Université de Rouen
Agrégation de mathématiques 2005-2006

Eléments de théorie des corps finis. Application : les codes correcteurs.

Nicolas Bruyère

Table des matières

Première partie

Les corps finis

Dans cette partie on ne présente que des résultats sur les corps finis qui nous seront utiles pour la suite concernant les codes correcteurs. C'est bien sûr loin d'être exhaustif, il y manque même certains résultats essentiels pour l'agrégation (voir [Per96] pour plus de détails).

1 Corps finis : définitions et premières propriétés

1.1 Introduction

Les corps finis les plus simples sont ceux de la forme $\mathbb{Z}/p\mathbb{Z}$ où p est premier. Ce sont d'ailleurs les uniques corps finis de cardinaux premiers (uniques à isomorphisme *unique* près). On va voir dans la suite que ces corps particuliers interviennent dans la structure de tous les corps finis.
Rappelons aussi que si $K \subset L$, où K et L sont des corps finis alors on peut munir L d'une structure de K-espace vectoriel. On a alors une relation entre les cardinaux de ces corps et la dimension de L sur K donnée par $|L| = |K|^{\dim_K L}$

1.2 Caractéristique et cardinal

Définition 1.1 *Soit K un corps (pas forcément fini). On appelle sous-corps premier de K le plus petit sous-corps de K contenant* 1.

On va chercher à décrire ces sous corps premiers. Pour cela on pose l'homomorphisme d'anneaux $\varphi : \mathbb{Z} \to K$ défini par $\varphi(n) = n.1 = 1 + \cdots + 1$. $\ker \varphi$ est alors un idéal de $\mathbb{Z}$, donc de la forme $p\mathbb{Z}$. D'autre part, les théorèmes d'isomorphisme donnent $\mathbb{Z}/p\mathbb{Z} \sim \varphi(\mathbb{Z})$ qui est inclus dans K donc intègre. Donc deux cas : $p = 0$ ou p est un nombre premier.

Définition 1.2 *Le nombre p est appelé la caractéristique du corps K. Il est noté* $\mathrm{car}(p)$.

Remarques
- Si $\mathrm{car}(K) > 0$ alors $px = 0$ pour tout $x \in K$.
- Si $\mathrm{car}(K) = 0$ alors $\varphi(\mathbb{Z}) \sim \mathbb{Z}$, donc K est infini. Alors $\mathbb{Q}$ est le sous-corps premier de K.
- Si K est fini on a $p = \mathrm{car}(K) > 0$ alors le sous-corps premier de K est $\mathbb{Z}/p\mathbb{Z}$ que l'on note aussi $\mathbb{F}_p$. Donc $|K| = p^m$. Le cardinal d'un corps fini est une puissance d'un nombre premier (par exemple il n'y a pas de corps à 6 éléments).
- Si $\mathrm{car}(K) > 0$ alors K n'est pas nécessairement fini. Par exemple, la clôture algébrique de $\mathbb{F}_2$.

Proposition 1.1 (Frobenius) *Soit K un corps de caractéristique $p > 0$. L'application $x \in K \mapsto x^p \in K$ est un homomorphisme de corps appelé homomorphisme de Frobenius.*

Démonstration — (exercice). □

Proposition 1.2 *Soit K un corps de caractéristique $p > 0$.*
- *Soit $x \in K$. Alors $x \in \mathbb{F}_p$ si et seulement si $x^p = x$.*
- *Soit $Q \in K[X]$. Alors $Q \in \mathbb{F}_p[X]$ si et seulement si $Q(X^p) = (Q(X))^p$.*

Démonstration — Soit $x \in K$. Si $x \in \mathbb{F}_p$ alors par le théorème de Lagrange $x^p = x$. Réciproquement, le polynôme $X^p - X$ a au plus p racines, or les éléments de $\mathbb{F}_p$ au nombre de p sont tous racines.
Si $Q \in \mathbb{F}_p[X]$ alors d'après l'homomorphisme de Frobenius $Q(X^p) = (Q(X))^p$. Réciproquement si $Q(X^p) = (Q(X))^p$ alors par l'homomorphisme de Frobenius les coefficients de Q vérifient l'équation $x^p = x$ donc appartiennent à $\mathbb{F}_p$. □

Avant d'aller plus loin dans les propriétés sur les corps finis, il convient d'établir l'existence et l'unicité (dans un sens à préciser) de ces corps.

Théorème 1.1 *Soit p un nombre premier et soit $m \in \mathbb{N}^*$. On pose $q = p^m$.*
Il existe un corps K à q éléments, c'est le corps de décomposition du polynôme $X^q - X$ sur $\mathbb{F}_p$.
En particulier K est unique à isomorphisme (non unique) près. On le note $\mathbb{F}_q$.

