Memento 6/6 - Codes

1. Généralités La transmission d'un message à travers un *canal de communication* nécessite au préalable un *codage*. Formellement, on se fixe un ensemble de tous les messages que l'on veut transmettre et chacun de ces messages est transformé en un (appelé *mot de code*) sur un alphabet fini A arbitraire. C'est l'ensemble de tous ces mots de code qui constitue un *code*. Si tous les mots du code ont la même longueur l, on dit que c'est un code *en blocs de longueur* l, sinon c'est un code de longueur *variable*. Si q est le cardinal de l'alphabet A, le code est dit q-aire (binaire si $q = 2$).

Exemples :

(1) Sur un porte-avions on peut transmettre visuellement (le canal de transmission est l'air) une quinzaine de messages différents à un pilote (haut, bas, roues, volets, etc.) tous *codés* à l'aide de deux raquettes de signalisation manipulées par l'officier d'appontage (code LSO, Landing Signal Officer's).

(2) Le code *morse* est un code de longueur variable qui contient une soixantaine de séquences plus ou moins longues de "traits" et de "points" représentants des lettres, chiffres ou des ponctuations.

(3) Le code ASCII est un code en bloc qui contient 128 séquences binaires de 7 bits représente 128 symboles (lettres, chiffres, ponctuations, etc.).

Si le canal de communication est *bruité*, on cherche à construire des codes qui permettent, dans une certaine mesure, de reconstituer le message d'origine si le message reçu ne correspond à aucun mot de l'ensemble des messages possibles.

2. Codes linéaires On note $\mathbf{F}_p$ le corps des entiers $\mathbf{Z}/p\mathbf{Z}$ modulo un premier p. En munissant l'ensemble produit $\mathbf{F}_p^n$ d'une addition interne $(x_1, \ldots, x_n) + (y_1, \ldots, y_n) := (x_1 + y_1, \ldots, x_n + y_n)$ et d'une multiplication externe $\lambda(x_1, \ldots, x_n) := (\lambda x_1, \ldots, \lambda x_n)$ avec $\lambda \in \mathbf{F}_p$, on obtient un espace vectoriel de dimension n (à la manière de $\mathbf{R}^n$). Un *code* linéaire de longueur n est tout simplement un sous-espace vectoriel de F_p^n. Si la dimension du code est k, alors il contient p^k mots de code. On appelle *matrice génératrice* d'un code linéaire, toute matrice G telle que tout mot du code s'obtient comme combinaison linéaire des lignes de cette matrice. Exemple sur $\mathbf{F}_3$: les matrices génératrices

$$G = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \qquad G' = \begin{pmatrix} 1 & 0 & 1 & 10 & 1 & 2 & 1 \end{pmatrix}$$

engendre le code $C = \{0000, 0121, 0212, 1011, 2022, 1102, 2110, 2221\}$ de dimension 2 et de longueur 4. On suppose qu'une matrice génératrice d'un code de dimension k contient exactement k lignes et elle est dite *normalisée* si les k premières colonnes forment la matrice identité. La matrice G' ci-dessus est normalisée.

3. Distance de Hamming La *distance de Hamming* sur $\mathbf{F}_p^n$ est définie par $d(x, y) := \#\{i,\ x_i \neq y_i\}$ où $x = (x_1, \ldots, x_n)$ et $y = (y_1, \ldots, y_n)$. On appelle *poids de Hamming* d'un vecteur $x = (x_1, \ldots, x_n)$ le nombre de ses composantes non-nulles que l'on note $\text{poids}(x)$ et on démontre que

$$d(x, y) = \text{poids}(x - y). \tag{1}$$

La *distance minimale* d'un code C est la plus plus petite distance non-nulle entre deux mots de codes. Pour déterminer la distance minimale d'un code linéaire, il est plus simple de chercher le mot de plus petit poids d'après l'égalité (1).

4. Capacité de correction Quand un mot de code x est envoyé via un canal de transmission, certaines de ses composantes peuvent être modifiées et on reçoit $y \neq x$. On cherche alors le mot de code x le plus proche de y et on considère que le message envoyé a été x. Si des mots de code sont trop proches les uns des autres, par exemple $x = 0110$ et $x' = 1010$, et que l'on reçoit $y = 1110$, il est impossible de savoir si c'est x qui a été transmis ou x', il faut que la distance entre les mots de code soit au moins 3 pour corriger une erreur (le premier symbole ici). Plus généralement on montre que si un code est de distance minimale d, alors on peut corriger $(d-1)/2$ erreurs, c'est la *capacité de correction* du code.

5. Codes parfaits On appelle *rayon de recouvrement* d'un code C de longueur n sur $\mathbf{F}_p$, le plus petit rayon r tel que la réunion des boules de rayon r centrées en les mots de C soit égale à l'espace $\mathbf{F}_p^n$ tout entier. Si les boules sont deux-à-deux disjointes alors le code est dit *parfait*, ce qui équivaut à dire que la capacité de correction du code est égale à son rayon de recouvrement.

Le sous-espace vectoriel orthogonal à un code C de longueur n et de dimension k est aussi un espace vectoriel donc un code, il a même longueur et sa dimension est $n - k$. On trouve aisément la matrice génératrice du code orthogonal à partir d'une matrice génératrice.