

## 1 Les codes en blocs

### Définition des codes en blocs

Principe :  $k$  bits  $\xrightarrow[m]{c \in \mathcal{C}}$   $n$  bits avec  $n > k$ . On a  $\dim(\mathcal{C}) = k$ .

**Def.** Rendement :  $R = \frac{k}{n}$ .

**Not.**  $\mathcal{C}(k, n, d_{\min})$ .

**Def.** Capacité de détection :  $t \leq d_{\min} - 1$ . Capacité de correction :  $t \leq \lfloor \frac{d_{\min} - 1}{2} \rfloor$ .

### Les codes linéaires en blocs

$\mathcal{C}$  est un sev de  $\text{GF}(2)^n$ . Alors  $d_{\min} = \min_{c \neq 0} w_H(c)$  (poids de Hamming).

**Def.** Matrice génératrice :  $G = [I_{k \times k} \mid P_{k \times (n-k)}] \in \mathfrak{M}_{k \times n}$  sous forme systématique telle que  $c = m \cdot G = [m \mid n - k \text{ bits de parité}]$ .

**Def.** Matrice de parité :  $H \in \mathfrak{M}_{(n-k) \times n}$  la matrice génératrice de  $\mathcal{C}^\perp$ , donc  $\forall c \in \mathcal{C}, c \cdot H^T = 0$ . Sous forme systématique :  $H = [-P^T \mid I_{n-k}]$ .

**Def.** Vecteur syndrôme :  $s = rH^T$  avec  $r$  le mot reçu. Alors  $s = 0$  ssi  $R$  est un mot de code.

**Th** (Borne de singleton).  $d_{\min} \leq n - k + 1$ , d'où la correction d'erreur  $2t \leq d_{\min} - 1 \leq n - k$ .

On effectue alors un décodage par maximum likelihood : si les éléments de l'alphabet de départ sont équiprobables, on cherche  $\max p(y \mid x)$ .

### Transformations

- Extension : rajouter des bits de parité.
- Allongement : rajouter des bits d'info.
- Perforation : supprimer des bits de parité.
- Raccourcissement : supprimer des bits d'info.
- Augmentation : ajouter des bits d'info sans modifier la longueur.
- Expurgation : supprimer des bits d'info sans modifier la longueur.

## 2 Les codes cycliques

**Def.** Code cyclique : code linéaire en bloc  $\mathcal{C}$  défini sur  $\text{GF}(q)$  tel que  $\forall c = (c_0, \dots, c_{n-1}) \in \mathcal{C}, (c_{n-1}, c_0, \dots, c_{n-2}) \in \mathcal{C}$ .

### Représentation polynomiale

**Def.**  $c(X) = c_0 + c_1X + \dots + c_{n-1}X_{n-1}$

Décalage cyclique :  $Xc(X) \pmod{X^n - 1}$ .

**Def.** Polynôme générateur :  $g(X)$  l'unique mot de code unitaire de degré minimal.

Tous les mots de code sont multiples de  $g(X)$ , et  $g(X) \mid X^n - 1$ . Pour écrire  $c(X) = g(X)m(X)$  de manière unique, avec  $m$  un mot d'info de degré  $< k$ , il faut  $\deg(g(X)) = n - k$ .

**Th.**  $\mathcal{C}$  est un sous-ensemble cyclique de  $\text{GF}(q)[X]/X^n - 1$  ssi  $\mathcal{C}$  est un idéal de  $\text{GF}(q)[X]/X^n - 1$ .

On a le morphisme  $\phi : p(X) \in \text{GF}(q)[X] \rightarrow p(X) \pmod{X^n - 1}$ .  $\phi^{-1}(\mathcal{C})$  est un idéal de  $\text{GF}(q)[X]$ , qui est un corps, donc tous ces idéaux sont principaux et il y a existence et unicité de  $g(X)$ .

