Cours d'algèbre. Primalité. Divisibilité. Codes.*, Nouvelle Bibliothèque Mathématique [New Mathematics Library], 1. Cassini, Paris, 1997. xviii+302 pp.

[vLi] J.H. van Lint, *Introduction to coding theory.* Springer-Verlag, New-York

[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

[Pey] Emmanuel Peyre, *Corps finis et courbes elliptiques.* DESS Cryptologie, sécurité et codage d'information, Modules A1A et A1B, Grenoble, 2002, pp. 1-128

[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

Institut Fourier,
B.P.74, 38402 St.–Martin d'Hères,
FRANCE