**Projet de Recherche (PRe)**

**Spécialité : STIC**

**Année scolaire : 2021-2022**

**Titre du rapport de stage**

**Complément de titre**

|  |
| --- |
| **Illustration**  **(facodemodem)** |

**Demodem de demodemodemodem**

**(Demodemo de demo de demodemodemodem de de demo demodemode de dem de demodemodemodem)**

**Demode : Demod Demodem Odemodemo : 2023**

**Demode DEMOD Demod : Demode demodemod d’demodem :**

**Demod demodemo de \_\_ /\_\_ /\_\_\_\_ de \_\_ /\_\_ /\_\_\_\_**

**DEM de d’demodemod d’demodem :**

**Demodem :**

# Résumé

# Remerciements

Table des matières

[Résumé 3](#_Toc107484101)

[Remerciements 4](#_Toc107484102)

[Chapitre 1. – Equivalences et égalité entre codes de Reed Solomon et codes de Reed Solomon généralisés 7](#_Toc107484103)

[I. Généralités sur les codes de Reed Solomon 7](#_Toc107484104)

[II. Génération des codes 7](#_Toc107484105)

[1. Corps de Galois 7](#_Toc107484106)

[2. Codes de Reed Solomon 9](#_Toc107484107)

[3. Codes de Reed-Solomon généralisés 9](#_Toc107484108)

[4. Exemples 10](#_Toc107484109)

[5. Notes théoriques sur les RS et GRS 10](#_Toc107484110)

[III. Comparaisons pratiques des codes RS et GRS 11](#_Toc107484111)

[1. Condition pour la cyclicité des codes GRS 11](#_Toc107484112)

[2. Répartitions des poids des mots de codes [5,7] 12](#_Toc107484113)

[3. Classes d’équivalences des codes [5;7] [5;15] et [7;15] 14](#_Toc107484114)

[4. Comparaison des matrices des codes 14](#_Toc107484115)

[5. Comparaison des polynômes générateurs 16](#_Toc107484116)

[IV. Egalité entre RS et GRS 17](#_Toc107484117)

[1. Condition d’égalité entre codes GRS et RS 17](#_Toc107484118)

[Chapitre 2. Décodage des codes GRS 19](#_Toc107484119)

[Annexes 20](#_Toc107484120)

[Annexe A : Preuve du Lemme IV.1.1 20](#_Toc107484121)

[Annexe B : CryptoExplorer 22](#_Toc107484122)

# – Equivalences et égalité entre codes de Reed Solomon et codes de Reed Solomon généralisés

## Généralités sur les codes de Reed Solomon

Les codes correcteurs sont une manière d’encoder une information avant de la transmettre. Le but de ces codes est de rendre possible la détection et la correction d’erreurs de transmission. On introduit de la redondance en transmettant des bits d’informations supplémentaires qui permettent de corriger plus ou moins d’erreur ; plus on veut pouvoir corriger d’erreur, plus il faut introduire de redondance et donc de bits supplémentaires.

Les codes de Reed Solomon sont une catégorie particulière de codes correcteurs adaptés à la correction de paquets d’erreurs. En effet dans un canal, les erreurs arrivent souvent par paquet sur plusieurs bits consécutifs à cause d’une perturbation. Les codes de Reed Solomon permettent de traiter ce problème mais ne permettraient pas de corriger le même nombre d’erreur réparties aléatoirement. Cependant, en pratique, les erreurs arrivent toujours par paquet.

Les codes de Reed Solomon classiques sont générés par division polynomiale dans un corps de Gallois. M. Arnaud Dagnelies présente dans sa thèse une méthode de généraliser les codes de Reed Solomon en encodant cette fois ci avec des évaluations polynomiales.

Le but de cette étude est de comprendre à quelles conditions on a équivalence ou égalité entre les codes de Reed-Solomon (RS) et les codes de Reed-Solomon généralisés (GRS).

## Génération des codes

### Corps de Galois

#### Corps à 8 éléments

Encoder des données à l’aide d’un code de Reed-Solomon consiste à faire des opérations dans un cors fini. Un mot de code est une suite de symboles qui sont des éléments de ce corps. On va donc choisir des corps de Galois.

Dans un premier temps, pour simplifier, on va s’intéresser à des codes de petite taille. On prend donc le corps de Galois à 8 engendré par le polynôme à coefficients dans {0, 1}

|  |  |
| --- | --- |
| Polynôme générateur : | |
| Forme Polaire | Forme Cartésienne |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Avec ce corps de Galois, on a les tables d’addition et de multiplication suivantes :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

#### Corps à 16 éléments

On s’intéressera aussi à des codes dans un corps à 16 éléments. On prendra le corps généré par le polynôme

|  |  |
| --- | --- |
| Polynôme générateur : | |
| Forme Polaire | Forme Cartésienne |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Codes de Reed Solomon

Les mots de code sont écrits sur n=7 symboles, chaque symbole étant un élément du corps de Galois choisi.

