The Minimal polynomials, characteristic subspaces, normal bases and the frobenius form

Paul Camion, Daniel Augot

▶ **To cite this version:**

Paul Camion, Daniel Augot. The Minimal polynomials, characteristic subspaces, normal bases and the frobenius form. [Research Report] RR-2006, INRIA. 1993. inria-00074666

HAL Id: inria-00074666
https://inria.hal.science/inria-00074666

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Minimal Polynomials, Characteristic Subspaces, Normal Bases and the Frobenius Form

Daniel AUGOT
Paul CAMION

N° 2006
Août 1993

PROGRAMME 2

Calcul symbolique,
programmation
et génie logiciel

The Minimal Polynomials, Characteristic Subspaces, Normal bases and the Frobenius Form

Polynômes minimaux, sous-espaces caractéristiques, bases normales et forme de Frobénius

Daniel AUGOT*, Paul CAMION†

July 27, 1993

Résumé

Divers algorithmes reliés au calcul du polynôme minimal d'une matrice carrée $n \times n$ sur un corps commutatif k sont exposés ici. Nous n'utilisons que l'arithmétique classique. La complexité n'est évaluée que pour $\mathrm{k} = \mathsf{F}_q$. Elle est exprimée en nombre d'opérations élémentaires dans k. La complexité du premier algorithme, pour lequel la factorisation du polynôme caractéristique est nécessaire, est de $O(\sqrt{n}n^3)$. L'algorithme fournit le polynôme minimal et tous les sous-espaces caractéristiques. On utilise la forme de Hessenberg à décalages connue des automaticiens et qui existe pour toute matrice. La complexité est alors réduite à $O(n^3 + m_{\mathbf{A}}^2 n^2)$, où $m_{\mathbf{A}}$ est un paramètre de la matrice $\mathbf{A}$ qui en général est petit. On présente de plus un algorithme itératif pour le polynôme minimal, qui a une complexité de $O(n^3 + n^2 m^2)$ où m est un paramètre lié à la matrice de Hessenberg à décalages utilisée. Il n'exige pas la connaissance du polynôme caractéristique. Un perfectionnement fournit un algorithme où jusqu' à m processeurs peuvent opérer indépendamment en parallèle. Le fait important est que la valeur moyenne de m ou $m_{\mathbf{A}}$ est $\approx \log n$. Ensuite nous nous intéressons à la construction d'un vecteur cyclique, d'abord pour une matrice dont le polynôme caractéristique est sans facteur carré. L'utilisation de la forme de Hessenberg à décalages permet d'obtenir un algorithme dont le coût est de $O(n^3 + m^2 n^2)$. Une méthode plus élaborée donne le résultat en $O(n^3)$ calculs élémentaires. En particulier, une base normale pour l'extension d'un corps fini sera obtenue de facon déterministe et aussi probabiliste, sur la donnée matricielle de l'opérateur de Frobenius, avec cette complexité. Finalement la forme de Frobenius est obtenue avec une complexité moyenne asymptotique de $O(n^3 \log n)$. Une retombée est l'obtention d'un vecteur cyclique pour une matrice quelconque. Tous les algorithmes sont déterministes. Dans tous les cas, la complexité obtenue est meilleure que pour les algorithmes déterministes connus à ce jour. La

*Paris 6, INRIA Domaine de Voluceau - 78153 Le Chesnay Cedex -France
†CNRS,INRIA

valeur asymptotique de l'espérance de m ou $m_{\mathbf{A}}$ est $\log n$. Les résultats sont repris dans les tables 1, 2, 3 et 4. L'étude des propriétés de base de la forme de Hessenberg à décalages aboutit à un algorithme qui construit tout élément du centralisateur dans $GL(n, \mathrm{k})$, ou peut-être dans un sous-groupe particulier de $GL(n, \mathrm{k})$, d'une matrice donnée quelconque. Cette étude du centralisateur nous conduit à étendre un résultat obtenu par R.Stong [20], ce qui nous permet d'établir nos évaluations de complexité et aussi de donner la formule explicite de la taille du centralisateur d'un opérateur linéaire, après calcul de ce qui est ici nommé forme de Frobénius développée.

