L'algèbre dans la correction des erreurs

Dany-Jack Mercier

L'Algèbre dans la Correction des Erreurs

Dany-Jack Mercier
IUFM Antilles-Guyane
Equipe Applications de l'Algèbre et de l'Arithmétique
de l'Université des Antilles-Guyane

avril-mai 1998

Abstract

Cet article montre à quel point l'algèbre linéaire, les polynômes et les corps finis sont présents dans de nombreuses applications concrètes touchant aux domaines de l'information et de la communication. Il informe sur l'enjeu actuel du transfert des informations et sur les moyens algébriques mis en oeuvre pour réaliser une "communication parfaite".

1 Introduction

Si la première moitié du vingtième siècle a été celle de la révolution analogique par la radio et la télévision, la seconde moitié de ce siècle est celle de la révolution numérique et de l'utilisation systématique de l'algèbre dans la transmission de données.

Ainsi, après l'apparition des CD Audio dans les années 1980, il faut compter sur le développement de la diffusion par satellite et l'utilisation de nouveaux moyens de communication tels la télécopie, le réseau internet ou le téléphone numérique. Les données (textes, images et sons) sont maintenant engrangées sur des CD-ROM dont les lecteurs sont munis de systèmes de correction d'erreurs. Même la photographie et la radio deviennent numériques.

Les techniques de restitution d'images ou de sons sont liées à la transmission et à la lecture correcte de nombreux messages numériques, encore appelés "mots". Un message est formé de mots eux mêmes constitués de symboles (ou bits) pris dans un alphabet. Prenons le message 00101 formé de 5 bits valant chacun 0 ou 1. Si nous transmettons le message tel quel, une erreur de

[0] [ccod0002] v1.10 APMEP **415**, 1998, pp.173-191. (dernière révision : 13 juin 2008)

transmission ou de lecture peut avoir lieu et rendre le message inintelligible. Décidons de répéter ce message trois fois, et d'envoyer :

001010010100101.

Si le message reçu comporte moins d'une erreur, cette erreur peut être corrigée. S'il comporte moins de deux erreurs, le récepteur est capable de détecter qu'il y a eu erreur mais ne peut pas toujours récupérer le message originel. Enfin, s'il se produit plus de deux erreurs pendant la transmission, le récepteur peut ne pas les détecter.

Nous venons de voir un premier exemple de **code correcteur d'erreur**, appelé **code à répétition**. Ce code, qui corrige une erreur et en détecte deux, est utilisé dans certains lecteurs de CD Audio possédant trois têtes de lecture. Le signal 0 ou 1 est lu indépendamment par chacune de ces trois têtes pour donner un mot de trois chiffres, et une erreur de lecture peut être corrigée.

Remarquons bien qu'il est naturel d'allonger un message pour le protéger. Prenons les mots d'un language. Ils sont en général très "éloignés" les uns des autres, deux mots différant selon leurs longueurs et selon les lettres et les syllabes utilisées. Ainsi on confondra difficilement les mots "bibliothèque" et "armoire" même si ces mots sont mal prononçés ou entendus, et l'on reconstituera naturellement le message dans une conversation quand bien même certains sons seraient supprimés ou déformés. Les aviateurs quant à eux épèlent leurs numéros d'immatriculation en disant "alpha zoulou" pour "AZ"...

Un deuxième exemple de détection d'erreurs largement utilisé en informatique est l'adjonction d'un **bit de parité**. Reprenons le message 00101 et ajoutons lui un dernier bit obtenu en additionnant les cinq bits du départ modulo 2. Le message devient 001010 et permet de détecter une erreur sans toutefois pouvoir la corriger. Pour cela, on fait la somme de tous les bits pour obtenir 0 s'il n'y a pas eu d'erreur, et 1 dans le cas contraire. Ce code, appelé **code de parité**, est utilisé un peu partout : dans les numéros de sécurité sociale où l'on rajoute la "clé", dans ceux des comptes bancaires ou encore dans les code-barres des supermarchés où c'est le 13-ième chiffre qui constitue la clé de contrôle.

Ces deux exemples fondamentaux sont à la base de la théorie du codage et montrent que l'on peut maîtriser l'apparition d'erreurs en allongeant volontairement le message avant sa transmission ou sa lecture. Des techniques algébriques plus sophistiquées sont ensuite utilisées pour améliorer les performances du codage, le but étant :

- de savoir si des erreurs se sont produites (problème de la détection),
- de retrouver le message correct à partir du message reçu (problème de la correction),

- de corriger le plus d'erreurs possibles tout en utilisant le moins de bits supplémentaires possibles (problème de la performance du codage).

Du point de vue mathématique, l'un des intérêts de la théorie des codes est de montrer que l'algèbre s'applique fondamentalement dans notre vie de tous les jours dès que l'on écoute de la musique ou que l'on s'installe devant son téléviseur, et que des notions aussi abstraites que celles d'espaces vectoriels ou de polynômes sur des corps finis nous permettent de lire des messages, d'écouter de la musique ou de regarder des films dans des conditions optimum. Ces applications pratiques me servent parfois à justifier aux yeux des étudiants de DEUG l'apprentissage des techniques d'algèbre linéaire. Elles donnent aussi une réponse aux lycéens qui se demandent à quoi peuvent bien servir les polynômes.

2 Code correcteur d'erreurs

Soit Q un ensemble fini à q éléments. Soient k et n deux entiers naturels non nuls avec $k \leq n$. L'ensemble des messages sera une partie E de Q^k, et l'on introduit une application injective

$$\begin{array}{rccc} f: & E & \to & Q^n \\ & a = (a_1, a_2, ..., a_k) & \mapsto & c = (c_1, c_2, ..., c_n) \end{array}$$

appelée **application de codage** ou **encodeur**. Le message ou mot a est un élément de E. Il est modifié pour fournir le mot $f(a) = c \in Q^n$. C'est le mot c qui sera transmis et lu par un sytème quelconque pour donner un message reçu $x = (x_1, ..., x_n)$ éventuellement entaché d'erreurs.

