

6

Codes correcteurs

La partie élémentaire est constituée des sections 6.1 à 6.4. Elle explique la nécessité des codes correcteurs, introduit le corps fini \mathbb{F}_2 de deux éléments et décrit une famille de codes correcteurs non triviaux, les codes de Hamming. L'arithmétique du corps \mathbb{F}_2 sera nouvelle pour la plupart des lecteurs, mais la partie élémentaire n'utilise que les concepts relatifs aux espaces vectoriels et à l'algèbre matricielle (sur \mathbb{F}_2). Elle peut être couverte en trois heures. Les sections 6.5 et 6.6 forment la partie avancée. Les corps finis \mathbb{F}_{p^r} , où p est premier, sont construits par l'introduction sur les polynômes de degré inférieur à r d'une multiplication modulo un polynôme irréductible. La pratique sur quelques exemples permet d'assimiler facilement ce concept qui peut, au premier abord, sembler déroutant. Les codes de Reed-Solomon sont enfin présentés à la dernière section. Il faut au moins trois heures supplémentaires pour couvrir la partie avancée.

6.1 Introduction : numériser, détecter et corriger

La transmission d'information à distance a été utilisée très tôt dans l'histoire de l'humanité¹. La découverte des lois physiques de l'électromagnétisme et de leurs applications a permis d'envoyer des messages sous forme électrique dès la seconde moitié du XIX^e siècle. Que la communication se fasse directement à l'aide d'une langue humaine (tel le français ou l'anglais) ou soit préalablement encodée (par exemple, à l'aide du code Morse (1836)), l'utilité de pouvoir détecter et corriger des erreurs lors de la transmission se fait rapidement sentir.

Une première méthode permettant d'améliorer la fidélité de la transmission d'un message possède une importance historique. Au début de la téléphonie (avec ou sans fil), la qualité de la transmission laissait beaucoup à désirer. Il était donc usuel d'épeler

¹Selon la légende, le soldat chargé de rapporter la victoire des Athéniens sur les Perses en l'an 490 aurait couru la distance entre Marathon et Athènes et serait mort d'épuisement à son arrivée. La longueur du marathon olympique est maintenant de 41,195 km.

un mot en remplaçant chaque lettre par un mot dont la première lettre coïncidait avec la lettre à épeler. Ainsi, pour transmettre le mot « erreur », l'interlocuteur aurait dit les mots « Echo, Roméo, Roméo, Écho, Uniforme, Roméo ». Les armées américaine et britannique avaient de tels « alphabets » dès la Première Guerre mondiale. Ce code pour améliorer la transmission d'un message multiplie l'information; on espère que le récepteur puisse extraire du message codé (Écho, Roméo, Roméo, Écho, Uniforme, Roméo) le message original (« erreur »), et ce, avec plus de constance et de précision que si le mot « erreur » avait été simplement dit ou épelé. Cette « multiplication de l'information » ou *redondance* est la clé de tout code détecteur et correcteur.

Notre second exemple sera celui d'un code détecteur : il permet de diagnostiquer qu'une erreur a été commise lors de la transmission, mais pas de corriger cette erreur. En informatique, il est usuel de remplacer les caractères de notre alphabet étendu (a, b, c, ..., A, B, C, ..., 0, 1, 2, ..., +, -, :, ;, ...) par un chiffre entre 0 et 127. En représentation binaire, il faut sept caractères 0 ou 1 (chacun appelé *bit*, une contraction de *binary digit*) pour étiqueter ces $2^7 = 128$ caractères. Par exemple, supposons que la lettre *a* corresponde au nombre 97. Puisque $97 = 64 + 32 + 1 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^0$, la lettre *a* est encodée comme 1100001. Ainsi, une correspondance possible est donnée par le tableau suivant.

	décimal	binaire	parité+binaire
A	65	1000001	01000001
B	66	1000010	01000010
C	67	1000011	11000011
:	:	:	:
a	97	1100001	11100001
b	98	1100010	11100010
c	99	1100011	01100011
:	:	:	:

Pour détecter une erreur, on ajoute au code de sept bits un huitième bit, dit *bit de parité*. Il est placé à gauche des sept bits initiaux. Ce huitième bit est choisi de façon à ce que la somme des huit bits du code soit paire. Par exemple, la somme des sept bits de « A » est $1 + 0 + 0 + 0 + 0 + 0 + 1 = 2$, le bit de parité sera alors 0, et « A » sera représenté par 01000001. Cependant, la somme des sept bits de « a » est $1 + 1 + 0 + 0 + 0 + 0 + 1 = 3$, et « a » sera représenté par les huit bits 11100001. Ce bit de parité est un code de détection. Il permet de détecter qu'une erreur a été commise lors de la transmission, mais il ne permet pas de la corriger, car le récepteur ne sait pas lequel des huit bits est le bit fautif. Le récepteur, constatant l'erreur, peut cependant demander à ce que le caractère lui soit retransmis. Notons que ce code détecteur repose sur l'hypothèse qu'au maximum un bit

est erroné. Cette hypothèse est raisonnable si la transmission est presque parfaite et que la probabilité de deux erreurs au sein d'une transmission de huit bits est presque nulle.

Le troisième exemple présente une idée simple pour construire un code correcteur, c'est-à-dire un code permettant de détecter *et* de corriger une erreur. Il consiste à répéter la totalité du message suffisamment de fois. Par exemple, tous les caractères d'un texte pourraient être répétés deux fois. Ainsi, le mot « erreur » pourrait être transmis sous la forme « eerrrreeuurr ». Cette première version de ce code simple n'est cependant pas un code correcteur, car, même si on suppose qu'au plus une lettre par paire puisse être erronée, il ne permet pas la correction des erreurs. Quel était le message original si nous recevons « ââgmee » ? Était-ce « âge » ou « âme » ? Pour faire de cette idée simple un code *correcteur*, il suffit de répéter trois fois chaque lettre. Si l'hypothèse d'au plus une erreur par groupe de trois lettres est raisonnable, alors « âââgmeeee » sera décodé en « âme ». En effet, même si les trois lettres « gmm » ne coïncident pas, une seule est erronée, par hypothèse, et les trois lettres originales ne peuvent donc être que « mmm ». Voici donc un premier exemple de code correcteur ! Ce code est peu utilisé, car il est coûteux : il demande que toute l'information soit transmise en triple. Les codes que nous présenterons dans ce chapitre sont beaucoup plus économiques. Comme pour tous les codes, il n'est pas impossible que, dans un groupe de trois lettres, deux ou même trois soient erronées ; notre hypothèse est que ces événements sont *très* peu probables. Comme l'exercice 8 le montre, ce code fort simple conserve cependant un léger avantage sur le code de Hamming qui sera introduit à la section 6.3.

Les codes détecteurs et correcteurs existent donc depuis longtemps. Avec la numérisation de l'information, ces codes sont devenus de plus en plus nécessaires et aisés à mettre en œuvre. Leur nécessité est facile à comprendre quand on connaît la grandeur des fichiers typiques contenant des photos et de la musique. Voici, à la figure 6.1, une toute petite photo numérisée : les deux copies représentent le sommet de la tour d'un des bâtiments de l'Université de Montréal. À gauche, la photo est dans son format original. À droite, la même photo a été agrandie huit fois horizontalement et verticalement : on y voit clairement les pixels, c'est-à-dire les carrés de gris constant. En fait, ces deux photos ont été fractionnées en 72×72 carrés de gris constant (les pixels), et la profondeur du gris a été repérée sur une échelle de $256 = 2^8$ niveaux de gris (le blanc étant à une extrémité de cette échelle, le noir étant à l'autre). Il faut donc transmettre $72 \times 72 \times 8 = 41,472$ bits pour transmettre cette petite photo en noir et blanc. Et nous sommes fort loin d'une photo couleur grand format puisque les caméras numériques actuelles ont des capteurs de plus de 2000×3000 pixels couleur² !

Le son et, en particulier, la musique sont de plus en plus souvent numérisés. Par rapport à la numérisation des images, celle du son est plus difficile ... à visualiser. Il faut savoir que le son est une onde. Les vagues sur la mer sont une onde qui se propage

²Ceux qui s'intéressent à l'informatique sont habitués à voir les grandeurs de fichier et les capacités des espaces-disques mesurées en octets, en kilooctets (Ko, c'est-à-dire 1000 octets), en mégaoctets (1 Mo = 10^6 octets), en gigaoctets (1 Go = 10^9 octets), etc. Un octet est égal à huit bits, et notre petite photo noir et blanc occupe $41\,472/8$ o = 5184 o = 5,184 Ko.

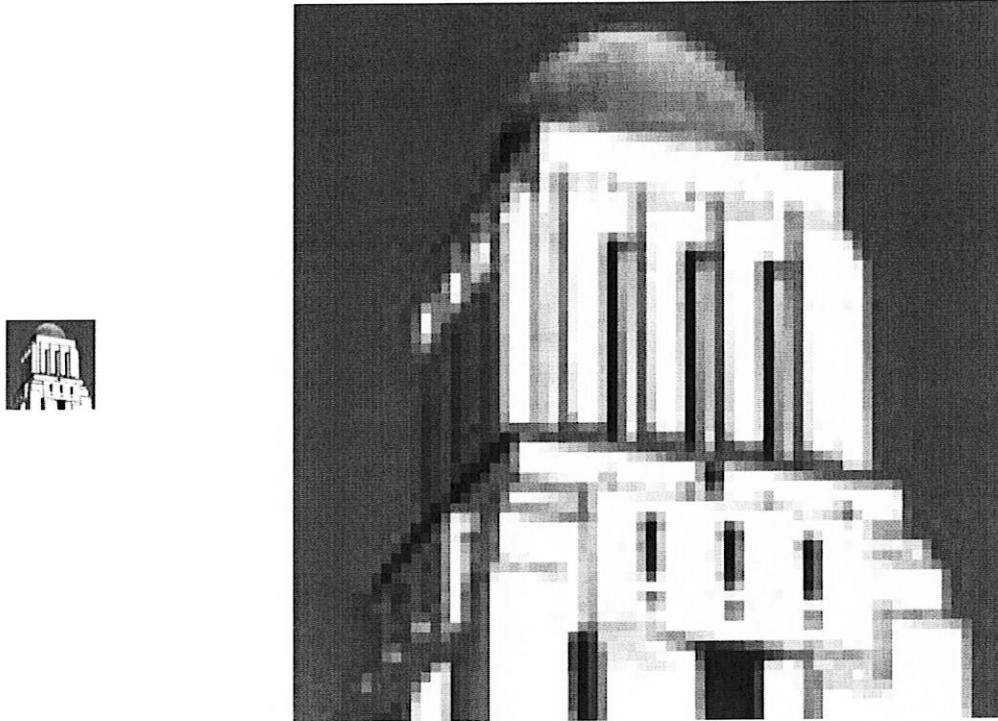


Fig. 6.1. Une photo numérisée : à gauche, l'« original » et à droite, la photo agrandie huit fois en hauteur et en largeur.

à la surface de l'eau, la lumière est une onde des champs électrique et magnétique, et le son est une onde de la densité de l'air. Si nous pouvions mesurer la densité en un point fixe de l'air près d'un piano qui sonne la note *la* au centre du clavier, nous verrions que la densité de l'air augmente et diminue environ 440 fois par seconde. La variation est minuscule, mais nos oreilles la détectent, la transforment en onde électrique, transmettent cette onde à notre cerveau, qui l'analyse et la « perçoit » comme le *la* au centre du clavier d'un piano. La figure 6.2 donne une représentation de cette onde de densité. (L'axe horizontal repère le temps alors que le vertical donne l'amplitude de l'onde. Les unités importent peu pour ce qui suit.) Lorsque la fonction est positive, la densité est supérieure à celle de l'air au repos (c'est-à-dire dans le silence absolu), et lorsque la fonction est négative, la densité est inférieure. La numérisation du son consiste à remplacer cette fonction continue par une fonction en escalier. Pour chaque période très courte de Δ seconde, la fonction « son » est remplacée pour toute cette période par la valeur au centre de cet intervalle de temps. Si Δ est assez court, l'approximation par la fonction escalier sera suffisamment bonne pour que l'oreille ne puisse pas percevoir

de différence entre la fonction son et la fonction escalier. (Une paire de fonctions son et escalier est représentée à la figure 6.3.) Cette étape réalisée, il suffit d'énumérer les valeurs de la fonction escalier pour chacune des périodes Δ .

