Codes correcteurs d'erreurs

Autor(en): **Sigrist, François**

Objekttyp: **Article**

Zeitschrift: **Elemente der Mathematik**

Band (Jahr): **48 (1993)**

PDF erstellt am: **14.08.2024**

Persistenter Link: https://doi.org/10.5169/seals-44634

El. Math. *48* (1993)

0013-6018/93/040157-13$1.50 + 0.20/0

Codes correcteurs d'erreurs

François Sigrist

François Sigrist, né en 1940, a étudié les mathématiques a l'EPFZ, où il a obtenu le doctorat en 1967 sous la direction de Beno Eckmann. Il a enseigné à l'University of British Columbia à Vancouver avant d'être nommé à l'Université de Neuchâtel en 1968. Ses domaines de recherche sont la topologie algébrique (H-espaces, K-théorie, homotopie) et depuis quelques années les empilements de sphères et les formes quadratiques réelles.

Viele denken beim Wort *Code* wohl vor allem an Verschlüsselungen für Geheimbotschaften oder vielleicht an Passwörter, welche die Zugangsberechtigung zum Bankkonto steuern. Die Eingeweihten wissen hingegen um die sehr viel breitere Bedeutung des Wortes "Code". Sie wissen, dass Codes heute in vielen Situationen des täglichen Lebens eine unauffällige, aber wichtige Rolle spielen. Zum Beispiel an den modernen Ladenkassen, welche in der Lage sind, den auf den Waren angebrachten Strichcode fehlerfrei zu lesen; im Buchladen, wo die ISBN-Nummer auf einem Code basiert, welcher Übermittlungsfehler erkennen lässt; beim Hören einer CD, wo Störungen durch kleine Kratzer vom Gerät selbständig ausgeglichen werden; am Computer, wo auf der Festplatte die Dateien so gespeichert sind, dass Aufzeichnungs- oder Lesefehler automatisch korrigiert werden, etc. Die meisten dieser Codes benötigen für ihren Einsatz nichttriviale Mathematik, die noch bis vor wenigen Jahren als zu abstrakt eingestuft wurde, um in anderen Gebieten von Nutzen zu sein! — In seinem Beitrag gibt François Sigrist eine Einführung in die Theorie dieser fehlerkorrigierenden Codes. Von einfachsten Beispielen ausgehend gelangt er zu zyklischen Codes, behandelt den Golay-Code (so genannt nach seinem Entdecker, dem Schweizer Elektroingenieur Marcel Golay) und kommt schliesslich kurz auf die Codes zu sprechen, die in den amerikanischen Raumsonden Voyager benutzt wurden, um die spektakulären Jupiter- und Saturnbilder in allerbester Qualität zur Erde zu übermitteln. — Der Beitrag geht auf einen "Cours de perfectionnement" zurück, den François Sigrist im Herbst 1992 in Grimentz VS für eine Gruppe von Lehrern der welschen Schweiz durchgeführt hat. *ust*

1 Un exemple illustratif

Les habitués du Sport-Toto connaissent généralement par coeur la fameuse liste de 9 colonnes de 4 matches garantissant 3 points:

$$\begin{pmatrix} x & x & x & 1 & 1 & 1 & 2 & 2 & 2 \\ x & 1 & 2 & x & 1 & 2 & x & 1 & 2 \\ x & 1 & 2 & 1 & 2 & x & 2 & x & 1 \\ x & 1 & 2 & 2 & x & 1 & 1 & 2 & x \end{pmatrix}$$

Fig. 1 Sport-Toto à 4 matches.

Comment s'assurer de cette étonnante propriété, de préférence sans balayer les 81 résultats possibles pour les 4 matches? Voici une première idée: la sphère d'influence d'une des colonnes de la liste contient 9 colonnes (elle-même, et les 8 autres colonnes réalisant 3 points). Si l'on veut couvrir les 81 possibilités, il est donc nécessaire que toutes les sphères d'influence soient *disjointes!* A cet effet, il faut contrôler que deux colonnes du tableau n'ont jamais *deux résultats communs*, ou encore qu'elles diffèrent en au moins *trois* lettres. La vérification est plus rapide, mais nous verrons plus loin qu'elle peut même s'effectuer instantanément en utilisant habilement la structure du tableau.

Introduisons un peu de vocabulaire: Les 9 colonnes sont les *mots* d'un *code* de *longueur* 4, sur un *alphabet* de 3 lettres x, 1, et 2. On appelle *distance* de deux mots le nombre de différences. Celle-ci donne à un code une structure d'espace métrique. La *distance d'un code* est la distance minimale de deux mots distincts appartenant au code. La notation (n, M, d) pour un code précise que sa longueur est n, sa distance d, et qu'il contient M mots. Notre exemple est donc un code $(4, 9, 3)$. Il nous enseigne qu'un code à distance 3 est *correcteur d'erreur*: si l'on modifie une lettre dans un mot, il est possible de la corriger automatiquement en cherchant le mot du code le plus proche.