### Polynôme de parité

**Def.** Le polynôme de parité est  $h(X) = \frac{X^n - 1}{g(X)}$ .

Pour avoir unicité de l'écriture des mots de code :  $\deg(h) < k$ .

### Forme systématique

$c(X) = m(X)g(X)$  : pour rendre  $m(X)$  visible dans  $g(X)$ , on mettra ses coefficients dans les plus hauts degrés.

Forme souhaitée :  $c(X) = X^{n-k}m(X) + t(X)$ .

Division euclidienne :  $X^{n-k}m(X) = \underbrace{q(X)g(X)}_{\in \mathcal{C}} + r(X)$ . On écrit donc  $c(X) = X^{n-k}m(X) - r(X)$ .

### Polynôme syndrôme

Le mot reçu est  $r(X) = c(X) + e(X)$ , où  $e$  est la représentation polynomiale de l'erreur. Alors  $s(X)$  est égal au reste de  $r(X)/g(X)$  ou de  $e(X)/g(X)$ .

Si  $\deg(e(X)) < \frac{d_{\min}}{2}$  alors  $s(X)$  est unique.

### 3 Rappels sur les corps finis

#### Polynôme sur un corps

Tout polynôme  $P(X)$  défini sur un corps  $F$  peut être factorisé de façon unique en produit de polynômes premiers (irréductible, unitaire, de degré  $> 1$ ).

Soit un anneau quotient  $F[X]/P(X)$ . Si  $p(X)$  est unitaire, c'est l'ensemble des polynômes de degré inférieur à  $P(X)$ . C'est un corps ssi  $P(X)$  est premier, et c'est alors une extension de  $F$ .

#### Construction de $\text{GF}(p^m)$

Soit  $P$  premier dans  $\text{GF}(p)[X]$ , de degré  $m$ . Alors  $\text{GF}(p)[X]/P(X)$  est un corps fini à  $p^m$  éléments.

#### Élément primitif

$\alpha \in \text{GF}(p^m)$  est **primitif** si tout élément de  $\text{GF}(p^m) \setminus \{0\}$  est une puissance de  $\alpha$ . Tout corps fini en possède au moins un.

$(\text{GF}(p^m) \setminus \{0\}, \cdot)$  est un groupe cyclique généré par  $\alpha$ .

**Def.**  $P(X)$  est un polynôme **primitif** ssi il annule un élément primitif.

#### Factorisation de $X^n - 1$ , où $n = p^m - 1$

Soit  $\beta \in \text{GF}(p^m) \setminus \{0\}$ , d'ordre  $r$ . Alors  $\beta^{p^m-1} = (\beta^r)^{\frac{p^m-1}{r}} = 1$  donc  $\beta$  est racine de  $X^n - 1$  ( $r$  divise l'ordre du groupe).

$X^n - 1 = \prod_{\beta \in \text{GF}(p^m) \setminus \{0\}} (X - \beta)$  et on veut factoriser dans  $\text{GF}(p)$  maintenant.

#### Polynôme minimal

**Def.** Le **polynôme minimal** de  $\beta \in \text{GF}(p^m)$  est le polynôme de plus petit degré dans  $\text{GF}(p)$  qui annule  $\beta$ .

**Prop** (de Frobenius).  $\forall q \in \text{GF}(p^m)[X], \forall a \in \mathbb{N}, q(X)^{p^a} = \left[ \sum_{i=0}^{\deg(q)} q_i X^i \right]^{p^a} = \sum_{i=0}^{\deg(q)} q_i^{p^a} X^{ip^a}$ .

**Th.** Si  $f(X)$  est le polynôme minimal de  $\beta \in \text{GF}(p^m)$  alors c'est aussi le polynôme minimal de  $\beta^p$ .

Deux éléments de  $\text{GF}(p^m)$  sont conjugués s'ils ont le même polynôme minimal. Les conjugués de  $\beta$  sont  $\{\beta, \beta^p, \beta^{p^2}, \dots, \beta^{p^{r-1}}\}$  où  $r = \min\{i \in \mathbb{N}^* \mid \beta^{p^i} = \beta\}$ .

