Étude algèbrique des mots de poids minimum des codes cycliques, méthodes d'algèbre linéaire sur les corps finis.

Daniel Augot

THÈSE de DOCTORAT de l'UNIVERSITÉ PARIS 6

Spécialité : INFORMATIQUE

Présentée par

Daniel AUGOT

pour obtenir le titre de DOCTEUR DE L'UNIVERSITÉ PARIS 6

Sujet de la thèse :

ÉTUDE ALGÉBRIQUE DES MOTS DE POIDS MINIMUM DES CODES CYCLIQUES, MÉTHODES D'ALGÈBRE LINÉAIRE SUR LES CORPS FINIS.

Soutenue le 2 décembre 1993, devant le jury composé de :

Daniel LAZARD	*Président*
Philippe FLAJOLET	*Rapporteur*
Jacques WOLFMANN	*Rapporteur*
Maurice MIGNOTTE	*Examinateur*
Paul CAMION	*Examinateur*
Pascale CHARPIN	*Directeur*
Thomas ERICSON	*Examinateur*

Je remercie Philippe Flajolet d'avoir consacré une partie de son talent et de son temps à la lourde tâche de rapporteur. J'en suis très honoré.

Merci à Jacques Wolfmann de la gentillesse qu'il a témoigné à mon égard, et d'avoir accepté d'être rapporteur.

Merci à Daniel Lazard. C'est lui qui m'a mis sur les rails du calcul formel en m'ouvrant les portes de la "salle 105", ou résidait Scratchpad II. Il a aussi bien voulu se prêter à d'intéressantes discussions, dont le théorème I.14 du chapitre I est le fruit.

Jag är mycket ärad av Herr Professor Thomas Ericsons närvaro. Han har visat stort intresse för mitt arbete och han har accepterat att vara juryman avseende min avhandling. För detta är jag mycket tacksam.

Je remercie aussi le professeur Maurice Mignotte d'avoir accepté d'être membre du jury.

Merci à Pascale Charpin de m'avoir confié ce sujet, à la croisée de la théorie des codes et du calcul formel. Pascale a su être à mes cotés à chaque instant, prodiguant conseils et encouragements, avec ténacité.

Je remercie Paul Camion de m'avoir associé à ses réflexions sur le calcul du polynôme minimal d'une matrice, problème qui a conduit à tous les algorithmes du chapitre IV.

Merci aux joviaux Nicolas Sendrier et Hervé Chabanne, dont le contact m'a été profitable. Outre que Pascale, Nicolas et moi avons eu une collaboration fructueuse, Nicolas m'a fait profiter de ses compétences en informatique. N'oublions pas non plus Gaétan Haché, Dominique Le Brigand et Guy Chassé, les géomètres. Un remerciement aussi à Françoise Levy-dit-Vehel, qui m'a soumis l'exemple des duaux des BCH.

Retournons au LITP, et à l'équipe de Daniel Lazard : Renaud Rioboo, qui m'a largement aidé dans ma première confrontation avec les "machines", et qui fut d'une grande patience ; Jean-Charles Faugère, dont on ne vantera jamais assez les qualités, qui mis à ma disposition Gb pour le corps à deux éléments, dont les résultats ont guidé mes intuitions ; Annick Valibouze, que je remercie pour les références précises sur les relations de Girard-Newton, et pour son aide dans ma charge de moniteur à l'université ; et Marc Moreno-Maza, qui me fit découvrir la cucaracha.

Entre ces deux équipes, j'ai eu l'occasion de séjourner au Laboratoire d'Informatique de l'Ecole Polytechnique, grâce à l'accueil que m'a accordé Jean-Marc Steyaert. Cette année me fut très profitable, aussi grâce à François Morain. Ceci m'amène au projet ALGO, dont je salue tous les membres.

Enfin, j'adresse mes remerciements à Marie, Côme et Bia, demeurant à Polytechnique, c'est-à-dire à Joel Marchand, qui sut mettre à la disposition des utilisateurs des machines et des outils, et qui est toujours prodigue de conseils pour l'utilisateur moins averti que lui.

Table des matières

Table des tables

Introduction générale

1 Utilisation des équations de Newton

L'origine de ce travail est un article de T. Kasami, S. Lin et W.W. Peterson [KLP68]. Cet article présentait, entre autres sujets, une étude des mots de poids minimal des codes de Reed et Muller raccourcis, pour caractériser leurs mots de poids minimal, qui sont ceux dont le support est un sous-espace vectoriel.

Le code de Reed et Muller d'ordre r est inclu dans le code BCH de distance construite $2^{m-r}-1$, et la question que posait Pascale Charpin était d'étudier les mots de poids minimal de ces codes BCH. T. Kasami utilisait les identités de Newton pour caractériser les mots de poids minimal des codes de Reed et Muller. Nous avons repris ces idées, et étudié le système défini par les équations de Newton. Le sujet de ce travail est donc d'étudier les relations entre le système algébrique défini par les équations de Newton et les propriétés des mots de poids minimal.

2 Les équations de Newton : un système nécessaire

Nous avons prouvé que les mots de poids minimal des codes $BCH(n = 2^m - 1, \delta = 2^{m-2} - 1)$ sont les mots du code de Reed et Muller d'ordre 2 raccourci. Pour prouver ce résultat, nous avons utilisé les identités de Newton comme un ensemble de conditions *nécessaires* vérifiées par les fonctions puissances et symétriques des localisateurs des mots de poids minimal d'un code cyclique. Le même principe a permis aussi de prouver qu'il n'existait pas de mots de poids minimal dans certains codes BCH, car le système des équations de Newton est contradictoire. Ce travail a été mené avec P. Charpin et N. Sendrier, et a permis de déterminer complètement la table des codes BCH de longueur 255 donnée dans [WS86][table 9.1, page 267].

3 Les équations de Newton : un système suffisant

Le principal apport de cette thèse est de montrer que les solutions *algébriques* du système des équations de Newton sont de "bonnes" solutions : s'il existe des solutions au système algébrique défini par les équations de Newton, alors le problème posé dans le domaine des codes correcteurs admet une solution.

Dans le premier chapitre, nous étudions les identités de Newton, principalement pour formuler un système d'équations algébriques, que nous appellons le *système des équations de Newton*. Nous définissons des problèmes sur les corps finis, que nous appellons *problèmes des*

fonctions puissances. Ceci nous permet de conduire une courte étude théorique sans faire intervenir les codes correcteurs d'erreurs. Nous présentons alors les principales propriétés des systèmes des équations de Newton ainsi définis. Principalement, si le système des équations de Newton admet une solution, alors le problème des fonctions puissances admet une solution. Déterminer si un système d'équations polynômes admet des solutions peut être fait algorithmiquement au moyen du calcul de *bases standards*, que nous introduisons brièvement, pour présenter au lecteur les notions qui environnent le calcul effectif de solutions de systèmes algébriques.

Dans le deuxième chapitre, nous introduisons brièvement les codes correcteurs cycliques, et nous montrons le lien entre le problème de l'existence de mots de poids minimal dans des codes cycliques et les problèmes de fonctions puissances définis au premier chapitre. Ainsi nous transformons un problème difficile, le problème de l'existence de mots de poids donné dans un code donné, en un problème (peut être trop) difficile, le problème du calcul des solutions d'un système d'équations algébriques. Toutefois, si le système des équations de Newton est "résolu" par la méthode du calcul d'une base standard, des informations pertinentes sur les mots de poids minimal peuvent être obtenues. Nous présentons quelques exemples d'étude.

Dans ce même chapitre 2, nous utilisons une méthode employée par T. Schaub [Sch88], pour déterminer une borne inférieure sur la distance minimale d'un code cyclique. Ceci nous est très utile, car le système algébrique des équations de Newton est défini pour un poids déterminé. Pour éviter d'étudier des systèmes superflus, il est nécessaire d'écrire le système des équations de Newton pour un poids aussi proche que possible de la distance minimale du code étudié. Il nous faut disposer d'une borne très fine sur la distance minimale du code étudié, et la méthode de Schaub donne de très bons résultats. Appliquée en particulier sur les duaux des codes BCH, il apparaît que la distance minimale de ceux-ci est très au-dessus des bornes habituelles (voir les tables II.2, II.3 et II.4).

4 Une étude théorique

Dans le troisième chapitre, nous déterminons complètement l'ensemble des mots de poids minimal des codes BCH de longueur $2^m - 1$ de distance construite $2^{m-2} - 1$. L'outil est le système des équations de Newton, et il possible de déterminer d'une manière générale la forme que prennent certaines équations. La preuve du résultat est longue et difficile, et l'idée de la démonstration peut sembler tombée du ciel. Elle ne l'est pas, et en réalité, ce sont des études conduites avec des outils de calcul formel (Maple, Axiom), qui ont permis de constater quelle forme prenait certaines équations de Newton. La preuve du théorème consistait, en quelque sorte, à comprendre pourquoi l'ordinateur produisait des équations de Newton très creuses.

5 Calcul de bases normales

La notion de base normale est une notion importante intervenant en théorie des corps finis. Dans le chapitre 4, nous nous posons le problème de déterminer une base normale d'un corps fini. Nous rappelons le commentaire de J. von Zur Gathen et M. Giesbrecht [vZGG90] :

> We think that Lenstra's comment "Although the algorithms presented in this [Lenstra's] paper are not necessarily inefficient, I do not expect that in practice they can compete with the probabilistic algorithms..." also applies to the methods of Section 4 [de [vZGG90]], and expect methods avoiding linear algebra [...] to perform better in practice.

Nous employons précisément des techniques d'algèbre linéaire pour calculer une base normale sur $\mathbf{F}_{q^m}$, et nous obtenons ainsi la meilleure complexité *déterministe* pour résoudre ce problème, et notre méthode rivalise avec des méthodes probabilistes, lorsqu'on se contente d'employer des méthodes naïves de multiplication de polynômes.

Le calcul d'une base normale n'a pas été notre seul souci, et nous présentons aussi une méthode de calcul du polynôme minimal d'une matrice. Si la factorisation du polynôme caractéristique est connue, alors nous disposons aussi d'une méthode dont la complexité du pire est $O(n^{3.5})$. Enfin nous nous posons le problème du calcul de la forme rationnelle canonique d'une matrice, et lorsque la factorisation du polynôme caractéristique est connue, la complexité obtenue est meilleure que celle de P. Ozello dans sa thèse [Oze87].

6 Caractéristique première en Axiom

Un brève étude est conduite en appendice B, pour montrer un problème d'implantation en Axiom. La librairie d'Axiom est fort bien fournie en ce qui concerne les corps finis. Ceux-ci sont disponibles avec différentes implantations : la représentation usuelle est la représentation *polynomiale* ; pour les corps finis de petite taille la représentation par le groupe cyclique est disponible, et c'est la plus efficace ; enfin la représentation vectorielle, au moyen d'une base normale, existe aussi en Axiom.

Toutefois, il existait un problème pour élever à une puissance élevée des polynômes à coefficients dans un corps fini, l'algorithme utilisé par défaut ne s'avérant pas très performant. Nous présentons comment résoudre le problème. La difficulté n'est pas algorithmique, car l'algorithme est très simple, mais plutôt de conception. En Axiom, le principe général est de dégager le plus haut degré de généralité avant de définir une implantation, un type, ou un algorithme. Avec la librairie actuelle, définir une implantation générique n'était pas possible, principalement parce que les catégories traitant de la caractéristique sont trop faibles. Il faut donc définir une nouvelle catégorie, et c'est ce que nous présentons dans cet appendice B.

Chapitre I

Identités de Newton et systèmes algébriques

Dans ce chapitre, nous introduisons les identités de Newton reliant entre elles les fonctions puissances élémentaires et les fonctions symétriques élémentaires (partie 1) de w indéterminées. Ces identités se définissent dans un contexte plus général lorsque les fonctions puissances ne sont plus symétriques mais simplement une combinaison linéaire des indéterminées à la même puissance (*fonctions puissances généralisées*). Nous présentons aussi ces identités dans le cas d'un corps fini, cas qui nous intéressera dans toute la suite (partie 1.4). A ces présentations est jointe une formulation matricielle, mettant en avant les propriétés linéaires de ces identités.

Notre travail consiste à considérer ces identités, non plus comme des relations, mais comme un système d'équations algébriques dont les inconnues sont les fonctions puissances et les fonctions symétriques. Cette présentation est effectuée dans la partie 2 de ce chapitre, où nous formalisons une série de problèmes *de fonctions puissances* dans les corps finis, que nous notons $\mathcal{PG}_{i_1,\ldots,i_l}(w)$. Nous montrons ensuite un moyen d'étude de ces problèmes, et que le système *algébrique*, noté $\mathcal{S}_{i_1,\ldots,i_l}(w)$, défini par les identités de Newton, permet d'étudier le problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$. Nous utilisons essentiellement le fait que les identités de Newton forment un système de conditions nécessaires vérifiées par les solutions des problèmes de fonctions puissances. Nous introduisons aussi la tranformée de Fourier d'un vecteur, et surtout le lien entre le poids d'un mot d'un mot et le rang d'une matrice circulante obtenue à partir de la transformée de Fourier de ce mot (théorème I.9).

Quelques propriétés de ces systèmes algébriques sont établies en 2.3, théorème I.13. Principalement, notre apport est de montrer comment le système des équations de Newton $\mathcal{S}_{i_1,\ldots,i_l}(w)$ est aussi un système *suffisant*, c'est-à-dire que les solutions de ce système sont toutes des fonctions puissances de solutions de $\mathcal{PG}_{i_1,\ldots,i_l}(w)$. Nous déterminons ainsi complètement la nature des solutions des systèmes $\mathcal{S}_{i_1,\ldots,i_l}(w)$. L'étude du problème de fonctions puissances est équivalente à l'étude du système algébrique. Pour aborder cette étude, nous rappelons brièvement dans la troisième partie de ce chapitre les notions de base standard, et de réduction, qui seront pour nous un outil utile de caractérisation d'idéaux, pour déterminer des propriétés ensemblistes de ces idéaux, comme l'existence ou non de solutions, et le nombre de solutions.

1 Identités de Newton

Dans un premier temps, nous donnons la formulation habituelle des identités de Newton.

1.1 Présentation des équations

Définition I.1 *Soit* $\mathbf{k}$ *un corps,* $\mathbf{X} = (X_i)_{i=1\ldots w}$*, w indéterminées. Pour tout $k \geq 0$, la k-ième* fonction symétrique élémentaire *des X_i, dénotée σ_k, $\sigma_k \in \mathbf{k}[X_1, \ldots, X_w]$, est :*

$$\sigma_k = (-1)^k \sum_{1 \leq i_1 < \ldots < i_k \leq w} X_{i_1} X_{i_2} \ldots X_{i_k}, \qquad k \in [1, w],$$

et, pour $k = 0$, σ_0=1.

Les fonctions puissances élémentaires*, pour $k \geq 0$, notées A_k sont :*

$$A_k = \sum_{i=1}^{w} X_i^k, \qquad A_0 = w. \tag{I.1}$$

Il s'agit de deux familles de fonctions symétriques, où, pour tout k, σ_k et A_k sont des polynômes homogènes de degré k. La définition usuelle des fonctions symétriques élémentaires est

$$\sigma_i = \sum_{1 \leq i_1 < \ldots < i_k \leq w} X_{i_1} X_{i_2} \ldots X_{i_k}, \qquad k \in [1, w],$$

mais la définition adoptée ici évite de manipuler les signes dans les relations entre coefficients et racines, et dans les identités de Newton.

Théorème I.1 *Soit* $\mathbf{k}$ *un corps,* $\mathbf{X} = (X_i)_{i=1\ldots w}$*, w indéterminées, soient $\sigma_1, \ldots, \sigma_w$ les fonctions symétriques des X_i, et A_i, $i \geq 1$, les fonctions puissances des X_i. Pour tout entier i, $i \geq 1$, on a*

$$A_i + \sum_{k=1}^{i-1} \sigma_k A_{i-k} + i\sigma_i = 0, \qquad i \leq w, \tag{I.2}$$

$$A_i + \sum_{k=1}^{w} \sigma_k A_{i-k} = 0, \qquad i > w. \tag{I.3}$$

Ces identités sont appelées les identités de Newton*.*

Preuve : On considère le polynôme $p(Z) = \prod_{i=1}^{i=w}(1 - X_i Z) \in \mathbf{k}(X_1, \ldots, X_n)[Z]$. La dérivée logarithmique de $p(Z)$ donne :

$$\frac{p'(Z)}{p(Z)} = \sum_{i=1}^{w} \frac{-X_i}{1 - X_i Z}. \tag{I.4}$$

Or le développement de $p(Z)$ est $p(Z) = \sum_{i=0}^{w} \sigma_i Z^i$, et le développement en série formelle du deuxième terme de (I.4) est égal à :

$$\begin{aligned} \frac{p(Z)'}{p(Z)} &= \sum_{k=1}^{w} \frac{-X_k}{1 - X_k Z} = \sum_{k=1}^{w} -X_k \sum_{l=0}^{\infty} X_k^l Z^l = \sum_{l=0}^{\infty} \sum_{k=1}^{w} -X_k^{l+1} Z^l \\ &= \sum_{l=0}^{\infty} -A_{l+1} Z^l \end{aligned}$$

En écrivant :

$$p'(Z) = (\sum_{l=0}^{\infty} -A_{l+1}Z^l)p(Z) = \sum_{n=0}^{\infty}(\sum_{l+k=n} -A_{l+1}\sigma_k)Z^n,$$

et en identifiant, on obtient les relations suivantes :

$$\begin{aligned}(i+1)\sigma_{i+1} &= -\sum_{l+k=n} A_{l+1}\sigma_k \qquad i < w,\\ 0 &= \sum_{l+k=n} A_{l+1}\sigma_k \qquad i \geq w.\end{aligned}$$

□

N'étant pas un historien des mathématiques, je ne saurais présenter un aperçu bibliographique des relations de Newton. Ces propriétés semblaient être connues au 17-ème siècle, où les relations entre coefficients et racines sont dégagées. Rappelons qu'à cette époque, on avançait qu'un polynôme de degré n a n racines (éventuellement "impossibles"), comptées avec multiplicité (notes historiques dans [Bou81, Bou84]). A. Lascoux [Las86] semble attribuer même ces relations à A. Girard. Annick Valibouze m'a indiqué une référence de Lagrange, "Notes sur la théorie des équations algébriques", où l'auteur déclare : "Newton, et longtemps avant lui Albert Girard, avaient donné la manière de déterminer la somme des puissances des racines d'une équation par des fonctions de ses coefficients", faisant référence à [Gir29]. Cependant, nous en resterons à la terminologie de relations de Newton.

Plus récemment, les relations de Newton interviennent d'un point de vue algorithmique, dans différents domaines. Je citerai le problème du calcul du polynôme caractéristique d'une matrice, par la méthode de Leverrier, améliorée par Faddeev (voir [Gan77]). Les fonctions puissances peuvent servir comme famille génératrice des fonctions symétriques, pour manipuler celles-ci [Val87], ou pour modéliser la clôture algébrique d'un corps, en manipulant les fonctions puissances élémentaires au lieu des fonctions symétriques élémentaires (qui sont les coefficients des polynômes minimaux) [DT87, Wee90, AV92]. Les identités de Newton montrent comment déterminer simplement les fonctions symétriques élémentaires avec les fonctions puissances, en caractéristique zéro, ou tout du moins lorsque la caractéristique du corps est supérieure au nombre w d'indéterminées. Tel ne sera pas le cas dans notre étude, où nous considérons un nombre élevé d'indéterminées, pour des corps de petite caractéristique, presque toujours 2.

Notation 1 *Nous notons P_i le polynôme*

$$A_i + \textstyle\sum_{k=1}^{i-1} \sigma_k A_{i-k} + i\sigma_i, \quad si\ i \leq w, \tag{I.5}$$

$$A_i + \textstyle\sum_{k=1}^{w} \sigma_k A_{i-k}, \qquad si\ i > w, \tag{I.6}$$

de sorte que P_i est homogène de degré i en les X_i. Pour la suite nous aurons besoin de pouvoir référer à une équation donnée. Nous les numérotons ainsi :

$$\begin{aligned} id_1 :&\quad A_1 + \sigma_1 = P_1 = 0,\\ id_2 :&\quad A_2 + \sigma_1 A_1 + 2\sigma_2 = P_2 = 0,\\ &\quad\ \vdots\\ id_i :&\quad A_i + \sigma_1 A_{i-1} + \cdots + \sigma_w A_{i-w} = P_i = 0.\end{aligned}$$

1.2 Formulation matricielle

Soit T (comme triangulaire) la matrice :

$$T = \begin{pmatrix} A_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_2 & A_1 & 2 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & \\ A_i & A_{i-1} & \cdots & A_1 & i & 0 & 0 & 0 \\ \vdots & & & & & & & \\ A_w & A_{w-1} & \cdots & \cdots & \cdots & \cdots & A_1 & w \end{pmatrix}$$

Et soit C_∞ la matrice en nombre infini de lignes :

$$C_\infty = \begin{pmatrix} A_{w+1} & A_{w+2} & \dots & A_1 \\ A_{w+2} & A_{w+3} & \dots & A_2 \\ \vdots & & & \\ A_i & A_{i-1} & \dots & A_{i-w} \\ \vdots & & & \end{pmatrix}$$

Alors les équations de Newton prennent la forme suivante :

$$\begin{pmatrix} T \\ C_\infty \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Nous dirons que le système :

$$T \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{I.7}$$

constitue la *forme triangulaire* des identités de Newton, et que l'équation suivante :

$$C_\infty \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{I.8}$$

est la *forme circulante* des identités de Newton.

1.3 Identités de Newton généralisées

On s'intéresse au cas où les fonctions puissances ne sont pas seulement la somme des puissances des indéterminées, mais une combinaison linéaire de ces puissances.

Définition I.2 *Soit* $\mathbf{k}$ *un corps,* $X_1, \ldots, X_w$*,* w *indéterminées. Soit* $(a_1, \ldots, a_w) \in \mathbf{k}^w$*, et soit :*

$$A_k = \sum_{i=1}^{w} a_i X_i^k, \qquad k \geq 0.$$

Par définition, les A_k *sont les* fonctions puissances généralisées des X_i, relativement aux a_i*.*

Théorème I.2 *Soit* $\mathbf{k}$ *un corps,* $X_1, \ldots, X_w$*,* w *indéterminées. Soit* $(a_1, \ldots, a_w) \in \mathbf{k}^n$*. Les fonctions puissances généralisées des* X_i *relativement aux* a_i *et les fonctions symétriques élémentaires vérifient les relations suivantes :*

$$\forall j \geq 0,\ A_{j+w} + \sigma_1 A_{j+w-1} + \cdots + \sigma_k A_{j+w-k} + \cdots + \sigma_w A_j = 0.$$

Ces relations sont appelées les identités de Newton généralisées*.*

Preuve : [WS86] On a :

$$\prod_{i=1}^{w} (1 - X_i Z) = 1 + \sigma_1 Z + \cdots + \sigma_w Z^w.$$

En posant $Z = 1/X_i$, et en multipliant par $a_i X_i^{j+w}$, on a :

$$a_i X_i^{j+w} + \sigma_1 a_i X^{j+w-1} + \cdots + \sigma_w a_i X^j = 0.$$

En sommant pour i de 1 à w, on obtient :

$$\sum_{i=0}^{w} a_i X_i^{j+w} + \sum_{i=0}^{w} \sigma_1 a_i X^{j+w-1} + \ldots + \sum_{i=0}^{w} \sigma_w a_i X^j = 0.$$

Ce qui est bien la relation désirée.

□

On remarque que les identités prennent la même forme, sauf qu'on ne retrouve pas la forme triangulaire des premières équations. Il ne reste que la forme circulante :

$$C_\infty \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Les équations de Newton définissent une relation de récurrence pour les fonctions puissances généralisées. Les fonctions symétriques σ_i ne s'expriment plus en fonction des fonctions puissances A_i, ce qui était prévisible car les fonctions puissances sont symétriques, alors que les fonctions puissances généralisées ne le sont pas.

1.4 Les identités de Newton sur un corps fini

Notation 2 *Nous réserverons pour toute la suite les notations suivantes :*

- $\mathbf{k} = \mathrm{F}_q$ *est le corps fini à* q *éléments, où* $q = p^m$ *avec* p *premier.*
- w *est un entier* > 0.
- $a_1, \ldots, a_w$ *sont* w *éléments de* $\mathbf{k}$.
- $\mathbf{F}$ *est une extension de* $\mathbf{k}$ *:* $\mathbf{F} = \mathrm{F}_{q^{m'}}$. *Nous notons* n *le cardinal du groupe multiplicatif de* $\mathbf{F}$ *:* $n = q^{m'} - 1$.
- $X_1, \ldots, X_w$ *sont* w *éléments de* $\mathbf{F}$.

Nous supposons le lecteur familier avec les corps finis, et ne rappelons pas les propriétés des corps finis. L'ouvrage de MacEliece [Eli87] constitue à ce sujet une excellente introduction, et la somme de Lidl et Niederreiter [LN83] est une référence en ce domaine.

1.4.1 Forme des équations

Théorème I.3 *Soient* $(X_1, \ldots, X_w) \in \mathbf{F}^w$, *soient* $\sigma_1, \ldots, \sigma_w$ *les fonctions symétriques élémentaires des* X_i, *et* A_k, $k \geq 1$, *les fonctions puissances des* X_i.

Les fonctions puissances A_i *et les fonctions symétriques élémentaires* σ_i *sont liées par les relations suivantes :*

$$\begin{aligned}
id_1 &: \quad A_1 + \sigma_1 \equiv 0 \\
id_2 &: \quad A_2 + \sigma_1 A_1 + 2\sigma_2 \equiv 0 \\
id_i &: \quad A_i + \sigma_1 A_{i-1} + \cdots + i\sigma_i \equiv 0, \qquad si\ p \not| \ i,\ i \leq w \\
id_i &: \quad A_i + \sigma_1 A_{i-1} + \cdots + A_1 \sigma_{i-1} \equiv 0, \qquad si\ p \mid i,\ i \leq w \\
id_{w-1} &: \quad A_{w-1} + \sigma_1 A_{w-2} + \cdots + \sigma_{w-2} A_1 + (w-1)\sigma_{w-1} \equiv 0 \\
id_w &: \quad A_w + \sigma_1 A_{w-1} + \cdots + \sigma_{w-1} A_1 + w\sigma_w \equiv 0,
\end{aligned}$$

et pour $w < k \leq n + w$ *:*

$$A_k + \cdots + \sigma_i A_{k-i} + \cdots + \sigma_w A_{k-w} \equiv 0.$$

Soient $(a_1, \ldots, a_w) \in \mathbf{k}^w$, *les identités des Newton généralisées entre les fonctions puissances généralisées* A'_i *des* X_i *relativement aux* a_i *et les fonctions symétriques deviennent*

$$A'_k + \cdots + \sigma_i A'_{k-i} + \cdots + \sigma_w A'_{k-w} \equiv 0, \qquad k \in [w+1, w+n].$$

Nous employons le symbole $\equiv$ pour indiquer l'effet de la caractéristique p.

Preuve : Si la caractéristique p est inférieure à w, et si p divise i, alors le terme $i\sigma_i$ dans l'identité id_i disparaît. De plus, étant donné la propriété :

$$\forall x \in \mathbf{F}, x^{n+1} = x$$

on a $A_{i+n} = A_i$, et l'équation $id_{n+i}, i \geq w$ se réduit à :

$$\begin{aligned} id_{n+i} &= A_{n+i} + A_{n+i-1}\sigma_1 + A_{n+i-2}\sigma_2 + \cdots + A_{n+i-w}\sigma_w \\ &= A_i + A_{i-1}\sigma_1 + A_{i-1}\sigma_2 + \cdots + A_{n-w}\sigma_w \\ &= id_i. \end{aligned}$$

Donc le système en nombre infini d'équations se réduit aux seules équations $(id_i), 1 \leq i \leq n+w$, dans le cas des identités de Newton, et à aux équations $(id_i), w < i \leq n+w$ dans le cas des identités de Newton généralisées.

□

On voit que les premières équations de Newton, pour $1 \leq i \leq w$ perdent leur forme triangulaire, si la caractéristique p est inférieure à w, et que ce système n'est plus nécessairement résoluble en les σ_i.

Du point de vue matriciel, les identités de Newton prennent la forme

$$\begin{pmatrix} T \\ C_{n,w} \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

où $C_{n,w}$ est la matrice :

$$\begin{pmatrix} A_{w+1} & A_{w+2} & \dots & A_1 \\ A_{w+2} & A_{w+3} & \dots & A_2 \\ \vdots & & & \\ A_i & A_{i-1} & \dots & A_{i-w} \\ \vdots & & & \\ A_{n+w} & A_{n+w-1} & \dots & A_n \end{pmatrix}.$$

Dans le cas des identités de Newton généralisées, le système se réduit à :

$$\begin{pmatrix} C_{n,w} \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

1.4.2 Cas de la caractéristique 2

Théorème I.4 *Soit* $\mathbf{k}$ *un corps de caractéristique* 2. *Pour* $r \geq 1$, *le polynôme* P_{2r} *est dans l'idéal engendré par* $(P_1, P_3, \dots, P_{2r-1})$, $i = 0 \dots r-1$. *Le système des équations de Newton est équivalent au système des équations de numéro pair :*

$$\begin{pmatrix} T' \\ C'_{n,w} \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{I.9}$$

où T'_n et I'_n sont les sous-matrices extraites de T et de I_n en ne considérant que les lignes d'indice pair.

Preuve : Nous allons montrer que le polynôme P_{2r} s'écrit comme combinaison des polynômes $P_1, \dots P_{2r-1}$ de la manière suivante :

$$P_{2r} = A_{2r-1}P_1 + A_{2r-2}P_2 + \cdots A_1 P_{2r-1}.$$

Dans le terme de droite, le coefficient C_{2j} de σ_{2j} est

$$\begin{aligned} C_{2j} &= A_{2r-(2j+1)}A_1 + A_{2r-(2j+1)-1}A_2 + \cdots + A_1 A_{2r-1-2j} \\ &= \sum_{i=1}^{r-j-1} A_{2r-2j-i}A_i + A_{2r-2j-(r-j)}A_{r-j} + \sum_{i=r-j+1}^{2r-2j-1} A_{2r-2j-i}A_i \\ &= \sum_{i=1}^{r-j-1} A_{2r-2j-1}A_i + A_{r-j}^2 + \sum_{i=1}^{r-j-1} A_i A_{2r-2j-i} = A_{2(r-j),} \end{aligned}$$

et le coefficient C_{2j+1} de σ_{2j+1} est

$$\begin{aligned} C_{2j+1} &= A_{2r-(2j+1)} + A_{2r-(2j+1)-1}A_1 + A_{2r-(2j+1)-2}A_2 + \cdots + A_1 A_{2r-(2j+1)-1} \\ &= A_{2r-(2j+1)} + \sum_{i=1}^{r-j} A_{2r-(2j+1)-i}A_i + \sum_{i=r-j+1}^{2r-2j} A_{2r-(2j+1)-i}A_i \\ &= A_{2r-2j+1} + \sum_{i=1}^{r-j} A_{2r-(2j+1)-i}A_i + \sum_{i=1}^{r-j} A_{2r-(2j+1)-i}A_i \\ &= A_{2r-(2j+1)}. \end{aligned}$$

Ce qui donne bien les coefficients de P_{2r}. Il reste

$$A_{2r-1}A_1 + A_{2r-2}A_2 + \cdots + A_r A_r + \cdots + A_1 A_{2r-1} = A_r^2 = A_{2r},$$

les termes s'annulant deux à deux, sauf $A_r A_r$.

□

Nous pensons que dans le cas de la caractéristique p, le polynôme P_{kp} est dans l'idéal engendré par les polynômes $P_1, \dots, P_{kp-1}$. Nous n'avons pu le montrer, et la preuve semble beaucoup plus technique que celle-ci.

2 Le système algébrique défini par les équations de Newton

Le lien entre ce chapitre I et la théorie des codes correcteurs d'erreurs sera établie dans le chapitre II. Les problèmes que nous étudierons en théorie des codes correcteurs d'erreurs sont formulés ici indépendamment de celle-ci, et nous introduisons, dans cette partie, ce que nous appelons les problèmes des fonctions puissances, et notons $\mathcal{PG}_{i_1,\dots,i_l}(w)$.

L'existence de solutions algébriques aux équations de Newton est d'abord introduite comme une condition *nécessaire* à l'existence de solutions au problème $\mathcal{PG}_{i_1,\dots,i_l}(w)$ (théorème I.11). Nous définissons clairement le système $\mathcal{S}_{i_1,\dots,i_l}(w)$ associé au problème $\mathcal{PG}_{i_1,\dots,i_l}(w)$, et nous étudions les solutions de ces systèmes algébriques, pour les déterminer complètement.

Nous rappelons le cadre dans lequel nous nous plaçons :

- $\mathbf{k} = \mathrm{F}_q$ est le corps fini à q éléments, où $q = p^m$ avec p premier.
- w est un entier > 0.
- $a_1, \ldots, a_w$ sont des éléments de $\mathbf{k}$.
- $\mathbf{F}$ est une extension de k : $\mathbf{F} = \mathrm{F}_{q^{m'}}$. Nous notons n le cardinal du groupe multiplicatif de $\mathbf{F}$: $n = q^{m'} - 1$.

Bien que ce soit le problème des équations de Newton pour un poids donné qui nous intéresse, nous formulons aussi des problèmes plus généraux, qui nous serviront d'un point de vue théorique.

2.1 Le problème des fonctions puissances généralisées, de poids indéterminé

Définition I.3 *Nous appelons* problème des fonctions puissances généralisées*, le problème posé par $A_{i_1} = A_{i_2} = \cdots = A_{i_l} = 0$, où les A_i sont les fonctions puissances généralisées des inconnues $(X_1, \ldots, X_w) \in \mathbf{F}^w$, relativement à $(a_1, \ldots, a_w) \in \mathbf{k}^w$.*

1. *w est inconnu,*
2. *les a_i sont inconnus.*

Nous notons ce problème $\mathcal{PG}_{i_1,\ldots,i_l}$.

Une solution du problème *est la donnée d'un entier $w > 0$, de $(a_1, \ldots, a_w) \in \mathbf{k}^w$, et de $(X_1, \ldots, X_w) \in \mathbf{F}^w$, tels que les fonctions puissances généralisées des X_i relativement aux a_i vérifient $A_{i_1} = \ldots = A_{i_l} = 0$. Nous supposons que les a_i et les X_i, $i \in [1, w]$, sont tous différents de 0, afin d'éliminer des solutions redondantes évidentes. Nous dirons que $(\mathbf{k}, \mathbf{F})$ est le* contexte *du problème, et que w est le* poids *de la solution.*

Nous montrerons comment traiter ce problème : pour chaque ensemble de valeurs des fonctions puissances, sous certaines conditions (elles sont *closes* par classes cyclotomiques), il existe une solution $(w, a_1, \ldots, a_w, X_1, \ldots, X_w)$, qui vérifie $\mathcal{PG}_{i_1 \ldots i_l}$. Il s'agit d'une transformée de Fourier inverse. Avant d'étudier cette transformation, énonçons des conditions nécessaires à l'existence d'une solution.

2.1.1 Classes cyclotomiques

Proposition I.1 *Soit w un entier, $(a_1, \ldots a_w) \in \mathbf{k}^w$, $(X_1, \ldots X_w) \in \mathbf{F}^w$, les fonctions puissances des $X_1, \ldots, X_w$, relativement aux a_i, vérifient :*

$$\forall\ i\ \in [0, n-1],\ A_{qi \bmod n} = A_i^q. \tag{I.10}$$

Preuve : Les a_i étant dans $\mathbf{k} = \mathrm{F}_q$ et les X_i dans $\mathbf{F}$, nous avons :

$$\begin{aligned} A_i^q &= \left(\sum_{k=1}^{w} a_k X_k^i\right)^q = \sum_{k=1}^{w} a_k X_k^{iq} = \sum_{k=1}^{w} a_k X_k^{iq \bmod n} \\ &= A_{iq \bmod n}. \end{aligned}$$

En effet, les X_i étant dans $\mathbf{F} \setminus \{0\}$, ils vérifient tous $X_i^n = 1$.

□

Ceci conduit à la définition suivante :

Définition I.4 *Soit p un nombre premier, soit n un entier positif premier à p et soit $q = p^m$, les* classes cyclotomiques *de q modulo n sont les ensembles :*

$$cl(i) = \{i \bmod n, iq \bmod n, \ldots, iq^j \bmod n, \ldots\}, \qquad i \in [0, n-1].$$

Les classes cyclotomiques de q modulo n définissent une partition de $[0, n-1]$.

Nous dirons qu'un n-uplet $(\gamma_0, \ldots, \gamma_{n-1}) \in \mathbf{F}^n$ est clos pour les classes cyclotomiques *si $\gamma_i^q = \gamma_{qi \bmod n}$, $i \in [0, n-1]$.*

Exemple 1 Dans le contexte $\mathbf{k} = \mathrm{F}_2$, $\mathbf{F} = \mathrm{F}_{16}$, $n = 15$, le système $\mathcal{PG}_{1,2,4,8}$ est défini, car $cl(1) = \{1, 2, 4, 8\}$.

Les solutions du système $\mathcal{PG}_{1,2,4,8,3}$ vérifieront $A_1 = A_2 = A_4 = A_8 = 0$ et $A_3 = A_6 = A_{12} = A_9 = 0$, car $cl(3) = \{3, 6, 12, 9\}$. Le problème $\mathcal{PG}_{1,2,4,8,3}$ admet les mêmes solutions que le système $\mathcal{PG}_{1,2,4,8,3,6,12,9}$.

Notation 3 *Puisque la nullité de A_i entraîne la nullité de $A_{iq \bmod n}$, et que les problèmes $\mathcal{PG}_{i_1,\ldots,i_l}$ et $\mathcal{PG}_{cl(i_1)\cup\ldots\cup cl(i_l)}$ ont les mêmes solutions, nous emploierons la notation $\mathcal{PG}_{i_1,\ldots,i_l}$ pour le système $\mathcal{PG}_{cl(i_1)\cup\ldots\cup cl(i_l)}$.*

2.1.2 Les solutions par transformation de Fourier

Nous introduisons la tranformée de Fourier d'une séquence $a = (a_0, \ldots, a_{n-1})$. La terminologie habituelle en théorie des codes est celle de *polynôme de Mattson-Solomon.*

Notation 4 *Dorénavant, pour tout corps fini $\mathbf{k}$, et n premier à la caractéristique de $\mathbf{k}$, nous estimons fixée une racine primitive n-ième de l'unité, dans une extension convenable de $\mathbf{k}$. Nous la noterons α.*

Définition I.5 ([WS86, p. 239]) *Soit $a = (a_0, \ldots, a_{n-1}) \in \mathbf{k}^n$ et soit α un racine primitive n-ième de l'unité. Le* polynôme de Mattson-Solomon *$A(Z)$ de a est défini comme suit :*

$$A(Z) = \sum_{i=1}^{w} A_i Z^{n-i}$$

avec

$$A_i = a(\alpha^i) = \sum_{j=0}^{n-1} a_j \alpha^{ij}.$$

Théorème I.5 *Soit $a = (a_0, \ldots, a_{n-1}) \in \mathbf{k}^n$ et $A(Z)$ le polynôme de Mattson-Solomon de a. Alors on a :*

$$na_i = A(\alpha^i), \qquad i \in [0, n-1].$$

L'application $a \mapsto A(X)$ est injective.

Preuve : Calculons $A(\alpha^i)$:

$$\begin{aligned} A(\alpha^i) &= \sum_{j=1}^{n} A_j \alpha^{i(n-j)} \\ &= \sum_{j=1}^{n} \sum_{k=0}^{n-1} a_k \alpha^{kj} \alpha^{-ij} \\ &= \sum_{k=0}^{n-1} a_k \sum_{j=1}^{n} \alpha^{j(k-i)} \\ &= n a_i \end{aligned}$$

Car $\sum_{i=1}^{n} \alpha^{ik}$ est égal à n si $k = 0$, à 0 si $k \neq 0$.

□

Définition I.6 *Soit $a = (a_0, \ldots, a_{n-1}) \in \mathbf{k}^n$, et soit α une racine primitive n-ième de l'unité. Soient $X_1, \ldots, X_w$ définis par*

$$\{X_1, \ldots, X_w\} = \{\alpha^i,\ i \in [0, n-1] \mid\ a_i \neq 0\}.$$

Par définition, $X_1, \ldots, X_w$ sont les localisateurs *de a (l'ordre d'indexation importe peu). Pour le localisateur X_j de a, nous notons a_{j_i} la coordonnée correspondante non nulle de a.*

Le théorème suivant est une réciproque de la proposition I.1.

Théorème I.6 *Soit un n-uplet $(\gamma_0, \ldots, \gamma_{n-1}) \in \mathbf{F}^n$, clos par classes cyclotomiques, et vérifiant $\gamma_{i_1} = \cdots = \gamma_{i_l} = 0$. Au n-uplet $(\gamma_0, \ldots, \gamma_{n-1})$ correspond au moins une solution w, $(X_1, \ldots, X_w)$, $(a_1, \ldots, a_w)$ de $\mathcal{PG}_{i_1,\ldots,i_l}$, cette solution étant obtenue par transformée de Fourier inverse.*

Preuve : Soient donné $(\gamma_0, \ldots, \gamma_{n-1}) \in \mathbf{F}^n$, vérifiant $\gamma_{i_1} = \cdots = \gamma_{i_l} = 0$, clos par classes cyclotomiques. Nous formons le polynôme :

$$\gamma(Z) = \sum_{i=0}^{n-1} \gamma_i Z^{n-i}$$

et nous posons $a_i = \gamma(\alpha^i)$, pour $i \in [0, n-1]$. Alors, pour tout $i \in [0, n-1]$, nous avons $a_i \in \mathbf{k} (= \mathrm{F}_q)$, puisque

$$\begin{aligned} a_i^q &= \left(\frac{1}{n} \sum_{k=0}^{n-1} \gamma_k (\alpha^i)^{n-k} \right)^q = \frac{1}{n} \sum_{k=0}^{n-1} \gamma_k^q \alpha^{qi(n-k)} \\ &= \frac{1}{n} \sum_{k=0}^{n-1} \gamma_{qk \bmod n} (\alpha^i)^{n-qk} = \frac{1}{n} \sum_{k'=0}^{n-1} \gamma_{k'} (\alpha^i)^{n-k'} \\ &= a_i. \end{aligned}$$

Notons X_i, $i = 1, \ldots, w$ les localisateurs de $a = (a_0, \ldots, a_{n-1})$. Soit maintenant A'_i, $i = 1 \ldots n$, les fonctions puissances de $X_1, \ldots, X_w$ relativement aux $a_{j_1}, \ldots, a_{j_w}$, alors :

$$\begin{aligned} A'_k &= \sum_{j=0}^{w} a_{i_j} X_j^k = \sum_{i=0}^{n-1} a_i \alpha^{ik} \\ &= \gamma_k. \end{aligned}$$

et en particulier $A'_{i_1} = \cdots = A'_{i_l} = 0$.

□

Définition I.7 *Une solution* w, $(a_1, \ldots, a_w) \in \mathbf{k}^n$, $(X_1, \ldots, X_w) \in \mathbf{F}^n$ *est* multiple *si il existe* i, j, $i \neq j$ *tel que* $X_i = X_j$.

Propriété I.1 *Soit une solution multiple de* $\mathcal{PG}_{i_1,\ldots,i_l}$, *donnée par* w, $(a_1, \ldots, a_w) \in \mathbf{k}^w$, $(X_1, \ldots, X_w) \in \mathbf{F}^w$. *Il existe alors une solution non multiple de* $\mathcal{PG}_{i_1,\ldots,i_l}$ *définie par* $w' < w$, $a'_1, \ldots, a'_{w'}$, *et* $\{X'_1, \ldots, X'_{w'}\} \subset \{X_1, \ldots X_w\}$.

Preuve : En effet supposons $X_1 = X_2$, alors $w - 1$, $a'_1 = a_1 + a_2, a_3, \ldots, a_w$, $X_1, X_3, \ldots X_w$ est aussi une solution. Ainsi, en groupant les X_i communs, on arrive à une solution non multiple.

□

Théorème I.7 *Soit un* n*-uplet* $(A_0, \ldots, A_{n-1}) \in \mathbf{F}^n$, *clos par classes cyclotomiques, et vérifiant* $A_{i_1} = \cdots = A_{i_l} = 0$. *Au* n*-uplet* $(A_0, \ldots, A_{n-1})$ *correspond une unique solution non multiple* w, $(X_1, \ldots, X_w)$, $(a_1, \ldots, a_w)$ *de* $\mathcal{PG}_{i_1,\ldots,i_l}(w)$. *Précisément cette solution est celle obtenue par la transformée de Fourier inverse de la séquence* $(A_0 \ldots A_{n-1})$.

Le problème est de connaître les w pour lesquels il existe une solution w, $(a_1, \ldots a_w) \in \mathbf{k}^w$, $(X_1, \ldots X_w) \in \mathbf{F}^n$. Nous avons très exactement :

Théorème I.8 *Soit* $A = (A_0, \ldots, A_{n-1}) \in \mathbf{F}^n$, *clos par classes cyclotomiques, et soit* $A(Z)$ *le polynôme de Mattson-Solomon dont les coefficients sont les* A_i. *Soit la solution non multiple* w, $(a_1, \ldots a_w) \in \mathbf{k}^w$, $(X_1, \ldots X_w) \in \mathbf{F}^n$, *correspondant à* $A = (A_0, \ldots, A_{n-1})$. *Soit* r *le nombre de racines* n*-ième de l'unité qui sont racines de* $A(Z)$. *Alors* w *est égal à* $n - r$.

Preuve : D'après la propriété d'inversion de la transformation de Fourier (théorème I.5), tous les a_i sont obtenus par $a_i = A(\alpha^i)/n$, $i = 0, \ldots, n-1$. Donc les a_i non nuls sont pour les racines n-ièmes de l'unité α^i qui ne sont pas racines de $A(Z)$, soit $n - r$.

□

Par exemple, le nombre de coordonnées non nulles de a est supérieur ou égal à $n - \deg(A(Z))$.

2.2 Le problème des fonctions puissances généralisées de poids déterminé

Définition I.8 *Pour* $w > 0$, *nous appelons* problème des fonctions puissances généralisées de poids w *le problème posé par* $A_{i_1} = A_{i_2} = \cdots = A_{i_l} = 0$, *où les* A_i *sont les fonctions puissances généralisées de* w *inconnues* $(X_1, \ldots, X_w) \in \mathbf{F}^w$, *relativement à* $(a_1 \ldots, a_w) \in \mathbf{k}^w$.

1. *les* a_i *sont inconnus.*

Nous notons ce problème $\mathcal{PG}_{i_1,\ldots i_l}(w)$.

Une solution du problème *est la donnée de* $(a_1, \ldots a_w) \in \mathbf{k}^w$, $(X_1, \ldots X_w) \in \mathbf{k}^w$ *tels que les fonctions puissances généralisées des* X_i *relativement aux* a_i *vérifient* $A_{i_1} = \ldots = A_{i_l} = 0$. *Nous supposons que les* a_i *et les* X_i, $i = 1 \ldots w$, *sont tous différents de* 0, *afin d'éliminer des solutions redondantes évidentes. Nous dirons que* $(\mathbf{k}, \mathbf{F})$ *est le contexte du problème.*

Il est clair que ce problème est un sous problème du problème précédent, en ce sens qu'une solution de $\mathcal{PG}_{i_1,\dots i_l}(w)$ est une solution de $\mathcal{PG}_{i_1,\dots i_l}$. L'étude précédente peut s'appliquer, et une solution existe s'il existe un ensemble de valeurs de fonctions puissances clos par classes cyclotomiques, telle que le poids de la transformée de Fourier inverse est exactement w.

Il est cependant utile de pouvoir fixer le poids, car cela nous permet d'écrire un système d'équations.

2.2.1 Le poids de la solution

Si une solution $(a_1,\dots,a_w)$, $(X_1,\dots,X_w)$ de $\mathcal{PG}_{i_1,\dots i_l}(w)$ existe, en vertu des relations de Newton généralisées établies dans le paragraphe 1 de ce chapitre, les fonctions symétriques élémentaires de $X_1,\dots,X_w$, solutions du problème, vérifieront la forme circulante des équations de Newton :

$$\begin{pmatrix} A_{w+1} & A_{w+2} & \dots & A_1 \\ A_{w+2} & A_{w+3} & \dots & A_2 \\ \vdots & & & \\ A_{n+w} & A_{n+w-1} & \dots & A_n \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{I.11}$$

Le système I.11 montre que la rang de la matrice

$$\begin{pmatrix} A_{w+1} & A_{w+2} & \dots & A_1 \\ A_{w+2} & A_{w+3} & \dots & A_2 \\ \vdots & & & \\ A_{n+w} & A_{n+w-1} & \dots & A_n \end{pmatrix}$$

est inférieur à w. Le poids w est très lié au rang d'une matrice circulante définie par les A_i, comme le montre le théorème suivant.

Théorème I.9 *Soit $a=(a_0,\dots,a_{n-1})\in\mathbf{k}^n$. Le nombre de coordonnées non nulles de a est égal au rang de la matrice*

$$C = \begin{pmatrix} A_0 & A_1 & \dots & A_{n-2} & A_{n-1} \\ A_1 & A_2 & \dots & A_{n-1} & A_0 \\ \vdots & & & & \\ A_{n-1} & A_0 & \dots & A_{n-3} & A_{n-2} \end{pmatrix},$$

où les A_i sont les coefficients du polynôme de Mattson-Solomon de a.

Preuve : Les coefficients du polynôme de Mattson-Solomon peuvent se calculer matriciellement comme suit

$$\begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_{n-1} \end{pmatrix} = F \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix},$$

avec

$$F = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \alpha^1 & \dots & \alpha^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-1} & \dots & \alpha^{(n-1)(n-1)} \end{pmatrix}$$

De même

$$\begin{pmatrix} A_i \\ A_{i+1} \\ \vdots \\ A_{i+n-1} \end{pmatrix} = F \begin{pmatrix} a_0 \\ \alpha^i a_1 \\ \vdots \\ \alpha^{(n-1)i} a_n \end{pmatrix}.$$

D'où

$$\begin{pmatrix} A_0 & A_1 & \dots & A_{n-2} & A_{n-1} \\ A_1 & A_2 & \dots & A_{n-1} & A_0 \\ \vdots & & & & \\ A_{n-1} & A_0 & \dots & A_{n-3} & A_{n-2} \end{pmatrix} = F \begin{pmatrix} a_0 & a_0 & \dots & a_0 \\ a_1 & \alpha a_1 & \dots & \alpha^{n-1} a_1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{n-1} & \alpha^{n-1} a_{n-1} & \dots & \alpha^{(n-1)(n-1)} a_{n-1} \end{pmatrix}$$

$$= F \begin{pmatrix} a_0 & 0 & 0 & \dots & 0 \\ 0 & a_1 & 0 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & \dots & 0 & a_{n-1} \end{pmatrix} F$$

Le rang d'une matrice diagonale étant égal au nombre de termes non nuls sur la diagonale, le résultat s'obtient car les deux matrices sont équivalentes.

□

Théorème I.10 *Si l'ensemble $i_1, \dots, i_l$ contient $\delta - 1$ entiers consécutifs, alors le système $\mathcal{PG}_{i_1,\dots,i_l}(w)$ n'admet pas de solutions, si w est inférieur à δ.*

Preuve : Soit en effet une solution de $\mathcal{PG}_{i_1,\dots,i_l}(w)$, avec $[s, s+1, \dots, s+\delta-2] \subset \{i_1, \dots, i_l\}$ alors le poids de cette solution est égal au rang de la matrice

$$\begin{pmatrix} & & & \underbrace{\delta - 1 \text{ zéros}} & & & \\ A_{n-1} & \dots & A_{s+\delta-1} & 0 \dots 0 & A_{s-1} & \dots & A_0 \\ A_{n-2} & \dots & 0 & 0 \dots A_{s-1} & \dots & A_0 & A_{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}.$$

Si au moins un A_i est différent de zéro, alors les δ premières lignes d'une telle matrice sont linéairement indépendantes et le rang est supérieur ou égal à δ. Sinon tous les A_i sont nuls.

□

Remarque I.1 *Ce théorème est en fait le théorème de la borne BCH (théorème II.3), hors du contexte des codes correcteurs d'erreurs.*

2.2.2 Le système algébrique

Pour déterminer l'existence ou non de solutions au problème $\mathcal{PG}_{i_1,\dots,i_l}(w)$, nous considérerons le système des équations suivantes:

$$\begin{aligned} A_{w+1} + A_w\sigma_1 + \cdots A_2\sigma_{w-1} + A_1\sigma_w &= 0 \\ A_{w+2} + A_{w+1}\sigma_1 + \cdots A_3\sigma_{w-1} + A_1\sigma_w &= 0 \\ &\vdots \\ A_{n+w} + A_{n+w-1}\sigma_1 + \cdots A_{n+1}\sigma_{w-1} + A_n\sigma_w &= 0 \end{aligned}$$

Problème spécifique Résoudre le système *spécifique* pour des valeurs $A_i = \gamma_i \in \mathbf{F}$ closes par classes cyclotomiques revient à déterminer l'existence de σ_i solutions de

$$\begin{aligned} \gamma_w + \gamma_{w-1}\sigma_1 + \cdots \gamma_1\sigma_{w-1} + \gamma_0\sigma_w &= 0 \\ \gamma_{w+1} + \gamma_w\sigma_1 + \cdots \gamma_2\sigma_{w-1} + \gamma_1\sigma_w &= 0 \\ &\vdots \\ \gamma_{n+w} + \gamma_{n+w-1}\sigma_1 + \cdots \gamma_{n+1}\sigma_{w-1} + \gamma_n\sigma_w &= 0 \end{aligned}$$

auquel cas il s'agit de déterminer l'existence de solutions à un système linéaire.

L'existence de telles solutions est une condition *nécessaire* à l'existence de solutions au problème spécifique. Mais une solution $\sigma_1, \ldots, \sigma_w$ n'implique pas l'existence de solutions au problème. En effet, les σ_i étant les fonctions symétriques des X_i, le polynôme

$$\sum_{i=0}^{w} \sigma_{w-i} Z^i = \prod_{i=1}^{w} (Z - X_i)$$

admet comme racines les indéterminées X_i. Pour que celles-ci soient dans $\mathbf{F}$, il faut que le polynôme $\sum_{i=0}^{w} \sigma_{w-i} Z^i$ soit scindé dans $\mathbf{F}$. De plus, pour éviter des solutions multiples, il faut que ce polynôme soit à racines simples.

Si ce polynôme est scindé, pour déterminer complètement la solution, il s'agit de déterminer les coefficients a_i. Ceux-ci vérifient le système :

$$\left\{ \begin{aligned} a_1X_1 + a_2X_2 + \cdots + a_wX_w &= \gamma_1 \\ a_1{X_1}^2 + a_2{X_2}^2 + \cdots + a_w{X_w}^2 &= \gamma_2 \\ &\vdots \\ a_1{X_1}^w + a_2{X_2}^w + \cdots + a_w{X_w}^w &= \gamma_w \end{aligned} \right. \tag{I.12}$$

Dés lors que les X_i sont distincts, ce système est inversible car c'est un système de Vandermonde.

Problème générique Nous considérons de nouveau le système

$$\begin{aligned} A_{w+1} + A_w\sigma_1 + \cdots A_2\sigma_{w-1} + A_1\sigma_w &= 0 \\ A_{w+2} + A_{w+1}\sigma_1 + \cdots A_3\sigma_{w-1} + A_1\sigma_w &= 0 \\ &\vdots \\ A_{n+w} + A_{n+w-1}\sigma_1 + \cdots A_{n+1}\sigma_{w-1} + A_n\sigma_w &= 0 \end{aligned} \tag{I.13}$$

dans lequel seront introduites les conditions $A_{qi \bmod n} = A_i^q$ et $A_{i+n} = A_i$. En pratique, nous choisirons pour chaque classe cyclotomique le plus petit représentant i_0, et les A_i tels que i est dans la même classe cyclotomique que i_0 seront exprimés en tant que puissance de A_{i_0}. Si la classe cyclotomique de i se réduit à un seul élément, on introduit la condition $A_i^q = A_i$. La condition $A_{i+n} = A_i$ est introduite dans les équations eq_i, $i \geq n$.