Notons $C = f(E)$ l'image de f. Comme f est injective, f réalise une bijection de E sur C et C peut être considéré comme l'ensemble de tous les messages possibles. C est appelé **code de longueur** n, et les éléments de C s'appellent les **mots** du code. Le **cardinal** du code est par définition celui de C. On le notera $\#C$ ou M. Pour mesurer le degré de différence entre deux mots x et y de Q^n, on utilise la **distance de Hamming** d définie par

$$\forall x, y \in Q^n \quad d(x, y) = \#\{i \in \mathbb{N}_n \,/\, x_i \neq y_i\}.$$

On vérifie facilement que d est bien une distance sur Q^n. La **distance minimale** du code C est la distance minimum entre deux mots distincts de ce code. On la note $d(C)$ ou d, et

$$d = \text{Min}\{d(x, y) \,/\, x, y \in C \text{ avec } x \neq y\}.$$

Un code C de longueur n, de cardinal M et de distance minimale d est appelé code (n, M, d). Les nombres n, M, d sont les **paramètres** du code.

d joue un rôle important car se trouve en relation étroite avec le nombre d'erreurs susceptibles d'être corrigées. Supposons que le message soit $c = (c_1, ..., c_n)$ et qu'il y ait eu moins de t erreurs de transmission ou de lecture. Le message obtenu $x = (x_1, ..., x_n)$ vérifie $d(x, c) \leq t$. On peut retrouver c à partir de x si, et seulement si, il existe un seul mot de code situé à une distance de x inférieure ou égale à t. Autrement dit, il faut et il suffit que les boules fermées de rayon t centrées sur les éléments du code C soient disjointes. Un **code corrigera t erreurs** si cette condition est vérifiée. En notant $[r]$ la partie entière d'un réel r :

Théorème 1 *Un code C de distance minimale d corrige au plus $e = \left[\frac{d-1}{2}\right]$ erreurs et en détecte $d-1$.*

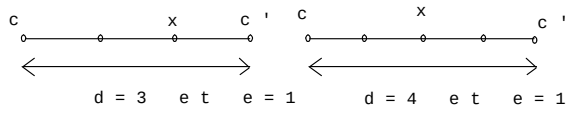
Preuve : Si x vérifie simultanément $d(x, c) \leq e$ et $d(x, c') \leq e$ où c, c' sont des mots de C, alors

$$d \leq d\left(c, c'\right) \leq d(c, x) + d\left(x, c'\right) \leq 2e$$

ce qui est absurde puisque $2e \leq d-1$. Ainsi C corrige e erreurs. Pour vérifier que C ne corrige pas plus de $e+1$ erreurs, on envisage le cas le plus défavorable où les mots c et c' de C sont à distance minimum, i.e. où $d(c, c') = d$, et l'on choisit un mot x du segment $[c, c']$ tel que $d(c, x) = e + 1$. Dans ce cas $d(c, x) + d(x, c') = d(c, c') = d$. Supposons que c soit le message envoyé et x le message reçu dans lequel il y a $e + 1$ erreurs.

$$d\left(x, c'\right) = d - (e + 1) \leq d(c, x)$$

montre que c' est un mot de code plus rapproché de x que ne l'est c. Le décodage consistant à remplacer x par le mot de code le plus proche de x donnera c' si l'inégalité précédente est stricte, ou bien hésitera entre c et c' si l'inégalité est une égalité (voir figure ci-dessous). Dans les deux cas la correction de l'erreur est impossible. ■



L'entier $e = \left[\frac{d-1}{2}\right]$ représente la **capacité de correction** du code C, et C est appelé **code e-correcteur d'erreurs**.

3 Codes linéaires

3.1 Définitions

Rappelons que si q (distinct de 1) est une puissance d'un nombre premier, il existe un et un seul corps fini de cardinal q à isomorphisme près (cf. [3], [6]) Ce corps est noté $\mathbb{F}_q$. L'ensemble des messages $E = \mathbb{F}_q^k$ est maintenant structuré en espace vectoriel de dimension k sur $\mathbb{F}_q$. Si l'on décide de n'utiliser que des encodeurs linéaires f, le code $C = f\left(\mathbb{F}_q^k\right)$ devient un sous-espace vectoriel de $\mathbb{F}_q^n$.

Définition 1 *Un **code linéaire** de dimension k et de longueur n est un sous-espace vectoriel de dimension k de $\mathbb{F}_q^n$. Si la distance minimal de C est d, on dit que C est un code $[n, k, d]$ (ou simplement $[n, k]$). Si $q = 2$, on dit que C est un **code binaire**.*

Définition 2 *Le nombre de coordonnées non nulles de $x \in \mathbb{F}_q$ est appelé **poids de Hamming** de x, et noté*

$$w(x) = \#\{i \in \mathbb{N}_n \,/\, x_i \neq 0\}.$$

Comme $d(x, y) = w(x - y)$, la distance minimale de C s'écrit

$$d = \min\{w(x) \,/\, x \in C^*\}.$$

Notons ${}^tG$ la matrice de l'application linéaire $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ dans les bases canoniques. (${}^tG$ désigne la transposée de la matrice G). G est du type $k \times n$ (i.e. à k lignes et n colonnes) et tout mot de C s'écrit $c = f(x) = xG$ où $c = (c_1, ..., c_n) \in \mathbb{F}_q^n$ et $x = (x_1, ..., x_k) \in \mathbb{F}_q^k$ sont des vecteur-lignes.

Définition 3 *La matrice G est une **matrice génératrice** du code C. Toute matrice H de type $(n - k) \times n$ vérifiant*

$$c \in C \Leftrightarrow H\,{}^tc = 0$$

*est appelée **matrice de contrôle** de C. On a $C = \operatorname{Ker} H$ et $\operatorname{rg} H = n - k$.*

3.2 Codes systématiques

A chaque mot $x = (x_1, ..., x_k)$ du message on adjoint $n-k$ symboles $c_{k+1}, ..., c_n$ dépendant linéairement des x_i pour obtenir le mot de code $c = f(x)$. Les symboles c_i sont appelés **bits de contrôle**, et

$$c = (x_1, ..., x_k, c_{k+1}, ..., c_n) = (x_1, ..., x_k)\,(I_k \,|\, A)$$

où $(I_k \mid A)$ désigne la matrice $k \times n$ obtenue en écrivant côte à côte la matrice identité I_k de taille k et une matrice quelconque A. On dira qu'un code C est **systématique** s'il possède une matrice génératrice de la forme $G = (I_k \mid A)$. Dans ce cas, $c \in C$ si, et seulement si, $c = xG = x\,(I_k \mid A)$, et

$$\left(-\,{}^tA \mid I_{n-k}\right)\,{}^tc = \left(-\,{}^tA \mid I_{n-k}\right)\begin{pmatrix} I_k \\ {}^tA \end{pmatrix}\,{}^tx = -\,{}^tA\,{}^tx + {}^tA\,{}^tx = 0.$$

C est inclus dans $\operatorname{Ker} H$ où $H = \left(-\,{}^tA \mid I_{n-k}\right)$. Comme H est de rang $n-k$, les sous-espaces C et $\operatorname{Ker} H$ ont même dimension k et donc $C = \operatorname{Ker} H$. La matrice $H = \left(-\,{}^tA \mid I_{n-k}\right)$ est par conséquent une matrice de contrôle de C.