Le polynôme minimal de  $\beta$  s'écrit  $f(X) = (X - \beta)(X - \beta^p) \cdots (X - \beta^{p^{r-1}})$ .

### 4 Codes BCH

#### Construction d'un code cyclique

Code cyclique primitif : de longueur  $n = p^m - 1$  avec  $p$  premier.

**Th.** Soit  $\beta_1, \dots, \beta_r$  les racines dans  $\text{GF}(p^m)$  de  $g(X)$ , polynômes générateur d'un code cyclique primitif. Alors  $c(X) \in \text{GF}(p)[X]$  est un mot de code ssi  $\forall i, c(\beta_i) = 0$ . De plus  $g(X) = \text{ppcm}(f_{\beta_1}(X), \dots, f_{\beta_r}(X))$  où les  $f_i$  sont les polynômes minimaux.

On obtient  $g(X)$  en choisissant les racines. On a alors  $k$  via  $\deg(g(X)) = n - k$ .

Les zéros du code sont les  $i$  tels que  $\alpha^i$  est racine de  $g(X)$ , avec  $\alpha$  un élément primitif.

#### Les codes BCH

**Def.** **Code BCH** : code cyclique ayant  $2t$  zéros consécutifs (qui corrige  $t$  erreurs,  $d_{\min} \geq 2t + 1$ ).

Construction d'un code BCH, avec  $n = p^m - 1$ ,  $p$  premier :

- 1) choisir  $p(X)$  premier, de degré  $m$  sur  $\text{GF}(p)[X] \rightarrow \text{GF}(p^m)$ ,
- 2) calculer les polynômes minimaux des  $\alpha^i$  pour  $1 \leq i \leq 2t$ ,
- 3) calculer  $g(X) = \text{ppcm}((f_{\alpha^i})_{i=1, \dots, 2t})$ .

#### Décodage par calcul du syndrome

On a  $R(X) = c(X) + e(X)$ . Le polynôme syndrome  $s(X)$  est le reste de  $R(X)/g(X)$ .

**Def.** **vecteur syndrome** :  $S = (s_1, \dots, s_{2t})$  où  $s_i = R(\alpha^i)$ .

Si  $S = 0$ ,  $R(x)$  est un mot de code. On a  $s_{2i} = R(\alpha^{2i}) = R(\alpha^i)^2 = s_i^2$ , donc il y a de la redondance.

**Ex** (Code de Hamming).  $BCH(2^m - 1, 2^m - 1 - m, 3)$ . On a  $\text{GF}(2^m) = \text{GF}(2)[X]/P(X)$ ,  $g(X) = p(X)$ . Le code corrige une erreur. Algorithme de décodage : calculer  $s_1$ , si  $s_1 = 0$ ,  $e(x) = 0$ , sinon  $s_1 = \alpha^i \neq 0$  et  $e(X) = X^i$ .

**Ex** (Code BCH binaire correcteur de 2 erreurs).  $S = (s_1, s_2, s_3, s_4)$ ,  $s_2 = s_1^2$  et  $s_4 = s_1^4$ . Les composantes non

redondantes sont  $s_1$  et  $s_3$ .

$e(X) = 0$	pas d'erreur	$s_1 = 0$	$s_3 = 0$
$e(X) = X^i$	erreur en position $i$	$s_1 = \alpha^i$	$s_3 = \alpha^{3i}$
$e(X) = X^i + X^j$	erreurs en positions $i$ et $j$	$s_1 = \alpha^i + \alpha^j$	$s_3 = \alpha^{3i} + \alpha^{3j}$

**Def.** **Polynôme localisateur d'erreurs** :  $\Lambda(X)$  dont les racines sont les inverses des positions des erreurs dans  $\text{GF}(p^m)[X]$  (corps localisateur d'erreurs).