Définition I.9 *Le système*

$$\begin{array}{rcl} A_{w+1} + A_w\sigma_1 + \cdots + A_1\sigma_w & = & 0 \\ A_{w+2} + A_{w+1}\sigma_1 + \cdots + A_1\sigma_w & = & 0 \\ & \vdots & \\ A_{n+w} + A_{n+w-1}\sigma_1 + \cdots + A_n\sigma_w & = & 0 \\ A_{qi \bmod n} & = & A_i^q \\ A_{i+n} & = & A_i \\ A_{i_1} = \cdots = A_{i_l} & = & 0 \end{array} \tag{I.14}$$

est le système algébrique *associé au problème* $\mathcal{PG}_{i_1,\ldots,i_l}(w)$ *dans le contexte* $(\mathbf{k}, \mathbf{F})$. *Nous le notons* $\mathcal{S}_{i_1,\ldots,i_l}(w)$. *Soit* $\overline{\mathbf{k}}$ *la clôture algébrique de* $\mathbf{k}$, *une* solution algébrique *de* $\mathcal{S}_{i_1,\ldots,i_l}(w)$ *est* $(\sigma_1, \ldots, \sigma_w, A_0, \ldots, A_{n-1}) \in \overline{\mathbf{k}}^{n+w}$ *satisfaisant I.14.*

Exemple 2 $\mathbf{k} = \mathbb{F}_2$, $\mathbf{F} = \mathbb{F}_8$. On considère $\mathcal{PG}_1(3)$. Les classes cyclotomiques de 2 modulo 7 sont

$$\{\{0\}, \{1,2,4\}, \{3,6,5\}\}.$$

Le système se transforme ainsi

$$\left\{\begin{array}{c} A_4 + A_3\sigma_1 + A_2\sigma_2 + A_1\sigma_3 = 0 \\ A_5 + A_4\sigma_1 + A_3\sigma_2 + A_2\sigma_3 = 0 \\ A_6 + A_5\sigma_1 + A_4\sigma_2 + A_3\sigma_3 = 0 \\ A_7 + A_6\sigma_1 + A_5\sigma_2 + A_4\sigma_3 = 0 \\ A_8 + A_7\sigma_1 + A_6\sigma_2 + A_5\sigma_3 = 0 \\ A_9 + A_8\sigma_1 + A_7\sigma_2 + A_6\sigma_3 = 0 \\ A_{10} + A_9\sigma_1 + A_8\sigma_2 + A_7\sigma_3 = 0 \\ A_0^2 = A_0 \end{array}\right. \rightarrow \mathcal{S}_1(3) \; : \; \left\{\begin{array}{rcl} A_3\sigma_1 & = & 0 \\ A_3^4 + A_3\sigma_2 & = & 0 \\ A_3^2 + A_3^4\sigma_1 + A_3\sigma_3 & = & 0 \\ A_0 + A_3^2\sigma_1 + A_3^4\sigma_2 & = & 0 \\ A_0\sigma_1 + A_3^2\sigma_2 + A_3^4\sigma_3 & = & 0 \\ A_0\sigma_2 + A_3^2\sigma_3 & = & 0 \\ A_3 + A_0\sigma_3 & = & 0 \\ A_0^2 & = & A_0 \end{array}\right.$$

après introduction des conditions $A_1 = 0$, $A_{iq \bmod n} = A_i^q$, $A_{i+n} = A_i$.

Théorème I.11 *Si le problème* $\mathcal{PG}_{i_1,\ldots,i_l}(w)$ *admet des solutions autres que la solution nulle, alors le système* $\mathcal{S}_{i_1,\ldots,i_l}(w)$ *admet une solution algébrique différente de* $(0, \ldots, 0)$.

Exemple 3 Dans le contexte $\mathbf{k} = \mathbb{F}_3$, $\mathbf{F} = \mathbb{F}_9$, soit $\mathcal{PG}_{1,2,5}(4)$. Les classes cyclotomiques de 3 modulo 8 sont

$$\{\{0\}, \{1,3\}, \{2,6\}, \{4\}, \{5,7\}\}.$$

Le système $\mathcal{S}_{1,2,5}(4)$ s'écrit alors

$$\begin{array}{rcl} A_4\ \sigma_1 & = & 0 \\ A_4\ \sigma_2 & = & 0 \\ A_4\ \sigma_3 & = & 0 \\ A_4\ \sigma_4 + A_0 & = & 0 \\ A_0\ \sigma_1 & = & 0 \\ A_0\ \sigma_2 & = & 0 \\ A_0\ \sigma_3 & = & 0 \\ A_0\ \sigma_4 + A_4 & = & 0 \\ {A_0}^3 & = & A_0 \\ {A_4}^3 & = & A_4 \end{array}$$

Ce système admet les solutions algébriques suivantes :

1. $A_4 = A_0 = 0$, les σ_i sont indéterminés.
2. $A_4 = A_0 = 1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0$, $\sigma_4 = -1$. Les X_i sont racines de $\sigma(Z) = -X^4 + 1$.
3. $A_4 = 1$, $A_0 = -1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0$, $\sigma_4 = 1$. Les X_i sont racines de $\sigma(Z) = X^4 + 1$.
4. $A_4 = A_0 = -1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0$, $\sigma_4 = -1$. Les X_i sont racines de $\sigma(Z) = -X^4 + 1$.
5. $A_4 = -1$, $A_0 = 1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0$, $\sigma_4 = 1$. Les X_i sont racines de $\sigma(Z) = X^4 + 1$.

La première solution correspond à la solution $X_1 = X_2 = X_3 = X_4 = 0$. Les solutions correspondant aux cas 2 et 4 sont

$$(X_1, X_2, X_3, X_4) = (\alpha^0, \alpha^2, \alpha^4, \alpha^6).$$

Les solutions correspondant aux cas 3 et 5 sont

$$(X_1, X_2, X_3, X_4) = (\alpha^1, \alpha^3, \alpha^5, \alpha^7).$$

La résolution du système de Vandermonde I.12 donne les quatre solutions :

$$\begin{aligned}(1,1,1,1),(\alpha^0,\alpha^2,\alpha^4,\alpha^6)\\(-1,-1,-1,-1),(\alpha^0,\alpha^2,\alpha^4,\alpha^6)\\(1,1,1,1),(\alpha^1,\alpha^3,\alpha^5,\alpha^7)\\(-1,-1,-1,-1),(\alpha^1,\alpha^3,\alpha^5,\alpha^7)\end{aligned}$$

Exemple 4 Dans le contexte $\mathbf{k} = \mathbb{F}_3$, $\mathbf{F} = \mathbb{F}_{27}$ ($n = 26$), soit le problème $\mathcal{PG}_{1,2,4,7,14}(5)$. les classes cyclotomiques de 3 modulo 26 sont

$$\{\{0\},\{1,3,9\},\{2,6,18\},\{4,12,10\},\{5,15,19\},\{7,21,11\},\{8,24,20\},\{13\},\{14,16,22\},\{17,25,23\}\}$$

Le système $\mathcal{S}_{1,2,4,7,14}(5)$ s'écrit

$$\begin{aligned}A_5\ \sigma_1 &= 0\\ A_5\ \sigma_2 &= 0\\ A_5\ \sigma_3 + A_8 &= 0\\ A_5\ \sigma_4 + A_8\ \sigma_1 &= 0\\ A_5\ \sigma_5 + A_8\ \sigma_2 &= 0\\ A_8\ \sigma_3 &= 0\\ &\vdots\end{aligned}$$

L'ensemble des fonctions puissances nulles contient $\{1,2,3,4,6,7,9,10,11,12\}$. Si nous écartons la solution nulle, nous avons $A_5 \neq 0$. En effet si $A_5 = 0$, l'ensemble des fonctions puissances généralisées devant être annulées contient $\{1,2,3,4,5,6,7\}$, ce qui est impossible pour $w = 5$ à cause du théorème I.10. Les équations $A_5\ \sigma_3 + A_8 = 0$ et $A_8\ \sigma_3 = 0$ impliquent alors la nullité simultanée de σ_3 et de A_8. Ce qui est impossible car A_8 ne peut être nul, pour la même raison que A_5.

Le système $\mathcal{PG}_{1,2,4,7,14}(5)$, $\mathbf{k} = \mathbb{F}_3$, $\mathbf{F} = \mathbb{F}_{27}$ n'admet donc pas de solutions.

2.3 Les solutions du système algébrique

2.3.1 Une condition nécessaire et suffisante

Etant donné le problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$, nous avons construit, à partir des identités de Newton, un système d'équations algébriques, $\mathcal{S}_{i_1,\ldots,i_l}(w)$. Nous avons vu que l'existence de solutions à ce système est une condition *nécessaire* à l'existence de solutions au problème des fonctions puissances (théorème I.11).

Nous étudions maintenant les solutions *algébriques* du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$, et le lien entre ces solutions et les solutions du problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$.

Théorème I.12 *Le problème des fonctions puissances $\mathcal{PG}_{i_1,\ldots,i_l}(w)$ admet une solution non multiple si et seulement si il existe une solution algébrique $(\sigma_1,\ldots,\sigma_w,A_0,\ldots,A_{n-1})$ du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$, telle que :*

$$\sigma(Z) \mid Z^n - 1 \text{ où } \sigma(Z) = 1 + \sum_{i=1}^{w} \sigma_i Z^i. \tag{I.15}$$

Preuve : Soit $(\sigma_1,\ldots,\sigma_w,A_0,\ldots,A_{n-1})$ une solution du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$, vérifiant la condition de divisibilité I.15, et soient $(X_1,\ldots,X_w)$ les inverses des racines de $\sigma(Z)$ défini dans I.15. La condition de divisibilité implique alors que $X_1,\ldots,X_w$ sont tous distincts, différents de 0, et appartiennent à $\mathbf{F}_{q^m}$, où $n = q^m - 1$. Il reste à déterminer les coefficients $a_1,\ldots,a_w$ tel que $A_i = \sum_{k=1}^{w} a_k X_k^i$. Ceci amène à résoudre le système

$$\begin{aligned} a_1X_1 + a_2X_2 + \cdots + a_wX_w &= A_1 \\ a_1{X_1}^2 + a_2{X_2}^2 + \cdots + a_w{X_w}^2 &= A_2 \\ &\vdots \\ a_1{X_1}^w + a_2{X_2}^w + \cdots + a_w{X_w}^w &= A_w \end{aligned}$$

qui admet une et unique solution, car il s'agit d'une système de Vandermonde, qui est inversible puisque les X_i sont tous distincts.

La solution correspondant à $(\sigma_1,\ldots,\sigma_w,A_0,\ldots,A_{n-1})$ est alors

$$(a_1,\ldots,a_w,X_1,\ldots,X_w).$$

□

2.3.2 Signification des autres solutions

Définition I.10 *Un n-uplet $(A_0,\ldots,A_{n-1}) \in \overline{\mathbf{k}}^n$ est dit* solution du système algébrique $\mathcal{S}_{i_1,\ldots,i_l}(w)$ *s'il existe $(\sigma_1,\ldots,\sigma_w) \in \overline{\mathbf{k}}^w$ tels que $(\sigma_1,\ldots,\sigma_w,A_0,\ldots,A_{n-1})$ est une solution de $\mathcal{S}_{i_1,\ldots,i_l}(w)$.*

Théorème I.13 *Soit posé le problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$ dans le contexte $(\mathbf{k},\mathbf{F})$, et soit $(A_0 = \gamma_0,\ldots,A_{n-1} = \gamma_{n-1})$ une solution de de $\mathcal{S}_{i_1,\ldots,i_l}(w)$. Alors $(\gamma_0,\ldots,\gamma_{n-1}) \in \mathbf{F}^n$, et la transformation de Fourier inverse de $(\gamma_0,\ldots,\gamma_{n-1})$ sont une solution de $\mathcal{PG}_{i_1,\ldots,i_l}(w')$, pour $w' < w$, dans le même contexte.*

Il y a bijection entre l'ensemble de toutes les solutions des problèmes $\mathcal{PG}_{i_1,\ldots,i_l}(w')$, $w' < w$ et l'ensemble des n-uplets $(A_0,\ldots,A_{n-1}) \in \overline{\mathbf{k}}^n$ solutions du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$.

Preuve : Soit $a = (a_0, \ldots, a_{n-1})$ la transformation de Fourier inverse de $(\gamma_0, \ldots, \gamma_{n-1})$. On montre d'abord que a est à coefficients dans $\mathbf{k}$. Pour celà, il suffit de montrer $a_i^q = a_i$, $i = 0, \ldots, n-1$. Rappellons que les γ_i étant solutions du système des équations de Newton, ils vérifient $\gamma_k^q = \gamma_{kq \bmod n}$. On a :

$$\begin{aligned} a_i^q &= \left(\frac{1}{n}\sum_{k=0}^{n-1} \gamma_k (\alpha^i)^{n-k}\right)^q \\ &= \frac{1}{n}\sum_{k=0}^{n-1} \gamma_k^q \alpha^{qi(n-k)} \\ &= \frac{1}{n}\sum_{k=0}^{n-1} \gamma_{qk \bmod n} (\alpha^i)^{n-qk} \\ &= \frac{1}{n}\sum_{k'=0}^{n-1} \gamma_{k'} (\alpha^i)^{n-k'} = a_i \end{aligned}$$

puisque $\mathrm{pgcd}(q, n) = 1$. Réciproquement, a fournit une solution au problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$, car la transformation de Fourier étant bijective, ses coefficients de Fourier sont $\gamma_0, \ldots, \gamma_{n-1}$, et donc $\gamma_{i_1} = \cdots = \gamma_{i_l} = 0$. En particulier $(\gamma_0, \ldots, \gamma_{n-1}) \in \mathbf{F}^n$.

Il reste à déterminer le poids de la solution, c'est-à-dire le poids de a. En effet, a peut conduire à une solution algébrique de $\mathcal{S}_{i_1,\ldots,i_l}(w)$ et sans être de poids w. Soit w' le poids de a. Nous allons montrer que $w' \leq w$. D'après le théorème I.9, w' est égal au rang de la matrice :

$$C = \begin{pmatrix} \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 & \gamma_0 \\ \gamma_{n-2} & \gamma_{n-3} & \cdots & \gamma_0 & \gamma_{n-1} \\ \vdots & & & & \\ \gamma_0 & \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 \end{pmatrix}.$$

Or, il existe $\sigma_1, \ldots, \sigma_w$ tel que $(\sigma_1, \ldots, \sigma_w, \gamma_0, \ldots, \gamma_{n-1})$ est solution du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$, ce qui s'écrit matriciellement

$$\begin{pmatrix} \gamma_w & \gamma_{w-1} & \cdots & \gamma_0 \\ \gamma_{w+1} & \gamma_w & \cdots & \gamma_1 \\ \vdots & & & \\ \gamma_{n+w} & \gamma_{n+w-1} & \cdots & \gamma_n \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \vdots \\ \sigma_w \end{pmatrix} = 0.$$

Cette même combinaison permet d'annuler C de la manière suivante:

$$\begin{pmatrix} \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 & \gamma_0 \\ \gamma_{n-2} & \gamma_{n-3} & \cdots & \gamma_0 & \gamma_{n-1} \\ \vdots & & & & \\ \gamma_0 & \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \sigma_1 \\ \vdots \\ \sigma_w \end{pmatrix} = 0,$$

et C étant une matrice circulante, on a de même

$$\begin{pmatrix} \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 & \gamma_0 \\ \gamma_{n-2} & \gamma_{n-3} & \cdots & \gamma_0 & \gamma_{n-1} \\ \vdots & & & & \\ \gamma_0 & \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \sigma_1 \\ \vdots \\ \sigma_w \\ 0 \end{pmatrix} = 0.$$

Et ainsi de suite. Les $n-w$ premières colonnes de C sont ainsi combinaisons linéaires de w dernières, on en déduit que le rang de C est inférieur ou égal à w.

□

Corollaire I.1 *Le nombre total de solutions des $\mathcal{PG}_{i_1,\ldots,i_l}(w')$, $w' < w$, est égal au nombre de n-uplets $(A_0, \ldots, A_{n-1})$ solutions du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$.*

Corollaire I.2 *Soit posé le problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$, et supposons qu'il n'existe pas de solutions à $\mathcal{PG}_{i_1,\ldots,i_l}(w')$, $w' < w$. Alors le nombre de solutions de $\mathcal{PG}_{i_1,\ldots,i_l}(w)$ est égal au nombre de n-uplets $(A_0, \ldots, A_{n-1})$ solutions du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$.*

Théorème I.14 *Soit $(\gamma_0, \ldots, \gamma_{n-1})$ une solution de $S_{i_1,\ldots,i_l}(w)$, et soit $a = (a_0, \ldots, a_{n-1})$, la tranformée de Fourier inverse de $(\gamma_0, \ldots, \gamma_{n-1})$. Soit w_0 le poids a et $\sigma_a(Z) = 1 + \sum_{i=1}^{w_0} \Sigma_i Z^i$ le polynôme localisateur de a. Alors l'ensemble des solutions $(\sigma_1, \ldots, \sigma_w)$ de*

$$\begin{pmatrix} \gamma_{w+1} & \gamma_w & \cdots & \gamma_1 \\ \vdots & & & \\ \gamma_{w+n} & \gamma_{n+w-1} & \cdots & \gamma_n \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \vdots \\ \sigma_w \end{pmatrix} = 0 \tag{I.16}$$

est

$$\begin{aligned} \mathcal{F} \quad = \{(\sigma_1, \ldots, \sigma_w) \in \mathbf{F}^w \mid \quad & \exists (\lambda_1, \ldots, \lambda_{w-w_0}) \in \mathbf{F}^{w-w_0}, \\ & 1 + \sum_{i=1}^{w} \sigma_i Z^i = \sigma_a(Z)(1 + \lambda_1 Z + \cdots + \lambda_{w-w_0} Z^{w-w_0})\}. \end{aligned}$$

C'est-à-dire que les polynômes $1 + \sum_{i=1}^{w} \sigma_i Z^i$ formés à partir des solutions $(\sigma_1, \ldots, \sigma_w)$ sont les multiples du polynôme localisateur de a.

Preuve : Soit $\mathcal{F} \subset \mathbf{F}^w$ l'ensemble des solutions de I.16. Le rang de la matrice

$$C_{n,w} = \begin{pmatrix} \gamma_w & \gamma_{w-1} & \cdots & \gamma_0 \\ \gamma_{w+1} & \gamma_w & \cdots & \gamma_1 \\ \vdots & & & \\ \gamma_{n+w} & \gamma_{n+w-1} & \cdots & \gamma_n \end{pmatrix}$$

est exactement w_0. De plus le système I.16 est compatible car il admet la solution

$$(1, \Sigma_1, \ldots, \Sigma_{w_0}, 0, \ldots, 0)$$

où $\sigma_a(Z) = 1 + \sum_{i=1}^{w_0} \Sigma_i Z^i$. L'ensemble $\mathcal{F}$ est donc un espace affine de dimension $w - w_0$.

Soit $\mathcal{F}'$ l'espace

$$\begin{aligned}\mathcal{F}' \quad = \{(\sigma_1, \ldots, \sigma_w) \in \mathbf{F}^w \mid \quad & \exists (\lambda_1, \ldots, \lambda_{w-w_0}) \in \mathbf{F}^{w-w_0}, \\ & 1 + \sum_{i=1}^{w} \sigma_i Z^i = \sigma_a(Z)(1 + \lambda_1 Z + \cdots + \lambda_{w-w_0} Z^{w-w_0})\}.\end{aligned}$$

C'est un espace affine qui contient $(\Sigma_0, \ldots, \Sigma_{w_0}, 0, \ldots, 0)$ et $\dim \mathcal{F}' = w - w_0 = \dim \mathcal{F}$.

Soit $(\sigma_1, \ldots, \sigma_w) \in \mathcal{F}'$: il existe $(\lambda_1, \ldots, \lambda_{w-w_0}) \in \mathbf{F}^{w-w_0}$ tel que

$$1 + \sum_{i=1}^{w} \sigma_i Z_i = \sigma_a(X)(1 + \lambda_1 Z + \cdots + \lambda_{w-w_0} Z^{w-w_0}).$$

Alors

$$\begin{aligned}\sigma_j &= \lambda_j + \sum_{i-1}^{j} \Sigma_i \lambda_{j-i} + \Sigma_j, \qquad j \leq w_0, \\ \sigma_j &= \lambda_j + \sum_{i-1}^{w_0} \Sigma_i \lambda_{j-i}, \qquad j > w_0.\end{aligned}$$

En convenant $\lambda_j = 0$ si $j > w - w_0$. Montrons que $\sigma_1, \ldots, \sigma_w$ sont solutions du système I.16. Il faut vérifier que

$$\gamma_{w+1} + \gamma_w \sigma_1 + \cdots + \gamma_1 \sigma_w = \gamma_{w+1} + \gamma_w(\lambda_1 + \Sigma_1) + \cdots + \gamma_1(\lambda_w + \sum_{i-1}^{w_0} \Sigma_i \lambda_{w-i})$$

est nul. Le coefficient de λ_{w-w_0} est

$$\gamma_{w_0+1} + \gamma_{w_0} \Sigma_1 + \cdots + \gamma_1 \Sigma_{w_0} = 0,$$

car c'est la relation de Newton généralisée id_{w_0+1}.

De même le coefficient de λ_{w-j} est

$$\gamma_{w_0+j+1} + \gamma_{w_0+j} \Sigma_1 + \cdots + \gamma_{j+1} \Sigma_{w_0} = id_{w_0+j+1} = 0.$$

Le terme constant (par rapport aux λ_i) est

$$\gamma_w + \gamma_w \Sigma_1 + \cdots + \gamma_{w-w_0} \Sigma_{w_0} = id_w = 0.$$

On montre ainsi que le système I.16 est satisfait par $\sigma_1, \ldots, \sigma_w$. Les deux espaces affines $\mathcal{F}$ et $\mathcal{F}'$ sont de même dimension, et $\mathcal{F}' \subset \mathcal{F}$. Donc $\mathcal{F} = \mathcal{F}'$.

□

Exemple 5 Nous reprenons l'exemple page 28. $\mathbf{k} = \mathbb{F}_3$, $\mathbf{F} = \mathbb{F}_9$, soit $\mathcal{PG}_{1,2,5}(6)$. Les solutions du système $\mathcal{S}_{1,2,5}(6)$ vérifient

$$\sigma_6 + \sigma_2 A_4 A_0 = 0,\ \sigma_5 + \sigma_1 A_4 A_0 = 0,\ \sigma_4 + A_4 A_0 = 0,\ \sigma_3 = 0,\ {A_4}^2 + 2 = 0,\ {A_0}^2 + 2 = 0.$$

Les couples (A_0, A_4) solutions sont les mêmes que les solutions de $\mathcal{S}_{1,2,5}(4)$. Il n'y a donc pas de solution de poids 6 au système $\mathcal{S}_{1,2,5}(6)$. On voit de plus que les solutions $(\sigma_1, \ldots, \sigma_6)$ sont telles que

$$1 + \sum_{i=1}^{6} \sigma_i Z^i = (1 - A_0 A_4 Z^4)(1 + \sigma_1 Z + \sigma_2 Z^2).$$

Corollaire I.3 *Soit $(\gamma_0, \ldots, \gamma_{n-1})$ une solution de $S_{i_1,\ldots,i_l}(w)$, et soit $a = (a_0, \ldots, a_{n-1})$, la tranformée de Fourier inverse de $(\gamma_0, \ldots, \gamma_{n-1})$. Soit w_0 le poids de a et $\sigma_a(Z)$ le polynôme localisateur de a. Si $w = w_0$ alors les solutions de I.16 sont les coefficients du polynôme localisateur de a.*

Proposition I.2 *S'il n'existe pas de solutions à $\mathcal{PG}_{i_1,\ldots,i_l}(w')$, $w' < w$, alors le système $S_{i_1,\ldots,i_l}(w)$ admet un nombre fini de solutions.*

Preuve : Il n'existe qu'un nombre fini de n-uplets $(A_0, \ldots, A_{n-1})$ solutions de $S_{i_1,\ldots,i_l}(w)$, d'après le théorème I.13. Pour chaque $(A_0, \ldots, A_{n-1})$ solution, soit $a = (a_0, \ldots, a_{n-1})$ la transformée de Fourier inverse de $(A_0, \ldots, A_{n-1})$. Alors le poids de a est w par le corollaire I.2. Dans ce cas là, le corollaire I.3 indique qu'il y a un unique w-uplet $(\sigma_1, \ldots, \sigma_w)$ tel que $(\sigma_1, \ldots, \sigma_w, A_0, \ldots, A_w)$ soit solution de $S_{i_1,\ldots,i_l}(w)$.

□

2.4 Problèmes particuliers

2.4.1 Le problème des fonctions puissances de poids déterminé

Dans la plupart des cas traités en pratique, nous considérerons le cas $\mathbf{k} = \mathbb{F}_2$. Dans ce cas-là le seul coefficient possible aux fonctions puissances généralisées est 1, et les fonctions puissances généralisées sont les fonctions puissances.

Définition I.11 *Nous appelons le* problème des fonctions puissances de poids w *le problème posé par $A_{i_1} = A_{i_2} = \cdots = A_{i_l} = 0$ où les A_i sont les fonctions puissances de w inconnues $(X_1, \ldots X_w) \in \mathbf{F}^w$. De plus :*

Nous notons ce problème $\mathcal{P}_{i_1,\ldots i_l}(w)$. Une solution du problème est $(X_1, \ldots X_w) \in \mathbf{F}^w$ tels que les fonctions puissances des X_i vérifient $A_{i_1} = \ldots A_{i_l} = 0$.

Pour étudier ce problème nous écrirons le système :

$$\begin{pmatrix} T \\ C_{n,w} \end{pmatrix} \begin{pmatrix} 1 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{I.17}$$

où T est la matrice introduite en 1.2. Dans le cas de la caractéristique 2, nous n'écrirons que les équations d'indice impair, car les équations d'incide pair sont redondantes, en vertu du théorème I.4.

Définition I.12 *Le système I.17 avec les conditions* $A_{qi \bmod n} = A_i^q$, $A_{i+n} = A_i$, $A_{i_1} = \cdots = A_{i_l} = 0$ *est le* système algébrique *associé au problème* $\mathcal{P}_{i_1,\ldots,i_l}(w)$ *dans le contexte* $(\mathbf{k}, \mathbf{F})$. *Nous le notons* $\mathcal{S}_{i_1,\ldots,i_l}(w)$.

Ce système est beaucoup plus riche que la forme circulante seule, car il est quasiment triangulaire en les σ_i. Les seuls σ_i pour lesquels le système n'est pas triangulaire sont tels que $p \mid i$.

Exemple 6 $\mathbf{k} = \mathrm{F}_2$, $\mathbf{F} = \mathrm{F}_{32}$, $n = 31$, on considére $\mathcal{P}_{1,5,7}(5)$. Le système $\mathcal{S}_{1,5,7}(5)$ s'écrit :

$$\begin{aligned} \sigma_1 &= 0 \\ \sigma_3 + A_3 &= 0 \\ \sigma_5 + A_3\ \sigma_2 &= 0 \\ A_3\ \sigma_4 + {A_3}^2\ \sigma_1 &= 0 \\ {A_3}^2\ \sigma_3 &= 0 \\ &\vdots \end{aligned}$$

C'est-à-dire que A_3 et σ_3 sont tous deux nuls. Toute solution de ce problème est donc solution de $\mathcal{PG}_{1,3,5,7}(5)$, ce qui est impossible d'aprés le théorème I.10. Le problème $\mathcal{PG}_{1,5,7}(5)$ n'admet pas de solutions.

2.4.2 Les problèmes idempotents

Nous aurons besoin dans la suite des sous-problèmes particuliers suivant :

Définition I.13 *A chacun des problèmes précédents nous associons le problème suivant :* $A_{i_1} = A_{i_2} = \cdots = A_{i_l} = 0$ *et les autres valeurs sont égales à* 0 *ou* 1. *Nous appellons ce problème le* problème idempotent *associé au problème indéterminé (c'est à dire sans imposer de valeurs aux autres* A_i*). Nous notons successivement ces problèmes* $\mathcal{PGI}_{i_1,\ldots i_l}$, $\mathcal{PGI}_{i_1,\ldots i_l}(w)$, $\mathcal{PI}_{i_1,\ldots i_l}(w)$.

Nous rappellons la propriété de morphisme de la tranformation de Fourier :

Propriété I.2 *Soit* $\odot$ *le produit composante à composante dans* $\mathbf{F}^n$, *c'est-à-dire*

$$(A_0, \ldots, A_{n-1}) \odot (B_0, \ldots, B_{n-1}) = (A_0B_0, \ldots, A_{n-1}B_{n-1})$$

et soit T *la transformation qui à* $a \in \mathbf{k}^n$ *associe son polynôme de Mattson-Solomon.*

Alors

$$T(a(X)b(X) \bmod X^n - 1) = T(a) \odot T(b)$$

Preuve : Si $c(X) = a(X)b(X) \bmod X^n - 1$ et $C = C_1Z + \cdots + C_nZ^n$ est le polynôme de Mattson-Solomon de c alors $a(\alpha^i)b(\alpha^i) = c(\alpha^i)$ et donc $C_i = A_iB_i$.

□

La terminologie de problème idempotent est justifiée par le théorème suivant :

Théorème I.15 *Soit* $a \in \mathbf{k}^n$ *tel que les coefficients de son polynôme de Mattson-Solomon vérifient* $A_i = 0$ *ou* $A_i = 1$. *Alors le polynôme* $a(X) = a_0 + a_1x + \cdots + a_{n-1}X^{n-1}$ *vérifie*

$$a(X)^2 = a(X) \bmod X^n - 1.$$

Preuve : Si tous les coefficients du Mattson-Solomon $A(Z)$ de a sont 0 ou 1, alors $A(Z)$ vérifie $A(Z) \odot A(Z) = A(Z)$. La tranformation de Fourier étant un morphisme injectif de $\mathbf{k}[X]/(X^n-1)$ dans $(\mathbf{F}[Z], \odot)$, on a $a(X)^2 = a(X) \bmod X^n - 1$.

□

Définition I.14 *Le système*

$$\begin{array}{rcl}
A_{w+1} + A_w\sigma_1 + \cdots A_2\sigma_{w-1} + A_1\sigma_w & = & 0 \\
A_{w+2} + A_{w+1}\sigma_1 + \cdots A_3\sigma_{w-1} + A_1\sigma_w & = & 0 \\
\vdots & & \\
A_{n+w} + A_{n+w-1}\sigma_1 + \cdots A_{n+1}\sigma_{w-1} + A_n\sigma_w & = & 0 \\
A_{qi \bmod n} & = & A_i^q \\
A_{i+n} & = & A_i \\
A_i^2 & = & A_i \\
A_{i_1} = \cdots = A_{i_l} & = & 0
\end{array} \tag{I.18}$$

est le système algébrique *associé au problème* $\mathcal{PGI}_{i_1,\ldots,i_l}(w)$ *dans le contexte* $(\mathbf{k}, \mathbf{F})$. *Nous le notons* $\mathcal{SI}_{i_1,\ldots,i_l}(w)$.

Les conditions $A_{iq \bmod n} = A_i^q$ et $A_i^2 = A_i$ seront introduites de la manière suivante : pour chaque classe cyclotomique un plus petit représentant i_0 est choisi, et les A_i tels que i est dans la même classe cyclotomique que i_0 égaux à A_i. Si la classe cyclotomique de i se réduit à un seul élément, on introduit la condition $A_i^2 = A_i$.

Exemple 7 Nous reprenons l'exemple de la page 28 : $\mathbf{k} = \mathbb{F}_2$, $\mathbf{F} = \mathbb{F}_8$. On considère $\mathcal{PGI}_1(3)$. Le système se transforme ainsi

$$\left\{\begin{array}{c}
A_4 + A_3\sigma_1 + A_2\sigma_2 + A_1\sigma_3 = 0 \\
A_5 + A_4\sigma_1 + A_3\sigma_2 + A_2\sigma_3 = 0 \\
A_6 + A_5\sigma_1 + A_4\sigma_2 + A_3\sigma_3 = 0 \\
A_7 + A_6\sigma_1 + A_5\sigma_2 + A_4\sigma_3 = 0 \\
A_8 + A_7\sigma_1 + A_6\sigma_2 + A_5\sigma_3 = 0 \\
A_9 + A_8\sigma_1 + A_7\sigma_2 + A_6\sigma_3 = 0 \\
A_{10} + A_9\sigma_1 + A_8\sigma_2 + A_7\sigma_3 = 0 \\
A_0^2 = A_0
\end{array}\right. \rightarrow \mathcal{SI}_1(3) \; : \; \left\{\begin{array}{rcl}
A_3\sigma_1 & = & 0 \\
A_3 + A_3\sigma_2 & = & 0 \\
A_3 + A_3\sigma_1 + A_3\sigma_3 & = & 0 \\
A_0 + A_3\sigma_1 + A_3\sigma_2 & = & 0 \\
A_0\sigma_1 + A_3\sigma_2 + A_3\sigma_3 & = & 0 \\
A_0\sigma_2 + A_3\sigma_3 & = & 0 \\
A_3 + A_0\sigma_3 & = & 0 \\
A_0^2 & = & A_0
\end{array}\right.$$

après introduction des conditions $A_1 = 0$, $A_{iq \bmod n} = A_i$ si $cl(i) \neq \{i\}$, $A_i^2 = A_i$ si $cl(i) = \{i\}$, $A_{i+n} = A_i$.

Proposition I.3 *Le nombre total de solutions des* $\mathcal{PGI}_{i_1,\ldots,i_l}(w')$, $w' < w$, *est égal au nombre de n-uplets* $(A_0, \ldots, A_{n-1})$ *solutions du système* $\mathcal{SI}_{i_1,\ldots,i_l}(w)$.

3 Aspect symbolique

3.1 Traitement heuristique des équations de Newton

3.1.1 Cas de la caractéristique nulle

Afin d'illustrer l'originalité du problème en caractéristique p, nous montrons grâce à un exemple comment ce problème se résout en caractéristique nulle.

Exemple 8 Soit à déterminer X_1, X_2, X_3 dans C vérifiant $A_1 = A_7 = A_{10} = 0$, où A_i sont les fonctions puissances de X_1, X_2, X_3. Le système des équations s'écrit

$$\begin{aligned}
eq_1 &: \sigma_1 = 0 \\
eq_2 &: 2\sigma_2 + A_2 = 0 \Rightarrow \sigma_2 = -\frac{1}{2}A_2 \\
eq_3 &: 2\sigma_3 + A_3 = 0 \Rightarrow \sigma_3 = -\frac{1}{3}A_3 \\
eq_4 &: A_4 - \frac{1}{2}A_2^2 = 0 \Rightarrow A_4 = \frac{1}{2}A_2^2 \\
eq_5 &: A_5 - \frac{5}{6}A_2A_3 = 0 \Rightarrow A_3 = \frac{5}{6}A_2A_3 \\
eq_6 &: A_6 - \frac{1}{3}A_3^2 - \frac{1}{4}A_2^3 = 0 \Rightarrow A_6 = \frac{1}{3}A_3^2 + \frac{1}{4}A_2^3 \\
eq_7 &: -\frac{7}{12}A_2^2A_3 = 0 \\
eq_8 &: A_8 - \frac{4}{9}A_2A_3^2 - \frac{1}{8}A_2^4 = 0 \Rightarrow A_8 = \frac{4}{9}A_2A_3^2 + \frac{1}{8}A_2^4 \\
eq_9 &: A_9 - \frac{1}{9}A_3^3 - \frac{1}{12}A_2^3A_3 = 0 \Rightarrow A_9 = \frac{1}{9}A_3^3 + \frac{1}{12}A_2^3A_3 \\
eq_{10} &: -\frac{2}{9}A_2^2A_3^2 - \frac{1}{16}A_2^5 = 0
\end{aligned}$$

l'équation eq_{10} avec l'équation eq_7 montrent $A_2 = 0$. Tous les autres A_i sont ensuite déterminés, sauf A_3, et le polynôme localisateur à la forme :

$$\sigma(Z) = -\frac{1}{3}A_3Z^3 + 1.$$

3.1.2 Cas des corps finis

Nous présentons essentiellement un traitement heuristique des équations de Newton.

On écrit d'abord le système des identités de Newton, engendrées par calcul dans un système de calcul formel.

La première étape est la suivante : on détermine d'abord les fonctions symétriques σ_i, en fonction des fonctions puissances A_i. Le problème principal qui se pose dans le cas des corps finis est que le système n'est pas facilement résoluble en les σ_i, dès que $p < w$.

Toutefois nous pouvons faire les remarques suivantes au sujet du système des équations de Newton.

Cyclicité des solutions

En effet si $X_1, \ldots, X_w$ est solution d'un problème de fonctions puissances (déterminées ou indéterminées), alors pour tout $\beta \in \mathbf{F}$, $\beta X_1, \ldots, \beta X_w$ est aussi solution.

Si nous notons A'_i la fonction puissance associée à $\beta X_1, \ldots, \beta X_w$ alors nous avons :

$$A'_i = \beta^i A_i.$$

Pour $k \in [1, n-1]$, $(\beta^i)^k$ décrit un sous groupe multiplicatif de $\mathbf{F}^*$. Si $A_i \neq 0$, et si le sous-groupe engendré par β^i est $\mathbf{F}^*$, il existe k tel que

$$(\beta^i)^k A_i = 1.$$

Par exemple, si $A_i \neq 0$, et si i et n sont premiers entre eux, il existe toujours k tel qu'en multipliant la solution par α^k, on ait $\alpha^k A_i = 1$, ce qui permet d'écrire le système avec $A_i = 1$. Cette opération ne peut s'effectuer que sur une seule indéterminée A_i.

Cyclicité des fonctions puissances

Chaque fonction puissance vérifie $A_i^{q^m} = A_i$ et $A_i^n = 1$, si $A_i \neq 0$. Nous nous servirons de cette relation pour traiter des équations du type

$$A_i^{q^{m_1}} + P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1}) = 0,$$

où A_i n'apparaît pas dans P. Alors A_i peut être exprimer de la manière suivante :

$$A_i = -P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1})^{q^{m-m_1}}.$$

De même pour une équation du type

$$A_i^k + P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1}) = 0, \qquad \gcd(k, n) = 1,$$

et si $A_i \neq 0$, nous pourrons exprimer A_i ainsi :

$$A_i = -P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1})^{k'}$$

où $kk' = 1 \bmod n$. Toutefois cette résolution est à éviter car, contrairement à la précédente, il y a une croissance du nombre de termes dans le calcul de $P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1})^{k'}$, qui n'apparaît pas dans le calcul de $P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1})^{q^{m-m_1}}$, où la linéarité de l'application $P \mapsto P^q$ préserve le nombre de termes.

Ces propriétés interviennent dans les équations du type

$$A_j^k A_i + P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1}) = 0,$$

qui permettent aussi d'exprimer A_i, si $A_j \neq 0$

$$A_i = -P(A_0, \ldots, \hat{A}_i, \ldots, A_{n-1}) A_j^{n-k}.$$

Exemple 9 Nous nous plaçons dans le contexte $\mathbf{k} = \mathrm{F}_2$, $\mathbf{F} = \mathrm{F}_{128}$, $n = 127$, et on se pose le problème $\mathcal{P}_{1,\ldots,28}(29)$. Nous allons prouver que le système $\mathcal{S}_{1,\ldots,28}(29)$ n'a pas de solutions pour $w = 29$. C'est un exemple simple, et rapide, qui illustre bien notre démarche.

Par le théorème I.10, nous savons qu'il n'existe pas de solutions pour $w < 29$.

Conformément à notre heuristique, nous essayons d'abord d'éliminer les σ_i. La première partie triangulaire des équations de Newton permet d'éliminer les σ_i d'indice impair. Nous avons $\sigma_i = 0$ pour i impair inférieur à 28. Ensuite nous pouvons constater que, pour $i \leq 29$ pair, l'équation eq_{i+29} a la forme suivante :

$$eq_{i+29} = A_{29+i} + A_{29+i-2}\sigma_2 + \cdots + A_{31}\sigma_{i-2} + A_{29}\sigma_i = 0$$

En effet $A_i = 0$ pour $i < 29$. De plus le terme général de l'équation eq_{i+29} est $A_k \sigma_{29+i-k}$, $k < 58$. Si k est pair, alors $A_k = A_{2k'} = A_{k'}^2$, avec $k' < 29$, et donc $A_k = 0$.

De plus nous pouvons supposer $A_{29} \neq 0$, à cause du théorème I.10. Puisque $\mathrm{pgcd}(29, 127) = 1$, nous supposons $A_{29} = 1$. Nous introduisons donc dans le système $A_{29} = 1$ et le système des équations eq_{i+29}, $1 \leq i \leq 29$, i pair, est triangulaire en les σ_i d'indice pair, et nous permet alors d'éliminer σ_i, i pair. Les fonctions symétriques élémentaires σ_i sont donc toutes déterminées. Les seules indéterminées intervenant alors dans $\mathcal{S}_{1,\ldots,28}(29)$ sont $A_{31}, A_{43}, A_{47}, A_{55}, A_{63}$.

Nous présentons un exemple de preuve que le système est contradictoire.Nous en sommes au stade où toutes les fonctions symétriques élémentaires σ_i ont été éliminées, et remplacées par une expression en fonction des A_i.

La première équation non encore traitée et non nulle est l'équation eq_{59}. Nous rappelons aussi qu'il n'est pas nécessaire de traiter les équations d'indice pair, car elles sont combinaisons des équations précédentes (théorème I.4).

eq 59

$$A_{55}{}^8 + A_{31}{}^2\ A_{55} + A_{31}{}^6\ A_{47} + A_{31}\ A_{43}{}^2 + A_{31}{}^8\ A_{43} + A_{31}{}^{15} + A_{31}{}^{10} + A_{31}{}^5 + 1$$

eq 61

$$A_{31}\ A_{55}{}^8 + A_{31}{}^3\ A_{55} + A_{47}{}^4 + A_{31}{}^7\ A_{47} + A_{31}{}^4\ A_{43}{}^{16} + A_{43}{}^8 + A_{31}{}^8\ A_{43}{}^4 + A_{31}{}^2\ A_{43}{}^2 + A_{31}{}^9\ A_{43} + A_{31}{}^{11} + A_{31}{}^6 + A_{31}$$

— *On fait la combinaison suivante des l'équation eq_{59} et eq_{61} afin d'éliminer A_{47}. L'élévation a la puissance 32 est pour ensuite réduire les exposants, en tenant compte que $A_i^{128} = A_i$.*

tempeq:=(eq(61)+a.31 * eq(59))**32

$$A_{47}{}^{128} + A_{31}{}^{128}\ A_{43}{}^{512} + A_{43}{}^{256} + A_{31}{}^{256}\ A_{43}{}^{128} + A_{31}{}^{512}$$

tempeq:=reduceExponents tempeq

$$A_{47} + A_{31}\ A_{43}{}^4 + A_{43}{}^2 + A_{31}{}^2\ A_{43} + A_{31}{}^4$$

— *Une fois les exposants réduits, on peut alors éliminer A_{47}.*

a.47:=tempeq+a.47

$$A_{31}\ A_{43}{}^4 + A_{43}{}^2 + A_{31}{}^2\ A_{43} + A_{31}{}^4$$

eq 63

$$A_{63} + A_{31}{}^2\ A_{55}{}^8 + A_{31}{}^4\ A_{55} + A_{31}\ A_{43}{}^8 + A_{31}{}^9\ A_{43}{}^4 + A_{31}{}^8\ A_{43}{}^2 + A_{43} + A_{31}{}^{17}$$

— *Cette équation permet d'exprimer simplement A_{63}.*

a.63:=eq(63)+a.63;

eq 69

$$A_{55}{}^8 + A_{31}{}^2\ A_{55} + A_{31}{}^7\ A_{43}{}^4 + \left(A_{31}{}^6 + A_{31}\right)\ A_{43}{}^2 + A_{31}{}^{15} + A_{31}{}^5$$

— *La combinaison suivantes des équations eq_{59} et eq_{69} fournit la contradiction.*

eq(69)+eq(59)

$$1$$

Le système $\mathcal{P}_{1,\ldots,28}(29)$ n'admet pas de solutions.

3.2 L'outil algorithmique des bases standards

Les méthodes heuristiques de la section précédentes peuvent échouer en général, et il nous faut disposer d'un outil plus algorithmique.

La notion de base standard est une notion intervenant dans l'étude algorithmique de systèmes d'équations algébriques. Le problème est le suivant : étant donnés une famille $(f_1, \ldots, f_s)$ de polynômes à n variables dans un corps $\mathbf{k}$, "trouver" les zéros communs des f_i dans la clôture algébrique de $\mathbf{k}$. "Trouver" consistera à produire un système d'équations équivalent à $(f_1, \ldots, f_s)$, que l'on sait mieux utiliser.

Principalement, nous nous intéressons au problème de l'existence de solutions, et au problème de la détermination du nombre de solutions. Les bases standards permettent exactement de répondre à ces questions, et il existe un algorithme pour les calculer. Nous les considérerons donc comme un outil algorithmique de manipulation de systèmes algébriques.

Cette section est consacrée à introduire les notions intervenant dans de tels outils. Nous serons brefs dans les définitions, en cherchant à donner au lecteur une intuition des notions, le théorème I.16 pouvant être vu comme une analogie avec le cas d'une seule variable. L'exposé est essentiellement celui de [LJ85].

3.2.1 Ordre sur les monômes

Dans le cas d'une variable, l'anneau $\mathbf{k}[X]$ est euclidien, et les idéaux sont engendrés par un unique polynôme. Ainsi déterminer si deux polynômes ont une racine commune consiste à vérifier que le pgcd de ces deux polynômes est différent de 1. Dans le cas de plusieurs variables, ce n'est plus le cas. La division de Hironaka d'un polynôme par une famille de polynôme est l'analogue de la division euclidienne. Pour la définir, nous avons besoin de pouvoir comparer les polynômes. Pour cela nous avons besoin d'un ordre sur les monômes. Soit $f \in \mathbf{k}[X_1, \ldots, X_n]$, nous notons $f = \sum f_\alpha X^\alpha$, où les α sont les exposants des monômes de f, $\alpha \in \mathrm{N}^n$. On considère N^n muni d'un ordre total $<$ compatible avec l'addition (c'est-à-dire que $(\mathrm{N}^n, +, <)$ est un monoïde ordonné), et tel que

$$\forall \alpha \in \mathrm{N}^n,\ \forall \beta \in \mathrm{N}^n,\ \beta \neq 0,\ \alpha < \alpha + \beta.$$

Les ordres les plus classiques sont les suivants :

Définition I.15 *Par définition* l'ordre lexicographique *sur* N^n *est l'ordre suivant :*

$$(\alpha_1, \alpha_2 \ldots, \alpha_n) < (\beta_1, \beta_2, \ldots, \beta_n) \Leftrightarrow \exists\ s \in [1, n],\ (\forall i < s,\ \alpha_i = \beta_i)\ \textit{et}\ (\alpha_s < \beta_s)$$

*L'*ordre diagonal (ou ordre du degré) *est l'ordre suivant :*

$$(\alpha_1, \alpha_2 \ldots, \alpha_n) < (\beta_1, \beta_2, \ldots, \beta_n) \Leftrightarrow \begin{cases} \sum_{i=1}^n \alpha_i < \sum_{i=1}^n \beta_1 \\ \textit{ou} \\ \begin{cases} \sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_1 \\ \textit{et} \\ \exists\ s \in [1, n],\ \forall i < s,\ \alpha_i = \beta_i\ \textit{et}\ \alpha_s < \beta_s. \end{cases} \end{cases}$$

Toutes les définitions qui suivent sont relatives à un ordre donné :

Définition I.16 *Soit* $f = \sum c_\alpha X^\alpha \in \mathbf{k}[X_1, \ldots, X_n]$, l'exposant privilégié *de* f *noté* $\exp f$ *est* α *maximal tel que* $c_\alpha \neq 0$. *La* forme initiale *de* f, *noté* inf *est* $c_\alpha X^\alpha$ *avec* $\alpha = \exp f$.

Proposition I.4 (Hironaka) *Soit* $f_1, \ldots f_s \in \mathbf{k}[X_1, \ldots X_n]$ *et soient* $\Delta, \Delta_1, \ldots, \Delta_s$ *définis comme suit :*

$$\begin{array}{rcll} \Delta_1 & = & \exp(f_1) + \mathrm{N}^n & \\ \Delta_i & = & \exp(f_i) + \mathrm{N}^n - \cup_{j<i} \Delta_j, & i = 2 \ldots s \\ \Delta & = & \mathrm{N}^n \setminus \cup_{i \in [1,s]} \Delta_s, & \end{array}$$

alors pour tout $f \in \mathbf{k}[X_1, \dots X_n]$, *il existe* $h, h_1, \dots h_s$ *uniques tels que :*

$$f = h_1 f_1 + \cdots + h_s f_s + h,$$

et tels que

$$\forall i \in [1, s],\ si\ h_i = \sum_\alpha c_\alpha X^\alpha, \qquad (c_\alpha \neq 0) \Rightarrow (\exp(f_i) + \alpha \in \Delta_i)\,,$$
$$si\ h = \sum_\alpha c_\alpha X^\alpha, \qquad (c_\alpha \neq 0) \Rightarrow (\alpha \in \Delta)\,.$$

Si $f \neq 0$, *alors* $\exp h \leq \exp f$ *ou* $h = 0$, *et* $\exp f_i + \exp h_i \leq \exp f$, *ou* $h_i = 0$. *Le polynôme* h *étant unique, on le note* $h = f\mathcal{R}\{f_1, \dots, f_s\}$.

Cette technique de réduction peut être vue comme l'analogue du calcul du reste d'un polynôme par rapport à un autre polynôme dans $\mathbf{k}[X]$. Toutefois ce calcul est dépendant de l'ordre des polynômes $f_1, \dots f_s$, et $h = f\mathcal{R}\{f_1, \dots, f_s\}$ peut être différent de $h' = f\mathcal{R}\{f_s, \dots, f_1\}$, $\{f_s, \dots, f_1\}$ pris dans un ordre différent.

Définition I.17 *Soit* I *un idéal de* $\mathbf{k}[X_1, \dots X_n]$, *on note* $E(I)$ *l'ensemble :*

$$E(I) = \{\exp(f), f \in I\}.$$

Un ensemble $S \subset E(I)$ *est une frontière de* $E(I)$ *si*

$$E(I) = \sum_{\alpha \in S} \alpha + \mathbf{N}^n.$$

Théorème-Définition I.1 *Soit* I *un idéal de* $\mathbf{k}[X_1, \dots X_n]$, *alors il existe un ensemble fini* S *qui soit frontière de* $E(I)$. *L'*escalier *d'un idéal* I *est une frontière de* $E(I)$ *de cardinal minimal.*

3.2.2 Bases standards et réduction

Définition I.18 *Soit* I *un idéal de* $\mathbf{k}[X_1, \dots, X_n]$, *d'ensemble d'exposants privilégiés* $E(I)$, *une* base standard *de* I *est un ensemble fini de polynômes* $(g_1, g_2 \dots g_s)$ *tel que :*

$$E(I) = \cup_{i=1}^s \exp g_i + \mathbf{N}^n.$$

Définition I.19 *Une base standard minimale est une base standard de cardinal minimal.*

La technique de division de Hironaka prend tout son sens lorsqu'un polynôme est réduit par rapport à une base standard. Nous énumérons sans démonstrations les propriétés suivantes d'une base standard de I :

Proposition I.5 *Soit* I *un idéal de* $\mathbf{k}[X_1, \dots, X_n]$, *et* $(f_1, \dots f_s)$, $(g_1, \dots g_s)$ *deux bases standards de* I *(pour le même ordre sur les monômes), alors :*

$$f\mathcal{R}\{f_1, \dots f_s\} = f\mathcal{R}\{g_1, \dots g_s\}.$$

Cette propriété nous permet de caractériser un idéal, c'est à dire d'avoir un critère d'appartenance à I et de déterminer un représentant de $f \bmod I$:

Théorème I.16 *Soit $f_1, \dots f_s$ une base standard de I, nous notons $f\mathcal{R}I$ pour $f\mathcal{R}\{f_1 \dots f_s\}$. Nous avons les propriétés :*

$$\begin{aligned} \left(f\mathcal{R}I = \sum_\alpha c_\alpha X^\alpha\right),\ c_\alpha \neq 0 &\Rightarrow \alpha \notin E(I) \\ f\mathcal{R}I \neq 0 &\Rightarrow exp(f\mathcal{R}I) \leq \exp(f) \\ f \in I &\Leftrightarrow f\mathcal{R}I = 0 \end{aligned}$$

En particulier une base standard de l'idéal I est un système de générateur de I.

Définition I.20 *Soit I un idéal de $\mathbf{k}[X_1, \dots, X_n]$. On dit que $(f_1, \dots, f_s)$ est une* base standard réduite *de I si*

1. *$(f_1, \dots, f_s)$ est une base standard de I.*
2. *$inf_i = X^{\exp f_i}$, $i = 1 \dots s$.*
3. *$f_i\mathcal{R}\{f_1, \dots, \hat{f}_i, \dots, f_s\} = f_i$ (où $(f_1, \dots, \hat{f}_i, \dots, f_s) = (f_1, \dots, f_{i-1}, f_{i+1}, \dots, f_s)$).*

Proposition I.6 *Pour un ordre donné sur N^n, les bases standards réduites d'un idéal I de $\mathbf{k}[X_1, \dots, X_n]$ sont égales, à permutation des polynômes de la base près.*

Ainsi, l'ordre sur N^n étant fixé, nous parlerons de *la* base standard d'un idéal I, en parlant d'une base standard réduite de I.

La propriété suivante nous permet de déterminer si l'idéal a des solutions :

Proposition I.7 *Le système algébrique $(g_1, \dots g_s)$ a des solutions dans $\overline{\mathbf{k}}$ si et seulement si une base standard de l'idéal $I = (g_1, \dots, g_s)$ ne contient pas 1.*

Propriété I.3 *Soit $I \subset \mathbf{k}[X_1, \dots, X_n]$ engendré par $(f_1, \dots, f_s)$. Si $\mathcal{C}ard(\mathrm{N}^n \setminus E(I)) < \infty$, alors le système $(f_1, \dots, f_s)$ admet un nombre fini s de solutions dans $\overline{\mathbf{k}}$ et*

$$s \leq \mathcal{C}ard(\mathrm{N}^n \setminus E(I)).$$

Le nombre de solutions, comptés avec multiplicité, de $(f_1, \dots, f_s)$ est $\mathcal{C}ard(\mathrm{N}^n \setminus E(I))$.

Calcul de bases standards et le logiciel Gb Buchberger a présenté en 1965 un algorithme de calcul de base standard. Le principe de l'algorithme de Buchberger est présenté dans [LJ85], et pour une description plus détaillée, un chapitre de [GCL92] est consacré aux raffinement de cet algorithme. Le principal problème est la grande complexité pratique du calcul d'une base standard, et une borne supérieure de complexité, dans le cas où le système $(f_1, \dots, f_s)$ admet un nombre fini de solutions est $d^{O(n^2)}$, où n est le nombre de variables, et d le degré total maximal des f_i, $i = 1, \dots, s$ [Laz93].

La plupart des systèmes de calcul formel (Maple, Axiom) proposent une implantation de l'algorithme de Buchberger, ou de variantes. Toutefois, étant donné la taille des problèmes que nous étudierons, ainsi que le grand nombre de variables, ces systèmes, trop généralistes, ne permettent pas de calculer la base standard. C'est pourquoi nous avons utilisé "Gb". Il s'agit d'un système de calcul de base standard, écrit par Jean-Charles Faugère au LITP [Fau93]. Ce système présente pour nous les avantages suivants :

1. Il existe une interface commode avec Axiom. Il est possible de mener des calculs avec Axiom, et à tout moment de lancer le calcul d'une base standard d'une famille de polynômes obtenus en Axiom. Le calcul est alors effectué par Gb, qui retourne le résultat sous forme d'objets Axiom. Pour les systèmes que nous considérons, nous ne pouvons pas les créer à la main, à cause de leur taille. Cette interface est donc d'une grande utilité.

2. Gb est le "recordman" des calculs de base standard, pour deux raisons. La première est théorique : l'état de l'art en algorithmique des bases standards y est implanté. La stratégie du "sucre" [GMN+91] semble être empiriquement très bonne. De plus, il est possible, à partir d'une base standard obtenue pour l'ordre du degré, de calculer une base standard pour l'ordre lexicographique, par des techniques d'algèbre linéaire [FGLM]. La deuxième raison est pratique : il est du à la qualité de l'implantation, et notamment à la gestion de la mémoire. Les facteurs de comparaison des calcul sont les suivants : Gb est 3 à 6 fois plus rapides que Macaulay, 30 à 100 fois rapide que Axiom.