Passons maintenant du football aux télécommunications. Imaginons que nous avons 9 mots de deux lettres dans notre vocabulaire. Prenons, par exemple, les deux premières lettres des 9 mots de notre code[1]):

$$(xx, x1, x2, 1x, 11, 12, 2x, 21, 22)$$

Supposons maintenant que ces 9 messages puissent être envoyés sur une ligne de transmission imparfaite, mais à capacité double: pour transmettre un mot de deux lettres, nous avons quatre lettres à disposition. Cependant, la ligne fait une faute dans une lettre avec la probabilité p (disons 0.01).

Première méthode: Répéter le mot. A l'arrivée, si les deux moitiés du mot reçu ne coïncident pas, le destinataire peut employer toutes les martingales possibles et imaginables pour reconstituer le message: elles sont *toutes* équivalentes au choix de la première moitié! La probabilité de décodage correct est par conséquent $(1-p)^2 = 0,9801$. Un mot sur 50 sera donc mal lu.

1) En remplaçant x par zéro, on constate que ce sont les entiers de 0 à 8, dans leur écriture en base 3: les colonnes du tableau sont donc numérotées par leurs deux premières lettres.

Deuxième méthode: Envoyer pour chaque mot la colonne correspondante du Sport-Toto. A la réception, le destinataire regarde le tableau, y cherche son meilleur résultat, et en prend les deux premières lettres. Si la ligne a fait 0 ou 1 erreur, il retrouvera le mot juste! En effet, ceci ne fait qu'exprimer le fait que le tableau garantit 3 points. La probabilité de décodage correct devient $(1-p)^4 + 4p(1-p)^3 = 0,9994\ldots$ Seul un mot sur 1689 sera mal lu. Comparativement, le gain en performance est spectaculaire.

Retournons maintenant à notre tableau de départ, en remplaçant x par zéro. L'alphabet devient l'ensemble $\{0,1,2\}$ des restes de division par 3. Nos mots vivent donc dans un espace vectoriel de dimension 4 sur le corps $\mathbf{F}_3$. De plus, ils y forment un *sous-espace vectoriel*, comme on le vérifie aisément sur le tableau. Nous avons ici, par définition, un *code linéaire*. Une des premières retombées de cette propriété est la démonstration instantanée du fait que le code a bien la distance 3. Introduisons la notion de *poids* d'un mot: c'est le nombre de lettres non-nulles; c'est aussi la distance du mot à l'origine. La distance de deux mots est alors le poids de leur différence. Par conséquent, la distance d'un code linéaire est égale au *poids minimal*. Pour notre tableau, on constate immédiatement que tous les colonnes, sauf l'origine, ont le poids 3, montrant donc sans le moindre calcul que la distance est 3.

En présence d'un code linéaire, on peut évidemment faire appel aux techniques de l'algèbre linéaire. Voici un échantillon typique, qui sera analysé en détail plus loin: supposons que le résultat des matches du Sport-Toto soit (en colonne) $(1,1,x,1)$. Comment retrouver, par calcul, notre meilleur résultat? Autrement dit, le récepteur du message reçoit le mot $(1,1,x,1)$. Quel est le message original de deux lettres?

Tout d'abord, la linéarité montre que les colonnes ont la forme $(a,b,a+b,2a+b)$. Par conséquent, le résultat des matches (x,y,z,t) fournit 4 points si $z = x+y$ et $t = 2x+y$. Matriciellement, ceci peut s'écrire

$$\begin{pmatrix} 1 & 1 & 2 & 0 \\ 2 & 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Si *une* erreur s'est introduite dans le mot, celui-ci a été modifié par *un multiple d'un vecteur de base!* Alors, le produit matriciel ci-dessus n'est pas nul, mais est égal à un multiple d'une colonne de la matrice de gauche. Il suffit donc de localiser l'erreur en calculant

$$\begin{pmatrix} 1 & 1 & 2 & 0 \\ 2 & 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

On obtient le double de la deuxième colonne de la matrice. Par conséquent, l'erreur introduite est $(0,2,0,0)$, et la bonne colonne se calcule en retranchant l'erreur:

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 1 \end{pmatrix}.$$

Il paraît évident, à ce stade, qu'un tel calcul peut être complètement automatisé, et qu'il fournit une méthode de décodage très rapide.

2 Les codes de Hamming

Les codes correcteurs d'erreurs ne sont pas seulement importants pour les transmissions, mais aussi (et surtout) pour la protection des informations stockées sous forme statique.

Un ordinateur, ou même une simple calculette, contient une quantité importante de données câblées. Celles-ci ne sont pas à l'abri de perturbations: il y a d'infimes traces d'uranium et de thorium radioactifs dans pratiquement tous les matériaux. Ces éléments lourds émettent des particules alpha, susceptibles de modifier le contenu des cellules de mémoire. Expérimentalement, on considère qu'une altération se produit une fois par cellule et par million d'années. Pour un ordinateur actuel, la période de sécurité est donc de quelques semaines. Avec un code correcteur d'erreur, la sécurité s'étend sur plusieurs années, car il faut seulement exclure l'altération de deux lettres dans un même mot. Le calcul est bien connu sous le nom de "paradoxe des anniversaires", et je recommande la lecture de l'excellent article de McEliece [3] sur ce sujet. En pratique, les ordinateurs réactualisent systématiquement leurs données, et sont donc virtuellement à l'abri d'erreurs dans leur mémoire morte.

