4.8 Application des corps finis aux codes correcteurs d'erreur

Cette section se veut une introduction succincte et très partielle à la vaste théorie des codes correcteurs, l'une des applications pratiques les plus célèbres de la théorie des corps finis (qui n'a jamais entendu parler de corps finis sans être informés qu'ils sont indispensables à la technologie des disques compacts ?). Les exigences du programme officiel en la matière ne sont pas claires : le contenu exact est « codes cycliques », sans aucune autre précision. On se concentre ici sur les aspects directement liés aux outils mathématiques qu'on a vus dans les sections précédentes, sans vraiment insister sur l'aspect pratique de la mise en oeuvre effective des codes correcteurs et l'efficacité des algorithmes de décodage ; il faudrait déjà pour cela introduire la notion de complexité d'un algorithme ; ceux qui connaissent cette notion pourront essayer d'estimer la complexité des algorithmes présentés et/ou se reporter aux références ci-dessous. La démonstration de certains résultats fait l'objet d'exercices de TD.

Si vous souhaitez approfondir le sujet, l'ouvrage destiné aux étudiants de licence le plus complet en ce qui concerne la théorie des codes est sans doute le *Cours d'algèbre*, de Michel Demazure, aux éditions Cassini. Il contient entre autre un chapitre décrivant précisément le codage réellement employé sur les disques compacts.

Nous signalons aussi comme références utiles :

- le complément 1 du chapitre 4 de *Mathématiques L2*, ouvrage collectif aux éditions Pearson ;
- le chapitre 21 de *Toute l'algèbre de la licence*, de Jean-Pierre Escofier, aux éditions Dunod.

Ces deux références couvrent peu ou prou le même matériel que ce qui suit, avec sans doute plus d'exemples.

4.8.1 Introduction en agitant les mains

L'idée de départ qui sous-tend la théorie des codes correcteurs d'erreurs est la suivante.

Soit, dans une situation donnée, $\mathcal{A}$ un ensemble (fini) d'informations susceptible d'être transmises par un certain canal. Le canal de transmission n'étant pas parfait, ces informations sont également susceptibles d'être altérées au cours de la transmission.

Pour pallier ce problème, on construit une bijection (le codage) de $\mathcal{A}$ sur une partie $\mathcal{C}$ d'un plus gros ensemble d'informations $\mathcal{B}$. Cet ensemble $\mathcal{B}$ est munie d'une certaine distance qui mesure à quel point deux éléments de $\mathcal{B}$ sont semblables.

Le codage doit vérifier, vis-à-vis de cette distance, les propriétés suivantes :

- si la transmission d'un élément de $\mathcal{C}$ n'est pas trop perturbée, le résultat sera un élément de $\mathcal{B}$ à distance pas trop grande de c ;
- les éléments de $\mathcal{C}$ sont assez éloignés les uns des autres.

Ainsi, connaissant le résultat $b \in B$ de la transmission pas trop perturbée d'un élément c de $\mathcal{C}$, on peut retrouver l'élément c initialement transmis : c'est l'élément $c \in C$ qui se trouve à distance minimale de b.

Les alphabets radio (« Alpha, Bravo, Charlie… ») constituent une illustration basique mais assez parlante de ce type de construction.

Ici l'ensemble $\mathcal{A}$ est l'ensemble des lettres de l'alphabet, destinées à être transmises oralement par ondes radios (typiquement un indicatif de navigation aérienne du genre « ATC ») et l'ensemble $\mathcal{B}$ est (disons) l'ensemble des mots de la langue utilisée pour la communication radio. La distance entre deux éléments de $\mathcal{B}$ mesure à quel point les prononciations de ces deux éléments sont phonétiquement différentes. Le codage associe à chacune des lettres de l'alphabet un mot qui commence par la lettre en question (Alpha pour A, Bravo pour B, Charlie pour C, etc). Ce codage est choisi de sorte que l'ensemble des mots associés aux lettres de l'alphabet soient mutuellement assez éloignés phonétiquement les uns des autres. Ainsi même si la communication est un peu parasitée, il sera facile de retrouver que l'émetteur a voulu transmettre par exemple Bravo (donc la lettre B) et pas un autre mot du code. En particulier il y a peu de risque de confondre après une communication pas trop parasitée les mots Bravo et Charlie, alors que phonétiquement la prononciation des lettres B et C peut facilement être confondue dès qu'il y a un peu de « friture » sur la ligne.