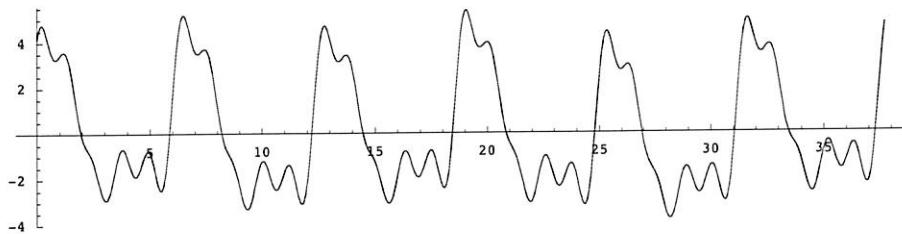


Fig. 6.2. La fonction « onde de densité » pendant une courte fraction de seconde

Sur un disque compact, le son est haché en 44 100 morceaux par seconde (l'équivalent des pixels de l'image ci-dessus), et la valeur de la fonction escalier durant chacune de ces $\Delta = \frac{1}{44\,100}$ seconde est repérée sur une échelle de $65\,536 = 2^{16}$ intensités³. Si on se rappelle que les disques compacts reproduisent le son en stéréo, chaque seconde de musique demande $44\,100 \times 16 \times 2 = 1\,411\,200$ bits, et un disque d'environ 70 minutes devra transmettre fidèlement $1\,411\,200 \times 60 \times 70 = 5\,927\,040\,000$ bits = 740 880 000 octets, soit approximativement 740 Mo. La possibilité de détecter et de corriger les erreurs semble attrayante.

Dans ce qui suit, deux codes classiques sont présentés, celui de Hamming et celui de Reed et Solomon. Le premier a été retenu par France Télécom pour la transmission du signal du Minitel, un précurseur d'Internet tel que nous le connaissons aujourd'hui. Le code de Reed-Solomon confère, aux disques compacts, leur grande robustesse. Le Consultative Committee for Space Data System, créé en 1982 pour harmoniser les pratiques des différentes agences spatiales, recommande également ce code pour la transmission de données par satellite.

6.2 Le corps \mathbb{F}_2

Pour comprendre le code de Hamming, nous devons connaître les règles de calcul applicables au corps à deux éléments \mathbb{F}_2 . Un corps est un ensemble de nombres sur lequel sont définies deux opérations appelées « addition » et « multiplication », opérations qui doivent satisfaire aux propriétés qui nous sont familières pour les nombres rationnels

³Sony et Philips sont les deux compagnies qui ont établi conjointement le standard du disque compact. Après avoir hésité entre une échelle fragmentée en 2^{14} ou en 2^{16} niveaux, les ingénieurs des deux compagnies ont opté pour l'échelle la plus fine, celle de 2^{16} niveaux [1]. Voir aussi le chapitre 10.

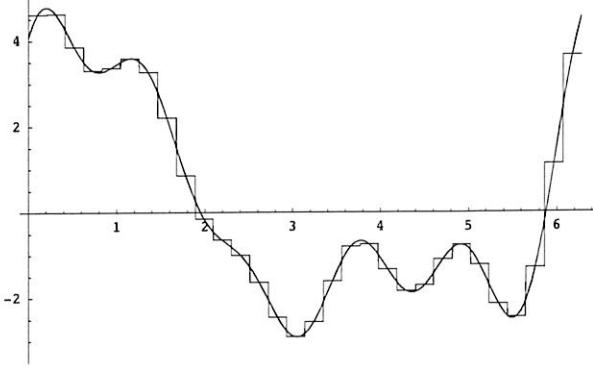


Fig. 6.3. La fonction « son » et une fonction « escalier » qui l'approche

et réels : associativité, commutativité, distributivité de la multiplication sur l'addition, existence de neutres pour l'addition et la multiplication, existence d'un inverse additif, existence d'un inverse multiplicatif pour tout élément non nul. Le lecteur connaît sûrement l'ensemble des rationnels \mathbb{Q} , celui des réels \mathbb{R} et probablement celui des complexes \mathbb{C} . Ces trois ensembles, munis des opérations + et \times usuelles, sont des corps. Mais il existe beaucoup d'autres corps que ceux-ci !

Nous examinerons la structure mathématique d'un corps en plus de détails à la section 6.5 ; il nous suffira, pour le présent exemple, de donner les règles de l'addition + et de la multiplication \times qui sont définies sur cet ensemble de deux éléments $\{0, 1\}$. Les tables d'addition et de multiplication se lisent comme suit

$+$	0	1	\times	0	1
0	0	1	0	0	0
1	1	0	1	0	1

(6.1)

Ces opérations répondent aux règles que l'on connaît pour les nombres rationnels \mathbb{Q} , les réels \mathbb{R} et les complexes \mathbb{C} : associativité, commutativité et distributivité, existence des neutres et des inverses. Par exemple, en utilisant les tables d'addition et de multiplication ci-dessus, on vérifie que, pour tout $x, y, z \in \mathbb{F}_2$, la distributivité

$$x \times (y + z) = x \times y + x \times z$$

est vérifiée. Puisque x, y et z prennent chacun deux valeurs, la distributivité représente huit relations correspondant aux huit valeurs possibles du triplet $(x, y, z) \in \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)\}$. Voici la vérification explicite de la distributivité pour le triplet $(x, y, z) = (1, 0, 1)$:

$$x \times (y + z) = 1 \times (0 + 1) = 1 \times 1 = 1$$

et

$$x \times y + x \times z = 1 \times 0 + 1 \times 1 = 0 + 1 = 1.$$

Comme dans \mathbb{Q} , \mathbb{R} et \mathbb{C} , 0 est le neutre pour $+$ et 1, le neutre pour \times . Tous les éléments possèdent un inverse additif. (Exercice : quel est l'inverse additif de 1?) Et tous les éléments de $\mathbb{F}_2 \setminus \{0\}$ possèdent un inverse multiplicatif. Dans ce dernier cas, l'affirmation est très simple, car il n'y a qu'un élément dans $\mathbb{F}_2 \setminus \{0\} = \{1\}$, et l'inverse multiplicatif de 1 est 1 puisque $1 \times 1 = 1$.

Tout comme il est possible de définir des espaces vectoriels \mathbb{R}^3 , \mathbb{R}^n ou \mathbb{C}^2 , il est possible de parler des espaces vectoriels à trois composantes, chacune d'entre elles étant un élément de \mathbb{F}_2 . Il est possible d'additionner deux vecteurs de \mathbb{F}_2^3 et d'en faire des combinaisons linéaires (avec coefficients dans \mathbb{F}_2 évidemment!). Par exemple :

$$\begin{aligned}(1, 0, 1) + (0, 1, 0) &= (1, 1, 1), \\ (1, 0, 1) + (0, 1, 1) &= (1, 1, 0),\end{aligned}$$

et

$$0 \cdot (1, 0, 1) + 1 \cdot (0, 1, 1) + 1 \cdot (1, 1, 0) = (1, 0, 1).$$

Puisque les composantes doivent être dans \mathbb{F}_2 et que seules les combinaisons linéaires avec coefficients dans \mathbb{F}_2 sont permises, le nombre de vecteurs dans \mathbb{F}_2^3 (et dans tout $\mathbb{F}_2^n, n < \infty$) est fini! Attention, même si la dimension de \mathbb{R}^3 est 3 (et donc finie), le nombre de vecteurs dans \mathbb{R}^3 est infini. Pour \mathbb{F}_2^3 , ce nombre de vecteurs est huit, et la liste *complète* des vecteurs de cet espace vectoriel est

$$\{(0, 0, 0), (0, 0, 1), (0, 1, 0), (1, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}.$$

(Exercice : rappeler la définition de dimension d'un espace vectoriel et calculer la dimension de \mathbb{F}_2^3 .) Les espaces vectoriels sur des corps finis tel \mathbb{F}_2 sont déroutants, car leurs vecteurs sont en nombre fini, et les cours d'algèbre linéaire n'en parlent pas ou peu même si beaucoup des méthodes qui y sont développées (le calcul matriciel, entre autres) s'appliquent à eux.

6.3 Le code de Hamming C(7, 4)

Voici un premier exemple de code correcteur moderne. Plutôt que les lettres usuelles (a, b, c, ...), il utilise un alphabet constitué de deux lettres (\mathbb{F}_2) que nous désignerons par 0 et 1⁴. Nous nous limiterons de plus à transmettre des mots ayant précisément quatre lettres (u_1, u_2, u_3, u_4). (Exercice : est-ce une restriction grave?) Notre vocabulaire ou

⁴Comme nous l'avons vu dans l'introduction, ceci n'est pas une restriction puisqu'il existe des « dictionnaires » traduisant notre alphabet latin en une série de caractères 0 et 1.

code $C \subset \mathbb{F}_2^4$ ne contiendra donc que 16 mots ou *éléments*. Plutôt que de transmettre ces quatre lettres, nous transmettrons les sept lettres suivantes :

$$\begin{aligned}v_1 &= u_1, \\v_2 &= u_2, \\v_3 &= u_3, \\v_4 &= u_4, \\v_5 &= u_1 + u_2 + u_4, \\v_6 &= u_1 + u_3 + u_4, \\v_7 &= u_2 + u_3 + u_4.\end{aligned}$$

Ainsi, pour transmettre le mot $(1, 0, 1, 1)$, nous enverrons le message

$$(v_1, v_2, v_3, v_4, v_5, v_6, v_7) = (1, 0, 1, 1, 0, 1, 0)$$

puisque

$$\begin{aligned}v_5 &= u_1 + u_2 + u_4 = 1 + 0 + 1 = 0, \\v_6 &= u_1 + u_3 + u_4 = 1 + 1 + 1 = 1, \\v_7 &= u_2 + u_3 + u_4 = 0 + 1 + 1 = 0.\end{aligned}$$

(Attention : « + » est l'addition dans \mathbb{F}_2 .)

Puisque les quatre premières composantes de (v_1, v_2, \dots, v_7) sont précisément les quatre lettres du mot à transmettre, à quoi peuvent bien servir les trois autres lettres ? Ces lettres sont *redondantes* et permettent de corriger *une lettre* erronée, quelle qu'elle soit. Comment ce « miracle » peut-il être accompli ?

En voici un exemple. Le récepteur reçoit les sept lettres $(w_1, w_2, \dots, w_7) = (1, 1, 1, 1, 0, 0, 0)$. Nous distinguons les v_i des w_i , car, lors de la transmission, une des lettres, disons v_j , peut avoir été corrompue. Alors $v_j \neq w_j$. À cause de la qualité de son canal de transmission, le récepteur peut, avec une bonne assurance, faire l'hypothèse qu'aucune ou au plus une lettre est erronée. Le récepteur calcule donc

$$\begin{aligned}W_5 &= w_1 + w_2 + w_4, \\W_6 &= w_1 + w_3 + w_4, \\W_7 &= w_2 + w_3 + w_4,\end{aligned}$$

et les compare avec les w_5, w_6 et w_7 qu'il a reçus. S'il n'y a pas eu d'erreur lors de la transmission, W_5, W_6, W_7 devraient coïncider avec w_5, w_6, w_7 . Voici ce calcul pour l'exemple présent :

$$\begin{aligned}W_5 &= w_1 + w_2 + w_4 = 1 + 1 + 1 = 1 = w_5, \\W_6 &= w_1 + w_3 + w_4 = 1 + 1 + 1 = 1 \neq w_6, \\W_7 &= w_2 + w_3 + w_4 = 1 + 1 + 1 = 1 \neq w_7.\end{aligned}\tag{6.2}$$

Le récepteur constate qu'une erreur a dû se produire, car deux des trois lettres reçues, w_5, w_6 et w_7 , ne reproduisent pas celles qu'il calcule, c'est-à-dire W_5, W_6, W_7 . Mais où est l'erreur ? Touche-t-elle une des quatre premières lettres du message original ou une des trois lettres ajoutées ? Il est facile d'exclure la possibilité que w_5 ou w_6 ou w_7 soit erronée. Si nous changeons *une seule* de ces trois lettres, au moins une des trois égalités (6.2) ne sera pas satisfaite. Il faut donc que la lettre erronée soit une des quatre premières. Quelle lettre, parmi w_1, w_2, w_3 et w_4 , peut-on changer de façon à corriger simultanément les deux dernières égalités fausses de (6.2) tout en préservant la première qui est juste ? La réponse est simple : la lettre à corriger est w_3 . En effet, la première somme ne contient pas w_3 et ne sera pas affectée par le changement de cette lettre. Les deux autres changeront et seront donc « corrigées ». Ainsi, même si les quatre premières lettres reçues par le récepteur sont $(w_1, w_2, w_3, w_4) = (1, 1, 1, 1)$, le message original (correct) devait être $(v_1, v_2, v_3, v_4) = (1, 1, 0, 1)$.

Dressons maintenant la liste de toutes les possibilités. Supposons que le récepteur reçoive les lettres (w_1, w_2, \dots, w_7) . La seule chose dont ce récepteur est assuré est que ces sept lettres $w_i, i = 1, \dots, 7$ coïncident avec les lettres $v_i = i, \dots, 7$, à l'exception, peut-être, d'une seule lettre (qu'il ne connaît cependant pas). Huit possibilités se présentent au récepteur. Les voici :

- (0) il n'y a aucune lettre erronée ;
- (1) w_1 est erronée ;
- (2) w_2 est erronée ;
- (3) w_3 est erronée ;
- (4) w_4 est erronée ;
- (5) w_5 est erronée ;
- (6) w_6 est erronée ;
- (7) w_7 est erronée.