Dans l'exemple :  $\Lambda(X) = (1 + \alpha^i X)(1 + \alpha^j X) \in \text{GF}(2^m)[X]$  avec 2 erreurs. En développant on obtient  $\Lambda(X) = 1 + s_1 X + \left(s_1^2 + \frac{s_3}{s_4}\right) X^2$ .

### Transformée de Fourier discrète dans les corps finis

- TDF dans  $\mathbb{C}^n : (h_0, \dots, h_{n-1}) \mapsto (H_0, \dots, H_{n-1})$  avec  $H_k = \sum_{l=0}^{n-1} h_l \exp\left(-\frac{2i\pi k l}{n}\right)$ .
- TDF dans  $\text{GF}(p^m) : \text{soit } \alpha \in \text{GF}(p^m) \text{ tel que } \alpha^n = 1, \text{ on a}$

$$(v_0, \dots, v_{n-1}) \in \text{GF}(p) \longleftrightarrow (V_0, \dots, V_{n-1}) \in \text{GF}(p^m)$$

$$v_i = \frac{1}{n \bmod p} \sum_{j=0}^{n-1} V_j \alpha^{-ij} \quad V_j = \sum_{i=0}^{n-1} v_i \alpha^{ij}$$

Dans le cas  $p = 2$  et  $n = 2^m - 1$ , on a donc  $v_i = \sum_{j=0}^{n-1} V_j \alpha^{-ij}$  et  $V_j = \sum_{i=0}^{n-1} v_i \alpha^{ij}$ .

On définit  $v(X) = \sum_{i=0}^{n-1} v_i X^i$  et  $V(X) = \sum_{j=0}^{n-1} V_j X^j$ . On a  $v_i = V(\alpha^{-i})$  et  $V_j = v(\alpha^j)$ .

### Produit de convolution cyclique :

$$v_i = h_i u_i \xrightarrow{\text{TDF}} V_j = \sum_{i=0}^{n-1} H_i U_{(j-i) \bmod n}$$

$$v_i = \sum_{j=0}^{n-1} h_j u_{(j-i) \bmod n} \xrightarrow{\text{TDF}} V_j = H_j U_j$$

### Technique spectrale de décodage des codes BCH

Vecteur reçu :  $v(X) = c(X) + e(X)$  ( $v_i = c_i + e_i$ ). Syndromes :  $\forall i \in \llbracket 1; 2t \rrbracket, S_i = v(\alpha^i) = e(\alpha^i) = E_i$ .

$$\begin{matrix} e(X) & \xrightarrow{\text{TDF}} & E(X) \\ e_0, \dots, e_{n-1} & & E_0, \dots, E_{n-1} \end{matrix}$$

Or  $(E_1, \dots, E_{2t}) = (S_1, \dots, S_{2t})$  est la fenêtre spatiale sur le motif d'erreur.

Supposons  $\nu$  erreur, avec  $\nu \leq t$ , de positions  $i_k, k \in \llbracket 1; \nu \rrbracket$ .

On a le polynôme localisateur d'erreurs  $\Lambda(X) = \prod_{k=1}^{\nu} (1 + \alpha^{i_k} X) = \sum_{i=0}^{n-1} \Lambda_i X^i$ . On passe à  $\lambda(X) = \sum_{i=0}^{n-1} \lambda_i X^i, \lambda_i = \Lambda(\alpha^i) = \sum_{j=0}^{n-1} \Lambda_j \alpha^{-ij} \in \text{GF}(p)$ .