Exemples : L'application $f\,(x_1, x_2, x_3) =$(x_1, x_2, x_3, x_1, $x_2 + x_3$, $x_1 + x_2 + x_3$, x_3) définit un code systématique C de type $[7, 3]$ sur $\mathbb{F}_2$. L'écriture

$$(c_1, ..., c_6) = (x_1, x_2, x_3)\begin{pmatrix} 1 & 0 & 0 & \mid & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & \mid & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & \mid & 0 & 1 & 1 & 1 \end{pmatrix}$$

met en évidence une matrice génératrice de C d'où l'on peut déduire la matrice de contrôle

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

Le **code de parité binaire** vu dans l'introduction consiste à ajouter le bit de parité $c_{k+1} = \sum_{i=1}^{k} x_i$ à $(x_1, ..., x_k)$. C'est un code systématique $[k+1, k]$. Le **code à répétition** consistant à répéter n fois le symbole x_1 sera un code systématique $[n, 1]$.

3.3 Codes MDS

Théorème 2 *Si C est un code linéaire $[n, k, d]$ de matrice de contrôle H, alors*

$$d = \operatorname{Min}\left\{s \in \mathbb{N}^* \,/\, \textit{il existe } s \textit{ colonnes de } H \textit{ linéairement dépendantes}\right\}.$$

Preuve : Il existe un mot de code $x = (x_1, ..., x_n)$ de poids d.
Notons $H = [h_1, ..., h_n]$ où h_i représente la i-ième colonne de H. $x \in C$ signifie que $H\,{}^tx = 0$, soit $\sum_{i=1}^{k} x_i h_i = 0$ et constitue une relation de dépendance d'exactement d colonnes de H. Il existera donc d colonnes linéairement dépendantes. D'autre part, si les s colonnes $h_{i_1}, ..., h_{i_s}$ de H sont linéairement dépendantes, il existe une s-liste $(x_{i_1}, ..., x_{i_s}) \neq (0, ..., 0)$ telle que $\sum_{j=1}^{s} x_{i_j} h_{i_j} = 0$. En posant $x_i = 0$ si $i \notin \{i_1, ..., i_s\}$ et $x = (x_1, ..., x_n)$, on constate que $x \in C \setminus \{0\}$ et $w\,(x) \leq s$, donc $d \leq s$. ■

Corollaire 1 *Pour tout code* $[n,k,s]$*, on a* $k+d \leq n+1$.

Preuve : H est une matrice de rang $n-k$, donc $n-k+1$ de ses colonnes sont toujours liées et le Théorème entraine $d \leq n-k+1$. ■

La majoration $d \leq n-k+1$ est appelée **borne de Singleton** et un code $[n,k,d]$ vérifiant $d = n-k+1$ est appelé **code MDS** (Maximum Distance Separable). Il s'agit alors d'un bon code puisque pour n et k fixés, d est maximum et le nombre d'erreurs qu'il est possible de corriger aussi. On notera que C est un code MDS si, et seulement si, $n-k$ colonnes quelconques de H sont toujours linéairement indépendantes.

4 Codes cycliques

Un code linéaire C est dit **cyclique** s'il est stable par permutation à droite de ses composantes, i.e.

$$(a_0,...,a_{n-2},a_{n-1}) \in C \Rightarrow (a_{n-1},a_0,...,a_{n-2}) \in C.$$

Notons $\mathbb{F}_q[x]$ l'algèbre des polynômes à coefficients dans le corps de Galois $\mathbb{F}_q$ et (x^n-1) l'idéal engendré par le polynôme x^n-1. L'algèbre quotient $\mathbb{F}_q[x]/(x^n-1)$ est un $\mathbb{F}_q$-espace vectoriel de dimension n, dont une base est $\left(\dot{1},\dot{x},...,\dot{x}^{n-1}\right)$. Cela permet d'identifier (vectoriellement) un élément $a=(a_0,...,a_{n-1})$ de $\mathbb{F}_q^n$ au polynôme $a(x) = a_0 + a_1\dot{x} + ... + a_{n-1}\dot{x}^{n-1}$ de $\mathbb{F}_q[x]/(x^n-1)$. Pour simplifier les notations, on oubliera de mettre le point au dessus de la classe de x, de sorte que l'on écrira

$$a(x) = a_0 + a_1x + ... + a_{n-1}x^{n-1} \in \mathbb{F}_q[x]/(x^n-1)$$

le polynôme associé au vecteur $a=(a_0,...,a_{n-1}) \in \mathbb{F}_q^n$. Avec cette identification, C apparaît comme un sous-espace vectoriel de $\mathbb{F}_q[x]/(x^n-1)$. Si C est un code cyclique,

$$a(x) = a_0+a_1x+...+a_{n-1}x^{n-1} \in C \Rightarrow x.a(x) = a_{n-1}+a_0x+...+a_{n-2}x^{n-1} \in C$$

et de proche en proche, $x^i a(x) \in C$ pour tout entier i. Ainsi $b(x)\,a(x)$ sera dans C pour tout polynôme b et C sera un idéal de $\mathbb{F}_q[x]/(x^n-1)$. Réciproquement, tout idéal de $\mathbb{F}_q[x]/(x^n-1)$ est un sous-espace vectoriel stable par permutation à droite des coordonnées dans la base $\left(1,x,...,x^{n-1}\right)$, et l'on peut énoncer :

Théorème 3 *Moyennant l'identification des* $\mathbb{F}_q$*-espaces vectoriels* $\mathbb{F}_q^n$ *et* $\mathbb{F}_q[x]/(x^n-1)$*, un code cyclique est un idéal de* $\mathbb{F}_q[x]/(x^n-1)$.

Il s'agit maintenant de préciser la nature des idéaux de $\mathbb{F}_q[x]/(x^n-1)$. Dans le Théorème suivant, on distingue encore une dernière fois g et $\dot{g}$ pour plus de clarté.