On veut ici un code 1-correcteur (t=1), donc les mots à encoder sont des mots de   
 symboles. Il y a donc mots de codes.

#### Méthode d’encodage systématique

Pour avoir un code 1-correcteur, on doit choisir un polynôme à deux racines consécutives. On choisit dans un premier temps

On écrit le mot à encoder sous forme polynômiale  de degré au plus 4 : chaque symbole du mot est un coefficient de . Par exemple le mot sera noté

On divise ensuite par et on note le reste.

Le mot de code à transmettre est

### Codes de Reed-Solomon généralisés

La méthode d’encodage présentée ici est expliquée dans la thèse de Arnaud Dagnelies. Ici l’encodage est réalisé par évaluation polynomiale.

#### Méthode d’encodage

Soit le vecteur des points où l’on va évaluer le code GRS, avec

Soit le vecteur de normalisation non nul.

On note comme précédemment le mot à encoder sous forme polynomiale.

Alors le mot encodé est {}

### Exemples

Soit le mot de code . On a donc

#### Reed-Solomon

Soit

On fait la division euclidienne :

Donc

Donc

Donc le mot encodé est

#### Reed-Solomon généralisé

Soit et

Donc ici le mot encodé est

### Notes théoriques sur les RS et GRS

Les codes correcteurs sont caractérisés par leurs matrices de génération G et de contrôle G. Encoder un mot, revient à le multiplier par G, et la matrice G permet de détecter les erreurs.

On se place dans un corps de Galois à éléments (n éléments non nuls). On note le code de Rees-Solomon obtenu par division polynômiale par , et le code de Rees-Solomon généralisé obtenu avec de .

#### Codes RS (sous forme non systématique)

Le code où a pour matrice G :

Soit tel que , Alors à pour matrice de contrôle H :

#### Codes GRS

On note la longueur du message et la longueur des mots de codes.

Soit et , le code a pour matrice génératrice G :

La matrice de contrôle H est de la forme :

Où les sont obtenus en résolvant le système :

avec

C’est-à-dire

## Comparaisons pratiques des codes RS et GRS

### Condition pour la cyclicité des codes GRS

Pour pouvoir cmparer les codes RS et GRS, on doit savoir à quelles conditions un codes GRS est en fait un code RS cyclique. La proposition suivante, tirée du livre Notes en Coding theory de J.I Hall, donne ces conditions.

Notation :

Pour une racine primitive de un corps de Galois, on note :

Remarques :

* est le vecteur dont toutes les composantes sont **1.**
* est la liste des puissances de .
* . C’est-à-dire qu’un décalage cyclique de est un multiple de .

Proposition :

est cyclique. C’est-à-dire est cyclique si et avec

Preuve :

On a vu que pour , le terme de la matrice génératrice est

Donc pour , le terme de la matrice génératrice est  
, avec . Donc les lignes de la matrice sont les pour .

On a vu que les sont des décalages de , donc le code est cyclique.

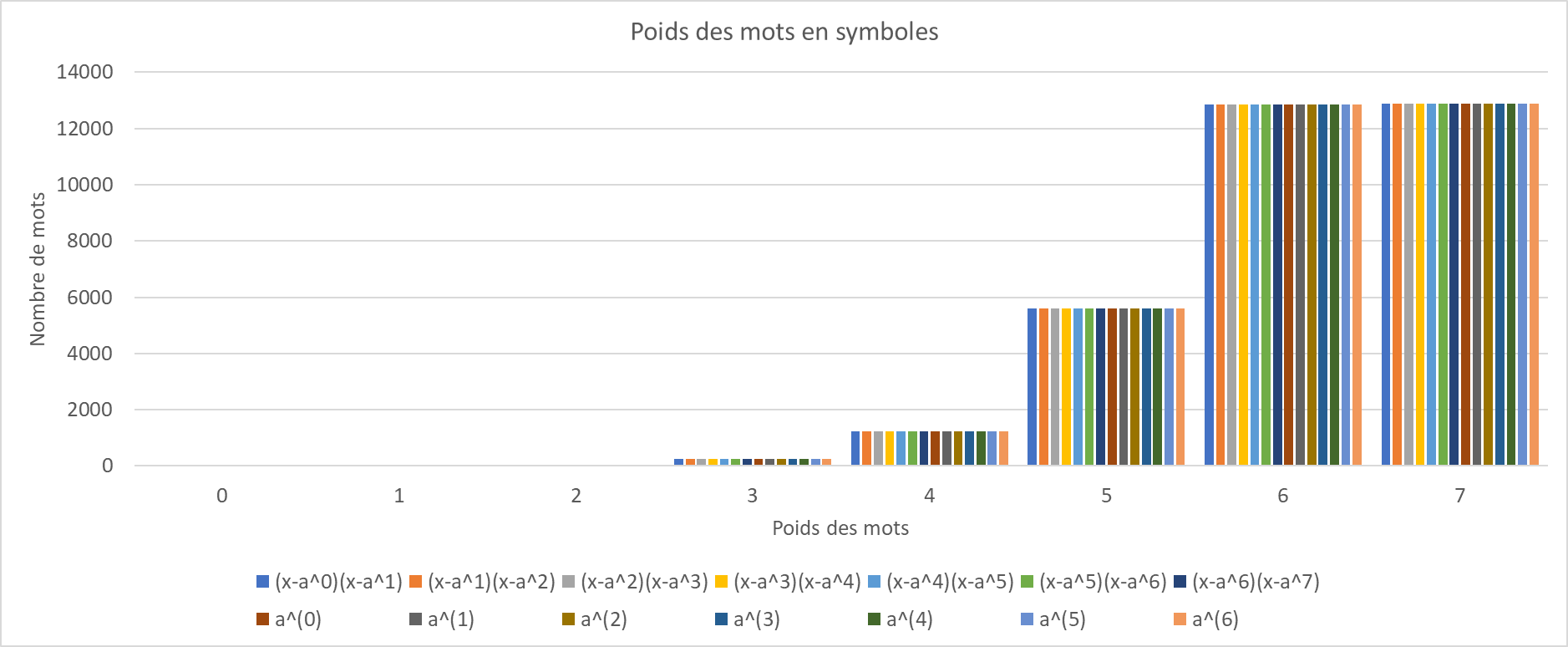
### Répartitions des poids des mots de codes [5,7]