**Prop.** On a  $\forall i, e_i \lambda_i = 0$ .

*Démonstration.* Si  $i$  n'est pas la position d'une erreur,  $e_i = 0$ , sinon  $i$  est la position d'une erreur, donc  $e_i \neq 0$  mais  $\lambda_i = \Lambda(\alpha^{-i}) = 0$ .  $\square$

Système fondamental de décodage,  $t$  équations avec  $t$  inconnues :

$$\begin{cases} \sum_{j=0}^{n-1} \Lambda_j E_{(k-j) \bmod n} = 0 \\ \forall i \in \llbracket t+1; 2t \rrbracket, \sum_{j=1}^t \Lambda_j S_{k-j} = S_k \end{cases}$$

**Algorithme de PGZ.** Écriture du système sous forme matricielle :

$$\underbrace{\begin{pmatrix} S_1 & S_2 & \dots & S_t \\ S_2 & S_3 & \dots & S_{t+1} \\ \dots & \dots & \dots & \dots \\ S_t & S_{t+1} & \dots & S_{2t} \end{pmatrix}}_{\text{matrice des syndromes}} \cdot \begin{pmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \dots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} S_{t+1} \\ S_{t+2} \\ \dots \\ S_{2t} \end{pmatrix}$$

S'il existe  $t$  erreurs, la matrice des syndromes est inversible, donc le système est résoluble et il existe une solution unique. Sinon on reprend mais en testant pour une erreur de moins.

## 5 Codes Reed-Solomon

Cas particulier des codes BCH.

**Def.** Un code RS correcteur de  $t$  erreurs est un code cyclique de longueur  $2^m - 1$  ayant uniquement  $2t$  zéros consécutifs.

**Ex.** Avec  $\{1, 2, 3, 4\}$ ,  $g(X) = (X - \alpha)(X - \alpha^2)(X - \alpha^3)(X - \alpha^4) \in \text{GF}(2^m)[X]$ .

Pour les RS, le corps des symboles et le corps localisateur d'erreurs sont les mêmes. Se sont des codes non binaires.

On a  $\deg(g(X)) = 2t$ ,  $g(X) = (X - \alpha) \cdots (X - \alpha^{2t})$  et  $n - k = 2t$ , d'où  $d_{\min} = 2t + 1$  (code à distance maximale).

**Ex.** On prend  $RS(15, 11, 5)$ , sur  $\text{GF}(16) \simeq \text{GF}(2)[X]/1 + X + X^4$ , avec  $n = 2^4 - 1$ .

Il corrige 2 erreurs :  $g(X) = X^4 + \alpha^{13}X^3 + \alpha^6X^2 + \alpha^3X + \alpha^{10} \in \text{GF}(16)[X]$ . On a  $n - k = 4$ , donc  $k = 11$  et  $d_{\min} = 5$ .

**Ex.** DVB :  $RS(204, 188)$  (c'est un RS raccourci,  $204 \neq 2^m - 1$ ). La chaîne de transmission (le code source) impose d'utiliser 188 octets, donc sur  $\text{GF}(2^8) = \text{GF}(256) \simeq \text{GF}(2)[X]/X^8 + X^6 + X^3 + X^2 + 1$ .

Il corrige 8 erreurs :  $2t = 16$  et l'on a  $g(X) = (X + 1)(X + \alpha) \cdots (X + \alpha^{15})$ . Alors  $\deg(g(X)) = 16 \implies k = 239$  mais on veut  $k = 188$ . Donc on raccourcit le code : on rajoute des zéros pour le codage, on les enlève pour la transmission et on les rajoute pour le décodage.

### Algorithme d'Euclide

Les zéros du code sont  $\{0, \dots, 2t - 1\}$ . On a  $\forall i \in \llbracket 0; n - 1 \rrbracket$ ,  $\Lambda(\alpha^{-i})E(\alpha^{-i}) = \lambda_i e_i = 0$ . Donc  $\Lambda(X)E(X) = 0 \pmod{X^n - 1}$ , d'où  $\Lambda(X)E(X) = \Omega(X)(X^n - 1)$ .

Il vient  $[\Lambda(X)E(X) = \Omega(X)(X^n - 1)] \pmod{X^{2t}}$ , puis  $\Lambda(X)[E(X) \pmod{X^{2t}}] = \Omega(X) \pmod{X^{2t}}$  et  $\Lambda(X)S(X) = \Omega(X) \pmod{X^{2t}}$ . Donc  $\Lambda(X)S(X) + q(X)X^{2t} = \Omega(X)$ ,  $\Omega(X)$  est un diviseur commun de  $S(X)$  et  $X^{2t}$ .