Théorème 4 *Soit k un corps et $f \in k[x]$. L'anneau $A = k[x]/(f)$ est principal, et tout idéal de A est de la forme $\left(\dot{g}\right)$ où g est un polynôme unitaire de $k[x]$ qui divise f. De plus un tel polynôme g est unique.*

Preuve : Soit $\pi : k[x] \to A$ la projection canonique. Si I est un idéal de A, $J = \pi^{-1}(I)$ est un idéal de l'anneau principal $k[x]$, et l'on sait que le polynôme $g \in k[x]$ unitaire de plus bas degré de J vérifie $J = (g)$. Comme π est surjective,

$$\pi(J) = \pi\left(\pi^{-1}(I)\right) = I$$

d'où $I = \left(\dot{g}\right)$. De plus $\pi^{-1}(0) = (f) \subset J$ montre que g divise f.
Montrons maintenant l'unicité de g. Si $h \in k[x]$ vérifie $I = \left(\dot{g}\right) = \left(\dot{h}\right)$, $\dot{h} = \dot{g}\dot{u}$ pour un polynôme convenable u, donc il existe $v \in k[x]$ tel que $h = gu + fv$. Comme g divise f, g divisera h. En recommençant de la même façon, on montre que h divise g, de sorte que les polynômes g et h soient unitaires et associés, donc égaux. ■

Définition 4 *Soit C un code cyclique. L'unique polynôme unitaire g de $\mathbb{F}_q[x]$ qui divise $x^n - 1$ et tel que $C = (g)$ s'appelle le* ***polynôme générateur*** *du code cyclique C.*

Si C est le code cyclique de longueur n sur $\mathbb{F}_q$ et de polynôme générateur g, posons $k = n - \deg g$. L'application

$$\begin{array}{rccc} \gamma : & \mathbb{F}_q^k & \to & \mathbb{F}_q^n \simeq \mathbb{F}_q[x]/(x^n-1) \\ & a = (a_0, ..., a_{k-1}) & \mapsto & a(x)\, g(x) \end{array}$$

est injective et applique $\mathbb{F}_q^k$ sur C, donc réalise un encodage simple de C et montre que C est de dimension $k = n - \deg g$.

5 Codes BCH (Bose-Chaudhuri-Hocquenghem)

Soient n un entier premier avec q et α une racine primitive n-ième de l'unité dans une extension algébriquement close de $\mathbb{F}_q$. Le corps de décomposition de $x^n - 1$ est $\mathbb{F}_{q^m}$ où m représente l'ordre multiplicatif de q modulo n, de sorte que toutes les puissances de α appartiennent à $\mathbb{F}_{q^m}$. Avec ces notations, on

appelle **code BCH de longueur n et de distance construite d sur $\mathbb{F}_q$** tout code cyclique de longueur n et de polynôme générateur

$$g(x) = \text{ppcm}\left(p_r(x), p_{r+1}(x), ..., p_{r+d-2}(x)\right)$$

où r et d sont deux entiers naturels non nuls avec $2 \leq d \leq n$ et où $p_i(x)$ représente le polynôme minimal de α^i sur $\mathbb{F}_q$.
On notera que g divise $x^n - 1$ et que g est le polynôme de plus petit degré dans $\mathbb{F}_q[x]$ admettant les éléments distincts $\alpha^r, \alpha^{r+1}, ..., \alpha^{r+d-2}$ comme racines.
Si $r = 1$, on dit que C est un **code BCH au sens strict**. Si $n = q^m - 1$, C est appelé **code BCH primitif** (i.e. α est un élément primitif de $\mathbb{F}_{q^m}$). Enfin un **code de Reed-Solomon** de longueur $q-1$ est un code BCH de longueur $q-1$ sur le corps $\mathbb{F}_q$ (i.e. α est un élément primitif de $\mathbb{F}_q$).

Théorème 5 *La distance minimale d'un code BCH est supérieure à sa distance construite.*

Preuve : Un mot $a = (a_0, a_1, ..., a_{n-1})$ appartient à C si, et seulement si, le polynôme associé $a(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1}$ est multiple de $g(x)$. Cela revient à dire que $a(x)$ est divisible par chacun des polynômes $p_i(x)$, ou encore que $a(x)$ admet $\alpha^r, \alpha^{r+1}, ..., \alpha^{r+d-2}$ comme racines. Cela s'écrit

$$a_0 + a_1\alpha^{r+j} + ... + a_{n-1}\left(\alpha^{r+j}\right)^{n-1} = 0$$

pour tout j, ou encore $H\,{}^t a = 0$ avec

$$H = \begin{pmatrix} 1 & \alpha^r & \alpha^{2r} & \alpha^{(n-1)r} \\ 1 & \alpha^{r+1} & \alpha^{2(r+1)} & \alpha^{(n-1)(r+1)} \\ \vdots & & & \vdots \\ 1 & \alpha^{r+d-2} & \alpha^{2(r+d-2)} & \alpha^{(n-1)(r+d-2)} \end{pmatrix}.$$

H est une matrice de contrôle de $\acute{C}$ et $d-1$ colonnes quelconques de cette matrice sont indépendantes (utiliser un déterminant de Vandermonde). On applique le Théorème 2. ■

Théorème 6 *Un code de Reed-Solomon de longueur $q-1$ et de distance construite d ($2 \leq d \leq q-1$) est un code cyclique dont le polynôme générateur s'écrit*

$$g(x) = (x - \alpha^r)\left(x - \alpha^{r+1}\right)...\left(x - \alpha^{r+d-2}\right)$$

où α est un élément primitif de $\mathbb{F}_q$. Ses paramètres sont $[q-1, q-d, d]$ et c'est donc un code MDS.

Preuve : α est racine $(q-1)$-ième primitive de l'unité i.e. un élément primitif de $\mathbb{F}_q$. Le polynôme minimal $p_i(x)$ de α^i sera $x-\alpha^i$ donc

$$g(x) = (x-\alpha^r)\left(x-\alpha^{r+1}\right)...\left(x-\alpha^{r+d-2}\right)$$

engendrera C. Comme $\dim C = k = n - \deg g = q-d$, la distance minimale $\operatorname{dist} C$ de C vérifiera compte tenu de la Borne de Singleton et du Théorème précédent

$$d \leq \operatorname{dist} C \leq n-k+1 = d. \blacksquare$$

6 Minitel

On se réfère à l'article [1]. On rappelle qu'un octet est une succession de 8 bits, un bit (abréviation de "binary digit") désignant l'un des symboles 0 ou 1.

6.1 Codage

Les informations qui circulent sur les lignes téléphoniques entre deux modems sont regroupées en blocs de 15 octets, et le dernier bit de chacun de ces 15 octets est un bit de parité (i.e. la somme des 7 bits qui le précède). A ce bloc de 15 octets on ajoute systématiquement un octet de Contrôle de Redondance cyclique (CRC) et un octet de validation. De cette façon, le minitel reçoit les informations par blocs de 17 octets.