4.8.2 Un cadre théorique (un peu trop) général

Les messages (ou « mots ») susceptibles d'être transmis sont écrits dans un certain *alphabet*, en d'autres termes un certain ensemble fini A. L'ensemble des messages susceptibles d'être transmis est alors A^k, où k est un certain entier. Si le message à transmettre à une longueur strictement supérieure à k, on le découpe en « blocs » de taille fixe égale à k.

Un *codage* est une application injective $c\colon A^k \to A^n$, où n est un entier supérieur à k (on code en « ajoutant de la redondance »). Le *code* proprement dit est alors l'image $\mathscr{C} := c(A^k) \subset A^n$, et n est appelé la *longueur* du code : c'est la longueur des mots qui vont être réellement transmis, afin de permettre la correction de certaines erreurs éventuelles de transmission.

On définit la *distance* entre deux éléments $(a_i)_{1\leqslant i\leqslant n}$ et $(b_i)_{1\leqslant i\leqslant n}$ de A^n comme étant

$$d((a_i),(b_i)) = \mathrm{card}(\{i \in \{1,\dots,n\}, a_i \neq b_i\}).$$

On vérifie que $d\colon A^n \times A^n \to \mathbf{N}$ est bien une distance sur A^n, appelée *distance de Hamming*.

Le processus de codage/transmission/correction/décodage peut alors se décrire ainsi ;

1. Soit $\boldsymbol{m} \in A^k$ un mot à transmettre ; **(A)** on code $\boldsymbol{m}$ en $\boldsymbol{n} := c(\boldsymbol{m}) \in A^n$
2. On transmet $\boldsymbol{n}$; à la réception on obtient un élément $\boldsymbol{n'}$ de A^n éventuellement distinct de $\boldsymbol{n}$ à cause des erreurs de transmission
3. **(B)** On calcule $\boldsymbol{n''} \in \mathscr{C}$ un élément qui minimise la distance de $\boldsymbol{n'}$ à $\mathscr{C}$ et **(C)** on calcule $\boldsymbol{m''} \in A^k$ tel que $c(\boldsymbol{m''}) = \boldsymbol{n''}$.

Le *nombre d'erreurs de transmission* dans le processus décrit ci-dessus est $d(\boldsymbol{n}, \boldsymbol{n'})$. S'il n'y a pas d'erreur, on aura $\boldsymbol{n''} = \boldsymbol{n'}$ et $\boldsymbol{m''} = \boldsymbol{m}$.

La *distance minimale* du code $\mathscr{C}$ est la distance minimale entre deux éléments distincts [10] de $\mathscr{C}$. On la note δ.

Soit τ le plus grand entier strictement inférieur à $\delta/2$. On constate alors : *le processus décrit ci-dessus corrige de manière certaine τ erreurs de transmission (ou moins).* On dit que le code est τ-correcteur. Plus précisément, dès que $d(\boldsymbol{n}', \boldsymbol{n}) \leqslant \tau$, on a nécessairement $\boldsymbol{n}'' = \boldsymbol{n}$ d'après la définition de δ et l'inégalité triangulaire. En revanche, on peut trouver au moins un couple $(\boldsymbol{n}, \boldsymbol{n}') \in \mathscr{C} \times A^n$ tel que $d(\boldsymbol{n}, \boldsymbol{n}') \geqslant \tau + 1$ et $\boldsymbol{n}$ ne minimise pas la distance de $\boldsymbol{n}'$ à $\mathscr{C}$ ou, dans le meilleur des cas, n'est pas l'unique élément de $\mathscr{C}$ vérifiant cette propriété.

Il est intuitivement clair que k étant fixé, on peut, en augmentant la longueur n, rendre δ (et donc τ) aussi grand que l'on veut pour un bon choix du code. Bien entendu augmenter n est coûteux dans la pratique, puisque cela revient à augmenter la taille de l'information effectivement transmise. Il est moins évident a priori de construire des codes efficaces *et* peu coûteux, c'est-à-dire qui permettent de corriger de manière certaine un nombre raisonnable d'erreurs tout en gardant une longueur raisonnable. Noter que dans la pratique il est important aussi de disposer d'algorithmes efficaces pour effectuer les opérations **(A)**, **(B)** et **(C)** de la procédure ci-dessus. L'opération **(A)** s'appelle *codage*, l'opération **(B)** *correction* et **(B)**+**(C)** *correction + décodage* ; parfois **(B)** est appelée décodage.