Dans l'algorithme d'Euclide de base on calcule  $\text{pgcd}(a(X), b(X))$ , en supposant  $\deg(b) \leq \deg(a)$ , par :

$$\begin{aligned} a(X) &= b(X)q_1(X) + r_1(X) \\ b(X) &= r_1(X)q_2(X) + r_2(X) \\ &\dots\dots\dots \\ r_i(X) &= r_{i+1}(X)q_{i+2}(X) + r_{i+2}(X) \end{aligned}$$

et on s'arrête dès que le degré d'un reste est nul.

Dans la version généralisée on a

$$f_i(X)a(X) + g_i(X)b(X) = r_i(X)$$

avec

$$\begin{aligned} f_i(X) &= f_{i-2}(X) + f_{i-1}(X)q_i(X) \\ g_i(X) &= g_{i-2}(X) + g_{i-1}(X)q_i(X) \\ f_{-1}(X) &= 1 & f_0(X) &= 0 & f_1(X) &= 1 \\ g_{-1}(X) &= 0 & g_0(X) &= 1 & g_1(X) &= q_1(X) \end{aligned}$$

L'algorithme d'Euclide généralisé appliqué à  $S(X)$ ,  $X^{2t}$  donne donc  $\Lambda(X)$  et  $\Omega(X)$ . On arrête l'algorithme dès que  $r_i(X)$  est de degré  $\leq t - 1$ .

### Algorithme de Forney pour connaître les valeurs des erreurs

Soit  $\Omega(X)$  le polynôme évaluateur d'erreur. En dérivant  $\Lambda(X)E(X) = \Omega(X)(X^n - 1)$  on obtient

$$\Lambda'(X)E(X) + E'(X)\Lambda(X) = \Omega'(X)(X^n - 1) + n\Omega(X)X^{n-1}$$

Pour  $i$  une position d'erreur,  $\lambda_i = \Lambda(\alpha^{-i}) = 0$  et  $e_i = E(\alpha^{-i})$ . Donc  $\Lambda'(\alpha^{-i})E(\alpha^{-i}) = n\Omega(\alpha^{-i})\alpha^{n-1-i}$ .

## 6 Codes convolutifs

Les codes convolutifs ont été inventés par Elias en 1954 et sont différents des codes en blocs.

### Principe

Il y a un effet mémoire : la sortie d'un codeur convolutif dépend d'un symbole courant à coder ainsi que du symbole précédent et du résultat de codage du symbole précédent.

Un **codeur convolutif** est une machine à états finis.

Le code possède un **rendement**  $R = \frac{k}{n}$  où  $k$  est le nombre de symboles en entrée et  $n$  le nombre de symboles en sortie. Il a également une **longueur de contrainte** qui correspond au nombre de temps horloge influençant les sorties du codeur. Pour un 1/2-taux codeur,  $L = n + 1$ .

**Ex.** Soit un codeur avec  $R = \frac{1}{2}$  et  $L = 3$ .

## Représentation polynomiale

Soit une séquence binaire  $e_i(1), \dots, e_i(n)$  sur l'entrée  $i$ . On lui associe le polynôme  $e_i(X) = \sum_{t=1}^n e_i(t)X^t$ . Alors chaque sortie possède une représentation polynomiale comme  $s_i(X) = \sum_{j=1}^k g_{ij}(X)e_j(X)$ .

Les  $g_{ij}$  sont appelés les **polynômes générateurs** du codes.