3. Jean-Charles Faugère a écrit un module spécial de Gb pour calculer sur $\mathbf{k} = \mathrm{F}_2$, qui est le corps qui nous intéresse principalement.

Toutefois, pour de grands exemples, il apparaît que le calcul de base standard n'est pas très efficace, notamment dans certains cas qu'il est possible de traiter à la main, en utilisant les heuristiques indiquées dans le chapitre précédent.

Deux leviers permettraient d'obtenir des algorithmes plus rapides. L'algorithme de calcul de base standard est très sensible à l'ordre défini sur les monômes. L'expérience montre que les indéterminées σ_i sont difficiles à éliminer, et qu'un pré-traitement à la main pour éliminer ceux-ci (comme indiqué en 3.1.2) facilite la tâche de Gb. Les ordres d'*élimination* à la Macaulay [SSB89] pourraient être meilleurs que l'ordre lexicographique, ou l'ordre du degré.

Le deuxième levier est la stratégie dans le calcul de la base standard. Les actions indiquées dans les heuristiques du chapitre précédent ne sont guère compatibles avec les techniques de calcul de base standard, car elles créent une augmentation du degré, alors que le but est de réduire le terme de tête. Il reste à concevoir des stratégies dédiées aux problèmes $\mathcal{S}_{i_1,\ldots,i_l}(w)$, qui semblent de nature très particulière.

3.2.3 Une utilisation directe

Nous discutons ici du problème $\mathcal{P}_{i_1,i_2,\ldots,i_l}(w)$, dans le contexte $(\mathbf{k}, \mathbf{F})$. Il est possible de formuler en termes de système algébrique le problème $\mathcal{P}_{i_1,i_2,\ldots,i_l}(w)$, sans utiliser le système des équations de Newton. Notre problème est de savoir s'il existe des solutions dans $\mathbf{F}$. On étudiera donc l'idéal engendré par la famille suivante :

$$\begin{aligned} X_1^{i_1} + \cdots + X_w^{i_1} &= 0 \\ X_1^{i_2} + \cdots + X_w^{i_2} &= 0 \\ &\vdots \\ X_1^{i_l} + \cdots + X_w^{i_l} &= 0 \\ X_1^n = \cdots = X_w^n &= 1 \end{aligned}$$

Définition I.21 *Dans le contexte* $(\mathbf{k}, \mathbf{F})$*, le système algébrique* brut $\mathcal{B}_{i_1,\ldots,i_l}(w)$ *est le système*

$$\begin{aligned} X_1^{i_1} + \cdots + X_w^{i_1} &= 0 \\ X_1^{i_2} + \cdots + X_w^{i_2} &= 0 \\ &\vdots \\ X_1^{i_l} + \cdots + X_w^{i_l} &= 0 \\ X_1^n = \cdots = X_w^n &= 1 \end{aligned}$$

A la différence du système algébrique $\mathcal{S}_{i_1,\ldots,i_l}(w)$, ce système d'équations ne permet pas de discriminer les solutions telles que les indéterminées X_i sont deux à deux distinctes. Toutefois, dans le cas de la caractéristique 2, il y a équivalence entre solutions algébriques et solutions de $\mathcal{P}_{i_1,i_2,\ldots,i_l}(w)$, sous certaines conditions :

Proposition I.8 *Soit* $\mathbf{k}$ *un corps fini de caractéristique 2. Dans le contexte* $(\mathbf{k}, \mathbf{F})$*, si les problèmes* $\mathcal{P}_{i_l,\ldots,i_l}(w')$ *n'admettent pas de solutions pour* $w' < w$*, alors le nombre de solutions algébriques de* $\mathcal{B}_{i_1,\ldots,i_l}(w)$ *est le nombre de solutions de* $\mathcal{P}_{i_l,\ldots,i_l}(w)$*.*

Preuve : En effet supposons qu'il existe une solution $(X_1, \ldots, X_w)$ de $\mathcal{B}_{i_1,\ldots,i_l}(w)$ telle que $X_1 = X_2$. Alors, l'effet de la caractéristique 2 est d'éliminer X_1 et X_2 dans les équations :

$$\begin{aligned} X_1^{i_1} + X_2^{i_1} + \cdots + X_w^{i_1} &= 0 \\ X_1^{i_2} + X_2^{i_2} + \cdots + X_w^{i_2} &= 0 \\ &\vdots \\ X_1^{i_l} + X_2^{i_l} + \cdots + X_w^{i_l} &= 0 \end{aligned}$$

de sorte que $X_3, \ldots, X_w$ est une solution de $\mathcal{B}_{i_1,\ldots,i_l}(w-2)$. En éliminant ainsi les X_i égaux, on obtient une solution de $\mathcal{B}_{i_1,\ldots,i_l}(w')$, où $w' < w$ et où les X_i sont tous distincts. C'est une solution de $\mathcal{P}_{i_l,\ldots,i_l}(w')$, qui par hypothèse n'en admet pas. Toutes les solutions $(X_1, \ldots, X_w)$ de $\mathcal{B}_{i_1,\ldots,i_l}(w)$ vérifient donc $X_i \neq X_j$ si $i \neq j$, et sont des solutions de $\mathcal{P}_{i_l,\ldots,i_l}(w)$.

□

Exemple 10 $\mathbf{k} = \mathbb{F}_2$, $\mathbf{F} = \mathbb{F}_{32}$, $A_1 = A_3 = A_5 = 0$, $w = 5$. Le système de générateurs est

$$\begin{aligned} X_1 + X_2 + X_3 + X_4 + X_5 &= 0 \\ X_1^3 + X_2^3 + X_3^3 + X_4^3 + X_5^3 &= 0 \\ X_1^5 + X_2^5 + X_3^5 + X_4^5 + X_5^5 &= 0 \\ X_1^{31} = X_2^{31} = X_3^{31} = X_4^{31} = X_5^{31} &= 1 \end{aligned}$$

Et la base standard obtenue se réduit à (1), ce qui était prévisible par le théorème I.10.

Exemple 11 $\mathbf{k} = \mathbb{F}_2$, $\mathbf{F} = \mathbb{F}_{32}$, $A_1 = A_3 = 0$, $w = 5$. Le système de générateur est

$$\begin{aligned} X_1 + X_2 + X_3 + X_4 + X_5 &= 0 \\ X_1^3 + X_2^3 + X_3^3 + X_4^3 + X_5^3 &= 0 \\ X_1^{31} = X_2^{31} = X_3^{31} = X_4^{31} = X_5^{31} &= 1 \end{aligned}$$

Le calcul de la base standard indique qu'il y a 22320 solutions. En identifiant les solutions $X_1, X_2, \ldots, X_w$ équivalentes par permutation des indices, on obtient $22320/120 = 168$ solutions.

Pour discriminer les inconnues égales, il est possible de rajouter les équations $(X_i - X_j)^n = 1$, $1 \leq i < j \leq w$, ce qui entraîne $X_i \neq X_j$. Toutefois, lorsque le nombre d'indéterminées devient trop grand, et la taille du corps trop élevée, les algorithmes de calcul standard, de forte complexité, deviennent impraticables. En effet le degré du système est n, et les cas pratiques qui nous intéressent sont pour $n \geq 63$. De plus le nombre de solutions du système $\mathcal{B}_{i_1,\ldots,i_l}(w)$, qui ne discrimine pas les solutions équivalentes à permutation près, est $w!$ fois le nombre de solutions de $\mathcal{I}_{i_1,\ldots,i_l}(w)$. Or moins il y a de solutions, mieux on se porte, du point de vue du calcul des bases standards.

3.2.4 Utilisation sur le système des identités de Newton

Pour déterminer l'existence de solutions au problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$, nous calculons directement une base standard de l'idéal engendré par les équations du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$.

Exemple 12 En reprenant l'exemple $\mathbf{k} = \mathrm{F}_2$, $\mathbf{F} = \mathrm{F}_{32}$, $A_1 = A_3 = 0$, $w = 5$, on trouve 186 points en dessous de l'escalier de la base standard.

Exemple 13 Nous donnons aussi la base standard obtenue pour l'ordre lexicographique obtenue pour $\mathbf{k} = \mathrm{F}_2, \mathbf{F} = \mathrm{F}_{64}$ et $\mathcal{PG}_{1\ldots8}(9)$. On voit que l'idéal est de dimension 0, il y a 2170 monômes sous l'escalier, et il n'y a pas de solutions multiples (ce qui n'est pas facile à montrer), donc il y a 2170 solutions de $\mathcal{S}_{1,\ldots,8}(9)$. De plus toutes les solutions sont dans F_{64}, c'est-à-dire qu'on peut vérifier que les polynômes $\sigma_8^{64} - \sigma_8 \ldots \sigma_2^{64} - \sigma_2$, $A_{31}^{64} - A_{31} \ldots A_{11}^{64} - A_{11}$ sont dans l'idéal engendré par les équations de Newton, en utilisant la réduction de Hironaka par rapport à la base standard obtenue. Ceci correspond bien au corollaire I.2.

Les G_i, $i = 1 \ldots 15$ sont les éléments d'une base standard obtenue pour l'ordre lexicographique. Les fonctions puissances A_{11}, A_{13} et A_{15} déterminent les autres symboles.

$$
\begin{aligned}
&\sigma_9 + A_9\ , \sigma_8 + {A_9}^5 {A_{13}}^2 + {A_9}^4 {A_{11}}^2 A_{13} + {A_9}^3 {A_{11}}^4, \sigma_7, \sigma_6 + {A_9}^6 A_{15} + {A_9}^4 {A_{11}}^3,\\
&\sigma_5, \sigma_4 + {A_9}^6 A_{13} + {A_9}^5 {A_{11}}^2, \sigma_3, \sigma_2 + {A_9}^6 A_{11}, \sigma_1,\\
&{A_9}^6 {A_{11}}^{24} {A_{13}}^7 + {A_9}^4 {A_{11}}^{28} {A_{13}}^5 + A_9 {A_{11}}^3 {A_{13}}^4\\
&+ {A_{11}}^5 {A_{13}}^3 + A_9 {A_{11}}^{18} A_{13} + {A_9}^5 {A_{11}}^{56} + {A_9}^6 {A_{11}}^{38} + {A_9}^3 {A_{11}}^{29} + A_9 {A_{11}}^2 + A_1,\\
&A_{27} + {A_9}^5 {A_{15}}^3 + A_9 {A_{11}}^{22} {A_{13}}^7 + {A_9}^6 {A_{11}}^{26} {A_{13}}^5 + {A_9}^2 {A_{11}}^3 {A_{13}}^3\\
&+ \left({A_9}^2 {A_{11}}^{34} + {A_{11}}^7\right) A_{13} + {A_{11}}^{54} + A_9 {A_{11}}^{36} + {A_9}^5 {A_{11}}^{27} + {A_9}^6 {A_{11}}^9 + {A_9}^3,\\
&A_{23} + {A_9}^3 {A_{11}}^{20} {A_{13}}^7 + A_9 {A_{11}}^{24} {A_{13}}^5 + A_9 {A_{11}}^8 {A_{13}}^4 + \left({A_{11}}^{10} + {A_9}^4 A_{11}\right) {A_{13}}^3\\
&+ \left({A_9}^4 {A_{11}}^{32} + {A_9}^5 {A_{11}}^{14} + {A_9}^2 {A_{11}}^5\right) A_{13} + {A_9}^2 {A_{11}}^{52} + {A_{11}}^{25} + {A_9}^4 {A_{11}}^{16},\\
&A_{21} + {A_9}^6 {A_{15}}^2 + {A_9}^5 {A_{13}}^3 + {A_9}^3 {A_{11}}^4 A_{13} + {A_9}^2 {A_{11}}^6,\\
&{A_{15}}^4 + {A_9}^5 A_{15} + {A_9}^5 {A_{13}}^6 + {A_9}^4 {A_{11}}^2 {A_{13}}^5 + {A_9}^2 {A_{11}}^6 {A_{13}}^3 + {A_{11}}^{10} A_{13} + {A_9}^6 {A_{11}}^{12} + {A_9}^3 {A_{11}}^3,\\
&A_{13} A_{15} + {A_{13}}^7 + {A_9}^6 {A_{11}}^2 {A_{13}}^6 + {A_9}^5 {A_{11}}^4 {A_{13}}^5 + {A_9}^4 {A_{11}}^6 {A_{13}}^4 + {A_9}^3 {A_{11}}^8 {A_{13}}^3 + {A_9}^2 {A_{11}}^{10} {A_{13}}^2\\
&+ \left(A_9 {A_{11}}^{12} + {A_9}^5 {A_{11}}^3\right) A_{13} + {A_9}^6 {A_{11}}^{32} + {A_{11}}^{14} + {A_9}^4 {A_{11}}^5,\\
&A_{11} A_{15} + \left({A_9}^3 {A_{11}}^{26} + {A_{11}}^{17}\right) {A_{13}}^7 + \left(A_9 {A_{11}}^{30} + {A_9}^5 {A_{11}}^{21} + {A_9}^6 {A_{11}}^3\right) {A_{13}}^5 + {A_9}^4 {A_{11}}^7 {A_{13}}^3\\
&+ {A_9}^3 {A_{11}}^9 {A_{13}}^2 + \left(A_9 {A_{11}}^{29} + {A_9}^5 {A_{11}}^{20} + {A_9}^6 {A_{11}}^2\right) A_{13} + {A_9}^2 {A_{11}}^{58} + {A_9}^6 {A_{11}}^{49} + {A_9}^3 {A_{11}}^{40}\\
&+ {A_9}^4 {A_{11}}^{22} + {A_9}^5 {A_{11}}^4,\\
&{A_{13}}^8 + {A_9}^6 {A_{11}}^2 {A_{13}}^7 + {A_9}^5 {A_{11}}^4 {A_{13}}^6 + {A_9}^4 {A_{11}}^6 {A_{13}}^5 + A_9 {A_{11}}^{12} {A_{13}}^2\\
&+ \left({A_9}^6 {A_{11}}^{32} + {A_9}^4 {A_{11}}^5\right) A_{13} + {A_9}^5 {A_{11}}^{34} + {A_9}^6 {A_{11}}^{16} + {A_9}^3 {A_{11}}^7,
\end{aligned}
$$

$$\left({A_{11}}^{27} + {A_9}^4{A_{11}}^{18} + {A_9}^5\right){A_{13}}^7 + \left({A_9}^5{A_{11}}^{31} + {A_9}^2{A_{11}}^{22} + {A_9}^3{A_{11}}^4\right){A_{13}}^5$$
$$+ \left({A_9}^5{A_{11}}^{30} + {A_9}^2{A_{11}}^{21} + {A_9}^3{A_{11}}^3\right)A_{13} + {A_9}^6{A_{11}}^{59}$$
$$+ {A_9}^3{A_{11}}^{50} + {A_{11}}^{41} + {A_9}^4{A_{11}}^{32} + A_9{A_{11}}^{23} + {A_9}^5{A_{11}}^{14} + {A_9}^2{A_{11}}^5,$$
$$\left({A_{11}}^{36} + A_9{A_{11}}^{18} + {A_9}^5{A_{11}}^9 + {A_9}^2\right)A_{13}, {A_{11}}^{64} + A_{11}, {A_9}^7 + 1$$

Remarque I.2 *Il serait commode de compter directement le nombre de solutions du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$ en comptant le nombre de monômes sous l'escalier d'une base standard de l'idéal engendré par le système $\mathcal{S}_{i_1,\ldots,i_l}(w)$. Pour cela, il faudrait que le système $\mathcal{S}_{i_1,\ldots,i_l}(w)$ n'admette pas de solutions multiples, ce que nous n'avons pas réussi à prouver.*

Chapitre II

Mots de poids minimum des codes correcteurs cycliques

Dans cette partie nous définissons les codes correcteurs d'erreurs, principalement les codes cycliques sur F_q de longueur première à q. Nous nous intéresserons au cas particulier des codes de longueur primitive, c'est à dire aux codes de longueur $n = q^m - 1$, pour $m > 0$.

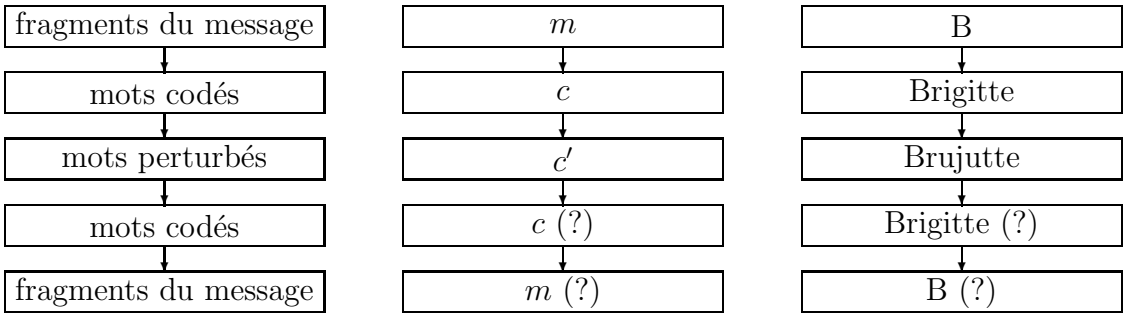
Notre objectif est de déterminer l'existence de mots de poids donnés dans un code cyclique. La difficulté du problème général pour les codes linéaires est présentée par E. R. Berlekamp, R. J. MacEliece et H. C. A. van Tilborg dans [BMvT78]. Dans le cas qui nous intéresse, ce problème se détermine en termes de fonctions puissances d'un certain nombre d'indéterminées devant être nulles, et nous pourrons l'aborder de manière très algébrique au moyen des notions introduites dans la première partie. Nous présentons quelques exemples, où une base standard décrivant l'ensemble des mots de petits poids a pu être calculée. En appendice, nous indiquerons comment traiter certains exemples, par une méthode originale mélangeant l'aspect algorithmique des bases standards et des heuristiques interactives.

Nous aborderons ce problème essentiellement pour calculer des mots de poids minimum de code cyclique. Pour cela il nous faut disposer d'une borne inférieure sur la distance minimale d'un code cyclique, pour éviter de mener des calculs inutiles. L'algorithme pour borner le rang présenté par T. Shaub dans sa thèse [Sch88], s'avère très performant, du point de vue de la borne produite. En particulier, cet algorithme présente des résultats saisissants pour les duaux des codes BCH, en produisant, en longueur 127 et 255, une borne supérieure à la borne de Carlitz-Ushiyama habituellement employée.

1 Codes correcteurs d'erreurs

1.1 Présentation

La notion de code correcteur d'erreurs remonte déjà à quelques décennies, le point de départ théorique de la recherche étant le théorème de Shannon (1948), qui assure l'existence de "bons" codes correcteurs d'erreurs. Pour ce théorème, son exposé, sa démonstration et sa signification nous renvoyons le lecteur à l'ouvrage de Van Lint d'introduction à la théorie des codes [vL82]. Nous ne nous intéressons ici uniquement aux codes en blocs (par opposition aux codes convolutionnels). Le principe des codes en blocs est le suivant :



Le message à transmettre est fragmenté en blocs de taille donnée, qui sont dans l'espace des messages $\mathcal{M}$. A chacun de ces fragments m est associé un mot c, par l'opération du *codage*. Le mot de code c est ensuite transmis sur la ligne bruitée, éventuellement entaché d'erreur pour devenir le mot c'. Le processus de décodage est d'essayer de retrouver le mot de code c, qui permettra ensuite de retrouver le fragment de message m.

Un exemple simple est le codage des lettres employé lorsqu'une ligne téléphonique est bruyante : l'auditeur dissociera mieux les mots Patricia et Brigitte que les lettres P et B. Ici les messages à transmettre sont des lettres et l'espace dans lequel on code est l'ensemble des prénoms. Les sonorités des prénoms sont suffisamment distinctes pour éviter de les confondre. Si un utilisateur sourd entend "Victorine" pour "Brigitte", il décodera la lettre "V", ce qui constitue une erreur de décodage.

Définition II.1 *Un* code C *de longueur* n *sur l'alphabet* A *est un sous ensemble de* A^n*. On notera* M *le cardinal du code* C*. On appellera* coder *un message* m *le fait de lui associer, d'une manière unique, un élément de* C*. On appellera* décoder *un mot de l'espace* A^n *le fait de lui associer de manière unique un élément de* C*.*

On espère, en décodant, obtenir le mot le plus proche du mot qui a été transmis. C'est le décodage à *maximum de vraisemblance*. Un algorithme de décodage peut aussi ne pas retourner de réponses. Ceci est important dans le cadre de certaines classes de codes (codes concaténés) où diverses stratégies de décodage sont disponibles. La théorie du décodage ne sera pas introduite ici, et nous renvoyons à la thèse de N. Sendrier [Sen91] pour un meilleur exposé sur les notions de décodage et d'algorithme de décodage.

1.2 Codes linéaires

1.2.1 L'alphabet des symboles

Dans tous les cas suivants, on considérera que A est un corps, ce qui nous permet de voir A^n comme un A-espace vectoriel. Pour tous les codes considérés dans la suite et dans cette thèse, A est un corps fini $\mathbf{k} = \mathrm{F}_q$, où q est une puissance d'un nombre premier p.

Dans la réalité, la plupart des codes utilisés en pratique sont définis sur le corps F_{2^m}, voire F_2, qui se prête à un codage facile dans les microprocesseurs.

1.2.2 Codes linéaires

La structure de l'espace ambiant ainsi enrichie, on définit la notion de code linéaire comme suit :

Définition II.2 *Un* $[n,k]$ code linéaire C *sur* $\mathbb{F}_q$*, est un sous-espace vectoriel de* $\mathbb{F}_q^n$*, de dimension* k*. La* longueur *du code* C *est* n*. On appelle* matrice génératrice *de* C *une* $[k,n]$ *matrice dont les lignes forment une base de* C*. On appelle un* mot *du code un élément de* C*.*

Définissons maintenant la distance utilisée pour les notions de "plus proche", "séparer les mots".

Définition II.3 (Distance de Hamming) *Soit* $a,b \in \mathbb{F}_q^n$*, alors la* distance de Hamming $d(a,b)$ *entre* a *et* b *est :*

$$d(a,b) = \mathcal{C}ard(\{i \in [1,n] \ \mid \ a_i \neq b_i\}).$$

Le poids $w(a)$ *d'un mot* a *est :*

$$w(a) = d(a,0) = \mathcal{C}ard(\{i \in [1,n] \ \mid \ a_i \neq 0\})$$

Soit C *un code sur* $\mathbb{F}_q$*, on note* d *la* distance minimale *de* C*, définie comme le poids minimal des mots de* C *non nuls :*

$$d = d(C) = \inf_{\{a,b \in C | a \neq b\}} d(a,b).$$

Dans le code d'un $[n,k]$ code linéaire C, la distance minimale d est égal au poids minimal des mots de C. Si d est la distance minimale du code C, alors chaque boule centrée en un mot du code de rayon $\lfloor (d-1)/2 \rfloor$ ne contient pas d'autre mot du code. En utilisant un décodage à maximum de vraisemblance, chaque mot c' à distance inférieure à e d'un mot du code c sera décodé en ce mot c. S'il y a moins de e erreurs, le décodage est correct. De plus si on est capable de déterminer si un mot de $\mathbf{k}^n$ est dans C, alors on est capable de détecter au plus $d-1$ erreurs. On dira que le code est $d-1$ détecteur.

Définition II.4 *Soit* C *un code de distance minimale* d*. La* capacité de correction *du code* C *est* $e = \lfloor (d-1)/2 \rfloor$*. La* capacité de détection *de* C *est* $d-1$*.*

Exemple 14 Le code de parité est le code $[n,n-1]$ sur $\mathbb{F}_2$, noté C_P, défini comme suit :

$$C_P = \{c = (c_0,\ldots,c_{n-1}) \in \mathbb{F}_2^n;\ \sum_{i=0}^{n-1} x_i = 0\}, \tag{II.1}$$

et le code à répétition est le $[n,1]$ code C_R défini par :

$$C_R = \{(1,\ldots,1),(0,\ldots,0)\}. \tag{II.2}$$

Le code C_P a pour dimension $n-1$ et pour distance minimale 2, sa capacité de détection est 1, et sa capacité de correction est 0, alors que le code C_R a pour dimension 1 et pour distance minimale n, sa capacité de détection est $n-1$, et sa capacité de correction est $\lfloor (n-1)/2 \rfloor$. L'un est de grande dimension mais de faible distance minimale, l'autre de faible dimension mais de grande de distance minimale.

1.2.3 Dual d'un code linéaire

Nous suivons l'exposé de [WS86], en extrayant succinctement les théorèmes qui nous intéressent, et en introduisant les mêmes confusions de notation : A_i pour les coefficients du polynôme de Mattson-Solomon, et pour les coefficients de l'énumérateur des poids.

Définition II.5 *Soit* $c = (c_0, \ldots, c_{n-1}) \in \mathbf{k}^n$ *et* $c' = (c'_0, \ldots, c'_{n-1}) \in \mathbf{k}^n$, *le* produit scalaire $c \cdot c'$ *de* c *et de* c' *est*

$$c \cdot c' = c_0 c'_0 + \cdots + c_{n-1} c'_{n-1}.$$

Définition II.6 *Soit* C *un* $[n, k]$ *code linéaire. Le* dual $C^\perp$ *de* C *est le code défini par :*

$$C^\perp = \{c \in \mathrm{F}_q^n;\ \forall c' \in C;\ c \cdot c' = 0\}.$$

Le code $C^\perp$ est un $[n, n-k]$ code.

Définition II.7 *Soit* C *un* $[n, k]$ *code linéaire. On note* A_i *le nombre de mots de* C *de poids* i, *et le* polynôme énumérateur des poids de C $W_C(X, Y)$ *est*

$$W_C(X, Y) = \sum_{i=0}^{n} A_i X^{n-i} Y^i.$$

Les distributions de poids d'un code cyclique et de son dual sont fortement reliés entre elles, par la transformée de MacWilliams. Nous donnons le théorème dans le cas binaire.

Théorème II.1 *Si* C *est un* $[n, k]$ *code linéaire sur* F_2, *de code dual* $C^\perp$, *alors*

$$W_{C^\perp}(X, Y) = \frac{1}{\mathcal{C}ard(C)} W_C(X + Y, X - Y).$$

Nous nous servirons d'une forme particulière de cette relation (les *identités de Pless*) dans le chapitre 3, pour étudier le nombre de mots de poids minimal des codes BCH 3-correcteurs.

2 Codes cycliques

2.1 Définition

La notion de code cyclique nous fournit un cadre mathématique plus riche dans un champ d'investigation plus restreint :

Définition II.8 *Un* $[n, k]$ *code linéaire* C *sur* F_q *est dit* cyclique *s'il vérifie la propriété suivante :*

$$\forall\ a = (a_0, \ldots a_{n-1}) \in C,\ (a_{n-1}, a_0, \ldots, a_{n-2}) \in\ C$$

Autrement dit un code cyclique est un code linéaire invariant par permutation circulaire sur ses coefficients. Considérons l'algèbre $\mathrm{F}_q[X]/(X^n - 1)$. Nous identifions $c = (c_0, \ldots c_{n-1}) \in \mathrm{F}_q^n$ et le polynôme $c(X) = c_0 + \cdots + c_{n-1} X^{n-1}$. La multiplication par X revient à effectuer une permutation circulaire des coordonnées de c. D'où

Proposition II.1 *Un code cyclique est un idéal de* $\mathrm{F}_q[X]/(X^n - 1)$.

Notation 5 *On notera dans toute la suite* E_n *pour* $\mathrm{F}_q[X]/(X^n - 1)$, *et la même notation* c *sera utilisée pour le polynôme* $c = c_0 + \cdots + c_{n-1} X^{n-1} \in E_n$ *et pour le mot* $c = (c_0, \ldots, c_{n-1}) \in \mathrm{F}_q^n$. *Nous supposons de plus que* n *est premier à* q.

2.2 Propriétés générales des codes cycliques

2.2.1 Polynôme générateur

L'anneau E_n est principal, donc tout idéal admet un générateur. Nous résumons ici les propriétés principales du polynôme générateur d'un code cyclique :

Théorème II.2 *Soit C un $[n,k]$ code cyclique. L'idéal C admet un polynôme générateur $g(X)$ unitaire de degré minimal. Par définition $g(X)$ est le* polynôme générateur *de C. De plus $g(X)$ est un diviseur de X^n-1.*

Le code C peut être engendré par plusieurs polynômes, même de degrés différents. Les contraintes du précédent théorème imposent l'unicité et permettent la terminologie.

2.2.2 Zéros d'un code cyclique

Notation 6 *Soit C un code cyclique de longueur n sur $\mathbf{k}=\mathrm{F}_q$, on note $\mathbf{F}=\mathrm{F}_{q^m}$ le corps de décomposition de X^n-1 sur $\mathbf{k}$, et on considérera comme fixée une racine primitive n-ième de l'unité, que l'on notera α.*

Nous rappelons la définition du polynôme de Mattson-Solomon.

Définition II.9 *Soit $c=(c_0,\ldots,c_{n-1})\in E_n$. Le* polynôme de Mattson-Solomon *de c est le polynôme $A(Z)=\sum_{i=1}^{n}A_iZ^{n-i}$, avec $A_i=c(\alpha^i)$.*

Le polynôme générateur divise X^n-1, il est donc scindé à racines simples dans F_q^*. Ceci induit la définition suivante :

Définition II.10 *Soit C un $[n,k]$ code cyclique de polynôme générateur $g(X)$, les* zéros *de C sont les zéros (dans $\mathbf{F}$) de $g(X)$.*

Propriété II.1 *Les zéros du code sont des racines n-ièmes de l'unité. Le code C de polynôme générateur $g(X)$ est l'ensemble*

$$\{c\in \mathrm{F}_q[X]/(X^n-1),\ c(z)=0,\ \forall z,\ g(z)=0\}$$

*L'*ensemble de définition *$I(C)$ de C est l'ensemble d'entiers*

$$I(C)=\{i\in[0,n-1],\ \alpha^i \textit{ est un zéro de } C\}.$$

Si i est dans l'ensemble de définition $I(C)$ de C, alors qi est dans $I(C)$. L'ensemble de définition de C est réunion de classes cyclotomiques de q modulo n disjointes.

Ceci permet de définir facilement n'importe quel code cyclique, en le caractérisant par son ensemble de définition. Par exemple les codes irréductibles :

Définition II.11 *Un code irréductible est un code cyclique dont l'ensemble de définition est $\{0,n-1\}$ privé d'une seule classe cyclotomique.*

Définition II.12 *Un mot $c\in E_n$ est* idempotent *si $c(X)^2=C(X)$.*

Proposition II.2 *Un mot $c\in E_n$ est idempotent si et seulement si les coefficients du polynôme de Mattson-Solomon de c vérifient $A_i^2=A_i$, $i=1,\ldots,n$. Soit C un code cyclique, il existe un unique idempotent e tel que C est engendré par e en tant qu'idéal. Par définition e est* l'idempotent générateur *de C.*

2.2.3 Distance minimale, Borne BCH et autres bornes

D'une manière assez surprenante, il est possible de déduire des bornes inférieures de la distance minimale d'un code cyclique au seul vu de l'ensemble de ses zéros. La borne la plus fameuse est la borne BCH du nom de ses découvreurs (Bose-Chaudhuri-Hocquenghem) :

Théorème II.3 *Soit C un code cyclique dont l'ensemble des zéros contient une séquence de δ entiers consécutifs, alors la distance minimale est supérieure ou égale à δ.*

D'autres bornes sont disponibles, qui sont des raffinements de la borne BCH [Roo82, vLW86b, Sch88, MS88, dRvL91].

La borne BCH conduit à la définition des célèbres codes BCH.

Définition II.13 *Un code de longueur n sur $\mathbb{F}_q$ est un* code BCH de distance construite δ *si son polynôme générateur $g(X)$ est égal à*

$$\mathrm{ppcm}\{\min_{\alpha^b}(X), \min_{\alpha^{b+1}}(X), \ldots, \min_{\alpha^{b+\delta-2}}(X)\},$$

où $\min_{\alpha^i}(X)$ désigne le polynôme minimal de α^i. Un mot $c \in E_n$ est dans le code si et seulement si

$$c(\alpha^b) = c(\alpha^{b+1}) = \cdots = c(\alpha^{b+\delta-2}) = 0.$$

Étant donné que l'ensemble de définition d'un code BCH contient un ensemble de $\delta-1$ entiers consécutifs, la distance minimale d'un code BCH de distance construite δ est supérieure ou égale à δ.

Si le degré du polynôme de générateur $g(X)$ est k, alors la dimension du code BCH est $n-k$. Le polynôme minimal de α^i est de degré inférieur à m, et il y a plus $\delta-1$ polynômes minimaux distincts.

Propriété II.2 *Un code BCH de longueur n et de distance construite δ a une distance minimale $d \geq \delta$ et une dimension $k \geq n - m(\delta-1)$.*

Cette remarque conduit à s'intéresser particulièrement aux codes de longueur $n = q^m - 1$, parce que les classes cyclotomiques sont petites relativement à la longueur du code, de sorte que la dimension du code reste élevée.

Définition II.14 *Un code BCH sur $\mathbb{F}_q$ est dit* primitif *(ou* de longueur primitive*) s'il est de longueur $q^m - 1$. Un code est dit code BCH* au sens strict *si $b = 1$ dans la définition II.13. Nous noterons $BCH(n, \delta)$ le code BCH primitif au sens strict de longueur $n = q^m - 1$ et de distance construite δ.*

La vraie distance minimale des codes BCH est en général inconnue, et seules quelques classes de BCH sont de distance minimale connue. Le principal but de ce qui suit est de proposer une méthode pour déterminer la vraie distance minimale des codes BCH, et pour calculer l'ensemble des mots de poids minimal, ou de faible poids. Les notions introduites dans la partie I nous permettront de transporter ce problème dans un autre contexte. De plus l'approche que nous proposons est valide pour tout code cyclique.

2.2.4 Groupe d'automorphisme des codes BCH

Définition II.15 *Soit C un $[n,k]$ code linéaire. Par définition le code étendu de C est le $[n+1,k]$ code C_e défini par*

$$C_e = \{(x_0,\ldots,x_{n-1},x_\infty) \mid (x_0,\ldots,x_{n-1}) \in C \text{ et } x_0+\cdots+x_{n-1}+x_\infty = 0\}.$$

Par exemple, dans le cas binaire, pour le $[n,n]$ code $C = \mathrm{F}_2^n$, le code C_e est le code à bit de parité, de longueur $n+1$.

Nous rappelons qu'une racine primitive n-ième est fixée, et que les localisateurs d'un mot c sont les $X_i = \alpha^{j_i}$, où $c_{j_i} \neq 0$.

Définition II.16 *Soit C un $[n,k]$ code cyclique, et C_e son code étendu. Les* localisateurs *de $c_e \in C_e$, $c_e = (c \mid c_\infty)$ sont les localisateurs $X_1,\ldots,X_w$ de c si $c_\infty = 0$, et $X_1,\ldots,X_w,0$ si $c_\infty \neq 0$ (On localise la position de parité par 0).*

Proposition II.3 *Soit C un code cyclique binaire d'ensemble de définition $I(C)$, et C_e son code étendu. Alors le mot c_e est dans le code C_e si et seulement si les fonctions puissances des localisateurs de c_e vérifient*

$$A_0 = 0, \text{ et } A_i = 0, \quad i \in I(C),$$

en convenant que $0^0 = 1$.

Preuve : Soit $c_e = (c \mid c_\infty)$ un mot de C_e et A'_i, $i = 0\ldots n-1$, les fonctions puissances des localisateurs de c_e, A_i les fonctions puissances des localisateurs de c. Si $c_\infty = 0$, alors $A'_i = A_i$ et $A'_0 = A_0 = 0$ car le mot est de poids pair. Si $c_\infty \neq 0$, alors $A'_i = A_i$, $i > 0$ et, c_e étant de poids pair, $A'_0 = 0$.

□

La définition que nous donnons d'un automorphisme de code est celle de groupe de permutations de coordonnées, comme dans [WS86].

Définition II.17 *Le groupe des permutations de coordonnées qui envoie un code C sur lui-même est le groupe d'automorphismes de C, noté $Aut(C)$.*

Pour un code cyclique, il est possible d'étiqueter les indices des coordonnées par les racines n-ièmes de l'unité. Le groupe d'automorphismes d'un code cyclique est alors un groupe de permutations du groupe des racines n-ièmes de l'unité. Opérer une permutation circulaire sur un mot c revient à multiplier ses localisateurs par α. Donc le groupe d'automorphismes d'un code cyclique contient le groupe $G = \{x \mapsto \gamma x \mid \gamma \in \mathrm{F}^\star_{q^{m'}},\ \gamma^n = 1\}$.

Exemple 15 Soit C un code cyclique de longueur n sur F_q, d'ensemble de définition $I(C)$. Pour tout mot $c \in \mathrm{F}_q^n$, soit $X_1,\ldots,X_w \in \mathrm{F}_{q^m}$ ses localisateurs. Si $c \in C$, alors

$$A_{i_1} = \cdots = A_{i_l} = 0 \quad \{i_1,\ldots,i_l\} = I(C)$$

où les A_i sont les fonctions puissances des localisateurs. Soit le mot c' de localisateurs $X_1^q,\ldots,X_w^q$, et $A'_1,\ldots,A'_n$ les fonctions puissances de ses localisateurs. Alors $A'_i = A_i^q$ et donc

$$A'_{i_1} = \cdots = A'_{i_l} = 0, \quad i_1,\ldots,i_l \in I(C).$$

Le mot c' est dans C.

Proposition II.4 *Tout code cyclique contient le groupe de Galois de* F_{q^m} *sur* F_q

Le théorème suivant est du à T. Kasami [KLP67].

Théorème II.4 ([WS86][p. 236, th. 16]) *Soit C un code BCH primitif de longueur $2^m - 1$, et de distance construite δ, et C_e le code étendu, de longueur 2^m. Le groupe d'automorphisme de C_e contient le groupe affine de F_{2^m}, c'est-à-dire les permutations des localisateurs de la forme $\beta \mapsto a\beta + b$, $a, b \in \mathrm{F}_{2^m}$, $a \neq 0$.*

Les codes étendus invariants sous le groupe affine sont les codes *affine-invariants*, qui forment une classe importante de codes, incluant les codes BCH, les codes de Reed et Muller, les codes de Reed et Solomon. Cette classe de codes a été étudiée par P. Charpin [Cha90, Cha87], et le groupe d'automorphisme de certaines classes de ces codes a été obtenu par T. Berger [Ber91].

Corollaire II.1 *Si le code BCH primitif de longueur $n = 2^m - 1$ admet un mot de poids pair w, alors il admet un mot de poids $w - 1$.*

Preuve : Soit c de poids pair w dans le code BCH. Soit le mot c_e dans le code étendu

$$c_e = (c_0, \ldots, c_{n-1}, 0),$$

et soient $X_1, \ldots, X_w$ les localisateurs de c. Puisque le groupe d'automorphisme du code étendu contient le groupe affine, le mot c'_e de localisateurs $X_1 - X_1, \ldots, X_w - X_1$ est aussi dans le code étendu, avec un symbole non nul sur la position de parité. Le mot c' obtenu par raccourci de c'_e est dans le code BCH, et son poids est $w - 1$.

□

Corollaire II.2 *La distance minimale du BCH primitif au sens strict de longueur $2^m - 1$ de distance construite δ est impaire.*

Nous traiterons aussi les codes à résidus quadratiques, et les codes de Reed et Muller raccourcis.

Définition II.18 *Soit p un nombre premier, on dit que $j \in \mathrm{N}$ est résidu quadratique modulo p, si j est un carré modulo p.*

Définition II.19 *Soit $\mathbf{k} = \mathrm{F}_p$, p premier, et soit l premier tel que l est un résidu quadratique modulo p. Le* code à résidus quadratiques $\mathcal{L}$ *de longueur l sur F_p est le code cyclique d'ensemble de définition*

$$I(\mathcal{L}) = \{i \in [1, l-1],\ i \text{ est un résidu quadratique modulo } p\}.$$

Le code $\overline{\mathcal{L}}$ d'ensemble de définition

$$I(\overline{\mathcal{L}}) = I(\mathcal{L}) \cup \{0\}$$

est le code à résidus quadratiques expurgé.

Nous donnons la définition cyclique des codes de Reed et Muller raccourcis, qui n'est pas celle introduite historiquement. Cette définition nous semble plus simple dans le contexte introduit ici. Nous nous contenterons du cas binaire.

Définition II.20 *Soit $j \in [0, 2^m - 1]$, le* 2-poids *de i, noté $w_2(i)$ est*

$$\sum_{i=0}^{m-1} j_i,$$

où $i = j_0 + j_1 2 + \cdots + j_{m-1} 2^{m-1}$ est la décomposition en base 2 de i.

Le code de Reed et Muller *raccourci de longueur $n = 2^m - 1$ sur F_2, d'ordre r, noté $RM(r, m)$ est le code C binaire de longueur $n = 2^m - 1$, d'ensemble de définition*

$$I(C) = \{i \in [0, 2^m - 1],\ w_2(i) < m - r\}.$$

Nous énonçons quelques propriétés des codes de Reed et Muller raccourci, qui sont bien connues [WS86].

Proposition II.5 *La distance minimale du code de Reed et Muller raccourci d'ordre r de longueur $n = 2^m - 1$ sur F_2 est $2^{m-r} - 1$. Les mots de poids $2^{m-r} - 1$ sont les mots dont l'ensemble des localisateurs est un sous-espace vectoriel, privé de 0. Le groupe d'automorphisme du code de Reed et Muller raccourci est le groupe linéaire.*

La propriété suivante, bien qu'admise généralement, est bien traitée dans [AK92].

Proposition II.6 *Le code de Reed et Muller raccourci d'ordre r est engendré par ses mots de poids minimal $2^{m-r} - 1$.*

3 Lien avec les équations de Newton

3.1 Introduction du problème des fonctions puissances

Nous allons formuler maintenant le problème de l'existence d'un mot de poids w dans un code cyclique en terme de problème de fonctions puissances. Nous rappelons que le problème $\mathcal{PG}_{i_1,\ldots,i_l}(w)$ dans le contexte $(\mathbf{k}, \mathbf{F})$ est le problème de la détermination de $(X_1, \ldots, X_w) \in \mathbf{F}^w$, $X_i \neq X_i$ si $i \neq j$, et de $(a_1, \ldots, a_w) \in \mathbf{k}^w$ tels que les fonctions puissances généralisées des X_i relativement aux a_i vérifient $A_i = 0$, $i \in \{i_1, \ldots, i_l\}$ (définition I.8, page 24).

Proposition II.7 *Soit C un code cyclique sur F_q de longueur $n = q^m - 1$, d'ensemble de définition $I(C) = \{i_1, \ldots, i_l\}$, et soit $c = (c_0, \ldots, c_{n-1})$ un mot de C de poids w_0. Soit $(X_1, \ldots, X_{w_0})$ les localisateurs de c, associés à $(a_{i_1}, \ldots, a_{i_{w_0}})$. Alors $(c_1, \ldots, c_{w_0}) \in {\mathrm{F}_q}^{w_0}$, $(X_1, \ldots, X_{w_0}) \in {\mathrm{F}_{q^m}}^{w_0}$ est solution du problème $\mathcal{PG}_{i_1,\ldots,i_l}(w_0)$ dans le contexte $(\mathrm{F}_q, \mathrm{F}_{q^m})$. Réciproquement à une solution non multiple de $\mathcal{PG}_{i_1,\ldots,i_l}(w_0)$ correspond un mot de C de poids w_0.*

Preuve : Soit c un mot de C, de poids w_0, et soient $X_1, \ldots, X_{w_0}$ les localisateurs de c, associés aux coefficients $c_{i_1}, \ldots, c_{i_{w_0}}$. Alors

$$\begin{aligned} c = (c_0, \ldots, c_{n-1}) \in C &\Leftrightarrow c(\alpha^i) = 0, \quad i \in I(C) \\ &\Leftrightarrow \sum_{k=0}^{n-1} c_k \alpha^{ik} = 0, \quad i \in I(C) \\ &\Leftrightarrow \sum_{k=0}^{n-1} c_k (\alpha^k)^i = 0, \quad i \in I(C) \\ &\Leftrightarrow \sum_{l=1}^{w} c_{k_l} X_{k_l}^i = 0, \quad i \in I(C) \\ &\Leftrightarrow A_i = 0, \quad i \in I(C), \end{aligned}$$

où les A_i sont les fonctions puissances généralisées des $X_1, \ldots, X_{w_0}$, relativement à $c_1, \ldots, c_{w_0}$.

Réciproquement, soit une solution non multiple $(a_1, \ldots, a_w) \in {\mathbb{F}_q}^{w_0}$, $(X_1, \ldots, X_w) \in {\mathbb{F}_{q^m}}^{w_0}$ du système $\mathcal{PG}_{i_1,\ldots,i_l}(w_0)$, il suffit de prendre le mot dont les localisateurs sont $X_1, \ldots, X_w$ et les coefficients associés à ces localisateurs sont $a_1, \ldots, a_w$.

□

Le corollaire important est le suivant

Corollaire II.3 *Le code cyclique C de longueur primitive $n = q^m - 1$ d'ensemble de définition $I(C) = \{i_1, \ldots, i_l\}$ admet un mot de poids $\leq w$ si et seulement si le système $\mathcal{S}_{i_1,\ldots,i_l}(w)$, dans le contexte $(\mathbb{F}_q, \mathbb{F}_{q^m})$, admet une solution. Si la distance minimale de C est d alors le système $\mathcal{S}_{i_1,\ldots,i_l}(d)$ admet un nombre fini de solutions, qui est le nombre de mots de poids d.*

Preuve : Les solutions du système $\mathcal{S}_{i_1,\ldots,i_l}(w)$ sont en correspondance avec les solutions du problème $\mathcal{PG}_{i_1,\ldots,i_l}(w_0)$, $w_0 \leq w$, par le théorème I.13. Les solutions de $\mathcal{PG}_{i_1,\ldots,i_l}(w_0)$, $w_0 \leq w$, sont en correspondance avec les mots de C de poids w_0, $w_0 \leq w$.

□

En ce qui concerne les codes BCH, ce corollaire devient

Corollaire II.4 *Le code $BCH(n, \delta)$, primitif au sens strict, de distance construite δ, a pour vraie distance minimale δ, si et seulement le système $\mathcal{S}_{1,\ldots,\delta-1}(\delta)$ admet une solution.*

En termes d'idéaux, l'existence de solutions à $\mathcal{S}_{1,\ldots,\delta-1}(\delta)$ signifie que l'idéal engendré par les équations de Newton ne se réduit pas à (1), en en calculant une base standard.

Cela conduit à une méthode constructive pour déterminer la distance minimale d'un code cyclique. Le schéma est le suivant :

1. Déterminer un poids w "candidat", par exemple en utilisant la borne BCH.

2. Écrire le système $\mathcal{S}_{i_1,\ldots,i_l}(w)$.

3. Si ce système admet une solution, alors le code C a pour distance minimale w.

4. Si le système est contradictoire, le distance minimale est supérieure à w, et on peut écrire le système $\mathcal{S}_{i_1,\ldots,i_l}(w')$, pour $w' > w$, et recommencer le procédé.

Rappelons une méthode pour déterminer la distance minimale d'un code cyclique, qui contraste avec la notre, et qui lui est très complémentaire. Soit un code C, on construit, en utilisant une matrice génératrice de C, des mots de C. A chaque poids w trouvé, on sait que la distance minimale de C est inférieure ou égale à w. On peut ainsi engendrer tous les mots de code (pour les codes de petites dimensions), ou en engendrer un échantillon assez important. Cette méthode peut être améliorée et permet donc de borner supérieurement la distance minimale d'un code. Elle a été mise en œuvre notamment par J. L. Dornstetter [Dor82], et nous nous servirons plus loin d'un de ses résultats. Nicolas Sendrier et David Audibert dans [AS93a] ont ainsi établi la répartition des poids de tous les codes cycliques de longueur inférieure à 63. Si la dimension du code est trop élevée, et si tous les mots du code n'ont pas été engendrés, les résultats retournés sont probabilistes, mais d'autres méthodes sont utilisées [Leo88]. Notre méthode, par contraste, permet de borner inférieurement la distance minimale d'un code, en prouvant que le système $\mathcal{S}_{i_1,\ldots,i_l}(w)$ n'admet pas de solutions. Notre méthode présente aussi l'avantage de pouvoir caractériser les mots de poids minimal des codes cycliques, et d'obtenir des résultats sur la structure de ces mots (voir le chapitre 3, et l'exemple qui suit).

3.1.1 Un exemple (significatif)

Nous introduisons le code C suivant[1] : C est un $[63, 21]$ code cyclique d'ensemble de définition

$$I(C) = cl(1)\} \cup \{cl(5)\} \cup \{cl(7)\} \cup \{cl(9)\} \cup \{cl(11)\} \cup \{cl(13)\} \cup \{cl(23)\} \cup \{cl(27).$$

Cet ensemble contient $7, 8, 9, 10, 11$, soit 5 entiers consécutifs, et d'après la borne BCH, la distance minimale est supérieure ou égale à 6. Nous voulons déterminer l'existence et le nombre de mots de poids 6.

Un mot de poids 6 dans C ne peut vérifier $A_3 = 0$, car d'après la borne BCH, il devrait alors avoir un poids ≥ 18. Nous écrivons donc le système $\mathcal{S}_{1,5,7,9,11,13,23,27}(6)$ et nous introduisons la condition $A_3 \neq 0$[2].

Le calcul de la base standard, pour l'ordre lexicographique, donne le résultat suivant :

$$\left[\sigma_6 + {A_3}^2,\ \sigma_5,\ \sigma_4,\ \sigma_3 + A_3,\ \sigma_2,\ \sigma_1,\ A_{31},\ A_{21} + {A_3}^7,\ A_{15} + {A_3}^5,\ {A_3}^{21} + 1,\ A_0\right] \qquad \text{(II.3)}$$

On en déduit les propriétés suivantes

1. La base standard ne se réduit pas à (1), donc il y a des solutions, et le code a pour vraie distance minimale 6. (Nous savons que les solutions du système algébrique sont toutes valides par le théorème I.13)

2. Les mots de poids minimal sont dans un sous-code, défini par les relations supplémentaires $A_{31} = 0, A_0 = 0$. La relation $A_0 = 0$ était évidente car les mots de poids 6 sont de poids pair, et nécessairement vérifient $A_0 = 0$. Cette relation aurait pu être ajoutée au système $\mathcal{S}_{1,5,7,9,11,13,23,27}(6)$, mais le calcul de la base standard retrouve cette propriété.

[1] Cet exemple a été découvert par Nicolas Sendrier, grâce à un programme de calcul de distribution de poids

[2] Préférant nous en tenir aux calculs de bases standards, la condition $A_3 \neq 0$ sera introduite sous la forme $A_3^{63} = 1$

3. Il y a 21 mots de poids 6. En effet, tous les A_i sont uniquement déterminés, sauf A_3 qui vérifie $A_3^{21} = 1$, et ce polynôme est scindé dans $\mathbb{F}_{64}$.

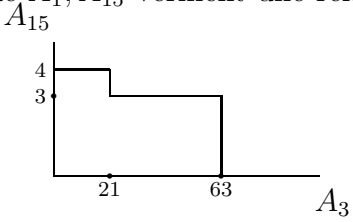
   Leurs polynômes localisateurs sont de la forme $A_3^2 Z^6 + A_3 Z^3 + 1$. Or comme A_3 vérifie $A_3^{21} = 1$, on peut écrire $A_3 = \gamma^3$ pour γ dans $\mathbb{F}_{63}$. Leurs polynômes localisateurs sont donc de la forme $Y^2 + Y + 1$ avec $Y = (\gamma Z)^3$. Le polynôme $Y^2 + Y + 1$ est scindé dans $\mathbb{F}_{63}$ avec les racines α^{21}, α^{42}, et les localisateurs sont donc $\alpha^7/\gamma, \alpha^{28}/\gamma, \alpha^{56}/\gamma, \alpha^{14}/\gamma, \alpha^{49}/\gamma, \alpha^{35}/\gamma$, pour $\gamma \in \mathbb{F}_{64}^\star$.

4. En prenant $A_3 = 1$ (c'est-à-dire $\gamma = 1$), on obtient un mot tel que tous les coefficients du polynôme localisateur sont dans $\mathbb{F}_2$. Ce mot est donc idempotent, qui n'admet que 21 conjugués par permutation circulaire, qui sont tous les mots de poids minimal. Le mot idempotent est $x^7 + x^{14} + x^{28} + x^{35} + x^{49} + x^{56}$.

Continuons notre étude. Nous nous posons le problème des mots de poids ≤ 12. Nous écrivons à nouveau le système $\mathcal{S}_{1,5,7,9,11,13,23,27}(12)$, avec la contrainte $A_3 \neq 0$. Les solutions sont

$$\begin{aligned}
&[\sigma_{12} + {A_3}^2\sigma_6 + {A_3}^{62}A_{15}, \sigma_{11}, \sigma_{10} + {A_3}^2\sigma_4, \sigma_9 + A_3\sigma_6 + {A_3}^3, \sigma_8 + {A_3}^2\sigma_2, \sigma_7 + A_3\sigma_4,\\
&\left(A_{15} + {A_3}^5\right)\sigma_6 + {A_3}^{13}{A_{15}}^3 + \left({A_3}^{60} + {A_3}^{39} + {A_3}^{18}\right){A_{15}}^2 + \left({A_3}^{44} + {A_3}^2\right)A_{15} + {A_3}^{49} + {A_3}^7,\\
&\left({A_3}^{21} + 1\right)\sigma_6 + \left({A_3}^{55} + {A_3}^{34}\right){A_{15}}^2 + \left({A_3}^{39} + {A_3}^{18}\right)A_{15} + {A_3}^{23} + {A_3}^2,\\
&\sigma_5 + A_3\sigma_2, \left(A_{15} + {A_3}^5\right)\sigma_4, \left({A_3}^{21} + 1\right)\sigma_4, \sigma_3 + A_3, \left(A_{15} + {A_3}^5\right)\sigma_2, \left({A_3}^{21} + 1\right)\sigma_2, \sigma_1, A_{31},\\
&A_{21} + {A_3}^{13}{A_{15}}^3 + \left({A_3}^{60} + {A_3}^{39} + {A_3}^{18}\right){A_{15}}^2 + \left({A_3}^{44} + {A_3}^2\right)A_{15} + {A_3}^{49},\\
&{A_{15}}^4 + \left({A_3}^{52} + {A_3}^{31} + {A_3}^{10}\right){A_{15}}^2 + {A_3}^{36}A_{15} + {A_3}^{62},\\
&\left({A_3}^{21} + 1\right){A_{15}}^3 + \left({A_3}^{26} + {A_3}^5\right){A_{15}}^2 + \left({A_3}^{31} + {A_3}^{10}\right)A_{15} + {A_3}^{57} + {A_3}^{15},\\
&{A_3}^{63} + 1, A_0]
\end{aligned}$$

Soit I l'idéal engendré par les équations de $\mathcal{S}_{1,5,7,9,11,13,23,27}(12)$, nous représentons l'escalier de $I \cup \mathbb{F}_2[A_3, A_{15}]$ (en effet, les équations faisant intervenir les σ_i ne contraignent pas les A_i, et de plus tous les A_i autres que A_1, A_{15} vérifient une relation de degré 1).



Il apparaît sur l'escalier de l'idéal qu'il y a $210 = (63 \times 3 + 1 \times 21)$ couples (A_3, A_{15}) qui sont solutions de $\mathcal{S}_{1,5,7,9,11,13,23,27}(12)$. Il y a ainsi 210 mots de poids ≤ 12.

Pour déterminer le nombre de mots de poids < 12, nous calculons une base standard de $\mathcal{S}_{1,5,7,9,11,13,23,27}(11)$. On obtient les mêmes uplets $(A_0, \ldots, A_{n-1})$ que les solutions de la base standard obtenue en II.3, qui correspondent à tous les mots de poids 6. Conclusion : il n'y a pas de mots de poids strictement compris entre 6 et 12. Il y a 21 mots de poids 6 et 189 mots de poids 12, et ces mots sont dans le sous-code défini par les relations $A_{31} = 0$ et $A_0 = 0$, c'est à dire le sous-code dont l'idempotent $x^7 + x^{14} + x^{28} + x^{35} + x^{49} + x^{56}$ est le polynôme générateur.

3.1.2 D'autres exemples

Nous présentons quelques exemples de base standard pour des codes de petites longueurs. Dans tous ces exemples, nous notons I l'idéal engendré par le système des équations de Newton $\mathcal{S}_{i_1,\ldots,i_l}(w)$, où l'ensemble de définition de C est $I(C) = \{i_1, \ldots, i_l\}$ et w est la distance minimale de C.