L'octet de validation est obtenu en positionnant tous ses bits à 0. C'est 00000000 et il sert essentiellement à dépister rapidement les pertes d'information graves sur la ligne. On imagine les perturbations que peuvent occasionner les intempéries et en particulier la foudre tombant à proximité d'une ligne téléphonique. Dans un tel cas, il y a peu de chance pour que quelques uns des 8 zéros de l'octet de validation n'aient pas été remplaçés par des 1. Le modem du récepteur se rendra rapidement compte de l'erreur et demandera à l'émetteur de répéter le message.

Intéressons-nous maintenant à l'octet CRC. Le message initial est formé d'un bloc de 15 octets et correspond à une succession de $15 \times 8 = 120$ bits que nous noterons $a = (a_0, a_1, ..., a_{119})$ et qui sera identifié au polynôme

$$a(x) = a_0 + a_1 x + ... + a_{119} x^{119}.$$

Le code cyclique C choisi par les ingénieurs de la Direction Générale des Télécommunications est de longueur $n = 127 = 2^7 - 1$ et de polynôme générateur $g(x) = x^7 + x^3 + 1$. On peut vérifier que $g(x)$ est le polynôme minimal sur $\mathbb{F}_2$

d'une racine 127-ième primitive de l'unité, de sorte que C soit un code BCH primitif de distance construite $d = 2$.

Le codage du message a se fait ainsi :

- Multiplier $a(x)$ par x^7,
- Ecrire la division euclidienne de $x^7 a(x)$ par $g(x)$, soit

$$x^7 a(x) = q(x) g(x) + r(x) \quad \text{avec } \deg r(x) < 7,$$

- Prendre les coefficients de $r(x)$ pour les 7 premiers octets de l'octet CRC,
- Le 8-ième bit de l'octet CRC est un bit de parité portant sur le mot tout entier.

Ainsi le message envoyé, que l'on notera $M = (a(x) | r(x) | \zeta | v(x))$, est formé de $a(x)$ suivi du reste $r(x)$ et d'un bit de parité ζ, puis suivi de l'octet de validation $v(x)$.
On remarquera que sa transmission revient à transmettre le mot $c = q(x) g(x)$ du code C puisque

$$c = q(x) g(x) = x^7 a(x) + r(x)$$

est obtenu facilement à partie de M.

Pour transmettre le message codé M, on transmet le message $a(x)$ puis la seule difficulté revient à calculer rapidement le reste $r(x)$. Cela se fait facilement à l'aide d'un circuit informatique appelé "registre à décalage" et basé sur le fait que le reste $r(x)$ de la division de $x^7 a(x)$ par $g(x)$ s'obtient en remplaçant systématiquement les x^7 intervenant dans le polynôme $x^7 a(x)$ par des $x^3 + 1$.

Plus précisemment, le circuit contient 7 cases de mémoires $r_6, ..., r_0$ correspondant aux coefficients de $r(x)$. Le coefficient de plus haut degré de $x^7 a(x)$ est entré en r_0, puis on décale tous les chiffres des mémoires r_i vers r_{i+1} et l'on entre le second coefficient de $x^7 a(x)$ en r_0.
On continue ainsi jusqu'à épuisement des coefficients de $x^7 a(x)$, et avec la convention suivante : tout chiffre de r_6 ne pouvant pas être décalé sur la gauche sera ajouté aux chiffres des mémoires r_3 et r_0. On traduit par là que $x^7 = x^3 + 1$ modulo $g(x)$.
Au bout du compte, le reste $r(x)$ est lu dans les mémoires $r_6, ..., r_0$. La figure ci-dessous montre les décalages successifs lorsque $a(x) = x^2 + x$ pour obtenir $r(x) = x^5 + x^4 + x^2 + x$ à la dernière ligne.

r_6	r_5	r_4	r_3	r_2	r_1	r_0
						1
					1	1
				1	1	
			1	1		
		1	1			
	1	1				
1	1					
1			1			1
		1	1		1	1
	1	1		1	1	

Ce type de codage est très rapide et s'effectue au fur et à mesure de la lecture des données $a_0, ..., a_{126}$. Il se fait en temps réel.

6.2 Décodage et correction d'une erreur

Le message reçu est $M' = \left(a'(x) | r'(x) | \zeta' | v'(x)\right)$. Si l'octet de validation $v'(x)$ n'est pas nul, on demande au modem émetteur de retransmettre le message. On suppose maintenant que $v'(x)$ est nul.

Si l'on sait que M' ne contient pas plus d'une erreur, on est sûr de la corriger par le moyen ci-dessous. Dans la pratique on demandera la retransmission du message chaque fois que l'on a détecté une erreur multiple, en estimant que la probabilité pour qu'il y ait eu erreur multiple non détectée soit faible. Voici comment l'on procède :

Si le bit de parité ζ' et tous les bits de parité inclus dans la séquence $a'(x)$ sont justes, il n'y a pas eu d'erreur (avec une forte probabilité) et l'on peut considérer $a'(x)$ comme étant le message envoyé.

Si le bit de parité ζ' est faux, on cherche confirmation de l'erreur dans les bits de parités de $a'(x)$. Si un seul bit de parité de $a'(x)$ est faux, on estime qu'il y a eu une seule erreur portant sur l'octet correspondant de $a'(x)$, et l'on passe à sa correction (voir plus bas en $(*)$). Si tous les bits de parités de $a'(x)$ sont justes, on estime que l'unique erreur a eu lieu dans $r'(x)$, et l'on passe à sa correction. Par contre si plusieurs bits de parité sont faux, une erreur multiple est détectée et l'on demande la retransmission du message.

Supposons ici que ζ' soit faux sans qu'une erreur multiple ait été détectée, et procédons à la correction. L'erreur est dans $(a'(x) | r'(x))$, de sorte que $(a'(x) | r'(x))$ et $(a(x) | r(x))$ ne diffèrent que d'une coordonnée. Il existe un

entier p tel que le mot de code envoyé

$$c = q(x)\, g(x) = x^7 a(x) + r(x)$$

diffère du mot $c' = x^7 a'(x) + r'(x)$ effectivement reçu de x^p, i.e. $c - c' = x^p$. En prenant les classes modulo $g(x)$, on obtient $c' \equiv x^p \ (g)$. Dans cette relation, on connaît c' et donc aussi la classe de x^p modulo $g(x)$. Le lemme ci-dessous permet de retrouver x^p et par conséquent de savoir exactement où l'erreur s'est produite. La correction d'une erreur est possible.