Il s'avère que pour construire des codes ayant de bonnes propriétés au sens ci-dessus, il est plus ou moins nécessaire de mettre plus de structure sur les objets considérés.

4.8.3 Codes linéaires

On s'intéresse désormais exclusivement à la famille particulière des codes *linéaires*.

Plus précisément, en conservant les notations précédentes, on suppose désormais que $A = \mathbf{K}$ est un corps fini et que $\varphi\colon \mathbf{K}^k \to \mathbf{K}^n$ est $\mathbf{K}$-linéaire. En particulier le code $\mathcal{C}$ est alors un sous-espace vectoriel de dimension k de $\mathbf{K}^n$. Dans la pratique, il est fréquent (mais pas systématique) de prendre pour $\mathbf{K}$ le corps à deux éléments, ce qui est très adapté à la représentation par bits utilisée par les ordinateurs.

On définit le *poids* $\omega(\boldsymbol{n})$ d'un élément $\boldsymbol{n}$ de $\mathbf{K}^n$ comme étant son nombre de composantes non nulles, c'est-à-dire $d(\boldsymbol{n}, (0, \ldots, 0))$. La distance minimale δ de $\mathcal{C}$ coïncide alors avec le poids minimal des élément non nuls de $\mathcal{C}$. Ceci vient du fait que $\mathcal{C}$ est un sous-groupe de $\mathbf{K}^n$ et que la distance de Hamming est invariante par translation.

Les paramètres essentiels d'un code linéaire sont sa longueur n, sa dimension k et sa distance minimale δ. On dit que le code est de paramètres $[n, k, \delta]$. Parfois on précise aussi dans les paramètres le cardinal q du corps fini $\mathbf{K}$.

Rappelons que l'on cherche à obtenir des codes de longueur n assez petite tout en étant de distance minimale δ assez grande. De ce point de vue, on a la contrainte suivante, appelée *borne de Singleton* [11].

10. On écarte systématiquement dans la suite le cas où le code $\mathscr{C}$ est réduit à un élément, de peu d'intérêt dans la pratique.

11. du nom de R.C. Singleton, auteur d'un article de 1964 où ce résultat est démontré

Proposition 14. *Avec les hypothèse précédentes (code linéaire de dimension k, de longueur n et de distance minimale δ) on a*

$$\delta \leqslant n - k + 1.$$

Cette borne quantifie le fait intuitif qu'on ne peut pas avoir à la foit n petit (peu de redondance) et δ grand (beaucoup d'erreurs corrigées de manière certaine). Les codes linéaires dont les paramètres vérifient l'égalité $\delta = n - k + 1$ sont dits de type MDS (pour l'anglais *maximal distance separable*). Nous en verrons quelques exemples.

Dans la pratique, l'application linéaire φ est donnée par une matrice $G \in \mathcal{M}_{k,n}(\mathbf{K})$ à k lignes et n colonnes à coefficients dans $\mathbf{K}$ et de rang k, appelée *matrice génératrice du code*. Les lignes de G forment donc une base de $\mathscr{C}$. *Bien noter qu'ici et dans la suite, conformément à la tradition en théorie des codes, les éléments de $\mathbf{K}^n$ sont identifiés à des vecteurs lignes*. Cette matrice génératrice permet l'opération de codage **(A)** de la manière très simple suivante : à un élément x de $\mathbf{K}^k$, on associe $x \cdot G \in \mathbf{K}^n$.

