

# AUTOUR DU DÉCODAGE DES CODES DE BOSE, RAY-CHAUDHURI ET HOCQUENGHEM

PHILIPPE LANGEVIN

ABSTRACT. Au début des années soixante A. Hocquenghem (1959) et un an plus tard, R. C. Bose et D. K. Ray-Chaudhuri découvrent une classe de codes importantes qui figure dans tous les manuels de codage : la classe BCH. La littérature ne manque pas sur ce sujet, signalons notamment le petit livre de van Lint [2] et la bible des codes correcteurs [1] dont les index renvoient tous les deux à la page 80 pour cette notion. Des points de vue pédagogique et pratique, la classe des codes BCH ne manque pas d'intérêt puisqu'elle offre des codes «décodables» de capacité de correction relativement élevée ce qui est conforme à l'objectif principal de cette note : présenter aux étudiants de troisième cycle d'informatique et de mathématiques quelques exemples d'algorithmes de décodages non triviaux c'est-à-dire non fondés sur les tables de syndromes. De tous les codes algébriques, les codes de Reed-Muller binaires sont sans aucun doute les plus faciles à mettre en oeuvre. Nous utiliserons la transformée de Fourier pour écrire un algorithme «diviser pour régner» particulièrement efficace. L'étude des coefficients de Fourier d'une fonction booléenne conduit aux fonctions courbes et hautement non-linéaires liées à la détermination du rayon de recouvrement des codes de Reed-Muller affines. Des notions capitales pour l'étude de la méthode de chiffrement au fil de l'eau qui utilise des suites de récursions linéaires engendrées par des registres à décalage, filtrés par des fonctions non-linéaires. Un contexte cryptographique où se pose le problème de calculer la complexité linéaire d'une suite donnée. Nous présenterons l'algorithme de Berlekamp-Massey répondant à cette question. Il s'agit d'une procédure initialement conçue par Berlekamp pour décoder les codes BCH. Il existe un algorithme plus facile à présenter pour le décodage de ces codes, et au moins aussi efficace : l'algorithme Euclidien. Comme son nom l'indique, il s'agit de détourner l'algorithme d'Euclide de son utilisation normale, le calcul du plus grand diviseur commun de deux polynômes, au profit du décodage des codes. Le sujet nécessite l'implantation des opérations élémentaires sur les corps finis : addition, produit, inversion, racine de l'unité, polynôme minimal etc. . . Pour les notions fondamentales : corps finis, polynômes etc. . . Je renvoie le lecteur informaticien vers le cours de mes collègues Papini et Wolfmann [5], et le lecteur mathématicien vers l'encyclopédie de Lidl et Niederreiter [4]. Nous profiterons de l'occasion pour effleurer quelques points fondamentaux de la théorie des codes correcteurs d'erreurs : NP-complétude du décodage des codes, bornes de Singleton et Varshamov-Gilbert, classe de bons codes, codes résidus quadratiques. Un petit échantillon de l'ensemble des questions issues de la théorie des codes correcteurs qui pourront être approfondies par les étudiants de troisième cycle de mathématiques.

---

*Date:* <http://langevin.univ-tln.fr/CDE/bch.pdf>

Première publication : Octobre 2000; dernière modification : Avril 2014.

## CONTENTS

1. Classes de bon codes	2
2. Borne de Singleton	3
3. Borne de Varshamov-Gilbert	4
4. bons codes explicites	4
5. Codes de Reed-Muller	4
6. Non-linéarité	7
7. Chiffrement au fil de l'eau	7
8. Algorithme de Berlekamp-Massey	7
9. Cyclotomie	9
10. Classe BCH	11
11. Implantation en Langage C	12
12. BCH 2-correcteur	13
13. Codes primitifs	13
14. Résidus quadratiques	14
15. Algorithme d'Euclide	15
16. Décodage des BCH	16
References	17

## 1. CLASSES DE BON CODES

La problématique du codage-décodage des codes correcteurs d'erreurs est balisée par deux grands théorèmes. Un résultat positif de Claude Shannon qui affirme l'existence de certaines classes de bons codes, et un résultat d'Alexander Vardy qui affirme l'impossibilité pratique de déterminer la capacité de correction d'un code arbitraire de grande dimension.

**Théorème 1.** *Quelque soit le corps de base  $K$ , il existe une suite de codes  $K$ -linéaires  $[N_n, K_n, D_n]$  tels que*

$$\lim_{n \rightarrow \infty} \frac{K_n}{N_n} \neq 0 \quad \text{et} \quad \lim_{n \rightarrow \infty} \frac{D_n}{N_n} \neq 0,$$

*on parle de suite de bons codes.*

*Proof.* La preuve est faite un peu plus loin. □

Le problème de décodage des codes correcteurs se formule en une question de décision. Étant donné : un corps fini  $K$ , deux entiers  $k$  et  $n$ , une matrice  $H$  à coefficients dans  $K$  composée de  $n$  lignes et  $(n-k)$  colonnes, un troisième entier  $w$ , et (enfin) un vecteur  $s$  de  $K^{n-k}$ , existe-t-il un mot  $x \in K^n$  tel que :

$$\text{wt}(x) \leq w, \quad xH = s.$$

**Théorème 2** (Berlekamp, McEliece, Van Tilborg, 1970). *Le problème de décodage des codes correcteurs est NP-complet.*

*Proof.* Il s'agit de faire une réduction au problème des mariages à trois. □

Le premier théorème montre qu'il est possible de construire des codes performants, mais ce n'est pas un résultat effectif. Dans tous les cas, le second théorème montre que l'on ne peut pas se contenter d'une méthode aléatoire

pour construire des bons codes au risque d'être incapable de les décoder ! Pour des raisons pratiques, on privilégie les mauvais codes décodables aux bons codes indécodables...

Les mêmes auteurs montrent aussi la NP-complétude du problème de décision résultant de la question : Existe-t-il un mot  $x \in K^n$  de poids  $w$  tel que  $xH = 0$  ? Mais ils terminent leur article en conjecturant la NP-complétude du problème dit de la distance minimale, formulé comme ci-dessus en remplaçant l'égalité sur le poids de  $x$  par une inégalité. Une conjecture qui a été démontrée récemment par Alexander Vardy.