Lemme 1 $1, x, ..., x^{126}$ *sont des classes distinctes de* $\mathbb{F}_q[x]/(g)$. *Autrement dit, si* $0 \leq p < q \leq 126$, *alors* $x^p \not\equiv x^q \ (g)$.

Preuve : Si g divise $x^q - x^p$ alors g divise $x^m - 1$ où $m = q - p \in \{1, ..., 126\}$. L'un des zéros de g est une racine 127-ième primitive de l'unité, que nous noterons α, et l'on a nécessairement $\alpha^m = 1$, ce qui est absurde. ■

Ce type de décodage qui s'effectue au fur et à mesure de l'arrivée des informations, est rapide puisque le cas le plus fréquent est celui de l'absence d'erreur. Alors ζ est juste et il n'y a rien à décoder, $a(x)$ ayant déjà été lu.

7 Compact Disc Audio : le code CIRC

Le code CIRC (Cross Interleaved Reed-Solomon Code) est un code correcteur d'erreur de type BCH particulièrement adapté aux problèmes du disque compact (CD). Ces problèmes se résument à retrouver l'information musicale, vidéo ou autre à partir d'un CD sale, rayé ou présentant une petit défaut de fabrication. Les normes de Philips exigent une correction d'une rayure de $0,2$ mm, et le code CIRC permettra de corriger des paquets d'erreurs ou d'effacements de 4096 bits consécutifs, ce qui correspond approximativement à une rayure d'un millimètre. Dans ce paragraphe, on se réfère à [7].

7.1 Codes raccourcis

Soient C un code linéaire $[n, k, d]$ et $t \leq k$. On suppose que C possède une matrice génératrice G de la forme $G = (A|B)$ où A est une matrice de taille $k \times t$ et de rang t, et l'on pose $C' = \{c \in C \,/\, c_1 = ... = c_t = 0\}$ et

$$C'' = \left\{(c_{t+1}, ..., c_n) \in \mathbb{F}_q^{n-t} \,/\, (0, ..., 0, c_{t+1}, ..., c_n) \in C'\right\}.$$

Le code C'', dont les mots sont obtenus en effaçant les t premières coordonnées des mots de C dont les t premières coordonnées sont nulles, est appelé **code raccourci** de C.

Théorème 7 *C'' est un code $[n-t, k-t, d'' \geq d]$.*

Preuve : L'application

$$\begin{array}{rccc} p: & C' & \to & C'' \\ & (0,...,0,c_{t+1},...,c_n) & \mapsto & (c_{t+1},...,c_n) \end{array}$$

est un isomorphisme d'espaces vectoriels, donc $\dim C'' = \dim C'$.
Notons $G = [g_1, ..., g_n]$. Les mots du code C sont décrits par les produits xG quand x parcourt $\mathbb{F}_q$, et l'on peut écrire $xG = ((x|g_1), ..., (x|g_n))$ où $(.|.)$ désigne la forme bilinéaire symétrique non dégénérée standard sur $\mathbb{F}_q^k$. Le code C' est l'image par l'application linéaire injective $x \mapsto xG$ de l'orthogonal $(\mathrm{Vect}(g_1, ..., g_t))^\perp$ dans $\mathbb{F}_q^k$. Comme g_1, ..., g_t est un système libre (en effet $\mathrm{rg}\, A = t$), on déduit

$$\dim C' = \dim (\mathrm{Vect}(g_1, ..., g_t))^\perp = k - t$$

et C'' sera bien un code $[n-t, k-t]$.
Enfin tout mot $c'' = (c_{t+1}, ..., c_n)$ de C'' provenant d'un mot

$$c = (0, ..., 0, c_{t+1}, ..., c_n)$$

de C dont les t premières colonnes sont nulles, son poids sera

$$w(c'') = w(c) \geq d.$$

La distance minimale de C'' sera $\geq d$. ■

Le code raccourci d'un code de Reed-Solomon conserve la même distance minimale. En effet, si d désigne la distance minimale du code de Reed Solomon C, et d'' celle du code raccourci C'', le Théorème précédent, et la borne de Singleton permettent d'écrire

$$d \leq d'' \leq (n-t) - (k-t) + 1 = d.$$

L'intérêt des codes raccourcis est de fournir une méthode construction de k nouveaux codes de longueurs plus petites à partir d'un seul code de paramètres $[n, k, d]$ tout en conservant la capacité de correction du code, celle-ci étant directement liée à la distance minimale. C'est d'autant plus important que de nombreuses classes de codes, tels les codes BCH, ont des paramètres très spéciaux qu'il s'agit d'adapter aux diverses contraintes techniques, et que les 3 paramètres n, k, d d'un code ne sont pas indépendants entre eux mais liés par de nombreuses relations (telles les bornes de Hamming ou celle de Singleton). En règle générale, fixer 2 de ces paramètres revient à déterminer le troisième.

7.2 Codes démultipliés

Dans la grande majorité des cas, des problèmes de lecture d'un disque compact surviennent lorsque le disque est sale, lorsque sa surface est rayée ou présente un défaut de fabrication. La perte d'information a lieu sur un nombre "important" de bits contigus, donnant naissance à ce que l'on appelle des "paquets d'erreurs". La correction de paquets d'erreurs est donc tout à fait typique des codes correcteurs présents sur un lecteur de CD, et fait en particulier appel aux codes démultipliés.

Soit C un code $[n,k]$ sur $\mathbb{F}_q$. Notons $q=p^m$ où p est premier, et appelons α un élément primitif de $\mathbb{F}_q$ (i.e. une racine primitive $(q-1)$-ième de l'unité dans $\mathbb{F}_q$). On sait que $\left(1,\alpha,...,\alpha^{m-1}\right)$ est une base du $\mathbb{F}_p$-espace vectoriel $\mathbb{F}_q$. Si $(x_{i1},...,x_{im})$ désignent les coordonnées de $x_i\in\mathbb{F}_q$ dans cette base, on peut définr l'application :

$$\begin{array}{rccc}\Psi: & \mathbb{F}_q^n & \to & \mathbb{F}_p^{nm} \\ & (x_1,...,x_n) & \mapsto & (x_{11},...,x_{1m},x_{21},...,x_{n1},...,x_{nm})\,.\end{array}$$