L'opération **(C)** de décodage après correction (étant donné $y \in \mathcal{C}$, trouver $x \in \mathbf{K}^n$ telle que $x \cdot G = y$) sera également très simple si la matrice génératrice G est sous forme dite *systématique*. Ceci signifie que la matrice G s'écrit par blocs $G = (I_k|\tilde{G})$ où $\tilde{G} \in \mathcal{M}_{k,n-k}(\mathbf{K})$. Noter qu'on peut toujours se ramener à ce cas quitte à bien choisir la base de $\mathcal{C}$ et *à effectuer une permutation adéquate des coordonnées*. Dans ce cas le décodage après correction est simplissime : étant donné $y \in \mathcal{C}$, l'unique $x \in \mathbf{K}^k$ tel que $y = x \cdot G$ est obtenu en ne gardant que les k premières coordonnées de y.

Identifiant le $\mathbf{K}$-espace vectoriel $\mathbf{K}^n$ à son espace dual, une *matrice de contrôle* pour le code $\mathscr{C}$ est une matrice $H \in \mathcal{M}_{n-k,k}(\mathbf{K})$ dont les lignes forment une base de l'orthogonal $\mathscr{C}^\perp$ de $\mathscr{C}$, où pour mémoire

$$\mathscr{C}^\perp = \{y \in \mathbf{K}^n, \quad \forall x \in \mathscr{C}, \quad \langle y\,,\, x\rangle = 0\}.$$

Dans la pratique, déterminer une matrice de contrôle revient donc à déterminer un système de $n - k$ équations linéaires homogènes en n variables décrivant $\mathscr{C}$. On a donc

$$\mathscr{C} = \{x \in \mathbf{K}^n, \quad x \cdot {}^tH = 0\}.$$

Si la matrice génératrice G est sous forme systématique $G = (I_k|\tilde{G})$, on constate aisément que $H = (-{}^t\tilde{G}|I_{n-k})$ est une matrice de contrôle. La connaissance de la matrice de contrôle permet, au moins théoriquement, de déterminer la distance minimale du code.

Proposition 15. *Avec les hypothèse et notations précédentes, soit H une matrice de contrôle de $\mathscr{C}$. Soit d l'unique entier strictement positif vérifiant la propriété suivante :*

1. *il existe d colonnes de H qui forment un système lié ;*
2. *tout sous-ensemble de colonnes de H de cardinal $d-1$ est un système libre.*

Alors d est la distance minimale de $\mathscr{C}$.

Le *syndrome* d'un élément $y \in \mathbf{K}^n$ est $s(y) := y \cdot {}^tH$. En particulier le syndrome définit une application linéaire surjective $s\colon \mathbf{K}^n \to \mathbf{K}^{n-k}$, de noyau $\mathscr{C}$.

Soit $x \in \mathscr{C}$ un mot à transmettre, et $y \in \mathbf{K}^n$ le mot finalement transmis. On appelle $e = y - x$ le *vecteur d'erreurs* ; on a en particulier $s(y) = s(e)$. Déterminer $x' \in \mathscr{C}$ tel que $d(y, x') = d(y, \mathscr{C})$ revient alors à déterminer un élément y' de poids minimal de l'ensemble

$$\{z \in \mathbf{K}^n, \quad s(z) = s(y)\}.$$

En effet, posant alors $x' = y - y'$, on aura $s(x') = s(y) - s(y') = 0$ donc $x' \in \mathscr{C}$, et par ailleurs si x'' est un élément quelconque de $\mathscr{C}$ on a $s(y - x'') = s(y)$ donc, par définition de y',

$$\omega(y - x'') \geqslant \omega(y') = \omega(y - x').$$

On a donc bien $d(y, x'') \geqslant d(y, x')$.

Cette remarque justifie la méthode suivante de décodage de codes linéaires, dite *par syndrome* : on détermine au préalable, pour chaque valeur possible σ du syndrome, l'élément $y(\sigma)$ de $\mathbf{K}^n$ de poids minimal parmi les éléments de $\mathbf{K}^n$ de syndrome σ et on stocke ces valeurs dans une table.

Ensuite, pour chaque mot $y \in \mathbf{K}^n$ reçu après transmission, on calcule $\sigma = s(y)$ et on décode y en $y - y(\sigma)$.

Noter qu'il y a $\mathrm{card}(\mathbf{K})^{n-k}$ valeurs possibles du syndrome. Dans la pratique la méthode précédente n'est réellement efficace que lorsque $n - k$ est petit. D'autres méthodes plus efficaces [12] ont été mises au point pour des familles particulières de codes linéraires. Nous décrivons ci-dessous une telle méthode pour les codes dits BCH, qui font partie de la famille plus générale des codes cycliques qui font l'objet de la partie suivante.