Nous sommes en 1946–47, dans les laboratoires Bell. Le mathématicien Richard Hamming côtoie, et utilise "à temps perdu" l'un de ces fameux monstres mis au point pendant la guerre, fonctionnant avec des lampes et des relais. Ces machines ne sont pas munies de code correcteur d'erreur, mais seulement d'un code détecteur d'erreur appelé "two-out-of-five": les chiffres de 0 à 9 sont chargés sur des rampes de 5 lampes, dont deux sont allumées et trois éteintes. La détection des erreurs (fréquentes) est facile, et dans ce cas, on arrête la machine pour recharger les données. Pour sa recherche, Hamming n'a droit à la machine que pendant le week-end, alors qu'elle est en pilotage automatique. Si une panne se produit, il lui reste à attendre le retour des techniciens le lundi matin! Il se fixe alors comme mission de mettre au point un système de correction automatique des erreurs.

Sa démarche est analogue à celle du *décodage* du Sport-Toto du paragraphe précédent. Autrement dit, il aborde le problème par l'autre bout, en cherchant comment corriger les messages *avant de savoir comment les écrire!* Le calcul (cf. paragraphe suivant) est sans surprise: on tombe, en lieu et place de

$$(a, b, a+b, 2a+b)$$

sur l'alphabet $\{0,1,2\}$, sur un code linéaire de longueur 7 sur $\{0,1\}$

$$(a, b, c, d, b+c+d, a+c+d, a+b+d)$$

que les fameux diagrammes de Venn-Euler ont rendu célèbre il y a fort longtemps.

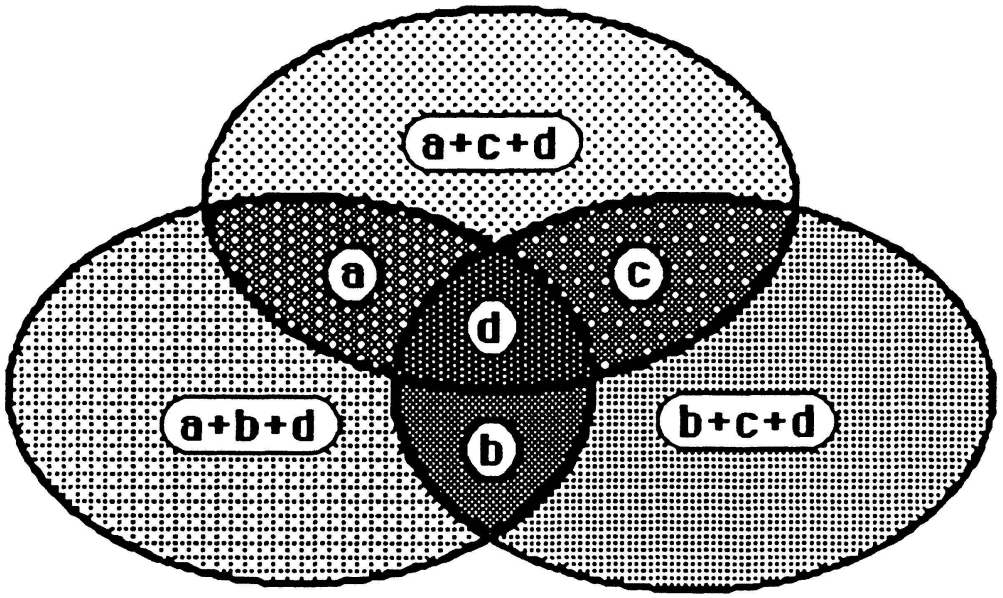

Fig. 2 Le code de Hamming (7,16,3)

Il a donc fallu attendre 1947, non seulement pour découvrir les semi-conducteurs, mais également pour avoir la possibilité de les employer à grande échelle. Nous verrons ci-après comment Hamming a ainsi d'un seul coup découvert toute une famille de codes, y compris celui du Sport-Toto!

3 Généralités sur les codes linéaires

Un code linéaire $[n, k]$ est un sous-espace vectoriel $\mathscr{C}$ de dimension k de $(\mathbf{F_q})^n$, $\mathbf{F_q}$ désignant le corps fini à q éléments. On le décrit généralement par la matrice $\mathbf{G}$ de l'injection du code: un message $m = (m_1, \ldots, m_k)$ est codé en $m\mathbf{G} = c = (c_1, \ldots, c_n)$.