Dans un corps de Galois à 8 éléments, il y a 7 codes de Reed-Solomon [5,7] :

Il existe également 7 codes de Reed-Solomon généralisés qui respectent la condition de cyclicité donnée par la proposition précédente :

#### Equivalence en poids en symboles

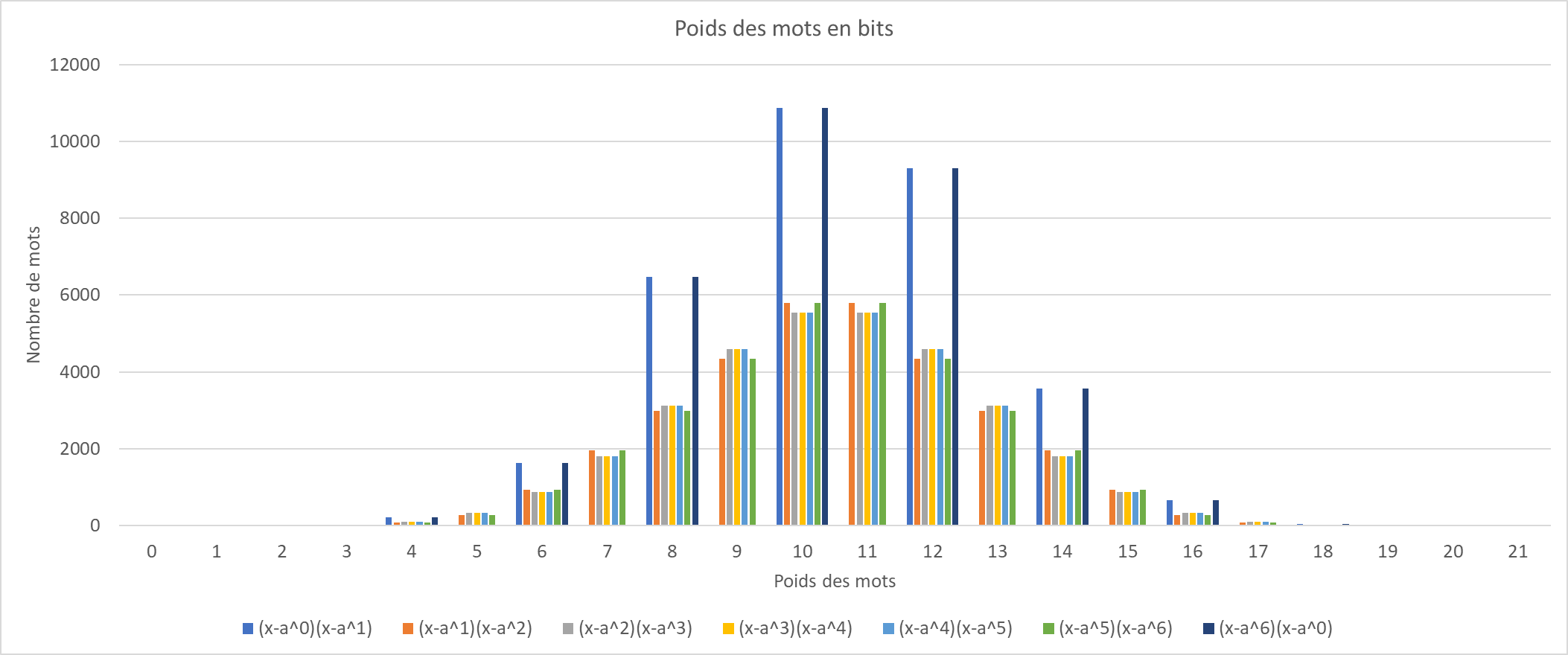
Tout d’abord, on regarde si les codes sont équivalents en poids en symboles. C’est-à-dire qu’on compte le nombre de symboles non nuls dans chaque mot des codes.