Abstract

Various algorithms connected with the computation of the minimal polynomial of a square $n \times n$ matrix over a field k are presented here. Only classical arithmetic is used. The complexity is evaluated only for $\mathrm{k} = \mathbf{F}_q$. The complexity of the first algorithm, where the complete factorization of the characteristic polynomial is needed, is $O(\sqrt{n}n^3)$. It produces the minimal polynomial and all characteristic subspaces. Using the Shift-Hessenberg form, known to automation scientists and which exists for any matrix, the complexity of this algorithm is reduced to $O(n^3 + m_{\mathbf{A}}^2 n^2)$, where $m_{\mathbf{A}}$ is a parameter for the matrix $\mathbf{A}$, expected to be low. Furthermore an iterative algorithm for the minimal polynomial is presented with complexity $O(n^3 + n^2 m^2)$, where m is a parameter of the used Shift-Hessenberg matrix. It does not require knowledge of the characteristic polynomial. A refinement leads to an algorithm where up to m processes can be done independently in parallel. Important here is the fact that the average value of m or $m_{\mathbf{A}}$ is $\approx \log n$. Next we are concerned with the topic of finding a cyclic vector first for a matrix whose characteristic polynomial is square-free. Using the Shift-Hessenberg form leads to an algorithm at cost $O(n^3 + m^2 n^2)$. A more sophisticated recurrent procedure gives the result in $O(n^3)$ steps. In particular, a normal basis for an extended finite field will be obtained with that complexity with a deterministic algorithm and with a probabilistic algorithm as well on the data of a matrix representing the Frobenius operator. Finally the Frobenius form is obtained with asymptotic average complexity $O(n^3 \log n)$. As a by-product we there obtain a cyclic vector for any matrix. All algorithms are deterministic. In all four cases, the complexity obtained is better than for the heretofore best known deterministic algorithm. The asymptotic expected value of m or $m_{\mathbf{A}}$ is $\log n$. The results are summarized in Tables 1, 2, 3 and 4. Studying basic properties of the Shift-Hessenberg form leads to an algorithm to construct any element in $GL(n, \mathrm{k})$ or maybe in a particular subgroup of $GL(n, \mathrm{k})$, of the centralizer for any given matrix. That investigation into the centralizer lead us to extend a result obtained by R.Stong [20], which allows the needed complexity evaluations to be established and also the size of the centralizer of a linear operator to be given explicitly, for k a finite field, after computation of what is here called an Expanded-Frobenius form.

Keywords: characteristic polynomial, polynomial factorization, Hessenberg form, characteristic subspace, minimal polynomial, cyclic vector, finite filtration, normal basis, Frobenius form, elementary divisor, centralizer of a matrix.

1 Introduction

We present various low complexity algorithms for computing the objects in the title. The naïve algorithm for constructing the minimal polynomial of a matrix $\mathbf{A}$ consists in computing $I, \mathbf{A}, \mathbf{A}^2, \ldots, \mathbf{A}^n$ and then obtaining a non-trivial linear combination

$$\sum_{i=0}^{t} c_i \mathbf{A}^i = 0$$

with smallest possible t. The complexity is $O(n^4)$ with required memory size $O(n^4)$ as well. Significantly better algorithms for obtaining the minimal polynomial are probabilistic. They essentially consist in computing the minimal polynomial of $\mathbf{A}$ at random vectors with a good probability but no certainty that the minimal polynomial of $\mathbf{A}$ over the whole space is finally obtained. We observe that MAPLE preferred the naïve deterministic algorithm. If matrices submitted at computation were taken at random, such a probabilistic algorithm would be satisfactory since most characteristic polynomials of matrices over finite fields have few factors, which entails that many vectors are cyclic for such matrices. But this is not the case in the real world. For instance the characteristic polynomial of the Frobenius operator $\mathbf{F}$ of F_{q^n} over F_q is $X^n - 1$, and constructing a cyclic vector under $\mathbf{F}$ is precisely constructing a normal element, thus a normal basis. This is of particular interest in cryptology [1, 2, 3]. That topic is within the scope of the present paper. The recent results obtained by J. von zur Gathen and M. Giesbrecht [10] which are summarized in their introduction as follows:

' a fast algorithm in Section 2 for computing a normal basis of degree n over F_q, requiring an expected number $O^{\sim}(n^2 \log q)$ operations in F_q with fast arithmetic, and an expected number $O(n^3 \log q)$ operations in F_q with "naïve" arithmetic; this compares favourably with the previous known $O(n^{3.39} \log q)$ and "naïve" $O(n^3 \log q)$ operations in F_q respectively, based on linear algebra; '

Notice that the exact value of $O^{\sim}(n^2 \log q)$ is $O(M(n)(M(n) \log n + n \log q))$ where $O(M(n))$ is the cost, i.e. the number of operations in F_q for multiplying two polynomials of degree n with coefficients in F_q. Relying on algorithms for fast multiplication, the number $n \log n \log \log n$ is here taken for the value of $M(n)$. That is known to be beneficial only for huge values of n, i.e. more than one thousand.

An account of deterministic algorithms for the construction of normal basis is given in: Applications of finite fields, by Ian. F. Blake et al. [5, pp. 87-89]. The algorithm of H.W.Lenstra, published in 1991, is there described. The cost is of $(O((n^2 + \log q)(n \log q)^2)$ bit operations which is also the cost of the algorithms of E. Bach, J. Discoll and J. Shallit. In particular we here obtain a normal basis deterministically on the data of a presentation of the field F_{q^n} together with the matrix representing the Frobenius map in the given basis of F_{q^n} over F_q. The cost is in $O(n^3)$ operations in F_q for any n. To sum up, our deterministic algorithm in $O(n^3)$ for computing a normal basis compares favourably not only with previous deterministic algorithms but also with probabilistic algorithms when confining ourselves to classical arithmetic. Notice that a very simple probabilistic algorithm is derived from our processes which will give a normal basis with classical arithmetic in $O(n^3)$ steps. There is no difficulty in deriving vectors with given exponent as was done in [10] deterministically, and with complexity $O(n^3)$ as well. Regarding the minimal polynomial, Patrick Ozello [16]

gives a deterministic algorithm with asymptotic average complexity $O(n^3 \log n)$. That goal is here achieved with asymptotic average complexity reduced to $O(n^3)$.

Arnold Schönhage, in his encouragement to improve the deterministic algorithms for the minimal polynomial presented by the authors at Oberwolfach in February 1993 suggested we try to obtain a Sparse Hessenberg form for a matrix. We actually use the Shift-Hessenberg form met in the work of Patrick Ozello [16] but first introduced by automation scientists.

In Section 2, we recall Wilkinson's algorithm to compute the characteristic polynomial of a matrix with $O(n^3)$ elementary operations in F_q, using the Hessenberg form for a square matrix.

Sections 3, 4, 5 are concerned with the problem of obtaining the minimal polynomial, a different algorithm being presented in each section. In Section 3, we introduce an algorithm which produces the minimal polynomial and all characteristic subspaces at cost $O(\sqrt{n}n^3)$. In Section 4 the Shift-Hessenberg form is introduced. Any matrix is similar to a matrix with that form. Thanks to that form, the complexity of the algorithm of Section 3 is reduced to $O(n^3 + m_{\mathbf{A}}^2 n^2)$, where the number $m_{\mathbf{A}}$ is the size of a maximum increasing sequence of invariant subspaces of $\mathbf{A}$. Both algorithms use as data the matrix and its factorized characteristic polynomial. Notice that the best time bound for factoring a polynomial over F_q is $O(n^2 \log n \log\log n . \log q)$ using fast multiplication. If this cannot be reasonably considered in the context of this paper, we here only need that it can be done using classical arithmetic with $O(n^3 + n^2 \log q)$ operations in F_q and space for $O(n^2)$ elements of F_q [11, Section 8].

Our algorithms appeal to a recurrent "divide-and-conquer" procedure. The surprising fact is that the total complexity is the same as for the terminal stage.