4.8.4 Codes cycliques

Rappelons les hypothèses et notation : $\mathbf{K}$ est un corps fini, n et k sont des entiers strictement positifs, $\varphi\colon \mathbf{K}^k \to \mathbf{K}^n$ est une application $\mathbf{K}$-linéaire injective et $\mathscr{C} = \varphi(\mathbf{K}^k)$ est le code linéaire associé.

Définition 16. Le code $\mathscr{C}$ est dit *cyclique* s'il est invariant par l'application de décalage

$$T\colon \begin{array}{ccc} \mathbf{K}^n & \longrightarrow & \mathbf{K}^n \\ (a_0, a_1, \ldots, a_{n-2}, a_{n-1}) & \longmapsto & (a_{n-1}, a_0, \ldots, a_{n-3}, a_{n-2}) \end{array}.$$

On obtient une reformulation importante de cette propriété en identifiant $\mathbf{K}^n$ à l'espace vectoriel sous-jacent à la $\mathbf{K}$-algèbre quotient $\mathbf{K}[X]/\langle X^n - 1\rangle$. Notant abusivement X l'image de X dans cette algèbre quotient, on sait (proposition 8 de la section 3) que $\{1, X, \ldots, X^{n-1}\}$ est une base du $\mathbf{K}$-espace vectoriel sous-jacent, au moyen de laquelle on identifie cet espace

12. Pour préciser ce que l'on entend par « efficace », il faudrait introduire la notion de complexité d'un algorithme, ce que nous ne ferons pas dans le cadre de ce cours.

vectoriel sous-jacent à $\mathbf{K}^n$. On vérifie alors que via cette identification, l'application de décalage T ci-dessus correspond exactement à la multiplication par X. Ainsi les codes cycliques sont exactement les sous $\mathbf{K}$-espaces vectoriels de $\mathbf{K}[X]/\langle X^n-1\rangle$ invariants par multiplication par X, ou, ce qui revient au même, par multiplication par n'importe quel élément de $\mathbf{K}[X]/\langle X^n-1\rangle$. En d'autres termes

Proposition 17. *Modulo l'identification* $\mathbf{K}^n \cong \mathbf{K}[X]/\langle X^n-1\rangle$ *décrite ci-dessous, les codes cycliques de* $\mathbf{K}^n$ *sont exactement les idéaux de* $\mathbf{K}[X]/\langle X^n-1\rangle$.

Par la proposition 19 de la section 2, les idéaux de $\mathbf{K}[X]/\langle X^n-1\rangle$ sont en bijection naturelle avec les idéaux de $\mathbf{K}[X]$ contenant l'idéal $\langle X^n-1\rangle$. Au vu de la proposition 29 de la section 2, ce dernier ensemble d'idéaux est en bijection naturelle avec l'ensemble des diviseurs unitaires de X^n-1. Si g est un tel diviseur unitaire, le code cyclique correspondant sera dit engendré par g.

Nous avons décrit les codes cycliques mais pas les applications de codage correspondantes. Si $\mathscr{C}$ est un code cyclique correspondant à l'idéal engendré par un diviseur unitaire g de X^n-1, sa dimension est $k := n - \deg(g)$. Un codage systématique est alors donné par l'application qui à $(a_1,\ldots,a_k) \in \mathbf{K}^k$ associe

$$\sum_{i=1}^{k} a_i X^{n-i} - r_{\boldsymbol{a}}(X) \in \mathbf{K}[X]/\langle X^n-1\rangle$$

où $r_{\boldsymbol{a}}(X)$ est (l'image dans $\mathbf{K}[X]/\langle X^n-1\rangle$ du) le reste de la division euclidienne de $\sum_{i=1}^{k} a_i X^{n-i}$ par g.

Noter que dans le cas d'un code cyclique engendré par un polynôme g, la borne de Singleton peut se réécrire

$$\deg(g) \geqslant \delta - 1.$$

Nous allons à présent donner une minoration de la distance minimale d'un code cyclique de $\mathbf{K}[X]/\langle X^n-1\rangle$, sous l'hypothèse supplémentaire que n et $q := \mathrm{card}(\mathbf{K})$ sont premiers entre eux.