Démonstration — Si K est un corps à q éléments et si $x \in K^*$, alors $x^{q-1} = 1$ (Lagrange), donc tout élément de K est racine de $X^q - X$. Or $X^q - X$ est à coefficients dans $\mathbb{F}_p$ le sous-corps premier de K. Donc K est bien le corps de décomposition de $X^q - X$ sur $\mathbb{F}_p$.
Réciproquement, soit K le corps de décomposition de $X^q - X$ sur $\mathbb{F}_p$ et soit $k \subset K$ l'ensemble des racines de $X^q - X$. Alors k est un corps (le vérifier). D'autre part le polynôme dérivé de $X^q - X$ est toujours non nul donc $X^q - X$ n'admet que des racines simples, donc $|k| = q$. D'où $k = K$. □

Remarques

- L'énoncé du théorème indique que K est unique à isomorphisme non unique près. Précisons ce point. Si $q = p$ est premier, K est unique à isomorphisme unique près. Car si K, K' sont deux corps de cardinal p alors il n'y a qu'un seul isomorphisme entre ces deux corps, celui qui envoie 1_K sur $1_{K'}$. En revanche si q n'est pas premier il n'y a pas unicité de cet isomorphisme. Par exemple dans le cas d'un corps K à 4 éléments, qui contient $\mathbb{F}_2$ (i.e. 0 et 1) et deux élements α et β. Alors il n'existe aucun moyen de distinguer α et β en ce sens qu'ils se comportent de la même façon face aux opérations de ce corps. Et donc si K' est un autre corps à 4 éléments dont les éléments distincts de 0 et 1 sont γ et δ alors il y a deux façons d'identifier K et K', celle qui associe α à γ et celle qui associe α à δ.
- En revanche, s'étant fixé $\mathbb{F}_p$. Si on note $\overline{\mathbb{F}_p}$ sa cloture algébrique. Alors il existe dans $\overline{\mathbb{F}_p}$ un unique corps à $q = p^m$ éléments. C'est l'ensemble des racines du polynôme $X^q - X$.

On a un analogue de la proposition 1.2, mais avec q à la place de p.

Proposition 1.3 *Soit K un corps de caractéristique $p > 0$ de cardinal $q = p^m$ et soit L une extension de K.*

- *Soit $x \in L$. Alors $x \in K$ si et seulement si $x^q = x$.*
- *Soit $Q \in L[X]$. Alors $Q \in K[X]$ si et seulement si $Q(X^q) = (Q(X))^q$.*

Démonstration — La démonstration est analogue à celle de la propositon 1.2. □

2 Les polynômes

2.1 Polynômes cyclotomiques

Dans $\mathbb{C}$ les racines n-ièmes de l'unité sont les $e^{2ik\pi/n}$ où $k \in \mathbb{Z}$. Leur ensemble $\mathbb{U}_n$ forme un groupe cyclique. Les générateurs de ce groupe sont appelés racine primitives n-ièmes de l'unité, leur ensemble est noté $\mathbb{P}_n$. Il sont de la forme $e^{2ik\pi/n}$ où $k \wedge n = 1$ ie k est inversible dans $\mathbb{Z}/n\mathbb{Z}$. Donc $|\mathbb{P}_n| = \varphi(n)$.

Définition 2.1 *Soit $n \in \mathbb{N}^*$. On appelle n-ième polynôme cyclotomique le polynôme*

$$\Phi_n(X) = \prod_{\zeta \in \mathbb{P}_n} (X - \zeta) = \prod_{\substack{k \in \mathbb{Z}/n\mathbb{Z} \\ k \wedge n = 1}} (X - e^{\frac{2ik\pi}{n}}) .$$

Cette définition est à rapprocher de l'identité

$$X^n - 1 = \prod_{k \in \mathbb{Z}/n\mathbb{Z}} (X - e^{\frac{2ik\pi}{n}}) .$$

Proposition 2.1 *Φ_n est un polynôme unitaire à coefficients entiers, de degré $\varphi(n)$. De plus il est irréductible sur $\mathbb{Q}$ donc sur $\mathbb{Z}$. On a*

$$X^n - 1 = \prod_{d|n} \Phi_d(X) \, .$$

Donnons quelques exemples.

$$\begin{aligned}
\Phi_1(X) &= X - 1 \\
\Phi_2(X) &= X + 1 \\
\Phi_3(X) &= X^2 + X + 1 \\
\Phi_4(X) &= X^2 + 1 \\
\Phi_5(X) &= X^4 + X^3 + X^2 + X + 1 \\
\Phi_p(X) &= X^{p-1} + \cdots + X + 1 \quad \text{si } p \text{ premier.}
\end{aligned}$$

On finit par le résultat suivant très important, placé ici car la démonstration utilise les polynômes cyclotomiques.

Théorème 2.1 (Wedderburn) *Tout corps fini est commutatif.*

2.2 Racines de l'unité dans un corps fini

Soit K un corps fini. Un élément ζ de K est appelé racine n-ième de l'unité si $\zeta^n = 1$. Leur ensemble $\mathbb{U}_n(K)$ est un groupe cyclique dont les générateurs sont appelés racines primitives n-ièmes de l'unité. Comme Φ_n est à coefficients entiers on peut définir $\Phi_n(x)$ pour tout x dans un corps K. Le résultat suivant donne un lien entre les polynômes cyclotomiques et les racines primitives n-ièmes de l'unité dans un corps fini.

Proposition 2.2 *Soit K un corps fini de cardinal p^m où p est premier et soit n premier avec p.*
- *Les racines n-ièmes de l'unité dans K sont des racines simples du polynôme $X^n - 1$.*
- *Les racines de Φ_n dans K sont exactement les racines primitives n-ième de l'unité dans K.*

Démonstration — Par l'absurde, supposons qu'il existe $Q \in K[X]$ non constant tel que $X^n - 1 = Q^2(X)R(X)$ où $R \in K[X]$. En dérivant, il vient

$$nX^{n-1} = Q(X)(2Q'(X)R(X) + Q(X)R'(X)) \, .$$

D'où dans $K[X]$:

$$\begin{aligned}
n.1_K &= X(nX^{n-1}) - n(X^n - 1) \\
&= Q(X)(2XQ'(X)R(X) + XQ(X)R'(X) - nQ(X)R(X)) \, .
\end{aligned}$$

Donc le polynôme Q divise la constante $n.1_K$ non nulle (pourquoi ?), donc Q est constant. D'ou la contradiction et la démonstration de la première partie. Donc, une racine n-ième de l'unité dans K est une racine d'exactement un des polynômes Φ_d où d divise n (d'après la proposition 2.1). Or, l'ensemble des racines des Φ_d pour d divisant n, d distinct de n est aussi l'ensemble des racines des $X^d - 1$ pour d divisant n, d distinct de n, i.e. l'ensemble des racines de l'unité sur K qui ne sont pas primitives. Les racines primitives sont alors les racines de Φ_n. $\square$.

