Codes correcteurs d'erreur

Judith Louis-Alexandre, Camille Huynh, Dorian Lesbre

Juin 2017

Table des matières

Table des fonctions

1 Théorie des codes

1.1 Principe

Un code correcteur sert à assurer une transmission ou un stockage d'information par un canal ou support qui pourrait la détériorer. Ils sont utilisés notamment lors de l'échange de données sur internet, sur réseau mobile, ou lors du stockage de données sur un disque DVD. L'idée est de rajouter judicieusement un certain nombre de redondances afin de pouvoir retrouver le message initial si celui-ci a été légèrement altéré. Le nombre de redondance dépendant de la fiabilité du support et du besoin de correction : une sonde spatiale en utilise beaucoup pour transmettre des données, tandis que le protocole http se contente d'une simple détection d'erreur, préférant redemander le message au serveur plutôt que de le corriger.

Disons que nous voulons transmettre le message suivante :

"Le cheval blanc d'Henri 4"

Le code le plus simple est celui de répétition, on c'est-à-dire que l'on transmet

"Le cheval blanc d'Henri 4Le cheval blanc d'Henri 4"

Cependant, si une erreur se produit, on est incapable de dire à priori si la première répétition où la deuxième est la bonne :

"Le cheval blanc d'Henri 4Le cheval blanc d'Henri 5"

Ce code ne permet donc pas de corriger d'erreurs. Des codes plus complexes permettent d'obtenir de bien meilleurs résultats (exemple de code de Reed-Solomon)

"℘♣Öb%&Le cheval blanc d'Henri 4"

Ce code rajoute 6 symboles de contrôles, et peut corriger jusqu'à 5 erreurs. Ainsi si le décodeur reçoit

"℘♣Öb%&Le cheval rouge d'Henri 4"
ou
"℘êÖb%&Le μ△eva Ⓢ blanc d'Henri X"

Il pourra retrouver au message initial. Ce code est donc plus efficace que le précédent : il insère nettement moins de symboles (6 et non 25) et peut corriger 5 erreurs.

L'objectif de l'étude mathématique des codes correcteurs est de trouver des solutions optimales (comment corriger un maximum d'erreur) à partir d'une longueur de message à transmettre ou d'un nombre de symboles ajoutés fixé.

1.2 Définitions générales

Caractéristiques d'un code : afin de formaliser la notion de code correcteur, nous utiliserons les définitions suivantes :

- **L'alphabet** est un ensemble fini de symboles avec lequel sont écrits les mots du code. On prendra pour alphabet $\mathbb{F}_q$, un corps fini à q éléments muni des opérations somme et produit.
- **Le message** A est une liste de k symboles à transmettre, on l'assimile à un élément de $\mathbb{F}_q^k$
- **L'encodeur** est un algorithme qui transforme le message A en un mot du code C de longueur n comportant des redondances. On l'assimile à une fonction injective $enc : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$
- **Le code** est l'ensemble des mots possibles : $\mathcal{C} = \left\{\text{enc}(B), B \in \mathbb{F}_q^k\right\}$
- **Le mot reçu** $D \in \mathbb{F}_q^n$ est une version potentiellement corrompue du message émis $D = C + E$, avec E les erreurs introduites.

— **La distance de Hamming** entre deux mots C_1, C_2 noté $\mathrm{d}(C_1, C_2)$ est le nombre de symboles distincts entre C_1, C_2.

On appelle poids la distance au mot nul : $w(C) = \mathrm{d}(C, 0)$

On introduit alors les grandeurs suivantes pour caractériser un code

— q le cardinal de l'alphabet (nombre de symboles distincts)

— k la dimension d'un code (la longueur du message initial)

— p le nombre de symboles de contrôle

— $n = k + p$ la longueur d'un mot du code

— la distance minimale d'un code $d_{\min} = \min\limits_{\substack{(c_1,\, c_2) \in \mathcal{C}^2 \\ c_1 \neq c_2}} \mathrm{d}(c_1, c_2)$

On condensera toutes ces notations en disant que $\mathcal{C}$ est un code $(n, k, d_{\min})_q$.

Le décodeur est un algorithme qui retrouve C à partir de $D = C + E$ si E est de poids assez faible. On l'assimile à un une fonction $\mathrm{dec} : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^k$ vérifiant $\forall A \in \mathbb{F}_q^k, \mathrm{dec}(\mathrm{enc}(A)) = A$ et

$$\forall A \in \mathbb{F}_q^k,\ \forall E \in \mathbb{F}_q^n,\ w(E) \leqslant \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \Rightarrow \mathrm{dec}\big(\mathrm{enc}(A) + E\big) = A$$

Proposition 1 : (Propriétés de la distance de Hamming)
La distance de Hamming est symétrique et vérifie pour tout $(c_1,\, c_2,\, c_3) \in (A^n)^3$:

$$\mathrm{d}(c_1, c_2) = \mathrm{d}(c_1 - c_2, 0) = w(c_1 - c_2) \tag{1}$$

$$\mathrm{d}(c_1, c_2) + \mathrm{d}(c_2, c_3) \geqslant \mathrm{d}(c_1, c_3) \geqslant |\mathrm{d}(c_1, c_2) - \mathrm{d}(c_2, c_3)| \tag{2}$$

Démonstration : notons $c_1 = (\alpha_1, \ldots, \alpha_n)$, $c_2 = (\beta_1, \ldots, \beta_n)$ et $c_3 = (\gamma_1, \ldots, \gamma_n)$. On a alors

$$\mathrm{d}(c_1, c_2) = \sum_{i=1}^{n} (1 - \delta_{\alpha_i, \beta_i})$$

où δ est le symbole de Kronecker. On sait que pour tout $i \in [\![1,\, n]\!]$, $\delta_{\alpha_i,\beta_i} = \delta_{\beta_i,\alpha_i} = \delta_{\alpha_i - \beta_i, 0}$ d'où la symétrie et le premier point.

Soit $i \in [\![1,\, n]\!]$,

Cas 1 : $\delta_{\alpha_i,\gamma_i} = 0$, alors $\alpha_i \neq \gamma_i$ donc $\alpha_i \neq \beta_i$ ou $\gamma_i \neq \beta_i$. donc $\delta_{\alpha_i,\beta_i} = 0$ ou $\delta_{\gamma_i,\beta_i} = 0$. Ce qui donne $(1 - \delta_{\alpha_i,\beta_i}) + (1 - \delta_{\gamma_i,\beta_i}) \geqslant 1$ et $|(1 - \delta_{\alpha_i,\beta_i}) - (1 - \delta_{\gamma_i,\beta_i})| \leqslant 1$ donc

$$|(1 - \delta_{\alpha_i,\beta_i}) - (1 - \delta_{\gamma_i,\beta_i})| \leqslant (1 - \delta_{\alpha_i,\gamma_i}) \leqslant (1 - \delta_{\alpha_i,\beta_i}) + (1 - \delta_{\gamma_i,\beta_i})$$

Cas 2 : $\delta_{\alpha_i,\gamma_i} = 1$. On a évidemment $(1 - \delta_{\alpha_i,\gamma_i}) \leqslant (1 - \delta_{\alpha_i,\beta_i}) + (1 - \delta_{\gamma_i,\beta_i})$.

Si $\alpha_i = \beta_i$ alors $\gamma_i = \beta_i$ donc $|(1 - \delta_{\alpha_i,\beta_i}) - (1 - \delta_{\gamma_i,\beta_i})| = |(0) - (0)| = 0$

Si $\alpha_i \neq \beta_i$ alors $\gamma_i \neq \beta_i$ donc $|(1 - \delta_{\alpha_i,\beta_i}) - (1 - \delta_{\gamma_i,\beta_i})| = |(1) - (1)| = 0$

En sommant sur i, on obtient $|\mathrm{d}(c_1, c_2) - \mathrm{d}(c_2, c_3)| \leqslant \mathrm{d}(c_1, c_3) \leqslant \mathrm{d}(c_1, c_2) + \mathrm{d}(c_2, c_3)$ □

1.3 Codes correcteurs

Code t-correcteurs : soit $t \geqslant 0$, notons $\mathcal{C} \in \mathcal{P}(\mathcal{A}^n)$ un code sur l'alphabet $\mathcal{A}$. Pour chaque mot $c \in \mathcal{C}$ du code, on définit l'ensemble

$$\mathcal{E}(c, t) = \{a \in \mathcal{A}^n \,/\, \mathrm{d}(c, a) \leqslant t\}$$

. C'est une boule de centre c et de rayon t (au sens de la distance de Hamming).

— $\mathcal{C}$ est t-correcteur si les ensembles $\big(\mathcal{E}(c,\, t)\big)_{c \in \mathcal{C}}$ sont disjoints.

— $\mathcal{C}$ est parfait si les ensembles $\big(\mathcal{E}(c,\, t)\big)_{c \in \mathcal{C}}$ forment une partition de $\mathcal{A}^n$

Si $\mathcal{C}$ code t-correcteur, on peut attribuer à chaque élément de $\mathcal{A}^n$ distant de moins de t du code un unique mot du code le plus proche. Si $\mathcal{C}$ est parfait, alors tout élément de $\mathcal{A}^n$ admet un unique mot du code le plus proche.

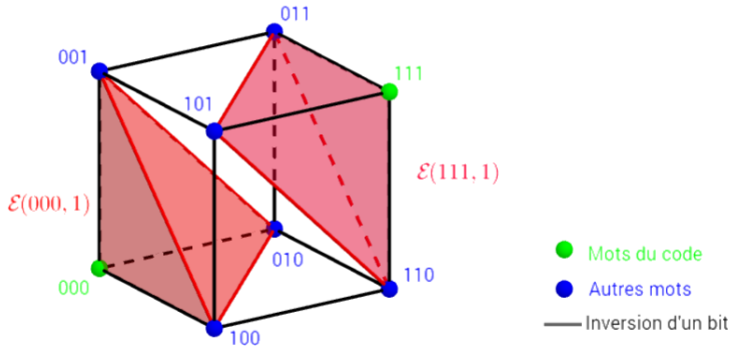

Illustration des ensembles $\mathcal{E}(c,t)$ pour le code 1-correcteur parfait Hamming $(3,1)$: $enc : \begin{cases} \mathbb{F}_2 & \to & \mathbb{F}_2^3 \\ 0 & \mapsto & 000 \\ 1 & \mapsto & 111 \end{cases}$

Proposition 2 :
Soit $\mathcal{C}$ un code possédant q^k éléments sur $\mathbb{F}_q^n$, t un entier positif et $c \in \mathcal{C}$. On a

1. $\#\mathcal{E}(c,\, t) = 1 + n(q-1) + \binom{n}{2}(q-1)^2 + \ldots + \binom{n}{t}(q-1)^t = \sum_{\ell=0}^{t} \binom{n}{\ell}(q-1)^\ell$
2. si $\mathcal{C}$ est t-correcteur alors $\#\mathcal{E}(c,t) \leqslant q^{n-k}$
3. si $\mathcal{C}$ est parfait alors $\#\mathcal{E} = q^{n-k}$.