À l'aide des lettres redondantes, le récepteur peut déterminer laquelle est juste. En effet, en calculant W_5, W_6 et W_7 comme ci-dessus, il pourra diagnostiquer lequel des huit cas $(i), i = 0, \dots, 7$, est le bon à l'aide du tableau suivant :

- (0) si $w_5 = W_5$ et $w_6 = W_6$ et $w_7 = W_7$;
- (1) si $w_5 \neq W_5$ et $w_6 \neq W_6$;
- (2) si $w_5 \neq W_5$ et $w_7 \neq W_7$;
- (3) si $w_6 \neq W_6$ et $w_7 \neq W_7$;
- (4) si $w_5 \neq W_5$ et $w_6 \neq W_6$ et $w_7 \neq W_7$;
- (5) si $w_5 \neq W_5$;
- (6) si $w_6 \neq W_6$;
- (7) si $w_7 \neq W_7$.

L'hypothèse qu'au maximum une lettre est erronée est cruciale. Si deux lettres pouvaient être erronées, alors le récepteur ne pourrait distinguer, par exemple, entre « w_1 est erronée » et « w_5 et w_6 sont toutes les deux erronées » et ne pourrait donc effectuer de correction. Connaissant, le cas échéant, la lettre erronée, il la corrigera, tronquera le message de ses trois dernières lettres, et les quatre lettres restantes seront à coup sûr le message que l'émetteur voulait transmettre. Le processus est donc symbolisé par

$$\begin{array}{c} \overline{(u_1, u_2, u_3, u_4) \in C \subset \mathbb{F}_2^4} \xrightarrow{\text{encodage}} \overline{(v_1, v_2, v_3, v_4, v_5, v_6, v_7) \in \mathbb{F}_2^7} \\ \xrightarrow{\text{transmission}} | (w_1, w_2, w_3, w_4, w_5, w_6, w_7) \in \mathbb{F}_2^7 | \\ \xrightarrow{\text{correction et décodage}} | (w'_1, w'_2, w'_3, w'_4) \in C \subset \mathbb{F}_2^4 | \end{array}$$

Comment le code de Hamming $C(7, 4)$ se compare-t-il aux autres codes correcteurs ? Cette question est trop vague. En effet, la qualité d'un code ne peut être jugée qu'en fonction des besoins : le taux d'erreur du canal de transmission, la longueur moyenne des messages à émettre, la rapidité d'encodage et de décodage requise, etc. Nous pouvons tout de même le comparer au code qui consiste à simplement répéter l'information envoyée. Par exemple, chacune des lettres $u_i, i = 1, 2, 3, 4$, peut être envoyée de façon répétée jusqu'à ce qu'un bon niveau de confiance soit atteint. Reprenons l'hypothèse qu'une seule erreur puisse se produire dans quelques bits (< 15 bits). Alors, chacune des quatre lettres peut être répétée. Comme nous l'avons déjà vu, si chacune des u_i est envoyée deux fois, seule une détection d'erreur peut être accomplie. Il faut transmettre chaque lettre trois fois pour assurer la correction d'une erreur. Transmettre trois fois les quatre lettres requiert 12 bits, et le code de Hamming en requiert sept. Il s'agit d'une amélioration significative.

6.4 Les codes de Hamming $C(2^k - 1, 2^k - k - 1)$

Le code de Hamming $C(7, 4)$ que nous venons d'étudier est le premier d'une famille de codes de Hamming $C(2^k - 1, 2^k - k - 1)$ que nous allons maintenant introduire. Tous ces codes ne permettent la correction que d'une erreur. Les deux nombres $2^k - 1$ et $2^k - k - 1$ indiquent respectivement la *longueur* des mots du code et la *dimension* du sous-espace formé par les mots transmis. Ainsi, pour $k = 3$, on retrouve le code $C(7, 4)$ où 7 est la longueur des mots transmis, c'est-à-dire que les mots transmis $\in \mathbb{F}_2^7$, alors que les mots (sans erreur) forment un sous-espace de dimension 4 isomorphe à \mathbb{F}_2^4 .

Deux matrices jouent un rôle important dans la description du code de Hamming (et de tous les codes dits linéaires, dont le code de Reed-Solomon fait partie) : la *matrice génératrice* G et la *matrice de contrôle* H . La matrice génératrice G_k est une matrice $(2^k - k - 1) \times (2^k - 1)$ et possède comme lignes une base du sous-espace isomorphe à $\mathbb{F}_2^{(2^k - k - 1)}$ des mots du code C , c'est-à-dire des mots sans erreur. Tout mot du code sera une combinaison linéaire de ces lignes. Pour $C(7, 4)$, la matrice G_3 peut être choisie sous la forme

$$G_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Par exemple, la première ligne de G_3 correspond au mot du code tel que $u_1 = 1$ et $u_2 = u_3 = u_4 = 0$. Alors, par les règles que nous avons choisies, $v_1 = 1, v_2 = v_3 = v_4 = 0$, et $v_5 = u_1 + u_2 + u_4 = 1, v_6 = u_1 + u_3 + u_4 = 1$ et $v_7 = u_2 + u_3 + u_4 = 0$. Ce sont les éléments de la première ligne. Les 16 mots du code C seront déduits des 16 différentes combinaisons linéaires possibles des quatre lignes de G_3 . Puisque G est définie à l'aide du choix d'une base, G n'est pas définie uniquement.

La matrice de contrôle H est une matrice $k \times (2^k - 1)$ dont les k lignes forment une base du complément orthogonal du sous-espace engendré par les lignes de G . Le produit scalaire est le produit usuel : si $v, w \in \mathbb{F}_2^n$, alors $(v, w) = \sum_{i=1}^n v_i w_i \in \mathbb{F}_2$. (L'appendice à la fin de ce chapitre rappelle la définition de produit scalaire et souligne les différences importantes entre cette structure sur les corps usuels (\mathbb{Q}, \mathbb{R} et \mathbb{C}) et sur les corps finis. Certaines de ces différences ne sont pas très intuitives !) Pour $C(7, 4)$ et le choix de G_3 ci-dessus, la matrice de contrôle H_3 peut être choisie ainsi :

$$H_3 = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Puisque les lignes de G et de H sont orthogonales deux à deux, les matrices G et H satisfont à

$$GH^t = 0. \quad (6.3)$$

Par exemple, pour $k = 3$:

$$G_3 H_3^t = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}}_{4 \times 7} \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{7 \times 3} = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{4 \times 3}.$$

Le code de Hamming général $C(2^k - 1, 2^k - k - 1)$ est défini par la donnée de la matrice de contrôle H . Cette matrice possède comme *vecteurs colonnes* tous les vecteurs non nuls de \mathbb{F}_2^k . Puisque \mathbb{F}_2^k contient 2^k vecteurs (dont le vecteur nul), H est bien une matrice $k \times (2^k - 1)$. La matrice H_3 en est un exemple. Comme nous l'avons dit, les lignes de la matrice génératrice G engendrent le complément orthogonal des lignes de H . Ceci termine la définition du code de Hamming $C(2^k - 1, 2^k - k - 1)$.

Voici comment l'encodage et le décodage sont faits.

Dans le choix de la matrice G_3 que nous avons fait, chacune des lignes correspond à un des mots à transmettre suivants : $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$ et $(0, 0, 0, 1)$. Pour obtenir le mot général (u_1, u_2, u_3, u_4) , il suffit de faire une combinaison linéaire des quatre lignes de G_3 :

$$(u_1 \ u_2 \ u_3 \ u_4) G_3 \in \mathbb{F}_2^7.$$

(Exercice : vérifier que le produit matriciel $(u_1 \ u_2 \ u_3 \ u_4) G_3$ donne bien une matrice 1×7 .) L'encodage de $u \in \mathbb{F}_2^{2^k-k-1}$ du code $C(2^k - 1, 2^k - k - 1)$ se fait exactement de la même façon :

$$v = uG \in \mathbb{F}_2^{2^k-1}.$$

L'encodage est donc une simple multiplication matricielle sur le corps à deux éléments \mathbb{F}_2 .

Le décodage est plus subtil ! Les deux observations suivantes sont au cœur de cette étape. La première est assez directe : un mot du code $v \in \mathbb{F}_2^{2^k-1}$ sans erreur est annihilé par la matrice de contrôle :

$$Hv^t = H(uG)^t = HG^t u^t = (GH^t)^t u^t = 0$$

du fait de l'orthogonalité des lignes de G et H .

La seconde observation est plus difficile. Soit $v \in \mathbb{F}_2^{2^k-1}$ un mot (sans erreur) du code et $v^{(i)} \in \mathbb{F}_2^{2^k-1}$ le mot obtenu de v en additionnant 1 à la i -ième composante de v . Ainsi, $v^{(i)}$ est un mot erroné en position i . Notons que $H(v^{(i)})^t \in \mathbb{F}_2^k$ est indépendant de v ! En effet,

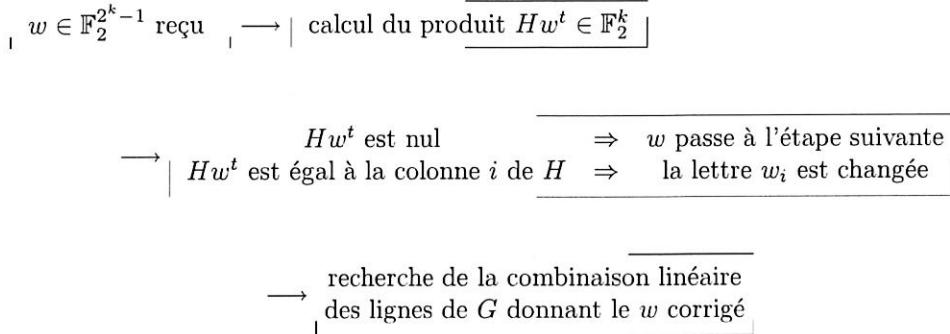
$$v^{(i)} = v + (0, 0, \dots, 0, \underbrace{1}_{\text{position } i}, 0, \dots, 0)$$

et

$$H(v^{(i)})^t = Hv^t + H \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = H \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{position } i,$$

puisque v est un mot du code. Ainsi, $H(v^{(i)})^t$ est la i -ième colonne de H . Puisque toutes les colonnes de H sont distinctes, par définition de H , une erreur sur la lettre i dans le mot reçu $w \in \mathbb{F}_2^{2^k-1}$ revient à obtenir la i -ième colonne de H par le produit Hw^t .

Le décodage fonctionne comme suit :



Quoique ces codes ne corrigeant qu'une erreur, ils sont très économiques pour k suffisamment grand. Par exemple, pour $k = 7$, il suffit d'ajouter sept bits à un message de 120 bits pour être sûr de pouvoir corriger une erreur. C'est précisément le code de Hamming $C(2^k - 1, 2^k - k - 1)$, $k = 7$, qui est utilisé pour le Minitel.

6.5 Corps finis

Pour présenter le code de Reed-Solomon, nous aurons besoin de connaître quelques propriétés des corps finis. Cette section couvre les éléments requis.

Définition 6.1 Un corps \mathbb{F} est un ensemble muni de deux opérations $+$ et \times et contenant au moins deux éléments notés 0 et $1 \in \mathbb{F}$, tel que les cinq propriétés suivantes soient satisfaites :

(P1) commutativité

$$a + b = b + a \quad \text{et} \quad a \times b = b \times a, \quad \forall a, b \in \mathbb{F}$$

(P2) associativité

$$(a + b) + c = a + (b + c) \quad \text{et} \quad (a \times b) \times c = a \times (b \times c), \quad \forall a, b, c \in \mathbb{F}$$

(P3) distributivité

$$(a + b) \times c = (a \times c) + (b \times c), \quad \forall a, b, c \in \mathbb{F}$$

(P4) neutres additif et multiplicatif

$$a + 0 = a \quad \text{et} \quad a \times 1 = a, \quad \forall a \in \mathbb{F}$$

(P5) existence des inverses additif et multiplicatif

$$\begin{aligned} & \forall a \in \mathbb{F}, \exists a' \in \mathbb{F} \text{ tel que } a + a' = 0, \\ & \forall a \in \mathbb{F} \setminus \{0\}, \exists a' \in \mathbb{F} \text{ tel que } a \times a' = 1. \end{aligned}$$

Définition 6.2 Un corps \mathbb{F} est dit fini si le nombre d'éléments dans \mathbb{F} est fini.

Exemple 6.3 Les trois corps les plus familiers sont \mathbb{Q} , \mathbb{R} et \mathbb{C} , c'est-à-dire les nombres rationnels, réels et complexes. Ils ne sont pas finis. La liste des propriétés ci-dessus est probablement familière au lecteur. Le but de donner la définition de corps est donc d'axiomatiser les propriétés de ces trois ensembles. L'avantage est de pouvoir étendre à des corps moins intuitifs les techniques de calcul développées pour \mathbb{Q} , \mathbb{R} et \mathbb{C} et qui ne reposent que sur (P1), (P2), (P3), (P4) et (P5).

Exemple 6.4 \mathbb{F}_2 muni des opérations $+$ et \times données à la section 6.2 est un corps. Les calculs faits durant l'étude des codes de Hamming vous ont sans doute convaincu que $(\mathbb{F}_2, +, \times)$ est bien un corps. Une vérification systématique est proposée à l'exercice 4 en fin de chapitre.