Sur nos deux exemples précédents, les matrices de codage sont

$$\text{Sport-Toto:} \quad \mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\text{Hamming:} \quad \mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Il est essentiel d'observer ici que cette description matricielle n'obéit pas aux règles d'invariance usuelles. Les changements de base dans $(\mathbf{F_q})^n$ ne sont *PAS* autorisés, car ils modifient en général la distance du code! Seules les permutations des coordonnées, évidemment inoffensives, peuvent servir de relation d'équivalence pour les codes. En revanche, tout est permis dans $(\mathbf{F_q})^k$: la donnée d'un sous-espace vectoriel ne dépend pas du choix d'une base. Pratiquement, ceci signifie qu'on ne s'autorise à modifier la matrice $\mathbf{G}$ que par des opérations de lignes.

$\mathbf{G}$ est une matrice $(k \times n)$ de rang k, appelée *matrice de codage*. On lui associe une *matrice de contrôle* $\mathbf{H}$: c'est une matrice $((n-k) \times n)$ de rang $(n-k)$ qui annule $\mathbf{G}$. En termes techniques, $\mathbf{H}$ décrit le code $\mathcal{C}$ comme noyau d'une application linéaire, alors que $\mathbf{G}$ le présente comme image. Matriciellement, $\mathbf{G}$ et $\mathbf{H}$ sont liées par l'équation $\mathbf{H}\mathbf{G}^t = \mathbf{0}$, $\mathbf{0}$ désignant la matrice nulle de dimensions $((n-k) \times k)$. Le *code dual* $\mathcal{C}^\perp$ d'un code $\mathcal{C}$ est défini de façon évidente, en échangeant les rôles de $\mathbf{G}$ et $\mathbf{H}$.

On montre facilement qu'en permutant au besoin les coordonnées, on peut toujours donner à la matrice de codage la forme $\mathbf{G} = (\mathbf{I_k}\ \mathbf{C})$, permettant ainsi de donner directement une matrice de contrôle $\mathbf{H} = (-\mathbf{C}^t\ \mathbf{I_{n-k}})$.

La distance d'un code n'est pas visible sur la matrice $\mathbf{G}$. En revanche, la matrice $\mathbf{H}$ permet, en principe tout au moins, de la déterminer:

Théorème 1 *Un code linéaire a la distance $\geq d$ si et seulement si $(d-1)$ colonnes de la matrice de contrôle sont toujours linéairement indépendantes sur $\mathbf{F_q}$.*

Démonstration. Soit c un mot de poids $(d-1)$ dans le code. Alors $\mathbf{H}c^t = 0$, une combinaisaison linéaire nulle de $(d-1)$ colonnes de $\mathbf{H}$. **Cqfd.**

Le cas particulier $d = 3$ sur $\mathbf{F_2}$ est le plus parlant: il faut que les colonnes de $\mathbf{H}$ soient **distinctes!** C'est ainsi que Hamming découvrit le code [7,4]: il y a évidemment 7 colonnes de hauteur 3 distinctes sur $\mathbf{F_2}$, ce sont les entiers de 1 à 7 écrits en base 2. A partir de

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}, \text{ on détermine facilement}$$

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \text{ ou } (a+b+d, a+c+d, a, b+c+d, b, c, d).$$

Nul doute qu'avec ce procédé de codage, Richard Hamming passa pour un sorcier: *le contrôle écrit en base deux la position de l'erreur!*

Sur $\mathbf{F_q}$ il y a $q^m - 1$ colonnes non-nulles de hauteur m. En ne prenant qu'une colonne parmi celles qui sont proportionnelles, on obtient donc une matrice $\mathbf{H}$ de taille $(m \times \frac{q^m-1}{q-1})$. Le code de Hamming général est donc de longueur $\frac{q^m-1}{q-1}$, et de dimension $\frac{q^m-1}{q-1} - m$. Pour $q = 3$, les premiers codes de Hamming ont les longueurs 4 (c'est notre exemple de départ), et 13 (Il existe donc un code garantissant 12 points au vrai Sport-Toto!).

Quelques mots enfin sur le décodage: à réception d'un message m, on commence par contrôler s'il appartient au code en calculant son *syndrome* $s = \mathbf{H}m^t$. Le syndrome d'un mot est une information suffisante pour retrouver les erreurs que le code est apte à corriger (c'est le théorème 1). Mais, sauf pour les codes de Hamming pour lesquels la distance 3 permet les sorcelleries ci-dessus, la reconstitution de l'erreur est un problème complexe qui n'a pu être résolu de façon satisfaisante que pour des codes ayant des propriétés

mathématiques supplémentaires. La même remarque s'applique à la détermination de la distance d'un code: l'information est complètement contenue dans la matrice **H**, mais son extraction est d'une complexité encore plus grande!

Pour les codes à utiliser dans les sondes spatiales, de bonnes propriétés de distance sont plus importantes qu'un décodage rapide. Les codes cycliques, qui feront l'objet du prochain chapitre, répondent particulièrement bien à ces exigences, avec en outre un codage rapide, simple et robuste. De plus, leur structure mathématique est fascinante et riche d'enseignements, ce qui en fait l'un des chapitres fétiches de la théorie des codes.