Nous rappelons que le nombre de solutions est égal au nombre de monômes en dessous de l'escalier, ces solutions étant comptés avec multiplicité. Dans tous les exemples que nous avons rencontrés, nous avons du vérifier que le système $\mathcal{S}_{i_1,\ldots,i_l}(w)$ n'admet pas de solutions multiples, le plus souvent par des calculs de résultants, que nous ne détaillerons pas ici. Comme cela a déjà été signalé dans la remarque I.2, il serait souhaitable de prouver que les systèmes $\mathcal{S}_{i_1,\ldots,i_l}(w)$ n'admettent pas de solutions multiples, ce qui éviterait cette vérification à chaque calcul.

Nous rappelons aussi qu'un mot c est idempotent si et seulement si les coefficients A_i du polynôme de Mattson-Solomon de c vérifient $A_i^2 = A_i$. C'est une condition qu'il est facile de tester au vu d'une base standard de l'idéal engendré par $\mathcal{S}_{i_1,\ldots,i_l}(w)$.

Le dual du BCH de longueur 63 et de distance construite 7 La distance minimale de ce code est 16, et la base standard, pour l'ordre lexicographique, du système $\mathcal{S}_{0,1,3,5,7,9,11,13,21,27}(16)$ est

$$[\sigma_{16} + {A_{15}}^{20}\ A_{31} + {A_{15}}^{19}\ {A_{23}}^2,\ \sigma_{15} + A_{15},\ \sigma_{14} + {A_{15}}^{12}\ A_{23},\ \sigma_{12},\ \sigma_{10},\ \sigma_8 + {A_{15}}^{20}\ A_{23},\ \sigma_6,$$
$$\sigma_4,\ \sigma_2,\ {A_{31}}^3 + {A_{15}}^{20}\ {A_{23}}^2\ {A_{31}}^2 + {A_{15}}^2,\ {A_{23}}^3 + {A_{15}}^{13},\ {A_{15}}^{21} + 1]$$

Il y a $189 = 3 \times 3 \times 21$ solutions distinctes (comptées sur les exposants de A_{15}, A_{23}, A_{31}) donc 189 mots de poids 16. Les mots de poids minimal ne sont dans aucun sous-code : on vérifie qu'on ne peut avoir $A_{15} = 0$, ou $A_{23} = 0$, ou $A_{31} = 0$. Il n'y a pas d'idempotents de poids minimal. En effet d'après les polynômes de la base standard, un idempotent devrait vérifier $A_{15} = 1$, $A_{23} = 1$, et $A_{31}^3 + A_{31}^2 + 1 = 0$, or ce polynôme n'a pas de zéros dans $\mathbb{F}_2$. Toutefois, on peut avoir $A_{15} = 1$ et $A_{23} = 1$, auquel cas A_{31} est racine de $X^3 + X^2 + 1$. Les polynômes localisateurs correspondant aux racines γ de $X^3 + X^2 + 1$ sont

$$\sigma(Z) = 1 + Z^8 + Z^{14} + Z^{15} + (1 + \gamma)Z^{16}.$$

Il y a trois mots qui admettent de tels polynômes localisateurs, chacun de ces mots admet 63 conjugués par permutation circulaire.

Le dual de BCH de longueur 63 et de distance construite 9 La distance minimale de ce code est 14. La base standard, pour l'ordre lexicographique, du système $\mathcal{S}_{0,1,3,5,9,11,13,21,27}(14)$ est

$$[\sigma_{14} + {A_{23}}^3\ A_{15}\ {A_7}^8 + {A_{23}}^2\ {A_{15}}^3\ {A_7}^7 + A_{23}\ {A_{15}}^5\ {A_7}^6 + {A_{15}}^7\ {A_7}^5 + {A_7}^2,$$
$$\sigma_{13} + {A_{23}}^3\ A_7 + {A_{23}}^2\ {A_{15}}^2 + A_{23}\ {A_{15}}^4\ {A_7}^8 + {A_{15}}^6\ {A_7}^7, \sigma_{12} + {A_{23}}^2\ A_{15}\ {A_7}^2 + {A_{15}}^5,$$
$$\sigma_{11} + {A_{23}}^2\ {A_7}^4 + {A_{15}}^4\ {A_7}^2, \sigma_{10} + A_{23}\ A_{15}\ {A_7}^5 + {A_{15}}^3\ {A_7}^4, \sigma_9 + A_{23}\ {A_7}^7 + {A_{15}}^2\ {A_7}^6,$$
$$\sigma_8 + A_{15}\ {A_7}^8, \sigma_7 + A_7, \sigma_6 + {A_{23}}^3 + {A_{23}}^2\ {A_{15}}^2\ {A_7}^8 + A_{23}\ {A_{15}}^4\ {A_7}^7 + {A_{15}}^6\ {A_7}^6,$$
$$\sigma_4 + {A_{23}}^2\ {A_7}^3 + {A_{15}}^4\ A_7, \sigma_2 + A_{23}\ {A_7}^6 + {A_{15}}^2\ {A_7}^5,$$
$$A_{31} + {A_{23}}^5\ {A_7}^6 + {A_{23}}^4\ {A_{15}}^2\ {A_7}^5 + A_{23}\ A_{15}\ {A_7}^8,$$

$A_{23}{}^8 + A_{23}\ A_7{}^5 + A_{15}{}^2\ A_7{}^4,$
$A_{23}{}^6\ A_{15} + A_{23}{}^5\ A_{15}{}^3\ A_7{}^8 + A_{23}{}^3\ A_{15}{}^7\ A_7{}^6 + A_{23}{}^2\ A_{15}{}^2\ A_7{}^2 + A_{15}{}^6,$
$A_{15}{}^8 + A_{15}\ A_7{}^6, A_7{}^9 + 1]$

Il y a $450 = 8\times 8\times 9 - 2\times 7\times 9$ (l'exposant du monôme $A_{23}^6 A_{15}$ ôte $2\times 7\times 9$ monômes du pavé de taille $8\times 8\times 9$) solutions distinctes. On remarque que ce code peut contenir des mots de poids minimal qui sont dans les sous-codes suivants : $I(C)\cup\{15\}$, $I(C)\cup\{15, 23, 31\}$ (pour trouver les mots éventuellement dans les sous-codes, nous ajoutons les conditions $A_i = 0$ pour un entier i qui n'est pas dans l'ensemble de définition du code, et calculons à nouveau une base standard).

Le code à résidus quadratiques binaire de longueur 31 La distance minimale est 7. La base standard du système $\mathcal{S}_{1,5,7}(7)$ est

$[\sigma_7 + A_{11}{}^4 A_3{}^{29} + A_{11}{}^2 A_3{}^{26}, \sigma_6 + A_3{}^2, \sigma_5 + A_{11} A_3{}^{29}, \sigma_4 + A_{11}{}^4\ A_3{}^{28} + A_{11}{}^2 A_3{}^{25}, \sigma_3 + A_3,$
$\sigma_2 + A_{11} A_3{}^{28}, \sigma_1, A_{15} + A_{11}{}^3 A_3{}^{25} + A_3{}^5, A_{11}{}^5 + A_{11}{}^4 A_3{}^{14} + A_{11}{}^2 A_3{}^{11} + A_{11} A_3{}^{25} + A_3{}^8,$
$A_3{}^{31} + 1]$

Il y a $155 = 31\times 5$ solutions, donc 155 mots de poids minimal. Il n'y a pas de mots de poids minimal qui soit un idempotent, et les mots de poids minimal ne sont dans aucun sous-code.

L'étendu du code à résidus quadratiques binaire de longueur 31 Les mots de poids 8. La base standard du système $\mathcal{S}_{1,5,7,0}(8)$ est

$[\sigma_8 + A_{11}{}^{14} A_3{}^3 + A_{11}{}^{12} + A_{11}{}^{11} A_3{}^{14} + A_{11}{}^9 A_3{}^{11} + A_{11}{}^6 A_3{}^{22} + A_{11}{}^5 A_3{}^5,$
$\sigma_7 + A_{11}{}^{14} A_3{}^{13} + A_{11}{}^{13} A_3{}^{27} + A_{11}{}^{11} A_3{}^{24} + A_{11}{}^8 A_3{}^4 + A_{11}{}^7 A_3{}^{18} + A_{11}{}^3 A_3{}^{12} + A_3{}^{23},$
$\sigma_6 + A_3{}^2,$
$\sigma_5 + A_{11}{}^{14} A_3{}^2 + A_{11}{}^{12} A_3{}^{30} + A_{11}{}^{11} A_3{}^{13} + A_{11}{}^9 A_3{}^{10} + A_{11}{}^6 A_3{}^{21} + A_{11}{}^5 A_3{}^4 + A_{11} A_3{}^{29},$
$\sigma_4 + A_{11}{}^{14} A_3{}^{12} + A_{11}{}^{13} A_3{}^{26} + A_{11}{}^{11} A_3{}^{23} + A_{11}{}^8 A_3{}^3 + A_{11}{}^7 A_3{}^{17} + A_{11}{}^3 A_3{}^{11} + A_3{}^{22},$
$\sigma_3 + A_3,$
$\sigma_2 + A_{11}{}^{14} A_3 + A_{11}{}^{12} A_3{}^{29} + A_{11}{}^{11} A_3{}^{12} + A_{11}{}^9 A_3{}^9 + A_{11}{}^6 A_3{}^{20} + A_{11}{}^5 A_3{}^3 + A_{11} A_3{}^{28},$
$\sigma_1,$
$A_{15} + A_{11}{}^{13} A_3{}^9 + A_{11}{}^{12} A_3{}^{23} + A_{11}{}^{10} A_3{}^{20} + A_{11}{}^9 A_3{}^3 + A_{11}{}^8 A_3{}^{17} + A_{11}{}^7$
$+ A_{11}{}^6 A_3{}^{14} + A_{11}{}^5 A_3{}^{28} + A_{11}{}^4 A_3{}^{11} + A_{11} A_3{}^{22},$
$A_{11}{}^{15} + A_{11}{}^{14} A_3{}^{14} + A_{11}{}^{12}\ A_3{}^{11} + A_{11}{}^{11} A_3{}^{25} + A_{11}{}^{10} A_3{}^8 + A_{11}{}^8 A_3{}^5 + A_{11}{}^6 A_3{}^2$
$+ A_{11}{}^4 A_3{}^{30} + A_{11}{}^3 A_3{}^{13} + A_{11}{}^2 A_3{}^{27} + A_3{}^{24},$
$A_3{}^{31} + 1]$

Il y a $465 = 31\times 15$ solutions. On vérifie qu'il n'y a pas de mots dans les sous-codes, et qu'il n'y a pas d'idempotents. Ce code a été traité très profondément dans [CCM92], où la distribution complète des poids du code a été établie, ainsi que celle de ses translatés, et les résultats concordent, pour le nombre de mots de poids minimal.

3.2 Obtention d'une borne : la méthode de Schaub

La méthode d'investigation que nous avons présentée jusqu'ici s'applique pour un poids donné. Etant donné un code cyclique C d'ensemble de définition $I(C)$, lorsque le poids w, pour lequel on veut déterminer l'ensemble des mots de poids w, est connu, on peut écrire le système $\mathcal{S}_{i_1,\ldots,i_l}(w)$, et le traiter, soit d'une manière heuristique, soit avec l'outil des bases standards.

Il reste à déterminer un bon poids w candidat. Notons que notre méthode permet de trouver le poids candidat : on calcule δ la borne BCH du code C en étudiant l'ensemble de définition, et par contradictions successives, on détermine $w \geq \delta$ tel que $\mathcal{S}_{i_1,\ldots,i_l}(w)$ a des solutions. Toutefois chaque calcul de base standard est lourd, et ce n'est guère praticable ; il faut donc être capable de déterminer le plus finement possible une borne sur la distance minimale d'un code cyclique dont on connaît l'ensemble de définition.

Nous présentons d'abord des bornes autres que la borne BCH, qui la généralisent (borne de Hartmann-Tzeng, borne de Roos). Ces bornes restent d'ordre général. Pour traiter un code donné, une méthode algorithmique, due à T. Schaub [Sch88], existe pour borner inférieurement la distance minimale d'un code donné par l'ensemble de ses zéros, d'une manière très fine. Nous présenterons cet algorithme, et l'appliquerons en particulier aux duaux des codes BCH. Pour ces codes, il apparaît que les bornes usuelles sont très en dessous de la distance minimale.

3.2.1 Autres bornes

La borne BCH est un exemple de borne inférieure sur la distance minimale d'un code à partir de son ensemble de définition. Des raffinements de cette borne existent, et nous présentons les bornes de Hartmann-Tzeng et de Roos. La borne BCH s'applique lorsque l'ensemble de définition $I(C)$ d'un code cyclique contient un ensemble de $\delta-1$ entiers consécutifs. Pour les autres bornes, le résultat que donne ces bornes est essentiellement le suivant : si un intervalle de zéros consécutifs apparaît plusieurs fois, alors la distance minimale du code est la borne BCH donnée par chacun de ces intervalles, augmentée du nombre de fois où apparaissent ces intervalles.

Théorème II.5 (Borne de Hartmann-Tzeng) *Soit C un code cyclique de longueur n et d'ensemble de définition $I(C) = \{i_1, \ldots, i_l\}$. S'il existe un entier a tel que $I(C)$ contient les $s+1$ intervalles $\{i+ja, i+1+ja, \ldots, i+\delta-2+ja\}$, $0 \leq j \leq s$, et si $\mathrm{pgcd}(a,n) < \delta$, alors la distance minimale de C est supérieure ou égale à $\delta + s$.*

La borne de Roos [Roo82, Roo83] est une généralisation de la borne de Hartmann-Tzeng :

Définition II.21 *Pour $B = \{i_1, \ldots, i_l\}$, avec $i_1 < i_2 < \cdots < i_l$, nous notons $\overline{B}$ l'intervalle d'entiers $[i_1, i_l]$. Pour $A, B \subset [1,n]$ nous notons AB l'ensemble $\{ab,\ a \in A,\ b \in B\}$.*

Théorème II.6 (Borne de Roos) *Si A est l'ensemble de définition d'un code de distance minimale d_A, et si B est un ensemble d'entiers inclus dans $[0, n-1]$ tel que $\mathcal{C}ard(\overline{B}) \leq \mathcal{C}ard(B) + d_A - 2$, alors la distance minimale du code d'ensemble de définition AB est supérieure ou égale à $\mathcal{C}ard(B) + d_A - 1$.*

Les preuves de ces deux théorèmes reposent sur des études de rang de matrices, dont le lien est fait avec les poids du code en vertu du théorème théorème II.7 ci-après. Ce

sujet est traité en profondeur dans [vLW86b], où des preuves simplifiées de ces bornes sont présentées. De plus une méthode pratique est introduite pour déterminer une borne inférieure de la distance minimale d'un code défini par son ensemble de zéros. Cette technique consiste à construire des ensembles dits *indépendants* par rapport à l'ensemble des zéros de C. La propriété est alors que le poids de chaque mot de C est supérieur au cardinal de tout ensemble indépendant. Cette technique semble être la plus fine, mais reste très heuristique, les auteurs ne propose pas d'algorithmes de construction d'ensemble admissible de cardinal maximal pour un code donné. Dans sa "dissertation", T. Schaub a introduit un algorithme pour borner inférieurement la distance minimale d'un code cyclique [Sch88].

3.2.2 Le principe

Nous rappelons le théorème I.9 :

Théorème II.7 *Soit $c = (c_0, \dots c_{n-1}) \in \mathbf{k}^n$. Le nombre de coordonnées non nulles de c est égal au rang de la matrice*

$$M_a = \begin{pmatrix} A_0 & A_1 & \dots & A_{n-2} & A_{n-1} \\ A_1 & A_2 & \dots & A_{n-1} & A_0 \\ \vdots & & & & \\ A_{n-1} & A_0 & \dots & A_{n-3} & A_{n-2} \end{pmatrix} \tag{II.4}$$

où les A_i sont ici les coefficients du polynôme de Mattson-Solomon de c.

Si $c = (c_0, \dots c_{n-1}) \in \mathbf{k}^n$ est dans le code cyclique d'ensemble de définition $I(C) = \{i_1, \dots, i_l\}$, alors les coefficients A_i du polynôme de Mattson-Solomon de vérifient $A_{i_1} = \dots = A_{i_l} = 0$. La distance minimale de C est donc le rang minimal de toutes les matrices M_c définies par l'équation II.4 pour tous les mots c de C. Il n'est pas question de passer un revue toutes ces matrices, et l'idée est de déterminer une borne inférieure du rang de toutes les matrices dont un certain ensemble de coefficients est nul. Par exemple pour une matrice M de $M_3(\mathbf{k})$ de la forme

$$\begin{pmatrix} \times & \times & 0 \\ \times & 0 & \times \\ 0 & \times & \times \end{pmatrix},$$

où $\times$ désigne n'importe quel élément non nul de $\mathbf{k}$, le rang de M est supérieur ou égal à 2.

3.2.3 L'algorithme pour borner le rang

Soit donnée une matrice $M_{gen} \in M_n(k)$, dont sont connus les coefficients nuls, et dont les autres coefficients sont certifiés non nuls. Nous décrivons l'algorithme de Schaub, pour borner le rang d'une telle matrice.

Nous utiliserons les trois symboles 0, Ξ^+, Ξ :

- 0 désigne 0.
- Ξ^+ désigne un élément non nul de $\mathbf{k}$.
- Ξ désigne un élément de k, dont la nullité ou non-nullité n'est pas assurée.

Et nous munirons $\{0, \Xi, \Xi^+\}$ de l'arithmétique suivante :

$+$	0	Ξ	Ξ^+
0	0	Ξ	Ξ^+
Ξ	Ξ	Ξ	Ξ
Ξ^+	Ξ^+	Ξ	Ξ

pour l'addition, et

$*$	0	Ξ	Ξ^+
0	0	0	0
Ξ	0	Ξ	Ξ
Ξ^+	0	Ξ	Ξ^+

, pour la multiplication. Nous dirons qu'en élément$x$ de $\{0, \Xi, \Xi^+\}$ est non égal à zéro si $x \neq 0$.

Soit donc une matrice dont les coefficients sont 0 ou Ξ^+, dont nous désirons obtenir une borne inférieure du rang. Nous construisons un ensemble $S = \{i_1, \ldots, i_r\}$ de lignes nécessairement indépendantes. Le rang est alors supérieur à r. A l'initialisation, cet ensemble sera constitué de la première ligne contenant au moins un coefficient non nul.

Supposons que l'ensemble de lignes déjà prouvées indépendantes est $S = \{i_1, \ldots, i_s\}$. La ligne courante $M[i]$ est inspectée, l'algorithme essayant d'écrire $M[i]$ comme combinaison linéaire (dont les coefficients sont Ξ^+, Ξ ou 0) des lignes $M[i_1], \ldots, M[i_s]$. Si une telle combinaison est réalisable, alors la ligne suivante est inspectée, sinon l'algorithme a *démontré* que cette ligne est linéairement indépendante des lignes précédentes, quels que soient les valeurs des symboles Ξ^+ apparaissant dans la matrice M. Cette ligne est alors ajoutée à l'ensemble S et la ligne suivante est inspectée. Quand toutes les lignes ont été inspectées, la borne retournée pour le rang sera alors le cardinal de l'ensemble S.

3.2.4 Détermination de l'indépendance possible d'une ligne

L'algorithme suppose au début que la ligne courante $L = M[i]$ est combinaison linéaire des lignes déclarées indépendantes, et les coefficients $(c_1, \ldots, c_s)$ sont $(\Xi, \Xi \ldots \Xi)$, de sorte que $L = c_1 M[i_1] + \cdots + c_s M[i_s]$, où $S = \{i_1, \ldots, i_s\}$. L'étude des coefficients de la ligne L permet d'établir des propriétés des coefficients $(c[1], \ldots, c[s])$, à savoir déterminer s'ils sont nuls ou sûrement non nuls (égaux à 0 ou Ξ^+).

Nous passons en revue chaque élément de L. Pour le coefficient courant d'indice k de la ligne L, nous calculons le *tableau t des termes* de la combinaison, $t = (t[1], \ldots, t[s])$, qui est le tableau des $c[j]M[i_j, k]$, $j = 1, \ldots, s$, de sorte que $L[k] = t[1] + \cdots + t[s]$. Au début $k = 1$, et seuls les cas suivants permettent d'établir une déduction :

1. Le coefficient courant de la ligne est 0 et il n'y a qu'un terme $t[l]$, non égal à zéro dans le tableau des termes, et $t[l] = \Xi$. Le coefficient correspondant $c[l]$ de la combinaison linéaire doit être 0. Ceci respecte la règle $c\Xi^+ = 0 \Rightarrow c = 0$.

2. Le coefficient courant de la ligne est 0, il n'y a qu'un seul terme sûrement non nul ($= \Xi^+$) dans le tableau des termes. Alors la ligne est déclarée indépendante. En effet $\Xi^+\Xi^+ \neq 0$.

3. Le coefficient courant de la ligne est 0 et il y a deux termes $t[l]$ et $t[l']$ non égaux à zéro dans le tableau des termes, tels qu'un des deux, $t[l']$, soit Ξ^+. Alors dans la combinaison linéaire, le coefficient $c[l]$, correspondant à l'autre terme, $t[l]$, doit être Ξ^+. La règle est $c\Xi^+ + \Xi^+ = 0 \Rightarrow c \neq 0$.

4. Le coefficient courant de la ligne est Ξ^+ et il n'y a pas de termes autres que 0 dans le tableau des termes, alors la ligne est indépendante. En effet $c \times 0 \neq \Xi^+$.

5. Le coefficient courant de la ligne est Ξ^+ et il n'y a qu'un terme $t[l]$ différent de 0 dans le tableau des termes, alors le coefficient $c[l]$ correspondant à ce terme doit être Ξ^+. La règle est $c\Xi^+ = \Xi^+ \Rightarrow c = \Xi^+$

Le tableau suivant permet de visualiser mieux les cas qui se présentent :

coefficient courant	0	0	0	Ξ^+	Ξ^+
termes	$\begin{matrix}0\\\vdots\\\Xi \quad l\\\vdots\\0\end{matrix}$	$\begin{matrix}0\\\vdots\\\Xi^+ \quad l\\\vdots\\0\end{matrix}$	$\begin{matrix}0\\\vdots\\\Xi \quad l\\\\\Xi^+ \quad l'\\\vdots\\0\end{matrix}$	$\begin{matrix}0\\\vdots\\\\\\\vdots\\0\end{matrix}$	$\begin{matrix}0\\\vdots\\0\\\Xi \quad l\\0\\\vdots\\0\end{matrix}$
conclusion	$c_l = 0$	indépendante.	$c_l = \Xi^+$	indépendante.	$c_k = \Xi^+$

Nous progressons ainsi le long de la ligne courante. Deux cas se présentent alors. Supposons qu à la fin du parcours, aucun coefficient de la combinaison linéaire n'a été modifié. La combinaison alors établie montre que la ligne peut être combinaison linéaire des lignes précédentes. Sinon, les coefficients de la combinaison linéaire ont été modifiés, et la ligne courante est à nouveau parcourue, pour vérifier la validité de la nouvelle liste des coefficients de la combinaison linéaire. Cette boucle de parcours de la ligne s'arrête lorsque la ligne a été déclarée indépendante, ou lorsqu'aucun changement n'a eu lieu sur la combinaison.

Une description de l'algorithme en pseudo-code est donnée dans la table II.1.

3.2.5 Application aux codes cycliques

Comme on le voit, l'algorithme fonctionne pour des matrices dont les termes sont soit nuls soit sûrement non nuls. Pour un code C d'ensemble de définition $I(C) = \{i_1 \dots i_k\}$, l'algorithme tel quel retourne une borne pour les mots du code qui vérifient exactement $A_{i_1} = \cdots = A_{i_l} = 0$ et $A_i \neq 0$, $i \notin I(C)$. Cette borne n'est pas forcément juste car il peut y avoir des mots dans des sous-codes de C dont le poids minimum est plus petit que la borne retournée pour les mots strictement dans C. Par exemple les mots de poids minimal du code décrit page 57, sont en fait dans un sous-code, et vérifient des relations supplémentaires.

Pour être assuré du résultat, il faut faire fonctionner la méthode pour tous les sous-codes cycliques de C. Ceci entraîne 2^k calculs si k classes cyclotomiques ne sont pas des zéros de C. La borne sera la plus petite borne trouvée pour chacun de ces sous-codes.

De plus, pour un code cyclique C sur $\mathbb{F}_2$, l'appartenance de 0 à $I(C)$ signifie que le code C ne contient que des mots de poids pair. C'est une chose que l'algorithme de Schaub ignore, et cet algorithme peut éventuellement retourner une borne impaire pour un code à poids pair. Il faut alors majorer le résultat retourné de un. Dans les tables de résultats que nous présenterons, cette remarque est prise en compte.

Cette méthode semble très efficace : T. Schaub montre que, dans le tableau de tous les codes cycliques de longueur inférieure à 57 traitées dans [vLW86b], toutes les distances

Données ; une matrice M de taille N×N dont les entrées sont les symboles $0, \Xi^+$.

Sortie : une borne inférieure du rang de M.

```
{Initialisations}
S[1]:=C[1] {première ligne indépendante}
s:=1 {le nombre de lignes assurées indépendantes}
pour i de 1 à N faire
    pour m de 1 à s faire c[m] := Ξ
    {initialisation des coefficients à priori indéterminés}
    repétér
    {jusqu'à ce que la ligne soit indépendante,
     ou ce que la combinaison semble valide}
        l:=1; change:=false; independant:=false
        tant que l ≤ N et independant=false faire
            pour m de 1 à s faire Terms[m]:=c[m]*s[m][i] {tableau des termes}
            si M[i][l]=0 alors {discussion des cas 1,2,3}
                cas 1 : c[l]:=0; change:=true
                cas 2 : independant:=true
                cas 3 : c[l']:=Ξ+; change:=true
            si M[i][l]=Ξ+ alors {discussion des cas 4,5}
                cas 4 : independant:=true
                cas 5 : c[l]:=Ξ+; change:=true
            l:=l+1
    jusqu'à change=false ou independant=true
    si independant=true alors
        s:=s+1; S[s]:=M[i]
retourner(k)
```

Les cas 1,2,3,4,5 sont décrits page 63.

Table II.1 : Méthode de Schaub.

minimales ont été retrouvées, sauf pour 18 codes, sur un total de 147. Ceci est très surprenant car l'algorithme dispose de très peu d'information, et ne peut se prononcer, pendant son déroulement, que pour très peu de cas. Ces bons résultats se justifient en partie par la structure très particulière des matrices circulantes étudiées. Nous verrons que dans le cas des duaux des BCH, les résultats produits par cet algorithme sont très surprenants.

Une autre remarque est que cette méthode est valide pour tout corps. En effet, aucune hypothèse n'est faite sur le domaine des coefficients de la matrice, et les règles de calcul définies sur $\{0, \Xi, \Xi^+\}$ reproduisent la définition de l'intégrité d'un anneau.

3.3 Etude des duaux de BCH

3.3.1 Duaux des BCH

Curieusement, cet algorithme n'est pas très connu, et n'a pas été employé pour des codes de plus grande longueur (≥ 63). T. Schaub semble s'être contenté des codes cycliques de petite longueur. Nous avons intensivement utilisé cet algorithme pour étudier la distance minimale des duaux des BCH en longueur primitive. Nous présentons la borne classique employée pour les duaux des codes BCH, qui est la borne de Carlitz-Uchiyama. Nous présentons une amélioration de cette borne. Les résultats pratiques produits par l'algorithme de Schaub sont très en dessus des résultats théoriques.

Théorème II.8 (Borne de Carlitz-Uchiyama [CU57]) *Soit C un code BCH binaire de longueur $n = 2^m - 1$ de distance construite $\delta = 2t + 1$ où :*

$$2t - 1 < 2^{\lceil \frac{m}{2} \rceil} + 1$$

Alors les poids w de tous les mots de $C^\perp$ vérifient :

$$2^{\frac{m}{2}} - (t-1)2^{\frac{m}{2}} \leq w \leq 2^{m-1} + (t-1)2^{\frac{m}{2}}.$$

F. Levy dans [LDV92] donne une amélioration de cette borne en établissant des propriétés de divisibilité des codes étendus, en utilisant la borne de Hartmann-Tzeng, et des propriétés d'inclusion des duaux des BCH étendus dans certains codes de Reed et Muller. Ses résultats sont, à ma connaissance, les plus fins sur la distance minimale des duaux des BCH. Nous citons donc sa borne :

Définition II.22 *Un code C est dit* t-divisible *si les poids des mots de C sont multiples de t.*

Théorème II.9 *Soit $\delta \geq 3$ et l tel que $2^l + 1 \leq \delta < 2^{l+1} + 1$. Alors le dual D du BCH de distance construite δ est $2^{\lceil \frac{m}{l} \rceil - 1}$-divisible. De plus soit $t = m - l$, alors :*

1. *si $\lfloor \frac{m}{2} \rfloor \leq l < m - 2$ et $\delta < 2^{l+1} - 3$, la distance minimale de D est supérieure ou égale à $2^{t+1} - 2^{t-1}$, sauf si m est impair et $l = \lfloor \frac{m}{2} \rfloor$ et $\delta = 2^l + 1$, auquel cas la borne de Carlitz-Uchiyama est meilleure.*

2. *si $\lfloor \frac{m}{2} \rfloor \leq l < m - 2$ et $\delta \geq 2^{l+1} - 3$ alors la distance minimale de D est supérieure ou égale à 2^t.*

3. *si* $l = m - 2$ *et* $m \geq 4$, *alors la distance minimale de* D *est supérieure à 6, sauf si* $\delta = 2^{m-1} - 1$ *ou* $\delta = 2^{m-1} - 2^{\frac{m}{2}-1} - 1$ *(*m *pair), auxquels cas la distance minimale est supérieure à* 4.

4. *si* $l < \lfloor \frac{m}{2} \rfloor$ *la borne de Carlitz-Uchiyama reste meilleure.*

Nous avons utilisé la méthode de Schaub pour les duaux des BCH en longueur 127 et en longueur 255. Toutefois, pour cette dernière longueur, certains duaux de BCH étaient de dimension trop grande pour que la méthode aboutisse.

Les bornes établies par la méthode de Schaub en longueur 127 sont données dans la table II.2. Le principal commentaire qui s'en dégage est que les résultats trouvé sont bien au-dessus de la borne de Carlitz-Uchiyama améliorée par F. Levy. Ce résultat est d'autant plus surprenant que la méthode de Schaub est très simple, et n'utilise aucune des particularités des duaux des codes BCH. Des résultats semblables sont vérifiés en longueur 255, ils sont compilés dans la table II.3.

La question qui se pose naturellement alors : quelle est la vraie distance minimale des duaux des codes BCH, et surtout quelle est la meilleure borne, car il apparaît que la borne de Carlitz-Uchiyama est très en dessous de la réalité.

3.3.2 Obtention de la distance minimale

A partir des bornes dans II.2, nous avons poursuivi l'étude des mots de poids minimum en utilisant la technique des bases standards. Les résultats sont très partiels, et le traitement des équations de Newton n'a pas abouti, en général. Pour chaque dual, nous avons essayé de calculer une base standard de l'idéal $\mathcal{SI}_{I(C)}(w)$ pour le problème des idempotents , afin de déterminer si le code contient des idempotents, car les polynômes générateurs de ces systèmes sont de degré au plus 2, et l'algorithme de calcul de base standard devient praticable. Lorsqu'il n'y a pas de mots idempotents, la distance minimale est inconnue. Tous ces résultats sont pour les duaux des codes BCH en longueur 127.

Nous notons δ la distance construire du code BCH dont le dual est étudié. Le borne calculée par l'algorithme de Schaub est notée d_S.

- $\delta = 3$, $d_S = 64$. C'est la vraie distance minimale, car il s'agit du code de Reed et Muller d'ordre 1.
- $\delta = 5$, $d_S = 56$. En longueur 2^m, m impair, la distance minimale est $2^{m-1} - 2^{\frac{m-1}{2}}$. La méthode de Schaub trouve la vraie distance minimale.
- $\delta = 7$, $d_S = 48$. En longueur 2^m, m impair, la distance minimale est $2^{m-1} - 2^{\frac{m+1}{2}}$. La méthode de Schaub trouve encore la vraie distance minimale.
- $\delta = 9$, $d_S = 40$. La base standard du système $\mathcal{S}_{I(C)}(40)$ contient 1. La distance minimale est donc strictement supérieure à 40. Le prochain candidat est 44, car le code est à poids multiples de 4, mais nous n'avons pu obtenir le résultat pour $\mathcal{S}(44)$, c'est-à-dire que la base standard n'a pu être obtenue.
- $\delta = 11$, $d_S = 32$. Soit D ce code, et $I(D)$ son ensemble de définition. Le sous code D' de D, d'ensemble de définition $I(D) \cup \{15\}$ contient l'intervalle $[1, 30]$, et est donc un sous code du BCH de distance construite 31. Nous verrons dans le chapitre 3 que ce code

Distance construite	zéros du dual	CU + L	Schaub
3	[1,3,5,7,9,11,13,15,19,21,23,27,29,31,43,47,55]	64	64
5	[1,3,5,7,9,11,13,15,19,21,23,27,29,43,47,55]	56	56
7	[1,3,5,7,9,11,13,15,19,21,23,27,29,43,55]	48	48
9	[1,3,5,7,9,11,13,19,21,23,27,29,43,55]	32	40
11	[1,3,5,7,9,11,13,19,21,23,27,29,43]	24	32
13	[1,3,5,7,9,11,13,19,21,23,27,43]	16	30
15	[1,3,5,7,9,11,13,19,21,27,43]	16	28
19	[1,3,5,9,11,13,19,21,27,43]	12	22
21	[1,3,5,9,11,13,19,21,43]	12	20
23	[1,3,5,9,11,13,19,21]	12	16
27	[1,3,5,9,11,19,21]	12	14
29	[1,3,5,9,11,21]	8	14
31	[1,3,5,9,21]	8	12
43	[1,5,9,21]	6	8
47	[1,5,9]	6	8
55	[1,9]	6	6
63	[1]	4	4

La distance contruite est celle du BCH dont on étudie le dual.

0 n'est pas indiqué dans l'ensemble des zéros du dual.

CU+L désigne le résultat de l'application de la borne théorique.

Schaub est la borne obtenue par le méthode de Schaub.

Table II.2 : Borne obtenue pour la distance minimale par l'algorithme de Schaub, en longueur 127.

	Longueur 255		
Distance construite	zéros du dual	CU + L	Schaub
3	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31, 37,39,43,45,47,51,53,55,59,61,63,85,87,91,95, 111,119]	128	128
5	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31, 37,39,43,45,47,51,53,55,59,61,85,87,91,95, 111,119]	112	112
7	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31, 37,39,43,45,47,51,53,55,59,61,85,87,91,111,119]	96	96
9	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,37, 39,43,45,47,51,53,55,59,61,85,87,91,111,119]	80	86
11	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,37, 39,43,45,47,51,53,55,59,61,85,87,91,119]	64	64
13	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,37, 39,43,45,47,51,53,55,59,85,87,91,119]	48	64
15	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,37, 39,43,45,51,53,55,59,85,87,91,119]	32	60
17	[1,3,5,7,9,11,13,17,19,21,23,25,27,29,37,39, 43,45,51,53,55,59,85,87,91,119]	24	42
19	[1,3,5,7,9,11,13,17,19,21,23,25,27,29,37,39, 43,45,51,53,55,59,85,87,91]	24	42
21	[1,3,5,7,9,11,13,17,19,21,23,25,27,29,37,39, 43,45,51,53,55,85,87,91]	24	40
23	[1,3,5,7,9,11,13,17,19,21,23,25,27,29,37,39, 43,45,51,53,55,85,91]	24	32
25	[1,3,5,7,9,11,13,17,19,21,23,25,27,37,39,43, 45,51,53,55,85,91]	24	32
27	[1,3,5,7,9,11,13,17,19,21,23,25,27,37,39,43, 45,51,53,85,91]	24	32
29	[1,3,5,7,9,11,13,17,19,21,23,25,27,37,43,45, 51,53,85,91]	16	28
31	[1,3,5,7,9,11,13,17,19,21,25,27,37,43,45,51, 53,85,91]	16	26
37	[1,3,5,9,11,13,17,19,21,25,27,37,43,45,51,53, 85,91]	12	22

La distance contruite est celle du BCH dont on étudie le dual.

0 n'est pas indiqué dans l'ensemble des zéros du dual.

CU+L désigne le résultat de l'application de la borne théorique.

Schaub est la borne obtenue par le méthode de Schaub. "?" signifie que la méthode ne s'est pas arretée, car le code possédait trop de sous-codes.

Table II.3 : Borne obtenue pour la distance minimale par l'algorithme de Schaub, en longueur 255.

	Longueur 255 (suite)		
Distance construite	zéros du dual	CU + L	Schaub
39	[1,3,5,9,11,13,17,19,21,25,27,37,43,45,51,53, 85]	12	22
43	[1,3,5,9,11,13,17,19,21,25,37,43,45,51,53,85]	12	20
45	[1,3,5,9,11,13,17,19,21,25,37,43,45,51,85]	12	20
47	[1,3,5,9,11,13,17,19,21,25,37,43,51,85]	12	16
51	[1,3,5,9,11,17,19,21,25,37,43,51,85]	12	16
53	[1,3,5,9,11,17,19,21,25,37,43,85]	12	16
55	[1,3,5,9,11,17,19,21,25,37,85]	12	?
59	[1,3,5,9,11,17,19,21,37,85]	12	?
61	[1,3,5,9,11,17,21,37,85]	8	?
63	[1,3,5,9,17,21,37,85]	8	?
85	[1,5,9,17,21,37,85]	6	?
87	[1,5,9,17,21,37]	6	?
91	[1,5,9,17,37]	6	?
95	[1,5,9,17]	6	?
111	[1,9,17]	6	?
119	[1,17]	4	?
127	[1]	4	?

La distance contruite est celle du BCH dont on étudie le dual.

0 n'est pas indiqué dans l'ensemble des zéros du dual.

CU+L désigne le résultat de l'application de la borne théorique.

Schaub est la borne obtenue par le méthode de Schaub. "?" signifie que la méthode ne s'est pas arretée, car le code possédait trop de sous-codes.

Table II.4 : Borne obtenue pour la distance minimale par l'algorithme de Schaub, en longueur 255 (suite).

	Différentes longueurs pour les duaux de BCH	
Longueur	Distance construite	Schaub
511	3	255
511	5	239
511	7	215
511	9	179
511	11	128
1023	5	479
1023	7	445
1023	9	356
1023	11	256
1023	13	256
1023	15	256

Table II.5 : Borne de schaub pour les "petits" duaux pour des longueurs plus grandes.

BCH contient des mots de poids 32. Cela ne suffit pas, car $I(D') = I(BCH(31)) \cup \{43\}$, donc $I(BCH(31)) \subset I(D')$, et D' est un sous code du code $BCH(31)$, et les mots de poids 32 du code BCH ne sont pas nécessairement dans D'. Cependant, dans le chapitre 3, nous verrons que les mots de poids minimum du code BCH vérifient la relation supplémentaire $A_{43} = 0$, et donc sont dans D'. La distance minimale de D est bien 32.

- $\delta = 15$, $d_S = 28$. Nous avons trouvé les idempotents suivants, dont nous donnons les polynômes localisateurs :

$$Z^{28} + (A_{47}+1)Z^{27} + (A_{47}+1)Z^{26} + Z^{25} + Z^{24} + A_{47}Z^{23} + A_{47}Z^{22} + (A_{47}+1)Z^{20} + (A_{47}+1)Z^{19} + A_{47}Z^{18} + A_{47}Z^{16} + Z^{15} + Z^{14} + Z^{10} + (A_{47}+1)Z^8 + (A_{47}+1)Z^4 + 1$$

où $A_{47} = 0$ ou 1. Si $A_{47} = 0$ le mot est dans le sous-code défini par la seule relation supplémentaire $A_{43} = 0$, sinon le mot n'est pas dans un sous-code. Nous avons aussi obtenu un idempotent de polynôme localisateur

$$Z^{28} + Z^{27} + Z^{26} + Z^{25} + Z^{23} + Z^{22} + Z^{18} + Z^{16} + Z^{14} + Z^6 + Z^4 + Z^2 + 1,$$

Ce dernier est dans le sous-code défini par les relations supplémentaires $A_{15} = 0, A_{47} = 0, A_{55} = 0$.

- $\delta = 19$, $d_S = 22$. On obtient un idempotent de poids 22. Ce mot est de plus dans le sous-code défini par les relations supplémentaires $A_{29} = 0$, $A_{31} = 0$, et $A_{63} = 0$. Son polynôme localisateur est

$$Z^{22} + Z^{20} + Z^{18} + Z^{17} + Z^{13} + Z^{10} + Z^8 + Z^7 + Z^6 + 1.$$

- Pour les autres valeurs de δ, nous n'avons pas obtenu de résultats exacts sur la distance minimale.

4 Longueurs plus grandes

Le principal problème qui se pose dans cette étude algébrique des mots de poids minimum des codes cycliques proposée ici est la grande complexité des algorithmes de calcul de base standard. Nous avons vu, par exemple, que dés la longueur 127 les résultats ne pouvaient être obtenus, dans le cas des duaux de BCH.

Nous expliquons ici deux alternatives qui se présentent pour étudier un code cyclique C de grande longueur. La première méthode est d'utiliser des heuristiques, comme celles dégagées en I. 3, en espérant découvrir des relations que la technique des bases standards n'établit pas. La deuxième méthode est de déterminer une forme *nécessaire* des polynômes localisateurs de mots de poids minimum de C. Cette expression dépend d'un certain nombre de coefficients du polynôme de Mattson-Solomon. Si ce nombre est petit, et la taille du corps pas trop grande, alors une recherche exhaustive est envisageable.

Nous présentons ces deux méthodes à travers deux exemples. Le premier exemple traite de deux codes BCH de longueur 255, dont la distance minimale était inconnue. Ce travail a été effectué avec P. Charpin et N. Sendrier. Le deuxième exemple est une classe de codes cycliques étudiées par van Lint et Wilson dans [vLW86a]. Les auteurs établissaient la distance minimale de tous les codes de cette famille, sauf pour trois longueurs. Une recherche exhaustive nous a permis de déterminer la distance minimale de ces codes.

4.1 Deux codes BCH de longueur 255

Nous avons pu traiter deux exemples de codes BCH en grande longueur (255), pour lesquels nous avons pu obtenir la vraie distance minimale. C'était les deux derniers codes BCH de cette longueur dont la distance minimale était inconnue (voir [WS86][table 9.1, page 267]). La méthode de calcul bases standards devient impraticable avec la puissance actuelle dont nous disposons.

La première preuve de ces résultats a été obtenue avec P. Charpin et N. Sendrier [ACS92], mais les preuves présentées ici sont plus simples.

Nous donnons les résultats.

Théorème II.10 *La distance minimale du code BCH primitif au sens strict de distance construite* 61 *de longueur* 255 *est 63. La distance minimale du code BCH primitif au sens strict de distance construite* 59 *de longueur* 255 *est* 61.

Les preuves de ces résultats sont reportées en appendice A. Nous attirons l'attention des spécialistes des bases standards sur la preuve de la contradiction pour le BCH de distance construite 59, qui utilise la technique de réduction de polynômes, combinée avec nos méthodes heuristiques.

4.2 Recherche exhaustive

Lorsque la longueur des codes devient très grande, un tel traitement symbolique devient impossible. Une méthode possible est de déterminer le polynôme localisateur des mots de poids minimum, en trouvant une forme *nécessaire* au polynôme localisateur des mots de poids minimum, en fonction d'un nombre minimum de fonctions puissances A_i. La deuxième étape est alors de construire effectivement des polynômes localisateurs en donnant des valeurs aux

fonctions puissances qui apparaissent dans le polynôme localisateur, et en testant s'il s'agit effectivement d'un polynôme localisateur.

Le système de calcul formel Axiom permet facilement de calculer effectivement dans le corps finis (non premiers). De cette manière, nous avons pu déterminer les cas restants du problème soulevé dans [vLW86a]. Cette étude a déjà été exposée dans [ACS90]. Cet exemple est simple et illustre bien la méthode.

Il s'agit de la famille des codes C_m cycliques de longueur $2^m - 1$ et de zéros α^1 et α^7. Van Lint et Wilson démontrent le théorème suivant :

Théorème II.11 *Le distance minimale de C_m est strictement inférieure à 5 sauf pour les cas éventuels $m = 5$, $m = 11$, $m = 13$, $m = 17$.*

Dans le cas $m = 5$, ils établissent que la distance minimale est 5. Les cas indéterminés demeurent $m = 11$, $m = 13$, $m = 17$.

La borne de Hartmann-Tzeng permet d'établir que la distance minimale δ est supérieure ou égale à 4. Nous allons prouver dans ces trois cas que la distance minimale est 4.

Supposons qu'il existe un mot x de poids 4, et considérons son polynôme localisateur $\sigma_x(Z)$:

$$\sigma_x(Z) = 1 + \sigma_1 Z + \sigma_2 Z^2 + \sigma_3 Z^3 + \sigma_4 Z^4 \ .$$

Et les fonctions puissances vérifient $A_1 = A_7 = 0$.

Les Identités de Newton donnent :

$$\begin{array}{lcll} (id_1) & : & A_1 + \sigma_1 = 0 & \Longrightarrow \ \sigma_1 = 0 \\ (id_3) & : & A_3 + \sigma_3 = 0 & \Longrightarrow \ \sigma_3 = A_3 \\ (id_5) & : & A_5 + A_3\sigma_2 = 0 & \Longrightarrow \ \sigma_2 = A_5/A_3 \\ (id_7) & : & A_5\sigma_2 + A_3\sigma_4 = 0 & \Longrightarrow \ \sigma_4 = \sigma_2^2 \end{array}$$

Puisque 3 est premier à $2^{11} - 1$, $2^{13} - 1$, et $2^{17} - 1$, nous pouvons supposer que $A_3 = 1$. Nous avons donc à déterminer s'il existe un mot du code dont le polynôme localisateur a la forme

$$\sigma_x(Z) = 1 + A_5 Z^2 + Z^3 + A_5^2 Z^4. \tag{II.5}$$

Pour un tel polynôme, la condition nécessaire et suffisante pour qu'il soit effectivement un polynôme localisateur est qu'il ait quatre racines distinctes dans $\mathbb{F}_{2^m}$, $m = 11$, $m = 13$, $m = 17$, c'est-à-dire $\sigma(Z) \mid Z^n - 1$, $n = 2^m - 1$, $m = 11$, $m = 13$, $m = 17$.

Plutôt que de calculer par l'algorithme d'Euclide le reste de la division de $Z^n - 1$ par $\sigma(Z)$, nous calculons $Z^{2^m} \bmod \sigma_x(Z)$ par carrés successifs. Le test est alors

$$Z^{2^m} \bmod \sigma_x(Z) = Z$$

Pour diminuer le nombre de calculs à effectuer dans un corps fini, nous faisons le changement d'indéterminée $Z^2 \leftarrow A_5 Z^2$, ce qui nous amène à rechercher des polynômes de la forme :

$$\sigma_x(Z) = 1 + Z^2 + Y Z^3 + Z^4,$$

où $Y = {A_5}^{-\frac{3}{2}}$.

Nous donnons trois mots dans les trois cas $m = 11$, $m = 13$, $m = 17$. Dans les trois cas nous avons $u = \overline{X} \bmod P(X)$ où P est un polynôme irréductible de degré $m = 11$, $m = 13$, $m = 17$, explicité à chaque fois.

- Cas $m = 11$:

$$\mathrm{F}_{2^{11}} = \mathrm{F}_2[X]/(X^{11} + X^2 + 1)\ ,\ u = \overline{X} \bmod (X^{11} + X^2 + 1)$$

On trouve un polynôme localisateur dont les racines sont

$$\alpha^{660}, \alpha^{487}, \alpha^{1769}, \alpha^{1178}$$

Le mot $\mathbf{x}$ dans $\mathrm{F}_2[X]/(X^{11} - 1)$ est donc $X^{487} + X^{660} + X^{1178} + X^{1769}$.

Cas $m = 13$:

-

$$\mathrm{F}_{2^{13}} = \mathrm{F}_2[X]/(X^{13} + X^4 + X^3 + X + 1)\ ,\ u = \overline{X} \bmod (X^{13} + X^4 + X^3 + X + 1)$$

On trouve un polynôme localisateur dont les racines sont

$$\alpha^{0}, \alpha^{6399}, \alpha^{2735}, \alpha^{6454}$$

Le mot est $\mathbf{x} = 1 + X^{6399} + X^{2735} + X^{6454}$.

- Cas $m = 17$:

$$\mathrm{F}_{2^{17}} = \mathrm{F}_2[X]/(X^{17} + X^3 + 1)\ ,\ u = \overline{X} \bmod (X^{13} + X^3 + 1)$$

On trouve un polynôme localisateur dont les racines sont

$$\alpha^{124733}, \alpha^{58930}, \alpha^{88726}, \alpha^{120824}$$

Le mot est $\mathbf{x} = X^{124733} + X^{58930} + X^{88726} + X^{120824}$.

Proposition II.8 *Les codes cycliques de longueur* $2^{11} - 1$, $2^{13} - 1$, $2^{17} - 1$, *de zéros* 1 *et* 7 *sont de distance minimale* 4.

Chapitre III

Mots de poids minimal des codes cycliques

Le chapitre précédent introduisait une méthode d'investigation des mots de poids minimal des codes cycliques, étant donné un code cyclique. C'est-à-dire qu'il est possible de transformer un problème d'existence de mots de faible poids dans un code cyclique en un problème de résolution de système algébrique.

Dans ce chapitre, nous utilisons les équations de Newton et la possibilité de déterminer les fonctions symétriques en fonctions de fonctions puissances, d'une manière théorique et non plus pour des exemples particuliers. Ceci nous permet de dégager un résultat général sur les mots de poids minimal des codes $BCH(2^m - 1, \delta = 2^{m-2} - 1)$, c'est-à-dire que les mots de poids minimal de ces codes sont les mots dont l'ensemble des localisateurs est un sous-espace vectoriel de dimension $m - 2$, privé de 0. Ce résultat est nouveau et permet de mieux situer le groupe d'automorphisme des $BCH(2^m - 1, \delta = 2^{m-2} - 1)$ bien qu'il ne soit pas encore complètement déterminé. Cette propriété était connue dans le cas des $BCH(2^m - 1, \delta = 2^{m-1} - 1)$, qui sont des codes de Reed et Muller dont les mots de plus petits poids sont connus. Nous montrons que ce résultat ne se généralise pas pour des $BCH(2^m - 1, \delta = 2^h - 1)$ avec h quelconque, et il semble difficile de déterminer la structure des mots de poids minimal dans ces cas-là.

La première partie de ce chapitre est un ensemble de rappels sur les polynômes *linéarisés*. La structure de ces polynômes est liée à celle de l'ensemble de leurs zéros, qui est un sous-espace vectoriel. Ces polynômes permettent de construire des mots de poids minimal des codes $BCH(2^m - 1, \delta = 2^h - 1)$.

Pour cette construction, nous utilisons un lemme pour caractériser les polynômes localisateurs de certains mots des codes BCH au sens strict. Nous ferons une digression, en utilisant ce lemme pour déterminer quels codes BCH voient leur distance minimale égale à leur distance construite, et tels qu'il existe un mot de poids minimal qui soit un idempotent.

Enfin, dans la partie 3, nous prouverons le résultat annoncé (théorème III.6), et étudierons un contre exemple, le cas des BCH de distance construite 7.

1 Polynômes linéarisés

Le but de cette partie est de parvenir à démontrer que les polynômes linéarisés sont exactement ceux dont l'ensemble des zéros est un espace vectoriel. L'exposé qui suit est emprunté

à [PJ86][ch. 6.8], et ce sont des rappels.

1.1 Définitions

Nous établissons quelques rappels.

Définition III.1 *Par définition, un polynôme $L(Z) \in \mathrm{F}_{q^m}[Z]$ est un polynôme* q-linéarisé *si et seulement si*

$$L(Z) = \sum_{i=0}^{l} l_i Z^{q^i}.$$

Soit $f(Z) = \sum_{i=0}^{s} a_i Z^i$, on note $\hat{f}(Z)$ le q-linéarisé associé *à $f(Z)$, défini par*

$$\hat{f}(Z) = \sum_{i=0}^{s} a_i Z^{q^i}.$$

On dira aussi que $f(Z)$ est le q-associé conventionnel *de $\hat{f}(Z)$.*

La terminologie de polynôme q-linéarisé vient du fait que de tels polynômes sont des F_q-applications linéaires de F_{q^m}. En particulier l'ensemble de leurs zéros est un F_q sous-espace vectoriel de F_{q^m}. Plus précisément :

Propriété III.1 *Soit $L(Z)$ un polynôme q-linéarisé. Alors l'ensemble des zéros de $L(Z)$ est un F_q-sous-espace vectoriel de F_{q^m}, et $L(\mathrm{F}_{q^m})$ est aussi un F_q sous-espace vectoriel de F_{q^m}.*

1.2 F_q-sous-espaces vectoriels de F_{q^m}

Théorème III.1 *Toute F_q-application linéaire de F_{q^m} peut se représenter d'une manière unique sous la forme $f(L)$, où L est l'automorphisme de Frobenius: $L(\beta) = \beta^q$, et f est un polynôme de $F_{q^m}[Z]$, de degré inférieur ou égal à $m-1$.*

Preuve :

Le nombre d'applications F_q-linéaires de F_{q^m} est q^{m^2}, et de même, le nombre de polynômes de $F_{q^m}[X]$ de degré inférieur ou égal $m-1$ est aussi q^{m^2}. Pour montrer que l'application qui à $p(X)$ associe $p(L)$ est injective, il suffit de montrer que seul le polynôme nul donne l'application linéaire nulle.

Soit $f(X) = a_0 + a_1 X + \ldots + a_s X^s$, avec $s < m$.

$$\forall \beta \in F_{q^m}, f(L)\beta = 0 \Rightarrow a_0\beta + a_1\beta^q + \ldots + a_s\beta^{q^s} = 0. \qquad \text{(III.1)}$$

Or $q^s < q^m$, et un polynôme non nul de degré q^s ne peut avoir plus de q^s racines. Donc $f = 0$.

□

Corollaire III.1 *Tout F_q-sous-espace vectoriel de F_{q^m} peut être vu comme l'ensemble des zéros, ou l'ensemble des images d'un polynôme de la forme $\hat{f}(X)$.*

Cela ne suffit pas, car nous voulons montrer qu'un polynôme dont l'ensemble des zéros est un sous-espace vectoriel est un polynôme linéarisé.

1.3 Un anneau non commutatif

On considère $F_{q^m}[X]$ muni de la loi de multiplication $*$ suivante, définie sur les monômes, étendue par linéarité :

$$\begin{aligned} X^i * X^j &= X^{i+j} \\ X * a &= a^q X. \end{aligned}$$

Alors $(F_{q^m}[X],+,*)$ est un anneau non commutatif. La multiplication ainsi définie correspond à la composition des opérateurs linéaires associés :

$$[f(X) * g(X)]_{X=L} = f(L) \circ g(L),$$

où L désigne l'automorphisme de Frobenius : $x \mapsto x^q$. Cet anneau est muni d'une division euclidienne à gauche.