**Théorème 3** (Vardy, 1997). *Le problème de la distance minimale : existe-t-il un mot  $x \in K^n$  tel que  $xH = 0$  et  $\text{wt}(x) \leq k$ , est NP-complet.*

*Proof.* □

## 2. BORNE DE SINGLETON

Les paramètres fondamentaux d'un  $[n, k, d]$  code linéaire satisfont à l'inégalité de Singleton

$$(1) \quad d \leq n - k + 1$$

Convenons d'appeler «dimension» d'un code non linéaire le logarithme de son cardinal.

**Exercice 1.** *Montrer qu'un code non linéaire vérifie la borne de Singleton.*

Les codes linéaires pour lesquels l'inégalité est une égalité sont des codes MDS pour (Maximal Distance Separable). Le dual d'un code MDS est un code MDS.

**Théorème 4.** *La distribution de poids d'un code MDS est complètement déterminée par  $n$ ,  $k$  et  $q$ . En particulier*

$$A_d = (q-1)C_n^{k-1}, \quad A_{d+1} = (q-1)C_n^{k-2}(q-n+k+1)$$

*Proof.* □

On déduit du théorème ci-dessus que, pour  $q$  fixé, il n'existe pas de code MDS arbitrairement long. Un constat qui nous amène à l'une des plus importantes conjectures de la théorie des codes correcteurs.

**Conjecture 1.** *La longueur maximale d'un code MDS non trivial sur un corps à  $q$  éléments est  $q+2$  si  $q$  est pair et  $k = \pm 3 \pmod n$ , sinon c'est  $q+1$ .*

On note  $A_q(n, d)$  le cardinal du plus grand code de longueur  $n$  et de distance minimale  $d$  sur le corps à  $q$  éléments. Pour comparer les classes de codes, on introduit la fonction :

$$(2) \quad \alpha_q(\delta) = \limsup_{n \rightarrow +\infty} \frac{1}{n} \log_q A_q(n, \delta n)$$

Toute borne donne lieu à une borne asymptotique, par exemple, la borne de Singleton montre que

$$\alpha(\delta) \leq 1 - \delta$$

On sait qu'il n'existe pas de codes MDS arbitrairement long : la majoration ci-dessus ne vaut pas grand-chose.

### 3. BORNE DE VARSHAMOV-GILBERT

Conformément à l'usage, notons  $V_q(n, d)$  le volume d'une boule de rayon  $d$  pour la distance de Hamming de  $\mathbf{F}_q^n$ . Un code  $C$  est dit maximal lorsque sa distance minimale est plus grande que celle de tous les codes qui le contiennent. En d'autres termes les boules de rayons  $d - 1$  centrées sur les mots de  $C$  recouvre l'espace de Hamming tout entier et donc

$$q^n \leq V_q(n, d - 1) |C|$$

**Lemme 1.** *Soient  $n, k$  et  $d$  trois entiers. Si  $q^{n-k} > V_q(n, d - 1)$  alors il existe un  $[n, k, d]$  code linéaire.*

*Proof.* Raisonnement par induction sur  $k$ . On construit une chaîne de codes de proche en proche, en partant d'un  $[n, 1, d]$  code arbitraire. Si  $l < k$  alors le  $[n, l, d]$  code de cette chaîne n'est pas maximal et on peut trouver un mot  $x$  à distance  $d$  qui permet d'étendre le dernier code en un  $[n, l + 1, d]$  code.  $\square$

Le lemme précédent donne sur le champ une minoration de la dimension du meilleur code linéaire de distance  $d$  et de longueur  $n$ , c'est la borne de Varshamov-Gilbert, ou encore borne d'empilement des sphères.

$$(3) \quad n - \log_q V_q(n, d - 1) < \log_q A_q(n, d)$$

En passant à la limite, pour  $0 < \delta < \frac{q-1}{q}$ , on obtient la borne de Varshamov-Gilbert asymptotique, en fonction de la «fonction d'entropie»

$$\begin{aligned} \alpha_q(\delta) &\geq 1 - \lim_{+\infty} \frac{\log_q V_q(n, n\delta)}{n} \\ &= 1 - H_q(\delta) \end{aligned}$$

avec

$$H_q(\delta) = \delta \log_q(q - 1) - \delta \log_q \delta - (1 - \delta) \log_q(1 - \delta).$$

Une sympathique expression déduite des formules d'approximation de Stirling de la fonction factorielle.

### 4. BONS CODES EXPLICITES



Il s'agit de parler de la famille des codes de Justesen : concaténation de codes par des codes de Reed-Solomon. Des codes de Goppa géométriques.

### 5. CODES DE REED-MULLER

Soit  $m$  un entier, on pose  $n = 2^m$  et on se donne une énumération  $P_1, P_2, \dots, P_n$  des  $m$ -uplets de  $\mathbf{F}_2^m$ . Une fonction booléenne  $f$  est une application de  $\mathbf{F}_2^m$  dans  $\mathbf{F}_2$  que nous représentons par une table de vérité :

$$f \mapsto [f(P_1), f(P_2), \dots, f(P_n)]$$

En particulier, les fonctions booléennes forment un espace de dimension  $n$  sur  $\mathbf{F}_2$ . Parmi les fonctions les plus simples, figurent les constantes et les fonctions coordonnées  $X_i$  qui projettent  $(a_1, a_2, \dots, a_m)$  sur  $a_i$ . Pour chaque partie  $S \subset \{1, 2, \dots, m\}$ , on note  $X_S$  le monôme  $\prod_{i \in S} X_i$ . On constate que l'indicatrice d'un point  $a$ , notée  $\delta_a$ , s'exprime polynomialement à partir des  $X_i$  puisque

$$\delta_a(x) = \prod_{i=1}^m (x_i + b_i + 1)$$

En conséquence toute fonction booléenne possède une représentation polynomiale sommes de monômes  $X_S$ . Plus précisément,

**Proposition 1.** *Le système des monômes  $X_S$  ( $S \subseteq \{1, 2, \dots, m\}$ ) est une base de l'espace des fonctions booléennes. On peut parler du degré d'une fonction. L'ensemble des applications de degré au plus  $k$  forme un espace vectoriel de dimension  $\sum_{i=0}^k C_m^i$ .*

L'espace des tables de vérités correspondant aux fonctions de degré au plus  $k$  est un code binaire de longueur  $n$ , de dimension  $\sum_{i=0}^k C_m^i$  et de poids minimum  $2^{m-k}$ . Ces codes apparaissent dans les travaux de Muller et Reed dans les années cinquante, on les note  $RM(k, m)$ .