Cette hypothèse garantit que $[q]_n$ est un élément du groupe $(\mathbf{Z}/n\mathbf{Z})^\times$. Soit $r \geqslant 1$ son ordre. Ainsi r est le plus petit entier strictement positif vérifiant $q^r = 1 \pmod n$. Soit maintenant $\mathbf{L}$ un corps à q^r éléments contenant $\mathbf{K}$ (*cf.* le théorème 13). Comme n divise $q^r - 1 = \mathrm{card}(\mathbf{L}^\times)$ et que $\mathbf{L}^\times$ est un groupe cyclique, il existe dans $\mathbf{L}^\times$ un élément α d'ordre n. Noter que comme $\alpha^n = 1$, l'expression α^i a un sens pour $i \in \mathbf{Z}/n\mathbf{Z}$. Comme α est d'ordre n, l'ensemble $\{\alpha^i\}_{i\in\mathbf{Z}/n\mathbf{Z}} \subset \mathbf{L}$ est de cardinal n. C'est donc l'ensemble des racines du polynôme X^n-1 dans $\mathbf{L}$, et ce polynôme est entièrement décomposé dans $\mathbf{L}$:

$$X^n - 1 = \prod_{i\in\mathbf{Z}/n\mathbf{Z}} (X - \alpha^i).$$

En particulier tout diviseur unitaire $g \in \mathbf{L}[X]$ de $X^n - 1$ s'écrit

$$g = \prod_{i \in \Sigma} (X - \alpha^i)$$

où Σ est une partie de $\mathbf{Z}/n\mathbf{Z}$. En fait l'application

$$\Sigma \mapsto g_\Sigma := \prod_{i \in \Sigma} (X - \alpha^i)$$

est une bijection de l'ensemble des parties de $\mathbf{Z}/n\mathbf{Z}$ sur l'ensemble des diviseurs unitaires *dans* $\mathbf{L}[X]$ de $X^n - 1$; attention, ce dernier ensemble contient l'ensemble des diviseurs unitaires dans $\mathbf{K}[X]$ de $X^n - 1$, c'est-à-dire l'ensemble des codes cycliques de $\mathbf{K}[X]/\langle X^n - 1 \rangle$, mais il est strictement plus gros dès que $r \geqslant 2$ (car alors $\alpha \notin \mathbf{K}$, donc $X - \alpha \notin \mathbf{K}[X]$. On démontrera en TD :

Proposition 18. *Avec les hypothèses et notations ci-dessus, $g_\Sigma \in \mathbf{K}[X]$ si et seulement si Σ est stable par multiplication par q.*

L'application réciproque de l'application ci-dessus sera notée $g \mapsto \Sigma_g$.

Ceci étant posé, on a l'estimation suivante pour la distance minimale d'un code cylique de $\mathbf{K}[X]$.

Proposition 19. *On conserve les hypothèse et notations précédentes. Soit $g \in \mathbf{K}[X]$ un diviseur unitaire de $X^n - 1$. On suppose qu'il existe $\nu \in \mathbf{Z}/n\mathbf{Z}$ et $2 \leqslant d \leqslant n$ un entier positif tels que*

$$\{\nu + [i]_n\}_{0 \leqslant i \leqslant d-2} \subset \Sigma_g.$$

Alors le code cyclique engendré par g est de distance minimale au moins d.

Cette propriété est à la base de la définition des codes cycliques dits BCH [13]. L'intérêt de ces codes est qu'ils bénéficient d'algorithmes de décodage efficaces. Noter que nous n'avons rien dit pour l'instant du problème du décodage pour les codes cycliques ; bien entendu, comme pour tout code linéaire, la méthode de décodage par syndrome est disponible, mais nous avons déjà signalé qu'elle était en général inefficace dans la pratique.