4 Codes cycliques

4.1 Définitions et propriétés algébriques

Un code est dit *cyclique* si toute permutation circulaire d'un mot du code appartient aussi au code. Dans le cas d'un code linéaire cyclique de longueur n sur $\mathbf{F_q}$, une judicieuse interprétation consiste à identifier $(\mathbf{F_q})^n$ aux polynômes de degré $< n$, en transcrivant[2]) $f = (f_0, f_1, \ldots, f_{n-1})$ en un polynôme $f(x) = f_0 + f_1 x + \ldots + f_{n-1}x^{n-1}$. Ceci permet d'utiliser la structure multiplicative de l'anneau des polynômes: la permutation circulaire des coefficients d'un polynôme $f(x)$ consiste à prendre *le reste de la division de* $xf(x)$ *par* $(x^n - 1)$. En d'autres termes, un code cyclique est un *IDÉAL* de l'anneau $\mathbf{F_q}[x]/(x^n - 1)$. Bien que cet anneau ne soit pas intègre, on a cependant une classification de ses idéaux:

Théorème 2 *Tout idéal* $\mathcal{C}$ *de l'anneau* $\mathbf{F_q}[x]/(x^n - 1)$ *est principal: il existe un unique polynôme* $g(x)$ *de degré* $< n$ *satisfaisant*

1. $g(x)$ *est de coefficient dominant* 1.
2. $\mathcal{C} = (g(x))$.
3. $g(x)$ *divise* $(x^n - 1)$.

Démonstration. On prend pour $g(x)$ le polynôme de plus petit degré à coefficient dominant 1 contenu dans $\mathcal{C}$. Par linéarité, $g(x)$ est unique. De plus, comme $\mathcal{C}$ est un idéal, $g(0) \neq 0$! Soit alors $c(x) \subset \mathcal{C}$. La division euclidienne montre que le reste de division de $c(x)$ par $g(x)$ est nul, et ceci démontre 2. De même, le reste de division de $(x^n - 1)$ par $g(x)$ est nul, démontrant 3. **Cqfd.**

Il y a donc correspondance entre les codes cycliques de longueur n sur $\mathbf{F_q}$ et les diviseurs de $(x^n - 1)$. Si l'on suppose que le polynôme générateur $g(x)$ est de degré $(n - k)$, il est facile de voir que le code est de dimension k. Pour s'en convaincre, le plus simple consiste à exhiber directement les matrices de codage et de contrôle:

Théorème 3 *Soit* $g(x)$, *de degré* $(n - k)$, *le générateur d'un code cyclique, et soit* $h(x)$ *le polynôme défini par* $(x^n - 1) = g(x)h(x)$. *Alors des matrices de codage et de contrôle*

2) Observer le décalage des indices.

sont données par

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & . & . & g_{n-k} & 0 & 0 & 0 \\ 0 & g_0 & g_1 & . & . & g_{n-k} & 0 & 0 \\ . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & g_0 & g_1 & . & . & g_{n-k} \end{pmatrix} \text{ et}$$

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & h_k & . & . & h_1 & h_0 \\ 0 & 0 & h_k & . & . & h_1 & h_0 & 0 \\ . & . & . & . & . & . & . & . \\ h_k & . & . & h_1 & h_0 & 0 & 0 & 0 \end{pmatrix}.$$

Démonstration. $\mathbf{G}$ est une matrice $(k \times n)$ de rang k, $\mathbf{H}$ une matrice $((n-k) \times n)$ de rang $(n-k)$. Il suffit alors de remarquer que les lignes de $\mathbf{G}$ sont divisibles par $g(x)$, et que $\mathbf{G}\mathbf{H}^t = \mathbf{0}$. Cette dernière propriété exprime simplement que $g(x)h(x) = (x^n - 1)$. **Cqfd.**

Le code dual d'un code cyclique est cyclique, comme on le constate immédiatement sur les matrices $\mathbf{G}$ et $\mathbf{H}$. Le générateur du code dual est $x^k h(\frac{1}{x})$.

Exemple: Si $g(x) = 1+x+x^3$, on trouve $h(x) = 1+x+x^2+x^4$. Ceci fournit la matrice de contrôle

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

dont les colonnes sont bien distinctes. Il s'agit donc d'une présentation du code de Hamming comme code cyclique.

Pour simplifier les considérations algébriques, on suppose *toujours*, dans la théorie des codes cycliques, que *n est premier à q*. L'explication de la pertinence de cette hypothèse est fournie par le fait que dans cette situation, les racines du polynôme sont toujours *simples*. En conséquence, les polynômes $g(x)$ et $h(x)$ sont premiers entre eux, une situation qui facilite grandement la description des codes.