**Exercice 2.** *Calculer le poids minimal de  $RM(k, m)$ .*

Le groupe des transformations affines de  $\mathbf{F}_2^m$  agit sur les fonctions, et même sur les fonctions de degré au plus  $k$ . En d'autres termes, le groupe des applications affines  $GA(\mathbf{F}_2, m)$  est inclus dans le groupe d'automorphisme de  $RM(k, m)$ .

**Exercice 3.** *Montrer que le groupe d'automorphismes du code  $RM(k, m)$  est exactement égal à  $GA(\mathbf{F}_2, m)$ .*

Les codes de Reed-Muller sont importants pour des raisons historiques. Les codes de Reed-Muller du premier ordre ont connu des heures de gloire puisque le code  $RM(1, 5)$  fût embarqué à bord de la sonde MARINER 9 pour l'exploration de Mars en janvier 1972. Tous les paramètres standards des codes de Reed-Muller du premier ordre sont connus sauf un, le rayon de recouvrement. Depuis les travaux de Rothaus, Dillon etc, la détermination du rayon de recouvrement de  $RM(1, m)$  est devenue une question centrale en cryptographie, un de mes sujets de recherche favoris, consulter mon HDR en ligne [3].

Un mot de  $RM(1, m)$  est complètement défini par un couple  $(a, b) \in \mathbf{F}_2^m \times \mathbf{F}_2$  auquel correspond la fonction

$$\Phi_{a,b}(x) = a \cdot x + b = a_1 x_1 + a_2 x_2 + \dots + a_m x_m + b$$

Mis de côté les constantes, tous les mots de  $RM(1, m)$  sont de poids  $2^{m-1}$ . On définit le coefficient de Fourier d'une fonction booléenne  $f$  en un point  $a \in \mathbf{F}_2^m$  par

$$\hat{f}(a) = \sum_{x \in \mathbf{F}_2^m} (-1)^{f(x) + ax}$$

la distance de entre  $f$  et la fonction affine  $\Phi_{a,b}$  est donnée par

```

ALGORITHME  FOURIER( f )
DONNEES   f : fonction booleenne.
VARIABLE  i, j, n : indice.
          t : entier;
DEBUT
  n := TAILLE(f);
  SI ( n >= 1 ) ALORS
    n := n/2;
    FOURIER( f );
    FOURIER(f+n);
    i := 0;
    j := n;
    TANTQUE ( i<n ) FAIRE
      t := f[ i ];
      f[ i ] := t + f[j];
      f[ j ] := t - f[j];
      INC(i);
      INC(j);
    FTQ
    SINON f[0] = 1 - 2*f[0]
  FSI
FIN

```

FIGURE 1. Transformation de Fourier.

$$d(f, \Phi_{a,b}) = 2^{m-1} - \frac{1}{2} \hat{f}(a).$$

En particulier, le spectre de Fourier d'une fonction du Reed-Muller d'ordre 1 est constitué d'une seule valeur non nulle :  $\pm 2^m$ .

**Exercice 4.** Montrer que  $\sum_a (\hat{f}(a))^2 = n^2$ . En déduire une majoration du rayon de recouvrement du code de Reed-Muller du premier ordre.

**Exercice 5.** Calculer les coefficients d'une forme quadratique. En déduire la valeur du rayon de recouvrement du code de Reed-Muller du premier ordre lorsque  $m$  est pair.

Identifions  $\mathbf{F}_2^m$  à  $\mathbf{F}_2^{m-1} \times \mathbf{F}_2$  et mettons  $a \in \mathbf{F}_2^m$  sous la forme  $(a', a_m)$  et notons  $u(x)$  la fonction de  $m-1$  variable obtenue par restriction de  $f$  à l'hyperplan  $x_m = 0$  i.e.  $u(x) = f(x, 0)$ ; De même notons  $v(x) = f(x, 1)$ . La relation

$$\hat{f}(a', a_m) = \hat{u}(a') + (-1)^{a_m} \hat{v}(a')$$

permet d'utiliser le principe «diviser pour régner» afin d'obtenir un algorithme (??) de complexité  $\Theta(mn)$  pour calculer «sur place» le spectre de Fourier de  $f$ .

**Exercice 6.** Supprimer la récursivité de l'algorithme proposé. En déduire un algorithme de décodage du code de Reed-Muller du premier ordre.

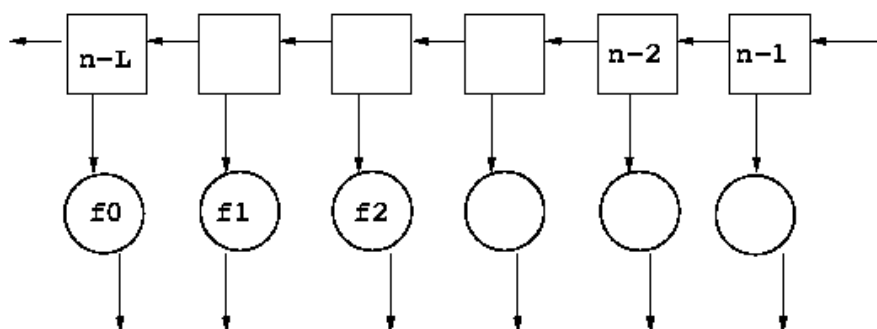


FIGURE 2. Registre à décalage

## 6. NON-LINÉARITÉ



Introduction à la non linéarité, fonctions courbes.

## 7. CHIFFREMENT AU FIL DE L'EAU



registres à décalage.