3 Exemple de construction de corps finis

On va mettre à profit les résultats des deux parties précédentes pour construire des corps finis et faire des calculs sur ces corps, choses que l'on ne pouvait faire pour le moment de part la définition abstraite des corps finis. On va construire $\mathbb{F}_{16}$ grâce à une racine primitive 15ème de

l'unité α sur le corps $\mathbb{F}_2$. Pour cela on choisit une racine de Φ_{15} sur $\mathbb{F}_2$ i.e. une racine d'un des facteurs irreductibles de Φ_{15} sur $\mathbb{F}_2$. On a

$$\Phi_{15}(X) = X^8 + X^7 + X^5 + X^4 + X^2 + X + 1 = (X^4 + X^3 + 1)(X^4 + X + 1)\,.$$

Choisissons α tel que $\alpha^4 + \alpha + 1 = 0$. α est donc connu à travers la relation $\alpha^4 = \alpha + 1$. Les éléments de $\mathbb{F}_{16}$ sont 0 et les puissances α^i avec $i = 0, \ldots, 14$. Donc tout élément non nul de $\mathbb{F}_{16}$ s'exprime en fonction de α^i où $i = 0, 1, 2, 3$. Par exemple, $\alpha^6 = \alpha^3 + \alpha^2$ donc $\alpha^6 + \alpha^3 = \alpha^2$.

Deuxième partie

Les codes correcteurs

4 Position du problème

4.1 Introduction

La transmission d'informations dans l'air (informations à envoyer à un satellite) ou par des câbles (lecture d'un CD) est souvent sujette à des perturbations (chocs du discman). C'est pour cela qu'il convient d'introduire les codes détecteurs et correcteurs d'erreurs. C'est-à-dire que l'on va coder les informations à envoyer de manière judicieuse afin de pouvoir justement détecter et corriger d'éventuels problèmes sous réserve qu'ils ne soient pas trop nombreux. En fait, le technicien connaît à l'avance la qualité de la transmission et aura effectué des tests statistiques pour prévoir le nombre (mais pas la position) des erreurs.

Dans la suite de ce texte on va tout d'abord expliquer le principe du codage et du décodage sur quelques exemples. Ensuite on présentera la théorie de codes plus généraux dans les corps finis, tout d'abord les codes linéaires qui sont aisés à manipuler de part la structure vectorielle (sur des corps finis) sous-jacente et ensuite les codes linéaires cycliques qui se caractérisent d'une jolie manière grâce à des espaces vectoriels, des polynômes, des idéaux. Enfin on étudiera un cas particulier de ces derniers codes qui sont les codes BCH.

4.2 Quelques exemples simples

Nous allons introduire quelques façons simples de coder des informations et faire ainsi apparaître les quelques difficultés qui se présentent dans la conception d'un codage efficace.
Supposons que nous voulions envoyer une séquence de 4 bits (une suite de quatre 0 ou 1). Au départ il y a donc 16 séquences possibles.

1. Supposons tout d'abord que nous ne codions pas le message. Alors à l'arrivée 16 séquences possibles donc si une erreur survient pendant la transmission aucune chance de la détecter encore moins de la corriger. D'où la nécessité de coder le message à envoyer.
2. Deuxième possibilité, introduisons un bit d'erreur, c'est-à-dire que l'on rajoute un 5ème bit au message à envoyer qui est la somme (modulo 2) des 4 premiers bits. La séquence reçue est de 5 bits et s'il y a une erreur pendant la transmission elle pourra être détectée (car le 5ème bit ne sera plus la somme des 4 premiers), cependant l'erreur ne pourra être localisée donc corrigée. En revanche s'il y a deux erreurs alors elles ne pourront être ni détectées ni donc corrigées. D'où la nécesssité de connaître au préalable quelques statistiques sur la transmission.
3. Enfin troisième possibilité on envoie chaque bit du message n fois. On pourra détecter et corriger des erreurs dès l'instant que l'on sait que le nombre d'erreurs de la transmission est $< n/2$. Le problème posé par un tel codage est la dilution du message original et par conséquent l'envoi du message prend beaucoup de temps par rapport à la nature du message (la transmission en temps réel des images télé ne serait pas possible).

5 Codes linéaires

5.1 Définition des codes linéaires

Dans la suite nous allons étudier une classe particulière de codes qui sont les codes linéaires. On voudra transmettre des éléments de $\mathbb{F}_q^k$ de **longueur** k que l'on appellera **mots**. La première opération est l'opération de codage. C'est-à-dire qu'à chacun des q^k mots on associe un **mot de**

code de longueur $n \geq k$ appartenant ainsi à $\mathbb{F}_q^n$. Ceci ce fait au moyen d'une application linéaire *injective* $\mathbb{F}_q^k \to \mathbb{F}_q^n$ dont l'image C est un sous-espace vectoriel de $\mathbb{F}_q^n$ appelé **code linéaire** de **longueur** n et de dimension k. Le rapport k/n est appelé le **taux** de C, il mesure la dilution du message envoyé.