Démonstration : pour construire un élément d de $\mathcal{E}(c,t)$, de manière unique :

1. on choisit $\ell \in [\![0,\, t]\!]$ la distance de Hamming entre d et c :
2. on choisit l'emplacement de ces ℓ coefficients distincts : $\binom{n}{\ell}$ possibilités.
3. on choisit la valeur de chaque coefficient : $(q-1)$ possibilités pour chaque (n'importe quel élément de $\mathcal{A}$ différent du coefficient correspondant de c), donc $(q-1)^\ell$ en tout.

En sommant, on obtient $\#\mathcal{E}(c,\, t) = \sum_{\ell=0}^{t} \binom{n}{\ell}(q-1)^\ell$.

Si le code est t-correcteur, alors $\bigcup_{c\in\mathcal{C}} E(c,t) \subset \mathcal{A}^n$ est une union disjointe (avec égalité si $\mathcal{C}$ est parfait), donc en passant aux cardinaux $\sum_{c\in\mathcal{C}} \sum_{\ell=0}^{t} \binom{n}{\ell}(q-1)^\ell \leqslant q^n$ (avec égalité si $\mathcal{C}$ est parfait), or $\#\mathcal{C} = q^k$, on obtient alors : $\sum_{\ell=0}^{t} \binom{n}{\ell}(q-1)^\ell \leqslant q^{n-k}$ avec égalité si $\mathcal{C}$ est parfait. □

Proposition 3 :
Soit $\mathcal{C}$ un code de distance minimale $d_{\min} \geqslant 1$. Alors

$$\mathcal{C} \text{ est } \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor \text{-correcteur mais n'est pas } \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor + 1\text{-correcteur.}$$

Démonstration : Posons $t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$.

Soit $(c_1, c_2) \in \mathcal{C}^2$, montrons que $\mathcal{E}(c_1, t) \cap \mathcal{E}(c_2, t) = \emptyset$. Soit $h \in \mathcal{E}(c_1, t)$. On sait que $\mathrm{d}(h, c_1) \leqslant t$. D'après l'inégalité $\mathrm{d}(h, c_2) \geqslant \mathrm{d}(c_2, c_1) - \mathrm{d}(h, c_1)$ on a $\mathrm{d}(h, c_2) \geqslant d_{\min} - \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor > \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$ donc $h \notin \mathcal{E}(c_2, t)$. Les ensembles sont disjoints.

Il existe $(c_1, c_2) \in \mathcal{C}^2$ tel que $\mathrm{d}(c_1, c_2) = d_{\min}$. Donc en modifiant $t + 1$ symboles de c_1 différents de ceux de c_2 en symboles de c_2, on obtient un mot h de distance $t + 1$ de c_1 et moins de $t + 1$ de c_2. Donc $h \in \mathcal{E}(c_1, t + 1) \cap \mathcal{E}(c_2, t + 1)$. Le code n'est pas $t + 1$ correcteur. □

1.4 Codes linéaires

Code linéaire : un code $\mathcal{C}$ est dit linéaire de longueur n et de dimension k sur l'alphabet $\mathcal{A}$ si $\mathcal{C}$ est un sous-espace vectoriel de $\mathcal{A}^n$ de dimension k. En pratique $\mathcal{A} = \mathbb{F}_q$ est un corps fini (et donc $\mathcal{A}^n$ est un $\mathbb{F}_q$ espace vectoriel de dimension finie n).

Matrice génératrice : soit $G \in \mathcal{M}_{n,k}(\mathbb{F}_q)$. On dit que G est une matrice génératrice du code linéaire $\mathcal{C}$ lorsque la famille des colonnes de G est un base de $\mathcal{C}$. Ainsi G est de rang k et $x \mapsto Gx$ est une fonction d'encodage possible.

Matrice de contrôle : $\mathbb{F}_q^n$ est muni du produit scalaire usuel (la somme des produits des coordonnées). Comme $\mathbb{F}_q^n$ est de dimension finie, il existe un unique sous-espace vectoriel $\mathcal{C}^\perp$ de dimension $n - k$. Notons $H \in \mathcal{M}_{n,n-k}(\mathbb{F}_q)$ une matrice génératrice de $\mathcal{C}^\perp$, appelée matrice de contrôle.

Proposition 4 :
Soit $\mathcal{C}$ un code linéaire. Il existe une matrice de contrôle H et une matrice génératrice G. De plus, pour toute matrice de contrôle H et pour toute matrice génératrice G on a :

1. ${}^tHG = 0$
2. $\forall x \in \mathbb{F}_q^n,\ x \in \mathcal{C} \Leftrightarrow {}^tHx = 0$

Démonstration : $\mathcal{C}$ étant un sous-espace vectoriel non-nul de $\mathbb{F}_q^n$, espace de dimension finie, $\mathcal{C}$ est de dimension finie. Donc il existe une base de $\mathcal{C}$, et donc une matrice G. De même il existe un supplémentaire orthogonal de dimension finie, et donc une base de ce supplémentaire.

${}^tHG = 0$ car les coefficients du produit correspondant au produit scalaire d'une colonne de H, donc d'un vecteur de $\mathcal{C}^\perp$ est d'une colonne de G (vecteur de $\mathcal{C}$).

Soit $x \in \mathbb{F}_q^n$. on note $(c_1, \ldots, c_n) \in (\mathbb{F}_q^n)^{n-k}$ les colonnes de H, c'est une base de $\mathcal{C}^\perp$. On a les équivalences suivantes :

$$\begin{aligned} x \in \mathcal{C} &\Leftrightarrow x \in (\mathcal{C}^\perp)^\perp \\ &\Leftrightarrow \forall c \in (\mathcal{C}^\perp),\ \langle x, c \rangle = 0 \\ &\Leftrightarrow \forall i \in [\![1, n-k]\!],\ \langle c_i, x \rangle = 0 \\ &\Leftrightarrow Hx = 0 \end{aligned}$$

□

Proposition 5 : (Distance minimale d'un code linéaire)
Soit $\mathcal{C}$ un code linéaire. La distance minimale $d_{\min}$ entre deux mots de $\mathcal{C}$ vérifie :

$$d_{\min} = \min_{c \in \mathcal{C} \setminus \{0\}} w(c)$$

Démonstration : on a par définition :

$$\begin{aligned} d_{\min} &= \min_{\substack{(c_1,c_2)\in\mathcal{C}^2 \\ c_1\neq c_2}} \mathrm{d}(c_1,c_2) \quad \text{or pour tout } (a,b)\in\mathcal{A}^2, \mathrm{d}(a,b)=\mathrm{d}(a-b,0) \\ &= \min_{\substack{(c_1,c_2)\in\mathcal{C}^2 \\ c_1\neq c_2}} \mathrm{d}(c_1-c_2,0) = \min_{\substack{(c_1,c_2)\in\mathcal{C}^2 \\ c_1\neq c_2}} w(c_1-c_2) \end{aligned}$$

le code étant linéaire, tout différence de mots du code est un mot du code et tout mot du code peut s'écrire sous la forme d'un différence : $\{c_1-c_2,\ (c_1,c_2)\in\mathcal{C}^2 \,/\, c_1\neq c_2\} = \mathcal{C}\setminus\{0\}$.
On a alors $d_{\min} = \min\limits_{c\in\mathcal{C}\setminus\{0\}} w(c)$ □

Définition : On définit la borne de Hamming d'un code t-correcteur par :

$$V_t = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$$

avec q le cardinal de l'alphabet, et n la longueur d'un mot du code.

Proposition 6 : (Majoration de Hamming)
Pour tout code $\mathcal{C}$ t-correcteur, on a la majoration suivante :

$$\#C \leqslant \frac{q^n}{V_t}$$

avec q le nombre de lettres dans l'alphabet du code et V_t la borne de Hamming du code.

Démonstration : C'est une reformulation de la proposition 2 □

Proposition 7 :
Un code est parfait si et seulement si sa borne de Hamming est atteinte.

Démonstration : Le sens direct a été montré dans la proposition 2.

Soit $\mathcal{C}$ un code t-correcteur. Montrons que si sa borne de Hamming est atteinte, $\mathcal{C}$ est un code parfait. Soit $\mathcal{B} = \{a\in\mathcal{A}^n \,/\, \exists c\in\mathcal{C}, a\in\mathcal{E}(c,t)\}$ l'ensemble des éléments de $\mathcal{A}^n$ appartenant à une boule de rayon t centrée en un mot du code. Alors on a exactement $\#\mathcal{B} = \#\mathcal{C}\times V_t$, les boules étant disjointes puisque le code est t-correcteur. On a alors $\#\mathcal{B} = q^n = \#\mathcal{A}^n$ puis $\mathcal{B} = \mathcal{A}^n$, ce qui montre que le code est parfait. □

2 Structure de corps finis

2.1 Construction des corps finis

2.1.1 Corps premiers

Construction : soit p un entier naturel non nul. La relation de congruence modulo p est une relation d'équivalence qui permet de construire l'ensemble $\mathbb{Z}/p\mathbb{Z} = \{\mathrm{Cl}(x), x\in\mathbb{Z}\}$. Cet ensemble est composé de p éléments : les classes des entiers de 0 à $p-1$ notées $\overline{0},\ldots,\overline{p}$. On définit les opérations $+$ et $\times$ dans $\mathbb{Z}/p\mathbb{Z}$ par :

— $\forall(\overline{a},\overline{b})\in(\mathbb{Z}/p\mathbb{Z})^2,\ \overline{a}+\overline{b} = \overline{a+b}$
— $\forall(\overline{a},\overline{b})\in(\mathbb{Z}/p\mathbb{Z})^2,\ \overline{a}\times\overline{b} = \overline{a\times b}$

Ces opérations sont bien définies car, pour $(a,b,c,d)\in\mathbb{Z}^4$ si $a\equiv c\ [p]$ et $b\equiv d\ [p]$ alors $a+b\equiv c+d\ [p]$ et $ab\equiv cd\ [p]$

Proposition 8 : (Structure de $(\mathbb{Z}/p\mathbb{Z}, +, \times)$)
L'ensemble $(\mathbb{Z}/p\mathbb{Z}, +, \times)$ est un anneau commutatif d'élément neutre $\overline{0}$ pour l'addition et $\overline{1}$ pour le produit. De plus $(\mathbb{Z}/p\mathbb{Z}, +, \times)$ est un corps si et seulement si p est premier.