## 8. ALGORITHME DE BERLEKAMP-MASSEY

On calcule dans le corps à deux éléments. La suite produite par le registre 'à décalage de la figure (FIG.2) vérifie une relation de récurrence linéaire d'ordre  $L$ . Pour tout entier naturel  $k$  :

$$s_{k+L} = \sum_{i=0}^{L-1} f_i s_{k+i}$$

ou encore, pour  $N$  supérieur à  $L$ ,

$$s_N = \sum_{i=1}^L f_{L-i} s_{N-i}$$

Ces suites qui sont manifestement périodiques à partir d'un certain rang forment un sous-espace de dimension au plus  $L$ . Il existe un polynôme  $r(T)$  tel que

$$(4) \quad S(T) = \sum_{n \geq 0} s_n T^n = \frac{r(T)}{g(T)}$$

où  $g(T)$  désigne le polynôme réciproque de  $f(T)$ .

*Proof.* Il suffit de montrer que pour  $n \geq L$ , le coefficient de degré  $N$  du produit  $f(T)S(T)$  est nul.  $\square$

Le polynôme de degré minimal  $g(T)$  satisfaisant à l'équation (4) s'appelle le polynôme caractéristique de  $S$ , et nous dirons que le polynôme  $r(T)$  qui lui correspond est le polynôme d'état. On dit que  $(r_n, g_n)$  est une approximation minimale à l'ordre  $n$  de  $S$  si

$$(5) \quad g_n(T)S(T) = r_n(T) \pmod{(T^n)}, \quad \deg r_n < \deg g_n.$$

avec  $g_n(T)$  de degré minimal. On note  $L_n$  le degré de  $g_n$ . La suite  $L_n$  est croissante, majorée par  $n$ . Il n'y a pas unicité.

**Lemme 2.** *La condition supplémentaire  $n \geq 2L_n$  implique l'unicité.*

*Proof.* Supposons deux approximations à l'ordre  $n$  de degré inférieur à  $n/2$ .

$$g(T)S(T) \equiv r(T) \pmod{T^n} \quad \text{et} \quad \dot{g}(T)S(T) \equiv \dot{r}(T) \pmod{T^n}$$

On remarque que  $T$  ne divise ni  $g$ , ni  $\dot{g}$ . Il suit une relation :

$$g\dot{r} \equiv \dot{g}r \pmod{T^n}.$$

□

L'algorithme de Berlekamp-Massey de la figure [BM] construit les approximations successives d'une suite de récursion linéaire. Il est fondé sur les deux faits qui suivent.

**Lemme 3.** *Si de plus  $L_n < L_{n+1}$  alors*

$$n - L_n \leq L_{n+1}$$

*Proof.* On suppose que  $n+1-L_n \geq 0$ . Notons  $f(T)$  le réciproque de  $g_{n-1}(T)$  et  $f'(T)$  celui de  $g_n(T)$  dont les degrés respectifs sont  $L := L_{n-1}$  et  $L' := L_n$ . Il suffit alors de remarquer que si  $L' \leq n - L$  alors les calculs suivants sont valides (indices positifs !).

$$\begin{aligned} S_{n-L'+L'} &= \sum_{i=0}^{L'-1} f'_i s_{n-L'+i} = \sum_{i=0}^{L'-1} f'_i s_{n-L'+i-L+L} \\ &= \sum_{i=0}^{L'-1} f'_i \sum_{j=0}^{L-1} f_j s_{n-L'+i-L+j} = \sum_{j=0}^{L-1} f_j \sum_{i=0}^{L'-1} f'_i s_{n-L'+i-L+j} \\ &= \sum_{j=0}^{L-1} f_j s_{n-L+j} \end{aligned}$$

Une égalité qui montre que les  $n+1$  premiers termes de la suite  $s$  sont engendrés par le plus court des registres, ce qui est contradictoire. □

Considérons la suite des approximations  $(R_n, f_n)_{0 \leq n}$  d'une série  $S(T)$ . La suite des degrés  $(L_n)_{0 \leq n}$  est croissante et nous dirons que  $m$  est un point de décrochage si  $L_m < L_{m+1}$ .

**Lemme 4.** *Soit  $(R_n, f_n)$  une approximation à l'ordre  $n$  sans être une approximation à l'ordre  $n+1$ . En notant,  $m$  le plus grand point décrochage inférieur à  $n$  alors on obtient une approximation  $(R_{n+1}, f_{n+1})$  à l'ordre  $n+1$  en posant :*

$$\begin{aligned} f_{n+1}(T) &= f_n(T) + T^{n-m} f_m(T) \\ R_{n+1}(T) &= R_n(T) + T^{n-m} R_m(T) \end{aligned}$$

```

ALGORITHME BERLEKAMP-MASSEY( s )
DONNEES   s : suite;
VARIABLE m, n : indice;
          Lm, Ln : indice;
          gm, gn : polynome;

DEBUT
  gm := 1;   Lm := 0;
  gn := 1;   Ln := 0;
  n := 1;   m := 0
  TANTQUE ( n < LONGUEUR[s] )
    d := s[n] + PREDICTION(rn, gn, Ln)
    SI ( d <> 0 ) ALORS
      gt := gn
      gn := gn + gm * T^(n - m)
      SI 2 * Ln <= n ALORS
        Ln := n + 1 - Ln
        m := n
  gm := gt
  FSI
  FSI
  FTQ
  RETOURNER gn
FIN

```

FIGURE 3. Berlekamp-Massey.