On voit que les 14 codes [5,7] sont parfaitement équivalents. Si on veut voir une différence il faut regarder l’équivalence en poids en bits.

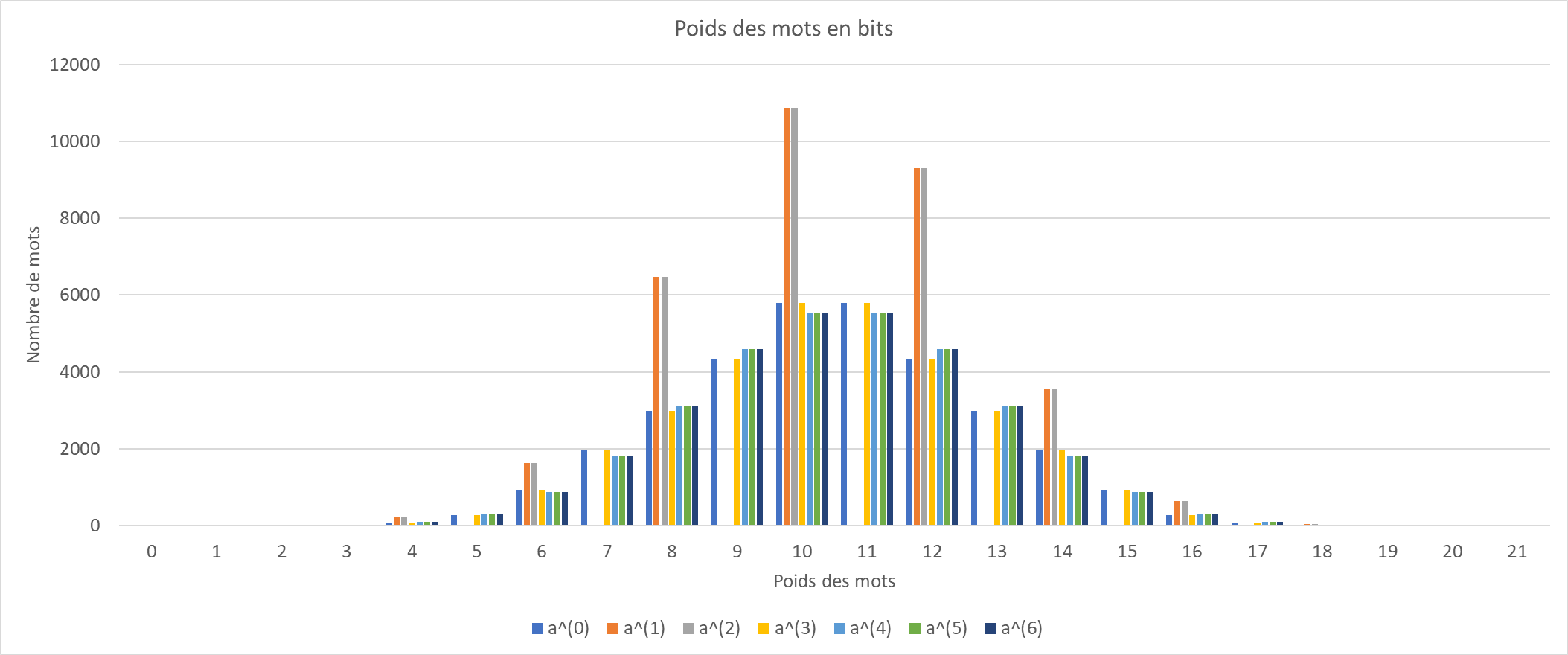
#### Equivalence en poids en bits

Tout d’abord on regarde la répartition des poids des mots des 7 codes de Reed Solomon.



On voit l’apparition de 3 familles d’équivalence :

On regarde maintenant la répartition des poids des 7 codes GRS.



On remarque de nouveau les 3 même familles d’équivalence.

Finalement, on a donc 3 familles d’équivalence :



### Classes d’équivalences des codes [5;7] [5;15] et [7;15]

On note

#### Codes [5 ; 7]

#### Codes [5 ; 15]

#### Codes [7 ; 15]

#### Remarques

On remarque que dans les 3 cas, on a un nombre tel que . De plus, on a pour les deux codes où et pour le code où .