Remarques
- Ce sont les mots de code de C qui seront envoyés et décodés à l'arrivée.
- L'application considérée est injective pour que l'on puisse retrouver sans équivoque le mot que l'on a codé au départ à partir du mot décodé à l'arrivée.

Dans toute la suite on ne s'intéressera pas directement à l'application mise en jeu, mais on travaillera directement sur C un code linéaire de longueur n et de dimension k, i.e. un sous-espace vectoriel de $\mathbb{F}_q^n$ de dimension k. On fixera l'entier n et on cherchera à construire C (l'espace des mots à envoyer) de telle sorte que la détection et la correction d'erreur(s) puissent se faire de manière optimale en fonction des contraintes initiales.

$$m \in C \overset{canal}{\longrightarrow} m' \in \mathbb{F}_q^n$$

5.2 Distance de Hamming

Le décodage s'effectuera selon *le principe du maximum de vraisemblance* c'est-à-dire que le mot décodé sera le mot qui a le plus de "composantes communes" avec le mot reçu d'où l'idée d'introduire les notions suivantes.

Définition 5.1 *Soient $m, m' \in \mathbb{F}_q^n$. On appelle* **poids** *de m que l'on note $w(m)$ le nombre de composantes non nulles de m. On appelle* **distance de Hamming** *entre m et m' et on note $d(m, m')$ l'entier $w(m - m')$. Et on appelle* **distance du code** *l'entier*

$$d = \min_{\substack{m,m' \in C \\ m \neq m'}} d(m, m') = \min_{\substack{m \in C \\ m \neq 0}} w(m)$$

Ainsi, si le mot m' est reçu, le décodage consiste à chercher le mot m_0 qui minimise $d(m, m')$ pour m parcourant C. Malheureusement ce mot m_0 n'est pas forcément unique et n'est même pas forcément le mot envoyé au départ (si par exemple il y a eu un nombre d'erreurs supérieur à celui du nombre d'erreurs pouvant être corrigées). On introduit alors la définition suivante.

Définition 5.2 *Soit $t \in \mathbb{N}$, on dit qu'un code C est* **t-correcteur** *s'il peut détecter et corriger au plus t erreurs.*

La définition précédente et d sont reliés par le résultat qui suit.

Proposition 5.1 *On peut détecter une erreur de poids $< d$. Si $d = 2t$ alors C est $(t-1)$-correcteur. Si $d = 2t + 1$ alors C est t-correcteur.*

Démonstration — Si $m \in C$ est envoyé et $m' = m + e$ est reçu avec $w(e) < d$ alors $m' \notin C$ car sinon $w(m' - m) < d$ avec $m' - m \in C$, ce qui contredit la distance minimale du code.
Si $m' \in \mathbb{F}_q^n$ peut s'écrire $m_1 + e_1$ et $m_2 + e_2$ avec $m_1, m_2 \in C$ et $w(e_1), w(e_2) < d/2$ alors :

$$\begin{aligned} w(m_1 - m_2) &= w(e_1 - e_2) \\ &\leq w(e_1) + w(e_2) \quad \text{(inég. triang.)} \\ &< d\,. \end{aligned}$$

D'où $m_1 = m_2$. □

Ainsi plus d est grand plus le code peut corriger d'erreurs. Cependant le triplet d'entiers (n, k, d), le **type** du code C, est relié par l'inégalité suivante.

Proposition 5.2 (Borne de Singleton) *On a* $d \leq n+1-k$

Démonstration — Soit E le sous-espace vectoriel de $\mathbb{F}_q^n$, constitué des mots dont les $k-1$ dernières composantes sont nulles. Alors :

$$\begin{aligned} \dim(C \cap E) &= \dim(C) + \dim(E) - \dim(\mathrm{Vect}(C \cup E)) \\ &= k + (n-k+1) - \dim(\mathrm{Vect}(C \cup E)) \\ &\geq 1. \end{aligned}$$

Il existe donc un élément non nul $m \in C$ qui de plus vérifie $w(m) < n+1-k$. □

Finalement à k fixé pour avoir un code optimal il faut avoir un n proche de k ($\geq k$) pour ne pas avoir trop de dilution et un d le plus grand possible pour avoir une capacité de correction grande. Mais la borne de Singleton montre que l'on ne peut avoir les deux à la fois, il faudra donc faire un compromis qui dépendra des contraintes initiales.

5.3 Quelques exemples

Revenons aux exemple de la section 4.2. Les exemples 1,2,3 de la section 4.2 sont respectivement des codes de type $(n, n, 1)$, $(n, n-1, 2)$, $(n, 1, n)$. Introduisons un quatrième exemple plus intéressant. On considère le code C qui est l'image de l'application

$$\begin{aligned} \Phi : \mathbb{F}_2^4 &\longrightarrow \mathbb{F}_2^7 \\ (a_0, a_1, a_2, a_3) &\longmapsto a_0 m_0 + a_1 m_1 + a_2 m_2 + a_3 m_3 \end{aligned}$$

où $m_0 = (1101000)$, $m_1 = (0110100)$, $m_2 = (0011010)$, $m_3 = (0001101)$. Alors le poids des mots du code est ≥ 3 (exercice). C'est donc un code de type $(7, 4, 3)$, il est donc 1-correcteur (code de Hamming de longueur 7).