Démonstration : soit $(\overline{a}, \overline{b}, \overline{c}, \overline{d}) \in (\mathbb{Z}/p\mathbb{Z})^4$
Groupe commutatif $(\mathbb{Z}/p\mathbb{Z}, +)$: elle s'hérite de la structure de groupe commutatif de $(\mathbb{Z}, +)$:
Associativité : $(\overline{a} + \overline{b}) + \overline{c} = \overline{a+b} + \overline{c} = \overline{a+b+c} = \overline{a} + \overline{b+c} = \overline{a} + (\overline{b} + \overline{c})$
Commutativité : $\overline{a} + \overline{b} = \overline{a+b} = \overline{b} + \overline{a}$
Neutre : $\overline{0} + \overline{a} = \overline{0+a} = \overline{a}$ et donc $\overline{a} + \overline{0} = \overline{a}$
Inversibilité : $\overline{a} + \overline{-a} = \overline{-a} + \overline{a} = \overline{0}$
Anneau $(\mathbb{Z}/p\mathbb{Z}, +, \times)$:
Associativité : $(\overline{a} \times \overline{b}) \times \overline{c} = \overline{a \times b} \times \overline{c} = \overline{a \times b \times c} = \overline{a} \times \overline{b \times c} = \overline{a} \times (\overline{b} \times \overline{c})$
Distributivité : $(\overline{a} + \overline{b}) \times (\overline{c} + \overline{d}) = \overline{a+b} \times \overline{c+d} = \overline{(a+b)(c+d)} = \overline{ac + ad + bc + bd}$
$= \overline{ac} + \overline{ad} + \overline{bc} + \overline{bd}$
Neutre : $\overline{a} \times \overline{1} = \overline{a \times 1} = \overline{a}$ et $\overline{1} \times \overline{a} = \overline{a}$
Commutativité : $\overline{a} \times \overline{b} = \overline{a \times b} = \overline{b} \times \overline{a}$
Supposons p premier et $a \not\equiv 0[p]$. Il existe $x \in [\![1,\ p-1]\!]$ tel que $\overline{a} = \overline{x}$. $x < p$ et p est premier, donc $x \wedge p = 1$. D'après le théorème de Bezout, il existe $(u, v) \in \mathbb{Z}^2$, $ux + vp = 1$.
$\overline{ux} + \overline{vp} = \overline{1}$, or $\overline{p} = 0$ et $\overline{x} = \overline{a}$, on a donc $\overline{av} = \overline{1}$ et $\overline{va} = \overline{av} = \overline{1}$
Supposons p non premier.
Cas 1 : $p = 1$. Alors $0 \equiv 1\ [p]$ donc $\overline{0} = \overline{1}$, $\overline{0} \times \overline{0} = \overline{1}$ donc $\overline{0} \in \mathcal{U}(\mathbb{Z}/p\mathbb{Z})$. Alors $\mathbb{Z}/p\mathbb{Z}$ n'est pas un corps.
Cas 2 : $p \geqslant 2$. p est non premier, donc $p \geqslant 3$. Il existe $(u, v) \in [\![2,\ p-1]\!]^2$ tel que $uv = p$, donc $\overline{uv} = \overline{0}$. Si $\mathbb{Z}/p\mathbb{Z}$ est un corps alors $\overline{u}^{-1}$ et $\overline{v}^{-1}$ existent. On a $\overline{v}^{-1}\overline{u}^{-1}\overline{uv} = \overline{v}^{-1}\overline{v} = \overline{1}$ or $\overline{v}^{-1}\overline{u}^{-1}\overline{uv} = \overline{v}^{-1}\overline{u}^{-1}\overline{0} = \overline{0}$ donc $\overline{1} = \overline{0}$. On en déduit $\overline{0} \times \overline{0} = \overline{1}$, c'est-à-dire $0 \equiv 1\ [p]$. $p \mid 1$ et $p \in \mathbb{N}^*$ donne $p = 1$ □

2.1.2 Corps de Galois

Construction : soit p un nombre premier et $n \in \mathbb{N}^*$. On notera $\mathbb{F}_p$ le corps $\mathbb{Z}/p\mathbb{Z}$. Soit P de $\mathbb{F}_p[X]$ de degré n. On définit la relation $\equiv$ par $\forall (A, B) \in \mathbb{F}_p[X], A \equiv B \Leftrightarrow P \mid (A - B)$. C'est une relation d'équivalence. On pose alors $\mathbb{F}_p[X]/P$ l'ensemble des classe d'équivalences. Il contient p^n éléments : les classes des polynômes de $\mathbb{F}_{p,\,n-1}[X]$ (qui est un $\mathbb{F}_p$ espace vectoriel de dimension n). On note $\overline{A}$ la classe de A. Similairement à la congruence sur les entiers, on définit les opérations :
— $\forall (\overline{A}, \overline{B}) \in (\mathbb{Z}/p\mathbb{Z})^2,\ \overline{A} + \overline{B} = \overline{A+B}$
— $\forall (\overline{A}, \overline{B}) \in (\mathbb{Z}/p\mathbb{Z})^2,\ \overline{A} \times \overline{B} = \overline{A \times B}$
Elle sont de même bien définies : pour $(\overline{A}, \overline{B}, \overline{C}, \overline{D}) \in \mathbb{F}_p[X]^4$ tel que $A \equiv B$ et $C \equiv D$, on sait que $(A + C) \equiv (B + D)$ et $(AC) \equiv (BD)$

Proposition 9 : (Structure de $(\mathbb{F}_p[X]/P, +, \times)$)
L'ensemble $(\mathbb{F}_p[X]/P, +, \times)$ ainsi défini est un anneau commutatif d'élément neutre $\overline{0}$ pour l'addition et $\overline{X^0}$ pour le produit. De plus $(\mathbb{F}_p[X]/P, +, \times)$ est un corps si et seulement si P est irréductible dans $\mathbb{F}_p[X]$.

Démonstration : la structure d'anneau commutatif s'hérite de celle de $\mathbb{F}_p[X]$
Supposons P irréductible dans $\mathbb{F}_p[X]$. Soit $\overline{A} \in \mathbb{F}_p[X]/P \setminus \{\overline{0}\}$.
Il existe $C \in \mathbb{F}_{p,\,n-1}[X]$ tel que $\overline{A} = \overline{C}$. $\deg C < \deg P$ et P est irréductible donc $C \wedge P = 1$. D'après le théorème de Bezout, il existe $(U, C) \in \mathbb{F}_p[X]^2$, $UC + VP = 1$.
$\overline{UC} + \overline{VP} = \overline{1}$, or $\overline{P} = 0$ et $\overline{C} = \overline{A}$, on a donc $\overline{AV} = \overline{1}$ et $\overline{VA} = \overline{AV} = \overline{1}$
Supposons P non irréductible dans $\mathbb{F}_p[X]$. Il existe $(C, D) \in \mathbb{F}_p[X]$ non constant tel que $CD = P$.
On a alors $\overline{CD} = \overline{0}$. Si $\mathbb{F}_p[X]/P$ est un corps alors $\overline{C}^{-1}$ et $\overline{D}^{-1}$ existent, et $\overline{D}^{-1}\overline{C}^{-1}\overline{CD} = \overline{1}$

or $\overline{CD} = \overline{0}$ donc $\overline{D}^{-1}\overline{C}^{-1}\overline{CD} = \overline{0}$. On a $\overline{1} = \overline{0}$, donc $\overline{0} \in \mathcal{U}(\mathbb{F}_p[X]/P)$, d'où $\mathbb{F}_p[X]/P$ n'est pas un corps. □

Proposition 10 :
Soit p premier et $n \geqslant 2$ entier. Le polynôme $X^{p^n} - X \in \mathbb{F}_p[X]$ est égal au produit de tous les polynômes irréductibles unitaires de $\mathbb{F}_p[X]$ dont le degré divise n.

Démonstration :
Soit $P \in \mathbb{F}_p[X]$ irréductible de degré d divisant n. Le corps $\mathbb{F}_{p^d} = \mathbb{F}_p[X]/P$ existe. On sait que le groupe multiplicatif d'un corps fini est cyclique d'après la proposition 13. Il existe donc un élément $\overline{A} \in (\mathbb{F}_p[X]/P) \setminus \{0\}$ générateur de $\left((\mathbb{F}_p[X]/P) \setminus \{0\}, \times\right)$.
$\overline{X} \neq 0$ donc il existe $i \in [\![1, p^d]\!]$ tel que $\overline{X} = \overline{A}^i$.
$\overline{X}^{p^d} = \left(\overline{A}^{p^d}\right)^i = \overline{A}^i = \overline{X}$. Or $p \mid n$ donc $\overline{X}^{p^n} = \overline{X}$, P divise $X^{p^n} - X$.
Soit P un diviseur irréductible de $\mathbb{F}_p[X]$ de degré d. Posons $f : \left\{ \begin{array}{ccc} \mathbb{F}_p[X]/P & \to & \mathbb{F}_p[X]/P \\ x & \mapsto & x^p \end{array} \right.$.
Soit $(\overline{a}, \overline{b}) \in (\mathbb{F}_p[X]/P)^2$ $f(\overline{a} + \overline{b}) = (\overline{a}\overline{b})^p = \overline{a}^p\overline{b}^p = f(\overline{a})f(\overline{b})$ et
$f(\overline{a} + \overline{b}) = (\overline{a} + \overline{b})^p = \sum_{i=0}^{p} \binom{p}{i} \overline{a}^i\overline{b}^{p-i}$ or p premier donc $p \left| \binom{p}{i} \right.$ d'où $\binom{p}{i} = 0$ dans $\mathbb{F}_p$,
$f(\overline{a} + \overline{b}) = \overline{a}^p + \overline{b}^p = f(\overline{a}) + f(\overline{b})$
f est donc un endomorphisme de corps.
Supposons $f(\overline{a}) = f(\overline{b})$, $f(\overline{a} - \overline{b}) = \overline{0}$, si $\overline{a} - \overline{b} \neq \overline{0}$ alors $f((\overline{a} - \overline{b})^{-1})f(\overline{a} - \overline{b}) = f(\overline{1}) = \overline{1}^p = \overline{1}$ absurde $\overline{0}$ non inversible. $\overline{a} = \overline{b}$, f est injective, et donc surjective (le corps étant fini).
f est un isomorphisme de corps, donc son itérée n-ième f^n est également un automorphisme de corps. On admet que l'ensemble des points fixes de f^n est un sous-corps $\mathbb{K}$ de $\mathbb{F}_p[X]/P$.
Or $\overline{X}^{p^n} - \overline{X}$ est un multiple de P dans $\mathbb{F}_p[X]/P$ donc $\overline{X}^{p^n} = \overline{X}$. On a $f^n(\overline{X}) = \overline{X}$ ce qui revient à $\overline{X} \in \mathbb{K}$. $\mathbb{K}$ est stable par $+$ et $\times$ donc on peut reconstruire tout $\mathbb{F}_p[X]/P$ à partir de $\overline{X}$ et $\overline{1}$, tous deux éléments de $\mathbb{K}$. D'où l'égalité $\mathbb{K} = \mathbb{F}_p[X]/P$.
Il existe $\overline{A}$ un élément d'ordre $p^d - 1$ de $(\mathbb{F}_p[X]/P \setminus \{0\}, \times)$. $\overline{A} \in \mathbb{K}$, donc $f^n(\overline{A}) = \overline{A}$. On a $\overline{A}^{p^n-1} = \overline{1}$ donc $p^d - 1 \mid p^n - 1$, d'où d divise n.
Montrons qu'il n'y a pas de facteur double. Soit $P \in \mathbb{F}_p[X]$ unitaire tel que $P^2 \mid X^{p^n} - X$. Il existe $C \in \mathbb{F}_p[X]$, $P^2C = X^{p^n} - X$. Ce qui en dérivant donne $P(P'C + PC') = p^nX^{p^n-1} - 1$ or $p^n \equiv 0\ [p]$ donc $P \mid -1$, $P = 1$ □