Exemple 6.5 \mathbb{F}_2 n'est que le premier d'une famille de corps finis. Soit p un nombre premier. On dit que deux nombres a et b sont congrus modulo p si p divise $a - b$. La congruence est une relation d'équivalence sur les entiers. On a exactement p classes d'équivalence représentées par $\bar{0}, \bar{1}, \dots, \bar{p-1}$. Par exemple, pour $p = 3$, les entiers \mathbb{Z} sont partitionnés en trois sous-ensembles :

$$\begin{aligned}\bar{0} &= \{\dots, -6, -3, 0, 3, 6, \dots\}, \\ \bar{1} &= \{\dots, -5, -2, 1, 4, 7, \dots\}, \\ \bar{2} &= \{\dots, -4, -1, 2, 5, 8, \dots\}.\end{aligned}$$

L'ensemble $\mathbb{Z}_p = \{\bar{0}, \bar{1}, \bar{2}, \dots, \bar{p-1}\}$ est l'ensemble de ces classes d'équivalence. On définit, entre ces classes, les opérations $+$ et \times comme l'addition et la multiplication modulo p . Pour faire l'addition modulo p des classes \bar{a} et \bar{b} , on choisit un élément de la classe \bar{a} et un autre de la classe \bar{b} . Le résultat de $\bar{a} + \bar{b}$ est $\bar{a} + \bar{b}$, c'est-à-dire la classe de \mathbb{Z}_p à laquelle appartient la somme des deux éléments choisis. (Exercice : pourquoi cette classe ne dépend-elle pas du choix des éléments, mais seulement des classes \bar{a} et \bar{b} ? Cette définition coïncide-t-elle avec celle donnée précédemment en 6.2 pour le cas \mathbb{F}_2 ?). La multiplication entre classes est définie de la même façon. Il est usuel d'omettre le « $\bar{}$ » qui dénote la classe d'équivalence. L'exercice 24 démontre que $(\mathbb{Z}_p, +, \times)$ est un corps.

Exemple 6.6 L'ensemble des entiers \mathbb{Z} n'est pas un corps, car l'élément 2, par exemple, n'y a pas d'inverse multiplicatif.

Exemple 6.7 Soit \mathbb{F} un corps. Dénotons par $\tilde{\mathbb{F}}$ l'ensemble de tous les quotients de polynômes en une variable x à coefficients dans \mathbb{F} ; ainsi, tous les éléments de $\tilde{\mathbb{F}}$ sont de la forme $\frac{p(x)}{q(x)}$, $p(x)$ et $q(x)$ étant des polynômes (de degré fini par définition) à coefficients dans \mathbb{F} et q étant différent du polynôme identiquement nul. Si nous munissons $\tilde{\mathbb{F}}$ de l'addition et de la multiplication usuelles pour les fonctions, alors $(\tilde{\mathbb{F}}, +, \times)$ est un corps. Les quotients de polynômes $0/1 = 0$ (c'est-à-dire le quotient tel que $p(x) = 0$ et $q(x) = 1$) et $1/1$ (c'est-à-dire tel que $p(x) = q(x) = 1$) sont les neutres additif et multiplicatif. On peut aisément vérifier les propriétés (P1) à (P5).

L'ensemble \mathbb{Z}_p ci-dessus mérite d'être étudié de plus près. Les tables d'addition et de multiplication dans \mathbb{Z}_3 sont

$+$	0	1	2		\times	0	1	2	
0	0	1	2		0	0	0	0	
1	1	2	0		1	0	1	2	
2	2	0	1		2	0	2	1	

(6.4)

et celles de \mathbb{Z}_5 sont

$+$	0	1	2	3	4	\times	0	1	2	3	4	
0	0	1	2	3	4	0	0	0	0	0	0	
1	1	2	3	4	0	1	0	1	2	3	4	
2	2	3	4	0	1	2	0	2	4	1	3	
3	3	4	0	1	2	3	0	3	1	4	2	
4	4	0	1	2	3	4	0	4	3	2	1	

(6.5)

(Exercice : vérifier que ces tables représentent bien l'addition et la multiplication modulo 3 et 5 respectivement.) L'exemple introduisant le corps \mathbb{Z}_p stipule que p doit être un nombre premier. Qu'arrive-t-il si p ne l'est pas ? Voici les tables d'addition et de multiplication modulo 6 définies sur l'ensemble $\{0, 1, 2, 3, 4, 5\}$:

$+$	0	1	2	3	4	5	\times	0	1	2	3	4	5
0	0	1	2	3	4	5	0	0	0	0	0	0	0
1	1	2	3	4	5	0	1	0	1	2	3	4	5
2	2	3	4	5	0	1	2	0	2	4	0	2	4
3	3	4	5	0	1	2	3	0	3	0	3	0	3
4	4	5	0	1	2	3	4	0	4	2	0	4	2
5	5	0	1	2	3	4	5	0	5	4	3	2	1

(6.6)

Comment prouver que $\{0, 1, 2, 3, 4, 5\}$ muni de ces tables ne forme pas un corps ? À l'aide des zéros en caractère gras dans la table de multiplication ci-dessus ! En voici la preuve.

Nous savons que $0 \times a = 0$ dans \mathbb{Q} et dans \mathbb{R} . Est-ce vrai pour tout élément non nul a dans un corps quelconque \mathbb{F} ? Oui ! La preuve qui suit est élémentaire. (En la lisant, noter que chaque étape découle directement d'une des propriétés du corps \mathbb{F} .) Soit a un élément quelconque de \mathbb{F} . Alors

$$\begin{aligned} 0 \times a &= (0 + 0) \times a && \text{(P4)} \\ &= 0 \times a + 0 \times a && \text{(P3).} \end{aligned}$$

Par (P5), tout élément de \mathbb{F} possède un inverse additif. Soit b l'inverse additif de $(0 \times a)$. Ajoutons cet élément aux deux membres de l'équation ci-dessus :

$$(0 \times a) + b = (0 \times a + 0 \times a) + b.$$

Le membre de gauche est nul (par définition de b) alors que celui de droite peut être réécrit

$$\begin{aligned} 0 &= 0 \times a + ((0 \times a) + b) && (\text{P2}) \\ &= 0 \times a + 0 \\ &= 0 \times a && (\text{P4}) \end{aligned}$$

à cause du choix de b . Ainsi, $0 \times a$ est nul quel que soit $a \in \mathbb{F}$. Revenons maintenant à la table de multiplication d'un corps \mathbb{F} . Soient a et $b \in \mathbb{F}$ deux éléments non nuls de \mathbb{F} tels que

$$a \times b = 0.$$

En multipliant les deux membres de cette équation par l'inverse multiplicatif b' de b qui existe de par (P5), on a

$$a \times (b \times b') = 0 \times b',$$

et, par la propriété qui vient d'être démontrée,

$$a \times 1 = 0$$

ou, par (P4),

$$a = 0,$$

ce qui est une contradiction, car a est non nul. *Donc, dans un corps \mathbb{F} , le produit d'éléments non nuls est non nul.* Ainsi, $(\mathbb{Z}_6, +, \times)$ n'est pas un corps... à cause des zéros en caractère gras.

Si p n'est pas un nombre premier, il existe q_1 et q_2 non nuls et différents de 1 tels que $p = q_1 q_2$. Dans \mathbb{Z}_p , on aura $q_1 \times q_2 = p = 0 \pmod{p}$. *Ainsi, si p n'est pas premier, \mathbb{Z}_p muni de l'addition et de la multiplication modulo p ne peut être un corps.* Nous allons utiliser cette observation (ainsi démontrée) pour introduire un résultat que nous ne prouverons pas.

On dénote par $\mathbb{F}[x]$ l'ensemble des polynômes en une variable x à coefficients dans \mathbb{F} . Cet ensemble peut être muni de l'addition et de la multiplication polynomiales usuelles. Attention : $\mathbb{F}[x]$ n'est pas un corps. Par exemple, l'élément non nul $(x + 1)$ n'a pas d'inverse multiplicatif.

Exemple 6.8 $\mathbb{F}_2[x]$ est l'ensemble de tous les polynômes à coefficients dans \mathbb{F}_2 . Voici un exemple de multiplication dans $\mathbb{F}_2[x]$:

$$(x + 1) \times (x + 1) = x^2 + x + x + 1 = x^2 + (1 + 1)x + 1 = x^2 + 1 \in \mathbb{F}_2[x].$$

De la même façon que nous avons appris à calculer « modulo p », il est possible de calculer « modulo un polynôme $p(x)$ ». Soit $p(x) \in \mathbb{F}[x]$ un polynôme de degré $n \geq 1$

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

tel que $a_i \in \mathbb{F}$, $0 \leq i \leq n$ et $a_n \neq 0$. On choisira, par la suite, un polynôme où $a_n = 1$. Les opérations $+$ et \times modulo $p(x)$ consistent à faire les opérations $+$ et \times polynomiales usuelles sur $\mathbb{F}[x]$, puis à utiliser de façon répétitive des multiples polynomiaux de $p(x)$ pour réduire le résultat à un polynôme de degré inférieur à n . Cette phrase est compliquée, mais deux exemples la clarifieront.

Exemple 6.9 Soit $p(x) = x^2 + 1 \in \mathbb{Q}[x]$ et soient $(x+1)$ et $(x^2 + 2x)$, deux polynômes appartenant également à $\mathbb{Q}[x]$, que nous désirons multiplier modulo $p(x)$. Les égalités qui suivent sont donc des égalités entre des polynômes différant par un multiple du polynôme $p(x)$. Ce ne sont pas des égalités strictes comme l'indique le « $(\text{mod } p(x))$ » à la dernière ligne. Voici les étapes :

$$\begin{aligned} (x+1) \times (x^2 + 2x) &= x^3 + 2x^2 + x^2 + 2x \\ &= x^3 + 3x^2 + 2x - x(x^2 + 1) \\ &= x^3 - x^3 + 3x^2 + 2x - x \\ &= 3x^2 + x \\ &= 3x^2 + x - 3(x^2 + 1) \\ &= 3x^2 - 3x^2 + x - 3 \\ &= x - 3 \quad (\text{mod } p(x)). \end{aligned}$$

Il est aisément vérifiable que le polynôme $(x-3)$ est également le reste de la division de $(x+1) \times (x^2 + 2x)$ par $p(x)$. Ceci n'est pas une coïncidence ! C'est une propriété générale qui donne une autre méthode pour calculer $q(x) \pmod{p(x)}$. Voir l'exercice 14.

Exemple 6.10 Soit $p(x) = x^2 + x + 1 \in \mathbb{F}_2[x]$. Le carré du polynôme $(x^2 + 1)$ modulo $p(x)$ est

$$\begin{aligned} (x^2 + 1) \times (x^2 + 1) &= x^4 + 1 = x^4 + 1 - x^2(x^2 + x + 1) = x^3 + x^2 + 1 \\ &= x^3 + x^2 + 1 - x(x^2 + x + 1) = x + 1 \quad (\text{mod } p(x)). \end{aligned}$$

On peut engendrer tous les corps finis à partir des ensembles de polynômes $\mathbb{F}[x]$ en copiant la construction des \mathbb{Z}_p munis de $+$ et \times modulo p où, rappelons-le, p doit être premier. Pour $\mathbb{F}[x]$, les opérations $+$ et \times se feront modulo un polynôme $p(x)$. N'importe quel polynôme ? Non ! De même que p doit être un nombre premier dans le cas de \mathbb{Z}_p , de même le polynôme $p(x)$ devra satisfaire à une condition particulière : être *irréductible*. Un polynôme non nul $p(x) \in \mathbb{F}[x]$ est irréductible si, pour tous $q_1(x)$ et $q_2(x) \in \mathbb{F}[x]$ tels que

$$p(x) = q_1(x)q_2(x),$$

alors $q_1(x)$ ou $q_2(x)$ est un polynôme constant. En d'autres mots, $p(x)$ n'est pas le produit de deux polynômes $\in \mathbb{F}[x]$ de degré inférieur à celui de $p(x)$.

Exemple 6.11 Le polynôme $x^2 + x - 1$ peut être factorisé sur \mathbb{R} . En effet, soient

$$x_1 = \frac{1}{2}(\sqrt{5} - 1) \quad \text{et} \quad x_2 = -\frac{1}{2}(\sqrt{5} + 1),$$

les racines de ce polynôme. Ces deux nombres appartiennent à \mathbb{R} et

$$x^2 + x - 1 = (x - x_1)(x - x_2).$$

Ainsi, $x^2 + x - 1 \in \mathbb{R}[x]$ n'est pas irréductible sur \mathbb{R} . Ce même polynôme est cependant irréductible sur $\mathbb{Q}[x]$, car ni x_1 ni x_2 n'appartiennent à \mathbb{Q} et $x^2 + x - 1$ ne peut être factorisé sur \mathbb{Q} .

Exemple 6.12 Le polynôme $x^2 + 1$ est irréductible sur \mathbb{R} , mais sur \mathbb{F}_2 , il peut être écrit comme $x^2 + 1 = (x + 1) \times (x + 1)$, et il n'est donc pas irréductible sur \mathbb{F}_2 .