6 Codes linéaires cycliques

On a signalé plus haut implicitement un algorithme de décodage. En effet, pour m' reçu on peut retrouver m (sous réserve que le nombre d'erreurs survenues soit en phase avec la capacité de correction du code) en minimisant $d(m, m')$ où $m \in C$. C'est un principe de décodage qui s'avère très lent quand C est grand. On va alors introduire les codes linéaires cycliques qui fournissent un cadre pour de bons algorithmes de décodage.

6.1 Définitions

Définition 6.1 *Un code linéaire $C \subset \mathbb{F}_q^n$ est dit* **cyclique** *s'il est stable par le shift à droite*

$$\begin{aligned} \sigma : \mathbb{F}_q^n &\longrightarrow \mathbb{F}_q^n \\ (m_0, \ldots, m_{n-1}) &\longmapsto (m_{n-1}, m_0 \ldots, m_{n-2}) \,. \end{aligned}$$

On va identifier $\mathbb{F}_q^n$ à l'algèbre $\mathbb{F}_q[X]/(X^n-1)$ via l'application

$$(m_0, \ldots, m_{n-1}) \longmapsto m_0 + m_1 X + \cdots + m_{n-1} X^{n-1} \ .$$

L'intérêt de cette identification est que le shift à droite s'identifie à la multiplication par X dans $\mathbb{F}_q[X]/(X^n-1)$. Ainsi, C, qui est un espace-vectoriel stable par tout itéré de shifts à droite, s'identifie à un sous-ensemble de $\mathbb{F}_q[X]/(X^n-1)$ stable par multiplication par les polynômes de $\mathbb{F}_q[X]/(X^n-1)$, il s'identifie donc à un idéal de $\mathbb{F}_q[X]/(X^n-1)$. D'où le résultat :

Proposition 6.1 *Un code linéaire est cyclique si et seulement si c'est un idéal de* $\mathbb{F}_q[X]/(X^n-1)$.

On peut aller un peu plus loin encore. On sait que la surjection canonique

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

induit une bijection entre l'ensemble des idéaux de $\mathbb{F}_q[X]/(X^n - 1)$ et les idéaux de $\mathbb{F}_q[X]$ qui contiennent $(X^n - 1)$ (l'idéal engendré par $X^n - 1$), idéaux qui sont engendrés par un diviseur (unitaire pour avoir unicité) de $X^n - 1$ car $\mathbb{F}_q[X]$ est principal. On peut alors définir :

Définition 6.2 *Le polynôme unitaire ainsi associé à un code linéaire cyclique C est appelé* **polynôme générateur** *du code cyclique.*

Construire un code linéaire cyclique revient donc à chercher les polynômes de $\mathbb{F}_q[X]$ qui divisent $X^n - 1$. Etant donné un tel polynôme g, la dimension du code est $k = n - \deg(g)$ car C est engendré (comme espace vectoriel sur $\mathbb{F}_q$) par $g, Xg, \ldots, X^{n-1-\deg(g)}g$. C est donc l'ensemble

$$\{gf : f \in \mathbb{F}_q[X], \deg(f) \leq n - 1 - \deg(g)\}$$

.

Pour finir on donne une caractérisation intéressante de la distance minimale du code grâce aux coefficients du polynôme générateur.

Proposition 6.2 *Soit $g(X) = \sum_{i=0}^{k} a_i X^i$ le polynôme générateur d'un code. Et soit $\rho = \#\{i : a_i \neq 0\} = w(g)$. Alors $d = \rho$.*

Démonstration — A voir. □

6.2 Retour aux exemples

Les trois exemples de la section 4.2 ainsi que l'exemple du code de Hamming sont des codes linéaires cycliques (le vérifier !). Les polynomes générateurs sont pour chacun d'entre eux 1, $1+X$, $1 + X + \cdots + X^{n-1}$, $1 + X + X^3$. (le vérifier !).

6.3 Construction des codes linéaires cycliques

On suppose désormais que n est premier avec q.

On a vu que construire un code sur $\mathbb{F}_q^n$ c'est trouver un polynôme unitaire g de $\mathbb{F}_q[X]$ diviseur de $X^n - 1$. On va maintenant voir comment construire ce polynôme.
On note K le corps de décomposition de $X^n - 1$ dans $\mathbb{F}_q$. C'est en fait le corps engendré par les racines n-ièmes de l'unité. Soit α une racine primitive n-ième de l'unité sur $\mathbb{F}_q$. On sait d'après la proposition 2.2 que cela revient à prendre une racine de Φ_n (c'est là que l'hypothèse n premier avec q est essentielle) . On a alors :

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

Ainsi g s'écrit :

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

où Σ est une partie convenable de $\mathbb{Z}/n\mathbb{Z}$.