**Ex.** Sur l'exemple précédent on a  $s_1^{(i)} = e^{(i)} + e^{(i-1)} + e^{(i-2)}$  et  $s_2^{(i)} = e^{(i)} + e^{(i-2)}$ . Donc  $s_1(X) = e(X) + Xe(X) + X^2e(X) = g_1(X)e(X)$  avec  $g_1(X) = 1 + X + X^2$  et  $s_2(X) = e(X) + X^2e(X) = g_2(X)e(X)$  avec  $g_2(X) = 1 + X^2$ .

**Ex.** Prenons le code avec  $R = \frac{2}{3}$  et  $L = 2$  donné par le circuit suivant :

On a alors :

$$\begin{aligned} s_1(X) &= \underbrace{(1+X)}_{g_{11}(X)} \cdot e_1(X) + \underbrace{X}_{g_{12}(X)} \cdot e_2(X) \\ s_2(X) &= \underbrace{1}_{g_{21}(X)} \cdot e_1(X) + \underbrace{(1+X)}_{g_{22}(X)} \cdot e_2(X) \\ s_3(X) &= \underbrace{(1+X)}_{g_{31}(X)} \cdot e_1(X) + \underbrace{0}_{g_{32}(X)} \cdot e_2(X) \end{aligned}$$

Donc  $G = \begin{pmatrix} 1+X & 1 & 1+X \\ X & 1+X & 0 \end{pmatrix} \in \mathfrak{M}_{k,n}$  est la matrice génératrice du code.

**Def. Code convolutif récursif** : une partie de la sortie est réintroduite dans les étages mémoire. Les polynômes signatures sont remplacés par des quotients de polynômes.

**Def. Code convolutif systématique** : une partie de ses sorties est exactement égale à ses entrées.

**Ex.** Sur l'exemple 1 : on crée un codeur systématique. On passe de  $e(X) \rightarrow \begin{cases} s_1(X) = (1+X+X^2)e(X) \\ s_2(X) = (1+X^2)e(X) \end{cases}$  à

$e(X) \rightarrow \begin{cases} f_1(X) = e(X) \\ f_2(X) = \frac{1+X^2}{1+X+X^2}e(X) \end{cases}$ . Donc  $f_2(X) \cdot (1+X+X^2) = (1+X^2)e(X)$ , d'où

$$f_2(X) = e(X) + X^2e(X) + Xf_2(X) + X^2f_2(X) = e(X) + Xf_2(X) + X^2(e(X) + f_2(X)).$$

**Ex.** Sur l'exemple 2 avec  $R = \frac{2}{3}$ . On prend les deux premières colonnes de  $G$  pour former  $G_1 = \begin{pmatrix} 1+X & 0 \\ 1 & 1+X \end{pmatrix}$ .

La nouvelle matrice est alors

$$G' = G_1^{-1} \cdot G = \begin{pmatrix} 1 & 0 & p_1(X)/p_2(X) \\ 0 & 1 & p_3(X)/p_1(X) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{1}{1+X} \\ 0 & 1 & \frac{1+X+X^2}{1+X^2} \end{pmatrix}$$

$$f_3(X) = \frac{1}{1+X}e_1(X) + \frac{1+X+X^2}{1+X}e_2(X)$$

$$(1+X^2)f_3(X) = \frac{1+X^2}{1+X}e_1(X) + (1+X+X^2)e_2(X)$$

$$f_3(X) = X^2f_3(X) + e_1(X) + Xe_1(X) + e_2(X) + Xe_2(X) + X^2e_2(X)$$

$$= e_1(X) + e_2(X) + X(e_1(X) + e_2(X)) + X^2(f_3(X) + e_2(X))$$

## Codes catastrophiques

**Ex.** Exemple avec  $R = \frac{1}{2}$ ,  $L = 3$  :

$$1111 \dots 1 \rightarrow 11010000 \dots 0.$$

**Def.** Un code est dit **catastrophique** s'il existe une séquence en entrée de poids infini qui donne une séquence en sortie de poids fini.