### Comparaison des matrices des codes

#### Rappels théoriques :

Codage sous forme systématique :

Un code sous forme systématique est un code dont la matrice génératrice est de la forme :

Propriété :

Tout code est équivalent à un code mis sous forme systématique que l’on trouve par l’algorithme du Pivot de Gauss.

#### Comparaisons des matrices

Comparons les matrices des codes d’une des classes d’équivalence, par exemple les codes :

Les formes des matrices sont données à la partie II.5

On note (resp. ) les matrices génératrices du code (resp. ) et (resp. ) les matrices des codes sous forme systématique équivalents.

Code  :

Code :

Code :

Code :

La première chose que l’on remarque, c’est que et . Ces codes sont en fait égaux comme on le démontrera dans la partie suivante. C’est-à-dire que leurs mots de code sont les mêmes.

Si on compare toutes les matrices, on a :

On a en fait : , c’est-à-dire qu’une matrice d’un code cyclique est égale à une matrice d’un code généralisé.

### Une image contenant texte Description générée automatiquementComparaison des polynômes générateurs

Voici les polynômes des codes [7 ; 15] :

Si on prend deux codes équivalents avec leurs polynômes notés et .

On remarque qu’on a

C’est-à-dire que le polynôme est le polynôme renversé et normalisé.

Cela se vérifie pour tous les codes testés.

Ce critère n’est pour l’instant pas prouvé, mais on pourra l’utiliser pour vérifier les classes d’équivalences de codes plus gros et donc trop long à calculer comme les codes . En effet calculer les polynômes est très simple pour toutes les tailles de codes. On ne pourra pas comparer aux codes GRS, mais on va donner dans la partie IV un critère d’égalité entre RS et GRS.

#### Application du critère

Ici, on va simplement appliquer le critère à d’autres familles de codes en supposant qu’il est vrai.

Codes [9 ; 15]

## 

Ici on a avec

Tableau des valeurs de en fonction des codes.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 3 | 7 | 4 | 2 |
| 3 | 15 | 12 | 2 |
| 4 | 7 | 3 | 6 |
| 4 | 15 | 11 | 10 |
| 5 | 7 | 2 | 3 |
| 5 | 15 | 10 | 3 |
| 6 | 7 | 1 | 0 |
| 6 | 15 | 9 | 11 |
| 7 | 15 | 8 | 4 |
| 7 | 31 | 24 | 4 |
| 8 | 15 | 7 | 12 |
| 9 | 15 | 6 | 5 |
| 10 | 15 | 5 | 13 |
| 11 | 15 | 4 | 6 |

## Egalité entre RS et GRS

Les résultats exhibés dans cette partie sont présentés par Jonathan I Hall dans son livre *Notes on coding theory*

### Condition d’égalité entre codes GRS et RS

Lemme IV.1.1 :

Preuve en Annexe.

Lemme IV.1.2 :

Pour un polynôme dans , un corps, et , tel que

Preuve :

Par division polynômiale, on a l’existence de et tels que avec , donc est une constante. On trouve que en calculant .

Conséquence :

Théorème :

Ce critère confirme bien les résultats de la partie III.4 où on avait trouvé des matrices sous forme systématiques identiques pour certains codes.

Preuve :

Soit . Par le **Lemme IV.1.1**, on a .

Donc comme vu précédemment, les lignes de la matrice génératrice de sont les pour .

Ainsi, pour et un mot de code et sa forme polynômiale,

Ainsi, par la conséquence du **Lemme IV.1.2**, on a :

Comme est unitaire et de degré , il s’agit du polynôme générateur de C.

De plus les sont des puissances consécutives de , donc .

# Décodage des codes GRS

## Introduction

### Principe de l’algorithme

Soit le mot envoyé tel que et le mot reçu. On prend tel que , .

Par exemple, si le polynôme utilisé pour encoder est sur , l’ensemble des points envoyés est :

.

On suppose que le mot reçu est tel que l’ensemble des points est :

.

Il faut maintenant trouver les polynômes de degré inférieur à 2 qui passent par le maximum de points. Par exemple ici les polynômes et sont des candidats équiprobables sans informations supplémentaires.

On voit que le problème est donc un problème d’interpolation polynomiale.

Cet algorithme, plus simple que celui qui est présenté dans la suite, est l’algorithme de Sudan. Cet algorithme ne permet que du décodage dur, c’est-à-dire qu’on assigne à chacun des points reçus une valeur donnée. L’algorithme suivant permet du décodage « doux ». Cela permet de prendre en compte plus d’informations présente dans le canal de transmission. C’est-à-dire qu’on assigne à une probabilité qu’il soit égal à chaque élément de , c’est ce que l’on appellera dans la suite la matrice de fiabilité. Cependant, le principe de l’algorithme présenté ici reste le même que l’algorithme de Sudan : il s’agit de trouver les polynômes correspondants aux mots les plus probables.