On dénotera par $\mathbb{F}[x]/(p(x))$ l'ensemble des polynômes à coefficients dans \mathbb{F} muni des opérations $+$ et \times modulo $p(x)$. Voici le résultat central dont nous aurons besoin.

Proposition 6.13 (i) Soit $p(x)$ un polynôme de degré n . Le quotient $\mathbb{F}[x]/p(x)$ peut être identifié à $\{q(x) \in \mathbb{F}[x] \mid \text{degré } q < n\}$ muni de l'addition et de la multiplication modulo $p(x)$.

(ii) $\mathbb{F}[x]/(p(x))$ est un corps si et seulement si $p(x)$ est irréductible sur \mathbb{F} .

Nous ne démontrerons pas ce résultat, mais nous l'utiliserons pour donner un exemple de construction explicite d'un corps fini qui ne soit pas un des corps \mathbb{Z}_p avec p premier.

Exemple 6.14 Construction de \mathbb{F}_9 , le corps à neuf éléments. Soit \mathbb{Z}_3 le corps à trois éléments dont les tables ont été données ci-dessus. Soit $\mathbb{Z}_3[x]$ l'ensemble des polynômes à coefficients dans \mathbb{Z}_3 et soit $p(x) = x^2 + x + 2$.

Convainquons-nous d'abord que $p(x)$ est irréductible. S'il ne l'est pas, alors il existe deux polynômes non constants q_1 et q_2 dont le produit est p . Ces deux polynômes doivent être de degré 1. Ainsi,

$$p(x) = (x + a)(bx + c) \tag{6.7}$$

pour certains $a, b, c \in \mathbb{Z}_3$. Si tel est le cas, $p(x)$ s'annulera pour l'inverse additif du nombre $a \in \mathbb{Z}_3$. Mais

$$\begin{aligned} p(0) &= 0^2 + 0 + 2 = 2, \\ p(1) &= 1^2 + 1 + 2 = 1, \\ p(2) &= 2^2 + 2 + 2 = 1 + 2 + 2 = 2; \end{aligned}$$

donc $p(x)$ ne s'annule pour aucun des trois éléments de \mathbb{Z}_3 . (Attention : les calculs sont faits dans \mathbb{Z}_3 !) Ainsi, $p(x)$ ne peut être mis sous la forme (6.7), et $p(x)$ est donc irréductible.

Commençons par trouver le nombre d'éléments du corps $\mathbb{Z}_3[x]/(p(x))$. Puisque tous les éléments de ce corps sont des polynômes de degré inférieur au degré de $p(x)$, ils sont tous de la forme $a_1x + a_0$. Puisque $a_0, a_1 \in \mathbb{Z}_3$ et que chacun peut prendre trois valeurs, il y aura donc $3^2 = 9$ éléments distincts dans $\mathbb{Z}_3[x]/(p(x))$.

Cherchons maintenant la table de multiplication. Deux exemples montrent comment faire :

$$(x+1)^2 = x^2 + 2x + 1 = (x^2 + 2x + 1) - (x^2 + x + 2) = x - 1 = x + 2$$

$$x(x+2) = x^2 + 2x = x^2 + 2x - (x^2 + x + 2) = x - 2 = x + 1.$$

La table de multiplication complète est

\times	0	1	2	x	$x + 1$	$x + 2$	$2x$	$2x + 1$	$2x + 2$
0	0	0	0	0	0	0	0	0	0
1	0	1	2	x	$x + 1$	$x + 2$	$2x$	$2x + 1$	$2x + 2$
2	0	2	1	$2x$	$2x + 2$	$2x + 1$	x	$x + 2$	$x + 1$
x	0	x	$2x$	$2x + 1$	1	$x + 1$	$x + 2$	$2x + 2$	2
$x + 1$	0	$x + 1$	$2x + 2$	1	$x + 2$	$2x$	2	x	$2x + 1$
$x + 2$	0	$x + 2$	$2x + 1$	$x + 1$	$2x$	2	$2x + 2$	1	x
$2x$	0	$2x$	x	$x + 2$	2	$2x + 2$	$2x + 1$	$x + 1$	1
$2x + 1$	0	$2x + 1$	$x + 2$	$2x + 2$	x	1	$x + 1$	2	$2x$
$2x + 2$	0	$2x + 2$	$x + 1$	2	$2x + 1$	x	1	$2x$	$x + 2$

(6.8)

Mais cette méthode est fastidieuse. Y a-t-il une manière de simplifier les calculs ? Énumérons plutôt les puissances du polynôme $q(x) = x$. À nouveau, toutes les égalités sont modulo $p(x)$. On a

$$\begin{aligned} q &= x, \\ q^2 &= x^2 = x^2 - (x^2 + x + 2) = -x - 2 = 2x + 1, \\ q^3 &= q \times q^2 = 2x^2 + x = 2x^2 + x - 2(x^2 + x + 2) = 2x + 2, \\ q^4 &= q \times q^3 = 2x^2 + 2x = 2x^2 + 2x - 2(x^2 + x + 2) = 2, \\ q^5 &= q \times q^4 = 2x, \\ q^6 &= q \times q^5 = 2x^2 = 2x^2 - 2(x^2 + x + 2) = x + 2, \\ q^7 &= q \times q^6 = x^2 + 2x = x^2 + 2x - (x^2 + x + 2) = x + 1, \\ q^8 &= q \times q^7 = x^2 + x = x^2 + x - (x^2 + x + 2) = 1. \end{aligned}$$

En prenant les puissances du polynôme $q(x) = x$, on obtient donc les huit polynômes non nuls de $\mathbb{Z}_3[x]/(p(x))$. La multiplication des éléments $\{0, q, q^2, q^3, q^4, q^5, q^6, q^7, q^8 = 1\}$ est aisée puisque $q^i \times q^j = q^{i+j}$ où $k = i + j \pmod{8}$ car $q^8 = 1$. Ceci nous donne une manière simple de calculer la table de multiplication. On transforme tout polynôme en une puissance de q . Alors, la multiplication de deux éléments se réduit à l'addition des exposants modulo 8. Nous pouvons refaire aisément les deux exemples ci-dessus

$$(x+1)^2 = q^7 \times q^7 = q^{14} = q^6 = x+2,$$

$$x(x+2) = q \times q^6 = q^7 = x+1.$$

Cette deuxième méthode nous permet également de vérifier nos calculs. Nous redonnons donc la table de multiplication ci-dessus en renommant les polynômes par leur puissance de q .

\times	0	1	q^4	q^1	q^7	q^6	q^5	q^2	q^3
0	0	0	0	0	0	0	0	0	0
1	0	1	q^4	q	q^7	q^6	q^5	q^2	q^3
q^4	0	q^4	1	q^5	q^3	q^2	q	q^6	q^7
q^1	0	q	q^5	q^2	1	q^7	q^6	q^3	q^4
q^7	0	q^7	q^3	1	q^6	q^5	q^4	q	q^2
q^6	0	q^6	q^2	q^7	q^5	q^4	q^3	1	q
q^5	0	q^5	q	q^6	q^4	q^3	q^2	q^7	1
q^2	0	q^2	q^6	q^3	q	1	q^7	q^4	q^5
q^3	0	q^3	q^7	q^4	q^2	q	1	q^5	q^6

(6.9)

Avec ces nouveaux noms, il est peut-être naturel de réordonner les lignes et les colonnes de façon à ce que les exposants de q croissent. Revoici donc une troisième fois la table de multiplication de \mathbb{F}_9 !

\times	0	q^1	q^2	q^3	q^4	q^5	q^6	q^7	1
0	0	0	0	0	0	0	0	0	0
q^1	0	q^2	q^3	q^4	q^5	q^6	q^7	1	q
q^2	0	q^3	q^4	q^5	q^6	q^7	1	q	q^2
q^3	0	q^4	q^5	q^6	q^7	1	q	q^2	q^3
q^4	0	q^5	q^6	q^7	1	q	q^2	q^3	q^4
q^5	0	q^6	q^7	1	q	q^2	q^3	q^4	q^5
q^6	0	q^7	1	q	q^2	q^3	q^4	q^5	q^6
q^7	0	1	q	q^2	q^3	q^4	q^5	q^6	q^7
1	0	q	q^2	q^3	q^4	q^5	q^6	q^7	1

(6.10)

La table d'addition peut également être obtenue facilement. Voici deux exemples de calcul :

$$q^2 + q^4 = (2x+1) + (2) = 2x + (2+1) = 2x = q^5,$$

$$q^3 + q^6 = (2x+2) + (x+2) = (2+1)x + (2+2) = 1 = q^8.$$

Voici la table d'addition complète de \mathbb{F}_9 . (Exercice : vérifier quelques éléments de cette table d'addition.)

+	0	q^1	q^2	q^3	q^4	q^5	q^6	q^7	1
0	0	q^1	q^2	q^3	q^4	q^5	q^6	q^7	1
q^1	q^1	q^5	1	q^4	q^6	0	q^3	q^2	q^7
q^2	q^2	1	q^6	q^1	q^5	q^7	0	q^4	q^3
q^3	q^3	q^4	q^1	q^7	q^2	q^6	1	0	q^5
q^4	q^4	q^6	q^5	q^2	1	q^3	q^7	q^1	0
q^5	q^5	0	q^7	q^6	q^3	q^1	q^4	1	q^2
q^6	q^6	q^3	0	1	q^7	q^4	q^2	q^5	q^1
q^7	q^7	q^2	q^4	0	q^1	1	q^5	q^3	q^6
1	1	q^7	q^3	q^5	0	q^2	q^1	q^6	q^4

(6.11)

Définition 6.15 Un élément non nul d'un corps tel que tous les autres éléments non nuls peuvent être obtenus de celui-ci par exponentiation est appelé élément primitif ou racine primitive.

Tous les éléments ne sont pas primitifs. Par exemple, dans \mathbb{F}_9 , l'élément q^4 n'est pas primitif ; les seuls éléments distincts qu'il engendre sont q^4 et $q^4 \times q^4 = q^8 = 1$. À l'exercice 13, vous trouverez tous les éléments (non nuls) primitifs de \mathbb{F}_9 . Dans l'exemple ci-dessus, le polynôme $q(x) = x$ est primitif puisqu'il nous a permis de construire les huit polynômes non nuls sous la forme q^i , $i = 1, \dots, 8$. Mais $q(x) = x$ n'est pas primitif pour tous les choix du corps de base \mathbb{F} et du polynôme irréductible $p(x)$. Nous en donnons deux exemples, le premier à l'exercice 17 ci-dessous et le second à l'exercice 6 du chapitre 8. Nous résumons cette analyse dans le théorème suivant.

Théorème 6.16 Tout corps fini \mathbb{F}_{p^r} possède une racine primitive, c'est-à-dire qu'il existe un élément non nul α dont l'ensemble des puissances coïncide avec l'ensemble des éléments non nuls de \mathbb{F}_{p^r} :

$$\mathbb{F}_{p^r} \setminus \{0\} = \{\alpha, \alpha^2, \dots, \alpha^{p^r-1} = 1\}.$$

Il est usuel de choisir la lettre α pour une racine primitive ; dans cette section, nous avons utilisé la lettre q , décrivant le polynôme $q(x) = x$, mais nous utiliserons α à la prochaine section. (Les lecteurs qui connaissent la structure de groupe noteront qu'un élément primitif est en fait un générateur du groupe multiplicatif des éléments non nuls du corps. Ceci indique que ces éléments forment un groupe cyclique. Nous n'utilisons cependant pas ce fait dans le présent chapitre.)

Avant de terminer notre introduction aux corps finis, nous énoncerons deux théorèmes sans les prouver.

Théorème 6.17 Le nombre d'éléments dans un corps fini est une puissance d'un nombre premier.

Théorème 6.18 *Si deux corps finis possèdent le même nombre d'éléments, alors ils sont isomorphes, c'est-à-dire qu'il existe un réordonnement des éléments du premier corps tel que les tables d'addition et de multiplication des deux corps coïncident. Un tel réordonnement s'appelle un isomorphisme entre les deux corps.*

6.6 Les codes de Reed et Solomon

Les codes de Reed et Solomon sont plus complexes que les codes de Hamming. Nous allons tout d'abord décrire l'encodage et le décodage. Puis nous prouverons trois propriétés qui caractérisent ces codes.

Soient \mathbb{F}_{2^m} le corps à 2^m éléments et α une racine primitive. Les $2^m - 1$ éléments non nuls de \mathbb{F}_{2^m} sont donc de la forme

$$\{\alpha, \alpha^2, \dots, \alpha^{2^m-1} = 1\},$$

et alors, tous ces éléments non nuls satisfont à $x^{2^m-1} = 1$.