4.8.5 Codes BCH

On rappelle le contexte et les notations : $\mathbf{K}$ est un corps fini de cardinal q, n est un entier positif premier à la caractéristique de $\mathbf{K}$, r est l'ordre [14] de q dans $(\mathbf{Z}/n\mathbf{Z})^\times$, $\mathbf{L}$ est un corps de cardinal q^r contenant $\mathbf{K}$ et $\alpha \in \mathbf{L}^\times$ est un élément d'ordre n. Soit $2 \leqslant d \leqslant n$ un entier. Soit $\mathcal{I} \in \mathbf{K}[X]$ l'idéal des polynômes $P \in \mathbf{K}[X]$ vérifiant

$$\forall i \in \{1, \ldots, d-1\}, \quad P(\alpha^i) = 0.$$

13. du nom de leurs inventeurs : R. Bose, D. K. Ray-Chaudhuri et A. Hocquenghem

14. En particulier, n divise $q^r - 1$; si $n = q^r - 1$, on parle de code BCH *primitif*.

C'est un idéal qui contient $\langle X^n - 1\rangle$ et qui définit donc un code cyclique de $\mathbf{K}[X]/\langle X^n-1\rangle$ de distance minimale δ *au moins* égale à d d'après la proposition 19 : on appelle un tel code un *code BCH de longueur n et de distance assignée d*. Soit t le plus grand entier strictement inférieur à $\frac{d}{2}$. L'algorithme de correction décrit ci-dessous permettra de corriger de manière certaines t erreurs. Noter que la distance minimale réelle du code est en général strictement supérieur à d (mais difficile à calculer dans la pratique) ; le code pourrait donc, en théorie, corriger de manière certaines plus d'erreurs, mais on ne dispose pas a priori d'algorithme efficace pour ce faire. Cependant nous verrons en TD le cas particulier où $\mathbf{L} = \mathbf{K}$, c'est-à-dire $n = q-1$ et α est un générateur de $\mathbf{K}^\times$. On obtient alors par la construction ci-dessus un code de distance minimale exactement d, qui est en outre de type MDS. De tels codes sont appelés codes de Reed-Solomon [15].

La dimension du code obtenu est $n - \deg(g)$ où $g \in \mathbf{K}[X]$ est le générateur unitaire de l'idéal $\mathcal{I}$ défini ci-dessus. D'après la proposition 18, on a $g = g_\Sigma$ où Σ est la plus petite partie de $\mathbf{Z}/n\mathbf{Z}$ contenant $\{[i]_n\}_{1\leqslant i\leqslant d-1}$ et stable par multiplication par q. Comme $q^r = 1 \pmod n$, l'ensemble

$$\{[q^s i]_n\}_{\substack{1\leqslant i\leqslant d-1\\ 0\leqslant s\leqslant r-1}}$$

contient $\{[i]_n\}_{1\leqslant i\leqslant d-1}$ et est stable par multiplication par q. On en déduit la majoration

$$\deg(g) \leqslant r(d-1).$$

Décrivons à présent l'algorithme de correction. Rappelons qu'on désigne par t le plus grand entier strictement inférieur à $\frac{d}{2}$. Soit $\boldsymbol{m}$ un mot de $\mathscr{C}$ et $\boldsymbol{m}'$ le mot transmis. On suppose comme annoncé que $f := d(\boldsymbol{m}, \boldsymbol{m}') \leqslant t$. Soit $\boldsymbol{e} = \boldsymbol{m} - \boldsymbol{m}'$.

Pour $j = 1, \ldots, 2t$, soit

$$s_j := \boldsymbol{m}'(\alpha^j) = \boldsymbol{e}(\alpha^j) \in \mathbf{L}$$

et

$$\boldsymbol{s}(Z) = \sum_{j=1}^{2t} s_j Z^{j-1} \in \mathbf{L}[Z].$$

Notons que $\deg(\boldsymbol{s}) \leqslant 2t-1$.

Écrivons

$$\boldsymbol{e}(X) = \sum_{i=1}^{f} a_i X^{r_i}$$

avec $1 \leqslant r_1 < r_2 < \cdots < r_f < n$ et les a_i sont tous non nuls. Il s'agit de déterminer les r_i et les a_i. Pour cela, il suffit de connaître le polynôme

$$\boldsymbol{\sigma}(Z) = \prod_{i=1}^{f} (1 - \alpha^{r_i} Z) \in \mathbf{L}[Z].$$

15. du nom de leurs inventeurs I.S. Reed et G. Solomon

En effet, connaissant $\boldsymbol{\sigma}(Z)$, on détermine quelles puissances de α en sont des racines (en les testant toutes), ce qui donne f et les r_i. La détermination des a_i se fait alors en résolvant le système linéaire

$$s_j = \sum_{i=1}^{f} a_i \alpha^{j r_i}, \quad 1 \leqslant j \leqslant f$$

dont la matrice est un Vandermonde.