Ψ est un isomorphisme de $\mathbb{F}_p$-espaces vectoriels appelé **démultiplication**, et la bijection réciproque Ψ^{-1} est appelée **contraction**. Le code C' démultiplié de C n'est autre que l'image $\Psi(C)$ de C par Ψ. C'est un code de paramètres $[nm,nk]$ sur $\mathbb{F}_p$. Si d désigne la distance minimale de C, rappelons que C corrige $e=\left[\frac{d-1}{2}\right]$ erreurs. Un mot $x=(x_1,...,x_n)$ de longueur n de C sera démultiplié en un mot x' de longueur nm de C' suivant le schéma

$$\begin{array}{lccccccc}x: & x_1 & & x_2 & ... & x_i & ... & x_n \\ x': & x_{11},...,x_{1m} & x_{21},...,x_{2m} & ... & x_{i1},...,x_{im} & ... & x_{n1},...,x_{nm}\end{array}$$

S'il y a moins de $m(e-1)+1$ erreurs consécutives dans le mot x', il y aura moins de e coordonnées erronées dans x, et la correction de x est possible. Autrement dit :

Théorème 8 *Si un code sur $\mathbb{F}_{p^m}$ corrige e erreurs, son démultiplié sur $\mathbb{F}_p$ corrigera jusqu'à $m(e-1)+1$ erreurs consécutives.*

7.3 Effacements

Il arrive souvent que le lecteur ne puisse déterminer la valeur du bit qu'il est en train de lire. On dit que l'on a affaire à un effacement. Le problème des effacements est un peu différent de celui de l'erreur dans le sens où le décodeur sait qu'il y a eu disparition d'information à l'endroit occupé par le symbole effacé. Dans le cas d'une erreur, le décodeur est à priori incapable de dire si le symbole proposé est vrai ou faux. Cette différence est mise en valeur dans les deux résultats suivants :

Théorème 9 *Un code (éventuellement non linéaire) de distance minimale d peut corriger $f = d - 1$ effacements.*

Preuve : Si c et c' sont deux mots de code qui possèdent $n - (d-1)$ coordonnées identiques, alors c et c' diffèrent sur au plus $d-1$ coordonnées et $d(c, c') \leq d - 1$. Cela entraine $c = c'$. ■

Théorème 10 *Un code (éventuellement non linéaire) de longueur n et de distance minimum $d \geq 2e + f + 1$ corrige e erreurs et f effacements.*

Preuve : Soit $x = (x_1, ..., x_n) \in C$ et x' le mot entaché de $p \leq e$ erreurs et $q \leq f$ effacements. Soit H l'ensemble des mots de C identiques à x sur les $n - q$ coordonnées non effacées. Il faut montrer l'existence d'un seul mot du code C situé à une distance $\leq e$ de H. Raisonnons par l'absurde : si y et z sont deux mots de C et si $h, k \in H$ vérifient $d(y, h) \leq e$ et $d(z, k) \leq e$. Alors

$$d \leq d(y, z) \leq d(y, h) + d(h, k) + d(k, z) \leq 2e + q \leq 2e + f$$

ce qui est contraire à l'hypothèse. ■

7.4 Entrelacements

L'objectif est encore ici de corriger de gros paquets d'erreurs. Considérons t mots $x_1, x_2, ..., x_t$ d'un code C et écrivons ces mots sur t lignes. On obtient le tableau

$$\begin{array}{lcccccc} \text{Mot } x_1 & : & x_{11} & x_{12} & ... & x_{1n} \\ \text{Mot } x_2 & : & x_{21} & x_{22} & ... & x_{2n} \\ \vdots & & \vdots & & & \vdots \\ \text{Mot } x_t & : & x_{t1} & x_{t2} & ... & x_{tn} \end{array}$$

Au lieu d'envoyer ce tableau ligne après ligne, on décide de l'envoyer colonne après colonne. On considère par conséquent le mot

$$x_{11}x_{21}...x_{t1}x_{12}x_{22}...x_{t2}...x_{tn}$$

obtenu par l'entrelacement de t mots de C. L'ensemble des ces mots forme un code C' de longueur nt et de dimension kt, appelé **code entrelacé de profondeur t du code C**. Si C corrige l erreurs consécutives, l'examen du tableau ci-dessus permet de voir que C' corrigera des paquets d'erreurs de longueur lt, ce qui constitue une amélioration substancielle.

Présentons maintenant une autre technique d'entrelacement très efficace. Considérons toujours les mêmes mots $x_1, ..., x_t$ de C. La technique d'entrelacement avec un retard r consiste dans un premier temps à construire le tableau suivant

dans lequel la première ligne est formée des premières coordonnées des mots $x_1, ..., x_t$, la seconde ligne est formée des secondes coordonnées de ces mots mais décalés sur la droite de r symboles 0, et ainsi de suite sans oublier de décaler de r symboles vers la droite à chaque passage à la ligne.

$$\begin{array}{cccccccccccc}
\underline{x_{11}} & x_{21} & ... & ... & ... & ... & x_{t1} & 0 & 0 & ... & & 0 \\
0 & 0 & ... & 0 & \underline{x_{12}} & x_{22} & ... & ... & x_{t2} & 0 & & \vdots \\
\vdots & & & & & & & & & & & 0 \\
0 & 0 & ... & ... & ... & ... & ... & 0 & \underline{x_{1n}} & x_{2n} & ... & x_{tn}
\end{array}$$

On transmet ensuite les colonnes de ce tableau pour obtenir un nouveau mot qui appartient à un code C''. C'' est le **code entrelacé de profondeur** t **avec un retard** r **du code** C.

Les symboles du premier mot x_1 ont été soulignés dans le tableau. r colonnes successives du tableau ne contiennent qu'un symbole figurant dans x_1. Par suite, si l'on perturbe les symboles de lr colonnes de ce tableau auxquels on peut rajouter l symboles "limitrophes", on touche à moins de l symboles consécutifs de chacun des mots $x_1, ..., x_t$ de C. Cela signifie que si C corrige l erreurs consécutives, alors C'' corrigera $l\,(rn+1)$ erreurs consécutives.

Exemple : Pour écrire l'entrelacement des trois mots

$$\begin{array}{lcccc}
\text{Mot } a & : & a_1 & a_2 & a_3 & a_4 \\
\text{Mot } b & : & b_1 & b_2 & b_3 & b_4 \\
\text{Mot } c & : & c_1 & c_2 & c_3 & c_4
\end{array}$$

avec un retard $r = 2$, on construit le tableau

$$\begin{array}{ccccccccc}
a_1 & b_1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & a_3 & b_3 & c_3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & a_4 & b_4 & c_4
\end{array}$$

et le mot entrelacé avec retard sera

$$\underbrace{a_1000b_1000c_1a_2000b_2000c_2}_{18 \text{ symboles}}a_3000b_3000c_3a_4000b_4000c_4.$$

Ici $(n, t, r) = (4, 3, 2)$ et si C corrige des paquets de $l = 2$ erreurs, alors C'' corrigera bien des paquets de 18 erreurs.