### Degré pondéré et énumération monomiale

Définition : Le (a,b) degré pondéré d’un monôme est :

Definition : Le (a,b) degré pondéré d’un polynôme est le plus grands degré pondéré de ses monômes.

L’algorithme est basé sur un polynôme bivarié . Si on remplace par , on a : . De plus, est de degré au plus . Donc le degré de est majoré par le degré pondéré de .

On note

Définition : L’ensemble est l’ensemble des monomes de (a,b) degré pondéré inférieur ou égal à

### Zéros de multiplicité m

Définition : un polynôme a un zéro de multiplicité en si   
.

Théorème : Si a un zéro de multiplicité en et , alors

### Expression des contraintes

Le but de cette section est de déterminer les contraintes sur pour qu’il ait des zéros de multiplicité données à des points données.

Soit , comment choisir les pour que ait un zéro de multiplicité en . C’est-à-dire :

### Matrice de multiplicité

On note les points où le code GRS est évalué et le vecteur contenant tous les éléments de

Définition : La matrice de multiplicité est la matrice telle que pour tout , le polynôme a un zero de multiplicité au point .

Définition : Le coût de la matrice est défini par :

Où est la matrice dont toutes les composantes sont égales à .

Le coût d’une matrice est égal au nombre de contraintes sur les coefficients de qu’il faudra satisfaire lors de l’interpolation.

Notation : On note le plus petit entier tel que

Le but de cet entier est de savoir le degré minimum que doit avoir le polynôme afin d’avoir suffisamment de coefficients pour pouvoir satisfaire toutes les contraintes imposées par . Si , le polynôme devra satisfaire contraintes sur ses coefficients, et donc devra avoir au moins coefficients. Donc le degré du polynôme doit être tel que **.**

Définition : Le score du mot est :

Théorème : Soit le polynôme qui satisfait les multiplicités de la matrice ,  
, et soit un polynôme d’encodage évalué en et qui donne le mot . ()

Si et

Alors

Ce théorème signifie que si , alors le décodage à fonctionné.

## Construction de la matrice de multiplicité

### Matrice de fiabilité

La matrice de fiabilité est la matrice qui concentre toutes les informations du canal de communication. Elle est construite à partir des informations que l’on a, c’est-à-dire ce que l’on reçoit, et d’autres informations comme la probabilité qu’une erreur ait lieu si le canal transmet les bits un par un. L’avantage de cette modélisation par rapport à l’algorithme de Sudan, est que l’on peut prendre en compte des informations plus complètes. Par exemple :

Dans un canal, la tension -1.5V correspond à 0, -0.5V à 1, 0.5V à 2 et 1.5V à 3. On reçoit . Avec l’algorithme de Sudan, on devrait choisir ou et appliquer l’algorithme ensuite. Ici on peut dire par exemple que . Et donc utiliser plus d’informations que l’on a à disposition. Dans les faits on utilisera des lois normales pour calculer ces probabilités.

Définition : La matrice de fiabilité est une matrice telle que

est une variable aléatoire dans qui représente le mot envoyé, et est une variable aléatoire qui prend ses valeurs dans ce que l’ont reçoit du canal.

Dans un premier temps, on prendra également à valeurs dans ce qui permettra de traiter le cas particulier d’un canal dur, mais l’algorithme qui suit fonctionne exactement de la même manière pour un canal doux. Seule la construction de la matrice est différente.

### Exemple pour un canal dur

On reprend le corps de Galois à 8 éléments présenté au début.

|  |  |
| --- | --- |
| Polynôme générateur : | |
| Forme Polaire | Forme Binaire |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Les symboles sont transmis via le canal bit à bit, et la probabilité qu’un bit soit inversé lors de la transmission est .

On note la distance entre deux symboles. C’est le nombre de bits différents entre les deux. Par exemple, .

On a donc

Par la formule de Bayes,

### Remarques et approximations

On note et vecteurs de variables aléatoires.

On remarque que la probabilité n’est pas égale au produit . En effet, les vecteurs sont des éléments de , mais on n’a pas l’égalité entre et . On n’a que .

La matrice de multiplicité idéale, serait celle qui maximise la probabilité de décoder. C’est-à-dire telle que soit maximale. Cependant ce problème est extrêmement difficile car la matrice affecte les deux côtés de l’inéquation et les probabilités sont difficiles à calculer. Ainsi, on va faire deux approximations.

La première est que l’on ne va pas chercher la matrice la plus optimale parmi toutes, mais simplement la matrice qui maximise le score moyen parmi les matrices de même coût.

La seconde est que l’on va utiliser l’approximation :

Ainsi le problème revient à trouver parmi les matrices de coût qui maximise

### Algorithme

**Initialisation :** On pose la matrice nulle.

Itération : Choisir tel que le ratio soit maximal.