**Prop.** Pour un code de rendement  $R = \frac{1}{n}$ , le code est catastrophique ssi  $\text{pgcd}(g_i(X)) \neq 1$ . Pour un code de rendement  $R = \frac{k}{n}$ , le code est catastrophique ssi le  $\text{pgcd}$  des mineurs d'ordre  $k$  de la matrice  $G$  est différent de 1.

## Représentations graphiques

### Algorithme de décodage

Décodage optimal ML : on suppose les séquences d'entrées équiprobables. Pour un BSC on aurait  $ML = \min_{c \in C} d_H(r, c)$  avec  $r$  le message reçu.

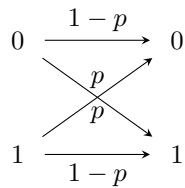


## 7 Performance des codes correcteurs

### Pour les codes en blocs

On prend  $C(n, k, d_{\min})$ ,  $R = \frac{k}{n}$  et  $d_{\min}$  qui détermine le pouvoir correcteur  $t \leq \lfloor \frac{d_{\min}-1}{2} \rfloor = \lceil \frac{d_{\min}}{2} \rceil - 1$ .

Rappel : canal de transmission BSC.



AWGN :  $y = x + b$  avec  $x \in 2\text{-PAM}$  est équivalent à un BSC avec  $p = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \simeq \exp\left(-\frac{E_b}{N_0}\right)$  où  $\sigma^2 = \frac{N_0}{2}$ .

$$\begin{bmatrix} k \text{ bits} \end{bmatrix} \xrightarrow[n \text{ textbitsdecode}]{\text{BSC}} \begin{bmatrix} c \end{bmatrix} \xrightarrow[n \text{ bits reçus}]{\text{BSC}} \begin{bmatrix} r \end{bmatrix}$$

Probabilité d'erreur :  $P_{e,\text{mot}} \leq \sum_{i=t+1}^n \binom{n}{i} p^i (1-p)^{n-i}$ . On travaille avec  $p \ll 1$  donc  $1-p \simeq 1$  (i.e. on travaille avec un fort rapport signal sur bruit).

Donc

$$\begin{aligned} P_{e,\text{mot}} &\leq \binom{n}{t+1} p^{t+1} \\ &\leq C^{te} \cdot p^{t+1} \\ &\leq C^{te} \cdot \exp\left(-\frac{E_s}{N_0}\right)^{\lceil d_{\min}/2 \rceil} \\ &\leq C^{te} \cdot \exp\left(-\frac{E_s}{N_0} \cdot \left\lceil \frac{d_{\min}}{2} \right\rceil\right) \end{aligned}$$

**Attention**  $k$  bits d'info  $\rightarrow E_b$  énergie bit,

$n$  bits d'info  $\rightarrow E_s$  énergie symbole,

et on a  $k \cdot E_b = n \cdot E_s$ .

$$P_{e,\text{mot}} \leq C^{te} \cdot \exp\left(-\frac{E_s}{N_0} \cdot R \cdot \left\lceil \frac{d_{\min}}{2} \right\rceil\right)$$

Le gain de codage pour une même  $P_e$  est :

$$\begin{aligned} CG &= 10 \log_{10} \left( \frac{RSB_{\text{codé}}}{RSB_{\text{non codé}}} \right) \\ &= 10 \log_{10} \left( \frac{\frac{E_s}{N_0} R \left\lceil \frac{d_{\min}}{2} \right\rceil}{\frac{E_b}{N_0}} \right) \\ &= 10 \log_{10} \left( R \left\lceil \frac{d_{\min}}{2} \right\rceil \right) \end{aligned}$$

Pour un code à répétition de rendement  $\frac{1}{3}$ ,  $d_{\min} = 3$ , on a  $CG = -1,7$  dB.

Pour un code de Hamming (7, 4, 3) :  $CG = 0,57$  dB.

Pour un code de Hamming (15, 11, 3) :  $CG = 1,66$  dB.

### Pour les codes convolutifs

$$CG = 10 \log_{10}(Rd_{\text{lib}}) - 2 \text{ dB}.$$