Attention, remarquons qu'a priori le polynôme $g \in K[X]$ alors qu'il faut qu'il soit dans $\mathbb{F}_q[X]$. On a pour cela la proposition suivante :

Proposition 6.3 *$g \in \mathbb{F}_q[X]$ si et seulement si Σ est stable par multiplication par $q \pmod{n}$.*

Démonstration — Si $g \in \mathbb{F}_q[X]$ alors d'après l'homomorphisme de Frobenius (cf proposition 1.1) appliqué n fois et le fait que les éléments de $\mathbb{F}_q$ sont stables par élévation à la puissance q-ième, on a $(g(X))^q = g(X^q)$, donc l'ensemble des racines de g est stable par passage à la puissance q-ième c'est-à-dire Σ est stable par multiplication par $q \pmod n$.
Réciproquement, si Σ est stable par multiplication par $q \pmod n$ alors :

$$\begin{aligned} g(X^q) &= \prod_{i\in\Sigma}(X^q - \alpha^i) \\ &= \prod_{i\in\Sigma}(X^q - \alpha^{qi}) \quad \text{(stabilité de } \Sigma) \\ &= \prod_{i\in\Sigma}(X - \alpha^i)^q \quad \text{(Frobenius)} \\ &= (g(X))^q \,. \end{aligned}$$

Les coefficients de g vérifient donc chacun l'équation $x^q = x$ ce qui implique, d'après la proposition 1.3, que $x \in \mathbb{F}_q$ et finalement $g \in \mathbb{F}_q[X]$. □

Finalement, pour construire un code linéaire cyclique il nous faut trouver les parties de $\mathbb{Z}/n\mathbb{Z}$ stables par multiplication par q. De plus la nature de l'ensemble Σ peut nous donner des informations sur la distance du code (donc sur la capacité de correction du code). En effet on a la proposition suivante :

Proposition 6.4 *S'il existe $k, s \in \mathbb{N}^*$ tels que Σ contienne $k+1, k+2, \ldots, k+s$ alors la distance du code est $\geq s+1$.*

Démonstration — Supposons que Σ contienne $k+1, k+2, \ldots, k+s$. Soit $m \in C$. Alors m est de degré $< n$ et $m(\alpha^i) = 0$ pour $i = k+1, \ldots, k+s$. Supposons que $w(m) \leq s$ i.e. m a au plus s coefficients non nuls. Il s'agit donc de montrer que m est identiquement nul.
Posons $m(X) = \sum_{j=1}^{s} \lambda_j X^{n_j}$ où $n_1, \ldots, n_j$ sont entiers et $0 \leq n_1 < \cdots < n_s < n$. On a alors pour tout $1 \leq i \leq s$:

$$\sum_{j=1}^{s} \lambda_j \alpha^{(k+i)n_j} = 0 \,.$$

On a ainsi un système en $\lambda_1, \ldots, \lambda_s$ dont le déterminant D est

$$\begin{vmatrix} \alpha^{(k+1)n_1} & \alpha^{(k+1)n_2} & \cdots & \alpha^{(k+1)n_s} \\ \alpha^{(k+2)n_1} & \alpha^{(k+2)n_2} & \cdots & \alpha^{(k+2)n_s} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(k+s)n_1} & \alpha^{(k+s)n_2} & \cdots & \alpha^{(k+s)n_s} \end{vmatrix} = \alpha^{(k+1)(n_1+\cdots+n_s)} \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{n_1} & \alpha^{n_2} & \cdots & \alpha^{n_s} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(s-1)n_1} & \alpha^{(s-1)n_2} & \cdots & \alpha^{(s-1)n_s} \end{vmatrix}$$

On reconnait un déterminant de Vandermonde, donc :

$$D = \alpha^{(k+1)(n_1+\cdots+n_s)} \prod_{1\leq j<i\leq s} (\alpha^{n_i} - \alpha^{n_j}) \,.$$

Donc D est non nul car α, racine primitive n-ième de l'unité, est non nul et vérifie $\alpha^{n_i} \neq \alpha^{n_j}$ pour tout $i \neq j$. Le système (homogène) admet donc l'unique solution $\lambda_1 = \cdots = \lambda_s = 0$, finalement m est nul. □

La proposition précédente montre que pour construire un code on est amené à imposer la présence de certains éléments dans Σ. Comment le construire alors pour qu'il contienne à la fois ces éléments et soit stable par la multiplication par q ? L'idée naturelle est de "mettre" dans Σ ces éléments et les multiplications successives par q (modulo n) de ces éléments. C'est ce que l'on va expliciter plus rigoureusement dans ce qui suit en définissant tout d'abord :

Proposition-Définition 6.1 *Soit $j \in \mathbb{Z}/n\mathbb{Z}$ et soit $s > 0$ le plus petit entier tel que $q^s j \equiv j \pmod{n}$. On pose $\Sigma_j = \{j, qj, \ldots, q^{s-1}j\}$. Les parties Σ_j sont appelées les classes cyclotomiques. Ce sont les classes d'équivalence pour la relation d'équivalence sur $\mathbb{Z}/n\mathbb{Z}$, $j \sim j'$ si et seulement s'il existe i tel que $q^i j \equiv j' \pmod{n}$.*

Les parties Σ stables sont donc les réunions de classes cyclotomiques. Le polynômes générateur g_j associé à Σ_j est le polynôme minimal de α^j. Donc étant donnée une partie Σ le polynôme générateur associé est le ppcm des polynomes minimaux de α^i où $i \in \Sigma$.
En pratique (Maple par exemple) une fois que l'on s'est fixé Σ, g est le produit des facteurs irréductibles de $X^n - 1$ (dans $\mathbb{F}_q[X]$) qui annulent les α^i où $i \in \Sigma$. La factorisation évoquée est faisable car il existe des algorithmes de factorisation de polynômes (algorithme de Berlekamp) sur des corps finis.