In Section 5, using the Shift-Hessenberg form of a matrix, we obtain an iterative algorithm ending in the minimal polynomial in $O(n^3 + n^2 m^2)$ elementary operations over F_q. It does not need any knowledge of the characteristic polynomial. Even if we don't consider zero-characteristic fields here, attention is drawn to the particular interest of that algorithm for a zero-characteristic k since factoring the characteristic polynomial is more expensive for such a field. The number m is a parameter of the Shift-Hessenberg form, and we have that $m \leq m_{\mathbf{A}}$.

We next are concerned with the topic of finding a cyclic vector. Notice that the Frobenius form obtained in last section solves the problem of obtaining a cyclic vector in general. However we will obtain a cheaper algorithm for matrices whose characteristic polynomial is square-free. Under that assumption, the Shift-Hessenberg form leads to an algorithm of complexity $O(n^3 + m^2 n^2)$ presented in Section 6 and to a more sophisticated recurrent procedure with complexity $O(n^3)$ presented in Section 7. In Section 9, the Frobenius form is obtained with asymptotic complexity $O(n^3 \log n)$. That is better than for the algorithm of Patrick Ozello [16], which is implemented in MAPLE.

To sum up, the algorithms for the minimal polynomial have *asymptotic average* complexity $O(n^3)$ and the algorithm for a cyclic vector has complexity $O(n^3)$ for matrices with square-free characteristic polynomial. Special attention is given to cyclic vectors for the Frobenius map. Indeed those vectors yield normal basis which are of particular interest. We will show how to compute a normal basis for $\mathsf{F}_q^{p^t}$ deterministically in $O(n^3)$ elementary operations. Furthermore we recall how a normal basis for $\mathsf{F}_q^{n_1 n_2}$ is constructed from the data

Input	Complexity	Average complexity	Section
Factorisation of $C(X)$	$O(n^{3.5})$	$O(n^{3.5})$	Section 3
Factorisation of $C(X)$	$O(n^3 + n^2 m_A^2)$	$O(n^3)$	Section 4
matrix A	$O(n^3 + n^2 m_A^2)$	$O(n^3)$	Section 5

Table 1: Algorithms for the minimal polynomial

Input	Complexity	Average Complexity	Section
A	$O(n^3 + n^2 m_A^2)$	$O(n^3)$	Section 6
A	$O(n^3)$	$O(n^3)$	Section 7

Table 2: Algorithms for a a cyclic vector of a matrix whose characteristic polynomial is square-free

of a normal basis for $\mathbf{F}_q^{n_1}$ and another for $\mathbf{F}_q^{n_2}$, provided n_1 and n_2 are coprime.

This ends in an algorithm for computing a normal basis for $\mathbf{F}_q^n$ *deterministically* in $O(n^3)$ elementary operations, for *any* n.

We summarize our results in Tables 1, 2, 3 and 4.

Notice that the product of two $n \times n$ matrices over $\mathbf{F}_q$ can be computed with $O(n^{2.376})$ operations in $\mathbf{F}_q$ by the algorithm of Coppersmith & Winograd [7]. This reduces the complexity $O(n^{3.5})$, each time it occurs in the table, to $O(n^{2.876} + n^3)$.

Note 1 *For all computations, a* presentation *of* $\mathbf{F}_q$ *is assumed. It means that we are able to compute sums, products and inverses in* $\mathbf{F}_q$. *We are not concerned with the complexities of those computations, and our complexity measures are given in terms of elementary operations over* $\mathbf{F}_q$. *For instance the greatest common divisor of two polynomials of degree less than* n *can be computed in* $O(n^2)$ *steps, and this means* $O(n^2)$ *elementary operations. Thus our complexity is not the bit complexity of the problems.*

It is important to keep this remark in mind, since all algorithms presented here can be applied to matrices over any field $\mathbf{k}$ *and in particular over* $\mathbf{Q}$, *but we don't give any measure of the growth of intermediate rational numbers. Over finite fields, we can assume that all elementary operations are performed at constant time. However, if fields of characteristic zero are involved, we would first obtain the Hessenberg form through orthogonal transformations, like Householder transformations or rotations which are recommended in the book*

Complexity	Section
$O(n^3)$	6, 7

Table 3: Algorithm for a normal basis of $\mathbf{F}_{q^n}$

Input	Complexity	Average complexity	Section
A	$O(n^3 m_A)$	$O(n^3 \log n)$	8

Table 4: Algorithm for computing the Frobenius Form

of S.H. Wilkinson[21], then we would change the subdiagonal entries for ones thanks to a rational transformation also given in that book, and finally achieve the Shift-Hessenberg form for finally obtaining the Hessenberg form all pivots from there on being reduced to one.