7.5 Code CIRC

Soit C le code de Reed-Solomon de polynôme générateur

$$g(x) = (x-\alpha)\left(x-\alpha^2\right)\left(x-\alpha^3\right)\left(x-\alpha^4\right)$$

où α est un élément primitif de $\mathbb{F}_{256}$. C est un code $[255, 251, 5]$ sur $\mathbb{F}_{256}$ qui permet de construire deux codes raccourcis de distance minimale encore égale à 5 (cf §7.1) :

-Un code C_1 de paramètres $[28, 24, 5]$,
-Un code C_2 de paramètres $[32, 28, 5]$.

Après échantillonage et quantification, le signal sonore se présente en trames de 24 octets. Ces 24 octets servent de symboles d'information du code C_1 et donnent naissance à des mots de 28 octets. Ces mots sont ensuite entrelacés avec une profondeur de 28 et un retard de 4 pour servir de symboles d'information du code C_2. Voyons maintenant comment s'opère le décodage des trames de longueur 32 lues par l'appareil...

• Action de C_2 : Le code C_2 est capable de corriger 2 erreurs, mais on l'utilise pour n'en corriger qu'une. Il peut détecter 3 erreurs. Si l'on note x le mot de $\mathbb{F}_{256}^{28}$ lu par l'appareil, trois cas sont possibles :

a1) Si $x \in C_2$, on estime que x représente le mot correct et qu'il n'y a pas eu d'erreur,

a2) Si x est à une distance de C_2 égale à 1, on estime qu'il y a eu une seule erreur, et le code C_2 la corrige,

a3) Dans tous les autres cas, on remplace x par 28 effacements qui seront désentrelacés puis envoyés sur le décodeur de C_1.

Les cas a1 et a2 peuvent donner lieu à une erreur de décodage, mais la probabilité d'erreur est infime. Pour le voir, supposons que tous les mots de $\mathbb{F}_{256}^{28}$ soient équiprobables, ce qui constitue une hypothèse pénalisante. Le mot x sera mal décodé par le processus a1 ou a2 dès que x appartient à l'une des boules fermées de rayon 1 centrée sur un mot du code C_2 sans être le centre de cette boule. Une boule fermée de rayon 1 de $\mathbb{F}_{256}^{28}$ contient $1 + 255.C_{32}^1$ mots. C_2 contient 256^{28} mots et $\mathbb{F}_{256}^{32}$ en contient 256^{32}. La probabilité d'une erreur au décodage sera donc majorée par :

$$\frac{256^{28}\left(1 + 255.C_{32}^1\right)}{256^{32}} \simeq 1,9.10^{-6}$$

ce qui est négligeable. Le même raisonnement pratiqué dans le cas où l'on utiliserait la double capacité de correction d'erreurs de C_2 fournirait le majo-

rant suivant bien trop grand pour être retenu :

$$\frac{256^{28}\left(1+255.C_{32}^{1}+255^{2}.C_{32}^{2}\right)}{256^{32}} \simeq 7,5.10^{-3}.$$

• Action de C_1 : Dans les cas a1 et a2, le mot, éventuellement corrigé par C_2, est désentrelaçé puis décodé par C_1 pour fournir l'information originelle. Plaçons-nous maintenant dans le cas a3. Les 28 symboles du mot ont été effacés et constituent une colonne de la matrice d'entrelacement. Pour le voir, il faut se référer au §7.4 et bien comprendre qu'ici, nous avons entrelaçé 28 mots de $\mathbb{F}_{256}^{28}$ donnés en lignes dans le tableau

$$\begin{array}{lcccccc}
\text{Mot } x_1 & : & x_{1,1} & x_{1,2} & \ldots & x_{1,28} \\
\text{Mot } x_2 & : & x_{2,1} & x_{2,2} & \ldots & x_{2,28} \\
\vdots & & \vdots & & & \vdots \\
\text{Mot } x_{28} & : & x_{28,1} & x_{28,2} & \ldots & x_{28,28}
\end{array}$$

et que la matrice d'entrelacements est

$$\begin{array}{ccccccccccc}
\underline{x_{1,1}} & x_{2,1} & x_{3,1} & x_{4,1} & \ldots & \ldots & & & & & x_{28,1} \\
0 & 0 & 0 & 0 & \underline{x_{1,2}} & x_{2,2} & x_{3,2} & x_{4,2} & \ldots & & x_{28,2} \\
\vdots & & & & & & & & \ddots & & \vdots \\
0 & 0 & 0 & 0 & 0 & \ldots & \ldots & 0 & \underline{x_{1,28}} & x_{2,28} \ldots & x_{28,28}
\end{array}$$

C_1 corrige 4 effacements donc sera capable de corriger $4 \times 4 = 16$ colonnes entières de cette matrice d'entrelacements. Chacune de ces 16 colonnes de 28 symboles provient d'un mot de 32 symboles (soit 32×8 bits 0 ou 1) avant traitement par le code C_2. Par suite, le code CIRC sera capable de corriger des effacements contigüs de longueur $16 \times 32 \times 8 = 4096$ bits sur la surface du disque compact, ce qui correspond à des rayures d'environ $1,23$ mm sur le disque et se trouve bien au dessus des normes de Philips ($0,2$ mm).

References

[1] Arnoux P., Minitel, codage de l'information et corps finis, Pour la Science n^{0}125, mars 1988

[2] Epreuve de Mathématique de l'Informatique de l'Agrégation externe de mathématiques, session 1996, rapport du jury, CNDP, 1996.

[3] Lidl R. & Niederreiter H., "Finite Fields", Encyclopedia of Mathematics and its Applications, vol. 20, Addison-Wesley Pubblishing Company, 1983.

[4] Mercier D-J., "Cryptographie Classique et Cryptographie publique à clé révélée", A.P.M.E.P., Bulletin n^{o}406 de septembre-octobre 1996.

[5] Mac William F.J. & Sloane N.J.A., "The theory of Error-Correcting Codes", North-Holland Mathematical Library, vol.16, 2nd reprint 1983.

[6] Papini O. & Wolfmann J., "Algèbre discrète et codes correcteurs", Springer-Verlag, Mathématiques & Applications n^{0}20, 1995.

[7] Zanotti J.P., "Codage d'un Signal Audionumérique sur un Support à Lecture Optique, Erreurs au décodage et Codes MDS", Mémoire de DEA, Université d'Aix-Marseille II, Faculté des Sciences de Luminy, 1992.