Reprenons l'exemple du code de Hamming. On a dans ce cas $q = 2$, $n = 7$. Il vient trois classes cyclotomiques $\{1, 2, 4\}$, $\{3, 5, 6\}$ et $\{0\}$. Alors $X^7 - 1 = (X - 1)g(X)g'(X)$ où

$$\begin{aligned} g(X) &= (X - \alpha)(X - \alpha^2)(X - \alpha^4) \\ g'(X) &= (X - \alpha^3)(X - \alpha^5)(X - \alpha^6)\,. \end{aligned}$$

L'un est $X^3 + X + 1$ et l'autre $X^3 + X^2 + 1$.

7 Les codes BCH binaires

On introduit maintenant un type de codages linéaires cycliques particuliers qui sont les codes BCH binaires, du nom de leurs créateurs, Bose, Chaudhuri et Hocquenghem. Ce sont des codes qui sont utilisés dans la lecture de CD et le minitel. Ces codes sont intéressants car ils possèdent un algorithme de décodage très performant.

7.1 Définition

On prendra $q = 2$ et $n = 2^m - 1$ où $m \in \mathbb{N}$. On choisira δ tel que $1 < \delta \leq 2^m - 1$, α une racine primitive n-ième de l'unité sur $\mathbb{F}_2$ c'est-à-dire une racine de Φ_n sur $\mathbb{F}_2$(d'après la propostion 2.2) et K le corps obtenu par adjonction de α à $\mathbb{F}_2$ (corps des racines n-ièmes de l'unité sur $\mathbb{F}_2$).
Grâce à la proposition 6.4, on va imposer au code construit d'être de distance minimale $\geq \delta$, pour cela on construit Σ de telle sorte qu'il contienne $1, 2, \ldots, \delta - 1$. On remarque que $\Sigma_i = \Sigma_{2i}$ (le vérifier !). On peut alors supposer que δ est impair, $\delta = 2t + 1$ ($t \in \mathbb{N}^*$). Ainsi :

$$\Sigma = \bigcup_{\substack{1 \leq i \leq 2t-1 \\ i \text{ impair}}} \Sigma_i = \bigcup_{i=1,3,\ldots,2t-1} \Sigma_i\ .$$

Le polynôme générateur de ce code est alors le ppcm des polynômes g_i pour $i = 1, 3, \ldots, 2t - 1$.

7.2 Exemple

Soit $n = 15$. La factorisation du polynôme cyclotomique Φ_{15} modulo 2 donne

$$\Phi_{15}(X) = (X^4 + X^3 + 1)(X^4 + X + 1).$$

On choisit α racine du polynôme $g_1 = 1 + X^3 + X^4$. Il y a 4 classes cyclotomiques non triviales

$$\begin{aligned} \Sigma_1 &= \{1, 2, 4, 8\} & \Sigma_3 &= \{3, 6, 9, 12\} \\ \Sigma_5 &= \{5, 10\} & \Sigma_7 &= \{7, 11, 13, 14\}\,. \end{aligned}$$

On répertorie dans le tableau suivant les codes BCH et leurs caractéristiques constructibles avec α.

δ	Σ	k	k/n	d	t	g
$2,3$	$\{1,2,4,8\}$	11	11/15	3	1	g_1
$4,5$	$\{1,2,3,4,6,8,9,12\}$	7	7/15	5	2	g_1g_3
$6,7$	$\{1,2,3,4,5,6,8,9,10,12\}$	5	1/3	7	3	$g_1g_3g_5$
$8,\ldots,15$	$\{1,\ldots,14\}$	1	1/15	15	7	$1+X+\cdots+X^{14}$

t est la capacité de correction, les autres symboles se comprennent aisément. Précisons tout de même, $g_3 = \Phi_5 = 1+X+X^2+X^3+X^4$, $g_5 = \Phi_3 = 1+X+X^2$ et $g_7 = 1+X+X^4$ (l'autre facteur de Φ_{15}) (le vérifier !). D'où finalement :

$$\begin{aligned} g_1g_3 &= 1+X+X^2+X^4+X^8 \\ g_1g_3g_5 &= 1+X^2+X^5+X^6+X^8+X^9+X^{10} \\ g_1g_3g_5g_7 &= 1+X+\cdots+X^{14} \quad \text{(heureusement !)}\,. \end{aligned}$$

Remarques
- Le dernier code de la liste est le code de répétition pure (exemple 3 de la section 4.2).
- On peut remarquer que la proposition 6.2 est vérifiée et donc la distance prescrite est la distance minimale.

7.3 Décodage des codes BCH

Les codes BCH possèdent un algorithme très efficace de décodage basé sur l'utilisation de polynômes. On garde les mêmes notations que dans la section 7.1. On a donc $\delta = 2t+1$ de telle sorte que le code est t-correcteur. On envoie un mot m du code, il vérifie donc

$$m(\alpha) = m(\alpha^2) = \ldots = m(\alpha^{2t}) = 0\ .$$

Il subit une erreur e de poids $\nu \leq t$ donc $e = X^{l_1}+\cdots+X^{l_\nu}$ avec $0 \leq l_1 < \cdots < l_\nu \leq n-1$, le mot reçu m' est donc $m' = m+e$. Le but est, connaissant m', de trouver les l_i pour reconstituer e donc m. Pour cela on pose :

Définition 7.1 *On appelle* **polynôme localisateur d'erreurs** *le polynôme de* $K[Z]$

$$\sigma(Z) = \prod_{i=1}^{\nu}(1-\beta_j Z)$$

où $\beta_j = \alpha^{l_j} \in K$.