4.2 Racines des codes cycliques

Si $g(x)$ est le générateur d'un code cyclique, toutes ses racines, dans une extension convenable de $\mathbf{F_q}$, sont des puissances d'une racine *primitive* α de l'unité. En d'autres termes:

$$(x^n - 1) = (x-1)(x-\alpha)(x-\alpha^2)\ldots(x-\alpha^{n-1}),$$

$$g(x) = (x-\alpha^{i_1})(x-\alpha^{i_2})\ldots(x-\alpha^{i_{n-k}})$$

A chaque polynôme $g(x)$ est donc attachée une liste d'indices $I = \{i_1, \ldots, i_{n-k}\}$. Cette liste dépend du choix de α, mais sa propriété fondamentale n'en dépend pas. Elle fait l'objet du

Théorème 4 *L'ensemble $I = \{i_1, \ldots, i_{n-k}\}$ est invariant par multiplication par q (modulo n).*

Démonstration. La preuve utilise l'identité remarquable des corps finis:

$$g(x^q) = (g(x))^q \text{ dans } \mathbf{F}_q[x]$$

Celle-ci montre que $g(\alpha^i) = 0 \Leftrightarrow g(\alpha^{qi}) = 0$. **Cqfd.**

Exemple. Prenons le code cyclique de longueur 7 sur $\mathbf{F}_2$, de générateur $(1+x+x^3)$. La factorisation de (x^7-1) sur $\mathbf{F}_2$ est

$$(x^7-1) = (x-1)(x^3+x+1)(x^3+x^2+1)\,.$$

La décomposition de l'ensemble $\{0,1,2,3,4,5,6\}$ en parties invariantes par multiplication par 2 (modulo 7) est $\{0\}\cup\{1,2,4\}\cup\{3,5,6\}$. Selon le choix d'une racine primitive 7-ième de l'unité sur $\mathbf{F}_2$, $I = \{1,2,4\}$ ou $I = \{3,5,6\}$.

L'ensemble I attaché à un code cyclique permet de donner une estimation de la distance du code. C'est le

Théorème 5 *Soit $\mathcal{C}$ un code cyclique de longueur n sur $\mathbf{F}_q$, de générateur $g(x)$, tel que $I = \{i_1, \ldots, i_{n-k}\}$. Si I contient $(d-1)$ entiers consécutifs, la distance de $\mathcal{C}$ est supérieure ou égale à d.*

Démonstration. Soit $c(x)$ un mot de poids $< d$ dans $\mathcal{C}$, et soient donc $\{c_{j_1}, \ldots, c_{j_{d-1}}\}$ les coefficients potentiellement non-nuls de $c(x)$. Désignons par $(s+1, s+2, \ldots, s+d-1)$ les entiers consécutifs contenus dans I. $c(x)$ doit s'annuler sur les racines de $g(x)$, et ceci fournit un système homogène de $(d-1)$ équations linéaires à $(d-1)$ inconnues $\{c_{j_r}\}$. Le déterminant de ce système est

$$\begin{vmatrix} \alpha^{(s+1)j_1} & \alpha^{(s+1)j_2} & \ldots & \alpha^{(s+1)j_{d-1}} \\ \alpha^{(s+2)j_1} & \alpha^{(s+2)j_2} & \ldots & \alpha^{(s+2)j_{d-1}} \\ \vdots & \vdots & \ldots & \vdots \\ \alpha^{(s+d-1)j_1} & \alpha^{(s+d-1)j_2} & \ldots & \alpha^{(s+d-1)j_{d-1}} \end{vmatrix}$$

$$= \alpha^{s(j_1+j_2+\ldots+j_{d-1})} \begin{vmatrix} \alpha^{j_1} & \alpha^{j_2} & \ldots & \alpha^{j_{d-1}} \\ \alpha^{2j_1} & \alpha^{2j_2} & \ldots & \alpha^{2j_{d-1}} \\ \vdots & \vdots & \ldots & \vdots \\ \alpha^{(d-1)j_1} & \alpha^{(d-1)j_2} & \ldots & \alpha^{(d-1)j_{d-1}} \end{vmatrix}.$$

Il s'agit d'un déterminant de Vandermonde, et comme les α^{j_r} sont distincts (α est une racine primitive de l'unité), il est non-nul. Par conséquent, $c(x) = 0$. **Cqfd.**

Exemples. Si l'on prend sur $\mathbf{F}_2$ le polynôme minimal d'une racine primitive n-ième de l'unité comme générateur, le code a au moins la distance 3, car l'ensemble I, qui est invariant par multiplication par 2, contient toujours au moins les entiers 1 et 2. Les codes de Hamming sur $\mathbf{F}_2$ s'obtiennent de cette manière, ils sont donc cycliques. Mais la distance peut être plus grande que la valeur fournie par le théorème, comme le montre le cas du code de Golay que nous étudierons ci-après.

4.3 Le code de Golay

L'ingénieur neuchâtelois Marcel Golay fit en 1949, alors qu'il travaillait aux Etats-Unis comme spécialiste du radar, la découverte de deux codes exceptionnels [2]. Tous deux sont des codes *parfaits*: ils garantissent $(n - e)$ points au "Sport-Toto sur $\mathbf{F_q}$" avec le minimum absolu du nombre de colonnes nécessaire. Le premier est un code [23, 12, 7] sur $\mathbf{F_2}$ ($e = 3$), le deuxième un code [11, 6, 5] sur $\mathbf{F_3}$ ($e = 2$). Rappelons ici que les codes de Hamming sont parfaits avec ($e = 1$). Ce n'est qu'en 1971 que les mathématiciens ont pu démontrer qu'un code parfait non-trivial avec ($e > 1$) est nécessairement un code [23, 12, 7] sur $\mathbf{F_2}$ ou [11, 6, 5] sur $\mathbf{F_3}$!