Les mots à encoder seront des mots de k lettres (chacune étant un élément de \mathbb{F}_{2^m}) où $k < 2^m - 2$. (Nous expliquerons sous peu comment cet entier k est choisi.) Ainsi, ce seront des éléments $(u_0, u_1, u_2, \dots, u_{k-1}) \in \mathbb{F}_{2^m}^k$. À chacun de ces mots nous ferons correspondre le polynôme

$$p(x) = u_0 + u_1x + u_2x^2 + \dots + u_{k-1}x^{k-1} \in \mathbb{F}_{2^m}[x].$$

Ces mots seront encodés dans un vecteur $v = (v_0, v_1, v_2, \dots, v_{2^m-2}) \in \mathbb{F}_{2^m}^{2^m-1}$ dont les composantes seront données par

$$v_i = p(\alpha^i), \quad i = 0, 1, 2, \dots, 2^m - 2$$

où α est la racine primitive choisie au départ. Ainsi, l'*encodage* consiste à calculer

$$\begin{aligned} v_0 &= p(1) = u_0 + u_1 + u_2 + \dots + u_{k-1}, \\ v_1 &= p(\alpha) = u_0 + u_1\alpha + u_2\alpha^2 + \dots + u_{k-1}\alpha^{k-1}, \\ v_2 &= p(\alpha^2) = u_0 + u_1\alpha^2 + u_2\alpha^4 + \dots + u_{k-1}\alpha^{2(k-1)}, \\ &\vdots = \vdots = \vdots \\ v_{2^m-2} &= p(\alpha^{2^m-2}) = u_0 + u_1\alpha^{2^m-2} + u_2\alpha^{2(2^m-2)} + \dots + u_{k-1}\alpha^{(k-1)(2^m-2)}. \end{aligned} \tag{6.12}$$

Le code de Reed-Solomon $C(2^m - 1, k)$ est l'ensemble des vecteurs $v \in \mathbb{F}_{2^m}^{2^m-1}$ ainsi obtenu. Une des conditions de base de tout encodage est que des mots différents aient des formes encodées distinctes. C'est ce qu'assure la propriété suivante du code de Reed-Solomon.

Propriété 6.19 *L'*encodage* $u \mapsto v$ tel que $u \in \mathbb{F}_{2^m}^k$ et $v \in \mathbb{F}_{2^m}^{2^m-1}$, est une application linéaire dont le noyau est nul, c'est-à-dire égal à $\{0\} \subset \mathbb{F}_{2^m}^k$.*

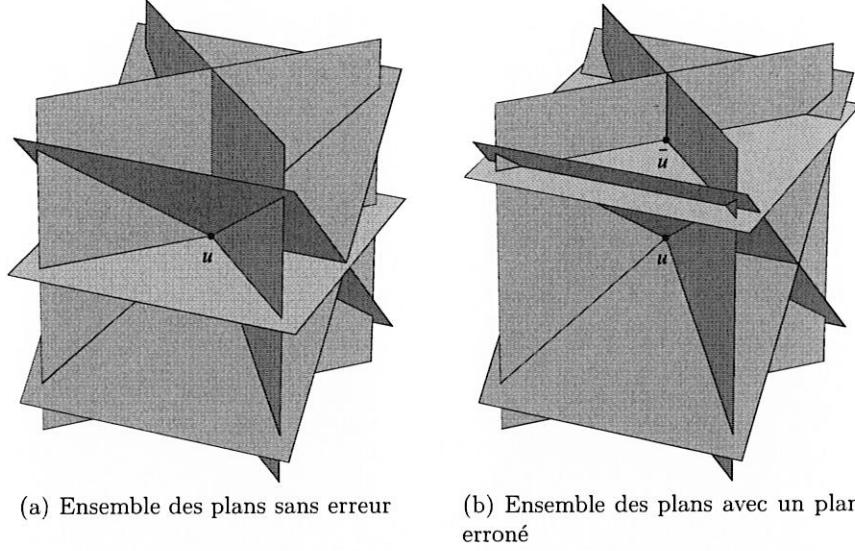
(Les preuves des propriétés 6.19 et 6.20 sont données à la fin de la section.)

La transmission peut introduire des erreurs dans le message encodé v . Ainsi, le message reçu $w \in \mathbb{F}_{2^m}^{2^m-1}$ pourrait différer de v par une composante ou même plus. Le décodage consiste à remplacer, dans le système (6.12), les v_i par les composantes w_i de w et à extraire de ce nouveau système linéaire les composantes u_j du message original et ce, malgré les erreurs possibles dans w . Pour comprendre comment ceci peut être fait, nous décrivons d'abord le système (6.12) géométriquement. Chacune des équations de (6.12) représente un plan dans l'espace \mathbb{F}_2^k paramétrisé par les coordonnées $(u_0, u_1, \dots, u_{k-1})$. Il y a donc 2^m-1 plans, plus que k , le nombre d'inconnues u_j . Utilisons notre intuition de \mathbb{R}^3 pour dessiner une représentation géométrique de la situation. La figure 6.4a présente cinq plans (plutôt que 2^m-1) dans \mathbb{R}^3 (plutôt que dans \mathbb{F}_2^k). S'il n'y a aucune erreur lors de la transmission (et alors, chacune des composantes w_i coïncide avec la composante correspondante v_i), alors les plans s'intersectent tous en un seul point u qui correspond au message original. De plus, chaque choix de trois plans parmi les cinq détermine uniquement la solution u . En d'autres mots, deux des cinq plans sont redondants et, dans cette transmission sans erreur, il y a plusieurs façons de reconstruire le message original u . Supposons maintenant qu'une des composantes de w soit erronée. L'équation qui la contient sera fausse, et le plan correspondant sera déplacé par rapport au plan original. C'est ce que représente la figure 6.4b où le plan horizontal a été déplacé vers le haut. Même si les quatre plans justes (sans erreur) s'intersectent encore en u , un choix de trois plans qui inclut le plan erroné donne une détermination \bar{u} erronée. Dans \mathbb{R}^3 , trois plans sont nécessaires pour déterminer u (justement ou erronément). Dans le système (6.12), il faut k plans (= équations) pour obtenir une valeur de u . Nous pouvons donc penser à un choix de k plans « votant » pour la valeur u où ils s'intersectent. Si quelques w_i sont faux, nous pouvons nous demander sous quelles conditions la valeur correcte de u obtiendra le plus grand nombre de votes. C'est à cette question que nous allons répondre maintenant. (Exercice : vérifier que, pour l'exemple de la figure 6.4b, la réponse u reçoit quatre votes alors que la réponse erronée \bar{u} n'en reçoit qu'un.)

Supposons qu'une fois le message transmis, nous recevions les 2^m-1 lettres de $w = (w_0, w_1, w_2, \dots, w_{2^m-2}) \in \mathbb{F}_{2^m}^{2^m-1}$. Si toutes ces lettres sont exactes, on peut retrouver le message original u en choisissant dans (6.12) n'importe quel sous-ensemble de k lignes (= plans) et en résolvant le système linéaire correspondant. Supposons qu'on choisisse les lignes i_0, i_1, \dots, i_{k-1} , tels que $0 \leq i_0 < i_1 < \dots < i_{k-1} \leq 2^m-2$, et que α_j dénote α^{i_j} . Alors, le système linéaire se lit

$$\begin{pmatrix} w_{i_0} \\ w_{i_1} \\ w_{i_2} \\ \vdots \\ w_{i_{k-1}} \end{pmatrix} = \begin{pmatrix} 1 & \alpha_0 & \alpha_0^2 & \alpha_0^3 & \dots & \alpha_0^{k-1} \\ 1 & \alpha_1 & \alpha_1^2 & \alpha_1^3 & \dots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \alpha_2^3 & \dots & \alpha_2^{k-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_{k-1} & \alpha_{k-1}^2 & \alpha_{k-1}^3 & \dots & \alpha_{k-1}^{k-1} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{k-1} \end{pmatrix}, \quad (6.13)$$

et on peut obtenir le message original u en inversant la matrice $\{\alpha_i^j\}_{0 \leq i,j \leq k-1}$, pour autant qu'elle soit inversible.

**Fig. 6.4.** Les plans du système (6.12)

Propriété 6.20 Pour tout choix $0 \leq i_0 < i_1 < i_2 < \dots < i_{k-1} \leq 2^m - 2$, la matrice $\{\alpha_i^j\}$ ci-dessus est inversible.

Ainsi, lorsque le message reçu ne contient aucune erreur, il y a autant de façons de déterminer le message original que de choix de k équations parmi les $2^m - 1$ équations du système (6.12), c'est-à-dire

$$\binom{2^m - 1}{k} = \frac{(2^m - 1)!}{k!(2^m - 1 - k)!}.$$

Supposons maintenant que s composantes parmi les $2^m - 1$ de w soient erronées. Alors seules $(2^m - s - 1)$ équations de (6.12) sont justes, et seules $\binom{2^m - s - 1}{k}$ déterminations de u parmi les $\binom{2^m - 1}{k}$ possibilités seront justes. Les autres seront erronées ; il y aura donc plusieurs déterminations de u , une seule étant la bonne. Soit \bar{u} une des valeurs fautives qu'on obtient en choisissant certaines des équations fausses de (6.12). Combien de fois peut-on obtenir \bar{u} en changeant les équations retenues ? La solution \bar{u} est l'intersection des k plans que représentent les k équations de (6.12) choisies. Au maximum $s + k - 1$ plans s'intersectent en \bar{u} , car s'il y en avait un seul de plus, il y aurait parmi ceux-ci k plans décrits par des équations justes, et alors, $\bar{u} = u$. Il y aura donc au maximum $\binom{s+k-1}{k}$ déterminations menant à \bar{u} . La valeur juste u obtiendra le plus de votes (c'est-à-dire le plus de déterminations) si

$$\binom{2^m - s - 1}{k} > \binom{s + k - 1}{k}$$

ou, de façon équivalente si

$$2^m - s - 1 > s + k - 1.$$

On en déduit que

$$2^m - k > 2s.$$

Puisque le nombre d'erreurs s est un entier, cette inégalité peut également être écrite

$$2^m - k - 1 \geq 2s.$$

En d'autres termes, tant que le nombre d'erreurs s est plus petit ou égal à $\frac{1}{2}(2^m - k - 1)$, la valeur juste u obtiendra le plus grand nombre de déterminations, et nous venons de prouver la dernière propriété.

Propriété 6.21 *Le code de Reed-Solomon peut corriger $\lceil \frac{1}{2}(2^m - k - 1) \rceil$ erreurs où la notation $[x]$ signifie la partie entière de x .*

Le *décodage* de w consiste donc à choisir, parmi toutes les déterminations de u à l'aide de (6.12), celle qui obtient le plus de votes.

Nous terminons cette section en prouvant les propriétés 6.19 et 6.20.

PREUVE DE LA PROPRIÉTÉ 6.19 Remarquons que chacune des composantes v_j de v , $j = 0, 1, \dots, 2^m - 2$, dépend linéairement des composantes u_i . Ainsi, l'encodage $u \mapsto v$ est une application linéaire de $\mathbb{F}_{2^m}^k$ dans $\mathbb{F}_{2^m}^{2^m-1}$.

Pour montrer que le noyau de cette application est trivial, il suffit de se convaincre que seul le polynôme nul sera envoyé dans $0 \in \mathbb{F}_{2^m}^{2^m-1}$. Si p est un polynôme non nul de degré $k - 1$ ou inférieur, il ne peut pas s'annuler pour plus de $k - 1$ valeurs. Les v_i sont les évaluations du polynôme p pour les puissances α^i , $i = 0, 1, 2, \dots, 2^m - 2$. Puisque α est une racine primitive, toutes les $2^m - 1$ valeurs α^i sont distinctes. Et puisque p n'a pas plus de $k - 1$ racines distinctes, seules $k - 1$ des $2^m - 1$ valeurs $v_i = p(\alpha^i)$ peuvent être nulles. Ainsi, tout polynôme p non nul est envoyé sur un vecteur v non nul. \square

La propriété 6.20 découle du lemme suivant que nous démontrerons tout d'abord.