*Proof.* Le degré de  $f_{n+1}(T)$  est compatible avec le lemme précédent. Toutes les hypothèses faites permettent d'écrire :

$$\begin{aligned}
 f_n(T)S(T) &= R_n(T) + T^n \pmod{(T^{n+1})} \\
 f_m(T)S(T) &= R_m(T) + T^m \pmod{(T^{m+1})}
 \end{aligned}$$

Il suffit alors d'ajouter la seconde ligne multipliée par  $T^{n-m}$  à la première.  $\square$

**Exercice 7.** Adaptez l'algorithme de Berlekamp-Massey sur un corps arbitraire.

## 9. CYCLOTOMIE

Soient  $p$  un premier,  $n$  un entier premier avec  $p$ ,  $K$  un corps fini de caractéristique  $p$  et de cardinal  $q$ . Dans nos séances de travaux pratiques, nous supposons que  $K = \mathbf{F}_2$ . Désignons par  $L$  le plus petit corps de décomposition de  $X^n - 1$  sur  $K$  i.e. le plus petit corps contenant  $K$  et les racines  $n$ -ème de l'unité. On sait que le degré  $f$  de l'extension  $L/K$  est égal à l'ordre multiplicatif de  $q$  modulo  $n$ .

TABLE 1. Le corps à 16 éléments.

log	$\alpha^i$	vec	num	log	$\alpha^i$	vec	num
0	1	0001	1	8	$\alpha^2 + 1$	0101	5
1	$\alpha$	0010	2	9	$\alpha^3 + \alpha$	1010	10
2	$\alpha^2$	0100	4	10	$\alpha^2 + \alpha + 1$	0111	7
3	$\alpha^3$	1000	8	11	$\alpha^3 + \alpha^2 + \alpha$	1110	14
4	$\alpha + 1$	0011	3	12	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
5	$\alpha^2 + \alpha^1$	0110	6	13	$\alpha^3 + \alpha^2 + 1$	1101	13
6	$\alpha^3 + \alpha^2$	1100	12	14	$\alpha^3 + 1$	1001	9
7	$\alpha^3 + \alpha + 1$	1011	11	15	1	0001	1

Pour concrétiser l'objet abstrait  $L$ , on choisit un polynôme irréductible  $\pi(X)$  à coefficients dans  $K$ , de degré  $f$ , et on sait que  $L$  est isomorphe à l'anneau quotient  $K[X]/(\pi(X))$ . Un théorème général de la théorie des corps affirme que tout sous-groupe fini d'un corps (commutatif) est cyclique. En particulier, le groupe multiplicatif de  $L$  est cyclique d'ordre  $q^f - 1$ . Un générateur de ce groupe s'appelle une racine primitive. Bien évidemment, le polynôme minimal d'une racine primitive est de degré  $f$ , on dit que c'est un polynôme primitifs. On suppose que c'est le cas de  $\pi(X)$  et on note  $\alpha$  la classe de  $X$  modulo  $\pi(X)$ , c'est une racine primitive de  $L$ . L'exponentiation de base  $\alpha$  qui envoie l'entier  $i$  sur  $\alpha^i$  définit une bijection de  $[0, q^f - 2]$  sur l'ensemble des éléments non nul de  $L$ . Chaque élément de  $z$  est identifié à un entier qu'on appelle le logarithme de base  $\alpha$  de  $z$ . Les  $f$  premières puissances successives de  $\alpha$  forment une base, un élément  $z \in L$  se décompose d'une et une seule manière en une somme  $\sum_{i=0}^{f-1} a_i \alpha^i$  ce qui permet de voir  $z$  comme un  $f$ -uplet de  $K^f$ . Dans le cas où  $K = \mathbf{F}_p$ , la décomposition en base  $p$  permet alors d'associer à  $z$  un numéro de  $[0, q^f - 1]$ .

$$(6) \quad \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_r \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_r^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \dots & \alpha_r^{r-1} \end{pmatrix}$$

La matrice ci-dessus est une matrice de type Vandermonde. Qui n'a jamais entendu parler de ce type de matrices ? Des tableaux qui portent assez mal leur nom puisque Vandermonde semble ne jamais avoir rien publié sur cette question ! Quoiqu'il en soit la matrice est inversible si et seulement si les éléments  $\alpha_1, \alpha_2, \dots, \alpha_r$  est sont tous distincts. Plus précisément, son déterminant vaut

$$\prod_{1 \leq i < j \leq r} (\alpha_j - \alpha_i).$$

**Exercice 8.** *Prouver ce qui vient d'être dit.*

Notons  $\beta$  une racine  $n$ -ième primitive, il y en a  $\phi(n)$  et par construction, le corps  $L$  les contient toutes. Soit  $\ell \in [0, n[$ . Toutes les sous matrices carrées

$r \times r$  de la matrice :

$$(7) \quad \begin{bmatrix} 1 & \beta^\ell & \beta^{2\ell} & \dots & \beta^{(n-1)\ell} \\ 1 & \beta^{\ell+1} & \beta^{2(\ell+1)} & \dots & \beta^{(n-1)(\ell+1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \beta^{\ell+r-2} & \beta^{2(\ell+r-2)} & \dots & \beta^{(n-1)(\ell+r-2)} \end{bmatrix}$$

sont de rang  $r$  sur  $L$ .

**Proposition 2.** *L'ensemble des polynômes à coefficients dans  $K$ , de tailles  $n$ , ayant pour racines  $\beta^\ell, \beta^{\ell+1}, \beta^{\ell+2}, \dots, \beta^{\ell+r-1}$  forme un code cyclique de distance minimale strictement supérieure à  $r$ , on dit que c'est un code BCH, abréviation de Bose-Chaudhuri-Hocquenghem.*

*Proof.* Tout a été dit ! □

En particulier, nous noterons  $\text{BCH}_\ell(t, K, n)$  (notation non standard) le code BCH  $t$ -correcteur obtenu en faisant  $r = 2t$  et lorsque  $\ell = 1$ , on parle de code BCH au sens strict.

## 10. CLASSE BCH

Il n'est pas inintéressant de prouver la proposition précédente par des moyens détournés. La transformée de Reed-Solomon du polynôme  $A(Z)$  est le polynôme  $\hat{A}(Z)$  dont le  $k$ -ième coefficient vaut  $A(\beta^k)$ .