Théorème III.2 *Soient $f(X)$, $g(X)$ dans $F_{q^m}[X]$, $g(X) \neq 0$. Il existe alors un unique couple de polynômes $(s(X), r(X))$ de $F_{q^m}[X]$ tels que*

$$f(X) = s(X) * g(X) + r(X),\ \deg(r) < deg(g).$$

Corollaire III.2 *Tout idéal à gauche de $(F_{q^m}[X], +, *)$ est principal.*

Théorème III.3 *Soit V un F_q-sous-espace vectoriel de F_{q^m}, de dimension k, et L l'automorphisme de Frobenius de F_{q^m} sur F_q. Il existe un unique F_q-sous-espace vectoriel U, de dimension $n-k$, et un unique couple de polynômes $g(X)$ et $h(X)$ respectivement de degré k et $n-k$, tels que*

$$\begin{aligned} V &= \ker g(L), \qquad U = \ker h(L), \\ U &= \operatorname{Im} g(L), \qquad V = \operatorname{Im} h(L). \end{aligned}$$

De plus $g(X)$ et $h(X)$ vérifient

$$X^m - 1 = g(X) * h(X) = h(X) * g(X).$$

Preuve : Soit V un F_q-sous-espace vectoriel de F_{q^m}, de dimension k, et soit

$$I = \{f(X) \in F_{q^m}[X] \mid f(L)(V) = \{0\}\}.$$

Alors I est un idéal à gauche de $F_{q^m}[X]$, $I \neq \{0\}$ car I contient $X^m - 1$. D'après le corollaire III.1, il existe un polynôme $f(X)$ tel que les zéros de $\hat{f}(x)$ sont exactement les éléments de V. Soit $g(X)$ le générateur de I, alors il existe $f_1(X)$ tel que $f(X) = f_1(X) * g(X)$, et :

$$f_1(L) \circ g(L)(\beta) = 0 \Longleftrightarrow \beta \in V,$$

ce qui implique que les zéros de $g(L)$ sont exactement les éléments de V, et $\ker g(L) = V$. De plus comme $X^m - 1 \in I$, il existe $h(X)$ tel que $X^m - 1 = h(X) * g(X)$. Or le monôme X^m commute avec tout polynôme, donc

$$\begin{aligned} g(X) * (X^m - 1) &= g(X) * h(X) * g(X) \\ &= (X^m - 1) * g(X). \end{aligned}$$

Comme l'anneau $(F_{q^m}[X], +, *)$ ne possède pas de diviseurs de zéro, on en déduit :

$$g(X) * h(X) = h(X) * g(X) = X^m - 1 \tag{III.2}$$

Soit maintenant $U = g(L)(F_{q^m})$. Le sous-espace U est un F_q-sous-espace vectoriel de dimension $n-k$, et $h(L)(U) = \{0\}$ en vertu de la relation III.2. Donc $\deg \hat{h}(X) \geq m-k$, et de même, $\deg \hat{g}(X) \geq k$. L'équation (III.2) donne $\deg \hat{g}(X) = k$, $\deg \hat{h}(X) = n-k$.

□

Corollaire III.3 *Soit V un F_q-sous-espace vectoriel de F_{q^m}, alors le polynôme*

$$\prod_{v \in V} (X - v)$$

est un polynôme q-linéarisé.

2 Mots de poids minimum de certains codes BCH

Nous rappelons un lemme utile pour caractériser les polynômes localisateurs des mots de code BCH. Ce lemme nous permet de prouver que la distance minimale des codes $BCH(2^m - 1, 2^h - 1)$ est 2^{h-1}. Nous utiliserons ensuite ce lemme pour étudier des codes BCH tels que leur distance minimale est atteinte par un idempotent.

2.1 Un lemme

Les équations de Newton donnent une condition nécessaire et suffisante facile pour qu'un polynôme localisateur soit le polynôme localisateur d'un mot d'un code BCH.

Lemme III.1 ([WS86][lemma 4, page 260]) *Soit $q = p^m$, p premier, et soit le polynôme $\sigma(Z) = \sum_{i=0}^{w} \sigma_i Z^i \in \mathrm{F}_{q^m}[Z]$. Alors $\sigma(Z)$ est le polynôme localisateur d'un mot dont les composantes sont 0 ou 1, appartenant au code BCH de longueur n et de distance construite δ si et seulement si les conditions suivantes sont vérifiées :*

1. *les zéros de $\sigma(Z)$ sont des racines n-ièmes distinctes de l'unité.*
2. *pour $1 \leq i < \delta$, si p ne divise pas i, alors $\sigma_i = 0$.*

Preuve : Si $\sigma(Z)$ est le polynôme localisateur d'un mot a du BCH de distance construite δ, alors les fonctions puissances symétriques des localisateurs de a vérifient $A_i = 0$, $1 \leq i < \delta$. La forme triangulaire des équations de Newton (voir chapitre 1, partie 1, équation 1.2) devient alors

$$i\sigma_i = 0, \quad 1 \leq i < \delta,$$

donc $\sigma_i = 0$, si $1 \leq i \leq \delta - 1$, et p ne divise pas i.

Réciproquement, soit $\sigma(Z)$ vérifiant les conditions 1 et 2 de l'énoncé, et soient $X_1, \ldots, X_w$ les inverses des zéros de $\sigma(Z)$. Ce sont bien les localisateurs d'un mot de longueur n, et de plus les fonctions puissances symétriques des X_i vérifient $A_i = 0$, $1 \leq i \leq \delta - 1$, par récurrence.

□

De cette manière nous pouvons établir l'existence de mots de poids minimum pour une certaine classe de codes BCH.

Théorème III.4 *La distance minimale du code BCH primitif au sens strict sur* $\mathbb{F}_q$*, de longueur* $n = q^m - 1$ *et de distance construite* $q^h - 1$ *est* $q^h - 1$*.*

Preuve : On trouve des mots de poids $q^h - 1$ de la manière suivante. Soit H un $\mathbb{F}_q$-sous-espace vectoriel de dimension h de $\mathbb{F}_{q^m}$, et soit les polynômes $\sigma(Z)$ et $l(Z)$ définis comme suit :

$$\sigma(Z) = \prod_{y \in H, y \neq 0} (1 - yZ), \quad l(Z) = \prod_{y \in H} (Z - y).$$

Alors on a la relation suivante entre $h(Z)$ et $l(Z)$:

$$\sigma(Z) = Z^{\mathcal{C}\mathrm{ard}(H)} l(Z^{-1}) = Z^{q^h}(Z^{-1}).$$

Or $l(Z)$ est un polynôme linéarisé, il s'écrit $l(Z) = \sum_{i=0}^{h} l_i Z^{q^i}$. Donc $\sigma(Z)$ a la forme :

$$\begin{aligned} \sigma(Z) = Z^{q^h} l(Z^{-1}) &= Z^{q^h} \sum_{i=0}^{h} l_i Z^{-q^i} \\ &= \sum_{i=0}^{h} l_i Z^{q^h - q^i}. \end{aligned}$$

Le polynôme $\sigma(Z)$ vérifie les conditions 1 et 2 du lemme III.1. Donc on a bien trouvé des mots de poids minimal des BCH de longueur $q^m - 1$ et de distance $\delta = q^h - 1$, en considérant des mots dont les localisateurs sont linéarisés.

□

Des prolongements de tels résultats ont été décrits dans [KL72], et repris dans [PJ86].

2.2 Recherche d'idempotents

Le lemme III.1, que nous venons de voir, peut être mis en œuvre pour rechercher des codes BCH dont la distance minimale est égale à la distance construite. Nous nous plaçons dans le cas des codes binaires.

Nous formulons la proposition suivante, qui illustrera notre démarche.

Proposition III.1 *Soit* δ *un entier impair. S'il existe un polynôme* $\sigma(Z) = \sigma_0 + \cdots + \sigma_w Z^w$ *dans* $\mathbb{F}_2[Z]$ *tel que*

$$\begin{cases} d = \deg \sigma \in \{\delta, \delta + 1\} \\ \sigma_d = \sigma_0 = 1 \\ \sigma_i = 0, \quad 1 \leq i < \delta, \ i \text{ impair}, \end{cases} \tag{III.3}$$

qui est scindé dans $\mathbb{F}_{2^m}$ *alors le code BCH de longueur* $2^m - 1$ *et de distance construite* δ *est bien défini, et contient un mot idempotent de poids* d*. Réciproquement si le code BCH de longueur* $2^m - 1$ *de distance construite* δ *contient un idempotent de poids* δ *ou* $\delta + 1$*, le polynôme localisateur de ce mot vérifie les conditions III.3.*

La distance minimale du BCH de longueur $2^m - 1$ *de distance construite* δ *est alors* δ*.*

Preuve : La réciproque est immédiate. Soit un polynôme $\sigma(Z)$ vérifiant III.3. On a $\sigma'(Z) = Z^{\delta - 1}$, et donc $\gcd(\sigma(Z), \sigma'(Z)) = 1$, car 0 n'est pas racine de $\sigma(Z)$ ($\sigma_0 = 1$). Les racines de $\sigma(Z)$ sont toutes dans $\mathbb{F}_{2^m}^{\star}$ et sont distinctes. D'après le lemme III.1, $\sigma(Z)$ est le polynôme localisateur d'un idempotent de poids w du BCH de longueur $2^m - 1$ de distance construite δ.

Si $\deg \sigma(Z) = \delta$, alors cet idempotent est de poids minimum et la distance minimale du BCH est δ. Si $\deg \sigma(Z) = \delta + 1$, alors le code BCH contient un mot de poids $\delta + 1$ et donc un mot de poids δ, en vertu du corollaire II.2 du chapitre 2.

□

Plutôt que de chercher dans un code BCH d'une longueur primitive $n = 2^m - 1$ donnée, de distance construite δ donnée, un mot de poids δ, on cherche pour un mot donné de poids δ la longueur pour laquelle le BCH de distance construite δ contient ce mot. Se donner un mot, lorsque la longueur est inconnue, revient à se donner un polynôme localisateur.

- On se donne un polynôme localisateur $\sigma(Z)$, de coefficients 0 ou 1, de degré δ, vérifiant $\sigma_i = 0$, $1 \leq i < \delta$, p - i.
- On calcule m tel que $K = \mathbb{F}_{2^m}$ est le corps de décomposition de $\sigma(Z)$.
- Alors le code BCH de longueur $2^m - 1$ et de distance construite δ admet pour vraie distance minimale δ, et le poids δ est atteint par un idempotent.

Calculer le corps de décomposition d'un polynôme sur F_2 est très facile, et peut être programmé en langage C. Ceci permet de tester un grand nombre de polynômes localisateurs. On peut se donner δ, et construire tous les polynômes vérifiant la condition 2 du lemme III.1 de degré δ ou $\delta + 1$, qui sont au nombre de $2^{(\delta+1)/2}$.

Nous présentons un algorithme simple de calcul du degré du corps de décomposition d'un polynôme dans $\mathbb{F}_2[Z]$ sans facteurs multiples. Rappelons que le polynôme $\sigma(Z)$ est scindé à racines simples dans $\mathbb{F}_{2^m}$ si et seulement si $\sigma(Z) \mid Z^{2^m} - Z$. Trouver le corps de décomposition de $\sigma(Z)$ revient à chercher m minimal tel que $\sigma(Z) \mid Z^{2^m} - Z$. Une méthode est de calculer $Z^2 \bmod \sigma(Z), \ldots, Z^{2^i} \bmod \sigma(Z), \ldots$ par élévations successives au carré, jusqu'à ce que $Z^{2^i} = Z \bmod \sigma(Z)$.

Cet algorithme, simple et efficace s'est avéré moins rapide que l'algorithme suivant.

INPUT: Un polynôme sans facteurs multiples $\sigma(Z)$ dans $\mathbb{F}_2[Z]$

1. $s_0(Z) = \sigma(Z)$, $p_0(Z) = Z$

2. pour $i \geq 0$, tant que $s_i(Z) \neq 1$, $\begin{cases} p_{i+1}(Z) &= p_i(Z)^2 \bmod s_i(Z) \\ r_{i+1}(Z) &= \gcd(s_i(Z), p_{i+1}(Z) - Z) \\ s_{i+1}(Z) &= \dfrac{s_i(Z)}{r_{i+1}(Z)} \end{cases}$

3. Soit $I = \{i, 1 \leq i \leq i_0 \mid r_i(Z) \neq 1\}$, où i_0 est le plus petit index tel que $s_{i_0}(Z) = 1$.

OUTPUT: $\mathrm{ppcm}(I)$

Cet algorithme est en fait le même que le précédent, à cette différence près les facteurs irréductibles de $\sigma(Z)$ se décomposant dans $\mathbb{F}_{2^i}$ sont ôtés au fur et à mesure.

Lemme III.2 *Pour tout i, $1 \leq i \leq i_0$, $p_i(Z) = Z^{2^i} \bmod s_{i-1}(Z)$.*

Preuve : Pour $i = 1$, $p_1(Z) = p_0(Z)^2 \bmod s_0(Z) = Z^2 \bmod s_0(Z)$.

Supposons que pour $i' < i$ nous avons $p_i(Z) = Z^{2^i} \bmod s_{i-1}(Z)$, c'est-à-dire $p_i(Z) = Z^{2^i} + \lambda_i(Z)s_{i-1}(Z)$, alors

$$\begin{aligned} p_{i+1}(Z) &= p_i(Z)^2 \bmod s_i(Z) \\ &= (Z^{2^i} + \lambda_i(Z)s_{i-1}(Z))^2 \bmod s_i(Z) \\ &= (Z^{2^i} + \lambda_i(Z)r_i(Z)s_i(Z))^2 \bmod s_i(Z) \\ &= Z^{2^{i+1}} \bmod s_i(Z) \end{aligned}$$

□

Lemme III.3 *Pour tout i, $1 \leq i \leq i_0$, $s_i(Z)$ est le produit des facteurs irréductibles de $\sigma(Z)$ de degré $> i$.*

Preuve : Si $i = 1$, alors $r_1(Z) = \text{pgcd}(\sigma(Z), Z^2 - Z)$ et $s_1(Z) = \sigma(Z)/r_1(Z)$ ne contient aucun facteur irréductible sur $\mathbb{F}_2$ de degré inférieur ou égal à 1. Supposons l'énoncé vrai pour $i' < i$, alors $s_i(Z) = s_{i-1}(Z)/r_i(Z)$, avec $r_i(Z) = \text{pgcd}(s_{i-1}(Z), Z^{2^i} - Z)$. Le polynôme s_{i-1} n'a que des facteurs irréductibles de degré $\geq i$, et les facteurs irréductibles de degré i de s_{i-1} sont des facteurs de r_i.

□

Lemme III.4 *Pour tout i, $1 \leq i \leq i_0$, $r_i(Z)$ est le produit des facteurs irréductibles de degré i de $\sigma(Z)$.*

Preuve : En effet $r_i(Z) = s_{i-1}(Z)/s_i(Z)$.

□

Proposition III.2 *L'algorithme présenté ci-dessus s'arrête et retourne le degré du corps de décomposition de $\sigma(Z)$.*

Preuve : Soit δ le degré du polynôme $\sigma(Z)$. Alors $\sigma(Z)$ ne contient pas de facteurs irréductibles de degré $> \delta$, donc $\sigma_\delta = 1$ et l'algorithme s'est arrêté avant. De plus

$$\sigma(Z) = \prod_{i=1 \ldots i_0} r_i(Z),$$

donc le corps de décomposition de $\sigma(Z)$ est le plus petit corps contenant les corps de décomposition des $r_i(Z)$, $i = 1 \ldots i_0$. Le degré de ce corps de décomposition est exactement le ppcm des i, $i = 1, \ldots i_0$ tel que $r_i \neq 1$.

□

Ce travail a été fait avec N. Sendrier [AS93b], et une table des codes BCH dont la distance minimale est atteinte par un idempotent a pu être compilée, voir la table III.2. L'algorithme de calcul du corps de décomposition est très proche de la factorisation en degré distincts (*distinct degree factorize* [Mig89], ou le fichier `ddfact.spad` de la librairie Axiom).

Données ; Un polynôme S dans $\mathrm{F}_2[Z]$, sans facteurs multiples.
Sortie : m tel que le corps de décomposition de S est F_{2^m}.
i:=0; s[0]:= S; p[0]:= Z; I:= {} **tant que** s[i] $\neq 1$ **faire** p[i+1] := p[i] (mod s[i]) r[i+1] := gcd(s[i],p[i]-Z) s[i+1] := s[i] / r[i+1] i:= i+1 **si** r[i] $\neq 1$ **alors** I:=I $\cup$ i **retourner**(ppcm(I))

Table III.1 : Algorithme du degré du corps de décomposition d'un polynôme sans facteurs multiples

Commentaires sur la table III.2 Dans la table sont rassemblés les degrés des corps de décomposition obtenus par l'algorithme, et qui sont les minimaux pour la relation de divisibilité : en effet si $\sigma(Z)$ de décompose dans F_{2^m}, alors $\sigma(Z)$ se décompose dans tout $\mathrm{F}_{2^{m'}}$ où m' est un multiple de m.

m=9, n=511

- Pour $\delta \in \{3, 7, 13, 15, 17, 19, 21, 27, 31, 35, 39, 45, 47\}$, le code $BCH(511, \delta)$ admet un idempotent de poids δ ou $\delta + 1$.
- Pour $\delta \in \{5, 9, 11, 23, 25\}$, le code $BCH(511, \delta)$ admet un mot de poids δ.
- Pour $\delta \in \{29, 37, 41, 43\}$, la véritable distance minimale est inconnue.

m=10, n=1023

- Pour $\delta \leq 49$, $\delta \neq 43$, le code $BCH(1023, \delta)$ admet un idempotent de poids δ ou $\delta + 1$.
- La véritable distance minimale du code $BCH(1023, 43)$ est inconnue.

Principalement on remarque que beaucoup de codes BCH ont leur distance minimale égale à leur distance construite et atteinte par un idempotent.

3 Problème réciproque

Nous revenons au problème des mots de poids minimum des codes BCH de longueur $2^m - 1$ de distance construite $2^h - 1$. Dans la section précédente, nous avons vu que ces codes admettaient des mots de poids $2^h - 1$, ces mots étant ceux dont l'ensemble des localisateurs formaient un espace vectoriel privé de 0 (théorème III.4). Nous nous intéressons au problème suivant : existe-t'il d'autres mots de poids minimum des codes BCH de distance construite $2^h - 1$, que ceux dont le support est un espace vectoriel ?

Nous prouvons qu'il n'y a pas d'autres mots de poids minimum dans les cas où la distance construite est $2^{m-1} - 1$, et $2^{m-2} - 1$. Le cas $2^{m-1} - 1$ était déjà connu, puisque le code $BCH(2^m - 1, 2^{m-1} - 1)$ est le code de Reed et Muller d'ordre 1.

δ	m
3	2, 3
5	4, 5, 6
7	3, 4, 7, 10
9	6, 8, 9, 10, 14, 15, 21
11	5, 6, 8, 11, 21, 28
13	8, 9, 10, 12, 13, 14, 21, 22, 33, 35
15	4, 5, 6, 7, 9, 26, 33, 39
17	8, 9, 10, 12, 14, 15, 17, 21, 35, 39, 44, 52, 55, 65, 66, 77
19	8, 9, 10, 12, 15, 19, 21, 28, 34, 35, 39, 51, 52, 65, 66, 77, 91
21	6, 7, 8, 9, 10, 11, 15, 38, 51, 57, 68, 85
23	6, 8, 10, 11, 14, 15, 21, 23, 35, 51, 52, 57, 65, 68, 76, 85, 95, 117, 119
25	8, 10, 12, 13, 15, 18, 21, 22, 25, 28, 33, 46, 57, 68, 69, 76, 77, 95, 102, 119, 133, 153
27	6, 7, 8, 9, 10, 13, 15, 33, 44, 55, 68, 69, 76, 85, 92, 115, 187
29	10, 12, 14, 15, 16, 18, 21, 25, 26, 27, 29, 35, 39, 44, 66, 68, 69, 76, 77, 92, 95, 99, 102, 114, 115, 153, 161, 171, 187, 209, 221, 715
31	5, 6, 8, 9, 14, 21, 31, 39, 44, 52, 58, 77, 87, 92, 119, 161, 209, 221, 247, 374, 561
33	10, 11, 12, 15, 16, 17, 18, 21, 27, 28, 39, 52, 62, 76, 87, 91, 92, 93, 95, 114, 115, 116, 133, 138, 145, 171, 175, 207, 247, 322
35	9, 10, 12, 14, 15, 16, 17, 21, 22, 25, 33, 35, 52, 65, 77, 78, 87, 91, 92, 93, 95, 114, 116, 124, 138, 143, 145, 152, 155, 203, 253, 299, 494, 741
37	8, 10, 12, 14, 15, 18, 19, 21, 27, 33, 34, 37, 44, 51, 52, 55, 65, 77, 78, 92, 93, 115, 116, 117, 119, 124, 138, 143, 155, 161, 174, 175, 203, 207, 217, 261, 299, 506
39	8, 9, 10, 12, 13, 15, 19, 21, 25, 28, 33, 35, 44, 51, 55, 68, 74, 77, 85, 111, 115, 116, 119, 124, 138, 145, 174, 186, 187, 217, 319, 322, 391, 406
41	10, 12, 14, 15, 16, 18, 21, 25, 26, 27, 35, 38, 39, 41, 44, 51, 57, 65, 66, 68, 77, 91, 99, 111, 116, 119, 124, 133, 138, 148, 155, 174, 184, 185, 186, 207, 209, 261, 279, 319, 341, 374, 377, 391, 437, 759, 1615, 2431
43	7, 8, 11, 12, 15, 18, 20, 27, 39, 43, 50, 52, 57, 65, 68, 76, 82, 85, 95, 102, 111, 115, 116, 123, 124, 138, 145, 148, 153, 174, 185, 186, 207, 221, 261, 279, 310, 377, 403, 437, 782, 1173
45	8, 9, 10, 11, 12, 14, 15, 21, 23, 25, 35, 39, 52, 57, 65, 68, 76, 85, 86, 91, 102, 119, 123, 124, 129, 133, 145, 148, 155, 164, 174, 186, 205, 217, 222, 247, 259, 403, 442, 493, 754
47	8, 9, 10, 12, 15, 21, 22, 23, 28, 33, 35, 47, 52, 55, 68, 76, 77, 78, 91, 95, 102, 114, 119, 123, 129, 133, 143, 148, 155, 164, 172, 174, 185, 186, 205, 215, 221, 222, 287, 325, 407, 425, 434, 493, 494, 518, 527, 551, 741, 806, 1131, 1209, 1885, 3553
49	10, 12, 14, 15, 16, 18, 21, 25, 27, 33, 35, 44, 46, 49, 52, 55, 65, 68, 69, 76, 78, 85, 94, 95, 102, 114, 117, 119, 129, 141, 143, 145, 148, 153, 164, 171, 172, 174, 186, 187, 203, 209, 215, 222, 232, 246, 248, 261, 279, 287, 299, 301, 333, 369, 407, 442, 481, 527, 551, 589, 663, 741, 986, 1131, 1209, 1479, 1771, 2387, 3059, 4199

Table III.2 : Quelques codes $BCH(2^m - 1, \delta)$ dont la distance est δ, atteinte par un idempotent.

3.1 Cas du BCH($n = 2^m - 1$,$\delta = 2^{m-1} - 1$)

Nous montrons d'abord que la réciproque est vraie dans le cas $n = 2^m - 1$, $\delta = 2^{m-1} - 1$. Dans ce cas là, le résultat était bien connu, car le code $BCH(n = 2^m - 1, \delta = 2^{m-1} - 1)$ est le code de Reed et Muller d'ordre 1. Nous en donnons cependant une démonstration personnelle, qui est une introduction au cas $\delta = 2^{m-2} - 1$, pour lequel la démonstration est sur le même principe, mais beaucoup plus technique et plus longue.

Nous notons $i_0 = 2^h - 1$. La proposition et le lemme suivant sont vrais pour tout h, et nous les utiliserons donc pour $h = m - 1$, $h = m - 2$.

Proposition III.3 *Pour tout $h \in [2, m-1]$, soit :*

$$J_h = \{2^h - 2^j \mid j \in [0, h]\} \tag{III.4}$$

Soit x un mot de poids $2^h - 1$ et soit $\sigma(Z) = \sum_{i=0}^{2^h-1} \sigma_i Z^i$ son polynôme localisateur. Alors les localisateurs de x forment un sous-espace vectoriel si et seulement si $i \notin J_h \Rightarrow \sigma_i = 0$.

Preuve : Les localisateurs de x forment un sous-espace vectoriel de dimension $m - 2$ privé de 0, si et seulement si le polynôme $L(Z) = X \prod_{i=1}^{2^{m-2}-1}(Z - X_i)$ est un polynôme linéarisé. Le polynôme localisateur de x est

$$\sigma(Z) = Z^{2^{m-2}} L(Z^{-1}),$$

dont on vérifie que les seuls coefficients non nuls sont pour les exposants éléments de J_h.

□

Montrer que les mots de poids minimum du $BCH(2^m - 1, 2^{m-1} - 1)$ sont ceux dont le support forme un espace vectoriel revient donc à montrer que $\sigma_i = 0$, $i \notin J_h$ ($h = m - 1$), pour tout mot de poids minimum du $BCH(2^m - 1, 2^{m-1} - 1)$.

Soit x un mot de poids minimum du $BCH(2^m - 1, 2^h - 1)$, et $\sigma_1, \ldots, \sigma_{i_0}$ les fonctions symétriques de ses localisateurs, nous formons l'hypothèse H_r :

$$H_r : \ r \in [1, i_0[\text{ et } r \notin J_h \Rightarrow \sigma_r = 0 \text{ et } A_{i_0+r} = 0.$$

Notons que cette hypothèse est vérifiée pour r impair, comme l'indique le lemme de caractérisation des polynômes localisateurs des codes BCH (lemme III.1).

Lemme III.5 *Soit x un mot de poids minimum du $BCH(2^m - 1, 2^h - 1)$, et $\sigma_1, \ldots, \sigma_{i_0}$ les fonctions symétriques de ses localisateurs. Soit $r < i_0$, et supposons que $H_{r'}$ est vraie pour $r' < r$. Alors*

$$A_{i_0+r} + A_{i_0}\sigma_r = 0. \tag{III.5}$$

Preuve : Nous numérotons les équations comme indiqué dans le chapitre 1, notation 1. L'équation eq_{i_0+r} est

$$eq_{i_0+r} : A_{i_0+r} + \sum_{k=1}^{i_0} A_{i_0+r-k}\sigma_k = 0$$

Examinons le terme $A_{i_0+r-k}\sigma_k$, pour $k \in [1, r[$.

- Si $k > r$ alors $i_0 + r - k < i_0$ et $A_{i_0+r-k} = 0$.

- Si $k < r$, on distingue deux cas.

  1. Si $k \notin J_h$, alors $\sigma_k = 0$, d'après H_k.
  2. Si $k \in J_h$, alors $k \geq 2^{h-1}$, et $r - k < 2^{h-1}$, ce qui implique que $r - k$ n'est pas dans J_h. L'hypothèse H_{r-k} implique alors que $A_{i_0+r-k} = 0$.

□

Nous écrirons la décomposition en base 2 d'un entier s modulo $2^m - 1$ de la manière suivante

$$\overbrace{s_0 \quad s_1 \quad \dots \quad s_{m-1}}^{m\text{bits}}$$

où $s = s_0 + s_1 2 + \cdots s_{m-1} 2^{m-1}$. Les bits de poids faible sont situés à gauche, et les bits les plus hauts nuls sont représentés. Par exemple nous écrirons, pour 2 mod 63, $2 = (0, 1, 0, 0, 0, 0)$, et nous dirons que 2 modulo 63 possède deux zéros consécutifs dans sa décomposition en base 2.

Les éléments de l'ensemble de définition du $BCH(2^m - 1, 2^{m-1} - 1)$ sont alors les entiers dans $[0, 2^m - 1[$ qui possèdent au moins deux zéros dans leur écriture binaire.

Théorème III.5 *Les mots de poids minimum du code $BCH(2^m - 1, 2^{m-1} - 1)$ sont les mots dont le support est un sous espace vectoriel de dimension $m - 1$.*

Preuve : Soit x un mot de poids minimum du $BCH(2^m - 1, 2^{m-1} - 1)$, $i_0 = 2^{m-1} - 1$, et $\sigma_1, \dots, \sigma_{i_0}$ les fonctions symétriques de ses localisateurs, nous allons prouver H_r par récurrence, pour i_0. La première étape est pour $r = 2$. Notons d'abord que $A_{i_0} \neq 0$, car x est un mot de poids i_0 : si $A_{i_0} = 0$, alors la borne BCH entraîne que le poids de x est strictement supérieur à i_0. L'équation eq_{i_0+2} devient $A_{i_0+2} + A_{i_0}\sigma_2 = 0$. Or

$$\begin{array}{lcccccc} 2 & = & 0 & 1 & 0 & \cdots & 0 & 0 \\ i_0 & = & 1 & 1 & 1 & \cdots & 1 & 0 \\ 2 + i_0 & = & 1 & 0 & 0 & \cdots & 0 & 1 \end{array}$$

c'est à dire que $2 + i_0$ contient plus de un "0" dans son écriture binaire, donc est dans l'ensemble de définition du code. Donc $A_{i_0+2} = 0$ et $\sigma_2 = 0$.

Supposons maintenant $H_{r'}$ vérifiée pour $r' < r$. L'équation eq_{i_0+r} est $A_{i_0+r} + A_{i_0}\sigma_r = 0$, en vertu du lemme III.5. Il faut montrer que si $r \notin J_{m-1}$ alors $A_{i_0+r} = 0$. Supposons $A_{i_0+r} \neq 0$. Alors $i_0 + r$ est dans la classe cyclotomique de $2^{m-1} - 1$, c'est-à-dire $i_0 + r = (2^{m-1} - 1)2^h \bmod 2^m - 1$ pour $h \in [1, m[$.

$$\begin{array}{rcl} i_0 + r = (2^{m-1} - 1)2^h \bmod 2^m - 1 & \Rightarrow & r = -2^{m-1} + 1 + 2^{h-1} - 2^h \bmod 2^m - 1 \\ & \Rightarrow & r = (2^m - 1) - 2^{m-1} + 1 + 2^{h-1} - 2^h \\ & \Rightarrow & r = 2^{m-1} + 2^{h-1} - 2^h = 2^{m-1} - 2^{h-1} \\ & \Rightarrow & r \in J_{m-1}. \end{array}$$

Donc si r n'est pas dans J_{m-1} alors $A_{i_0+r} = 0$ et $\sigma_r = 0$.

□

3.2 Cas du BCH($n = 2^m - 1$,$\delta = 2^{m-2} - 1$)

Nous reprenons ici la démonstration esquissée dans [ACS91]. Une démonstration a déjà été établie dans [ACS92], celle-ci présente l'avantage d'être plus détaillée. Le résultat a été redémontré par C. Carlet, en utilisant des techniques différentes [Car].

Nous établissons le théorème suivant :

Théorème III.6 *Les mots de poids minimum du code $BCH(2^m-1, 2^{m-2}-1)$ sont les mots dont le support est un sous-espace vectoriel de dimension $m-2$, privé de 0.*

3.2.1 Quelques points techniques

La démonstration étant assez technique et longue, nous isolons ici quelques faits dont nous nous servirons dans la démonstration. Nous notons I l'ensemble de définition du code BCH de longueur $n = 2^m - 1$, de distance construite $2^{m-2} - 1$: $I = cl(1) \cup \cdots \mathrm{cl}(2^{m-2} - 2)$.

Propriété III.2 *Soit $s \in [0, 2^m - 1]$, et soit $(s_0, \ldots, s_{m-1})$ la décomposition binaire de s. Alors $s \in I$ si et seulement si il existe $k < m$ tel que $s_k = s_{k+1 \pmod m} = 0$ et s'il existe de plus $j \in [0, 2^m - 1]$ tel que $s_j = 0$, $j \neq k, j \neq k+1 \pmod m$.*

Preuve : La décomposition en base 2 de $2^{m-2} - 1$ est

$$(1, 1, \ldots, 1, 0, 0).$$

Les entiers strictement inférieurs à $2^{m-2}-1$ possèdent donc un zéro supplémentaire dans leur décomposition en base 2. Soit maintenant un entier s possédant deux zéros consécutifs. En opérant une permutation circulaire sur s, on peut placer ces deux zéros en dernière position, et s'il existe un zéro supplémentaire, alors s est inférieur à $2^{m-2} - 1$.

□

La propriété III.2 signifie que s est dans I s'il existe deux zéros consécutifs dans la décomposition en base 2 de s, et un troisième zéro.

Nous avons besoin de quelques lemmes sur le comportement de combinaison $\alpha i + \beta j$ dans $[0, 2^m - 1]$.

Lemme III.6 *Soit $r \in [2, 2^{m-3}[$, r pair tel que $i_0 + r \notin I$. Alors :*

$$2i_0 + 3r \in I \text{ et } i_0 + 3r \in I.$$

Preuve : Si r a une configuration de la forme :

$$r = \quad (0, 0, \ldots, 0, 1, 0, \star, \ldots, \star, 0, 0, 0)$$

C'est-à-dire que le premier terme égal à 1 est suivi d'un zéro, alors $i_0 + r$ a la configuration suivante :

$$\begin{array}{rcccccccccccc} i_0 = & (1 & \ldots & 1 & 1 & 1 & 1 & \ldots & 1 & 1 & 0 & 0) \\ r = & (0 & \ldots & 0 & 1 & 0 & \star & \ldots & \star & 0 & 0 & 0) \\ i_0 + r = & (1 & \ldots & 1 & 0 & 0 & \star & \ldots & \star & \star & \star & 0), \end{array}$$

ce qui implique que $i_0 + r \in I$, ce qui est exclus.

Donc r a une configuration de la forme :

$$\begin{array}{rl} r = & (0 \ \ldots \ 0 \ 1 \ 1 \ \star \ \ldots \ \star \ 0 \ 0 \ 0) \\ r' = r/2 = & (0 \ \ldots \ 1 \ 1 \ \star \ \ldots \ \star \ 0 \ 0 \ 0 \ 0) \\ i_0 = & (1 \ \ldots \ 1 \ 1 \ 1 \ 1 \ \ldots \ 1 \ 1 \ 0 \ 0) \\ i_0 + r + r' = & (1 \ \ldots \ 0 \ 0 \ \star \ \star \ \ldots \ \star \ \star \ \star \ 0) \end{array}$$

Donc $i_0 + r + r' \in I$, et donc $2i_0 + 3r \in I$. De même :

$$\begin{array}{rl} i_0 = & (1 \ 1 \ \ldots \ 1 \ 1 \ 1 \ \ldots \ 1 \ 1 \ 1 \ 0 \ 0) \\ r = & (0 \ 0 \ \ldots \ 0 \ 1 \ 1 \ \star \ \ldots \ \star \ 0 \ 0 \ 0) \\ 2r = & (0 \ 0 \ \ldots \ 0 \ 0 \ 1 \ 1 \ \star \ \ldots \ \star \ 0 \ 0) \\ i_0 + 3r = & (1 \ 1 \ \ldots \ 1 \ 0 \ 0 \ \star \ \star \ \star \ \ldots \ \star \ 0) \end{array}$$

Et donc $i_0 + 3r \in I$.

□

Lemme III.7 *Soit $r \in]2^{m-3}, i_0[$, r pair et $r \notin J_{m-2}$. Alors il existe un élément de la même classe cyclotomique que $i_0 + 2r$ qui peut s'écrire sous la forme $i_0 + \epsilon$, avec $-i_0 < \epsilon < r$ et $\epsilon \notin J_{m-2}$.*

Preuve : Par hypothèse, $2^{m-1} - 1 < i_0 + 2r < 2^m - 1$. Nous considérons l'élément suivant dans la même classe cyclotomique que $i_0 + 2r$:

$$2(i_0 + 2r) - (2^m - 1) = i_0 + \epsilon$$

où $\epsilon = 4r + 2^{m-2} - 2^m$. Comme r est pair et que $2^{m-3} < r < 2^{m-2}$, nous avons :

$$\begin{array}{rcl} \epsilon & > & 4(2^{m-3} + 1) + 2^{m-2} - 2^m \\ & > & 4 - 2^{m-2} > -i_0. \end{array}$$

D'autre part, puisque $2^{m-2} - 2^m = -3 \cdot 2^{m-2}$,

$$\epsilon = r + 3(r - 2^{m-2}) < r,$$

car $r < i_0 < 2^{m-2}$. Supposons maintenant que ϵ est dans J_{m-2}, alors il existe j dans $[0, m-2]$ tel que $\epsilon = 2^{m-2} - 2^j$, et

$$4r + 2^{m-2} - 2^m = 2^{m-2} - 2^j \Rightarrow r = 2^{m-2} - 2^{j-2},$$

donc $r \in J_{m-2}$ ce qui est exclus.

□

3.2.2 Preuve du théorème III.6

Soit x un mot de poids minimal du $BCH(2^m - 1, 2^{m-2} - 1)$, et $\sigma_1, \ldots \sigma_{i_0}$ les fonctions symétriques de ses localisateurs. Comme dans la preuve du théorème III.3, nous formulons l'hypothèse de récurrence suivante H_r :

$$H_r : \ r \in [1, i_0[\text{ et } r \notin J_{m-2} \Rightarrow \sigma_r = 0 \text{ et } A_{i_0+r} = 0.$$

Il suffit de montrer l'hypothèse H_r pour r pair. Montrons d'abord H_2. L'équation eq_{i_0+2} est $A_{i_0+2} + A_{i_0}\sigma_2 = 0$. Or

$$\begin{array}{lcccccccc} 2 & = & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ i_0 & = & 1 & 1 & 1 & \cdots & 1 & 0 & 0 \\ 2+i_0 & = & 1 & 0 & 0 & \cdots & 0 & 1 & 0 \end{array}$$

c'est à dire que $2 + i_0$ contient plus de deux"0" dans son écriture binaire, et un troisième zéro, donc est dans l'ensemble de définition du code. La borne BCH implique que $A_{i_0} \neq 0$, donc $\sigma_2 = 0$.

Supposons $H_{r'}$ vérifiée pour $r' < r$. Alors d'après le lemme III.5, l'équation eq_{i_0+r} est

$$A_{i_0+r} + A_{i_0}\sigma_r = 0.$$

Ce qui entraîne la nullité ou la non-nullité simultanée de σ_r et de A_{i_0+r}. Si $i_0 + r \in I(C)$, alors $\sigma_r = 0$, et $A_{i_0+r} = 0$, sinon, on cherche une deuxième équation qui permet de prouver que l'un ou l'autre est nul.

Pour cela deux cas sont à distinguer, suivant que r est supérieur ou inférieur à 2^{m-3}. Le cas $r = 2^{m-3}$ n'est pas à traiter car 2^{m-3} est dans J_{m-2} ($2^{m-3} = 2^{m-2} - 2^{m-3}$).
cas $r < 2^{m-3}$ **:** Nous considérons l'équation eq_{2i_0+3r}.

$$eq_{2i_0+3r} \ : \ A_{2i_0+3r} + \sum_{k=1}^{i_0-1} A_{2i_0+3r-k}\sigma_k + A_{i_0+3r}\sigma_{i_0} = 0.$$

Etudions le terme $A_{2i_0+3r-k}\sigma_k$. Si k est impair, $\sigma_k = 0$, sinon, écrivons $r - k = 2k'$ et

$$A_{2i_0+3r-k}\sigma_k = A^2_{i_0+r+k'}\sigma_k, \quad k \in [1, i_0[.$$

Si $k' > 0$, alors $k < r$ et donc $\sigma_k = 0$, d'après H_k (k ne peut être dans J_{m-2}, puisque $k < r < 2^{m-3}$).

Si $k' < 0$ et $r + k' \neq 0$, alors $r + k' < r$ et $A_{i_0+r+k'} = 0$ (si $r + k' < 0$, $i_0 + r + k' \in I$, et si $r + k' > 0$ alors, $r + k' \notin J_{m-2}$ et $H_{r+k'}$ s'applique).

Le cas $k' < 0$ et $r = -k'$ donne le terme $A^2_{i_0}\sigma_{3r}$. Ceci implique aussi $k = 3r$ et donc $r < (i_0 - 1)/3$. Donc si $r \geq (i_0 - 1)/3$ ce terme n'apparaît pas.

L'équation se réduit donc à

$$A_{2i_0+3r} + A^2_{i_0+r}\sigma_r + A^2_{i_0}\sigma_{3r} + A_{i_0+3r}\sigma_{i_0} = 0$$

si $r < (i_0 - 1)/3$, et à

$$A_{2i_0+3r} + A^2_{i_0+r}\sigma_r + A_{i_0+3r}\sigma_{i_0} = 0$$

si $r \geq (i_0 - 1)/3$.

Le lemme III.6 entraîne la nullité de A_{2i_0+3r} et A_{i_0+3r}. Donc l'équation eq_{2i_0+3r} se réduit à

$$A^2_{i_0+r}\sigma_r + A^2_{i_0}\sigma_{3r} = 0$$

si $r < (i_0 - 1)/3$, et à

$$A^2_{i_0+r}\sigma_r = 0$$

si $r \geq (i_0 - 1)/3$, ce qui permet de conclure dans ce cas là. Dans le cas $r < (i_0 - 1)/3$, on montre directement que A_{i_0+r} est nul. Considérons en effet i_1 le plus petit représentant des

classes cyclotomiques, autres que celle de i_0, qui ne sont pas dans I. C'est l'entier le plus petit qui ne contient pas deux zéros consécutifs dans sa décomposition en base 2, c'est-à-dire

$$i_1 = \begin{cases} (1,0,1,0,\ldots,0,1,0) &= (2^m-1)/3 & \text{si } m \text{ est pair,} \\ (1,0,1,0,1,\ldots,0,1) &= 1+2\cdot(2^{m-1}-1)/3 & \text{si } m \text{ est impair.} \end{cases}$$

Si $r < (i_0-1)/3$, alors

$$\begin{aligned} i_0 + r &< 2^{m-2} - 1 + \frac{2^{m-2}-1}{3} \\ &< i_1, \end{aligned}$$

que m soit pair ou impair. Donc si $r < (i_0-1)/3$, i_0+r est dans I et $A_{i_0+r} = 0$.
cas $r > 2^{m-3}$ **:** Nous considérons l'équation eq_{2i_0+2r}.

$$eq_{2i_0+2r} \ : \ A_{i_0+r}^2 + \sum_{k=1}^{i_0-1} A_{2i_0+2r-k}\sigma_k + A_{i_0+2r}\sigma_{i_0} = 0.$$

Nous écrivons $k = 2k'$ et le terme général est alors $A_{i_0+r-k'}^2\sigma_k = 0$. De plus $k < i_0$, donc $k' < i_0/2$ et $r - k' > 2^h - 1 - i_0/2 > 0$, ce qui permet d'appliquer $H_{r-k'}$ si $r - k' \notin J_{m-2}$. Sinon $r-k'$ est dans J_{m-2}, et il existe un entier $j \in [1, m-3]$ tel que $r - k' = 2^{m-2} - 2^j$. Nous avons alors deux possibilités :

1. $2k' \geq r$. Alors $r - k' \leq k' \leq (2^{m-2}-2)/2$. Alors $r-k'$ ne peut pas être pas dans J_{m-2}. Ce cas ne peut pas se produire.

2. $2k' < r$. Si k n'est pas dans J_{m-2}, alors σ_k est nul, par H_k. Sinon il existe $j' \in [1, m-3]$ tel que $k = 2^{m-2} - 2^j$. A ce moment là, $r = 2^{m-2} + 2^{m-3} - 2^j - 2^{j'-1}$. Si $j < m-3$, on vérifie que $r > 2^{m-2} - 1 = i_0$, ce qui est impossible. Si $j = m-3$, on obtient $r = 2^{m-2} - 2^{j'-1}$ c'est-à-dire $r \in J_{m-2}$.

En fin de compte, l'équation eq_{2i_0+2r} se réduit à

$$A_{i_0+r}^2 + A_{i_0+2r}\sigma_{i_0} = 0.$$

Le lemme III.7 nous montre que $\mathrm{cl}(i_0+2r) = \mathrm{cl}(i_0+\epsilon)$, avec $-i_0 < \epsilon < r$, et $\epsilon \notin J_{m-2}$. Si $\epsilon < 0$ alors $i_0 + \epsilon$ est dans I, sinon, on applique H_ϵ. Dans les deux cas $A_{i_0+2r} = 0$. Pour finir $A_{i_0+r}^2 = 0$ et A_{i_0+r} est nul, ce qui permet de conclure.

3.2.3 Conséquences

Il est connu que les mots de poids minimum des codes de Reed et Muller raccourci sont ceux dont le support est un sous-espace vectoriel privé de 0.

Théorème III.7 *Les mots de poids minimum du code BCH primitif au sens strict, de longueur* $2^m - 1$ *de distance construite* $2^{m-2} - 1$ *sont les mots de poids minimum du code de Reed et Muller raccourci d'ordre 2.*

Corollaire III.4 *Le groupe d'automorphisme du code BCH au sens strict, de longueur* $2^m - 1$ *de distance construite* $2^{m-2} - 1$ *est contenu dans le groupe d'automorphisme du code de Reed et Muller raccourci d'ordre 2, c'est-à-dire le groupe linéaire.*

Preuve : En effet, les mots de poids minimum du code de Reed et Muller raccourci engendre le Reed et Muller raccourci (théorème II.6, chapitre 2). Tout automorphisme du code BCH conserve l'ensemble de ses mots de plus petit poids, donc les mots de poids minimum du Reed et Muller raccourci. Comme ceux-ci engendrent le Reed et Muller raccourci, par linéarité, tout automorphisme du BCH laisse le code de Reed et Muller raccourci invariant et est donc un automorphisme du code de Reed et Muller raccourci.

□

3.3 Généralisations

3.4 Autres valeurs de h

3.4.1 Les codes BCH 3-correcteurs

La formule des moments de Pless permet d'obtenir la distribution des poids d'un code à partir de la distribution des poids de son dual.

Théorème III.8 ([WS86][p. 131]) *Soit C un $[n,k]$ code linéaire, et $A_0,\ldots,A_n$ sa distribution des poids, et $A'_1,\ldots,A'_n$ la distribution des poids du dual C' de C. Alors*

$$\sum_{i=0\ldots n-w}\binom{n-i}{w}A_i = 2^{k-w}\sum_{i=0\ldots w}\binom{n-i}{n-w}A'_i,\quad i=0,\ldots,n. \tag{III.6}$$

La distribution des poids du dual des BCH 3-correcteurs est connue. Elle a été donnée par Berlekamp [Ber68] dans le cas m pair, et par Kasami [Kas69] dans le cas m impair. En utilisant la formule III.6, il est possible de calculer alors le nombre de mots de poids minimum des BCH 3-correcteurs. Nous avons mené ce calcul, en utilisant le système de calcul formel MAPLE, notamment pour la factorisation du nombre de mots de poids minimum.

Cas m pair Berlekamp a obtenu le résultat dans le cas m pair, pour le code étendu, de longueur $N=2^m$. Le code étendu ne contient pas de mots de poids 7, et nous allons compter le nombre de mots de poids 8 de l'étendu. Le dual du BCH 3-correcteur admet les paramètres suivants :

$$[N=2^m, 3m+1, 2^{m-1}-2^{(m+2)/2}],$$

et la distribution des poids est

i	A_i
0,2^m	1
$2^{m-1}+2^{(m+2)/2}$	$N(N-1)(N-4)/960$
$2^{m-1}-2^{(m+2)/2}$	$N(N-1)(N-4)/960$
$2^{m-1}+2^{m/2}$	$7N^2(N-1)/48$
$2^{m-1}-2^{m/2}$	$7N^2(N-1)/48$
$2^{m-1}+2^{(m-2)/2}$	$2N(N-1)(3N+8)/15$
$2^{m-1}-2^{(m-2)/2}$	$2N(N-1)(3N+8)/15$
2^{m-1}	$(N-1)(29N^2-4N+64)/32.$

En utilisant la formule III.6, on obtient

$$A_8 = \frac{N^4}{5040} - \frac{13\,N^3}{5040} + \frac{N^2}{30} - \frac{107\,N}{1260} + \frac{17}{315},$$

le nombre d'espace affines est

$$\frac{N^4}{1344} - \frac{N^3}{192} + \frac{N^2}{96} - \frac{N}{168},$$

et la différence entre le nombre de mots de poids 8 et le nombre de sous-espaces affines est

$$\frac{N\,(N-1)\,(N-4)\,(N-8)\,(N-16)}{40320}.$$

Propriété III.3 *Le code BCH de longueur 2^m-1, m pair, de distance construite 7 admet des mots de poids minimum dont le support n'est pas un sous-espace vectoriel dès que $m \geq 6$.*

Cas m impair Les paramètres sont les suivants

$$[N = 2^m - 1, 3m, 2^{m-1} - 2^{(m+1)/2}]$$

et la distribution des poids est

i	A_i
2^{m-1}	$(2^m-1)(9\cdot 2^{2m-4} + 3\cdot 2^{m-3} + 1)$
$2^{m-1} + 2^{(m+1)/2}$	$\frac{1}{3}2^{(m-5)/2}(2^{(m-3)/2}-1)(2^m-1)(2^{m-1}-1)$
$2^{m-1} - 2^{(m+1)/2}$	$\frac{1}{3}2^{(m-5)/2}(2^{(m-3)/2}+1)(2^m-1)(2^{m-1}-1)$
$2^{m-1} + 2^{(m-1)/2}$	$\frac{1}{3}2^{(m-3)/2}(2^{(m-1)/2}-1)(2^m-1)(5\cdot 2^{m-1}+4)$
$2^{m-1} - 2^{(m-1)/2}$	$\frac{1}{3}2^{(m-3)/2}(2^{(m-1)/2}+1)(2^m-1)(5\cdot 2^{m-1}+4)$
0	1.

En utilisant la formule III.6, on obtient

$$A_7 = \frac{N^4}{5040} - \frac{13\,N^3}{5040} + \frac{N^2}{30} - \frac{107\,N}{1260} + \frac{17}{315}.$$

Et le nombre d'espaces vectoriels est

$$\frac{(N-1)\,(N-2)\,(N-4)}{168}.$$

La différence entre le nombre de mots de poids minimum et le nombre de sous-espaces vectoriels est

$$\frac{(N-1)\,(N-2)\,(N-8)\,(N-32)}{5040}.$$

Propriété III.4 *Le code BCH de longueur impaire m de distance construite 7 admet des mots de poids minimum dont le support n'est pas un sous-espace vectoriel dès que $m \geq 7$.*

Il est très surprenant que dans les deux cas, le nombre de mots de poids minimum dont le support n'est pas un sous-espace vectoriel se factorise ainsi. Ceci laisse penser que les supports des mots de poids minimum sont d'une configuration très particulière, mais nous n'avons rien pu prouver.

3.4.2 Cas non binaire

Nous avons mené quelques expériences dans le cas des codes BCH sur $\mathbb{F}_3$. Dans le cas du code $BCH(n = 80, \delta = 26)$, nous avons obtenu uniquement des mots dont le support est un sous-espace vectoriel de dimension 3, et de plus les coefficients de ces mots étaient tous soit 1, soit -1. Nous n'avons pu prouver ce résultat, en toute généralité.

Dans le cas du code $BCH(n = 80, \delta = 8)$, nous avons obtenu des mots de poids 8 dont le support n'est pas un sous-espace vectoriel. Il semble donc que le résultat annoncé dans le cas binaire ne se généralise pas.

Chapitre IV

Calcul de bases normales sur un corps fini

Certains systèmes cryptographiques à clé publique sont basés sur la difficulté du problème du logarithme discret. Certaines implantations utilisent ce problème dans le cadre du groupe multiplicatif d'un corps fini non premier [MI88]. La difficulté de la résolution du problème du logarithme discret ne doit pas être le seul atout de tel systèmes : l'exponentiation d'un élément doit pouvoir être calculée rapidement. La notion de base normale de F_{q^n} fournit une réponse à ce problème.

Définition IV.1 *Soit* $\mathbf{K} = \mathrm{F}_{q^n}$*, considéré comme* F_q*-espace vectoriel. Une* base normale *de* $\mathbf{K}$ *est une base de* $\mathbf{K}$ *de la forme*

$$[\gamma, \gamma^q, \ldots, \gamma^{q^{n-1}}], \qquad \gamma \in \mathbf{K}.$$

Un élément γ *tel que* $[\gamma, \gamma^q, \ldots, \gamma^{q^{n-1}}]$ *est une base de* $\mathbf{K}$ *est dit* normal.

Soit $[\gamma, \gamma^q, \ldots, \gamma^{q^{n-1}}]$ une base normale de $\mathbf{K}$, et soit $x \in \mathbf{K}$, $x = x_0\gamma + \cdots + x_{n-1}\gamma^{q^{n-1}}$, alors $x^q = x_{n-1}\gamma + \cdots + x_{n-2}\gamma^{q^{n-2}}$. C'est-à-dire que l'élévation à la puissance q consiste simplement en une permutation circulaire des coordonnées dans la base normale. Ceci conduit à de bons algorithmes d'exponentiation [ABMV93].

Théorème IV.1 ([LN83, WS86, BGM+93, Alb56]) *Tout corps fini possède une base normale.*

La recherche de bases normales de F_{q^n}, et la détermination d'algorithmes pour calculer celles-ci, est donc un problème important, puisque celles-ci s'avèrent d'un grand intérêt pratique.

Parmi les bases normales, on distingue les *bases normales optimales*, c'est-à-dire celles qui fournissent une table de multiplication optimale, en ce sens qu'elle est la plus creuse possible. Ce sujet est considéré dans le cas de la caractéristique 2 dans [ABV89] et dans [MOVW89, BGM91] dans le cas général, et s'avère beaucoup plus difficile. La caractérisation des bases normales optimales est déterminée dans [GHWL92]. Nous n'aborderons pas le problème de la détermination de bases normales optimales.

Le point de vue abordé ici nous a conduit à considérer d'autres problèmes d'algèbre linéaire, que le calcul d'une base normale. Dans ce chapitre, nous présenterons dans la première partie nos résultats de complexité. La deuxième partie est une introduction à la forme

de Hessenberg à décalage d'une matrice, dont nous ferons grand usage dans les parties suivantes. Nous avons classé nos algorithmes par méthode : la troisième partie introduit des algorithmes directs, par opposition à des algorithmes récursifs, présenté dans la quatrième partie. Dans la cinquième partie, nous abordons le problème du calcul de la *forme rationnelle canonique* d'une matrice, et conclurons avec le problème du calcul d'une base normale en sixième partie.

1 Introduction

1.1 Algorithmes existants

Nous présenterons un algorithme déterministe de calcul de base normale sur $\mathbf{F}_{q^n}$. Cet algorithme ne construit pas nécessairement une base normale optimale. L'algorithme que nous présentons repose sur des techniques d'algèbre linéaire, et sa complexité est de $O(n^3 \log q)$ $\mathbf{F}_q$-opérations élémentaires. Nous rappelons les résultats de complexité déjà connus.

Algorithmes déterministes La meilleure complexité connue, en nombre d'opérations sur les bits, est $O((n^2 + \log q)(n \log q)^2)$ pour calculer une base normale de $\mathbf{F}_{q^n}$, cette complexité étant obtenue par l'algorithme de Bach, Driscoll et Shallit, présenté dans [BGM+93], ou par l'algorithme de H. W. Lenstra [Len91]. Ces algorithmes reposent essentiellement sur des manipulations de polynômes.

Algorithmes probabilistes J. Von zur Gathen et Marc Giesbrecht [vZGG90] établissent une estimation de la densité κ des éléments normaux dans $\mathbf{F}_{q^m}$. Il apparaît qu'il y en a beaucoup : $\kappa \geq 1/34$ si $n \leq q^4$, et $\kappa > (16 \log_q(n))^{-1}$ si $n \geq q^4$. Par essais successifs, en engendrant aléatoirement des éléments de $\mathbf{F}_{q^n}$, le temps moyen pour obtenir un élément normal est le même (à une constante près et à un facteur $\log n$ près) que le temps nécessaire à vérifier qu'un élément est normal.

J. Von zur Gathen et Marc Giesbrecht montrent alors qu'il est possible de calculer un élément normal en $O^\sim(n^2 \log q)$ opérations arithmétiques sur $\mathbf{F}_q$, en utilisant des méthodes rapides de multiplications rapides de polynômes [CK91], et en $O(n^3 \log q)$ en utilisant la multiplication naïve des polynômes. La notation $O^\sim$ est définie ainsi

$$g = O^\sim(h) \Leftrightarrow \exists k,\ g = O(h \log(h)^k).$$

Dans [BGM+93], un autre algorithme probabiliste est présenté, de complexité $O((n + \log q)(n \log q)^2)$ opérations sur les bits.

1.2 Un point de vue d'algèbre linéaire

Notre algorithme est essentiellement basé sur des considérations d'algèbre linéaire, et nous considérerons qu'un élément normal est un *vecteur cyclique* pour l'isomorphisme de Frobenius.

Définition IV.2 *Soit $A \in M_n(\mathbf{k})$ et $v \in \mathbf{k}^n$, le* polynôme minimal de A relativement à v *est le polynôme unitaire de plus petit degré $\pi_v(X)$ tel que $\pi_v(A)v = 0$.*

Notons que $\pi_v(X) \mid \pi(X)$, où $\pi(X)$ est le polynôme minimal de A.