Lorsque l’on augmente de , on augmente le coût de et le score moyen de . Maximiser le ratio ci-dessus revient donc à maximiser la probabilité de décoder le mot.

Fin : On s’arrête lorsque dépasse un seuil fixé. C’est-à-dire lorsque le coût devient trop important.

## Interpolation

## Factorisation

# Annexes

## Annexe A : Preuve du Lemme IV.1.1

Notations :

et

Lemme A.1 :

Si sont les n racines de , alors

Lemme A.2 : Code dual

où

*Preuve :*

Soient

est de degré inférieur à et est de degré inférieur à , donc est de degré inférieur à .

Par interpolation de Lagrange, on a :

On regarde la valeur des coefficients de des deux côtés de l’égalité :

Le produit scalaire des mots de codes est toujours égal à 0 ce qui démontre le Lemme.

Lemme A.3 :

Soient des éléments non nuls de , et un vecteur de dont toutes les composantes sont égales, alors

Lemme IV.1.1 :

*Preuve :*

Par le **Lemme A.2**, on a où pour   
 en notant , on a :

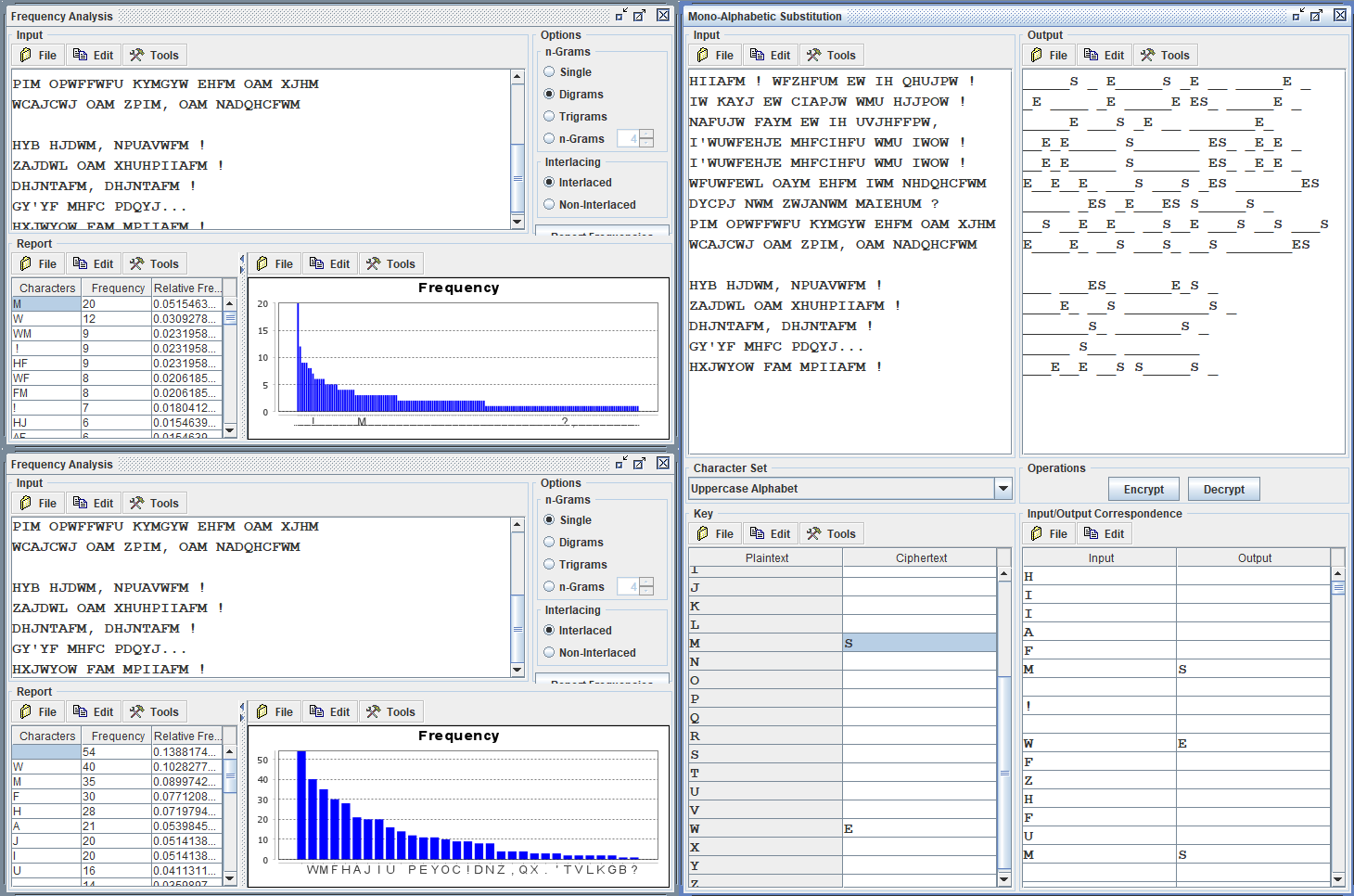
De plus par le **Lemme A.1**: Donc :