Lemme 6.22 (déterminant de Vandermonde) *Soient x_1, x_2, \dots, x_n des éléments quelconques d'un corps \mathbb{F} . Alors*

$$\begin{vmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{vmatrix} = \prod_{1 \leq i < j \leq n} (x_j - x_i).$$

PREUVE Si on soustrait la ligne j de la ligne i , la valeur du déterminant n'est pas changée, et la ligne i devient

$$(0 \quad x_i - x_j \quad x_i^2 - x_j^2 \quad x_i^3 - x_j^3 \quad \dots \quad x_i^{n-1} - x_j^{n-1}).$$

Puisque

$$x_i^k - x_j^k = (x_i - x_j) \sum_{l=0}^{k-1} x_i^l x_j^{k-l-1},$$

tous les éléments de cette nouvelle ligne i possèdent $(x_i - x_j)$ comme facteur. Le déterminant, vu comme polynôme en les variables x_1, x_2, \dots, x_n , possède donc $(x_i - x_j)$ comme facteur pour tout i et j . Le déterminant est donc le produit de

$$\prod_{1 \leq i < j \leq n} (x_j - x_i)$$

et d'un polynôme demeurant à déterminer. Notons que, dans $\prod_{1 \leq i < j \leq n} (x_j - x_i)$, la puissance maximale de x_n est $n-1$, car il y a $(n-1)$ termes tels que $j = n$. Dans le déterminant, la puissance maximale de x_n est également $n-1$, car les termes incluant x_n sont tous dans la même ligne, et c'est x_n^{n-1} qui, dans cette ligne, a la plus grande puissance. Ainsi, le polynôme multipliant $\prod_{1 \leq i < j \leq n} (x_j - x_i)$ ne peut pas dépendre de x_n . On peut répéter cet argument pour tous les autres x_i ; on conclut que le polynôme multipliant $\prod_{1 \leq i < j \leq n} (x_j - x_i)$ est une constante. Le terme $x_1^0 x_2^1 x_3^2 \cdots x_n^{n-1}$ dans le déterminant vient du produit de tous les termes diagonaux et a donc pour coefficient $+1$. Dans le produit $\prod_{1 \leq i < j \leq n} (x_j - x_i)$, ce même terme $x_1^0 x_2^1 x_3^2 \cdots x_n^{n-1}$ est obtenu par multiplication des *premiers* termes de tous les monômes $(x_j - x_i)$ et a également pour coefficient $+1$. (Pourquoi les *premiers* termes? Il y a précisément $n-1$ monômes du produit $\prod_{1 \leq i < j \leq n} (x_j - x_i)$ qui contiennent le terme x_n , et dans tous ces monômes, la variable x_n est le premier terme de $(x_j - x_i)$ puisque $i < j$. Il faut donc choisir les $n-1$ premiers termes de ces monômes. Parmi les monômes restants, il y en a précisément $n-2$ qui contiennent le terme x_{n-1} . À nouveau, dans tous ces monômes, la variable x_{n-1} est le premier terme. En répétant l'argument, on arrive à l'énoncé.) Donc, le déterminant et le polynôme sont égaux. \square

PREUVE DE LA PROPRIÉTÉ 6.20 Le lemme appliqué à la matrice du système (6.13) montre que son déterminant est égal à $\prod_{i < j} (\alpha_j - \alpha_i)$. Rappelons que les α_i sont des puissances distinctes de la racine primitive α et inférieures à $2^m - 1$. Donc, tous ces α_i sont distincts, le déterminant est non nul et la matrice inversible. \square

Voici un exemple concret des divers paramètres k, m et s du code. Nous avons vu au tout début de ce chapitre qu'il est usuel d'utiliser sept ou huit bits pour coder chacun des symboles de typographie (lettres, chiffres, signes de ponctuation, etc.). Si m est fixé à huit, alors chacune des lettres ($\in \mathbb{F}_{2^m}$) pourra représenter précisément une lettre de notre alphabet ou un caractère de ponctuation. Ainsi, la correspondance entre « lettre de l'alphabet » et « lettre dans \mathbb{F}_{2^m} » est biunivoque. Si nous choisissons

$m = 8$, le nombre k de lettres est borné par $2^m - 2 = 254$. Supposons maintenant que le canal de transmission soit assez fiable et qu'il soit pratiquement toujours suffisant de pouvoir corriger deux lettres. Puisque le nombre d'erreurs corrigibles s est égal à $\lfloor \frac{1}{2}(2^m - k - 1) \rfloor$, il faut donc que $(2^m - k - 1)$ soit supérieur ou égal à $2s = 4$. Nous pouvons donc transmettre le texte par blocs de $k = 2^8 - 4 - 1 = 251$ lettres. Notons qu'il pourrait y avoir plus d'un bit erroné au sein de chaque lettre transmise. Le code de Reed–Solomon corrige les lettres (et non les bits individuels).

La technologie du disque compact ne transmet pas des caractères latins, mais bien sûr un signal musical numérisé. Elle utilise malgré tout le code de Reed–Solomon avec les paramètres que nous venons d'étudier, soit $m = 8$ et un maximum de deux erreurs. Notons enfin qu'il existe des algorithmes efficaces de décodage qui évitent la solution de $\binom{2^m-1}{k}$ systèmes linéaires de k équations en k inconnues [2, 8]. Ces algorithmes accélèrent considérablement le décodage.

6.7 Appendice : le produit scalaire et les corps finis

Il est fort probable que votre cours d'algèbre linéaire ait défini le produit scalaire sur un espace vectoriel V sur le corps \mathbb{R} comme une fonction notée (\cdot, \cdot) qui associe à une paire d'éléments de V un nombre réel et telle que

- (i) $(x, y) = (y, x)$, pour tout $x, y \in V$;
- (ii) $(x + y, z) = (x, z) + (y, z)$ pour tout $x, y, z \in V$;
- (iii) $(cx, y) = c(x, y)$ pour tout $x, y \in V$ et $c \in \mathbb{R}$;
- (iv) $(x, x) \geq 0$ et $(x, x) = 0$ seulement pour $x = 0$.

Si le corps \mathbb{R} des nombres réels est remplacé par un corps fini, cette définition est conservée à l'exception de la dernière exigence qui devient

$$(iv)_{\text{fini}} \text{ si } (x, y) = 0 \text{ pour tout } y \in V, \text{ alors } x = 0.$$

C'est donc avec cette modification que le produit scalaire est utilisé dans le présent chapitre. Notons que la condition (iv) originale n'a pas de sens dans un corps fini, car il n'y a pas de relation d'ordre ($<$) préservée par l'addition. Par exemple, dans \mathbb{F}_2 , nous pourrions proposer que $0 < 1$. Cependant, cette inégalité ne peut être réconciliée avec l'affirmation usuelle dans les réels qui dit que, si $a < b$, alors $a + c < b + c$ pour tout nombre c . En effet, si le nombre $1 \in \mathbb{F}_2$ est ajouté aux deux membres de $0 < 1$, nous obtenons $0 + 1 < 1 + 1$, c'est-à-dire $1 < 0$, ce qui contredit clairement $0 < 1$!

La définition de complément orthogonal demeure la même pour le produit scalaire avec $(iv)_{\text{fini}}$. Nous la rappelons.

Définition 6.23 Si $W \subset V$ est un sous-ensemble de V , alors le complément orthogonal W^\perp est défini par $W^\perp = \{v \in V | (v, w) = 0 \text{ pour tout } w \in W\}$.

C'est un sous-espace vectoriel de V . La modification $(iv) \rightarrow (iv)_{\text{fini}}$ a une conséquence inattendue. Rappelons que, si $W \subset \mathbb{R}^n$ est un sous-espace vectoriel, alors lui et son complément n'ont que l'origine en commun : $W \cap W^\perp = \{0\}$. Dans un espace vectoriel sur un corps fini, ceci n'est plus toujours le cas ! Par exemple, considérons le sous-espace W engendré par le vecteur

$$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \in \mathbb{F}_2^3.$$

Les éléments $w = (w_1, w_2, w_3)^t \in \mathbb{F}_2^3$ du complément orthogonal W^\perp devront satisfaire à

$$(w_1 \quad w_2 \quad w_3) \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = 0$$

et donc à $w_1 + w_2 = 0$. Ainsi,

$$\left\{ \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

forme une base de W^\perp et

$$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \in W \cap W^\perp.$$

Nous devons donc utiliser notre intuition des compléments orthogonaux avec prudence !

6.8 Exercices

1. a) Dans le code de Hamming $C(7, 4)$, que sont les vecteurs à envoyer ($\in \mathbb{F}_2^7$) si on désire transmettre les mots $(0, 0, 0, 0)$, $(0, 0, 1, 0)$ ou $(0, 1, 1, 1)$?
b) Le récepteur reçoit les mots : $(1, 1, 1, 1, 1, 1, 1)$, $(1, 0, 1, 1, 1, 1, 1)$, $(0, 0, 0, 0, 1, 1, 1)$ et $(1, 1, 1, 1, 0, 0, 0)$. Quels étaient les mots transmis ?
2. a) On utilise le code de Hamming $C(15, 11)$ pour corriger des messages contenant au plus un bit erroné. Si la matrice de contrôle est

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

et le message reçu est

$$w = (1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1),$$

y a-t-il eu une erreur lors de la transmission ?

b) On désire utiliser le code de Hamming $C(2^k - 1, 2^k - k - 1)$ pour un certain k , mais on ne veut pas ajouter plus de 10 % de bits au mot original. Quelle est la longueur minimale du mot original et quel est le k caractérisant le code à utiliser ?

3. Les questions suivantes portent sur le code de Hamming $C(2^k - 1, 2^k - k - 1)$.
 - a) Dans ce code, combien de lettres ont les mots u à transmettre ? Combien y a-t-il de mots distincts que l'on peut transmettre ?
 - b) Combien de lettres ont les mots encodés v ?
 - c) Combien de mots reçus w distincts (erronés ou non) seront décodés comme le même message u ?
 - d) Existe-t-il des messages reçus qui ne peuvent pas être décodés ? (Une autre façon de poser cette question est : existe-t-il un $w \in \mathbb{F}_2^{2^k-1}$ qui ne soit pas, à une erreur possible près, l'encodage v d'un message $u \in \mathbb{F}_2^{2^k-k-1}$?)
4. Vérifier que l'addition $+$ et la multiplication \times dans \mathbb{F}_2 définies par les tables de la section 6.2 remplissent les conditions de la structure de corps définie en 6.5.
5. Soit $(\mathbb{F}, +, \times)$ un corps fini. Montrer que la table de multiplication des éléments non nuls de \mathbb{F} a la propriété suivante : toutes les lignes et toutes les colonnes contiennent tous les éléments non nuls de \mathbb{F} une et une seule fois.
6. a) Dans le code de Hamming $C(7, 4)$, existe-t-il un message reçu $(w_1, w_2, w_3, w_4, w_5, w_6, w_7) \in \mathbb{F}_2^7$ qu'il est impossible de décoder comme un des 16 éléments (mots) $\in \mathbb{F}_2^4$ lorsqu'on fait l'hypothèse d'un maximum d'une lettre erronée (voir aussi l'exercice 3 d)) ?
 - b) Montrer qu'un code du même type qu'un code de Hamming transformant un mot de trois bits en un mot de huit bits ne peut corriger deux erreurs.
 - c) Construire un code transformant un mot de trois bits en un mot de dix bits et corrigeant deux erreurs.
7. a) Soit H une matrice $k \times n$, $n > k$, dont les éléments appartiennent à \mathbb{F}_2 . Soit G une matrice $(n - k) \times n$ dont les éléments appartiennent à \mathbb{F}_2 , qu'obtient de H en demandant que G soit de rang maximal et que ses lignes soient orthogonales à celles de H . Si H a la forme

$$H = \left(\underbrace{M}_{k \times (n-k)} \mid I_{k \times k} \right)$$

où M est une matrice $k \times (n - k)$ et $I_{k \times k}$ est la matrice identité $k \times k$, montrer que G peut être choisie comme

$$G = \left(I_{(n-k) \times (n-k)} \mid \underbrace{M^t}_{(n-k) \times k} \right).$$

- b) Écrire G_4 et H_4 pour le code de Hamming $C(15, 11)$, c'est-à-dire pour $k = 4$. (Commencer par H_4 .)
- c) Quel est le message u que l'émetteur voulait envoyer si celui-ci utilisait le code $C(15, 11)$ et si le message reçu se lit $(1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)$?
8. Soit $p = \frac{1}{1000}$ la probabilité qu'un bit soit transmis erronément.
- a) Quelle est la probabilité d'avoir précisément deux bits fautifs lors de la transmission de sept bits, comme lors de la transmission d'un mot du code de Hamming $C(7, 4)$?
- b) Quelle est la probabilité d'avoir plus d'une erreur lors de la transmission de sept bits?
- c) Plutôt que le code de Hamming, on transmet un bit en le répétant trois fois. On décode à la majorité. Calculer la probabilité qu'on décode correctement le bit envoyé.
- d) On transmet quatre bits en répétant chacun trois fois. Quelle est la probabilité que les quatre bits soient décodés correctement? En comparant les résultats de cette question avec b) ci-dessus, on voit que le code simple possède un léger avantage sur le code de Hamming $C(7, 4)$, mais au prix de transmettre 12 bits plutôt que sept.
9. Chaque livre a un code ISBN (pour *International Standard Book Number*) qui lui est propre. Celui-ci est composé de dix chiffres. Par exemple, ISBN 2-12345-678-0. Les trois premiers segments identifient le groupe linguistique, la maison d'édition et le volume. Le dernier symbole est un symbole détecteur d'erreur choisi parmi $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, X\}$, où X représente 10 en chiffres romains. Appelons a_i , $i = 1, \dots, 10$, les 10 symboles. Alors, a_{10} est choisi comme le reste de la division de $b = \sum_{i=1}^9 ia_i$ par 11. Ainsi, dans notre exemple, $b = 1 \times 2 + 2 \times 1 + 3 \times 2 + 4 \times 3 + 5 \times 4 + 6 \times 5 + 7 \times 6 + 8 \times 7 + 9 \times 8 = 242 = 11 \times 22 + 0$.
- a) Montrer que ce code détecte une erreur.
- b) Montrer que la somme $\sum_{i=1}^{10} ia_i$ est divisible par 11.
- c) Trouver le dernier chiffre du code ISBN commençant par

ISBN 0-7267-3514-?.