Théorème IV.2 ([Gan77, Ch. VII§3 th. 2]) *Pour tout $A \in M_n(\mathbf{k})$, il existe un vecteur $v \in \mathbf{k}^n$ tel que $\pi_v(X) = \pi(X)$, où $\pi(X)$ est le polynôme minimal de A.*

Définition IV.3 *Soit A une matrice de $M_n(\mathbf{k})$. Un* vecteur cyclique *est un vecteur v de $\mathbf{k}^n$ tel que $\pi_v(X) = \pi(X)$, où $\pi(X)$ est le polynôme minimal de A.*

Le théorème IV.1 est alors une conséquence du théorème IV.2. En effet, le polynôme minimal de l'isomorphisme de Frobenius de $\mathbf{F}_{q^n} : x \mapsto x^q$ est $X^n - 1$. Le théorème IV.2 nous assure de l'existence d'un vecteur dont le polynôme minimal est $X^n - 1$, ce qui revient à dire que cet élément est normal.

Nous produirons donc un algorithme de recherche de vecteur cyclique d'un endomorphisme A, de complexité $O(n^3)$. Pour obtenir une base normale, on applique cet algorithme dans le cas de l'isomorphisme de Frobenius, dont la matrice peut être calculée en $O(n^3 \log q)$.

Ceci nous a amené à nous intéresser à d'autres problèmes d'algèbre linéaire.

1.3 Autres problèmes

Nous considérerons la thèse de P. Ozello, "Calculs exacts des formes de Jordan et de Frobenius d'une matrice" [Oze87], comme une référence en ce qui concerne l'algorithmique d'algèbre linéaire. Les programmeurs de la librairie Maple ont d'ailleurs choisi d'implanter les algorithmes décrits dans la thèse d'Ozello. Nous comparerons souvent les complexités de nos algorithmes à celles obtenues dans la thèse d'Ozello.

Dans la section 2, nous rappelons comment calculer le polynôme caractéristique d'une matrice en $O(n^3)$. Ensuite nous introduisons les matrices de Hessenberg à décalage. Nous définissons le paramètre m_A d'une matrice A, qui est le nombre de polynômes irréductibles, comptés avec multiplicités, du polynôme caractéristique de A. En utilisant les travaux de Richard Stong [Sto88], qui évalue le paramètre m_A en moyenne pour les matrices *inversibles* de taille n sur $\mathbf{F}_q$, nous conduirons une courte étude théorique qui nous mènera à une évaluation en moyenne de m_A pour toutes les matrices de $M_n(\mathbf{F}_q)$: m_A est en moyenne d'ordre $O(\log n)$ pour une matrice de taille n sur $\mathbf{F}_q$.

Nous aborderons le problème du calcul du polynôme minimal d'une matrice, et de la recherche d'un vecteur cyclique d'une matrice dont le polynôme caractéristique est sans facteurs multiples. Dans les sections 3 et 4, un algorithme sera présenté pour trouver une solution à chacun de ces problèmes. La section 3 présente des algorithmes *directs* pour résoudre ces problèmes, en opposition à des algorithmes respectant une stratégie *diviser pour régner*, qui sont présentés dans la section 4.

La complexité de ces algorithmes *directs* est donnée en fonction de n, la taille de la matrice, et de m_A, le nombre de facteurs irréductibles, comptés avec multiplicité, du polynôme caractéristique de A. Essentiellement, les deux algorithmes présentent la même complexité, qui est de $O(n^3 + m_A^2 n^2)$ opérations élémentaires sur $\mathbf{F}_q$.

Toutefois, lorsque le paramètre m_A de la matrice A est grand, la complexité de ces algorithmes devient mauvaise, de l'ordre de $O(n^4)$. Dans la section 4, nous présentons alors des algorithmes respectant une stratégie "diviser pour régner", qui présentent une meilleure complexité, lorsque m_A est grand. Pour calculer le polynôme minimal d'une matrice A, l'algorithme proposé est récursif et prend comme donnée supplémentaire la factorisation du

polynôme caractéristique de A, ce qui limite la portée de cet algorithme aux corps finis. Un algorithme pour calculer un vecteur cyclique d'une matrice dont le polynôme caractéristique est sans facteurs multiples est aussi formulé avec une stratégie diviser pour régner, et ceci conduit à un algorithme de calcul d'une base normale de F_{q^n}, lorsque n est premier à q, avec la complexité $O(n^3 \log q)$.

Enfin, dans la section 5, nous présentons un algorithme de calcul de la forme rationnelle canonique (et même d'une forme plus forte, en réalité) d'une matrice A, qui utilise à la fois des techniques diviser pour régner, et des méthodes directes. Cette forme résout tous les problèmes précédents, mais est plus coûteuse à obtenir, et de plus l'algorithme proposé prend comme donnée supplémentaire la factorisation du polynôme caractéristique de A. La connaissance apportée par cette étude nous permet alors de calculer une base normale pour F_{q^n}, n quelconque, sur la donnée d'une matrice représentant l'opérateur de Frobenius.

Les complexités de ces différents algorithmes sont présentés dans la table IV.1.

Commentaires sur la table IV.1 Pour tous nos calculs, nous supposons que nous sommes dans un corps *effectif*, c'est-à-dire un corps dans lequel nous sommes capables de calculer sommes, produits et inverses, et de tester l'égalité de deux éléments. En particulier, une *présentation* de F_q est supposée donnée (par exemple par une représentation des éléments comme éléments d'un groupe cyclique, en utilisant une table de logarithmes de Zech). La représentation *polynomiale* des éléments de F_q permet d'effectuer les opérations élémentaires à un coût $O(\log(q)^2)$.

Nous n'estimerons pas le coût de ces opérations élémentaires, et nos complexités seront données en nombre d'additions et de multiplications. Par exemple, nous dirons que le pgcd de deux polynômes dans $\mathbf{k}[X]$ de degré inférieur à n peut être calculé en $O(n^2)$ et cela signifie $O(n^2)$ opérations élémentaires dans $\mathbf{k}$. Ainsi notre mesure de complexité ne mesure pas le nombre d'opérations sur les bits.

Cette remarque est importante, puisque nos algorithmes peuvent s'appliquer sur tout corps $\mathbf{k}$ et en particulier sur Q, mais nous ne donnons pas de mesure de la taille des nombres rationnels intermédiaires. Sur les corps finis nous pouvons supposer que toutes les opérations élémentaires s'effectuent en temps constant.

Dans la table IV.1, nous présentons aussi la complexité moyenne des divers algorithmes introduits, dans le cas d'un corps fini. Nous entendons par là que les matrices données en entrée à l'algorithme sont uniformément distribuées, selon la mesure de comptage. On peut voir alors le nombre d'opérations effectuées par l'algorithme comme une variable aléatoire, à valeurs dans N, sur l'espace $M_n(\mathbf{k})$ muni de la mesure de comptage. Nous appelons *complexité moyenne* l'espérance de cette variable aléatoire.

2 Matrices de Hessenberg

2.1 Calcul du polynôme caractéristique

D'une manière surprenante le polynôme caractéristique d'une matrice A de taille n peut être calculé en $O(n^3)$ opérations élémentaires. La méthode est d'obtenir une matrice semblable à A pour laquelle le polynôme caractéristique est calculé facilement. Cette forme s'appelle forme de Hessenberg, présentée dans l'ouvrage de Wilkinson [Wil92] (cité dans [Knu81]).

Problème	Données	Complexité [moyenne]	partie
Polynôme minimal	A	$O(n^3 + n^2 m_A^2)$ $[O(n^3)]$	3.1
	Factorisation de $C(X)$	$O(n^{3.5})$ $[O(n^{3.5})]$	4.1
	Factorisation de $C(X)$	$O(n^3 + n^2 m_A^2)$ $[O(n^3)]$	4.1
Vecteur cyclique	A, $C(X)$ SQFR	$O(n^3 + n^2 m_A^2)$ $[O(n^3)]$	3.2
	A, $C(X)$ SQFR	$O(n^3)$ $[O(n^3)]$	4.3
Forme rationnelle canonique	Factorisation de $C(X)$	$O(m_A n^3)$ $[O(n^3 \log n)]$	5

m_A est le nombre de facteurs irréductibles, comptés avec multiplicité, de $C(X)$, le polynôme caractéristique de A. Les complexités moyennes valent dans le cas d'un corps fini.

Table IV.1 : Les complexités des différents algorithmes proposés.

2.1.1 La forme de Hessenberg d'une matrice et son calcul

Définition IV.4 *Une matrice $H \in M_n(\mathbf{k})$ est dite* matrice de Hessenberg *si elle a la forme suivante :*

$$\begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,n} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,n} \\ 0 & 0 & h_{4,3} & \cdots & h_{4,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & h_{n,n-1} & h_{n,n} \end{pmatrix},$$

c'est-à-dire si $H = h_{i,j}$ telle que $j < i - 1 \Rightarrow h_{i,j} = 0$.

Toute matrice est semblable à une matrice de Hessenberg. Toutefois, il ne s'agit pas d'un invariant des classes de similitude : deux matrices semblables peuvent être chacunes semblables à deux matrices de Hessenberg différentes.

Théorème IV.3 *Pour toute matrice A de $M_n(\mathbf{k})$, il existe une matrice de Hessenberg H et une matrice inversible P telles que $H = PAP^{-1}$ (Toute matrice est semblable à une matrice de Hessenberg). Les matrices H et P peuvent être calculées en $O(n^3)$ opérations élémentaires.*

Preuve : Nous prouvons le théorème en donnant l'algorithme dans la table IV.2, page 98.

Il s'agit d'une réduction par élimination de Gauss, les pivots étant choisis sur la sous-diagonale principale. A la différence de la méthode de Gauss, si le terme sur la première sous-diagonale est nul, on peut chercher un pivot en-dessous sur la même colonne, pour le placer sur la sous-diagonale principale. Ceci est fait par une permutation de lignes, et par la permutation de colonnes conjuguée, pour rester dans la même classe de similitude. Cette opération respecte les colonnes déjà traitées.

Le nombre maximum d'opérations élémentaires que nécessite cet algorithme est borné (grossièrement) par $2n^3$.

□

Données ; une matrice A de taille $n \times n$.

Sortie : H, matrice de Hessenberg, P inversible, $H = PAP^{-1}$.

H:=A; P:=I_n; i:=1
tant que i < n **faire** {*traiter chaque colonne sauf la dernière*}
    chercher le premier élément non nul dans la colonne i
    à partir du $i+1$-ième. Si un tel élément existe, soit j la position de cet élément
    si *aucun tel élément n'a été trouvé*
    alors i:=i+1 {*cette colonne reste échangée*}
    sinon
        pour H:
            échanger les lignes $i+1$ et j
            échanger les colonnes $i+1$ et j
        pour P: échanger les lignes $i+1$ et j
    pivot := 1/H[i+1,i]
    pour l **de** i+1 **à** n **faire**
        c:= pivot * H[l,i]
        pour H, $L_l \leftarrow L_l - c \times L_{i+1}$
        pour H, $C_{i+1} \leftarrow c \times C_l + C_{i+1}$
        pour P, $L_l \leftarrow L_l - c \times L_{i+1}$
    i:=i+1
retourner(H, P)

Table IV.2 : Algorithme de calcul d'une forme Hessenberg

Cette méthode peut admettre différentes stratégies quant au choix du pivot, dès lors que l'on s'intéresse au problème de la taille des coefficients. Le résultat peut être obtenu par une suite de transformations orthogonales dans le cas de nombres réels (méthode de Givens, méthode de Householder [Wil92]). Une remarque d'Ozello permet aussi de diminuer la taille des nombres intermédiaires [Oze87] dans le cas des nombres rationnels.

2.1.2 Calcul du polynôme caractéristique d'une matrice de Hessenberg

Le procédé suivant est montré dans [Wil92]. Soit $p_k(X)$ le polynôme caractéristique de la sous-matrice extraite des k premières lignes et des k premières colonnes d'une matrice de Hessenberg H. Calculer le polynôme caractéristique de A revient à calculer $p_n(X)$. On observe que les polynômes $p_k(X)$ vérifient la relation de récurrence suivante

$$\begin{aligned} p_k(X) &= (X - h_{k,k})p_{k-1}(X) - h_{k,k-1}(\\ &\quad (X - h_{k-1,k})p_{k-2}(X) - h_{k-1,k-2}(\\ &\quad (X - h_{k-2,k})p_{k-3}(X) - h_{k-2,k-3}(\\ &\quad \dots \\ &\quad (X - h_{3,k})p_2(X) - h_{3,2}(\\ &\quad h_{2,k}p_1(X) - h_{2,1}h_{1,k}))) \dots) \end{aligned}$$

Le calcul de $p_k(X)$ connaissant $p_{k-1}(X), p_{k-2}(X), \dots, p_1(X)$ s'opère à un coût $O(k^2)$. Le coût total pour obtenir $p_n(X)$ est $O(n^3)$.

2.2 Matrices de Hessenberg à décalage

2.2.1 La forme Hessenberg à décalage

Nous introduisons la terminologie de matrice de Hessenberg à décalage pour certaines matrices de Hessenberg. Cette matrice a déjà été considérée par P. Ozello dans [Oze87], comme une forme intermédiaire pour obtenir la forme rationnelle canonique d'une matrice. En particulier, on peut calculer le polynôme caractéristique d'une matrice de Hessenberg à décalage plus facilement, et sous forme partiellement factorisée.

Définition IV.5 *Une matrice H de $M_n(\mathbf{k})$ est dite de* Hessenberg à décalage *si elle a la forme*

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

c'est-à-dire que H est une matrice de Hessenberg telle que

$$(h_{i+1,i} \neq 0) \Rightarrow (h_{i+1,i} = 1,\ \forall j \leq i\ h_{j,i} = 0).$$

Le paramètre *m d'une matrice de Hessenberg à décalage est défini comme étant le nombre de zéros sur la sous-diagonale, plus un.*

Note 1 *Le paramètre m est le nombre de blocs diagonaux, chaque bloc étant une matrice compagnon, c'est-à-dire une matrice de la forme*

$$\begin{pmatrix} & & & & & c_0 \\ 1 & & & & & c_1 \\ & 1 & & & & c_2 \\ & & \ddots & & & c_i \\ & & & 1 & & c_{n-2} \\ & & & & 1 & c_{n-1} \end{pmatrix}.$$

Le polynôme minimal d'une telle matrice est égal à son polynôme caractéristique, qui est

$$X^n - c_{n-1}X^{n-1} - c_{n-2}X^{n-1} \cdots - c_1X^1 - c_0.$$

Si le paramètre m vaut 1, la matrice de Hessenberg à décalage est elle-même une matrice compagnon. L'autre cas extrême est $m = n$, où la matrice est triangulaire.

Nous notons $f_i(X)$ le polynôme minimal de la matrice compagnon située sur le i-ième bloc. Le polynôme caractéristique d'une matrice de Hessenberg à décalage est $f_1(X) \cdots f_m(X)$.

Nous avons le théorème suivant, équivalent au théorème IV.3.

Théorème IV.4 *Pour toute matrice A de $M_n(\mathbf{k})$, il existe une matrice de Hessenberg à décalage H et une matrice inversible P telle que $H = PAP^{-1}$ (Toute matrice est semblable à une matrice de Hessenberg à décalage). Les matrices H et P peuvent être calculées en $O(n^3)$ opérations élémentaires.*

Preuve : A nouveau nous prouvons le théorème en donnant un algorithme de calcul, voir table IV.3, page 101.

□

2.2.2 Matrices compagnons

Puisque les blocs diagonaux d'une matrice de Hessenberg à décalage sont des matrices compagnons, nous allons détailler des méthodes simples et efficaces pour résoudre certains problèmes d'algèbre linéaire où intervient une matrice compagnon. Ceci nous mènera ensuite à des algorithmes de faible complexité.

Dorénavant, étant donnée une matrice compagnon C de taille $n \times n$ de polynôme minimal $\pi(X)$ de degré n, et un vecteur $v = (v_0, \ldots v_{n-1})$, nous identifierons le vecteur v au polynôme

$$v(X) = v_0 + v_1X + v_2X^2 + \cdots + v_{n-1}X^{n-1}. \tag{IV.1}$$

Nous étudions d'abord le calcul de Cv pour tout vecteur v de $\mathbf{k}^n$. On observe que $Cv = Xv(X) \bmod \pi(X)$. Ceci signifie que le calcul de Cv consiste seulement en un décalage sur le vecteur v, modulo $\pi(X)$.

Données ; une matrice A de taille $n \times n$.

Sortie : H, matrice de Hessenberg à décalage, P inversible, $H = PAP^{-1}$.

H:=A; P:=I_n; i:=1
tant que i < n **faire** {*traiter chaque colonne sauf la dernière.*}
    chercher le premier élément non nul dans la colonne i
    à partir du i + 1-ième. Si un tel élément existe, soit j la position de cet élément
    si *aucun tel élément n'a été trouvé*
    alors i:=i+1 {*cette colonne reste inchangée*}
    sinon
        pour H:
            échanger les lignes $i+1$ et j
            échanger les colonnes $i+1$ et j
        pour P: échanger les lignes $i+1$ et j
    c := 1/H[i+1,i] {*pivot*}
    pour H
        $L_{i+1} \leftarrow L_{i+1} \times c$; $C_{i+1} \leftarrow C_{i+1}/c$ $\{h[i+1,i]$ *est égal à* $1\}$
    pour P $L_{i+1} \leftarrow L_{i+1} \times c$
    pour l **de** 1 **à** n **telque** $l \neq i+1$ **faire**
        h:= H[l,i]
        pour H: $L_l \leftarrow L_l - h \times L_{i+1}$; $C_{i+1} \leftarrow h \times C_l + C_{i+1}$
        pour P: $L_l \leftarrow L_l - h \times L_{i+1}$
    i:=i+1
retourner(H,P)

Table IV.3 : Algorithme de calcul d'une forme Hessenberg à décalage

Lemme IV.1 *Soit C une matrice compagnon, soit $v \in \mathbf{k}^n$, Cv peut être calculé en $2n$ opérations élémentaires.*

En particulier :

Lemme IV.2 *Soit C une matrice compagnon de polynôme minimal $\pi(X)$, soit $v \in \mathbf{k}^n$, et soit $P(X) \in \mathbf{k}[X]$ de degré inférieur à n, alors $P(C)v$ peut être calculé à un coût $O(n^2)$.*

Preuve : Calculer $P(C)v$ revient à calculer $P(X)v(X) \bmod \pi(X)$.

□

La solution de certains systèmes linéaires s'obtient de la manière décrite dans la preuve du lemme suivant. L'important est qu'un problème d'algèbre linéaire est résolu en calculant dans l'algèbre de polynômes $\mathbf{k}[X]/\pi(x)$.

Lemme IV.3 *Soit C une matrice compagnon de polynôme minimal $\pi(X)$, soit $v \in \mathbf{k}^n$, si $P(X) \in \mathbf{k}[X]$ est premier à $\pi(X)$, alors le système suivant en u*

$$P(C)u = v \tag{IV.2}$$

admet une unique solution qui s'obtient avec un coût $O(n^2)$.

Preuve : Puisque $P(X)$ est premier à $\pi(X)$, il existe $Q(X)$ tel que $P(X)Q(X) = 1 \pmod{\pi(X)}$. La solution u est donnée par $u = Q(C)v$. Calculer le polynôme $Q(X)$ se fait en $O(n^2)$ par l'algorithme d'Euclide étendu, et calculer $Q(C)v$ se fait en $O(n^2)$ par le lemme IV.2.

□

De la même manière, on peut calculer le polynôme minimal d'une matrice compagnon relativement à un vecteur.

Lemme IV.4 *Soit C une matrice compagnon de polynôme minimal $\pi(X)$, et soit $v \in \mathbf{k}^n$. Le polynôme minimal $\pi_v(X)$ de C relativement à v est donné par*

$$\pi_v(X) = \frac{\pi(X)}{\operatorname{pgcd}(\pi(X), v(X))},$$

et peut être calculé en $O(n^2)$ opérations élémentaires.

Preuve : Soit $\pi_v(X)$ le polynôme de degré minimal tel que $\pi_v(C)v = 0$. Il s'agit de trouver $\pi_x(X)$ de degré minimal tel que

$$\pi_x(X)v(X) = 0 \pmod{\pi(X)}.$$

Un solution est donnée par $\pi_u(X) = \pi(X)/\operatorname{pgcd}(\pi(X), v(X))$. Soit $\pi_w(X)$ une solution de $\pi_x(X)v(X) = 0 \bmod \pi(X)$, alors

$$\pi(X) \mid \pi_w(X)v(X),$$

et

$$\frac{\pi(X)}{\operatorname{pgcd}(\pi(X), v(X))} \mid \pi_w(X).$$

Ainsi $\pi_u(X)$ est le polynôme de degré minimal que nous recherchons.

□

2.2.3 Opérations avec une matrice de Hessenberg à décalage

Une matrice de Hessenberg à décalage a l'intérêt d'être creuse : elle présente au plus $2m$ termes non nuls par lignes. Dans cette sous-section, nous présentons quelques méthodes de calcul avec des matrices de Hessenberg à décalage.

Le premier lemme est immédiat, au vu du caractère creux d'une matrice de Hessenberg à décalage.

Lemme IV.5 *Soit $H \in M_n(\mathbf{k})$ une matrice de Hessenberg à décalage, de paramètre m, et soit M une matrice de taille $n \times n'$. Alors le produit HM peut être calculé en $O(mnn')$.*

Définition IV.6 *Soit $H \in M_n(\mathbf{k})$ une matrice de Hessenberg à décalage de paramètre m. Une matrice A est dite* polycyclique *pour H si ses colonnes sont, dans l'ordre,*

$$\left(v_1, Hv_1, \dots H^{n_1-1}v_1, v_2, Hv_2, \dots, H^{n_2-1}v_2, \dots, v_m, Hv_m, \dots, H^{n_m-1}v_m\right)$$

où $n_1, n_2, \dots n_m$ sont les tailles des blocs diagonaux de H, et $v_1, v_2, \dots, v_m$ sont des vecteurs de $\mathbf{k}^n$.

Proposition IV.1 *Soit $H \in M_n(\mathbf{k})$ une matrice de Hessenberg à décalage, et soit A, B deux matrices polycycliques pour H. Soit $\alpha, \beta \in \mathbf{k}$ alors $\alpha A + \beta B$, I_n, H, HA sont polycycliques pour H.*

En d'autres termes, l'ensemble des matrices polycycliques de H est un sous module du $\mathbf{k}[X]$-module induit par H sur $\mathbf{k}^n$.

Proposition IV.2 *Soit $H \in M_n(\mathbf{k})$ une matrice de Hessenberg à décalage de paramètre m. Alors le produit HA peut être calculé en $O(mn^2)$ pour un matrice quelconque $A \in M_n(\mathbf{k})$ et en $O(m^2n)$ si A est polycyclique pour H.*

Preuve : Si A est polycyclique, le produit HA est calculé en modifiant A comme suit. Les colonnes de A sont

$$\left(v_1, Hv_1, \dots H^{n_1-1}v_1, v_2, Hv_2, \dots, H^{n_2-1}v_2, \dots, v_m, Hv_m, \dots, H^{n_m-1}v_m\right),$$

et HA sera

$$\left(Hv_1, \dots H^{n_1}v_1, Hv_2, \dots, H^{n_2}v_2, \dots, Hv_m, \dots, H^{n_m}v_m\right),$$

de sorte qu'il suffit de décaler les colonnes de A vers la gauche, et de calculer seulement les produits $HH^{n_1-1}v_1, \dots, HH^{n_m-1}v_m$ soit m multiplications matrice-vecteur, chacune de coût $O(mn)$.

□

Corollaire IV.1 *Soit $H \in M_n(\mathbf{k})$ une matrice de Hessenberg à décalage, de paramètre m. Un polynôme $p(X) \in \mathbf{k}[X]$ de degré au plus t peut être évalué en H en $O(tm^2n)$.*

Preuve : Nous appliquons la règle de Horner pour évaluer $p(H) = p_tH^t + p_{t-1}H^{t-1} + \cdots + p_1H + p_0I$. Nous calculons $h_1 = p_tH + p_{t-1}I$, $h_2 = Hh_1 + p_{t-2}I$,...,$h_t = Hh_{t-1} + t_0I$. D'après la proposition IV.2, h_i est calculé sur la donnée de h_{i-1} à un coût $O(m^2n)$, d'où un coût total de $O(tm^2n)$ pour $p(H)$.

□

Ceci est à comparer avec le coût de l'évaluation d'un polynôme en une matrice A quelconque, qui est de tn^3, en utilisant la méthode de Horner et la multiplication naïve des matrices, de coût $O(n^3)$.

2.3 Bases à décalage et matrices de Hessenberg à décalage

La forme de Hessenberg à décalage sera pour nous l'outil algorithmique de base pour la plupart de nos algorithmes. Avant de présenter ceux-ci, nous allons montrer quelques propriétés des matrices de Hessenberg à décalage. Nous emploierons le terme opérateur et nous utiliserons la même notation pour l'opérateur A et pour la matrice A représentant l'opérateur A dans la base canonique de $\mathbf{k}^n$.

Les notions introduites dans cette section seront surtout employées en 2.4, pour obtenir une estimation du paramètre m d'une matrice à décalage.

2.3.1 Bases à décalages

Définition IV.7 *Soit $A \in M_n(\mathbf{k})$. Une* base à décalage *de A est une base de $\mathbf{k}^n$ de la forme*

$$\left[v_1, Av_1, \ldots, A^{n_1-1}v_1, v_2, Av_2, \ldots, A^{n_2-1}v_2, \ldots, v_m, Av_m, \ldots, A^{n_m-1}v_m\right]. \tag{IV.3}$$

La notion de bases à décalage s'entend pour une famille *ordonnée* de vecteurs.

Propriété IV.1 *Une matrice qui représente un opérateur A dans une de ses bases à décalages est une matrice de Hessenberg à décalage de A. Réciproquement une matrice de Hessenberg à décalage définit une base à décalage de A.*

Nous rappelons la forme suivante de matrice

Définition IV.8 *Une matrice $A \in M_n(\mathbf{k})$ est dite* rationnelle canonique *si elle de la forme suivante*

$$F = \begin{bmatrix} C_{p_1} & 0 & \cdots & 0 \\ 0 & C_{p_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{p_t} \end{bmatrix}, \tag{IV.4}$$

où C_{p_i} est la matrice compagnon de polynôme minimal p_i, et $p_1 \mid p_2 \mid \cdots \mid p_t$. Les polynômes p_i, $i = 1, \ldots, t$ sont les diviseurs élémentaires *de A.*

Théorème IV.5 ([Gan77]) *Toute matrice est équivalente à une matrice rationnelle canonique. Les diviseurs élémentaires sont invariants par classe de similitude, et caractérisent chaque classe.*

En particulier la forme rationnelle canonique d'une matrice est une matrice de Hessenberg à décalage dans une base à décalage particulière.

Etant donné un opérateur A, une base à décalage de A peut être obtenue de la manière suivante. Soit v_1 un vecteur de $\mathbf{k}^n$, on calcule les vecteurs $A^i v_1$, $i = 0, \ldots, n_1 - 1$ où n_1 est la plus petite valeur i telle que $A^i v_1$ est linéairement dépendant des $A^j v_1$, $j < i$. Ensuite v_2 est choisi indépendant des vecteurs précédents et de même on calcule $A^i v_2$, $i = 0, \ldots, n_2 - 1$, où n_2 est la plus petite valeur i telle que $A^i v_2$ est combinaison linéaire des $A^j v_2$, $j < i$ et des $A^j v_1$, $j < n_1$. Le processus s'arrête lorsque $n_1 + \cdots + n_m = n$. Une base à décalage est obtenue par l'algorithme décrit en IV.3, mais cette construction reflète la structure d'une base à décalage.

Plus précisément, à chaque base à décalage correspond une suite croissante de sous-espaces invariants $V_1, \ldots, V_m$. Chaque V_i est un sous-module du $\mathbf{k}[X]$-module induit par A

sur $\mathbf{k}^n$, et donc V_i/V_{i-1} est un $\mathbf{k}[X]$-module. Chaque module V_i/V_{i-1} est engendré par la classe $\overline{v}_i$ de v_i dans V_i/V_{i-1}, et $f_i(X)$ est le polynôme minimal de $\overline{v}_i$.

Plusieurs bases à décalage peuvent représenter l'opérateur A par la même matrice de Hessenberg à décalage. Il existe ainsi une partition des bases à décalage de A par rapport aux formes de Hessenberg à décalage auxquelles elles correspondent. La propriété suivante est claire d'après les définitions.

Propriété IV.2 *Soit A une matrice de $M_n(\mathbf{k})$, et soient deux bases à décalage B_1 et B_2:*

$$B_1 = \left[v_1, Av_1, \ldots, A^{n_1-1}v_1, v_2, Av_2, \ldots, A^{n_2-1}v_2, \ldots v_m, Av_m, \ldots, A^{n_m-1}v_m\right]$$

et

$$B_2 = \left[v'_1, Av'_1, \ldots, A^{n_1-1}v'_1, v'_2, Av'_2, \ldots, A^{n_2-1}v'_2, \ldots v'_m, Av'_m, \ldots, A^{n_m-1}v_m\right]$$

telles que

$$V_1 \subset V_2 \subset \ldots \subset V_m,$$

et

$$\left[v_1, v'_1 \in V_1 = AV_1; v_2, v'_2 \in V_2 \setminus V_1, V_2 = AV_2; v_m, v'_m \in V_m \setminus V_{m-1}, V_m = AV_m\right].$$

Si $A^{n_i}v_i$ est combinaison linéaire de $v_1, \ldots, v_{n_i-1}$ avec les mêmes coefficients que $A^{n_i}v'_i$, $i = 1, \ldots, m$ respectivement, alors B_1 et B_2 correspondent à la même forme de Hessenberg à décalage.

Définition IV.9 *Soit A une matrice de $M_n(\mathbf{k})$, et H une forme de Hessenberg à décalage* de *A. Nous dirons que B est une base à décalage* pour *H si l'opérateur A est représenté par la matrice H dans la base B.*

Nous établissons la distinction entre une base à décalage quelconque et une base à décalage représentant une matrice de Hessenberg à décalage donnée au moyen des prépositions *de* et *pour* (respectivement). Ceci est un peu faible, mais évite une terminologie plus lourde. Nous invitons le lecteur à y prêter attention, notamment dans les énoncés des théorèmes IV.3 et IV.6, et dans les preuves des corollaires IV.2 et IV.3.

2.3.2 Forme rationnelle canonique développée

Définition IV.10 *La* forme rationnelle canonique développée *d'un opérateur A de $M_n(\mathbf{k})$ est la matrice suivante, équivalente à A :*

$$D = \begin{bmatrix} F_{B_1,B_1} & 0 & \cdots & 0 \\ 0 & F_{B_2,B_2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & F_{B_d,B_d} \end{bmatrix}$$

où chaque matrice F_{B_i,B_i} est une matrice rationnelle canonique

$$\begin{bmatrix} C_{p_i^{s_{i,1}}} & 0 & \cdots & 0 \\ 0 & C_{p_i^{s_{i,2}}} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & C_{p_i^{s_{i,m_i}}} \end{bmatrix},$$

où $C_{p_i}^{s_{i,m_j}}$ *est la matrice compagnon de polynôme minimal* $p_i^{s_{i,m_j}}$, $s_{i,1} \leq s_{i,2} \leq \ldots \leq s_{i,m_i}$ *et les polynômes* p_i *sont premiers entre eux deux à deux.*

Au vu de la forme rationnelle canonique développée, $\mathbf{k}^n$ est la somme directe

$$\bigoplus_{i=1}^{d} \bigoplus_{j=1}^{m_i} V_{p_i}^{s_{i,j}},$$

où $V_{p_i}^{s_{i,j}}$ est le sous-espace vectoriel invariant pour A, tel que le polynôme minimal de la restriction de A à $V_{p_i}^{s_{i,j}}$ est $p_i^{s_{i,j}}$. Le $\mathbf{k}[X]$-module induit par A sur $\mathbf{k}^n$ est isomorphe au produit d'anneaux :

$$R = R_{1,1} \times \cdots \times R_{1,m_1} \times R_{2,1} \times \cdots \times R_{2,m_2} \times \cdots \times R_{d,1} \times \cdots \times R_{d,m_d} \tag{IV.5}$$

considérés comme $\mathbf{k}[X]$-modules avec $R_{i,j} = \mathbf{k}[X]/p_i^{s_{i,j}}$.

Pour tout vecteur u, nous notons $u_{|R_{i,j}}$ la composante de u dans l'anneau $R_{i,j}$, et nous considérons indifféremment $u_{|R_{i,j}}$ comme un vecteur ou comme un polynôme de degré inférieur à $s_{i,j} \deg(p_i)$.

Une base à décalage pour la forme rationnelle canonique développée est définie par une séquence de vecteurs

$$v'_{1,1}, v'_{1,2}, \ldots, v'_{1,m_1}, v'_{2,1}, \ldots, v'_{2,m_2}, \ldots, v'_{d,1}, \ldots, v'_{d,m_d}$$

telle que pour tout couple (i,j) le polynôme minimal de $v'_{i,j}$ est $p_i^{s_{i,j}}$.

Lemme IV.6 *Soit* u *un vecteur de* $\mathbf{k}^n$ *tel que* $p_i^{s_{i,j}} u = 0$. *Alors les composantes de* u *dans la décomposition IV.5 vérifient*

$$u_{|R_{k,l}} = 0 \text{ si } k \neq i.$$

Preuve : Supposons qu'il existe k, l, $k \neq i$ tels que $u_{|R_{k,l}} \neq 0$. Alors $p_i^{s_{i,j}} u_{|R_{k,l}}$ est non nul, puisque $u_{|R_{k,l}}$ est non nul et que $p_i^{s_{i,j}}$ est une unité de $R_{k,l} = \mathbf{k}[X]/p_k^{s_{k,l}}$ $(pgcd(p_i, p_k) = 1)$. □

Lemme IV.7 *Soit* u *un vecteur de* $\mathbf{k}^n$, *de polynôme minimal* $p_i^{s_{i,j}}$. *Alors les composantes de* u *dans* $R_{i,l}$ *vérifient*

- $l < j$; $u_{|R_{i,l}}$ *est un élément quelconque de* $R_{i,l}$,
- $l = j$; $u_{|R_{i,l}}$, *est un polynôme premier à* p_i,
- $l > j$; $u_{|R_{i,l}}$ *est un multiple de* $p_i^{s_{i,u} - s_{i,j}}$.

Preuve : Puisque le polynôme minimal de u est $p_i^{s_{i,j}}$, nous avons que $p_i^{s_{i,j}} v = 0$ pour tout vecteur v de $R_{i,l}$, dés que $l < j$, puisque $p_i^{s_{i,l}}$, qui divise $p_i^{s_{i,j}}$, est le polynôme minimal de A restreint à $R_{i,l}$. Ceci prouve le cas $l < j$.

Dans le cas $l = j$, nous avons qu'un vecteur est cyclique pour une matrice compagnon s'il est premier au polynôme minimal de cette matrice (lemme IV.4).

Dans le cas $l > j$, on doit avoir

$$p_i^{s_{i,j}} u = 0,$$

ce qui entraîne dans $R_{i,j}$ que

$$p_i^{s_{i,j}} u_{|R_{i,l}} = 0 \bmod p_i^{s_{i,l}},$$

et que $p_i^{s_{i,l}-s_{i,j}}$ divise $u_{|R_{i,l}}$.

□

Propriété IV.3 *Soit D une matrice rationnelle canonique. Toutes les bases à décalage pour D ont la forme*

$$[v'_{1,1}, Dv'_{1,1}, \ldots, D^{n_{1,1}-1}v'_{1,1}, v'_{1,2}, Dv'_{1,2}, \ldots, D^{n_{1,2}-1}v'_{1,2}, \ldots, v'_{d,m_d}, Dv'_{d,m_d}, \ldots, D^{n_{d,m_d}-1}v'_{d,m_d}]$$

où $n_{i,j} = s_{i,j} \deg p_i$ et chaque $v'_{i,j}$ dans $R_{i,l}$ admet $p_i^{s_{i,j}}$ pour polynôme minimal.

Preuve : C'est exactement la propriété IV.2, dans le cas d'une matrice rationnelle canonique développée.

□

2.3.3 Application au centralisateur d'une matrice

Définition IV.11 *Le* centralisateur $\mathcal{Z}(A)$ *d'une matrice A de $M_n(\mathbf{k})$ est le groupe des matrices inversibles de $GL_n(\mathbf{k})$ qui commutent avec A.*

Théorème IV.6 *Soit A un opérateur, et soit H une matrice Hessenberg à décalage semblable à A. Il y a correspondance bijective entre les base à décalage pour H et les matrices de $\mathcal{Z}(A)$, le groupe des matrices inversibles commutant avec A.*

Preuve : Soit B_1 et B_2 les matrices de changement de base pour deux bases à décalage pour H, vérifiant les conditions de la propriété IV.2, ${B_1}^{-1}AB_1 = {B_2}^{-1}AB_2$, puisque ces deux bases produisent la même matrice de Hessenberg à décalage, d'après la propriété IV.2. Alors la matrice $G = B_2{B_1}^{-1}$ commute avec A:

$$\begin{aligned} GA &= B_2{B_1}^{-1}A = B_2{B_1}^{-1}AB_1{B_1}^{-1} \\ &= B_2H_1{B_1}^{-1} \\ &= B_2{B_2}^{-1}AB_2{B_1}^{-1} \\ &= AB_2{B_1}^{-1} = AG. \end{aligned}$$

Soit maintenant une matrice inversible Q commutant avec A. Soient $v_1, \ldots, v_m$ définissant la base à décalage produite par B_1 comme dans la définition IV.7. Alors la matrice QB_1 est une matrice de base à décalage pour H. En effet soient $u_1, \ldots, u_m$ les vecteurs $u_1 = Qv_1, \ldots, u_mQv_m$, alors on a $A^kQv_i = QA^kv_i$, $i = 1, \ldots, m$. La famille

$$U = (u_1, Au_1, \ldots, A^{n_1-1}u_1, u_2, Au_2, \ldots, A^{n_2-2}u_2, \ldots, u_m, Au_m, \ldots, A^{n_m-1}u_m)$$

vérifie les mêmes conditions de dépendance linéaire que

$$V = (v_1, Av_1, \ldots, A^{n_1-1}v_1, v_2, Av_2, \ldots, A^{n_2-2}v_2, \ldots, v_m, Av_m, \ldots, A^{n_m-1}v_m),$$

puisque Q est inversible. D'après la propriété IV.2, U_1 produit le même forme de Hessenberg à décalage H_1. Si B_2 est la matrice de la base U_1, alors $B_2 = QB_1$, c'est-à-dire $Q = B_1{B_2}^{-1}$.

□

Définition IV.12 *La* somme directe *de deux matrices A_1 et A_2 est la matrice*

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}.$$

Corollaire IV.2 *Le centralisateur de la somme directe de deux matrices A_1 et A_2, dont les polynômes minimaux sont premiers entre eux, est le produit direct des centralisateurs de A_1 et de A_2.*

Preuve : Soit F_1 la forme rationnelle canonique développée de A_1 et F_2 la forme rationnelle canonique développée de A_2. Alors la matrice

$$F = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix}$$

est une matrice rationnelle canonique développée pour A. D'après le théorème IV.6, toute base à décalage pour F produit un élément du centralisateur de A. Puisque les polynômes minimaux de F_1 et F_2 sont premiers entre eux, le lemme IV.6 montre que les bases à décalage pour F sont des sommes directes des bases à décalage pour F_1 et pour F_2. Les bases à décalage pour F_1 (respectivement F_2) sont en correspondance avec les commutateurs de A_1 (respectivement A_2), d'où le résultat.

□

Corollaire IV.3 *Soit $\mathbf{k} = \mathrm{F}_q$, et soit A un opérateur dont la forme rationnelle canonique développée F est donnée comme dans la définition IV.10. Alors le cardinal du centralisateur de A est*

$$\prod_{i=1}^{d} \prod_{j=1}^{m_i} q^{\deg(p_i)\left(\sum_{w=1}^{j-1} s_{i,w} + (m_i - j)s_{i,j}\right)} \phi(p_i^{s_{i,j}}). \tag{IV.6}$$

où $\phi(g)$ est le nombre de polynômes de degré inférieur à $\deg(g)$ premiers à g.

Preuve : Il suffit de dénombrer les bases à décalage pour F. Chacune de ces bases à décalage est une famille

$$v'_{1,1}, v'_{1,2}, \ldots, v'_{1,m_1}, v'_{2,1}, \ldots, v'_{2,m_2}, \ldots, v'_{d,1} \ldots v'_{d,m_d}$$

telle que pour tout (i, j), le polynôme minimal de $v'_{i,j}$ est $p_i^{s_{i,j}}$.

Dans la formule (IV.6), le produit externe correspond au lemme IV.6. Chaque produit interne énumère, pour chaque $p_i^{s_{i,j}}$, le nombre de vecteurs tels que $p_i^{s_{i,j}} v = 0$. La somme

$$\sum_{w=1}^{j-1} s_{i,w}$$

vaut pour les anneaux $R_{i,l}$, $l < j$, pour lesquels tout vecteur v vérifie $p_i^{s_{i,j}} v = 0$. Le terme $(m_i - j)s_{i,j}$ est un résultat du fait que le nombre de polynômes multiples de $p_i^{s_{i,l} - s_{i,j}}$ dans $\mathbf{k}[X]/p_i^{s_{i,l}}$ est $q^{\deg(p_i)s_{i,j}}$.

Enfin, $\phi(p_i^{s_{i,j}}) = q^{s_{i,j} \deg(p_i)}(1 - q^{-\deg(p_i)})$ est le nombre de polynômes premiers à $p_i^{s_{i,j}}$, c'est-à-dire le nombre d'unités de $R_{i,j}$.

□

2.4 Estimation du paramètre m

Définition IV.13 *Soit A une matrice de $M_n(\mathbf{k})$, nous définissons m_A comme étant le nombre de facteurs du polynôme caractéristique de A, comptés avec multiplicité.*

Soit $V_1, \ldots, V_m$ une chaîne croissante d'espace invariants pour A, alors $m \leq m_A$. En particulier le paramètre m de toute forme de Hessenberg à décalage est inférieur ou égal à m_A.

Richard Stong, dans [Sto88] prouve le résultat suivant, sur la valeur moyenne de m_A pour les matrices inversibles.

Théorème IV.7 ([Sto88, Proposition 12]) *Soit $\mathbf{k} = \mathrm{F}_q$, et soit X_n la variable aléatoire donnée par le nombre de facteurs irréductibles, comptés avec multiplicité, du polynôme caractéristique des matrices inversibles de $GL_n(\mathbf{k})$. Alors EX_n est asymptotiquement équivalent à $\log n$, avec un écart moyen de $\log n$.*

Nous allons prouver le théorème :

Théorème IV.8 *Soit $\mathbf{k} = \mathrm{F}_q$, et soit Y_n la variable aléatoire donnée par le nombre de facteurs irréductibles des polynômes caractéristiques de matrices de $M_n(\mathbf{k})$, et soit EY_n l'espérance de Y_n. Alors, pour tout $\epsilon > 0$, il existe n_0 tel que $EY_n \leq 2(1+\epsilon)\log n$ pour $n \geq n_0$.*

C'est-à-dire $\limsup (EY_n / \log n) \leq 2$.

Nous allons d'abord prouver deux lemmes, qui interviendront dans la preuve.

Soit A une matrice de $M_n(\mathbf{k})$, et nous considérons sa forme rationnelle canonique développée comme suit :

$$\begin{bmatrix} N & 0 \\ 0 & S \end{bmatrix}$$

où N est une matrice rationnelle canonique développée de polynôme caractéristique X^{n_1} et S est une matrice rationnelle canonique développée inversible de taille $n_2 = n_1 - 1$.

Lemme IV.8 *L'espérance EZ_n du nombre de facteurs, comptés avec multiplicité, du polynôme caractéristique de S, pour les matrices de $M_n(\mathbf{k})$ vérifie :*

$$\forall \epsilon > 0, \exists n_0, \forall n \geq n_0 \Rightarrow EZ_n \leq (1+\epsilon)\log n.$$

Preuve : Soit S_{n_1} l'ensemble des matrices rationnelles canoniques de taille n_1 de polynôme caractéristique X^{n_1}, et S_{n_2} l'ensemble des matrices rationnelles canoniques inversibles de taille n_2. Nous notons z_{N,n_1} (respectivement z_{S,n_2}) le cardinal du centralisateur de $N \in S_{n_1}$ (respectivement $S \in S_{n_2}$).

Etant données S et N, alors, par le corollaire IV.2, le nombre de matrices qui admettent la forme rationnelle canonique

$$\begin{bmatrix} N & 0 \\ 0 & S \end{bmatrix} \tag{IV.7}$$

est

$$\frac{|GL(n,q)|}{z_{N,n_1} z_{S,n_2}}.$$

Le nombre de matrices qui admettent X^{n_1} dans la factorisation de leur polynôme caractéristique, et la matrice S dans leur forme rationnelle canonique comme dans (IV.7) est

$$\begin{aligned}\sum_{N\in S_{n_1}} \frac{|GL(n,q)|}{z_{N,n_1} z_{S,n_2}} &= \frac{|GL(n,q)|}{z_{S,n_2}} \sum_{s\in S_{n_1}} \frac{1}{z_{N,n_1}} \\ &= \frac{1}{z_{S,n_2}} \chi(n_1,n,q)\end{aligned}$$

où

$$\chi(n_1,n,q) = |GL(n,q)| \sum_{N\in S_{n_1}} \frac{1}{z_{N,n_1}}.$$

Soit maintenant $C_{n_2,k}$ l'ensemble des polynômes $C(X)$, $C(0)\neq 0$, de degré n_2, qui se décomposent en k facteurs irréductibles, et $S_{n_2,k}$ l'ensemble des matrices rationnelles canoniques de taille n_2 dont le polynôme caractéristique appartient à $C_{n_2,k}$. Le nombre de matrices dont le polynôme caractéristique est $X^{n_1}C(X)$, pour $C(X)$ dans $C_{n_2,k}$, est

$$\chi(n_1,n,q) \sum_{S\in S_{n_2,k}} \frac{1}{z_{S,n_2}}.$$

Soit θ la variable aléatoire définie par la taille de la partie non singulière d'une matrice, et soit η la variable aléatoire définie par le nombre de facteurs de la partie non singulière. La probabilité conditionnelle $P_n\{\eta = k \mid \theta = n_2\}$ que $C(X)$ appartienne à $C_{n_2,k}$, pour une matrice dont le polynôme caractéristique est $X^{n_1}C(X)$, est

$$\begin{aligned}\frac{\chi(n_1,n,q) \sum_{S\in S_{n_2,k}} \frac{1}{z_{S,n_2}}}{\chi(n_1,n,q) \sum_{S\in S_{n_2}} \frac{1}{z_{S,n_2}}} &= \frac{\sum_{S\in S_{n_2,k}} \frac{|GL(n_2,q)|}{z_{S,n_2}}}{\sum_{S\in S_{n_2}} \frac{|GL(n_2,q)|}{z_{S,n_2}}} \\ &= P_{n_2}\{\eta = k\},\end{aligned}$$

où $P_n\{\eta = k\}$ est la probabilité qu'une matrice inversible dans $GL(n,\mathbf{k})$ ait un polynôme caractéristique qui se factorise en k facteurs.

Nous pouvons alors conclure : le nombre moyen des facteurs différents de X du polynôme caractéristique d'une matrice dans $M_n(q)$ est donné par

$$\begin{aligned}\sum_{k=1}^{n} k \sum_{n_2=1}^{n} P\{\theta = n_2\} P_{n_2}\{\eta = k\} &= \sum_{n_2=1}^{n} P\{\theta = n_2\} \sum_{k=1}^{n} k P_{n_2}\{\eta = k\} \qquad \text{(IV.8)} \\ &= \sum_{n_2=1}^{n} P\{\theta = n_2\} EX_{n_2}. \qquad \text{(IV.9)}\end{aligned}$$

Ceci est une moyenne des EX_{n_2}, $n_2 = 1 \ldots n$, dont on montre qu'elle vérifie bien l'assertion de l'énoncé, sachant que $EX_{n_2} \approx \log n_2$ quand n_2 devient grand.

□

Le lemme suivant correspond à la partie singulière :

Lemme IV.9 *Soit Z_n la variable aléatoire définie par la multiplicité du facteur X dans le polynôme caractéristique des matrices de $M_n(\mathbf{k})$. Alors l'espérance EZ_n de Z_n est bornée asymptotiquement par* $\log n$.

Preuve : Considérons la translation $M \mapsto M + I_n$. Le facteur X^{n_1} du polynôme caractéristique d'une matrice devient $(X-1)^{n_1}$ dans la factorisation du polynôme caractéristique de $M' = M + I_n$. Considérons la forme rationnelle canonique de M'

$$\begin{bmatrix} N & 0 \\ 0 & S \end{bmatrix}$$

où N est nilpotente et S est inversible. Alors $(X-1)^{n_1}$ est la plus grande puissance de $X-1$ qui est un facteur du polynôme caractéristique $C(X)$ de S. Par le lemme IV.8, le nombre moyen des facteurs de $C(X)$ est borné asymptotiquement $\log n$.

□

Le théorème IV.8 résulte des lemmes IV.8 et IV.9.

3 Algorithmes directs

3.1 Calcul direct du polynôme minimal

3.1.1 Introduction

Nous donnons d'abord un algorithme direct de calcul du polynôme minimal d'une matrice A, dont on connaît une forme de Hessenberg à décalage. Ultérieurement nous présenterons un algorithme "diviser pour régner", de structure plus compliquée, mais dont la complexité peut être meilleure dans certains cas. L'algorithme présent utilise comme donnée la matrice A dont le polynôme minimal est à calculer, alors que l'algorithme récursif nécessite comme donnée supplémentaire la factorisation du polynôme caractéristique de A.

Nous introduisons un certain nombre de notations liées à une matrice de Hessenberg à décalage H d'une matrice A. D'abord H est décrite par blocs de la manière suivante.

$$H = \begin{pmatrix} H_{B_1,B_1} & H_{B_1,B_2} & \cdots & H_{B_1,B_m} \\ 0 & H_{B_2,B_2} & \cdots & H_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & H_{B_m,B_m} \end{pmatrix}.$$

Notation 7 *Nous notons B_k l'ensemble d'indices numérotant le k-ième bloc. Nous notons $B_{\geq k}$ l'ensemble d'indices $B_k \cup B_{k+1} \ldots \cup B_m$. Pour toute matrice $A \in M_n(\mathbf{k})$ nous notons A_{B_i,B_j} la matrice obtenue de A par extraction des lignes dans B_i et des colonnes dans B_j. Enfin, nous notons $A_{B_{\geq k}}$ la matrice carrée des lignes et des colonnes de A obtenue à partir du k-ième bloc. Dans le cas d'une matrice de Hessenberg à décalage, nous notons $f_i(X)$ le polynôme minimal de la matrice compagnon H_{B_i,B_i}.*

3.1.2 Une chaîne d'idéaux croissants liés à H

Propriété IV.4 *Soit I_k l'ensemble des polynômes $g(X)$ tel que*

$$g(H)_{B_i,B_i} = 0,\ i = 1,\ldots,m,\ \text{et } g(H)_{B_i,B_j} = 0, \qquad i < j,\ i,j = k,\ldots,m. \tag{IV.10}$$

Alors I_k est un idéal de $\mathbf{k}[X]$. On a la chaîne suivante d'inclusions

$$I_1 \subseteq I_2 \cdots \subseteq I_m. \tag{IV.11}$$

Preuve : Considérons d'abord le cas où $k = 1$. Alors

$$g(H)_{B_i,B_j} = 0,\ i \leq j,\ i,j = k,\ldots,m \Leftrightarrow g(H) = 0$$

et l'idéal $I_1 = (p_1(X))$ est l'idéal annulant la matrice H, c'est-à-dire l'idéal définissant le polynôme minimal $p_1(X)$ de H.

Soit maintenant I_k l'ensemble des polynômes tels que (IV.10) est vérifiée. Alors I_k est un idéal de polynômes, et $\mathbf{k}[X]$ étant principal, nous avons $I_k = (p_k)$. De plus $I_k \subseteq I_{k+1}$ et donc $p_{k+1}(X) \mid p_k(X)$.

□

Nous notons $\phi_k(X)$ le polynôme $p_k(X)/p_{k+1}(X)$. Nous montrons comment interpréter $\phi_k(X)$ sur la matrice H.

Soit $g(X) \in I_{k+1}$, alors, en se restreignant aux blocs d'indices de ligne dans B_k et de colonne dans B_j, $k \leq j \leq m$, on voit que la relation suivante est vérifiée :

$$(Hg(H))_{B_k,B_j} = H_{B_k,B_k}(g(H))_{B_k,B_j}$$

Soit maintenant $p(X)$ un polynôme de la forme $q(X)p_{k+1}(X)$, qui est la forme générale des polynômes de I_{k+1}. Nous avons :

$$p(H)_{B_k,B_j} = q(H)_{B_k,B_k}(p_{k+1}(H))_{B_k,B_j} = q(H_{B_k,B_k})(p_{k+1}(H))_{B_k,B_j}, \qquad k \leq j \leq n$$

Ainsi $p(H)_{B_k,B_j} = 0,\ k \leq j \leq n$ si et seulement si $q(H_{B_k,B_k})(p_{k+1}(H))_{B_k,B_j} = 0,\ k \leq j \leq n$, c'est-à-dire si et seulement si $q(H_{B_k,B_k})$ annule l'espace engendré par les colonnes des matrices $(p_{k+1}(H))_{B_k,B_j}, j = k,\ldots,m$.

Donc le polynôme $\phi_k(X)$ est le polynôme minimal de la restriction de H_{B_k,B_k} au sous-espace engendré par les colonnes de $(p_{k+1}(H))_{B_k,B_j}, j = k,\ldots,m$.

Remarquons que, puisque $f_k(X)$ est le polynôme minimal de H_{B_k,B_k} sur $\mathbf{k}^{B_k}$, nous avons que $\phi_k(X) \mid f_k(X)$.

3.1.3 L'algorithme pour le calcul du polynôme minimal de H

La méthode est de calculer successivement $p_m(X)$, $p_{m-1}(X),\ldots,p_1(X)$, pas à pas.

Première étape

On calcule le polynôme $p_m(X)$. Puisque tous les blocs diagonaux de $p_m(H)$ sont nuls, $p_m(X)$ est le ppcm des $f_i(X)$, $i = 1,\ldots,m$.

Étape itérative : calcul de $p_k(X)$ sur la donnée de $p_{k+1}(X)$

Supposons que $p_{k+1}(X)$ a déjà été calculé. Nous avons

$$p_{k+1}(H) = \begin{pmatrix} 0 & p_{k+1}(H)_{B_1,B_2} & \cdots & \cdots & p_{k+1}(H)_{B_1,B_m} \\ & \ddots & & & \\ & & 0 & p_{k+1}(H)_{B_k,B_{k+1}} & p_{k+1}(H)_{B_k,B_m} \\ & & 0 & 0 & 0 \\ & & 0 & 0 & 0 \end{pmatrix} \qquad \text{(IV.12)}$$

Nous devons trouver le polynôme $\phi_k(X)$, qui est le polynôme minimal H_{B_k,B_k} restreint au sous-espace engendré par les colonnes des matrices $p_{k+1}(H)_{B_k,B_j},\ k < j \leq m$. Nous aurons alors $p_k(X) = \phi_k(X)p_{k+1}(X)$. Le calcul de $\phi_k(X)$ se fait de proche en proche comme suit.

Soit $a^1 = (a^1_1, a^1_2, \ldots, a^1_{m_k})$ la première colonne non nulle des colonnes des matrices $p_k(H)_{B_k,B_j}$, $k < j \leq m$. On calcule le polynôme minimal $\phi_{k,a^1}(X)$ de H_{B_k,B_k} relativement à a^1. Ainsi $\phi_{k,a^1}(X)$ est un facteur de $\phi_k(X)$ et la matrice $H_{k,a^1} = \phi_{k,a^1}(H)p_{k+1}(H)$ est alors calculée. Ensuite le même procédé est appliqué à nouveau sur la première colonne non nulle a^2 des colonnes de H_{k,a^1}, pour obtenir un nouveau facteur $\phi_{k,a^2}(X)$ de $\phi_k(X)$. On calcule de nouveau $H_{k,a^2} = \phi_{k,a^2}(H)H_{k,a^1}$, et la première colonne non nulle de H_{k,a^2} est considérée. Le procédé s'arrête quand toutes les colonnes ont été annulées. Alors $\phi_k(X) = \phi_{k,a^1}(X)\phi_{k,a^2}(X)\cdots\phi_{k,a^l}(X)$ où a^l est la dernière colonne non nulle rencontrée.