Le code de Golay prolongé [24, 12, 8] sur $\mathbf{F_2}$ est le plus spectaculaire, car il corrige 3 erreurs et en détecte 4. Si on le compare à 3 copies du code de Hamming prolongé [8, 4, 4], le nombre de mots est le même, mais la capacité de correction des erreurs est bien meilleure (3 erreurs dans un mot de 24 lettres *vs.* 1 erreur dans chaque groupe de 8 lettres). Il s'agit donc d'un objet combinatoire exceptionnel, dont on a d'ailleurs pu démontrer *l'unicité.*

Les résultats précédents permettent de donner une description du code de Golay [23, 12, 7] sur $\mathbf{F_2}$ comme code cyclique.[3]) On prend le polynôme minimal d'une racine 23-ième de l'unité sur $\mathbf{F_2}$. La décomposition de l'ensemble $\{0, \ldots, 22\}$ en parties invariantes par multiplication par 2 (modulo 23) est

$$\{0\} \cup \{1,2,3,4,6,8,9,12,13,16,18\} \cup \{5,7,10,11,14,15,17,19,20,21,22\}.$$

Le polynôme $g(x)$ est donc de degré 11, fournissant un code [23, 12]. La distance est au moins 5, car les deux ensembles d'indices possibles contiennent 4 entiers consécutifs. Un choix possible est donné par[4])

$$g(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11}$$

En ajoutant un contrôle de parité, on obtient un code [24, 12] dont il est facile de montrer qu'il est auto-dual. Les poids sont donc tous divisibles par 4 (Exercice facile). La distance du code prolongé est donc égale à 8, et le code de départ a donc bien la distance 7, puisque $g(x)$ a le poids 7.

En outre, le code de Golay, 3-correcteur d'erreurs grâce à sa distance 7, est *parfait.* En effet:

$$\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 2^{11}$$

(Ce calcul est analogue à celui des sphères d'influence pour le Sport-Toto).

Ce code a été utilisé par les sondes Voyager pour transmettre les fabuleuses photos de la planète Jupiter et de ses satellites. Une éruption volcanique sur Io a même été transmise *en direct.*

---

3) Ce n'est pas la démarche de Golay. Pour connaître l'histoire du code de Golay, avec toutes ses retombées en géométrie et en théorie des groupes, il vaut la peine de se référer à l'excellent livre de Thompson [4].

4) Il existe une justification mathématique intéressante permettant de privilégier ce choix, basée sur les idempotents des codes cycliques. Une bonne référence est [5].

4.4 Codes de Reed-Solomon

Si l'on prend sur $\mathbf{F_q}$ un code de longueur $n = q - 1$, le polynôme $(x^n - 1)$ a pour racines tous les éléments non-nuls de $\mathbf{F_q}$. En prenant dans $\mathbf{F_q}$ un générateur multiplicatif α, on définit le *code de Reed-Solomon à distance d sur* $\mathbf{F_q}$ par le polynôme

$$g(x) = (x - \alpha)(x - \alpha^2) \ldots (x - \alpha^{d-1}) .$$

Il est clair que la distance de ce code est bien d, car $I = \{1, 2, \ldots, d - 1\}$ est formé de $(d - 1)$ entiers consécutifs, et $g(x)$ a au plus d coefficients non-nuls!

Exemple. Pour le code de Reed-Solomon à distance 3 sur $\mathbf{F}_7$, $g(x) = (x - 3^1)(x - 3^2) = 6 + 2x + x^2$ et $h(x) = 1 + 2x + 5x^2 + 5x^3 + x^4$.

La dimension du code de Reed-Solomon est $(n - d + 1)$, c'est la plus grande dimension possible pour un code de longueur n à distance d. En effet: Pour tout code linéaire $[n, k, d]$, $k \leq (n - d + 1)$ (Exercice facile).

Définition: Un code linéaire $[n, k]$ est dit MDS (maximum distance separable) si $d = (n - k + 1)$.

Théorème 6 *Les propriétés suivantes d'un code linéaire* $[n, k]$ *sont équivalentes*:

1. *Le code est MDS.*
2. *Le code dual est MDS.*
3. *Tous les* $(k \times k)$ *mineurs d'une matrice de codage sont inversibles.*
4. *Tout* k*-tuple de lettres d'un mot du code permet de le reconstituer.*

Démonstration. Si l'on supprime $(d - 1)$ coordonnées, la distance du code reste ≥ 1. Le mineur restant de la matrice de codage est donc inversible. Ceci démontre 1) $\Rightarrow$ 3). Pour que le code dual soit MDS, sa distance doit être $(k + 1)$. Ab absurdo, un mot de poids $\leq k$ sera annulé par le mineur correspondant de la matrice de codage, démontrant 3) $\Rightarrow$ 2). Les arguments restants sont (presque) évidents.