Proposition 11 : (Existence de $\mathbb{F}_{p^n}$)
Soit p un nombre premier et $n \in \mathbb{N}^*$. Il existe un polynôme irréductible P de $\mathbb{F}_p[X]$ de degré n, et donc un corps fini de cardinal p^n.

Démonstration : On a déjà montré que s'il existe un polynôme irréductible de $\mathbb{F}_p[X]$ de degré n, alors il existe un corps fini à p^n éléments. On note $m_n(p)$ le nombre de polynômes irréductibles unitaires de $\mathbb{F}_p[X]$ de degré n. Soit p un nombre premier. Montrons par récurrence forte sur $n \in \mathbb{N}^*$.

$$H(n) : "1 \leqslant m_n(p) \leqslant \frac{p^n}{n}"$$

Initialisation : $n = 1$. les polynômes irréductibles de degré 1 sont les $(X - p)_{p \in \mathbb{F}_p}$.
Donc $m_1(p) = \#F_p = p$. on a bien $1 \leqslant p \leqslant \frac{p^1}{1}$

Hérédité : soit $n \geqslant 2$, supposons que pour $k \in [\![1, n-1]\!]$, $H(k)$ vraie.
On sait que $X^{p^n} - X = \prod_{s \in \mathcal{D}_{\mathbb{N}}(n)} \left(\prod_{P \in G_s} P \right)$ où $G_s = \{P \in \mathbb{F}_p[X] \,/\, P \text{ irréductible et } \deg P = s\}$ et

$\mathcal{D}_{\mathbb{N}}(n)$ est l'ensemble des diviseurs positifs de n.
Ce qui donne en passant au degré : $p^n = \sum_{s\in\mathcal{D}_{\mathbb{N}}(n)} sm_s(p)$.

n divise n donc $p^n = \sum_{s\in\mathcal{D}_{\mathbb{N}}(n)\setminus\{n\}} sm_s(p) + nm_n(p)$ pour $s < n$ tel que $s \mid n$ on a par hypothèse de récurrence $m_s \geqslant 0$ d'où la majoration $m_n(p) \leqslant \dfrac{p^n}{n}$.

De plus $p^n - nm_n(p) = \sum_{s\in\mathcal{D}_{\mathbb{N}}(n)\setminus\{n\}} sm_s(p) \leqslant \sum_{s\in\mathcal{D}_{\mathbb{N}}(n)\setminus\{n\}} s\dfrac{p^s}{s} \leqslant \sum_{s=0}^{\lfloor n/2\rfloor} p^s = \dfrac{p^{\left\lfloor\frac{n}{2}\right\rfloor+1}-1}{p-1}$ ($p \neq 1$ car p premier).

On a par ailleurs $\dfrac{p^{\left\lfloor\frac{n}{2}\right\rfloor+1}-1}{p-1} \leqslant p^{\left\lfloor\frac{n}{2}\right\rfloor+1}$, d'où la minoration $m_n(p) \geqslant \dfrac{p^n - p^{\left\lfloor\frac{n}{2}\right\rfloor+1}}{n}$.

p et n sont supérieurs ou égaux à 2 donc $n > \left\lfloor\dfrac{n}{2}\right\rfloor + 1$, d'où $p^n - p^{\left\lfloor\frac{n}{2}\right\rfloor+1} > 0$. $m_n(p) > 0$ or $m_n(p)$ est entier. $m_n(p) \geqslant 1$

Conclusion : pour tout $n \in \mathbb{N}^*$, il existe un polynôme irréductible de $\mathbb{F}_p[X]$ de degré n et donc un corps fini de cardinal p^n □

2.2 Propriétés des corps finis

Proposition 12 :
Dans un groupe commutatif, l'ensemble des ordres des éléments est stable par ppcm

Démonstration : Soit $(G, \cdot)$ un groupe commutatif, et $(x, y) \in G$. On note m, n les ordres respectifs de x et y, alors $\text{ppcm}(m, n)$ est l'ordre de $x \cdot y$. □

Proposition 13 :
Soit $\mathbb{K}$ un corps fini commutatif. Alors $(\mathbb{K}^*, \times)$ est un groupe cyclique.

Démonstration : Montrons pour cela qu'un corps fini commutatif est un anneau intègre. Pour tout $a \in \mathbb{K}^*$, les applications $x \mapsto a^{-1} \cdot x$ et $x \mapsto a \cdot x$ sont réciproques l'une de l'autre, donc bijectives. On a donc pour tout $(a, b) \in \mathbb{K}^2, a \neq 0$:

$$a \cdot b = 0 \Leftrightarrow b \in \ker(x \mapsto a \cdot x) \Leftrightarrow b = 0$$

Par le caractère commutatif, la propriété pour $b \cdot a$ est également vérifiée, ce qui donne que $\mathbb{K}$ est un anneau intègre.

On note $\omega(\mathbb{K})$ le ppcm des ordres des éléments de $(\mathbb{K}, \times)$. Alors $\omega(\mathbb{K}) \leqslant \#\mathbb{K}$, puis $\omega(\mathbb{K}) = \#\mathbb{K}$ car $\mathbb{K}$ est un anneau intègre. Or l'ensemble des ordres des éléments d'un groupe commutatif est stable par ppcm. Il existe donc $\alpha \in \mathbb{K}$ tel que son ordre soit $\omega(\mathbb{K})$. On a donc $(e, \alpha, \alpha^2, \ldots, \alpha^{\#\mathbb{K}-1}) \subset \mathbb{K}$, puis par égalité des cardinaux, $\mathbb{K} = (\alpha^i)_{0\leqslant i\leqslant\#\mathbb{K}}$. □

3 Codes de Hamming

3.1 Théorie

Principe du code : Le code de Hamming est un code linéaire permettant de transmettre des messages. Le code prend en argument deux paramètres : n, la longueur du code et k, la dimension du code tel qu'il existe $r \geqslant 2$ tel que : $n = 2^r - 1$ et $k = 2^r - r - 1$. Le message à transmettre est découpé en paquets de n bits et l'information à transmettre est codé sur k bits. Chaque bit est un élément de l'ensemble $\{0, 1\}$. Le code de Hamming permet de détecter jusqu'à deux erreurs, qui auraient lieu lors de la transmission du message, mais ne peut en corriger qu'une seule.

Principe d'encodage : Le message à transmettre est découpé en paquets de n bits. L'information à transmettre est codé sur k bits. Les bits correspondant à des puissances de 2 sont utilisés pour contrôler le message transmis. On peut alors ajouter un bit de parité pour pouvoir déceler d'éventuelles erreurs de transmission. Ce bit de parité est ajouté de telle sorte qu'il y ait un nombre pair de 1 par paquet de n bits.
Les bits de contrôle complètent la parité de l'ensemble des bits d'information dans lesquels ils apparaissent dans la décomposition en base 2. Ainsi, pour un message codé sur 7 bits, le bit de contrôle 1 contrôle la parité des bits 3, 5 et 7, le bit de contrôle 2 contrôle la parité des bits 3 ,6 et 7 et le bit de contrôle 4 celle des bits 5, 6 et 7. Grâce à ces bits de contrôle une erreur commise lors de la transmission peut être détectée et corrigée.

Position des bits		2^0	2^1	3	2^2	5	6	7	2^3	9	10	11	12	13	14	15	2^4	17	18	$\cdots$
		p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	p_5	d_{12}	d_{13}	$\cdots$
Bits de parité	p_1	x		x		x		x		x		x		x		x		x		$\cdots$
	p_2		x	x			x	x			x	x			x	x			x	$\cdots$
	p_3				x	x	x	x					x	x	x	x				$\cdots$
	p_4								x	x	x	x	x	x	x	x				$\cdots$
	p_5																x	x	x	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

TABLE 1 – Répartition et symboles contrôlés par les bits de parité du code de Hamming

$d_1 \dots d_{2^r-r-1}$ est le message initial et $p_1 \dots p_r$ sont les symboles ajoutés. Le mot du code est alors la deuxième ligne du tableau : $p_1p_2d_1p_3d_2d_3 \dots$. Les croix indiquent les bits contrôlé par chaque symbole de contrôle.

Proposition 14 :
Le code de Hamming est linéaire de distance minimale 3.

Démonstration : La somme de deux bits de contrôle correspond à la parité de la somme des symboles qu'ils contrôlent, ce qui justifie la linéarité (dans $\mathbb{F}_2$, la multiplication externe est par 1 ou par 0). On peut donc appliquer le résultat de la proposition 5) :

$$d_{\min} = \min_{c \in \mathcal{C} \setminus \{0\}} w(c)$$

Soit $A \in \mathbb{F}_2^k$ un message non nul. On pose $B \in \mathbb{F}_2^n$ le message codé de A par le code Hamming (n, k).