3.1.4 Complexité

La clé est que calculer le polynôme minimal de H_{B_k,B_k} restreint à une colonne est peu coûteux en vertu du lemme IV.4, qui indique comment calculer le polynôme minimal d'une matrice compagnon relativement à un vecteur : cela se réduit à un calcul de pgcd de polynômes. Chacun de ces calculs a un coût de $O(n_k^2)$, et le nombre de ces calculs est au plus m_A. ceci donne une borne de $O(m_A n^2)$ pour tous les calculs de pgcd.

Les calculs les plus coûteux sont les évaluations des $H_k, H_{k,a^1}, H_{k,a^2}$. Chacune de ces opérations a un coût de $O(m^2n^2)$ d'après le corollaire IV.1. Il y a au plus m_A tel calculs.

Théorème IV.9 *Étant donnée une matrice de Hessenberg à décalage, son polynôme minimal peut être calculé en $O(m_A m^2 n^2)$ opérations élémentaires.*

Ceci est à comparer avec la procédure de P. Ozello qui est de $O(n^3 m)$.

Corollaire IV.4 *Le polynôme minimal d'une matrice A peut être obtenu en $O(n^3 + m_A^3 n^2)$ opérations élémentaires, où le terme n^3 provient du calcul d'une forme de Hessenberg à décalage de A.*

Preuve : On calcule d'abord H une forme de Hessenberg à décalage de la matrice A. Le paramètre m de H est inférieur à m_A, et le théorème IV.9 permet de conclure.

□

Remarque IV.1 *Notons que la complexité du pire est $O(n^5)$, ce qui est mauvais. Toutefois, quand m est grand, la stratégie suivante peut être employée pour évaluer les matrices $p_m(H)$, $p_{m-1}p_m(H)$, $p_{m-2}p_{m-1}p_m(H)\ldots$*

On utilise une sorte de règle de Horner "à l'envers". Soit $d_1, \ldots, d_m$ les degrés des polynômes $p_1, \ldots, p_m$. D'abord la matrice $p_m(H)$ est calculée à un coût de $d_m m^2 n$ en vertu du corollaire IV.1. Soit C_{k+1} la matrice $p_{k+1}p_k\cdots p_m(H)$, qui est une matrice polycyclique pour H, et soit $p_k(X) = X^{d_k} + a_{d_k-1}X^{d_k-1} + \cdots + a_1X + a_0$. On peut calculer $p_k(H)C_{k+1}$ comme suit :

$$(H^{d_k} + a_{d_k-1}H^{d_k-1} + \cdots + a_1H + a_0)C_{k+1} = (H^{d_k-1} + a_{d_k-1}H^{d_k-2} + \cdots + a_1)HC_{k+1} + a_0C_{k+1}$$

Le produit HC_{k+1} est obtenu à un coût de $O(m^2n)$ par la proposition IV.2, le produit a_0C_{k+1} à un coût de $O(n^2)$, et la somme de ces deux matrices à un coût de $O(n^2)$. Ainsi calculer $p_k(H)C_{k+1}$ a un coût de $O(d_k(m^2n + n^2))$, et le coût total est alors $O((d_1 + \cdots + d_m)(m^2n+n^2)) = O(m^2n^2+n^3)$. Cette méthode est meilleure lorsque m est grand, et conduit à une complexité du pire de $O(n^4)$.

3.2 Construction d'un vecteur cyclique

3.2.1 Introduction

Nous introduisons maintenant une méthode pour trouver un vecteur cyclique d'une matrice A. Bien qu'un vecteur cyclique soit obtenu lors du calcul de la forme rationnelle canonique d'une matrice sur laquelle n'est formulée aucune hypothèse, cette section est dédiée à l'objectif du vecteur cyclique, et on suppose de plus que le polynôme caractéristique de la matrice A est sans facteurs multiples. En effet un algorithme de bonne complexité existe sous ces hypothèses.

L'hypothèse que le polynôme caractéristique de la matrice A est sans facteurs multiples implique que le polynôme minimal de A est égal à son polynôme caractéristique. Aussi, les polynômes minimaux $f_k(X)$ des blocs compagnon diagonaux de toute forme de Hessenberg à décalage de A sont premiers deux à deux.

3.2.2 Un lemme technique

Notation 8 *Étant donné un vecteur v de $\mathbf{k}^n$, le vecteur de taille n_I, qui est la projection de v sur $\mathbf{k}^{B_I}$, est noté v_{B_I}. Nous notons $v^\star_{B_I}$ l'unique vecteur de $\mathbf{k}^n$ tel que sa projection sur $\mathbf{k}^{B_I}$ soit v_{B_I} et tel que sa projection sur $\mathbf{k}^{B_J}$ est 0, où J est l'ensemble complémentaire de I dans $[1,n]$: $(v^\star_{B_I})_{B_J} = 0$.*

Le lemme suivant établit l'étape de récurrence dans la recherche d'un vecteur cyclique. Nous établissons ce lemme pour une matrice A quelconque, bien que nous exploiterons la forme de Hessenberg à décalage dans notre algorithme.

Lemme IV.10 *Soit A une matrice en bloc de la forme*

$$\begin{pmatrix} A_{B_1,B_1} & A_{B_1,B_2} \\ 0 & A_{B_2,B_2} \end{pmatrix}$$

et soient v_{B_1}, v_{B_2} deux vecteurs cycliques pour A_{B_1,B_1} et A_{B_2,B_2} respectivement, de polynômes minimaux $f_1(X)$ et $f_2(X)$ respectivement. Si $f_1(X)$ et $f_2(X)$ sont premiers entre eux alors les équations en $u_{B_2} \in \mathbf{k}^{B_2}$ et $u_{B_1} \in \mathbf{k}^{B_1}$

$$v_{B_2} = u_{B_2} \tag{IV.13}$$
$$v_{B_1} = f_2(A_{B_1,B_1})u_{B_1} + (f_2(A)u^\star_{B_2})_{B_1}. \tag{IV.14}$$

admettent une solution unique $u = (u_{B_1}, u_{B_2})$, qui est un vecteur cyclique pour A.

Preuve : Montrons l'existence d'une solution unique. Il est clair que u_{B_2} est unique puisque $u_{B_2} = v_{B_2}$. Soit $h_2(X)$ tel que $h_2(X)f_2(X) = 1 \bmod f_1(X)$. Alors

$$u_{B_1} = h_2(A_{B_1,B_1})(f_2(A)u^\star_{B_2})_{B_1} - v_{B_1})$$

est solution de IV.14. Soit u'_{B_1} tel que

$$v_{B_1} = f_2(A_{B_1,B_1})u'_{B_1} + (f_2(A)u^\star_{B_2})_{B_1},$$

alors

$$u'_{B_1} = h_2(A_{B_1,B_1})(f_2(A)u^\star_{B_2})_{B_1} - v_{B_1}),$$

ce qui prouve l'unicité de u_{B_1}.

Puisque $f_1(X)f_2(X)$ est le polynôme minimal de A, nous avons à prouver que $f_1(X)f_2(X)$ est le polynôme minimal de la restriction de A à u. Supposons que $p(A)u = 0$ pour un polynôme $p(X)$ de degré minimal. Alors $p(X)$ est un diviseur de $f_1(X)f_2(X)$ et $p(X) = p_1(X)p_2(X)$ avec la condition que $p_1(X) \mid f_1(X),\ p_2(X) \mid f_2(X)$ et $\mathrm{pgcd}(p1(X), p2(X)) = 1$. Il s'ensuit

$$\begin{aligned} p(A)u = 0 &\Rightarrow (p(A)u)_{B_2} = 0 && \text{(IV.15)}\\ &\Rightarrow p(A_{B_2,B_2})u_{B_2} = 0 && \text{(IV.16)}\\ &\Rightarrow p_1(A_{B_2,B_2})p_2(A_{B_2,B_2})u_{B_2} = 0. && \text{(IV.17)} \end{aligned}$$

Puisque $\mathrm{pgcd}(p_1(X), f_2(X)) = 1$, il existe $h_1(X)$ tel que $p_1(X)h_1(X) = 1 \pmod{f_2(X)}$, c'est-à-dire

$$h_1(A_{B_2,B_2})p_1(A_{B_2,B_2}) = I.$$

En appliquant $h_1(A_{B_2,B_2})$ des deux cotés de IV.17 nous obtenons $p_2(A_{B_2,B_2})u_{B_2} = 0$, et ceci implique que $f_2(X) \mid p_2(X)$ puisque v_{B_2} est un vecteur cyclique pour A_{B_2,B_2}. Donc $p_2(X) = f_2(X)$.

Pour le premier bloc de coordonnées, nous avons les implications

$$\begin{aligned} p(A)u = 0 &\Rightarrow (p(A)u)_{B_1} = 0 && \text{(IV.18)}\\ &\Rightarrow (p(A)u^\star_{B_1})_{B_1} + (p(A)u^\star_{B_2})_{B_1} = 0 && \text{(IV.19)}\\ &\Rightarrow p_1(A_{B_1,B_1})(f_2(A_{B_1,B_1})u_{B_1} + (f_2(A)u^\star_{B_2})_{B_1}) = 0 && \text{(IV.20)} \end{aligned}$$

Par hypothèse, $f_2(A_{B_1,B_1})u_{B_1}+(f_2(A)u^\star_{B_2})_{B_1} = v_{B_1}$ est cyclique pour A_{B_1,B_1}. Alors d'après $(IV.20)$, $f_1(X) \mid p_1(X)$. Donc $p_1(X) = f_1(X)$ ce qui termine la preuve.

□

Remarque IV.2 *Résoudre les équations (IV.13) et (IV.14) nécessite trois principaux calculs. La matrice $f_2(A_{B_1,B_1})$ doit être calculée, ensuite $w_{B_1} = (f_2(A)u^\star_{B_2})_{B_1}$ est un vecteur à calculer, et enfin le système $f_2(A_{B_1,B_1})u_{B_1} = v_{B_1} - w_{B_1}$ est à résoudre.*

Nous observons que ces calculs peuvent être conduits à un faible coût :

Lemme IV.11 *Une solution u_{B_1}, u_{B_2} aux équations (IV.13) et (IV.14) peut être calculée en $O(n^3)$ opérations élémentaires.*

Preuve : Nous procédons comme suit. D'abord $w_{B_1} = (f_2(A)u^\star_{B_2})_{B_1}$ est calculé. Ceci coûte $O(n^3)$. Ensuite l'équation (IV.14) est résolue en trouvant un inverse $h_2(X)$ de $f_2(X)$ mod $f_1(X)$, la solution u_{B_1} étant donnée par

$$u_{B_1} = h_2(A_{B_1,B_1})(v_{B_1} - w_{B_1})$$

qui est évalué avec la complexité $O(n^3)$.

□

3.2.3 La récurrence

Nous rappelons que nous dénotons par $H_{B_{\geq k}}$ la matrice carrée obtenue de la matrice H par les blocs à partir du k-ième bloc

$$H_{B_{\geq k}} = \begin{pmatrix} H_{B_k,B_k} & H_{B_k,B_{k+1}} & \cdots & H_{B_k,B_m} \\ & H_{B_{k+1},k+1} & \cdots & H_{B_{k+1},n} \\ & \ddots & & \\ & & & H_{B_m,B_m} \end{pmatrix}$$

Notation 9 *Nous notons $u_{B_{\geq k}}$ un vecteur cyclique pour $H_{B_{\geq k}}$. Nous supposons de plus que les polynômes caractéristiques des blocs diagonaux sont premiers deux à deux.*

Première étape : Calcul de u_{B_m}. La matrice H_{B_m,B_m} est une matrice compagnon, et le vecteur $(1, 0, \ldots, 0)$ est un vecteur cyclique pour H_{B_m,B_m}.
Étape de récurrence : Supposons que le problème a été résolu pour $H_{B_{\geq k+1}}$, c'est-à-dire que nous connaissons un vecteur $u_{B_{\geq k+1}}$ qui est cyclique pour $H_{B_{\geq k+1}}$.

Nous construisons $u_{B_{\geq k}} = (u_{B_k}, u_{B_{\geq k+1}})$ qui est un vecteur cyclique pour $H_{B_{\geq k}}$, de la manière suivante. Notons w_{B_k} le vecteur

$$w_{B_k} = (f_{k+1}f_{k+2}\cdots f_m(H)u^\star_{B_{\geq k+1}})_{B_k}$$

Par le lemme IV.10, la relation suivante est résolue en u_{B_k}

$$(f_{k+1}f_{k+2}\cdots f_m)(H_{B_k,B_k})u_{B_k} + w_{B_k} = v^\dagger \tag{IV.21}$$

pour $v^\dagger$ cyclique pour H_{B_k,B_k}. Par exemple $v^\dagger = (1, 0 \ldots 0)$.

Le polynôme $f_{k+1}(X)\cdots f_m(X)$ est premier à $f_k(X)$ et admet pour inverse $h_k(X)$ mod $f_k(X)$. Ainsi nous avons

$$u_{B_k} = h_k(H_{B_k,B_k})(1 - w_{B_k}).$$

3.2.4 Une borne supérieure de complexité

Nous évaluons maintenant le nombre d'opérations nécessaires dans l'algorithme précédent. Les calculs les plus coûteux sont les calculs des vecteurs $w_{B_{m-1}}, w_{B_{m-2}}, \ldots, w_{B_1}$, qui sont obtenus au fur et à mesure en commençant par w_{B_m}.

$$\begin{aligned} w_{B_{m-1}} &= (f_m(H)w^\star_{B_m})_{B_{m-1}} \\ w_{B_{m-2}} &= ((f_{m-1}f_m)(H)w^\star_{B\geq m-1})_{B_{m-2}} \\ &\vdots \\ w_{B_k} &= ((f_{k+1}f_{k+2}\cdots f_m)(H)w^\star_{B\geq k+1})_{B_k} \end{aligned}$$

Le calcul de chaque vecteur w_{B_k} consiste à appliquer au plus n fois la matrice H sur un vecteur de taille n. Le coût de cette opération est $O(n^2m)$. L'obtention de u_{B_k} coûte $O(n^2m)$ opérations élémentaires. Enfin, un coût supplémentaire de $O(n_k^2)$ intervient pour calculer chacun des m pgcd de polynômes. Ceci donne $O(mn^2)$ opérations élémentaires supplémentaires.

Théorème IV.10 *Si le polynôme caractéristique d'une matrice A est sans facteurs multiples, alors l'algorithme décrit en 3.2.3 produit un vecteur cyclique pour A sur la donnée d'une forme de Hessenberg à décalage H de A, à un coût $O(m^2n^2)$.*

Corollaire IV.5 *Si le polynôme caractéristique d'une matrice A est sans facteurs multiples, un vecteur cyclique pour A peut être obtenu en $O(n^3 + m_A^2 n^2)$ opérations élémentaires.*

Remarque IV.3 *Notons que la complexité du pire est $O(n^4)$.*

Cet algorithme peut s'appliquer pour calculer une base normale de $\mathbf{F}_{q^n}$ lorsque n et q sont premiers entre eux. Toutefois, le paramètre m_A de l'isomorphisme de Frobenius est égal au nombre de facteurs de $X^n - 1$, qui peut être grand. Nous verrons en section 4.3 comment obtenir un vecteur cyclique à un coût $O(n^3)$, par une méthode "diviser pour régner".

4 Algorithmes "Diviser pour régner"

4.1 Calcul du polynôme minimal

4.1.1 Construction des sous-espaces caractéristiques

Nous montrons comment calculer une base d'un espace caractéristique d'une matrice A dont la factorisation du polynôme caractéristique est connue. Nous rappelons d'abord la définition des sous-espaces caractéristiques, et montrons comment une famille génératrice d'un espace caractéristique peut être obtenue en évaluant un polynôme en la matrice A. C'est cette opération qui est la plus coûteuse, et, au prix d'une mise en table de résultats intermédiaires, évaluer un polynôme de degré t en une matrice A peut s'effectuer en $O(n^3\sqrt{t})$. Dans le cas d'une matrice de Hessenberg à décalage, il est possible de mener ce calcul en $O(n^2 m^2)$.

Nous rappelons la définition des sous-espaces caractéristiques d'une matrice A.

Théorème IV.11 *Soit $C(X)$ le polynôme caractéristique d'une matrice $A \in M_n(\mathbf{k})$, et supposons que $C(X) = P(X)Q(X)$ où $P(X)$ et $Q(X)$ sont premiers entre eux. Alors l'espace $\mathbf{k}^n$ se décompose en sous-espaces invariants de la manière suivante*

$$\mathbf{k}^n = V_P \oplus V_Q, \qquad V_P = \ker P(A) \text{ et } V_Q = \ker Q(A).$$

De plus les deux sous-espaces V_P et V_Q sont

$$V_P = \operatorname{Im} Q(A) \text{ et } V_Q = \operatorname{Im} P(A).$$

Les sous-espaces caractéristiques sont alors définis de la manière suivante :

Définition IV.14 *Soit $A \in M_n(\mathbf{k})$ de polynôme caractéristique $C(X)$ et $C(X) = \prod_{i=1}^{k} f_i(X)^{r_i}$ la décomposition en facteurs irréductibles de $C(X)$. Les* sous-espaces caractéristiques *de A sont les sous espaces invariants $V_i = \ker f_i(A)^{r_i}$, $i = 1, \ldots, k$.*

Pour calculer une base de V_i, on considérera le polynôme $g_i(X) = C(X)/f_i(X)^{r_i}$. Les colonnes de la matrice $g_i(A)$ forment une famille génératrice de V_i, d'après le théorème IV.11. On pourra extraire une base de V_i en utilisant l'algorithme de Gauss.

Pour calculer le polynôme minimal d'une matrice A connaissant la factorisation de $C(X)$, nous procédons comme suit. Si le polynôme caractéristique de A est $C(X) = p(X)^r$ où $p(X)$ est irréductible, alors $\mathbf{k}^n$ est un sous-espace caractéristique, et trouver le polynôme minimal de A revient à calculer l'exposant minimal s tel que $p(A)^s = 0$.

Si le polynôme caractéristique n'est pas une puissance d'un polynôme irréductible, nous montrons qu'il est possible de scinder $C(X)$ en $C(X) = P(X)Q(X)$ où $P(X)$ et $Q(X)$ sont premiers entre eux, de sorte que : soit le polynôme caractéristique d'une des deux matrices $P(A)$ restreinte à V_Q ou $Q(A)$ restreinte à V_P est puissance d'un polynôme irréductible, soit nous avons $\deg P(X), \deg Q(X) \leq \frac{2}{3}n$. Nous appliquons récursivement la méthode sur V_P et sur V_Q. Les matrices des restrictions de A à V_P et V_Q sont décomposées à leur tour, jusqu'à ce que tous les sous-espaces caractéristiques A aient été obtenus. Finalement le polynôme minimal des restrictions de A à chaque sous-espace caractéristique est calculé. Le produit de ces polynômes minimaux donne le résultat final.

4.1.2 L'algorithme

Nous décrivons l'algorithme plus précisément.
Données : La matrice A et son polynôme caractéristique $C(X)$, donné avec sa factorisation en polynômes irréductibles :

$$C(X) = f_1(X)^{r_1} \dots f_k(X)^{r_k}.$$

Résultat : Le polynôme minimal de A et la décomposition de $\mathbf{k}^n$ en sous-espaces caractéristique de A.
Étape 1 : Grouper la factorisation de $C(X)$ en $C(X) = P(X)Q(X)$ où $P(X)$ et $Q(X)$ sont premiers entre eux. Trois groupements sont possibles.

- $C(X) = p(X)^r$, $p(X)$ irréductible. On calcule le polynôme minimal $p(X)^s$ de A par essai sur s. Ceci peut être effectué en $O(n^3\sqrt{n})$, en utilisant la technique d'évaluation présentée plus loin dans la preuve du théorème IV.12 pour calculer $p(A)$, l'exposant minimal s étant ensuite recherché par dichotomie, en $O(\lceil log_2 r \rceil)$ multiplications de matrices.

- Un des facteurs de la décomposition, $p_i(X)^{r_i}$, est de degré supérieur à $\frac{2}{3}n$. Alors on choisit $P(X) = p_i(X)^{r_i}$, c'est-à-dire $C(X) = p_i(X)^{r_i}Q(X)$, et $Q(A)$ donne une base d'un sous-espace caractéristique.

- Tous les facteurs $p_i(X)^{r_i}$ sont de degré inférieur à $\frac{2}{3}n$. Il est alors possible de grouper la factorisation $C(X) = P(X)Q(X)$ où $P(X)$ et $Q(X)$ sont premiers entre eux et où $\deg P(X) \leq \frac{2}{3}n$, $\deg Q(X) \leq \frac{2}{3}n$. Ceci sera décrit dans le lemme IV.12.

Étape 2 : Calcul de $Q(A)$, $P(A)$. Ceci donne deux familles de vecteurs générateurs pour les deux sous-espaces V_P et V_Q respectivement. Ceci est fait à un coût $O(n^3\sqrt{n})$ en utilisant la technique démontrée dans la preuve du théorème IV.12.
Étape 3 : Calcul de deux bases de V_P et V_Q respectivement, en utilisant la technique d'élimination de Gauss, à un coût $O(n^3)$.
Étape 4 : En changeant de base, prenant pour la nouvelle base l'union des deux bases précédemment calculées, les matrices A_P et A_Q des restrictions de A à V_P et V_Q respectivement sont calculées. Le coût est à nouveau $O(n^3)$.
Étape récursive : L'algorithme est appliqué récursivement à A_P et A_Q, les cas terminaux correspondant aux bases des sous-espaces caractéristiques de A.

Il reste à expliquer comment le regroupement des facteurs du polynôme caractéristique peut être effectué, et comment évaluer efficacement un polynôme en une matrice.

4.1.3 Évaluation d'un polynôme en une matrice

Soit A une matrice carrée de taille n. Nous montrons que $p(A)$ peut être calculé à un coût de $O(\sqrt{t}n^3)$, où t est le degré de $p(X)$. En utilisant la méthode de Horner, cela nous mènerait à un coût de $O(tn^3)$. Il est possible de faire mieux que la méthode de Horner, car l'évaluation d'un polynôme en une matrice fait intervenir deux lois de multiplication : l'une interne de coût n^3, l'autre externe de coût n^2.

La méthode est une variante de la méthode "pas de bébé, pas de géant" de Shank, et nécessite de conserver $\sqrt{t}$ matrices en table.

Théorème IV.12 *Pour toute matrice $A \in M_n(\mathbf{k})$, pour tout polynôme $U(X)$ de $\mathbf{k}[X]$ de degré au plus t, $U(A)$ peut être calculé en $O(\sqrt{t}n^3)$, en utilisant une taille mémoire de $O(\sqrt{t}n^3)$.*

Preuve : Pour simplifier, nous décrivons l'algorithme dans le cas où $t = d^2 - 1$, pour un entier d. Nous avons à calculer

$$U(A) = u_0 + u_1 A + u_2 A^2 + \ldots + u_t A^t. \tag{IV.22}$$

Soit $B = A^d$, nous décomposons le polynôme U en polynômes de taille d :

$$\begin{aligned} U(A) &= u_0 + u_1 A + u_2 A^2 + \ldots + u_{d-1} A^{d-1} \\ &\quad + (u_d + u_{d+1} A + u_{d+2} A^2 + \ldots + u_{d+d-1} A^{d-1}) B \\ &\quad + (u_{2d} + u_{2d+1} A + u_{2d+2} A^2 \ldots + u_{2d+d-1} A^{d-1}) B^2 \\ &\quad \ldots \\ &\quad + (u_{d(d-1)} + u_{d(d-1)+1} A + \ldots + u_{d(d-1)+(d-1)} A^{d-1}) B^{d-1} \\ &= U_0(A) + U_1(A) B + U_2(A) B \cdots + U_{d-1} B^{d-1} \end{aligned}$$

Un précalcul est mené pour enregistrer en table les matrices suivantes :

A	A^2	A^3	$\ldots$	A^{d-1}	$B = A^d$	B^2	$\ldots$	B^{d-1}

Le coût de ce précalcul est de $2d - 3$ multiplications de matrices. Calculer $U_i(A)$ ne fait intervenir aucune multiplication de matrice, puisque chaque matrice $A^i, 0 \leq i \leq d-1$ est lue dans la table. Ensuite chaque produit $U_i(A)B^i$ est à effectuer, ce qui mène à d multiplications de matrices supplémentaires.

D'où un coût total de $O(dn^3)$.

□

4.1.4 Regroupement des facteurs du polynôme caractéristique

La procédure de partition des facteurs du polynôme caractéristique peut s'appliquer à d'autres problèmes. Nous l'indiquons donc en toute généralité, d'autant plus que ces propriétés interviendront aussi pour évaluer la complexité de l'algorithme récursif pour calculer un vecteur cyclique.

Définition IV.15 *Un* multi-ensemble *est une application de E dans* N *où E est un sous-ensemble de* N.

Ainsi un multi-ensemble définit une suite d'entiers positifs $n_{i_1}, \ldots, n_{i_k}, \ldots$ Pour les définitions qui suivent nous supposons que E est de cardinal fini.

Définition IV.16 *Étant donné un multi-ensemble, nous notons $n(E)$ le nombre $\sum_{i\in E} n_i$. Pour $0 < \theta < 1$, une* partition θ-équitable *d'un multi-ensemble E est une partition $I \cup J$ de E qui vérifie : soit I (ou J) est réduit à un singleton $\{i\}$ et $n_i > \theta n(E)$, soit $n(I) \leq \theta n(E)$ et $n(J) \leq \theta n(E)$.*

Définition IV.17 *Une* partition récursive *de E est un arbre binaire dont les nœuds sont des sous-ensembles de E, les feuilles sont des singletons, et les sommets des deux sous-arbres d'un nœud sont une partition de ce nœud.*

Définition IV.18 *Pour $0 < \theta < 1$, une* partition θ-équitable récursive *de E, est une partition récursive de E, telles que les sommets des deux sous-arbres d'un nœud soit une partition θ-équitable de ce nœud.*

Soit un problème $P(E)$ défini pour tout multi-ensemble E. On suppose que résoudre le problème $P(E)$ est tel qu'il peut se réduire à résoudre les problèmes $P(I)$ et $P(J)$ quelle que soit la partition $I \cup J$ de E. On suppose de plus que le coût de la création de la partition $I \cup J$ de E est $O(n(E)^e)$ $(= \alpha n^e)$, et que le coût pour construire la solution du problème $P(E)$ à partir des solutions de $P(I)$ et de $P(J)$ est aussi $O(n(E)^e)$ $(= \beta n^e)$. Nous notons $C(E)$ le coût le plus faible pour résoudre $P(E)$ sur toutes les partitions récursives de E, et $C(n)$ est le coût le plus grand des $C(E)$ pour tous les multi-ensembles E tel que $n(E) = n$.

Le lemme suivant montre que le coût total pour résoudre $P(E)$ avec $n(E) = n$ reste $O(n^e)$, si e est assez grand.

Lemme IV.12 *Soit $P(E)$ avec $n(E) = n$, et soit un problème P vérifiant les conditions ci-dessus. Alors, si $\theta \geq \frac{2}{3}$, il existe une partition récursive de E θ-équitable . Si de plus $\theta = \frac{2}{3}$, on a $C(n) = O(n^e) = \gamma n^2$ avec $\gamma = \frac{\alpha+\beta}{1-2\theta^e}$, si e est supérieur à $\frac{\log 2}{\log 3 - \log 2} \simeq 1.71$.*

Preuve : Nous avons à montrer que, étant donné un multi-ensemble $n_1, n_2, \ldots, n_k$ avec $n(E) = n$, il existe une partition θ-équitable si $\theta \geq \frac{2}{3}$. D'après la définition d'une partition θ-équitable nous pouvons supposer que $n_i \leq \theta n, i = 1, \ldots, k$. Maintenant s'il existe i tel que $n_i > (1-\theta)n$, alors la partition $I = \{i\}$ et $J = E \setminus I$ est θ-équitable. Il reste le cas où $n_i \leq (1-\theta)n$, $i = 1, \ldots, k$. Soit alors J le sous-ensemble de E de cardinal maximal tel que $\sum_{j\in J} n_j \leq \theta n$. Montrons que $\sum_{i\in I} n_i \leq \theta n$, pour $I = E \setminus J$. Par l'absurde : si $\sum_{i\in I} n_i > \theta n$, en ôtant un élément i de I pour l'ajouter à J, I devenant I', J devenant J', la maximalité de J entraîne

$$\sum_{j\in J'} n_j > \theta n,$$

et donc

$$\sum_{i\in I'} n_i < (1-\theta)n.$$

Or

$$\sum_{i\in I'} n_i > n(I) - \theta n \geq (1-2\theta)n.$$

Ceci entraîne $2\theta - 1 < 1 - \theta$ et contredit l'hypothèse $\theta \geq \frac{2}{3}$.

Si $\theta = \frac{2}{3}$, montrons que $C(n)$ est borné supérieurement par γn^e. La preuve est par récurrence sur n. Le résultat est vrai pour $n = 2$ et pour $n > 2$, $\theta n \leq n - 1$. On a

$$\begin{aligned} C(n) &\leq \alpha n^e + \max(C((1-\theta)n) + \beta n^e, 2C(\theta n)) \\ &\leq \alpha n^e + \beta n^e + 2C(\theta) = (\alpha+\beta)n^e + 2\gamma\theta^e n^e. \end{aligned}$$

Ainsi nous avons que $C(n) \leq \gamma n^e$ avec $\gamma = \frac{\alpha+\beta}{1-2\theta^e}$, qui est positif si $e > \frac{\log 2}{\log 3 - \log 2}$.

□

D'un point de vue algorithmique cette partition peut être obtenue en triant par taille croissante les entiers n_i, $i \in E$. On les ajoute jusqu'à ce qu'une valeur $n_1 + \cdots + n_k$ supérieure à $\frac{2}{3}n$ est obtenue. Alors $I = \{1, \ldots, k-1\}$.

4.1.5 Complexité de l'algorithme

Théorème IV.13 *Soit A une matrice de $M_n(\mathbf{k})$, il est possible de calculer le polynôme minimal de A, et une matrice diagonale par blocs semblable à A, exhibant les sous-espaces caractéristiques de A, avec une complexité de $O(n^3\sqrt{n})$, et une taille mémoire de $O(n^3\sqrt{n})$.*

Preuve : Le théorème est prouvé en prenant $e = 3.5$ dans le lemme IV.12.

□

4.2 Le même algorithme sur la forme de Hessenberg à décalage

Cet algorithme peut être employé sur des matrices Hessenberg à décalage, plutôt que sur des matrices quelconques. Le point crucial est qu'évaluer un polynôme en une matrice H de Hessenberg à décalage est moins coûteux, si le paramètre m de H est petit. Cette amélioration se retrouvera dans la complexité finale de l'algorithme.
Données Une matrice H de Hessenberg à décalage et la factorisation de son polynôme caractéristique en facteurs multiples.
Résultat Le polynôme minimal de H et une décomposition de $\mathbf{k}^n$ en sous-espaces caractéristiques de H.

Étape 1 : Regrouper les facteurs $C(X) = P(X)Q(X)$, comme dans l'étape 1 de l'algorithme précédent.
Étape 2 : Calcul de $Q(H)$, et de $P(H)$ (corollaire IV.1). Ceci donne deux familles de vecteurs générateurs des sous-espaces V_P et V_Q respectivement.
Étape 3: Calcul de bases pour V_P et V_Q respectivement.
Step 4: On calcule les deux matrices H_P et H_Q des restrictions de H à V_P et V_Q respectivement. Le coût est à nouveau $O(n^3)$. On calcule des formes de Hessenberg à décalage H'_P et H'_Q des deux matrices H_P et H_Q.
Étape récursive On applique récursivement l'algorithme à H'_P et H'_Q. Ceci se termine en produisant les sous-espaces caractéristiques.

Corollaire IV.6 *Soit H une matrice de Hessenberg à décalage de $M_n(\mathbf{k})$, étant donnée la factorisation du polynôme caractéristique de H, il est possible de calculer le polynôme minimal d'une matrice de H et une matrice D bloc diagonale semblable à H en $O(n^3 + m_H^2 n^2)$ opérations élémentaires.*

Remarque IV.4 *Le terme n^3 dans l'évaluation précédente est dû au calcul de la forme de Hessenberg à décalage d'une matrice donnée, et à la construction des bases pour les sous-espaces invariants.*

Remarque IV.5 *En calculant une forme de Hessenberg à décalage d'une matrice donnée A, tous les résultats du corollaire IV.6 sont obtenus à un coût $O(n^3 + m_A^2 n^2)$ par le théorème IV.4.*

4.3 Calcul "diviser pour régner" d'un vecteur cyclique

Pour calculer un vecteur cyclique d'une matrice de Hessenberg à décalage H dont le polynôme caractéristique est sans facteurs multiples, la complexité de l'algorithme direct présenté en 3.2 devient $O(n^4)$ lorsque m est grand. Nous développons une procédure récursive plus sophistiquée, de complexité $O(n^3)$, pour toute valeur de m. Nous présentons une technique de décomposition de matrices de Hessenberg à décalage, et finalement donnons la description complète de l'algorithme. Cet algorithme fonctionne lorsque le polynôme minimal de H est sans facteurs multiples.

4.3.1 Stratégie

Soit à calculer un vecteur cyclique d'une matrice A dont le polynôme caractéristique est sans facteurs multiples. Une forme de Hessenberg à décalage H est d'abord calculée. Ensuite la stratégie est de découper la matrice de Hessenberg à décalage en deux sous matrices, dont les tailles restent sous contrôle.

Nous conservons les même notations que dans la partie 3.1. La matrice H est de la forme

$$H = \begin{pmatrix} H_{B_1,B_1} & H_{B_1,B_2} & \cdots & H_{B_1,B_m} \\ 0 & H_{B_2,B_2} & \cdots & H_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & H_{B_m,B_m} \end{pmatrix}$$

Notation 10 *Pour tout $I \subset [1,n], J \subset [1,n]$, nous notons $H_{I,J}$ la sous matrice formée des lignes de H dans I et des colonnes H dans J. La taille de I est notée n_I.*

Le découpage consiste en trouver une matrice H_{split} équivalente à H de la forme

$$H_{split} = \begin{pmatrix} H'_{B_I,B_I} & H'_{B_I,B_J} \\ 0 & H'_{B_J,B_J} \end{pmatrix}, \tag{IV.23}$$

qui est une matrice de Hessenberg à décalage telle que $n_I \leq \frac{2}{3}n, n_J \leq \frac{2}{3}n$. Nous appliquons récursivement l'algorithme sur les deux matrices H'_{B_I,B_I} et H'_{B_J,B_J}, pour trouver v_{B_I}, v_{B_J} qui sont des vecteurs cycliques de H'_{B_I,B_I} et H'_{B_J,B_J} respectivement.

Ensuite un vecteur u' cyclique pour H_{split} peut être calculé par le lemme IV.11, v_{B_I} et v_{B_J} étant connus. En changeant de base, on obtient un vecteur u cyclique pour H.

4.3.2 Le scindage

Nous expliquons dans un lemme comment scinder la matrice en deux sous-matrices. Avant de l'énoncer, nous expliquons un point technique au sujet des matrices de Hessenberg à décalage perturbées par une permutation de lignes et de colonnes.

Soit la matrice de Hessenberg à décalage :

$$H = \begin{pmatrix} H_{B_1,B_1} & H_{B_1,B_2} & \cdots & H_{B_1,B_k} & \cdots & H_{B_1,B_m} \\ 0 & H_{B_2,B_2} & \cdots & H_{B_2,B_k} & \cdots & H_{B_2,B_m} \\ \vdots & & \ddots & \vdots & \cdots & \vdots \\ \vdots & & & H_{B_k,B_k} & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & H_{B_m,B_m} \end{pmatrix},$$

sur laquelle on effectue une permutation de lignes et de colonnes telle que le bloc H_{B_k,B_k} est permuté avec le bloc H_{B_1,B_1}, de façon à obtenir la matrice H_{swap} semblable à H :

$$H_{swap} = \begin{pmatrix} H_{B_k,B_k} & 0 \cdots 0 & 0 & H_{B_k,B_{>k}} & \\ H_{B_{[2,k-1]},B_k} & H_{B_{[2,k-1]},B_{[2,k-1]}} & H_{B_{[2,k-1]},B_1} & H_{B_{[2,k-1]},B_{>k}} & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ H_{B_1,B_k} & H_{B_1,B_{[2,k-1]}} & H_{B_1,B_1} & H_{B_1,B_{>k}} & \\ H_{B_{>k},B_k} & H_{B_{>k},B_{[2,k-1]}} & H_{B_{>k},B_1} & H_{B_{>k},B_{>k}} & \end{pmatrix}$$

Ce n'est pas une matrice de Hessenberg à décalage, et nous en calculons une forme de Hessenberg à décalage H'. Ceci conduit à la matrice

$$H' = \begin{pmatrix} H'_{B_1,B_1} & H'_{B_1,B_2} & \cdots & H'_{B_1,B_k} & \cdots & H'_{B_1,B_m} \\ 0 & H'_{B_2,B_2} & \cdots & H'_{B_2,B_k} & \cdots & H'_{B_2,B_m} \\ \vdots & & \ddots & \vdots & \cdots & \vdots \\ \vdots & & & H'_{B_k,B_k} & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & H'_{B_m,B_m} \end{pmatrix}.$$

Le lemme suivant établit une relation entre le polynôme compagnon du bloc H'_{B_1,B_1} et polynôme compagnon du bloc H_{B_k,B_k}.

Lemme IV.13 *Soit f_k le polynôme compagnon du bloc H_{B_k,B_k} de la matrice H, et soit f'_1 le polynôme compagnon du bloc H'_{B_1,B_1} de la matrice H' obtenue par la transformation précédente. Alors f_k divise f'_1.*

Notation 11 *Nous notons ϵ_k le vecteur de k^n tel que $(\epsilon_k)_{B_k} = (1, 0, \ldots, 0)$ et $(\epsilon_k)_{B_j} = (0, \ldots, 0)$, $j \neq k$.*

Preuve : Nous savons que f_k divise le polynôme minimal de H relativement à ϵ_k. La permutation de H à H_{swap} revient à placer le vecteur ϵ_k comme premier vecteur de la nouvelle base. L'algorithme de réduction en forme de Hessenberg à décalage calcule une matrice dont le premier bloc est une matrice compagnon dont le polynôme compagnon est le polynôme minimal du premier vecteur. Ainsi f'_1 est le polynôme minimal de ϵ_k, qui est un multiple de f_k.

□

Voici maintenant le lemme de scindage de matrices de Hessenberg à décalage :

Lemme IV.14 (Scindage de la matrice) *Soit H une matrice de Hessenberg à décalage. Il est toujours possible de déterminer une matrice de Hessenberg à décalage H_{split} et une matrice P inversible telle que $H = PH_{split}P^{-1}$ avec H_{split} de la forme*

$$H_{split} = \begin{pmatrix} H'_{B_I,B_I} & H'_{B_I,B_J} \\ 0 & H'_{B_J,B_J} \end{pmatrix} \tag{IV.24}$$

et vérifiant l'une des deux possibilités

1. *soit H'_{B_I,B_I} est un bloc compagnon de taille $\geq \frac{2}{3}n$, et H'_{B_J,B_J} est une matrice de Hessenberg à décalage de taille $\leq \frac{1}{3}n$.*

2. *soit H'_{B_I,B_I} et H'_{B_J,B_J} sont des matrices de Hessenberg à décalage de taille inférieure à $\frac{2}{3}n$.*

De plus H_{split} et P peuvent être calculées en $O(n^3)$ opérations.

Preuve : Deux cas principaux sont à considérer.

$\exists k \in [1,m] \mid n_k \geq \frac{2}{3}n$.

On choisit $I = B_k$, $J = [1,m] \setminus I$. Alors $n_J \leq \frac{1}{3}n$ mais le bloc B_k peut ne pas être le premier bloc. Par permutation de lignes et de colonnes, le bloc B_k est placé en première position. Ceci mène à une matrice H_{swap} qui n'est pas Hessenberg à décalage. Nous appliquons l'algorithme de réduction en forme de Hessenberg à décalage pour produire la matrice H_{swap}. Par le lemme IV.13, la taille du premier bloc a augmenté, et donc est supérieure à $\frac{2}{3}n$. Ceci donne la matrice H_{split} du cas 1 à un coût $O(n^3)$.

$\forall j \in [1,m], n_j < \frac{2}{3}n$.

Supposons d'abord que tous les n_i sont inférieurs à $\frac{1}{3}n$. Dans la famille des ensembles $I_i = \{1,2,\ldots i\}$, nous notons I_{i_0} celui de cardinal maximal tel que $\sum_{j\in I_i} n_j < \frac{2}{3}n$. Alors $I = B_1 \cup B_2 \cdots B_{i_0}$ et $J = B_{i_0+1} \cup B_{i_0+2} \cdots \cup B_m$ vérifient $n_I \leq \frac{2}{3}n$ et $n_J \leq \frac{2}{3}n$. En effet puisque $n_{J\setminus\{i_0+1\}} < \frac{1}{3}n$, nous avons que $n_J < \frac{1}{3}n + n_{i_0+1} \leq \frac{2}{3}n$. Alors la matrice H_{split} est la matrice H. Ceci est le cas 2.

Sinon, il existe $n_k \geq \frac{1}{3}n$, et nous choisissons $I = B_k$, $J = [1,m] \setminus I$. Nous avons $n_I \leq \frac{2}{3}n$, $n_J \leq \frac{2}{3}n$. Par une permutation de lignes et de colonnes, le bloc H_I est amené en première position, et la matrice est ensuite réduite en matrice de Hessenberg à décalage en $O(n^3)$ opérations élémentaires. La taille du premier bloc a pu être accrue. Soit la taille du premier bloc demeure inférieure à $\frac{2}{3}n$ ce qui correspond au cas 2, soit la taille du premier bloc dépasse $\frac{2}{3}n$, ce qui est le cas 1. Dans ces deux cas, la taille du deuxième bloc reste inférieure à $\frac{2}{3}n$.

□

4.3.3 L'algorithme

Nous présentons maintenant l'algorithme complet pour calculer un vecteur cyclique d'une matrice A dont le polynôme minimal est sans facteurs multiples.

étape 1* : calcul d'une forme de Hessenberg à décalage de A. Celle-ci est obtenue en $O(n^3)$ opérations par le théorème IV.4. Ceci n'est effectué qu'une seule fois.

étape 2 : scindage et récursion. Le scindage indiqué par le lemme IV.14 est effectué, et nous obtenons deux sous-matrices H'_{B_I,B_I} et H'_{B_J,B_J}.

L'algorithme est alors appliqué récursivement sur les matrices H'_{B_I,B_I} et H'_{B_J,B_J} si elles ne sont pas des matrices compagnons. Dans le cas d'une matrice compagnon, le vecteur $(1, 0, \ldots, 0)$ est retourné.

étape 3 : reconstruction d'un vecteur cyclique. Les sous cas H'_{B_I,B_I} et H'_{B_J,B_J} ont été traités, et les solutions u_{B_1} et u_{B_2} ont été obtenues. Le lemme IV.11 nous indique comment construire un vecteur cyclique pour H_{split} à un coût $O(n^3)$.

étape 4 : changement de base. La donnée d'un vecteur cyclique pour H_{split} permet de calculer un vecteur cyclique pour H, par changement de base, en $O(n^3)$.

4.3.4 La complexité

Le coût de chaque étape est évalué.

étape 1* : le coût est $O(n^3)$, une seule fois.

étape 2 : le scindage coûte $O(n^3) = a_1 n^3$.

étape 3 : la reconstruction coûte $O(n^3) = a_2 n^3$.

étape 4 : le changement de base coûte $O(n^3)$.

étape 5*: Un changement de base de H à A est effectué pour obtenir un vecteur cyclique dans la base d'origine, à un coût $a_3 n^3$.

Seule les étapes 2,3 et 4 sont appliquées récursivement. Le coût total de ces étapes est $(a_1 + a_2)n^3 = an^3$.

Théorème IV.14 *Soit une matrice $A \in M_n(k)$ dont le polynôme minimal est sans facteurs multiples. Un vecteur cyclique pour A peut être calculé en $O(n^3)$ opérations élémentaires sur F_q.*

Preuve : On emploie le lemme IV.12, avec $e = 3$.

□

Corollaire IV.7 *Si n est premier à q, sur la donnée d'une matrice représentant l'opérateur de Frobenius $x \mapsto x^q$, il est possible de calculer une base normale de F_{q^n} en $O(n^3)$ opérations élémentaires.*

Preuve : Le polynôme minimal de l'automorphisme de Frobenius est $X^n - 1$, qui est sans facteurs multiples si $\mathrm{pgcd}(n, q) = 1$. Etant donnée la matrice F_n de l'automorphisme de Frobenius (calculée en $O(n^3 \log(q))$), nous sommes capables de calculer un vecteur cyclique pour F_n en $O(n^3)$. Ce vecteur est un élément normal.

□

5 Calcul de la forme rationnelle canonique

5.1 Définitions et Notations

Soit A un opérateur linéaire. Nous considérons $\mathbf{k}^n$ muni de la structure de $\mathbf{k}[X]$-module induite par $\mathbf{A}$.

Notation 12 *Soit $p \in \mathbf{k}[X]$, et $v \in \mathbf{k}^n$, nous utilisons la notation de $\mathbf{k}[X]$-module pv pour $p(A)v$. Nous utiliserons aussi la notation pv lorsque p est un polynôme $p(A_{B_i})$ évalué en la restriction de A à un sous-espace invariant $\mathbf{k}^{B_i}$, et lorsque v est un vecteur de $\mathbf{k}^{B_i}$.*

De plus nous avons besoin d'une notation spécifique pour les colonnes d'une matrice de Hessenberg à décalage.

Notation 13 *Soit H une matrice Hessenberg à décalage*

$$H = \begin{pmatrix} H_{B_1,B_1} & H_{B_1,B_2} & \cdots & H_{B_1,B_m} \\ 0 & H_{B_2,B_2} & \cdots & H_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & H_{B_m,B_m} \end{pmatrix}.$$

Nous notons ϵ_i le vecteur de la base canonique de $\mathbf{k}^n$ tel que $(\epsilon_i)_{B_i} = {}^t(1,\ldots,0)$. Toutes les colonnes de H sont les $A^i(\epsilon_j)$.

Nous fixons $e_i = f_i\epsilon_i$. Le vecteur e_i est le vecteur "au dessus" du i-ième bloc dans la forme de Hessenberg à décalage.

5.2 Calcul préliminaire

Nous calculons d'abord une forme bloc diagonale de la matrice A exhibant les sous-espaces caractéristiques. D'après l'algorithme présenté en 4.1.1, une telle matrice peut être obtenue en $O(n^{3.5})$ opérations élémentaires, ou, en utilisant la forme de Hessenberg à décalage en $O(n^3)$ en moyenne. Rappelons que la factorisation du polynôme caractéristique doit être connue, ce qui restreint notre méthode au corps finis. Notre objectif sera alors de produire la forme rationnelle canonique de la restriction de A aux sous-espaces caractéristiques, ce qui donnera la forme rationnelle canonique développée de A. Nous montrons comment recouvrer la forme rationnelle canonique d'une matrice A lorsque les formes rationnelles canoniques de ses sous-espaces caractéristiques ont été obtenues. En effet soit donnée une matrice bloc diagonale semblable à A :

$$D = \begin{pmatrix} F_{B_1,B_1} & 0 & \cdots & 0 \\ 0 & F_{B_2,B_2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & F_{B_d,B_d} \end{pmatrix}$$

où chaque matrice F_{B_i,B_i} a la forme rationnelle canonique

$$F_{B_i,B_i} = \begin{pmatrix} C_{p_i^{s_{i,1}}} & 0 & \cdots & 0 \\ 0 & C_{p_i^{s_{i,2}}} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & C_{p_i^{s_{i,m_i}}} \end{pmatrix}$$

et où $s_{i,1} \leq s_{i,2} \leq \cdots \leq s_{i,m_i}$, $i = 1,\ldots,d$. Nous avons ainsi que $p_i^{s_{i,m_i}}$ est le polynôme minimal de F_{B_i,B_i}. Les sous-espaces pour laquelle la matrice est une matrice compagnon sont notés $V_{p_i^{s_{i,1}}}, V_{p_i^{s_{i,2}}} \ldots V_{p_i^{s_{i,m_i}}}$ et un vecteur cyclique pour chacun de ces sous-espaces est noté $\epsilon_{p_i^{s_{i,j}}}$.

Nous considérons les sous-espaces

$$W_1 = V_{p_1^{s_{1,m_1}}} \oplus V_{p_2^{s_{2,m_2}}} \oplus \cdots \oplus V_{p_d^{s_{d,m_d}}}$$

$$W_2 = V_{p_1}^{s_{1,m_1-1}} \oplus V_{p_2}^{s_{2,m_2-1}} \oplus \cdots \oplus V_{p_d}^{s_{d,m_d-1}}$$

$$\vdots$$

formés en regroupant les sous-espaces cycliques par ordre décroissant d'exposant de chaque polynôme irréductible. Considérons aussi les vecteurs

$$E_1 = \epsilon_{p_1}^{s_{1,m_1}} + \epsilon_{p_2}^{s_{2,m_2}} + \cdots + \epsilon_{p_d}^{s_{d,m_d}}$$

$$E_2 = \epsilon_{p_1}^{s_{1,m_1-1}} + \epsilon_{p_2}^{s_{2,m_2-1}} + \cdots + \epsilon_{p_d}^{s_{d,m_d-1}}$$

$$\vdots$$

regroupés de la même manière. Alors chaque E_i est un vecteur cyclique de W_i, pour chaque valeur de i. Chaque vecteur cyclique définit ainsi un sous-espace invariant tel que le polynôme minimal f_{i+1} de A restreint à E_{i+1} divise le polynôme minimal f_i de A restreint à E_i. La matrice représentant la restriction de A dans la base $\{E_i, AE_i, A^2E_i, \dots\}$ est une matrice compagnon, et la matrice de A dans la base

$$\{E_1, AE_1, \dots E_i, AE_i, A^2E_i, \dots\}$$

correspond à la forme rationnelle canonique de A.

5.3 Cas d'un sous-espace caractéristique

Nous considérons donc un opérateur A dont le polynôme caractéristique est de la forme $C(X) = p(X)^r$, avec $r \geq 1$, pour $p(X)$ irréductible.

Nous appliquons l'algorithme de réduction en forme de Hessenberg à décalage H. La matrice H peut être écrite de la sorte

$$H = \begin{pmatrix} H_{B_1,B_1} & H_{B_1,B_2} & \cdots & H_{B_1,B_m} \\ 0 & H_{B_2,B_2} & \cdots & H_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & H_{B_m,B_m} \end{pmatrix}.$$

Le polynôme minimal f_i de la matrice compagnon H_{B_i,B_i} est p^{s_i}. Le but est d'obtenir une somme directe de deux blocs diagonaux, le premier étant un bloc compagnon, le deuxième bloc étant une matrice de Hessenberg à décalage. Le procédé sera ensuite appliqué au deuxième bloc diagonal.

Le cas favorable est lorsque les composantes des vecteurs e_i sur le premier bloc de coordonnées vérifient $(e_i)_{B_1} = e_i^\dagger p^{r_i}$, $\text{pgcd}(e_i^\dagger, p) = 1$, et $r_i \geq s_i$. Nous introduisons alors les vecteurs

$$\begin{aligned} \epsilon'_2 &= \epsilon_2 - e_2^\dagger p^{r_2-s_2}\epsilon_1 \\ \epsilon'_3 &= \epsilon_3 - e_3^\dagger p^{r_3-s_3}\epsilon_1 \\ &\vdots \\ \epsilon'_m &= \epsilon_m - e_m^\dagger p^{r_m-s_m}\epsilon_1, \end{aligned}$$

qui vérifient $(p^{s_i}\epsilon'_i)_{B_1} = 0$, $i = 2 \dots m$, puisque $p^{s_i}\epsilon_i = e_i$. Ils forment une base dans laquelle la matrice a la forme

$$\begin{pmatrix} C_{p^{s_1}} & 0 \\ 0 & H' \end{pmatrix},$$

et le procédé est ensuite appliqué à H'.

Sinon, il existe i tel que $r_i < s_i$. Soit ϵ_j le vecteur tel que $s_j - r_j$ est le plus grand. Par une permutation de lignes et de colonnes, nous plaçons ce vecteur ϵ_j en première position. En appliquant l'algorithme de réduction, nous calculons une nouvelle forme de Hessenberg à décalage, dont le premier bloc est une matrice compagnon, dont le polynôme minimal est le polynôme minimal de ϵ_j (voir lemme IV.13). Nous affirmons que l'exposant du polynôme minimal de ϵ_j est strictement supérieur à s_1. La taille du premier bloc a donc augmenté. Le processus s'arrête lorsque nous avons $s_i \leq r_i$ pour tout i, ce qui est la situation favorable, ou s'il ne reste plus qu'un seul bloc compagnon.

Nous prouvons maintenant notre affirmation. Soit j tel que $r_j < s_j$. Alors le polynôme minimal de ϵ_j est d'exposant supérieur à s_1.

Preuve : Nous calculons l'exposant minimal s tel que $p^s\epsilon_j = 0$. Nous devons avoir $p^{s_j}\epsilon_j$ à calculer. En effet $p^{s_j}(H)_{B_j,B_j} = 0$ et $(p^{s_j}\epsilon_j)_{B_j}$ est nul. Ceci s'écrit

$$p^{s_j}\epsilon_j = (e_j)^\star_{B_1} + (e_j)^\star_{B_2} + \cdots + (e_j)^\star_{B_{j-1}}.$$

Les coordonnées du bloc B_{j-1} doivent aussi être annulées, ce qui conduit à un facteur $p^{\lambda_{j-1}}$ tel que

$$p^{\lambda_{j-1}}p^{s_j}\epsilon_j = p^{\lambda_{j-1}}(e_j)_{B_1} + v(j-2).$$

Et le vecteur $v(j-2)$ est un vecteur de support inclus dans les blocs $B_1 \cup B_2 \cdots \cup B_{j-2}$. A chaque étape nous devrons avoir la relation

$$p^{\lambda_k} \cdots p^{\lambda_{j-1}}p^{s_j}\epsilon_j = p^{\lambda_k} \cdots p^{\lambda_{j-1}}(e_j)_{B_1} + v(k-1).$$

Jusqu'à ce que toutes les coordonnées soient nulles, sauf celles du premier bloc. Nous avons alors

$$\begin{aligned} p^{\lambda_2} \cdots p^{\lambda_{j-1}}p^{s_j}\epsilon_j &= p^{\lambda_2} \cdots p^{\lambda_{j-1}}(e_j)_{B_1} + v(1) \\ &= p^{\lambda_2} \cdots p^{\lambda_{j-1}}e_j^\dagger p^{r_j} + v(1). \end{aligned}$$

Sur le premier bloc de coordonnées, le $\mathbf{k}[X]$-module induit par H est $\mathbf{k}[X]/(p^{s_1}(X))$. Nous devons donc trouver l'exposant minimal l tel que

$$p^l(p^{\lambda_2} \cdots p^{\lambda_{j-1}}e_j^\dagger p^{r_j} + v(1)) = 0 \bmod p^{s_1}$$

et nous écrivons $v(1) = p^{r_0}v(1)^\dagger$ où $\gcd(p, v(1)^\dagger) = 1$.

Deux cas sont à considérer.

- $r_0 \geq \lambda_2 + \cdots + \lambda_{j-1} + r_j$.

  L'exposant l est

  $$l = s_1 - (\lambda_2 + \cdots + \lambda_{j-1} + r_j)$$

  et l'exposant du polynôme minimal de ϵ_j est

  $$l + \lambda_2 + \cdots + \lambda_{j-1} + s_j = s_1 - r_j + s_j > s_1$$

  puisque $r_j < s_j$.

- $r_0 < \lambda_2 + \cdots + \lambda_{j-1} + r_j$. L'exposant l est

$$l = s_1 - r_0 > s_1 - (\lambda_2 + \cdots + \lambda_{j-1} + r_j)$$

et l'exposant du polynôme minimal de ϵ_j est

$$\begin{aligned} s_1 - r_0 + \lambda_2 + \cdots + \lambda_{j-1} + s_j &> s_1 - (\lambda_2 + \cdots + \lambda_{j-1} + r_j) + \lambda_2 + \cdots + \lambda_{j-1} + s_j \\ &> s_1 - r_j + s_j > s_1 \end{aligned}$$

□

5.4 Complexité

Soit l'obtention directe du bloc supplémentaire, soit le processus d'augmentation de la taille du premier bloc a un coût $O(n^3)$. Ces processus sont appliqués au plus r fois. De plus les matrices de changement de base sont aussi obtenues. La complexité de cette méthode est donc $O(n^3 r)$.