- d) Un type d'erreur commun est l'inversion de deux chiffres. Par exemple, le code 0-1311-0362-8 sera entré erronément comme 0-1311-0326-8. Montrer que le code permet de détecter une telle erreur si les deux chiffres consécutifs ne sont pas identiques (auquel cas l'inversion n'est pas une erreur!).
- e) Dans d'autres références, on dit que a_{10} est choisi de telle sorte que la somme

$$\sum_{i=1}^{10} (11 - i)a_i$$

soit divisible par 11. Montrer que cette nouvelle définition est équivalente à celle qui est donnée ci-dessus.

10. La méthode suivante a été introduite par IBM pour construire un numéro de carte de crédit. Elle est aussi utilisée au Canada dans les numéros de carte d'assurance sociale. On construit des numéros de n chiffres, a_1, \dots, a_n , où $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Le numéro est valide si le nombre b construit comme suit est un multiple de 10 :

- si i est impair on pose $c_i = a_i$;
- si i est pair et $2a_i < 10$, on pose $c_i = 2a_i$;
- si i est pair et $2a_i \geq 10$, alors $2a_i = 10 + d_i$. On pose $c_i = 1 + d_i$, c'est-à-dire la somme des chiffres de $2a_i$;
- alors

$$b = \sum_{i=1}^n c_i.$$

- a) Montrer que, si i est pair, alors c_i est le reste de la division de $2a_i$ par 9.
 - b) Les 15 premiers chiffres d'une carte sont 1234 5678 1234 567. Calculer le 16^e chiffre.
 - c) Montrer que cette méthode détecte une erreur dans un des chiffres.
 - d) Un type d'erreur commun est l'inversion de deux chiffres consécutifs. La méthode IBM ne détecte pas toujours ce genre d'erreur. Montrer cependant qu'elle permet de détecter une telle erreur si les deux chiffres consécutifs ne sont pas identiques (auquel cas l'inversion n'est pas une erreur) et s'ils ne sont pas tous les deux dans l'ensemble $\{0, 9\}$.
11. Le code suivant est construit sur le même principe que le code de Hamming. On veut envoyer un mot de quatre bits (x_1, x_2, x_3, x_4) où les $x_i = 0, 1$. On l'allonge à un mot de 11 lettres en ajoutant les bits x_5, \dots, x_{11} définis comme suit (on utilise l'addition sur \mathbb{F}_2) :

$$\begin{aligned} x_5 &= x_1 + x_4, \\ x_6 &= x_1 + x_3, \\ x_7 &= x_1 + x_2, \\ x_8 &= x_1 + x_2 + x_3, \\ x_9 &= x_2 + x_4, \\ x_{10} &= x_2 + x_3 + x_4, \\ x_{11} &= x_3 + x_4. \end{aligned}$$

Montrer que ce code détecte deux erreurs.

12. Construire le corps fini \mathbb{F}_4 à quatre éléments. (Donner explicitement les tables d'addition et de multiplication.)
13. Donner tous les éléments primitifs du corps \mathbb{F}_9 de l'exemple 6.14 construit à l'aide du polynôme $p(x) = x^2 + x + 2$.

14. a) Soient $q(x)$ et $p(x)$ deux polynômes de $\mathbb{F}[x]$. Montrer qu'il existe des polynômes $s(x)$ et $r(x) \in \mathbb{F}[x]$ tels que $q(x) = s(x)p(x) + r(x)$ avec $0 \leq \text{degré de } r < \text{degré de } p$.
b) En conclure que $q(x) \equiv r(x) \pmod{p(x)}$.
15. Soit \mathcal{M}_n l'ensemble des matrices $n \times n$, et dénotons par $+$ et \cdot l'addition et la multiplication matricielles usuelles. Est-ce que $(\mathcal{M}_n, +, \cdot)$ est un corps ? Justifier.
16. Soient \mathcal{E} un ensemble fini et $U(\mathcal{E})$ l'ensemble de ses sous-ensembles. Par exemple, si $\mathcal{E} = \{a, b, c\}$, alors $U(\mathcal{E}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. On définit sur $U(\mathcal{E})$ des opérations $+$ et \times respectivement par l'union et par l'intersection ensemblistes usuelles. Pour $+$, le neutre est \emptyset , et pour \times , le neutre est \mathcal{E} . Est-ce que $U(\mathcal{E})$ muni de $+$ et de \times forme un corps ? Le montrer ou donner les propriétés qui ne sont pas satisfaites.
17. a) Soit \mathbb{F}_3 le corps à trois éléments. Il y a neuf polynômes de degré 2 de la forme $x^2 + ax + b$ où $a, b \in \mathbb{F}_3$. Énumérer ces neuf polynômes et identifier les trois qui sont irréductibles. (Suggestion : commencer par énumérer les polynômes de la forme $(x + c)(x + d)$).
b) Afin de construire le corps à neuf éléments, on considère le quotient $\mathbb{F}_3[x]/q(x)$ où $q(x) = x^2 + 2x + 2$. Montrer que x est une racine primitive en exprimant les huit éléments x^i , $i = 1, 2, \dots, 8$, comme polynômes de degré un ou zéro.
c) Dans la notation de b), pour quel i l'égalité $x^3 + x^5 = x^i$ est-elle juste ?
d) Le corps \mathbb{F}_9 a maintenant été construit de deux façons différentes, la première dans l'exemple 6.14 à la section 6.5 et ci-dessus à la question b). Pouvez-vous construire l'isomorphisme entre le résultat de ces deux constructions (voir le théorème 6.18 pour la définition d'isomorphisme) ?
e) En a), vous avez identifié trois polynômes irréductibles. Soient $p(x)$ celui qui est utilisé dans l'exemple 6.14, $q(x)$ celui qui l'est ci-dessus en b), et $r(x)$, le troisième. Est-ce que le polynôme $i(x) = x$ est une racine primitive de $\mathbb{F}_3[x]/r(x)$? Que faudrait-il faire pour obtenir les tables d'addition et de multiplication de $\mathbb{F}_3[x]/r(x)$?
18. a) Trouver le seul polynôme irréductible sur \mathbb{F}_2 de degré 2, les deux de degré 3 et les trois de degré 4.
b) Construire les tables d'addition et de multiplication du corps \mathbb{F}_8 à huit éléments.
19. a) Pour les ambitieux : construire \mathbb{F}_{16} .
b) Également pour les ambitieux : trouver un polynôme irréductible de degré 8 sur \mathbb{F}_2 . Ce polynôme vous permettrait de construire un corps à combien d'éléments ?
20. a) On considère le code correcteur d'erreurs qui consiste à répéter trois fois chaque bit. Pour envoyer un mot de sept bits, on commence par l'allonger à 21 bits. Par exemple, pour envoyer 0100111 on envoie
000 111 000 000 111 111 111

Ce code corrige au minimum une erreur. Mais il peut en corriger plus si les erreurs sont bien placées. Quel est le nombre maximum d'erreurs qu'il peut corriger ? Sous quelle condition ?

b) On considère maintenant le code de Reed–Solomon $C(7, 3)$. Les lettres sont des éléments du corps à huit éléments, \mathbb{F}_{2^3} , identifié à $\{0, 1\}^3$, sur lequel on a une addition et une multiplication. On décrit chaque lettre comme une suite de trois bits $\underbrace{b_0 b_1 b_2}$. Combien de bits au maximum ce code peut-il corriger ? Sous quelle condition ?

21. Soit le système suivant de trois équations à trois inconnues

$$\begin{aligned} 2x - \frac{1}{2}y &= 1, \\ -x + 2y - z &= 0, \\ -y + 2z &= 1. \end{aligned} \tag{*}$$

a) Résoudre ce système sur le corps \mathbb{F}_3 à trois éléments. (Le nombre $\frac{1}{2}$ est l'inverse multiplicatif du nombre 2.)

b) Considérons le système (*) sur le corps \mathbb{F}_p à p éléments où p est un nombre premier plus grand que 2. Pour quels p le système possède-t-il une solution unique ?

22. a) Calculer, dans le corps des réels \mathbb{R} , le déterminant d suivant

$$d = \begin{vmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{vmatrix}.$$

- b) Expliquer pourquoi le déterminant d_2 de la matrice

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \in \mathbb{F}_2^{3 \times 3}$$

est égal à $d \bmod 2$.

- c) Calculer dans \mathbb{F}_3 le déterminant d_3 de la matrice

$$\begin{pmatrix} 2 & 2 & 0 \\ 2 & 2 & 2 \\ 0 & 2 & 2 \end{pmatrix} \in \mathbb{F}_3^{3 \times 3}.$$

Pouvez-vous obtenir ce déterminant à partir de la réponse de a) ?

- d) Soit le système

$$\begin{aligned} 2a - b &= 1, \\ -a + 2b - c &= 1, \\ -b + 2c &= 1. \end{aligned} \tag{*}$$

Dans quels corps, parmi $\mathbb{R}, \mathbb{F}_2, \mathbb{F}_3$, ce système possède-t-il une unique solution ? (Les coefficients entiers du système sont compris modulo 2 ou 3 si la résolution est dans \mathbb{F}_2 ou \mathbb{F}_3 respectivement.)

e) Résoudre (\star) dans \mathbb{F}_3 .

23. Cet exercice a pour but d'encoder et de décoder un message à l'aide du code de Reed–Solomon avec $m = 3$ et $k = 3$. On doit avoir construit le corps \mathbb{F}_8 auparavant (voir l'exercice 18 ci-dessus). Les calculs sont assez directs, mais ils sont nombreux : travaillez en équipe. (Tous les participants doivent choisir la même racine primitive α et avoir les mêmes tables de \mathbb{F}_8 !)
- Combien d'erreurs au maximum le code de Reed–Solomon $C(7, 3)$ peut-il corriger ?
 - Quel est l'encodage du mot $(0, 1, \alpha) \in \mathbb{F}_2^3$?
 - L'équation (6.12) peut être réécrite

$$p = Cu,$$

où $p \in F_{2^m}^{2^m-1}$, $u \in \mathbb{F}_{2^m}^k$ et $C \in \mathbb{F}_{2^m}^{(2^m-1) \times k}$. Obtenir la matrice C pour le code $C(7, 3)$.

d) Supposons que le message reçu soit

$$w = (1, \alpha^4, \alpha^2, \alpha^4, \alpha^2, \alpha^4, \alpha^2) \in \mathbb{F}_{2^m}^{2^m-1}.$$

Choisir les lignes 0, 1 et 4 du système (6.12) et résoudre afin de trouver le vecteur $(u_0, u_1, u_2) \in \mathbb{F}_8^3$.

- Combien y a-t-il de choix possibles de trois équations distinctes parmi celles de (6.12) ? Combien faudra-t-il résoudre de systèmes comme celui de la question précédente pour être sûr que la réponse précédente est le message original ?
 - Est-ce que la solution de d) est le message original ?
24. Soit p un nombre premier. Cet exercice prouve que \mathbb{Z}_p est un corps. On dit que a et b sont congrus modulo p si leur différence $a - b$ est un multiple entier de p (voir l'exemple 6.5).
- Montrer que « être congru » est une relation d'équivalence, appelée *congruence modulo p*.
 - On identifie \mathbb{Z}_p à l'ensemble des classes d'équivalence des entiers modulo p . Soit $\bar{a}, \bar{b} \in \mathbb{Z}_p$. Soient $i, j \in \bar{a}$ et $m, n \in \bar{b}$. Montrer que si $i + m \in \bar{c}$ et $j + n \in \bar{d}$, alors $\bar{c} = \bar{d}$. Même question pour $i \times m$ et $j \times n$. Cet exercice montre que les définitions de $+$ et de \times données à l'exemple 6.5 ne dépendent pas de l'élément des classes \bar{a} et \bar{b} choisi.
 - Montrer que la classe $\bar{0}$ est le neutre pour $+$ et que $\bar{1}$ est le neutre pour \times .
 - Soit $\bar{a} \in \mathbb{Z}_p$ un élément différent de $\bar{0}$. Utiliser l'algorithme d'Euclide (corollaire 7.4 du chapitre 7) pour montrer qu'il existe $\bar{b} \in \mathbb{Z}_p$ tel que $\bar{a}\bar{b} = \bar{1}$.
 - Finir de démontrer que \mathbb{Z}_p est un corps.

Références

- [1] Pohlmann, K.C. *The compact disc handbook*, 2^e édition, Madison, A-R Editions, 1992.
- [2] Papini, O. et J. Wolfmann. *Algèbre discrète et codes correcteurs*, Berlin, Springer, 1995.
- [3] Lang, S. *Undergraduate algebra*, 2^e édition, New York, Springer, 1990.
- [4] Monforte, J. « The digital reproduction of sound », *Scientific American*, n^o décembre 1984, p. 78–84.
- [5] Arnoux, P. « Minitel, codage et corps finis », *Pour la Science*, n^o mars 1988.
- [6] Lachaud, D. et S. Vladut. « Les codes correcteurs d'erreurs », *La Recherche*, n^o hors-série août 1999.
- [7] Reed, I.S. et G. Solomon. « Polynomial codes over certain finite fields », *J. Soc. Ind. Appl. Math.*, vol. 8, p. 300–304, 1960. (Cet article est contenu dans le recueil de Berlekamp.)
- [8] Berlekamp, E.R., dir. *Key papers in the development of coding theory*, IEEE Press, 1974.