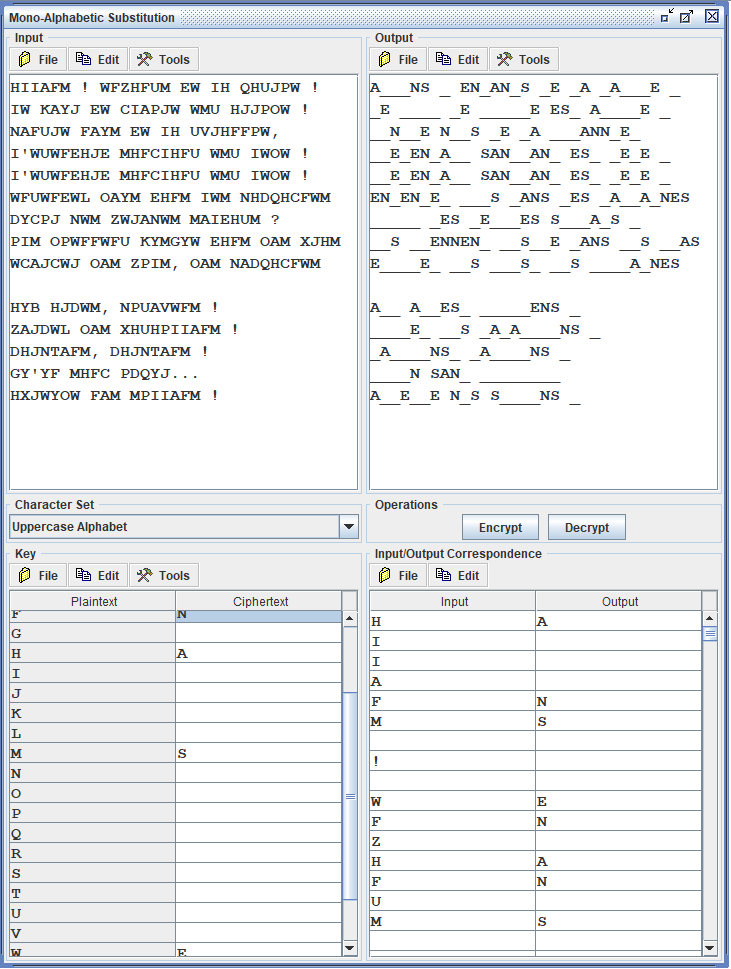
Donc

Donc par le **Lemme A.3 :**

## Annexe B : CryptoExplorer

Lien du programme : [The Home Page of Dr. Donald E. Spickler Jr. (salisbury.edu)](http://faculty.salisbury.edu/~despickler/personal/CryptographyExplorer.html)



* On voit que les lettres les plus courantes sont, dans l’ordre : W,M,F,H
  + On peut remplacer W et M par E et S.
* Les digrammes les plus courants sont WM, HF, WF, FM
  + Le digramme le plus courant en français est ES : il correspond à WM.
  + Le deuxième digramme le plus courant commençant par E est EN. Le F étant très présent dans le message chiffré et souvent après un W, on peut remplacer F par N.
* On a donc W->E, M->S, F->N, H->A.
* Ici on commence à reconnaitre des mots : par exemple le mot FAM correspond à N\_S. Il n’y a qu’une seule solution en français : NOS. Donc on a A->O.
* Le mot MHFC correspond à SAN\_. Il n’y a également qu’une seule solution en français : SANG. Donc C-> G
* Le mot WMU correspond à ES\_. On a donc U->T.

Une image contenant texte

Description générée automatiquement

* On peut également remarquer les lettres doubles. Le mot HIIAFM correspond à A\_\_ONS. En français, les consonnes qui peuvent se doubler sont : c,j,l,m,n,p,r,s,t,b,d,g,k,z. Seul L a du sens. On a donc I->L.
* Quelques lettres sont encore facilement reconnaissables comme le F de EN\_ANT, le D de SOL\_ATS, le I de \_LS et F\_LS puis le R de GLOI\_E.
* Ici le texte commence à être reconnaissable facilement.

Une image contenant texte

Description générée automatiquement

Remarques :

* Le A et le N ne sont pas dans le bon ordre de fréquence. Il est possible de le remarquer car le digramme AN est beaucoup plus fréquent que le digramme NA et le digramme EA n’existe presque pas alors que EN est l’un des plus fréquent.
* Les digrammes ne sont pas dans l’ordre, mais si on ne s’intéresse que au digrammes contenant une lettre donnée à une position donnée on peut retrouver des lettres.