La propriété MDS est, comme le montre le résultat ci-dessus, extrêmement utile pour les télécommunications à grande distance, car elle permet de reconstituer des messages tronqués ou parasités. Il n'est donc pas étonnant que la NASA y ait eu recours, comme nous verrons ci-après.

5 Les codes des sondes spatiales

Les télécommunications modernes doivent beaucoup aux pionniers du radar. Une de leurs ruses classiques pour se protéger contre les tronquages des messages consiste à écrire n messages codés de longueur n en ligne, puis à transmettre les colonnes de la matrice ainsi obtenue. De la sorte, un tronquage ou un effacement se ventile sur plusieurs mots, et peut être à l'arrivée réparé par un code correcteur d'erreurs.

Jusqu'à Jupiter, les images des sondes Voyager ont été transmises à l'aide du code de Golay. Mais pour son périple en direction des planètes lointaines Saturne, Uranus, et Neptune, la sonde Voyager II a été reprogrammée à l'aide d'un code de Reed-Solomon, pour des raisons de faiblesse du signal. Il fallait augmenter la capacité de correction, et à défaut de codes parfaits, ce sont les codes MDS qui offrent les avantages les plus

décisifs. Pour la sonde, la NASA a opté pour un code de Reed-Solomon à distance 15 sur $\mathbf{F}_{32}$.

C'est ici qu'intervient une autre astuce: $\mathbf{F}_{32}$ est un espace vectoriel de dimension 5 sur $\mathbf{F}_2$. Les *lettres* du code sont donc transmises comme *mots* de longueur 5. Un message du code est donc transmis avec l'alphabet $\mathbf{F}_2$, mais avec la longueur 155. Comme le code de Reed-Solomon est 7-correcteur, on constate par exemple que l'effacement de 30 lettres consécutives dans un mot de 155 lettres peut être reconstitué.

Une dernière ruse, encore plus diabolique, pour les codes de Reed-Solomon est de nature mathématique. Elle fournit un codage rapide et performant dans la sonde spatiale elle-même. De plus, elle permet de moduler la distance du code sans modifier le câblage!

Théorème 7 *Soit $a(x)$ un polynôme de degré $\leq (q-1-d)$ sur $\mathbf{F}_q$. Soit α un générateur multiplicatif de $\mathbf{F}_q$. On code $a(x)$ en un mot de longueur $q-1$*

$$c(x) = (a(1), a(\alpha), a(\alpha^2), \ldots, a(\alpha^{q-2})) .$$

Alors le code ainsi obtenu est le code de Reed-Solomon à distance d sur $\mathbf{F}_q$.

Démonstration. On pose $n = q-1$ pour simplifier les notations. Le code obtenu est clairement linéaire. De plus, il est cyclique, car le codage du polynôme $a(\alpha x)$ est

$$(a(\alpha), a(\alpha^2), a(\alpha^3), \ldots, a(1)).$$

Reste à montrer que le code a les racines voulues par le code de Reed-Solomon. Un mot du code peut s'écrire

$$c(x) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-d} a_j \alpha^{ij} x^i .$$

Il faut démontrer que $(0 < m < d) \Rightarrow c(\alpha^m) = 0$. Dans $c(\alpha^m)$, le coefficient de a_j est $\sum_{i=0}^{n-1} \alpha^{(j+m)i}$; il est donc nul si $(0 < j+m < n)$. En effet, toute racine n-ième de l'unité $\neq 1$ est une racine de $(1 + x + x^2 + \ldots + x^{n-1})$. Mais si $(0 \leq j \leq n-d)$, alors $(0 < j+m < n)$. **Cqfd.**

La méthode choisie pour le codage consiste à évaluer des polynômes. Ce sont justement des opérations très faciles à implanter et à miniaturiser (schéma de Horner). D'autre part, en modifiant le degré $(q-1-d)$ des messages à coder, on modifie en même temps la distance du code. Mais le schéma de Horner est inchangé! Par ailleurs, on peut directement construire une matrice de contrôle "flexible". Signalons cependant que le décodage est particulièrement difficile pour les codes à distance > 3, et qu'il faut avoir recours à un arsenal mathématique extrêmement sophistiqué. Mais ceci se passe sur Terre, et non dans la sonde spatiale. C'est donc une autre histoire.

Références

[1] Arnoux, Pierre: *Minitel, codage de l'information et corps finis.* Pour la Science (mars 1988).

[2] Golay, Marcel: *Notes on Digital Coding.* PIEEE 37 (1949) 637.

[3] McEliece, R.J.: *The reliability of computer memories.* Scientific American (January 1985).

[4] Thompson, T.M.: *From error-correcting codes through sphere packings to simple groups.* The Carus Mathematical Monographs 21, Mathematical Association of America (1983).

[5] Van Lint, J.H.: *Introduction to Coding Theory.* Graduate Texts in Mathematics 86, Springer-Verlag (1982).

François Sigrist
Institut de mathématiques
Université de Neuchâtel
2007 Neuchâtel
Suisse