Cas 1 : A possède un nombre supérieur ou égal à 3 de bits non nuls
Alors $w(B) \geq 3$
Cas 2 : A possède deux bit non nuls b_1 et b_2
Il existe au moins 1 bit de parité ne codant pas pour b_1 et pour b_2.
Ainsi, $w(B) \geq 3$
Cas 3 : A possède 1 unique bit b non nul
Alors, il existe α et $1 \leq \beta \leq 2^{\alpha-1}$ tel que : b soit le $(2^\alpha + \beta)^{ieme}$ bit du message codé. Donc, il y a au moins deux bits de parité codant pour ce bit.
Ainsi, $w(B) \geq 3$
Donc,

$$\min_{c \in \mathcal{C} \setminus \{0\}} w(c) = 3$$

Le code de Hamming est donc de distance minimale 3.

□

Proposition 15 :
Le code de Hamming est un code parfait.

Démonstration : Pour le code de Hamming (7,4), on a :

$$V_1 = \sum_{i=0}^{1} \binom{7}{i} (2-1)^i = 1 + 7 = 8$$

De plus :

$$\frac{2^7}{V_1} = \frac{128}{8} = 16$$

et

$$\#C = \#\mathbb{F}_2^4 = 2^4 = 16$$

On à le cas d'égalité dans la majoration de Hamming dans d'après la proposition 7 le code de Hamming est un code parfait. □

Proposition 16 :
Tout code binaire linéaire parfait de distance minimale 3 est un code de Hamming

Démonstration : Soit $\mathcal{C}(n,k)$ un code binaire linéaire parfait tel que : $d_{min}(\mathcal{C}) = 3$.
Soit $c \in \mathcal{C}$ et $a \in \{0,1\}^n$ tel que $d(c,a) \leq 1$.
La distance entre 2 mots du code est supérieure ou égale à 3. Donc il n'existe pas d'autre mot c' du code tel que $d(c',a) \leq 1$.
$\mathcal{C}$ est donc 1-correcteur parfait.
Ainsi, $V_t \times \#C = (n+1) \times \#C = 2^n$.
$(n+1)$ est un diviseur de 2^n, donc il existe r tel que : $n+1 = 2^r$, soit $n = 2^r - 1$. Or, entre 1 et $2^r - 1$, il y a r puissances de 2.
On a donc :

$$\begin{cases} n &= 2^r - 1 \\ k &= 2^r - r - 1 \end{cases}$$

$\mathcal{C}$ est donc un code de Hamming. □

3.2 Encodage : l'exemple du code de Hamming (7,4)

Codage du message : On pose A la matrice du message telle que :

$$A = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \text{ avec } (d_1, d_2, d_3, d_4) \in \mathbb{F}_2^4$$

On a :

$$\text{enc} : \begin{cases} \mathbb{F}_2^4 & \to & \mathbb{F}_2^7 \\ d_1d_2d_3d_4 & \mapsto & d_1d_2d_3d_4p_1p_2p_3 \end{cases}$$

avec p_1 la parité de d_1, d_2 et d_6, p_2 la parité de d_1, d_3 et d_4, etp_3 la parité de d_2, d_3 et d_4

La matrice génératrice du code est G telle que :

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Le message codé est la matrice B : $B = G \cdot A$
On a donc :

$$B = \begin{bmatrix} d_1 + d_2 + d_4 \\ d_1 + d_3 + d_4 \\ d_1 \\ d_2 + d_3 + d_4 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ d_1 \\ p_3 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$$

Vérification des erreurs : On pose H la matrice de contrôle :

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

(il s'agit en fait de la table 1).

On note D le message reçu. Si $H \cdot D = 0$, alors il n'y a pas d'erreur.

Démonstration : En effet, on a :

$$H \cdot B = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} . \begin{bmatrix} p_1 \\ p_2 \\ d_1 \\ p_3 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} p_3 + d_2 + d_3 + d_4 \\ p_2 + d_1 + d_3 + d_4 \\ p_1 + d_1 + d_2 + d_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

□

En cas d'erreur, on a $D = B + E_i$, avec $i \in [\![1,\ 7]\!]$, i correspondant à ligne où se situe le bit altéré. Ainsi, $H \cdot D = H \cdot E_i$. S'il n'y a qu'une seule erreur, on peut donc détecter où elle se situe et la corriger.

Démonstration : Avec la matrice de contrôle, il est possible de déceler quel bit a été altéré. Par exemple, pour E_1 on a :

$$H \cdot E_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

De la même façon, on obtient :

$$H \cdot E_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, H \cdot E_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, H \cdot E_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, H \cdot E_5 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, H \cdot E_6 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, H \cdot E_7 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

On remarque que $H \cdot E_i$ correspond à l'écriture binaire de i.
On peut donc corriger le signal reçu. □

Si deux erreurs sont commises, on peut détecter leur existence mais il est impossible de les corriger.

Démonstration : Soit $(i, j) \in [\![1,\ 7]\!]^2$ tel que $i \neq j$, alors : $E_i + E_j \in \{E_1, ..., E_7\}$
L'erreur peut donc être détectée. Cependant, il est alors impossible de corriger ces erreurs, ni même de savoir qu'il y a eu deux erreurs lors de la transmission, d'où l'intérêt de l'ajout d'un 8ème bit contrôlant la parité de l'ensemble de l'octet. □

Décodage : La matrice de décodage est R tel que :

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Elle permet d'extraire les bits d'information, en oubliant les autres, d'où la forme de pseudo matrice identité. Le message reçu corrigé est égal à B. Et on a bien :

$$R \cdot B = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ d_1 \\ p_3 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = A$$

4 Codes de Reed-Solomon

Paramètres : On prendra pour alphabet un corps fini de caractéristique q premier : $\mathbb{F}_{q^m}$. Le code de Reed-Solomon est défini par deux paramètres $(n, k) \in \mathbb{N}^2$:

— k est la longueur du message initial
— n est la longueur du mot du code (message et redondance)

On pose de plus $p = n-k$ le nombre de symboles de contrôle, et m tel que $q^m - 1 = n$. Ces paramètres doivent vérifier :

$$n = k + p = q^m - 1$$

Le message de longueur n est constitué de k symboles d'information et de p symboles de contrôle, d'où $n = k + p$. L'égalité $n = q^m - 1$ est nécessaire pour le décodage.

Le message est un élément de $\mathbb{F}_{q^m}^k$. Dans la suite, on identifiera les listes et les polynômes. Ainsi la liste $(a_0, \ldots, a_k) \in \mathbb{F}_{q^m}^k$ sera vue comme $a_0X^0 + a_1X^1 + \ldots + a_kX^k$, le message est donc un élément de $\mathbb{F}_{q^m,k}[X]$.

Convention : dans cette partie, toutes les fonction présentées prennent implicitement en argument les paramètres du code $(n, k, p = n - k)$ ainsi que l'élément générateur α de $(\mathbb{F}_{q^m}^*, \times)$ choisi. On emploiera les opérations suivantes :

— sur les ints : les opérations usuelles $+$, $-$, $*$ et la division entière //.
— sur les éléments de $\mathbb{F}_{q^m}$: les opérations usuelles $+$, $-$, $*$, l'exponentiation, et la division / (produit par l'inverse) et deux fonctions :
  `log_a` $(\beta) \mapsto i \in [\![1,\ 2^m - 1]\!]$ avec $\alpha^i = \beta$
  `exp_a` $(i) \mapsto \alpha^i$

— sur les polynômes (à coefficient dans $\mathbb{F}_{q^m}$) : les opérations usuelles $+$, $-$, $*$ et les fonctions
`divise` $(A, B) \to Q, R$ réalisant la division euclidienne
`evalue` $(A, \beta) \mapsto A(\beta)$
`prod_scalaire` $(A, \beta) \mapsto A * \beta$
`degre` $(A) \mapsto \deg(A)$
A[`i`] accès en lecture et écriture au i-ième coefficient

De plus X représentera toujours l'indéterminée polynômiale et les opérations $*, +, -, /, \%$ représentent les opérations usuels sur les entiers, les polynômes ou les éléments de $\mathbb{F}_{q^m}$ selon le contexte.

4.1 Encodage

Il existe α un générateur de $\mathbb{F}_{q^m}$. Fixons un tel générateur dans toute la suite. On définit le polynôme générateur par

$$G = \prod_{i=1}^{p}(X - \alpha^i)$$

L'encodage consiste à transformer A en multiple de G, de manière à avoir p racines connues.

Fonction 1 – code : encodage de Reed-Solomon

Donnees : A message, polynôme à coefficients dans $\mathbb{F}_{q^m}$ de degré au plus k
Resultat : C le mot du code correspondant à A
Algorithme :

```
G ← ∏_{i=1}^{p} (X − α^i)
T ← A*X^p
_, B ← divise(T, G)
renvoie T − B
```

Proposition 17 : (Propriétés du code de Reed-Solomon)
Notons $\mathcal{C}$ l'ensemble des mots du code. On a alors :

1. le code est l'ensemble des multiples de G de degré inférieur à n : $\mathcal{C} = \{QG, Q \in \mathbb{F}_{q^m,k}[X]\}$.
2. Le message d'origine d'un mot du code est la liste des ses coefficients d'ordre n à p (ce qui assure l'injectivité de l'encodage)
3. $\mathcal{C}$ est un code linéaire : c'est un sous-espace vectoriel de dimension k de $\mathbb{F}_{q^m,n}[X]$
4. $\mathcal{C}$ a pour distance minimale $d_{\min} = p + 1$

Le code de Reed-Solomon est donc un code linéaire $(n,\, k,\, n-k+1)_{q^m}$

Démonstration : montrons (1) par double inclusion :
Soit A un message. en reprenant les notations du code, il existe Q tel que $T = GQ + B$, donc $T - B = GQ$ De plus on a $\deg B < \deg C = p$, donc la soustraction $T - B$ ne modifie par les $n - p = k$ premier coefficients de T, qui sont ceux de A translaté de p (ce qui montre (2))
Soit C un multiple de G de degré inférieur ou égal à n. Notons D le résultat de l'encodage du polynôme $c_nX^k + c_{n-1}X^{k-1} + \ldots + c_p$.
On a $\deg(C-D) \leqslant \max\{\deg(D), \deg(C)\} \leqslant n$. or $\forall i \in [\![n,\, p]\!]\,,\ d_i = c_i$, donc $\deg(C-D) \leqslant p-1$.
$C - D$ étant un multiple de G et $\deg(G) = p$, on a $C - D = 0$, d'où $C = D$.

(3) Soit $f = \left\{ \begin{array}{ccc} \mathbb{F}_{q^m,k}[X] & \to & \mathbb{F}_{q^m,n}[X] \\ P & \mapsto & PG \end{array} \right.$. f est linéaire par linéarité du produit polynomial, ce qui justifie que $\mathrm{Im}(f) = \mathcal{C}$ est un sous-espace vectoriel de $\mathbb{F}_{q^m,n}[X]$. De plus pour $P \in \mathbb{F}_{q^m,k}[X]$,

$f(P) = 0 \Leftrightarrow P = 0$ car $G \neq 0$. donc $\ker(f) = \{0\}$. f est injective. $f^{|\mathcal{C}}$ est donc un isomorphisme, d'où $\dim \mathcal{C} = \dim \mathbb{F}_{q^m,k}[X] = k$.