**Proposition 3.** *La transformée de Reed-Solomon est isomorphisme de l'anneau  $L[Z]/(Z^n - 1)$  dans l'anneau des polynômes de degré au plus  $n - 1$  munis du produit termes à termes. L'isomorphisme réciproque envoie  $A(Z)$  sur  $\hat{A}(Z) = \frac{1}{n} \sum_{k=0}^{n-1} A(\beta^{-k}) Z^k$ .*

*Proof.* Exercice. □

Montrons d'une seconde façon le résultat de la proposition (2). Il suffit de montrer que le nombre de coefficients nuls dans un mot  $A$  du code  $\text{BCH}_\ell(t, K, n)$  est majoré par  $n - 2t - 1$ . Nous savons que  $A$  possède  $r$  zéros consécutifs. On a

$$A_k \beta^{k(-\ell-r)} = \frac{1}{n} \hat{A}(\beta^{-k}) \beta^{k(-\ell-r)}$$

Le nombre de composantes nulles dans  $A$  est inférieur ou égal au degré du polynôme  $\hat{A}(Z) Z^{-\ell-r}$ , soit  $n - r - 1$ .

La dimension du code  $\text{BCH}_\ell(t, K, n)$  est égal à au degré du plus petit multiple commun des polynômes minimaux des éléments  $\beta^\ell, \beta^{\ell+1}, \beta^{\ell+2}, \dots, \beta^{\ell+r-1}$ . Il n'existe pas de «formule» pour calculer la dimension de  $\text{BCH}(t, K, n)$ , mais cette dernière se déduit d'un bref examen cyclotomique. Si  $J$  désigne un système de représentant cyclotomique de l'intervalle  $[\ell, \ell + r - 1]$  alors le générateur du code est

$$\prod_{j \in J} M_{\beta^j}(X)$$

où  $M_{\beta^j}(X)$  désigne le polynôme minimal de  $\beta^j$  sur  $K$ , son degré  $f_j$  est égal au cardinal de la classe cyclotomique de  $j$  modulo  $n$  et la dimension du code

<pre> galois produit(galois x, y) { galois r = 0, ret;   ret = (galois) 1 &lt;&lt; DIMEN;   while ( !y ) {     if ( y &amp; 1 ) res = res^x;     x = x &lt;&lt; 1;     if(x &amp; ret) x = x ^ PRIM;     y = y &gt;&gt; 1;   }   return(r); } </pre>	<pre> galois inversion(galois x) { galois r = 1;   int i;   i = 1;   while ( i &lt; DIMEN ){     x = produit(x, x);     r = produit(r ,x)   }   return(r); } </pre>
--	---

FIGURE 4. opérations dans un corps de Galois.

vaut :

$$k = n - \sum_{j \in J} f_j.$$

Tout se simplifie dans le cas des codes de Reed-Solomon, c'est-à-dire lorsque  $L = K$ . Le générateur est égal au produit  $\prod_{j=\ell}^{\ell+r-1} (X - \beta^j)$  et définit un code MDS.

## 11. IMPLANTATION EN LANGAGE C

Les extensions du corps  $\mathbf{F}_2$  sont faciles et agréables à implanter en langage C tant que la dimension du corps reste raisonnable c'est-à-dire inférieure à 64 bits. Il est alors possible de définir un type «galois» par entier long non signé de 64 bits (long long) pour stocker un élément du corps de Galois. On peut utiliser deux variables globales : le degré de l'extension DIMEN, et le polynôme primitif PRIM également un entier de 64 bits qui est initialisé à partir de la table des polynômes primitifs :

```

galois prm[32]={
  1,3,7,11,19,37,67,131,391,529,1033,2053,4359,8231,20487,
  32771, 65581,131081,262183,524327,1048585,2097157,4194307,
  8388641,16777243,33554441,67108935,134217767,268435465,
  536870917,1073741907};

```

Pour des tables plus complète, vous pouvez télécharger le fichier  
<http://www.univ-tln.fr/~langevin/CDE/CORPS/primitif.dat>

La somme de deux éléments se fait par une disjonction exclusive, la multiplication et l'inversion sont décrites ci-dessous.

**Exercice 9.** *Utiliser la version Blankenship, voir [6], de l'algorithme d'Euclide étendu pour écrire une inversion plus performante.*

**Exercice 10.** *Justifier la correction des fonctions proposées. Écrire une fonction pour calculer l'ordre d'un élément, la puissance etc...*

Un polynôme de taille  $n$  est représenté par un pointeur sur un élément de Galois. Le lecteur n'aura pas de mal à implanter la construction des codes BCH à partir du calcul de polynôme minimal illustré par la figure 5.

```

void Xlin(galois *g, galois z, int n)
{ int i;
  for( i = n; i>0; i--)
    g[i] = g[i-1] ^ prd(g[i], z);
  g[0] = prd(g[0], z);
}

galois *Minimal(galois z, int n)
{ galois y, *r;
  int i;
  r = (galois *)
      calloc(n, sizeof(galois));
  r[0] = 1;
  y = z;
  do { Xlin( r , y, n);
      y = prd(y,y);
    }
  while ( y != z);
}
return(r);
}

```

FIGURE 5. Calcul du polynôme minimal

## 12. BCH 2-CORRECTEUR

Commençons par un cas vraiment très simple, celui du code BCH 1-correcteur sur le corps à deux éléments. Il s'agit du code constitué des polynômes  $f(X) \in \mathbf{F}_2[X]/(X^n - 1)$  vérifiant :

$$f(\beta) = 0.$$

Le cas des codes 2-correcteurs rest plus intéressant, pour l'essentiel, il faut savoir résoudre une équation du seconde degré dans un corps de caractéristique 2...

Désignons par  $g$  le mot reçu après émission de  $f$  avec une erreur à la position  $j$ . Nous avons  $g(X) = f(X) + X^j$  et le calcul du syndrome  $f(\beta)$  donne la position de l'erreur.