Il reste donc à explique comment calculer $\boldsymbol{\sigma}(Z)$ à partir des données déjà connues. Posons

$$\boldsymbol{\omega}(Z) = \sum_{i=1}^{f} a_i \alpha^{r_i} \prod_{1 \leqslant k \neq i \leqslant f} (1 - \alpha^{r_k} Z) \in \mathbf{L}[Z].$$

On vérifie aussitôt que $\deg(\boldsymbol{\omega}) < t$ et qu'aucune des racines de $\boldsymbol{\sigma}(Z)$ n'est racine de $\boldsymbol{\omega}(Z)$. En particulier $\boldsymbol{\sigma}(Z)$ et $\boldsymbol{\omega}(Z)$ sont premiers entre eux. En outre on vérifie facilement la relation

$$\boldsymbol{s}(Z)\boldsymbol{\sigma}(Z) = \boldsymbol{\omega}(Z) \pmod{Z^{2t}}.$$

Lemme 20. *Soit $\boldsymbol{\sigma}_0, \boldsymbol{\omega}_0 \in \mathbf{L}[Z]$ tels que $\deg(\boldsymbol{\sigma}_0) \leqslant t$, $\deg(\boldsymbol{\omega}_0) < t$ et*

$$\boldsymbol{s}(Z)\boldsymbol{\sigma}_0(Z) = \boldsymbol{\omega}_0(Z) \pmod{Z^{2t}}.$$

Alors il existe $\mathbf{C}(Z) \in \mathbf{L}[Z]$ tel que $\boldsymbol{\sigma}_0(Z) = \mathbf{C}(Z)\boldsymbol{\sigma}(Z)$ et $\boldsymbol{\omega}_0(Z) = \mathbf{C}(Z)\boldsymbol{\omega}(Z)$.

Démonstration. On a

$$\boldsymbol{\sigma}_0(Z)\boldsymbol{\omega}(Z) = \boldsymbol{s}(Z)\boldsymbol{\sigma}(Z)\boldsymbol{\sigma}_0(Z) = \boldsymbol{\omega}_0(Z)\boldsymbol{\sigma}(Z) \pmod{Z^{2t}}$$

Ainsi

$$\boldsymbol{\sigma}_0(Z)\boldsymbol{\omega}(Z) - \boldsymbol{\omega}_0(Z)\boldsymbol{\sigma}(Z) = 0 \pmod{Z^{2t}}$$

Mais $\deg(\boldsymbol{\sigma}_0(Z)\boldsymbol{\omega}(Z) - \boldsymbol{\omega}_0(Z)\boldsymbol{\sigma}(Z)) < 2t$ donc

$$\boldsymbol{\sigma}_0(Z)\boldsymbol{\omega}(Z) - \boldsymbol{\omega}_0(Z)\boldsymbol{\sigma}(Z) = 0$$

Or $\boldsymbol{\omega}(Z)$ et $\boldsymbol{\sigma}(Z)$ sont premiers entre eux. Par le lemme de Gauss, $\boldsymbol{\omega}(Z)$ divise $\boldsymbol{\omega}_0(Z)$ et on conclut facilement. □

Si on sait construire $\boldsymbol{\omega}_0$ et $\boldsymbol{\sigma}_0$ ayant la propriété de l'énoncé de lemme, on en déduit facilement $\boldsymbol{\sigma}$ et $\boldsymbol{\omega}$: comme $\boldsymbol{\omega}$ et $\boldsymbol{\sigma}$ sont premiers entre eux, le polynôme $\mathbf{C}$ de la conclusion de l'énoncé est le pgcd de $\boldsymbol{\omega}_0$ et $\boldsymbol{\sigma}_0$.

La construction de $\boldsymbol{\omega}_0$ et $\boldsymbol{\sigma}_0$ est basée sur l'algorithme d'Euclide étendu pour les polynômes, sur lequel nous reviendrons un peu plus tard dans la partie consacrée aux anneaux euclidiens.