Les racines de ce polynôme sont exactement les $\alpha^{-l_j} = \alpha^{n-l_j}$ donc la connaissance de ce polynôme implique la connaissance des l_j. Comment calculer ce polynôme ? Pour cela on introduit :

Définition 7.2 *On appelle* **polynôme syndrome** *le polynôme de* $K[Z]$

$$S(Z) = \sum_{i=1}^{2t} S_i Z^{i-1}$$

où $S_i = e(\alpha^i) = \sum_{j=1}^{\nu}\beta_j^i = m'(\alpha^i)$.

Ce polynôme est calculable à partir de m'. Grâce à ce polynôme on peut de plusieurs manières retrouver e. Nous ne nous attarderons que sur une méthode basée sur l'algorithme d'Euclide étendu. On peut aussi utiliser la résolution d'un système linéaire ou la transformée de Fourier discrète (cf. [Dem97], [PW95]). Dans ce qui suit on va déterminer σ à partir de S.

Proposition 7.1 *Il existe un polynôme $\omega(Z) \in K[Z]$ de degré $< t$ tel que $S(Z)\sigma(Z) \equiv \omega(Z) \pmod{Z^{2t}}$.*

Démonstration— On a :

$$S(Z) = \sum_{i=1}^{2t} \left(\sum_{j=1}^{\nu} \beta_j^i \right) Z^{i-1} = \sum_{j=1}^{\nu} \beta_j \left(\sum_{i=1}^{2t} (\beta_j Z)^{i-1} \right) = \sum_{j=1}^{\nu} \beta_j \frac{1 - \beta_j^{2\nu} Z^{2\nu}}{1 - \beta_j Z}$$

d'où

$$S(Z)\sigma(Z) = \sum_{j=1}^{\nu} \left(\beta_j (1 - \beta_j^{2\nu} Z^{2\nu}) \prod_{k \neq j} (1 - \beta_k Z) \right) .$$

D'où la proposition avec

$$\omega(Z) = \sum_{j=1}^{\nu} \left(\beta_j \prod_{k \neq j} (1 - \beta_k Z) \right)$$

où $\deg(\omega) \leq \nu - 1 < t$. □

Proposition 7.2 *Soient $\tilde{\omega}$ et $\tilde{\sigma}$ deux polynômes de $K[Z]$ vérifiant $\deg(\tilde{\sigma}) \leq t$, $\deg(\tilde{\omega}) < t$ et $S(Z)\tilde{\sigma}(Z) \equiv \tilde{\omega}(Z) \pmod{Z^{2t}}$. Alors il existe un polynôme $C \in K[Z]$ tel que $\tilde{\sigma}(Z) = C(Z)\sigma(Z)$ et $\tilde{\omega}(Z) = C(Z)\omega(Z)$.*

Démonstration— On a modulo Z^{2t} :

$$\omega(Z)\tilde{\sigma}(Z) \equiv S(Z)\sigma(Z)\tilde{\sigma}(Z) \equiv \tilde{\omega}(Z)\sigma(Z)$$

donc Z^{2t} divise $\omega(Z)\tilde{\sigma}(Z) - \tilde{\omega}(Z)\sigma(Z)$ qui est de degré $< 2t$. Donc ce polynôme est nul. D'autre part σ et ω sont premiers entre eux par la propositions précédente et le fait que les racines de σ sont non nulles. D'où la proposition. □

Finalement si par un moyen quelconque on arrive à construire des $\tilde{\sigma}$ et $\tilde{\omega}$ comme ci-dessus et si on arrive à déterminer C alors on connaitra σ. Ceci peut être fait grâce à l'algorithme d'Euclide étendu appliqué à $P_0 = Z^{2t}$ et $P_1 = S$ (qui donne le pgcd de deux polynômes et aussi les coefficients de la relation de Bezout qu'ils vérifient). Appliquons-le, il existe alors des suites de polynômes (P_i), (A_i) et (B_i) tels que $\deg(P_i) < \deg(P_{i-1})$ et $P_i = A_i Z^{2t} + B_i S$ donc $SB_i \equiv P_i \pmod{Z^{2t}}$. On choisit le premier i tel que $\deg(P_i) < t$. Alors, d'après une propriété de l'algorithme, on a $\deg(B_i) = \deg(P_0) - \deg(P_{i-1}) \leq 2t - t = t$. Donc les polynômes $\tilde{\sigma} = B_i$ et $\tilde{\omega} = P_i$ satisfont aux hypothèses de la proposition 7.2. Il existe donc $C(Z) \in K[Z]$ avec $B_i = C\sigma$ et $P_i = Cw$. On montre alors facilement grâce aux propriétés de l'algorithme que $C(Z)$ est constant égal à $B_i(0) \neq 0$. D'où

$$\sigma(Z) = B_i(Z)/B_i(0) .$$

Finissons en résumant les étapes du décodage.

- On calcule le polynôme syndrome S à l'aide de m'.
- Par l'algorithme d'Euclide étendu explicité ci-dessus on calcule le polynôme localisateur d'erreurs σ.
- Enfin, on calcule successivement $\sigma(\alpha^i)$ pour $i = 1, \ldots, n-1$ et comme l'on sait que les racines de σ sont les α^{n-l_j} on retrouve directement les l_j donc e.
- Le mot envoyé est alors $m = m' - e$

Références

[Dem97] Demazure (Michel). – *Cours d'algèbre*. – Paris, Cassini, 1997.

[Per96] Perrin (Daniel). – *Cours d'algèbre*. – Paris, Ellipses, 1996.

[PW95] Papini (Odile) et Wolfmann (Jacques). – *Algèbre discrète et codes correcteurs*. – Berlin, Springer-Verlag, 1995.