## 13. CODES PRIMITIFS

Un code BCH de longueur  $q^f - 1$  est dit primitif. Le groupe affine  $GA(L, 1)$  est inclus dans le groupe d'automorphisme de chacun des codes BCH primitifs étendus. Il peut être beaucoup plus important. On démontre par ailleurs que le code de Reed-Muller d'ordre  $k$  est inclus dans le BCH étendu de distance  $2^{m-k} - 1$ . L'objectif de cette section est de prouver que la classe des codes primitifs est mauvaise. La relation de Lucas (1878) est utile pour la suite. Si  $\sum_{i=0}^{\infty} k_i p^i$  et  $\sum_{i=0}^{\infty} l_i p^i$  désignent les décompositions en base  $p$  (premier) des entiers  $k$  et  $l$  alors

$$\text{(Lucas, 1878)} \quad C_l^k \equiv \prod_{i=0}^{\infty} C_{l_i}^{k_i} \pmod{p}$$

avec une conséquence

**Lemme 5.** *Soit  $r$  un entier et soit  $S$  un  $K$ -espace vectoriel. Si le poids binaire de l'entier  $r$  est strictement plus petit que la dimension de  $S$  alors la somme  $\sum_{x \in S} x^i$  est nulle.*

*Proof.* On raisonne par induction sur la dimension de  $S$ . La propriété est triviale si  $S$  est de dimension nulle, supposons la acquise pour la dimension  $r - 1$ . Donnons un espace de dimension  $r$ . Sans perdre en généralité, on peut le supposer contenant  $1 \in S$ , c'est-à-dire que  $S = T \oplus K$  avec  $T$  de dimension  $r - 1$ . Si  $i$  est de poids inférieur à  $r - 1$  c'est fini donc, considérons un entier  $i$  de poids  $r - 1$ .

$$\begin{aligned} \sum_{x \in S} x^i &= \sum_{y \in T} \sum_{z \in K} (x + z)^i = \sum_{y \in T} \sum_{z \in K} \sum_{j=0}^i C_i^j x^j z^{i-j} \\ &= \sum_{y \in T} \sum_{z \in K} \sum_{\text{wt}(j)=r-1} C_i^j x^j z^{i-j} = \sum_{y \in T} x^j \sum_{z \in K} z^0 \end{aligned}$$

La congruence de Lucas est utilisée pour pour affirmer :  $j < i$  et  $\text{wt}(j) = \text{wt}(i)$  implique  $C_i^j \equiv 0$ .  $\square$

**Proposition 4.** *Soit  $k$  un entier,  $k < f$ . Le code BCH primitif de longueur  $q^f - 1$  et de distance désignée  $q^h - 1$  est effectivement de distance minimale  $q^h - 1$ .*

*Proof.* En fait, dans le BCH de distance désignée  $q^h - 1$ , les mots de poids  $q^h - 1$  font légion. En effet, partons d'un espace  $S$  de dimension  $h$  et considérons le polynôme

$$A(Z) = \sum_{\beta^j \in S} Z^j$$

c'est un polynôme de poids  $q^h - 1$  et qui est bien dans le BCH primitif de distance désignée  $q^h - 2$  car si  $i \leq q^h - 2$  alors  $\text{wt}(i) < h$  et donc  $A(\beta^i) = \sum_{s \in S} s^i = 0$ .  $\square$

**Exercice 11.** *La distance minimale  $d$  d'un code BCH primitif de distance prescrite  $\delta$  est au plus égale à  $q\delta - 1$ .*

**Théorème 5.** *L'ensemble des BCH primitifs ne forme pas une classe de bons codes.*

*Proof.*  $\square$

## 14. RÉSIDUS QUADRATIQUES

Les codes BCH longs sont «mauvais» ce qui ne signifie pas que la classe BCH soit mauvaise. Presque tous les codes cycliques sont des BCH et peut raisonnablement conjecturer que la classe des codes

cycliques soit bonne à l'image des codes «résidus quadratiques» décrit ci-dessous.

Soit  $\ell$  un nombre premier.

### 15. ALGORITHME D'EUCLIDE

**Proposition 5** (Bâchet-Bézout). *Soient  $A(X)$  et  $B(X)$  deux polynômes à coefficients dans un corps  $K$ . Notons  $D(X)$  leur plus grand diviseur commun. Il existe deux polynômes  $U(X)$  et  $V(X)$  tels que*

$$AU + BV = D, \quad \deg U \leq \deg B, \quad \deg V \leq \deg A.$$

La démonstration qui nous occupera jusqu'à la fin de cette section donne lieu à un algorithme de décodage des codes BCH. Pour homogénéiser nos expressions récurrente, on pose  $R_{-1}(X) := A(X)$  et  $R_0(X) := B(X)$ . L'algorithme des divisions successives dû à Euclide donne une suite de reste  $R_1, R_2, \dots, R_n, 0$ .

$$\begin{aligned} R_{-1}(X) &= Q_1(X)R_0(X) + R_1(X) \\ R_0(X) &= Q_2(X)R_1(X) + R_2(X) \\ &\vdots \\ R_{i-2}(X) &= Q_i(X)R_{i-1}(X) + R_i(X) \\ &\vdots \\ R_{n-1}(X) &= Q_{n+1}(X)R_n(X) + 0 \end{aligned}$$

On associe à la suite des restes deux suites de polynômes  $U_i(X)$  et  $V_i(X)$  satisfaisant à

$$U_i(X)A(X) + V_i(X)B(X) = R_i$$

Il suffit d'initialiser correctement ces suites tout en vérifiant la même relation de récurrence que les  $R_i$  :

$$\begin{aligned} U_i(X) &= U_{i-2}(X) - Q_i(X)U_{i-1}(X) \\ V_i(X) &= V_{i-2}(X) - Q_i(X)V_{i-1}(X) \end{aligned}$$

sous les conditions initiales

$$\begin{aligned} U_{-1}(X) &= 1 & V_{-1}(X) &= 0, \\ U_0(X) &= 0, & V_0(X) &= 1, \end{aligned}$$

Nous obtenons

$$\begin{bmatrix} R_i \\ R_{i-1} \end{bmatrix} = \begin{bmatrix} -Q_i & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -Q_1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} R_0 \\ R_{-1} \end{bmatrix}$$

et par inversion

$$(8) \quad \begin{bmatrix} 0 & 1 \\ 1 & Q_1 \end{bmatrix} \cdots \begin{bmatrix} 0 & 1 \\ 1 & Q_i \end{bmatrix} \begin{bmatrix} R_i \\ R_{i-1} \end{bmatrix} = \begin{bmatrix} R_0 \\ R_{-1} \end{bmatrix}$$

de même

$$(9) \quad \begin{bmatrix} U_i & V_i \\ U_{i-1} & U_{i-1} \end{bmatrix} = \begin{bmatrix} -Q_i & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -Q_1 & 1 \\ 1 & 0 \end{bmatrix}$$

Cette dernière égalité montre que le degré de  $U_i$  vaut  $\sum_{j=1}^i \deg Q_j$  alors que (8) montre que  $\sum_{j=1}^i \deg Q_j + \deg R_{i-1} = \deg R_{-1}$ . Nous obtenons deux inégalités fondamentales

$$(10) \quad \begin{aligned} \deg U_i + \deg R_{i-1} &\leq \deg R_{-1} \\ \deg V_i + \deg R_{i-1} &\leq \deg R_0 \end{aligned}$$

ce qui fait plus que démontrer la proposition.