(4) Le code étant linéaire, on a $d_{\min} = \min_{c \in \mathcal{C}\setminus\{0\}} w(c)$ d'après la proposition 5

Soit C un mot du code non nul. Supposons que C possède au plus p coefficients non nuls. Alors il existe $(i_1, \ldots, i_p) \in [\![0, n]\!]^p$ et $(c_1, \ldots, c_p) \in \mathbb{F}_{q^m}^p$ tel que $i_1 < \ldots < i_p$ et $C = \sum_{\ell=1}^{p} c_\ell X^{i_\ell}$. On sait que les $(\alpha^j)_{j \in [\![1,\, p]\!]}$ sont des racines de ce polynôme, d'où $\forall s \in [\![1,\, p]\!]\,,\ C(\alpha^s) = \sum_{\ell=1}^{p} c_\ell (\alpha^s)^{i_\ell} = 0$

On a donc le système linéaire :

$$\begin{bmatrix} (\alpha^{i_1})^1 & \cdots & (\alpha^{i_p})^1 \\ \vdots & \ddots & \vdots \\ (\alpha^{i_1})^p & \cdots & (\alpha^{i_p})^p \end{bmatrix} \cdot \begin{bmatrix} c_{i_1} \\ \vdots \\ c_{i_p} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

La matrice des $\left((\alpha^{i_\ell})^k\right)$ est un matrice de Vandermonde. Or les $(\alpha^{i_\ell})_{\ell \in [\![1,\, p]\!]}$ sont tous non nuls et deux-à-deux distincts (en tant que puissances distinctes du générateur de $\mathbb{F}_{q^m}$ tout comprises dans $[\![0,\, q^m - 1]\!]$). La matrice est alors inversible. Son noyau est alors réduit à la matrice nulle, et on a ${}^t\begin{bmatrix} c_{i_1} & \cdots & c_{i_p} \end{bmatrix} = {}^t\begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}$, C est le mot nul ce qui est absurde. C possède donc au moins $p+1$ coefficients non-nuls, et $d_{\min} \geqslant p+1$
Par ailleurs, $1 \in \mathbb{F}_{q^m,k}[X]$ donc $G \in \mathcal{C}$. On sait que G est de degré p, il possède alors au plus $p+1$ coefficient non-nuls, d'où $d_{\min} \leqslant p+1$. Par double inégalité, $d_{\min} = p+1$ □

4.2 Détection d'erreurs et décodage

Détection d'erreurs : Le code de Reed-Solomon peut détecter les erreurs si moins de p erreurs se sont produites. En effet, le nombre de coefficients distincts de deux multiples de G est p, donc si moins de p erreurs se sont produites, alors le polynôme reçu D n'est pas un multiple de G. Vérifier les erreurs revient donc à calculer les $D(\alpha^i)$ pour $i \in [\![1,\, p]\!]$, si l'un d'entre eux est non nul, alors D n'est pas un multiple de G, il y a eu erreur. Sinon on peut directement décoder :

Fonction 2 – decode : décodeur de Reed-Solomon

Donnees : D message reçu, polynôme à coefficients dans $\mathbb{F}_{q^m}$ de degré au plus m
Resultat : A le message original
Algorithme :

```
erreur ← faux
Pour i allant de 1 a p faire
|    Si evalue(D, exp_a(i)) ≠ 0 alors
|    |    erreur ← vrai
Si erreur alors
|    D ← corrige(D)
renvoie la liste des coefficients de D d'ordre compris entre p et n
```

Correction d'erreur : s'il y a erreur, il existe $E \in \mathbb{F}_{q^m,n}[X]$ tel que $D = C + E$ avec C le message d'origine et D le message reçu. On suppose que E possède au plus $t = \frac{p}{2}$ coefficients non nuls. S'il en possède plus, le mot du code le plus proche n'est pas C, la correction sera erronée.

4.3 Correction d'erreurs par syndrome

Inconnues : on cherche le polynôme $E \in \mathbb{F}_{q^m,n}[X]$, on a supposé que E possède au plus $t = \frac{p}{2}$ coefficients non nuls. Il existe $v \leq t$ tel que v soit le nombre d'erreur. Il existe donc $(i_1, \ldots, i_v) \in [\![1, n]\!]^v$ deux-à-deux distincts et $(e_{i_1}, \ldots, e_{i_s}) \in \mathbb{F}_{q^m}^s$ éventuellement nuls tel que

$$E = \sum_{k=1}^{v} e_{i_k} X^{i_k}$$

Soit $j \in [\![1, p]\!]$, on connait $D(\alpha^j) = C(\alpha^j) + E(\alpha^j) = E(\alpha^j)$. Notons s_j cette quantité, nommé le j-ième syndrome. On a $s_j = \sum_{k=1}^{v} e_{i_k} \alpha^{j+i_k}$

Les $(S_j)_{j \in [\![1,\, p]\!]}$ constituent un système (non-linéaire) de p équations avec $2v$ inconnues.

Notations : Définissons les trois polynômes suivant :

- le polynôme syndrome : $S(X) = \sum_{k=1}^{2t} s_k X^{k-1}$
- le polynôme des positions d'erreur : $\Lambda(X) = \prod_{k=1}^{v} (1 - \alpha^{i_k} X)$.
- le polynôme d'évaluation d'erreur $\Omega(X) = \sum_{k=1}^{v} e_{i_k} \alpha^{i_k} \prod_{h=1,\, h \neq k}^{v} (1 - \alpha^{i_k} X)$

Déterminer les positions d'erreur revient à calculer le logarithme en base α des inverses des racines de Λ. Seul le polynôme syndrome est directement connu.

Structure générale de la correction : Le décodage et correction d'un polynôme D s'effectue alors ainsi :

1. On calcule les syndromes, s'ils sont tous nuls, il n'y a pas d'erreur et on décode, dans le cas contraire :
2. On détermine les polynômes Λ et Ω grâce à l'algorithme d'Euclide
3. On détermine les position d'erreurs $(i_k)_{k \in [\![1,\, p]\!]}$ à partir des racines de Λ
4. On détermine les valeurs d'erreurs $(e_{i_k})_{k \in [\![1,\, p]\!]}$ en évaluant judicieusement Λ et Ω
5. On construit le polynôme d'erreur et on le soustrait à D.

La suite vise à détailler les algorithmes utilisés lors des étapes 2, 3 et 4.

4.3.1 Déterminer les polynômes Λ et Ω

Proposition 18 : (Équation-clé)
Les polynômes Λ et Ω vérifient :

1. $S(X)\Lambda(X) \equiv \Omega(X)\ [X^{2t}]$
2. $\Lambda(X) \wedge \Omega(X) = 1$
3. $\forall (A, B) \in \mathbb{F}_{q^m}[X], \left\{ \begin{array}{l} \deg(A) \leqslant t \text{ et } \deg(B) < t \\ S(X)A(X) \equiv B(X)\ [X^{2t}] \end{array} \right. \Rightarrow \exists C \in \mathbb{F}_{q^m}[X],\ A = C\Lambda \text{ et } B = C\Omega$

Démonstration : pour l'équation (1), on a :

$$\begin{aligned} S(X) &= \sum_{k=1}^{2t} s_k X^{k-1} = \sum_{k=1}^{2t}\sum_{h=1}^{v} e_{i_h}\alpha^{ki_h}X^{k-1} = \sum_{h=1}^{v} e_{i_h}\alpha^{i_h}\left(\sum_{k=1}^{2t}(\alpha^{i_h}X)^{k-1}\right) \\ &= \sum_{h=1}^{v} e_{i_h}\alpha^{i_h}\frac{1-(\alpha^{i_h}X)^{2t+1}}{1-\alpha^{i_h}X} \\ \text{donc } \Lambda(X)S(X) &= \sum_{h=1}^{v} e_{i_h}a^{i_h}(1-(\alpha^{i_h}X)^{2t+1})\prod_{k=1,\,k\neq h}^{v}(1-\alpha^{i_k}X) \\ &= \sum_{h=1}^{v}\left(e_{i_h}\alpha^{i_h}\prod_{k=1,\,k\neq h}^{v}(1-\alpha^{i_k}X)\right) - X^{2t+1}\sum_{h=1}^{v}\left(e_{i_h}\alpha^{i_h+i_h+2t+1}\prod_{k=1,\,k\neq h}^{v}(1-\alpha^{i_k}X)\right) \\ &\equiv \Omega(X)\ [X^{2t}] \end{aligned}$$

Par ailleurs, pour $(k,h) \in [\![1,\ v]\!]^2$ tel que $h \neq k$ on a $i_h \neq i_k$ or $(i_h, i_k) \in [\![0,\ q^m]\!]$, on sait donc que $i_h - i_k \not\equiv 0\ [q^m]$, c'est-à-dire que $\alpha^{i_h - i_k} \neq 1$. On a alors $\Omega(\alpha^{i_k}) = e_{i_k}a^{i_k}\prod_{h=1,\,k\neq h}^{v}(1-\alpha^{i_h-i_k}) \neq 0$.