Pour une matrice quelconque, pour tous les sous-espaces caractéristiques, la complexité est

$$O(n_1^3 r_1) + O(n_2^3 r_2) + \cdots + O(n_d^3 r_d) \leq O(n^3(r_1 + r_2 + \cdots + r_d))$$

Où $r_1 + r_2 + \cdots + r_d$ est le nombre de facteurs du polynôme caractéristique, comptés avec multiplicité. Ce nombre est asymptotiquement $\log n$ en moyenne, par le théorème IV.8.

Théorème IV.15 *Soit $A \in M_n(\mathbf{k})$. Sur la donnée des sous-espaces caractéristiques de A, la forme rationnelle canonique de A et la matrice de changement de base peuvent être calculées en $O(n^3 m_A)$, où m_A est le nombre de facteurs irréductibles du polynôme caractéristique de A, comptés avec multiplicités. La complexité moyenne de cet algorithme sur un corps fini est asymptotiquement $O(n^3 \log n)$.*

La borne que donne Patrick Ozello pour son algorithme est $8n^4 + 2n^3$. En réalité notre algorithme est très proche du sien, en augmentant la taille du premier bloc dans les cas défavorables, mais dans notre méthode nous sommes assurés d'augmenter la taille du premier bloc d'au moins $\deg p$, alors que dans sa méthode la taille augmente d'au moins 1.

6 Calcul de base normale dans un corps fini

Nous avons montré en 4.3 comment calculer une base normale pour F_{q^n}, lorsque n est premier à p, la caractéristique de F_q.

Nous allons généraliser la construction pour tout n. Nous écrivons $n = n_1 n_2$ où $n_2 = p^t$, et n_1 est premier à p. D'après le corollaire IV.7, nous sommes capables de calculer un base normale de $\mathrm{F}_{q^{n_1}}$ en $O(n^3)$ opérations élémentaires. Nous montrons d'abord comment calculer une base normale sur la donnée d'une base normale pour $\mathrm{F}_{q^{n_1}}$ et d'une base normale pour $\mathrm{F}_{q^{n_1}}$.

6.1 Base normale pour un corps fini "composé"

Lorsque $K = \mathrm{F}_{q^{n_1 n_2}}$ avec $\mathrm{pgcd}(n_1, n_2) = 1$, nous dirons que K est le composé de $\mathrm{F}_{q^{n_1}}$ et de $\mathrm{F}_{q^{n_2}}$ ([Alb56]).

Le théorème suivant permet de construire un élément normal pour K.

Théorème IV.16 ([BGM+93, p. 72, th 4.3]) *Soit $n = n_1 n_2$ avec $\mathrm{pgcd}(n_1, n_2) = 1$. Alors, pour $\alpha \in \mathrm{F}_{q^{n_1}}$ et $\beta \in \mathrm{F}_{q^{n_2}}$, $\gamma = \alpha\beta \in \mathrm{F}_{q^n}$ est un élément normal si et seulement si α et β sont des éléments normaux de $\mathrm{F}_{q^{n_1}}$ et $\mathrm{F}_{q^{n_2}}$ respectivement.*

Notons que ce théorème fournit de plus une *présentation* de F_{q^n}, par la donnée du polynôme minimal de $\gamma = \alpha\beta$ qui peut être calculé sans la donnée explicite de γ.

Soit P_1 le polynôme minimal polynôme de θ_1, et P_2 le polynôme minimal de θ_2 et $\tilde{P}_2(X, Y) = X^{n_2} P_2(\frac{Y}{X})$. Considérons le résultant suivant [Mig89, p. 137]

$$\begin{aligned} R(Y) &= Res_X(\tilde{P}_2(X, Y), P_1(X)) \\ &= \prod_{\beta, P_1(\beta)=0} \tilde{P}_2(\beta, Y). \end{aligned}$$

Alors $R(Y) = 0$ si et seulement si il existe β tel que

$$\begin{cases} P_1(\beta) = 0 \\ \tilde{P}_2(\beta, Y) = 0 \end{cases} \Leftrightarrow \begin{cases} P_1(\beta) = 0 \\ \beta^{n_2} P_2(\frac{Y}{\beta}) = 0 \end{cases},$$

c'est-à-dire $R(Y) = 0$ si et seulement si Y est le produit d'une racine de P_1 et d'une racine de P_2. Le polynôme $R(Y)$ a $n_1 n_2$ racines, qui est le nombre de conjugués de $\theta_1\theta_2$. Puisque $\theta = \theta_1\theta_2$ est une racine de $R(Y)$, le polynôme $R(Y)$ est le polynôme minimal de θ. D'autres méthodes peuvent être considérées pour obtenir le polynôme minimal de $\theta = \theta_1\theta_2$, par exemple calculer $1, \theta, \ldots, \theta^i, \ldots, \theta^{n_1 n_2}$ et obtenir une relation de dépendance linéaire.

6.2 Base normale dans le cas $n = p^e$

Pour calculer une base normale pour F_{p^n}, où $n = p^e n_1$ et $\mathrm{pgcd}(p, n_1) = 1$, il suffit donc de calculer de calculer une base normale pour $\mathrm{F}_{p^{n_1}}$ et pour $\mathrm{F}_{p^{p^e}}$. Soit donc $\mathbf{K} = \mathrm{F}_{p^{p^e}}$.

Le polynôme minimal de l'isomorphisme de Frobenius est alors $X^{p^e} - 1 = (X-1)^{p^e}$. Soit H une matrice de Hessenberg à décalage présentée comme dans la partie 5.3, alors le polynôme minimal de ϵ_m est $X^{p^e} - 1$. En effet, on constate d'abord que $X^{p^e} - 1$ est le polynôme minimal d'un des vecteurs ϵ_i : si le polynôme minimal de chaque ϵ_i est $X^{p^{e_i}} - 1$, $e_i < e$, alors le polynôme minimal de H est $X^{p^{\max(e_i)}} - 1 \neq X^{p^e} - 1$. Ensuite, si ϵ_j est cyclique avec $l < m$, on peut le placer en première position, par permutation de lignes et de colonnes.

Le procédure de réduction en matrice en de Hessenberg à décalage ferait apparaître une matrice compagnon, de polynôme minimal $X^{p^e} - 1$. Or cette procédure laisse inchangés les blocs B_l, $l > j$, et en particulier le terme nul, sur la sous-diagonale, précédent ϵ_m demeure. La matrice obtenue ne peut pas être une matrice compagnon, et le vecteur ϵ_m est donc un vecteur cyclique.

Proposition IV.3 *Une base normale de $\mathrm{F}_{q^{p^e}}$, où $q = p^m$, p premier, peut être calculé en $O((p^e)^3)$ opérations élémentaires dans F_q.*

7 Conclusion

La complexité du pire de la plupart de nos algorithmes est de l'ordre de $O(n^4)$, où n est la taille de la matrice, sauf pour deux problèmes : le problème particulier de la recherche d'un vecteur cyclique d'une matrice dont le polynôme caractéristique est sans facteurs multiples, pour lequel la complexité $O(n^3)$ est obtenue ; et pour le calcul récursif du polynôme minimal d'une matrice dont la factorisation du polynôme caractéristique est connue, dont la complexité peut être bornée par $O(n^{3.5})$. Toutefois, la complexité pratique est meilleure en moyenne sur les corps finis, et il nous semble important d'avoir montré que, pour certains problèmes d'algèbre linéaire, il existe des algorithmes dont la complexité dépend en réalité de certains paramètres de la matrice A traitée, précisément m_A le nombre de facteurs du polynôme caractéristique, comptés avec multiplicité.

En ce qui concerne le problème du calcul de la base normale, nous avons montré que le coût du calcul d'une base normale de $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ est le même que le coût de l'obtention d'une matrice représentant l'isomorphisme de Frobenius. Ceci est dû à deux techniques importantes dégagées ici.

La première technique est l'emploi de techniques "diviser-pour-régner" qui permettent de maintenir une complexité constante, malgré une construction récursive de la solution. Cette complexité est essentiellement la même que la complexité nécessaire pour scinder le problème et pour reconstruire la solution à partir des solutions partielles. Nous avons vu deux exemples où cette démarche est fructueuse.

La deuxième technique est l'emploi de matrices de Hessenberg à décalage, qui peuvent être vues comme des formes faibles de la forme rationnelle canonique. La forme de Hessenberg à décalage présente la qualité d'être creuse, dans le cas des corps finis : il y a $O(\log n)$ termes non nuls par lignes. Ceci diminue considérablement le coût des calculs, notamment les multiplications de matrices. Un autre intérêt de la forme de Hessenberg à décalage est de présenter des propriétés de structure fortes, qui permettent de construire la solution, en utilisant les matrices compagnons qui interviennent sur la diagonale. Ceci permet de transformer les calculs matriciels en opérations sur les polynômes. Enfin l'obtention d'une forme de Hessenberg à décalage n'est pas très coûteuse.

Le prolongement naturel est le suivant : existe-t'il un algorithme de bonne complexité (par exemple $O(n^3 m_H)$ ou mieux $O(n^2 m_H^2)$) pour calculer la forme rationnelle canonique d'une matrice de Hessenberg à décalage H, sans connaître la factorisation du polynôme caractéristique de H. Ceci permettrait de généraliser notre algorithme de calcul de la forme rationnelle canonique à des matrices sur $\mathbb{Q}$.

Enfin, le problème de toutes ces méthodes est qu'il est impossible de faire baisser la complexité en dessous de $O(n^3)$ pour une matrice quelconque, car le précalcul requis est le calcul d'une forme de Hessenberg à décalage, qui s'obtient en $O(n^3)$. Existe-t'il alors des versions probabilistes de ces méthodes, de meilleure complexité ?

Annexe A

Détermination de la distance minimale de deux codes BCH de longueur 255

1 Premier exemple: Le BCH de longueur 255 de distance construite 61

Cet exemple est facile et peut être traité directement sans faire intervenir de bases standards. Le principe est d'éliminer le plus grand nombre de variables possible dans les équations, en espérant qu'une contradiction apparaîtra plus facilement. Afin de pouvoir éliminer une variable, il peut être utile de se servir d'une première variable comme "pivot", à condition qu'elle soit non nulle.

Dans cet exemple, nous présentons d'abord le cas ou l'on suppose $A_{63} = 0$. A ce stade toutes les fonctions symétriques ont déjà été éliminées, et la première équation à traiter est eq_{123}.

eq 123

$$A_{111}{}^{8} + A_{87}$$

%**32

$$A_{111}{}^{256} + A_{87}{}^{32}$$

— *réduction des exposants sachant que chaque variable vérifie* $A_i^{256} = A_i$.

reduceExponents(%)

$$A_{111} + A_{87}{}^{32}$$

— *cela permet alors d'éliminer* A_{111}.

a.111:=%+a.111; eq 125

$$A_{95}{}^{4} + A_{87}{}^{8}$$

reduceExponents(%**64)

$$A_{95} + A_{87}{}^{2}$$

a.95:=%+a.95; eq 127

$$A_{127} + A_{91} + {A_{85}}^2$$

a.127:=%+a.127; eq 133

$${A_{85}}^3 + 1$$

— *en particulier* $A_{85} \neq 0$*, ce qui est utilisé dans l'équation suivante.*
eq 135

$${A_{85}}^2\ A_{87}$$

a.87:=0; eq 143

$${A_{91}}^{32}$$

a.91:=0; eq 151

$${A_{85}}^3$$

— *ceci est en contradiction avec l'équation* eq_{133}*.*

Le code BCH de distance construite 61 et de longueur 255 ne contient donc pas de mots de poids 61 tel que $A_{63} = 0$, nous pouvons reprendre l'étude, en certifiant que $A_{63} \neq 0$.
assertNonZeroA(63)

true

eq 123

$$\begin{aligned}&{A_{63}}^2\ A_{119} + {A_{111}}^8 + {A_{63}}^6\ A_{111} + {A_{63}}^{14}\ A_{95} + {A_{63}}^8\ {A_{91}}^{32} + A_{63}\ {A_{91}}^2 + {A_{63}}^{16}\ A_{91}\\ &+ {A_{63}}^5\ {A_{87}}^2 + \left({A_{63}}^{18} + 1\right)\ A_{87} + {A_{63}}^{31} + {A_{63}}^{22} + {A_{63}}^{13} + {A_{63}}^4\end{aligned}$$

— A_{63} *peut servir pour éliminer, puisque* $A_{63} \neq 0$*.*
eq(123)*A.63**(255-2)

$$\begin{aligned}&{A_{63}}^{255}\ A_{119} + {A_{63}}^{253}\ {A_{111}}^8 + {A_{63}}^{259}\ A_{111} + {A_{63}}^{267}\ A_{95} + {A_{63}}^{261}\ {A_{91}}^{32} + {A_{63}}^{254}\ {A_{91}}^2\\ &+ {A_{63}}^{269}\ A_{91} + {A_{63}}^{258}\ {A_{87}}^2 + \left({A_{63}}^{271} + {A_{63}}^{253}\right)\ A_{87} + {A_{63}}^{284} + {A_{63}}^{275} + {A_{63}}^{266} + {A_{63}}^{257}\end{aligned}$$

reduceExponents %

$$\begin{aligned}&A_{119} + {A_{63}}^{253}\ {A_{111}}^8 + {A_{63}}^4\ A_{111} + {A_{63}}^{12}\ A_{95} + {A_{63}}^6\ {A_{91}}^{32} + {A_{63}}^{254}\ {A_{91}}^2\\ &+ {A_{63}}^{14}\ A_{91} + {A_{63}}^3\ {A_{87}}^2 + \left({A_{63}}^{253} + {A_{63}}^{16}\right)\ A_{87} + {A_{63}}^{29} + {A_{63}}^{20} + {A_{63}}^{11} + {A_{63}}^2\end{aligned}$$

A.119:=%+A.119; eq 125

$${A_{95}}^4 + {A_{63}}^8\ {A_{91}}^4 + {A_{63}}^4\ {A_{87}}^{16} + {A_{87}}^8 + {A_{63}}^{16}\ {A_{87}}^4 + {A_{63}}^8\ {A_{85}}^2 + {A_{63}}^{20}\ A_{85} + {A_{63}}^{32}$$

reduceExponents (%**64)

$$A_{95} + {A_{63}}^2\ A_{91} + A_{63}\ {A_{87}}^4 + {A_{87}}^2 + {A_{63}}^4\ A_{87} + {A_{63}}^2\ {A_{85}}^2 + {A_{63}}^5\ A_{85} + {A_{63}}^8$$

A.95:=%+A.95; eq 127

$$A_{127} + {A_{63}}^3 {A_{91}}^2 + A_{91} + A_{63} {A_{87}}^8 + {A_{63}}^7 {A_{87}}^2 + {A_{63}}^2 A_{87} + \left({A_{63}}^9 + 1\right) {A_{85}}^2 + {A_{63}}^{15} + {A_{63}}^6$$

A.127:=%+A.127; eq 133

$$ {A_{85}}^3 + 1$$

eq 135

$$ {A_{85}}^2\ A_{87} + A_{63}$$

A.63:=%+A.63; eq 141

$$\left({A_{87}}^{255} + 1\right)\ {A_{111}}^8 + {A_{85}}^2\ A_{87}\ {A_{91}}^2 + A_{85}\ A_{87}\ A_{91}$$

eq 143

$$ {A_{91}}^{32}$$

A.91:=0; eq 141

$$\left({A_{87}}^{255} + 1\right)\ {A_{111}}^8$$

eq 147

$$\left(A_{85}\ {A_{87}}^{255} + A_{85}\right)\ {A_{111}}^8 + A_{111} + {A_{85}}^2\ {A_{87}}^{16}$$

A.111:=A.85**2*A.87**16

$$ {A_{85}}^2\ {A_{87}}^{16}$$

eq 151

$$1$$

Ceci établit donc la contradiction.

Proposition A.1 *La distance minimale du code BCH primitif au sens strict de distance construite* 61 *de longueur* 255 *est strictement supérieure à* 61*.*

Comme le BCH de longueur 255 de distance construite 63 a pour vraie distance minimale 63, on en déduit :

Proposition A.2 *La distance minimale du code BCH primitif au sens strict de distance construite* 61 *de longueur* 255 *est 63.*

2 Deuxième exemple: Le BCH de longueur 255 de distance construite 59

L'exemple qui suit présente un cas plus difficile d'obtention de contradiction. Les mêmes techniques d'élimination sont employées, mais des équations apparraissent qui ne se prêtent pas à ces techniques. Notre méthode consiste à accumuler ces équations, et une base standard de l'idéal I engendré par ces équations est calculée. Les équations suivantes sont ensuite regardées modulo cet idéal. Plus précisément le déroulement est le suivant.

> Les équations sont traitées les unes après les autres, par ordre croissant. Chaque équation est traitée en utilisant une technique d'élimination, si possible.
>
> Sinon, on considère l'idéal I engendré par les équations déjà considérées parmi $eq_1, \ldots, eq_{i-1}$, qui n'ont pas pu permettre de simplification de variables. Nous calculons G une base standard de I.
>
> Pour chaque équation eq_i rencontrée, nous ne considérons $eq'_i = eq_i \mathcal{R} I$. Si cette réduction permet de faire apparaître une élimination, celle-ci est effectuée. Les équations engendrant I sont modifiées en conséquence, et une base standard est à nouveau calculée.
>
> Sinon l'équation eq_i est ajoutée à l'ensemble des générateurs de I, et une nouvelle base standard est calculée.

Nous présentons l'exemple suivant. On considère le BCH au sens strict de longueur 255 et de distance construite 59. Afin de pouvoir éliminer, nous devons supposer une variable non nulle.

Nous nous assurons d'abord de la non-nullité de A_{61}. En effet, en supposant $A_{61} = 0$, nous obtenons facilement une contradiction, qui apparaît simplement.

Une fois ce point établi, nous pouvons reprendre l'étude, en certifiant que $A_{61} \neq 0$.
assertNonZeroA(61)

true

eq 119

$$A_{119} + \left({A_{63}}^2 + {A_{61}}^4\right) A_{111} + \left({A_{63}}^6 + {A_{61}}^8 {A_{63}}^2 + {A_{61}}^{12}\right) A_{95} {A_{61}}^6 {A_{91}}^{32} + {A_{61}}^{14} A_{91} + A_{63} {A_{87}}^2 + \left({A_{63}}^8 + {A_{61}}^8 {A_{63}}^4 + {A_{61}}^{12} {A_{63}}^2 + {A_{61}}^{16}\right) A_{87} {A_{61}}^4 {A_{85}}^2 + {A_{63}}^{15} + {A_{61}}^{64} {A_{63}}^{10} + {A_{61}}^{68} {A_{63}}^8 + {A_{61}}^{16} {A_{63}}^7 + {A_{61}}^{72} {A_{63}}^6 + {A_{61}}^{128} {A_{63}}^5 + {A_{63}}^4 + {A_{61}}^{24} {A_{63}}^3 \left({A_{61}}^{80} + {A_{61}}^4\right) {A_{63}}^2 + \left({A_{61}}^{136} + {A_{61}}^{28}\right) A_{63} + {A_{61}}^{192} + {A_{61}}^{84} + {A_{61}}^{30} + {A_{61}}^8$$

a.119:=%+a.119; eq 121

$${A_{61}}^4 \ {A_{91}}^4 + {A_{87}}^{16} + {A_{61}}^{12} {A_{87}}^4 + {A_{61}}^3 {A_{85}}^2 + \left({A_{63}}^8 + {A_{61}}^8 {A_{63}}^4 + {A_{61}}^{16}\right) \ A_{85} + {A_{61}}^{29}$$

— *nous pouvons utiliser la technique simple d'élimination, comme dans l'étude précédente.*

eq(121)*a.61**(255-4)

$$A_{61}{}^{255}A_{91}{}^{4} + A_{61}{}^{251}A_{87}{}^{16} + A_{61}{}^{263}A_{87}{}^{4} + A_{61}{}^{254}A_{85}{}^{2} + \left(A_{61}{}^{251}A_{63}{}^{8} + A_{61}{}^{259}A_{63}{}^{4} + A_{61}{}^{267}\right)A_{85} + A_{61}{}^{280}$$

ree (%**64)

$$A_{91} + A_{61}{}^{254}A_{87}{}^{4} + A_{61}{}^{2}A_{87} + A_{61}{}^{191}A_{85}{}^{2} + \left(A_{61}{}^{254}A_{63}{}^{2} + A_{61}A_{63} + A_{61}{}^{3}\right)A_{85} + A_{61}{}^{70}$$

a.91:=%+a.91;eq 123

$$A_{111}{}^{8} + A_{61}{}^{2}A_{63}{}^{2}A_{111} + \left(A_{61}{}^{2}A_{63}{}^{6} + A_{61}{}^{10}A_{63}{}^{2}\right)A_{95} + \left(A_{61}{}^{223}A_{63}{}^{4} + A_{61}{}^{227}A_{63}{}^{2}\right)A_{87}{}^{128} + \left(A_{61}{}^{64}A_{63}{}^{4} + A_{61}{}^{68}A_{63}{}^{2}\right)A_{87}{}^{32} + A_{61}{}^{253}A_{87}{}^{8} + \left(A_{61}{}^{254}A_{63}{}^{8} + A_{61}{}^{7}A_{63}{}^{4} + A_{61}{}^{11}A_{63}{}^{2}\right)A_{87}{}^{4} + A_{61}{}^{2}A_{63}A_{87}{}^{2} + \left(A_{61}{}^{2}A_{63}{}^{8} + A_{61}{}^{10}A_{63}{}^{4}\right)A_{87} + \left(A_{61}{}^{223}A_{63}{}^{68} + A_{61}{}^{227}A_{63}{}^{66} + A_{61}{}^{32}A_{63}{}^{36} + A_{61}{}^{36}A_{63}{}^{34} + A_{61}{}^{191}A_{63}{}^{8} + (A_{61}{}^{253} + A_{61}{}^{199} + A_{61}{}^{96})A_{63}{}^{4} + \left(A_{61}{}^{203} + A_{61}{}^{100} + A_{61}{}^{2}\right)A_{63}{}^{2} + A_{61}{}^{4}A_{63}\right)A_{85}{}^{2} + \left(A_{61}{}^{254}A_{63}{}^{10} + A_{61}A_{63}{}^{9} + A_{61}{}^{3}A_{63}{}^{8} + A_{61}{}^{7}A_{63}{}^{6} + A_{61}{}^{9}A_{63}{}^{5} + A_{61}{}^{247}A_{63}{}^{4} + A_{61}{}^{13}A_{63}{}^{3} + \left(A_{61}{}^{251} + A_{61}{}^{15}\right)A_{63}{}^{2} + A_{61}{}^{127}\right)A_{85} + A_{63}{}^{16} + A_{61}{}^{2}A_{63}{}^{15} + A_{61}{}^{66}A_{63}{}^{10} + \left(A_{61}{}^{70} + A_{61}{}^{16}\right)A_{63}{}^{8} + A_{61}{}^{18}A_{63}{}^{7} + A_{61}{}^{74}A_{63}{}^{6} + A_{61}{}^{130}A_{63}{}^{5} + \left(A_{61}{}^{200} + A_{61}{}^{78} + A_{61}{}^{24}\right)A_{63}{}^{4} + A_{61}{}^{26}A_{63}{}^{3} + \left(A_{61}{}^{204} + A_{61}{}^{28} + A_{61}{}^{6}\right)A_{63}{}^{2} + A_{61}{}^{138}A_{63} + A_{61}{}^{194}$$

— *l'équation* 123 *ne permet pas d'effectuer de simplifications, on considère* $I = (eq_{123})$. *La base standard est immédiate.*

G:=[eq 123];
reduceExponents(normalForm(eq 125,G))

$$A_{95} + A_{61}{}^{254}A_{63}A_{87}{}^{4} + A_{61}{}^{191}A_{87}{}^{2} + A_{63}{}^{2}A_{87} + \left(A_{61}{}^{254}A_{63}{}^{3} + A_{61}A_{63}{}^{2}\right)A_{85} + A_{61}{}^{64}A_{63}{}^{4} + A_{61}{}^{68}A_{63}{}^{2}$$

— *l'équation réduite permet d'effectuer une simplification. La base standard est à recalculer.*

a.95:=%+a.95; G:=[eq 123];
a.127:=eq(127)+a.127;G:=[eq 123];
reduceExponents(normalFormeq 129,G))

$$0$$

reduceExponents(normalFormeq 131,G))

$$0$$

reduceExponents(normalFormeq 133,G))

$$0$$

reduceExponents(normalForm(eq 135,G))

$$0$$

reduceExponents(normalForm(eq 137,G))

$$A_{85}{}^{3}$$

— *A_{85} est nul. La base standard est à recalculer.*
a.85:=0;G:=[eq 123];
reduceExponents(normalForm(eq 139, G))

$$A_{61}$$

— *ceci est une contradiction, puisque A_{61} est supposé non nul. Nous montrons à titre d'exemple, l'équation eq_{139} non réduite, pour montrer que la réduction a considérablement réduit l'équation. Notons que la réduction seule seule crée des polynômes gigantesques, et c'est la réduction + la réduction des exposants qui donne le résultat.*
eq 139

$$\begin{aligned}
&\left(A_{61}{}^{2}A_{63}{}^{3}+A_{61}{}^{6}A_{63}\right)A_{111}{}^{8}+\left(A_{61}{}^{4}A_{63}{}^{5}+A_{61}{}^{8}A_{63}{}^{3}\right)A_{111}+\left(A_{61}{}^{225}A_{63}{}^{7}+A_{61}{}^{233}A_{63}{}^{3}\right)A_{87}{}^{128}\\
&+\left(A_{61}{}^{66}A_{63}{}^{7}+A_{61}{}^{74}A_{63}{}^{3}\right)A_{87}{}^{32}+\left(A_{63}{}^{3}+A_{61}{}^{4}A_{63}\right)A_{87}{}^{8}\\
&+\Big(A_{61}A_{63}{}^{11}+A_{61}{}^{3}A_{63}{}^{10}+A_{61}{}^{5}A_{63}{}^{9}+A_{61}{}^{7}A_{63}{}^{8}+A_{61}{}^{9}A_{63}{}^{7}\\
&+A_{61}{}^{11}A_{63}{}^{6}+A_{61}{}^{15}A_{63}{}^{4}+A_{61}{}^{17}A_{63}{}^{3}\Big)\,A_{87}{}^{4}\\
&+\left(A_{61}{}^{195}A_{63}{}^{9}+A_{61}{}^{199}A_{63}{}^{7}+A_{61}{}^{203}A_{63}{}^{5}+A_{61}{}^{4}A_{63}{}^{4}+\,A_{61}{}^{207}A_{63}{}^{3}+A_{61}{}^{8}A_{63}{}^{2}\right)A_{87}{}^{2}\\
&+A_{61}{}^{2}A_{63}{}^{19}+A_{61}{}^{4}A_{63}{}^{18}+A_{61}{}^{6}A_{63}{}^{17}+A_{61}{}^{8}A_{63}{}^{16}+A_{61}{}^{18}A_{63}{}^{11}+A_{61}{}^{20}A_{63}{}^{10}+A_{61}{}^{22}A_{63}{}^{9}\\
&+\left(A_{61}{}^{132}+A_{61}{}^{24}\right)A_{63}{}^{8}+\left(A_{61}{}^{202}+A_{61}{}^{26}\right)A_{63}{}^{7}+\left(A_{61}{}^{136}+A_{61}{}^{28}\right)A_{63}{}^{6}+A_{61}{}^{8}A_{63}{}^{5}\\
&+\left(A_{61}{}^{140}+A_{61}{}^{32}\right)A_{63}{}^{4}+\left(A_{61}{}^{210}+A_{61}{}^{196}+A_{61}{}^{34}+A_{61}{}^{12}\right)A_{63}{}^{3}+A_{61}{}^{144}A_{63}{}^{2}\\
&+A_{61}{}^{200}A_{63}+A_{61}
\end{aligned}$$

Proposition A.3 *La distance minimale du code BCH primitif au sens strict de distance construite* 59 *de longueur* 255 *est strictement supérieure à* 59*.*

Jean Louis Dornstetter [Dor82] a exhibé au mot de poids 61 dans ce code, on en déduit donc :

Proposition A.4 *La distance minimale du code BCH primitif au sens strict de distance construite* 59 *de longueur* 255 *est* 61*.*

Annexe B

Caractéristique première en Axiom

Dans les chapitres précédents, nous avons montré comment le système algébrique défini par les identités de Newton permet d'étudier les mots de poids minimum des codes cycliques. Dans ces systèmes particuliers d'équations algébriques, beaucoup d'indéterminées apparaissent au degré 1, et peuvent être éliminées par *substitution*. Sur certains exemples, il m'est apparu que ces substitutions étaient irréalisables. La fonction incriminée était la fonction `eval`, qui effectue la substitution. En lisant le code source des librairies d'Axiom, c'est finalement la fonction `**`, qui est mise en défaut.

Ce bref exposé montre comment j'ai résolu le problème, en restant fidèle à la philosophie générale d'Axiom [JS92]. Il y a beaucoup de notions d'Axiom dans ce qui suit, et peu d'algorithmes, l'algorithme choisi pour élever à une puissance étant le même que la méthode des carrés successifs, à la différence prés qu'on décompose l'exposant en base p plutôt qu'en base 2, où p est la caractéristique.

1 Le problème et son explication

1.1 Un calcul infaisable en Axiom

Voici un simple fichier d'input, qui montre la nature du problème. Il s'agit d'un calcul qui prend trop de place en mémoire, et tue la session Axiom.

```
)clear all
)se message time on
macro x^y==x**y
pol:POLY PF 2:=(t*x*y^32+t^3*y +t^3*x^8+t*x^3+t*x)*z;
pol:=pol+t^2*x^3*y^96+t^4*x^2*y^65;
pol:=pol+(t^4*x^10+t^2*x^5+t^2*x^3+1)*y^64+t*x^2*y^33;
pol:=pol+(t*x^67+t^2*x^7)*y^32+t^3*x*y^2;
pol:=pol+(t^3*x^66+t^3*x^9+t^4*x^6+t*x^4+t*x^2+t^2*x)*y+t^3*x^74;
pol:=pol+t*x^69+t*x^67+t^4*x^14+t^2*x^7+t*x+1  -- I yam what I yam

tpol:=pol for i in 1..6 repeat tpol:=tpol**2
-- on obtient ainsi pol**64, rapidement.

pol**64  -- le calcul direct explose.
```

```
Unrecoverable error: Can't allocate.  Good-bye!.
```

Que s'est-il passé ? En dépit des apparences, le problème est facile à résoudre, comme le calcul par carrés successifs de `tpol` le montre. De plus la taille du résultat est la même que celle de `pol`, qui ne comporte que 28 monômes, il ne devrait pas y avoir de croissance en mémoire.

1.2 Pourquoi un tel comportement

En étudiant la librairie, on découvre que le domaine `POLY` est dans la catégorie `Monoid`, et que l'opération `**` est implantée par défaut, en utilisant la méthode des carrés successifs, décrite dans le package `RepeatedSquaring`. On pourrait penser que le problème est déjà résolu, et que c'est cette méthode qui est choisie, comme l'exponentiation faite à la main dans l'exemple. Tel n'est pas le cas, donc l'implantation de `**` choisie par Axiom n'est pas celle-là, et $**$ est être redéfinie dans la *add-chain* du domaine `POLY`.

Le code source nous montre que la première exponentiation de `**` concernant les polynômes est donnée dans `PolynomialRing` (d'abréviation `PR`), cette implantation écrasant l'implantation par défaut. Cette implantation est la suivante :

```
PolynomialRing(R:Ring,E:OrderedAbelianMonoid): T == C
 where
  T == FiniteAbelianMonoidRing(R,E) with
    --assertions
       if R has canonicalUnitNormal then canonicalUnitNormal
          ++ canonicalUnitNormal guarantees that the function
          ++ unitCanonical returns the same representative for all
          ++ associates of any particular element.
  C == FreeModule(R,E) add
    --representations
       Term:=  Record(k:E,c:R)
       Rep:=  List Term
  .
  .
  .
       if R has CommutativeRing then
         p ** nn  ==
            null p => 0
            zero? nn => 1
            one? nn => p
            p.rest = [] => [[nn * p.first.k, p.first.c ** nn]]
            binomThmExpt([p.first], p.rest, nn)
         binomThmExpt(x,y,nn) ==
               nn = 0 => 1$$
               ans,xn,yn: $
               bincoef: Integer
               powl: List($):= [x]
```

```
for i in 2..nn repeat powl:=[x * powl.first, :powl]
yn:=y; ans:=powl.first; i:=1; bincoef:=nn
for xn in powl.rest repeat
   ans:= bincoef * xn * yn + ans
   bincoef:= (nn-i) * bincoef quo (i+1);  i:= i+1
   -- last I and BINCOEF unused
   yn:= y * yn
ans + yn
```

Le fonction `**` utilise la fonction locale `binomThmExpt` : il s'agit d'un calcul direct, qui essaye de conserver la caractère creux des polynômes intervenant dans le calcul. Cette implantation judicieuse ne fournit toutefois pas le résultat que nous désirons, et engendre, dans le cas modulaire, des calculs aussi importants que sur les entiers. Or la linéarité de $x \mapsto x^p$ en caractéristique p montre que le résultat peut être obtenu.

2 La redéfinition de **

2.1 La méthode

Dans le cas d'un corps de caractéristique première, nous définissons un nouvelle implantation de `**`, proche de la méthode des carrés successifs, la décomposition se faisant en base p, plutôt qu'en base 2. Voici du pseudo-code pour décrire cet algorithme, en caractéristique p.

q**n==
si n < p alors calculer q^p avec la méthode des carrés successifs
sinon
calculer a,b tel que n=a*p+b (division euclidienne)
calculer **spécialement** $qp := q^p$
calculer $qp^a * q^b$ (appel récursif à `**`)

Dans le cas des polynômes sur un corps de caractéristique première, l'exponentiation à la puissance p est facile : on multiplie tous les exposants par p, et, si le corps n'est pas premier, on élève tous les coefficients à la puissance p.

Maintenant que nous savons quel algorithme employer, il faut décider où l'implanter, et c'est là que la difficulté se trouve, la philosophie d'Axiom étant de définir cette implantation au plus haut niveau de généralité.

2.2 Implantation conditionnelle

Le plus haut niveau de généralité est `PolynomialRing` (Il s'agit du premier domaine dans la hiérarchie des domaines au-dessus de `Polynomial` qui exporte `**`). De plus, cette implantation doit être conditionnelle sur la catégorie du domaine des coefficients des polynômes. Il faut donc définir cette implantation dans `PolynomialRing(R)` lorsque `R` est dans la bonne catégorie.

La librairie Axiom qui nous est livrée présente deux catégories relatives à la caractéristique : `NonZeroCharacteristic` et `FieldOfPrimeCharacteristic` :

`NonZeroCharacteristic`

Cette catégorie ne convient pas. En effet la propriété de linéarité $(x+y)^p = x^p + y^p$ est vraie seulement pour une caractéristique p première. Or cette catégorie contient par exemple $\mathrm{Z}/4\mathrm{Z}[X]$, et est donc trop grande pour notre objectif.

`FieldOfPrimeCharacteristic`

C'est la première catégorie dans la hiérarchie des catégories qui spécifie de posséder une caractéristique première. Ceci est partiellement convenable, car il existe des anneaux de caractéristique première qui ne sont pas des corps. Toutefois nous choisissons cette catégorie.

Le code de `**` dans `PR`, conditionnel sur la caractéristique, est simple à écrire.

```
if R has FieldOfPrimeCharacteristic then
     -- give a smart exponentiation
     timesp: E->E
     charact:=characteristic()$R
     timesp e == charact*e
     powtop: R->R
     powtop r == r**charact
     p **nn ==
        zero? p=> 0
        zero? nn => 1
        one? nn => p
        nn<charact =>
         expt(p,nn::PositiveInteger)$RepeatedSquaring($)
        division:= divide(nn,charact)
        zero? division.remainder  =>
         map(powtop, mapExponents(timesp,p))**division.quotient
        map(powtop,mapExponents(timesp,p))**(division.quotient) *
         expt(p,division.remainder::PositiveInteger)$RepeatedSquaring($)
else
  -- l'implantation existante
```

Ce code est effectivement choisi par Axiom, et fonctionne très bien avec les polynômes à une indéterminée, et l'exponentiation est plus rapide, dans le cas de la caractéristique 2.

2.3 Echec en plusieurs indéterminées

Malheureusement, ce code ne permet pas de résoudre la problème présenté au début, dans le cas de plusieurs indéterminées. En effet, le calcul explose à nouveau, sur l'exemple présenté au début. Or l'implantation de polynômes de `POLY` est récursive, chaque polynôme étant, soit une constante, soit un polynôme sur les polynômes. Le constructeur choisi pour implanter cette représentation récursive est `SparseMultivariatePolynomial`, et dont l'implantation de `**` est celle de `PolynomialRing`.

En étudiant de prés la localisation de `**` dans le cas des polynômes en plusieurs indéterminées, on découvre que `**` est d'abord implanté au niveau de SparseMultivariatePolynomial (d'abréviation `SMP`) comme suit :

```
SparseMultivariatePolynomial(R: Ring,VarSet: OrderedSet): C == T where
  pgcd ==> PolynomialGcdPackage(IndexedExponents VarSet,VarSet,R,$)
  C == PolynomialCategory(R,IndexedExponents(VarSet),VarSet)
  T == add
    --representations
      D := SparseUnivariatePolynomial($)
      VPoly:=  Record(v:VarSet,ts:D)
      Rep:=  Union(R,VPoly)
      .
      .
      p ** k  ==
         p case R => p::R ** k
         PSimp(p.v, p.ts ** k)
```

(PSimp est une fonction locale de réarrangement de variables)

L'implantation de ** est basée sur la nature récursive de POLY, et la fonction ** employée est de signature **: SUP($) $->$ SUP($).

La catégorie de SMP($) n'exporte pas FieldOfPrimeCharacteristic, même si le corps de base est un corps fini, ce qui est normal (ce serait faux de l'exporter). Donc l'implantation conditionnelle ajoutée à PolynomialRing ne sera pas utilisée. La catégorie de SMP exporte NonZeroCharacteristic, ce qui est trop faible.

Conclusion : la catégorie de POLY doit être changée. Mais nous ne pouvons pas affirmer que PolynomialCategory exporte conditionnellement FieldOfPrimeCharacteristic, et la propriété de linéarité que nous désirons employer est fausse dans le cas général NonZeroCharacteristic. Nous avons donc besoin d'une catégorie pour distinguer les anneaux qui sont de caractéristique première.

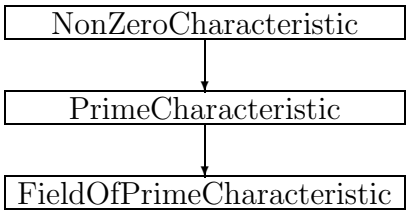
3 La catégorie PrimeCategory

3.1 Définition

C'est direct : il suffit de *nommer* la catégorie.

```
)abb category PCHAR PrimeCharacteristic
PrimeCharacteristic():Category == CharacteristicNonZero
```

Ce code, pour celui qui n'est pas habitué à Axiom, ne doit pas laisser penser que PrimeCharacteristic est la même catégorie que CharacteristicNonZero. En Axiom le nom des catégories est aussi important que ce qui est exporté (certains langages utilise la notion de catégorie *implicite*, définie simplement par les opérations exportées, le distinguo que nous faisons ne pourrait pas être codé dans de tels langages). Le code ci-dessus indique que tout domaine de la catégorie PrimeCharacteristic est dans la catégorie CharacteristicNonZero . La place de PrimeCharacteristic dans la hiérarchie des catégories est la suivante :



3.2 Quelles autres catégories doivent être modifiés

La catégorie est triviale à définir. Plus de travail est nécessaire pour déterminer quels domaines doivent être de cette catégorie.

Principalement, nous devons modifier les catégories suivantes, pour faire fonctionner notre exemple.

Les catégories des corps finis Nous avons besoin de quelques domaines de base qui appartiennent à la catégorie PrimeCharacteristic. Ces domaines sont les corps premiers, et les corps finis dans leurs implantations diverses (représentation polynomiale, cyclique, ou vectorielle par une base normale [GS]).

Dans le fichier `ffcat.spad`, seules quelques lignes sont à modifier.

- FieldOfPrimeCharacteristic : remplacer NonZeroCharacteristic par PrimeCharacteristic.
- ExtensionField(F) : Nous étendons conditionnellement la catégorie en ajoutant la ligne :

```
if F has PrimeCharacteristic then PrimeCharacteristic
```

AbelianMonoidRing C'est la catégorie de polynômes la plus sommaire spécifiant d'exporter `**` pour des polynômes. Toutes les autres catégories polynomiales étendent cette catégorie. Il suffit d'ajouter :

```
AbelianMonoidRing(R:Ring, E:OrderedAbelianMonoid): Category ==
        Join(Ring,BiModule(R,R)) with
        .
        .
        .
        if R has PrimeCharacteristic then PrimeCharacteristic
```

3.3 Quelles implantations doivent être changées

Nous avons vu comment redéfinir `**` pour le constructeur `PolynomialRing`. Ceci concerne à peu prés tous les domaines de polynômes. La principal problème, avec le compilateur livré actuellement, est que pour un changement apparemment mineur comme celui-ci, il faut recompiler tous les domaines susceptibles d'exporter `PrimeCharacteristic`, même si leur code n'a pas changé (les catégories qu'ils exportent ont juste été modifiés). La recompilation

nécessaire pour exécuter l'exemple présenté dans l'introduction prend quelques heures. A bon escient, car l'exemple fonctionne.

D'autres domaines susceptibles de voir leurs implantations modifiées sont `Expression`, `SimpleAlgebraicExtension`, `UnivariateTaylorSeries`, `UnivariateLaurentSeries`...

4 Un raffinement

La notion d'implantation par défaut permet de raffiner les choses, afin d'éviter de ré-écrire l'algorithme d'exponentiation successives pour tous les domaines de la catégorie.

On spécifie que `PrimeCharacteristic` exporte une opération, pour élever à la puissance p, en plus d'exporter `NonZeroCharacteristic`, et tous les domaines de cette catégorie, qui sont commutatifs, devront implanter cette opération.

Cette fonction nous permet d'implanter l'algorithme de calcul par élévation successives à la puissance p en fonction de cette opération, d'une manière générique. Ceci conduit à la définition suivante (le nom de la fonction est `powerToP`).

```
)abb category PCHAR PrimeCharacteristic
PrimeCharacteristic():Category == CharacteristicNonZero with
  if $ has CommutativeRing then
  -- on teste si le domaine lui-meme exporte CommutativeRing
   powerToP: $ -> $
   ++ powerToP(x) returns x**p, where p is the characteristic of $.
 add
  if $ has CommutativeRing then
   p:=characteristic()$$
   x:$ ** n:NonNegativeInteger==
       zero? x=> 0
       zero? n => 1
       one? n => x
       n<p =>
         expt(x,n::PositiveInteger)$RepeatedSquaring($)
       division:= divide(n,p)
       zero? division.remainder  =>
         powerToP(x)**division.quotient
       powerToP(x)**(division.quotient) *
         expt(x,division.remainder:: PositiveInteger)$RepeatedSquaring($)
```

Nous avons effectué cette définition et défini `powerToP` dans `PolynomialRing`. Cela fonctionne.

5 Conclusion

Ceci montre qu'en Axiom rien n'est simple, et que la difficulté ne se situe pas tant au niveau des algorithmes, qui en général sont connus, qu'au niveau de la description des objets mathématiques manipulés. Cela peut être très subtil, et le rapport [DT92] montre bien les difficultés de conception qui se présente. J. Davenport a qualifié la librairie d'Axiom, et la

hiérarchie des catégories, de “Bourbaki algorithmique”, ce qui souligne à la fois la rigueur du langage, mais aussi sa lourdeur.

S'agissant du problème d'élévation à de grandes puissances, il n'est pas sur que notre méthode soit la plus efficace pour de grandes caractéristiques, bien cette catégorie nous ait permet de continuer à travailler (c'est principalement la caractéristique 2 qui nous intéresse). Dans le cas de grandes caractéristiques, la méthode `binomThmExpt` est plus efficace. De plus ceci a des conséquences sur d'autres fonctions, notamment `eval`.

En réalité, il semble plus important d'être capable de décrire plus finement les catégories, et la librairie actuelle d'Axiom laisse un vide entre `NonZeroCharacteristic` et `FieldOfPrimeCharacteristic`.

Bibliographie

[ABMV93] G. B. Agnew, T. Beth, R.C. Mullin, and S.A. Vanstone. Arithmetic operations in $GF(2^m)$. *Journal of Cryptology*, 6:3–13, 1993.

[ABV89] D. W. Ash, I. F. Blake, and S. A. Vanstone. Low complexity normal bases. *Discrete Applied Mathematics*, pages 191–210, 1989.

[ACS90] D. Augot, P. Charpin, and N. Sendrier. Weights of some binary cyclic codes throughout the newton's identities. In G. Cohen and P. Charpin, editors, *Eurocode' 90*. Springer-Verlag, 1990.

[ACS91] D. Augot, P. Charpin, and N. Sendrier. Sur une classe de polynômes scindés de l'algèbre $F_{2^m}[Z]$. *CRAS*, 312:649–751, 1991.

[ACS92] D. Augot, P. Charpin, and N. Sendrier. Studying the locator polynomial of minimum weight codewords of BCH codes. *IEEE Transaction on Information Theory*, 38(3):960–973, 1992.

[AK92] E. F. Assmus and J. D. Key. *Designs and their Codes*. Cambridge University Press, 1992.

[Alb56] A. A. Albert. *Fundamental Concepts of Higher Algebra*. The University of Chicago Press, 1956.

[AS93a] D. Audibert and N. Sendrier. Distribution des poids des codes binaires cycliques de longueur 63, 1993. En préparation.

[AS93b] D. Augot and N. Sendrier. Idempotents and the BCH bound. *IEEE Transaction on Information Theory*, 1993. A paraitre.

[AV92] J-M Arnaudiès and A. Valibouze. Calculs de tables de longueurs d'orbites de résolvantes et de groupes de galois. manuscrit, 1992.

[Ber68] E.R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, 1968.

[Ber91] T. Berger. *Sur les groupes d'automorphismes des codes cycliques étendus primitifs affine-invariants*. PhD thesis, Université de Limoges, 1991.

[BGM91] T. Beth, W. Geiselmann, and F. Meyer. Finding (good) normal bases in finite fields. 1991.

[BGM+93] I.F. Blake, X.H. Gao, R.C. Mullin, S.A.Vanstone, and T. Yaghoobian. *Applications of finite fields*. Kluwer Academic Publishers, 1993.

[BMvT78] E.R. Berlekamp, R.J. MacEliece, and H.C.A. van Tilborg. On the inherent intractability of certain coding problem. *IEEE Transaction on Information Theory*, 24:384–386, May 1978.

[Bou81] N. Bourbaki. *Algèbre*. Masson, 1981.

[Bou84] N. Bourbaki. *Eléments d'Histoire des Mathématiques*. Masson, 1984.

[Car] C. Carlet. The divisors of $x^{2^m} + x$ of constant derivatives and degree $2^m - 2$. *SIAM Journal on Discrete Math.* à paraître.

[CCM92] P. Camion, B. Courteau, and A. Monpetit. Coset weight enumerators of the extremal self-dual binary codes of length 32. In P. Camion, P. Charpin, and S. Harari, editors, *Eurocode'92*, Lecture Notes in Computer Science. Springer-Verlag, 1992. à paraître.

[Cha87] P. Charpin. Codes cycliques affines invariants sous le groupe affine. Master's thesis, Université Paris VI, 1987.

[Cha90] P. Charpin. Codes cycliques étendus affines-invariants et antichaînes d'un ensemble partiellement ordonné. *Discrete Mathematics*, 80:229–247, 1990.

[CK91] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, pages 693–701, 1991.

[CU57] L. Carlitz and S. Uchiyama. Bounds for exponential sums. *Duke mathematical journal*, 24:37–41, 1957.

[Dor82] J-L. Dornstetter. Quelques résultats sur les codes BCH binaires en longueur 255. ENSTA stage report, Annex, July 1982.

[dRvL91] P.J.N. de Rooij and J.H. van Lint. More on the minimum distance of cyclic codes. *IEEE Transaction on Information Theory*, 37(1):187–189, January 1991.

[DT87] R. Dvornicich and C. Traverso. Newton symmetric functions and the arithmetic of algebraically closed fields. *Lecture Notes in Computer Science*, 356:216–224, June 1987.

[DT92] J. Davenport and B. Trager. Scratchpad's view of algebra (I) : Basic commutative algebra. Technical report, Nag Technical Reports, 1992.

[Eli87] R. J. Mac Eliece. *Finite Fields for the Computer Scientist and the Engineer*. Kluwer Academic Publisher, 1987.

[Fau93] Jean-Charles Faugère. *Résolution de systèmes d'équations algébriques avec GB*. PhD thesis, Université Paris VI, LITP, 1993. En préparation.

[FGLM] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero–dimensional grobner bases by change of ordering. *Journal of Symbolic Computation*. à paraître.

[Gan77] F. R. Gantmacher. *The Theory of Matrices*, volume 1. Chelsea, 1977.

[GCL92] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Academic Publishers, 1992.

[GHWL92] S. Gao and Jr H. W. Lenstra. Optimal normal bases. *Designs, Codes and Cryptography*, pages 315–323, 1992.

[Gir29] Albert Girard. Invention nouvelle en algèbre. *Amsterdam*, 1629.

[GMN+91] A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso. One sugar cube, please, or selection strategies in the Buchberger algorithm. In S. M. Watt, editor, *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*. ACM Press, 1991.

[GS] J. Grabmeier and B. Scheerhorn. Finite fields in axiom. Technical report, Nag Technical Reports.

[JS92] Richard D. Jenks and Robert S. Sutor. *Axiom, the Scientific Computation System*. Springer-Verlag, 1992.

[Kas69] T. Kasami. Weight distribution of Bose-Chaudhuri-Hocquenghem codes. In *Combinatorial Mathematic and its applications*, pages 335–357, 1969.

[KL72] T. Kasami and S. Lin. Some results on the minimum weight of primitive BCH codes. *IEEE Transaction on Information Theory*, pages 824–825, 1972.

[KLP67] T. Kasami, S. Lin, and W.W. Peterson. Some results on cyclic codes which are invariant under the affine group and their applications. *Information and Control*, 11:475–496, 1967.

[KLP68] T. Kasami, S. Lin, and W.W. Peterson. New generalisations of the Reed-Muller codes – Part I: Primitive codes. *IEEE Transaction on Information Theory*, 14(2):189–199, March 1968.

[Knu81] D. E. Knuth. *The Art of Computer Programming*, volume 2. Addison Wesley, 2 edition, 1981.

[Las86] A. Lascoux. Suites récurrentes linéaires. *Advances in Applied Mathematics*, 7:228–235, 1986.

[Laz93] Daniel Lazard. Systems of algebraic equations (algorithms and complexity). In *Proceedings of Cortona Conference*. University of Carolina Press, 1993.

[LDV92] F. Levy-Dit-Vehel. On duals of binary primitive BCH codes. In P. Camion, P. Charpin, and S. Harari, editors, *Eurocode'92*, Lecture Notes in Computer Science. Springer-Verlag, 1992. à paraître.

[Len91] H. W. Lenstra. Finding isomorphisms between finite fields. *Mathematics of Computation*, pages 329–347, 1991.

[Leo88] J.S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transaction on Information Theory*, 34(5):1354–1359, September 1988.

[LJ85] M. Lejeune-Jalabert. Effectivité de calculs polynomiaux. Cours de dea, Institut Fourier, 84-85.

[LN83] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1983.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In Christoph G. Gunther, editor, *Advances In Cryptology, Eurocrypt'88*, volume 330 of *Lecture Notes in Computer Science*, pages 419–453. Springer-Verlag, 1988.

[Mig89] M. Mignotte. *Mathématiques pour le calcul formel*. Presses Universitaires de France, 1989.

[MOVW89] R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone, and R.M. Wilson. Optimal normal bases in GF(p^n). *Discrete Applied Mathematics*, pages 149–161, 1988-1989.

[MS88] J. L. Massey and T. Schaub. Linear complexity in coding theory. *Lecture Notes in Computer Science*, (311), 1988.

[Oze87] Patrick Ozello. *Calcul exact des formes de Jordan et de Frobenius d'une matrice*. PhD thesis, Université Scientifique Technologique et Médicale de Grenoble, 1987.

[PJ86] W. W. Peterson and E. J. Weldon Jr. *Error-Correcting Codes*. MIT Press, 1986.

[Roo82] C. Roos. A generalization of the BCH bound for cyclic codes, including the hartmann-tzeng bound. *Journal of Combinatorial Theory*, 33:229–232, 1982.

[Roo83] C. Roos. A new lower bound for the minimum distance of a cyclic code. *IEEE Transaction on Information Theory*, 29(3):330–332, May 1983.

[Sch88] T. Schaub. *A Linear Complexity Approach to Cyclic Codes*. PhD thesis, Swiss Federal Institute of Technology, Zuerich, 1988.

[Sen91] N. Sendrier. *Codes Correcteurs à Haut Pouvoir de Correction*. PhD thesis, Université Paris VI, 1991.

[SSB89] M. Stillman, M. Stillman, and D. Bayer. *Macaulay User Manual*, 1989.

[Sto88] Richard Stong. Some asymptotic results on finite vector spaces. *Advances in Applied Mathematics*, 9:167–199, 1988.

[Val87] A. Valibouze. Fonctions symmétriques et changements de bases. *Lecture Notes in Computer Science*, 378:323–332, June 1987.

[vL82] J.H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, 1982.

[vLW86a] J.H. van Lint and R.M. Wilson. Binary cyclic codes generated by $m_1 m_7$. *IEEE Transaction on Information Theory*, 32(2):283, March 1986.

[vLW86b] J.H. van Lint and R.M. Wilson. On the minimum distance of cyclic codes. *IEEE Transactions on Information Theory*, IT-32(1):23–40, January 1986.

[vZGG90] J. von Zur Gathen and M. Giesbrecht. Constructing normal bases in finite fields. *Journal of Symbolic Computation*, 10:547–570, 1990.

[Wee90] D. Weeks. Embarrassingly parallel algorithms for algebraic number arithmetic — and some less trivial issues. In R. E. Zippel, editor, *Computer Algebra and Parallelism*, number 584 in Lecture Notes in Computer Science, pages 63–70, May 1990.

[Wil92] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford Science Publications, 1992.

[WS86] F.J. Mac Williams and N.J.A. Sloane. *The Theory of Error Correcting Codes*. North-Holland, 1986.

Index

Résumé : Nous étudions les mots de poids minimal des codes correcteurs d'erreurs cycliques. Les fonctions symétriques élémentaires et les fonctions puissances des localisateurs de ces mots vérifient les identités de Newton. Dans le premier chapitre celles-ci sont étudiées comme un système d'équations algébriques, dont les solutions sont étudiées par transformation de Fourier.

Dans le chapitre II, le lien est fait avec les codes correcteurs d'erreurs cycliques. Sur quelques exemples, il est montré comment étudier les mots de poids minimal sur la donnée d'une base standard de l'idéal engendré par les équations de Newton.

Dans le chapitre III, les relations de Newton sont utilisées d'un point de vue théorique, et des résultats sur les mots de poids minimal de certains codes BCH sont obtenus.

Ces calculs se placent dans le contexte de la théorie des corps finis. Dans le chapitre IV, un algorithme est développé pour calculer une base normale sur un corps fini. Un point de vue d'algèbre linéaire est choisi, et d'autres problèmes sont abordés (calcul du polynôme minimal, de la forme de Frobenius d'une matrice, lorsque la factorisation du polynôme caractéristique est connue).

Abstract : Minimum weight codewords of cyclic error-correcting codes are considered here. The elementary symmetric functions and the power-sum symmetric functions of the locators of these codewords are related by the Newton's identities. In the first chapter, they are viewed as a system of algebraic equations, whose solutions are studied by means of the Fourier transform.

In chapter II, the link with cyclic error-correcting codes is established. On some examples, it is shown how to study the minimum weight codewords on the data of a Grobner basis of the ideal generated by the Newton's identities.

In chapter III, the Newton's identities are considered from a theoretical point of view, and results about minimum weight codewords of a family of BCH codes are obtained.

These computations occurs in the context of finite fields. In chapter IV, an algorithm is constructed for computing a normal basis of a finite field. A point of view from linear algebra is chosen, and other problems are dealt with (the computation of the minimal polynomial, of the Frobenius form of a matrix, when the factorization of the characteristic polynomial is known).