## 16. DÉCODAGE DES BCH

On se donne le code  $\text{BCH}_\ell(t, K, n)$  annulant les racines  $\beta^\ell, \beta^{\ell+1}, \dots, \beta^{\ell+r-1}$  où on a posé  $r = 2t$ . La distance minimale est au moins  $2t + 1$ , et le code corrige au moins  $t$ -erreur effectivement décodable. La méthode décrite ci-dessous s'appuie sur l'algorithme des restes Euclidiens successifs, mais je suggère au lecteur de s'intéresser à l'algorithme de Berlekamp qu'on retrouve dans la cryptanalyse des chiffrements au fil de l'eau.

Donnons nous un mot reçu,  $R(X)$ , c'est-à-dire la somme d'un mot du code et d'un certain mot erreur :

$$E(X) = \sum_{j \in J} E_j X^j.$$

Il s'agit de déterminer  $E$  connaissant les  $r$  syndromes  $S_i := R(\beta^{\ell+i}) = E(\beta^{\ell+i})$ , pour  $i \in \{0, 1, \dots, r-1\}$  sachant que le poids de  $E$  est inférieur ou égal à  $t$ .

On introduit trois polynômes supplémentaires. Le syndrome  $S(X) := \sum_{i=0}^{r-1} X^i S_i$ , le localisateur (d'erreurs)

$$\Lambda(X) = \prod_{j \in J} (1 - \beta^j X)$$

qui vaut zéro en  $\beta^k$  si et seulement si une erreur apparaît à la position  $n - k$ , et déterminateur d'erreurs

$$\Delta(X) = \sum_{j \in J} E_j \beta^{\ell j} \prod_{k \neq j} (1 - \beta^k X)$$

La modulation par  $\beta^{\ell}$  deviendra claire un petit peu plus loin. Dans l'anneau des séries formelles,

$$\begin{aligned} \frac{\Delta(X)}{\Lambda(X)} &= \sum_{j \in J} \frac{E_j \beta^{\ell j}}{1 - \beta^j X} \\ &= \sum_{j \in J} E_j \beta^{\ell j} \sum_{k=0}^{\infty} \beta^{kj} X^k \\ &= \sum_{k=0}^{\infty} \sum_{j \in J} E_j \beta^{(\ell+k)j} X^k \\ &\equiv \sum_{k=0}^{r-1} S_k X^k \pmod{X^r} \end{aligned}$$

d'où la congruence polynomiale

$$(11) \quad \Delta(X) \equiv \Lambda(X)S(X) \pmod{X^r}$$

Pour un degré de  $\Lambda$  minimal, la congruence précédente contrainte par  $\deg \Lambda \leq r/2 = t$  et  $\deg \Delta \leq r/2 - 1$  possède une et une solution à un facteur scalaire multiplicatif près.

**Exercice 12.** *Prouvez le!*

Pour déterminer une solution minimale, on applique l'algorithme Eulidien avec  $A := X^r$  et  $B := S(X)$ . On calcule les restes successifs  $R_1, R_2$ , etc... On note  $R_i$  le premier reste de degré strictement inférieur à  $t$ . Avec les notations de la section précédente,

$$U_i(X)X^r + V_i(X)S(X) = R_i(X)$$

Mais les inégalités (10) montrent que  $\deg V_i \leq \deg S - \deg R_{i-1} < t$ .

Pour terminer le décodage, il faut trouver les racines de  $\Delta(X)$ . Si  $\beta^j$  est une racine de  $\Lambda(X)$  alors il y a une erreur à la position  $n - j$ , le motif de l'erreur est obtenu par la relation :

$$E_j \beta^{(l-1)j} = \frac{\Delta}{\Lambda'}(\beta^j).$$

## REFERENCES

- [1] F. J. MACWILLIAMS AND N. J. A. SLOANE. *The theory of error correcting codes*, volume 16. North Holland, 1977.
- [2] J. H. VAN LINT. *Introduction to coding theory*, volume 86. Springer Verlag, 1982.
- [3] LANGEVIN PH. *La formule de Poisson dans la théorie des codes, séquences et fonctions booléennes*. <http://langevin.univ-tln.fr/>, 1999.
- [4] LIDL R., NIEDERREITER H. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, 1983.
- [5] O. PAPINI ET J. WOLFMANN. *Algèbre Discrète et Code Correcteurs*, volume 20. Springer-Verlag, 1995.
- [6] SEROUL R. *math-info, informatique pour mathématiciens*. InterEditions, 1995.
- [7] STICHTENOTH H. *Algebraic function fields and codes*. Universitext. Springer-Verlag, 1993.

```
ALGORITHME EUCLIDIEN(S, t)
DONNEES      S : POLYNOME;
VARIABLE     Q, R : POLYNOME;
             A, B, V: VECTEUR;
DEBUT
  A := [ 1, 0, X^2t ];
  B := [ 0, 1, S ];
  TANTQUE ( DEGRE(B[2]) > t )
    R := A[2] MOD B[2];
    Q := A[2] DIV B[2];
    V := A - Q * B;
    A := B;
    B := V;
  FINTQUE
  LOCALISATEUR : B[2]
  EVALUATEUR   : B[1]
FIN
```

FIGURE 6. Décodage BCH.