Les racines de Λ ne sont pas racines de Ω. Par ailleurs, $\deg(\Omega) \leqslant v-1 < v$ et $\deg(\Lambda) = v$ donc $\Omega \wedge \Lambda = 1$

(3) soit $(A,B) \in \mathbb{F}_{q^m}[X]$ tel que $\deg(A) \leqslant t$, $\deg(B) < t$ et $AS \equiv B\ [X^2t]$. On a alors modulo $2t$: $\Omega A \equiv \Lambda S A \equiv \Lambda B$. Le polynôme $P = \Omega A - \Lambda B$ est divisible par X^{2t}.
Or $\deg P \leqslant \max\{\deg(A)+\deg(\Omega), \deg(B)+\deg(\Lambda)\} \leqslant \max\{t+(t-1),(t-1)+t\} < 2t$.
Donc $P = 0$, c'est-à-dire $\Omega A = \Lambda B$.
Λ divise ΩA et $\Lambda \wedge \Omega = 1$ donc Λ divise A. Il existe $C \in \mathbb{F}_{q^m}[X]$, $A = \Lambda C$. On a $\Lambda C\Omega = \Lambda B$ or $\Lambda \neq 0$, d'où $B = \Omega C$. □

Algorithme d'Euclide étendu : définissons les suites polynômiales d'Euclide (P_n) et (Q_n) par

$$\begin{cases} P_0 = X^{2t} \text{ et } P_1 = S \\ \forall n \geqslant 1, \left| \begin{array}{ll} \text{si } P_n \neq 0, & P_{n-1} = Q_n P_n + P_{n+1} \text{ et } \deg(P_{n+1}) < \deg(P_n) \\ \text{si } P_n = 0, & P_{n+1} = 0 \end{array} \right. \end{cases}$$

$(\deg(P_n))_{n\in\mathbb{N}}$ étant une suite d'entiers strictement décroissante, on a $P_n \longrightarrow 0$. On notera n_f le premier entier tel que $P_{n_f} = 0$ et $P_{n_f-1} \neq 0$. On pose alors les suites polynomiales

$$(U_n) : \begin{cases} U_0 = 1 \text{ et } U_1 = 0 \\ \forall n \in [\![1,\ n_f]\!], U_{n+1} = U_{n-1} - Q_n U_n \end{cases} \quad \text{et} \quad (V_n) : \begin{cases} V_0 = 0 \text{ et } V_1 = 1 \\ \forall n \in [\![1,\ n_f]\!], V_{n+1} = V_{n-1} - Q_n V_n \end{cases}$$

Proposition 19 :
Avec les notations précédentes, on a

1. $\forall n \in [\![0,\ n_f]\!]$, $P_n = U_n P_0 + V_n P_1$
2. $\forall n \in [\![1,\ n_f]\!]$, $\deg(V_n) \leqslant \deg(P_0) - \deg(P_{n-1})$
3. $\forall n \in [\![1,\ n_f]\!]$, $\deg(U_n) \leqslant \deg(P_1) - \deg(P_{n-1})$

Démonstration : prouvons ces propositions par récurrence à deux pas sur $n \in [\![1,\ n_f]\!]$.
Initialisation : $n = 1$.
(1) : $P_1 = 0 \times P_0 + 1 \times P_1$
(2) : $V_1 = 1$ donc $\deg(V_1) = 0 \leqslant \deg(P_0) - \deg(P_0)$
(3) : $U_1 = 0$ donc $\deg(U_1) = -\infty \leqslant \deg(P_1) - \deg(P_0)$

$n = 2$:

(1) : $P_2 = P_1Q_1 - P_0 = (U_1P_0 + V_1P_1)Q_1 - P_0 = (V_1Q_1 - V_0)P_1 + (U_1Q_1 - U_0)P_0 = V_2P_1 + U_2P_0$

(2) : $V_2 = V_0 - Q_1V_1 = -Q_1$ donc $\deg(V_1) = \deg(Q_1) = 1 \leqslant 2t - (2t-1) = \deg(P_0) - \deg(P_1)$.

(3) : $U_2 = U_0 - Q_1U_1 = 1$ donc $\deg(U_2) = 0 \leqslant 2t - 1 - (2t-1) = \deg(P_1) - \deg(P_1)$

Hérédité : soit $n \in [\![3,\ n_f]\!]$, supposons que (1) (2) et (3) soit vraie au rangs $n-1$ et $n-2$.

(1) : $P_n = P_{n-1}Q_{n-1} - P_{n-2} = (V_{n-1}P_1 + U_{n-1}P_0)Q_1 - (V_{n-2}P_1 + U_{n-2}P_0)$
$= (V_{n-1}Q_{n-1} - V_{n-2})P_1 + (U_{n-1}Q_{n-1} - U_{n-2})P_0$
$= V_nP_1 + U_nP_0$

On a $Q_{n-1}P_{n-1} = P_{n-2} - P_n$ donc $\deg(Q_{n-1}) \leqslant \max\{\deg(P_{n-2}), \deg(P_n)\} - \deg(P_{n-1})$
$\leqslant \deg(P_{n-2}) - \deg(P_{n-1})$ car $\deg(P_n) \leqslant \deg(P_{n-2})$

(2) : $V_n = V_{n-1}Q_{n-1} - V_{n-2}$ d'où la majoration :
$\deg(V_n) \leqslant \max\left\{\deg(V_{n-1}) + \deg(Q_{n-1}),\ \deg(V_{n-2})\right\}$
$\leqslant \max\{\deg(P_0) - \deg(P_{n-2}) + \deg(P_{n-2}) - \deg(P_{n-1}),\ \deg(P_0) - \deg(P_{n-3})\}$
$\leqslant \deg(P_0) - \deg(P_{n-1})$ car $-\deg(P_{n-1}) \geqslant -\deg(P_{n-3})$

(3) : $U_n = U_{n-1}Q_{n-1} - U_{n-2}$ on procède donc de même que pour (2)
$\deg(U_n) \leqslant \deg(P_1) - \deg(P_{n-1})$

Ce qui achève l'hérédité. (1), (2) et (3) sont vrai pour tout $n \in [\![1,\ n_f]\!]$. De plus (1) est vrai pour $n = 0$ car $P_0 = 0 \times P_1 + 1 \times P_0$ □

Proposition 20 : (Résolution de l'équation-clé)
Il existe $n_0 \in [\![2,\ n_f]\!]$ vérifiant $\deg(P_{n_0}) \leqslant t$ et $\forall n \in [\![0,\ n_0 - 1]\!], \deg(P_n) > t$. On a alors :

$$V_{n_0}(0) \neq 0 \qquad \Lambda(X) = \frac{V_{n_0}(X)}{V_{n_0}(0)} \qquad \Omega(X) = \frac{P_{n_0}(X)}{V_{n_0}(X)}$$

Démonstration : on sait que $P_n \longrightarrow 0$ donc $\deg(P_n) \longrightarrow -\infty$.
Par ailleurs $\deg(P_1) = \deg(S) = 2t - 1 > t$, et $\deg(P_{n_f}) = -\infty$ ce justifie l'existence de $n_0 \in [\![2,\ n_f]\!]$ vérifiant les propriétés de degré.
$P_{n_0} = U_{n_0}X^{2t} + V_{n_0}S$ d'après la proposition 19.1. Donc $P_{n_0} \equiv V_{n_0}S[X^{2t}]$.
Or $\begin{cases} \deg(V_{n_0}) \leqslant \deg(P_0) - \deg(P_{n_0-1}) \leqslant 2t - (t+1) < t \\ \deg(P_{n_0}) \leqslant t \end{cases}$.
On peut donc appliquer la proposition 18.3 :

$$\text{il existe } C \in \mathbb{F}_{q^m}[X] \text{ tel que } P_{n_0} = C\Omega \text{ et } V_{n_0} = C\Lambda$$

En passant au degré : $\deg(V_{n_0}) = \deg(C) + \deg(\Lambda)$ donc $\deg(C) + t \leqslant t$, donc $\deg(C) \leqslant 0$ ce qui justifie que C est une constante. C est non-nulle car $C\Lambda = V_{n_0} \neq 0$
$\Lambda(0) = 1$ donc $C = V_{n_0}(0)$. D'où le résultat : $\Lambda(X) = \dfrac{V_{n_0}(X)}{V_{n_0}(0)}$ et $\Omega(X) = \dfrac{P_{n_0}(X)}{V_{n_0}(X)}$ □

Fonction 3 – euclide : résolution de l'équation-clé

```
Donnees : D Le polynôme de F_{q^m,n}[X] reçu contenant au moins une erreur
Resultat : Lambda, Omega les polynômes des positions et évaluations d'erreurs
Algorithme :
# calcul des syndromes
    S ← polynôme 0 de F_{q^m}[X]
    Pour i allant de 1 a p faire
    |    S[i] ← evalue(D, exp_a(i))
# algorithme d'euclide
    P0 ← polynôme X^{2t} de F_{q^m}[X]
    P1 ← S
    V0 ← polynôme 0 de F_{q^m}[X]
    V1 ← polynôme 1 de F_{q^m}[X]
    t ← (n-k) // 2
    Tant que degre(P1) > t faire
    |    Q, R ← divise(P0, P1)
    |    P0, P1 ← P1, R
    |    V0, V1 ← V1, V0 - Q*V1
    C ← 1_{F_{q^m}} / evalue(V1, 0)
    renvoie prod_scalaire(V1, C), prod_scalaire(P1, C)
```

Complexité : cet algorithme a également l'avantage d'être rapide. En effet, la suite $\big(\deg(P_n)\big)$ étant strictement décroissante. On peut majorer le nombre d'itérations de l'algorithme d'Euclide ci-dessus par t. Euclide réalise donc au plus t divisions, produits et sommes de petits polynômes (tous les polynômes ont un degré majoré par $2t$).

4.3.2 Déterminer les positions d'erreurs

Détermination des racines : Λ étant connu, il suffit de calculer ses racines, les α^{-i_k} pour connaître les positions d'erreur. Comme on est dans un corps fini ayant un petit nombre d'élément, il suffit donc de faire une recherche exhaustive sur les $q^m - 1 = n$ éléments du groupe $(\mathbb{F}_{q^m}^*, \times)$.

Une fois les racines α^{-i_k} connues, on calcule leur logarithme en base α, pour obtenir $-i_k$ (modulo $[q^m - 1]$), le logarithme renvoyant une valeur positive entre 0 et $q^m - 1$. On a donc $i_k = q^m - 1 - \texttt{log\_a}(\alpha^{-i_k}) = n - \texttt{log\_a}(\alpha^{-i_k})$

Fonction 4 – calcul_racines : calcule des racines de Λ

```
Donnees : Lambda le polynôme de positions d'erreur
Resultat : racines liste des i_k, les puissances des inverses des racines de Lambda
Algorithme :
    racines ← liste d'ints de longueur degre(Lambda)
    k ← 0
    Pour i allant de 0 a n faire
    |    Si evalue(Lambda, exp_a(i)) = 0 alors
    |    |    racines[k] ← n-i
    |    |    k ← k + 1
    renvoie racines
```

4.3.3 Déterminer les valeurs d'erreurs

Proposition 21 : (Formule de Forney)
Étant donné les polynômes Λ et Ω ainsi que les positions d'erreur $(i_k)_{k\in[\![1,\,v]\!]}$, on a

$$\forall k \in [\![1,\,v]\!]\,,\; e_{i_k} = -\frac{\Omega(\alpha^{-i_k})}{\Lambda'(\alpha^{-i_k})}$$

avec $\Lambda'(X) = \sum_{k=1}^{v} k\cdot\lambda_k X^{k-1}$ est la dérivé de Λ ($\cdot$ désigne l'opérateur additif itéré et non l'opérateur produit)

Démonstration : soit $k \in [\![1,\,v]\!]$, on a $\Lambda'(X) = \sum_{h=1}^{v} -\alpha^{i_h} \prod_{j=1,\,j\neq h}^{v} (1-\alpha^{i_j}X)$ donc

$$\frac{\Omega(\alpha^{-i_k})}{\Lambda'(\alpha^{-i_k})} = \frac{\sum_{h=1}^{v} e_{i_h}\alpha^{i_h} \prod_{j=1,\,j\neq h}^{v} (1-\alpha^{i_j}\alpha^{-i_k})}{\sum_{k=1}^{v} -\alpha^{i_h} \prod_{j=1,\,j\neq h}^{v} (1-\alpha^{i_j}\alpha^{-i_k})} = \frac{e_{i_k}\alpha^{i_k} \prod_{h=1,\,h\neq k}^{v} (1-\alpha^{i_h}\alpha^{-i_k})}{-\alpha^{i_k} \prod_{h=1,\,h\neq k}^{v} (1-\alpha^{i_h}\alpha^{-i_k})}$$

$$\frac{\Omega(\alpha^{-i_k})}{\Lambda'(\alpha^{-i_k})} = \frac{e_{i_k}\alpha^{i_k}}{-\alpha^{i_k}} = -e_{i_k}$$

□

Calcul des coefficients de E : il suffit simplement d'appliquer cette formule pour calculer un par un les coefficients non-nuls de E :

Fonction 5 – Forney : calcul des coefficients e_{i_k}

Donnees : `racines` liste des positions d'erreur i_k
`Lambda`, `Omega` les polynômes des positions et évaluations d'erreurs
Resultat : `coeff` liste des coefficients e_{i_k} (dans le même ordre que leurs indices dans `racines`)
Algorithme :

```
coeff ← liste d'éléments de F_{q^m} de longueur len(racines)
d_Lambda ← derive(Lambda)
Pour i allant de 0 a len(coeffs) faire
    O ← evalue(Omega, exp_a(n-i))
    L ← evalue(d_Lambda, exp_a(n-i))
    coeff[i] ← - O / L
renvoie coeff
```

4.3.4 Synthèse de la correction

Méthode de correction : pour corriger le polynôme $D = C + E$, avec $E = \sum_{k=1}^{v} e_{i_k} X^{i_k}$

- on calcule les syndromes $(S_j)_{j\in[\![1,\,p]\!]}$, s'ils sont tous nuls pas besoin de corriger, sinon
- on calcule Λ et Ω grâce à l'algorithme d'Euclide
- on déterminer les positions d'erreurs $(i_k)_{k\in[\![1,\,v]\!]}$ à partir des racines de Λ
- on calcule les coefficients $(e_{i_k})_{k\in[\![1,\,v]\!]}$ grâce à la formule de Forney.
- on soustrait E à D

La fonction de correction s'écrit alors

Fonction 6 – corrige : correcteur du code de Reed-Solomon

```
Donnees : D polynôme erroné (contient au moins une erreur)
Resultat : C le mot du code le plus proche de D
Algorithme :
    Lambda, Omega ← euclide(D)
    racines ← calcul_racine(Lambda)
    coeff ← Forney(racines, Lambda, Omega)
    E ← polynôme nul de F_{q^m}[X]
    Pour i allant de 1 a len(racines) faire
    |    E[racines[i]] ← coeff[i]
    renvoie D − E
```

4.4 Notes d'implémentation : calcul dans $\mathbb{F}_{2^m}$ et $\mathbb{F}_{2^m}[X]$

Le corps de $\mathbb{F}_{2^m}$: nous avons montré qu'il existe des codes de Reed-Solomon pour n'importe quel corps fini de cardinal q^m avec q premier. Le plus courant étant d'utiliser un corps fini à 2^m éléments : on peut en effet représenter un élément de $\mathbb{F}_{2^m}$ par un entier sur m bits. [01001010] représente $X^6 + X^4 + X^2$ dans $\mathbb{F}_8$. Or dans $\mathbb{Z}/2\mathbb{Z}$, on a $1+1=0$, $1+0=1$, $0+1=1$ et $0+0=0$. L'addition dans $\mathbb{F}_{2^m}$ correspond donc au ou exclusif bit-à-bit.

Pour le produit, on se sert du caractère cyclique du groupe $(\mathbb{F}_{2^m}^*, \times)$. On choisit un générateur a et on précalcule deux tables `exp_a` et `log_a` telle que pour $i \in [\![0, m-1]\!]$, $a^i = \texttt{exp\_a}[i]$ et $i = \texttt{log\_a}[i]$. Ainsi pour $(c, d) \in \mathbb{F}_{2^m}^2$ non-nuls, le produit se lit dans ces tables :

$$c \times d = \texttt{exp\_a}[\texttt{log\_a}[c] + \texttt{log\_a}[b]]$$

L'inverse d'un élément se calcule alors simplement : $c^{-1} = \texttt{exp\_a}[m - 1 - \texttt{log\_a}[c]]$

Ainsi les opérations dans $\mathbb{F}_{2^m}$ sont des opérations élémentaires.

Les polynômes : les polynômes sont représentés par des listes d'éléments de $\mathbb{F}_{2^m}$. L'addition est le ou-exclusif coefficient par coefficient, et le produit est le produit polynomial classique. Finalement, on utilise l'algorithme de division synthétique pour calculer les restes et quotients de polynômes. Ainsi, pour $(P, Q) \in \mathbb{F}_{2^m}[X]$ de degrés respectifs n et m on a les complexités :

Opération	Complexité
$P + Q$	$\mathcal{O}(\max\{n, m\})$
$P \times Q$	$\mathcal{O}(n + m)$
P/Q et $P \bmod Q$	$\mathcal{O}(nm)$

Toutefois, les paramètre du code étant fixé, on étudie plutôt le temps d'exécution avec des paramètres de tailles fixés.

5 Comparaisons de codes correcteurs

Comparaison des codes : on parcourt le mot du code en permutant chaque bit si un nombre aléatoire est inférieur à une certaine probabilité. On trace alors la pourcentage de correction (le mot renvoyé par le décodeur est le même que celui initial) en fonction de la probabilité d'erreur.

On constate sans surprise que les message ayant un ratio $\frac{k}{n}$ plus faible, et donc un plus grand nombre de bits de contrôle par rapport aux bits de message, ont une meilleure correction.

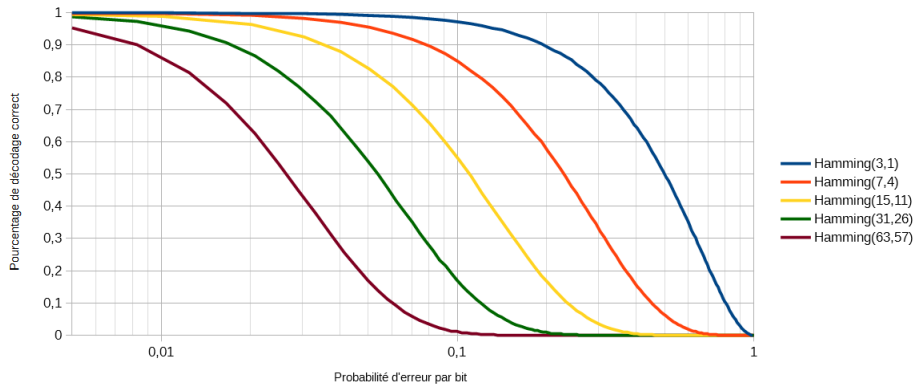

FIGURE 1 – Pourcentage de correction en fonction de la probabilité d'erreur pour le code de Hamming

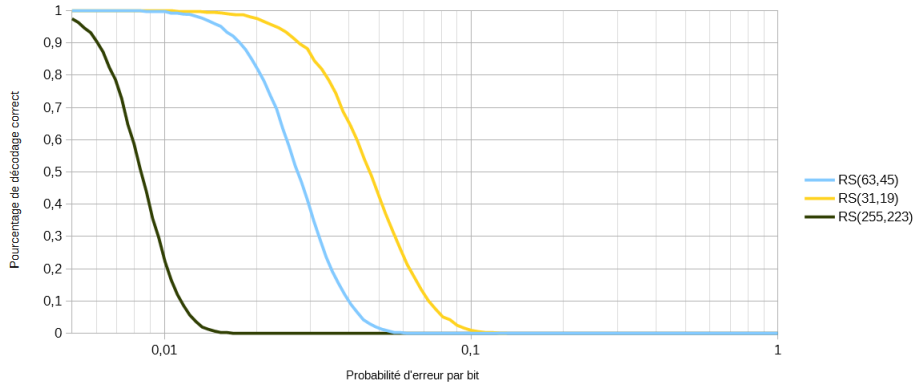

FIGURE 2 – Pourcentage de correction en fonction de la probabilité d'erreur pour le code de Reed-Solomon

En pratique, cette expérience n'est fait qu'avec des erreurs placés de façon aléatoire. Cependant, lors du stockage et de la transmission de données, les bouffées d'erreurs dues au rayures sur un disque ou au bruit du canal sont plus fréquentes. Le code de Reed-Solomon a un bien meilleur comportement vis-à-vis de ces bouffées d'erreur. En effet, le décodeur ne fait par la différence entre l'altération d'un bit du symbole ou de tous les bits du symbole (on travaille souvent sur $\mathbb{F}_{256}$ donc un symbole comprend 8 bits). Le code de Hamming, en revanche, échoue dès qu'il y a deux erreurs dans une même portion de message.

Références

[1] Andrew Brown. Reed-Solomon Encoder and Decoder, 2010. `https://github.com/brownan/Reed-Solomon`, 2015.

[2] Michel Demazure. *Cours d'Algèbre*. Cassini, 2008. chapitres 9-10-11, ISBN 978-2-84225-127-7.

[3] Jonathan I. Hall. Hamming codes. `http://users.math.msu.edu/users/jhall/classes/codenotes/Hamming.pdf`.

[4] Tony Hill. *Reed-Solomon Codes Explained*. mai 2013. `https://downloads.bbc.co.uk/rd/pubs/whp/whp-pdf-files/WHP031.pdf`.

[5] Munsif Jatoi. *Forward Error Correction using Reed- Solomon Coding and Euclid Decoding in Wireless Infrared Communications*. août 2014.

[6] Bruce Maggs. *Decoding Reed-Solomon Codes*. octobre 2000. Duke Computer Science, cours. `www.cs.duke.edu/courses/spring10/cps296.3/decoding_rs_scribe.pdf`.

[7] Priti Shankar. Decoding Reed-Solomon Codes Using Euclid's Algorithm. *Resonance*, avril 2007.

[8] Wikipédia. Code correcteur, 2017. `https://fr.wikipedia.org/wiki/Code_correcteur`.