Note 2 *All polynomials in this paper have their coefficients in* $\mathbf{k}$. *In particular "a polynomial p(X)" always means "a polynomial p(X) in* $\mathrm{k}[X]$*".*

2 Computing the characteristic polynomial

This section does not introduce any new result, but merely recalls how to compute the characteristic polynomial of a square matrix. The material can be found in [21] which is quoted in [12].

2.1 The Hessenberg form of a matrix

From [21], computing the characteristic polynomial of a $n \times n$ matrix is feasible in $O(n^3)$ elementary operations. The computation starts with a Hessenberg form of the matrix.

Definition 1 *A Hessenberg matrix* $\mathbf{H} \in M_n(\mathrm{k})$ *has the following form*

$$\begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,n} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,n} \\ 0 & 0 & h_{4,3} & \cdots & h_{4,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & h_{n,n-1} & h_{n,n} \end{bmatrix}$$

i.e. $\mathbf{H} = h_{i,j}$ *such that* $j < i - 1 \Rightarrow h_{i,j} = 0$.

Algorithm for computing the Hessenberg form of a matrix The following theorem holds:

Theorem 1 *For all* $\mathbf{A}$ *in* $M_n(\mathrm{k})$, *there exists a Hessenberg matrix* $\mathbf{H}$ *and a invertible matrix* $\mathbf{P}$ *such that* $\mathbf{H} = \mathbf{PAP}^{-1}$ *i.e. every matrix is similar to a Hessenberg matrix. The matrices* $\mathbf{H}$ *and* $\mathbf{P}$ *can be computed in* $O(n^3)$ *elementary operations.*

Proof: We prove the theorem by describing the algorithm.
Input $\mathbf{A} \in M_n(\mathrm{k})$, where k is a field.
Output $\mathbf{H}$ a Hessenberg form for $\mathbf{A}$, and $\mathbf{P}$ such that $\mathbf{H} = \mathbf{PAP}^{-1}$.

H:=**A**; **P**:=I_n; i:=1 {*ith row is denoted by* L_i, *ith column by* C_i}
while i < n **do** {*treat each column, do not treat the last one.*}
Search for the first non zero element in column i
starting at $i+1$. *If such an element exists, let* j *be the position of that entry*
if *no such element has been found* **then** i:=i+1 {*that column remains unchanged up to the end*}
else
with H:
swap rows $i+1$ and j
swap columns $i+1$ and j
with P: swap rows $i+1$ and j
pivot := 1/**H**[i+1,i]{*pivoting element*}
for l **from** i+1 **to** n **do**
c:= pivot∗**H**[l,i]
with H, $L_l \leftarrow L_l - c \times L_{i+1}$
with H, $C_{i+1} \leftarrow c \times C_l + C_{i+1}$
with P, $L_l \leftarrow L_l - c \times L_{i+1}$
i:=i+1
return(H,P)

□

2.2 Obtaining the characteristic polynomial from a Hessenberg form

Let us denote by $p_k(X)$ the characteristic polynomial of the diagonal submatrix of **H** extracted from the first k rows of **H**.

Computing the characteristic polynomial of **A** consists in computing $p_n(X)$. Observe that the polynomials $p_k(X)$ satisfy the following recurrence relations

$$\begin{aligned} p_k(X) = \; & (X - a_{k,k})p_{k-1}(X) - a_{k,k-1}(\\ & (X - a_{k-1,k})p_{k-2}(X) - a_{k-1,k-2}(\\ & (X - a_{k-2,k})p_{k-3}(X) - a_{k-2,k-3}(\\ & \dots \\ & (X - a_{3,k})p_2(X) - a_{3,2}(\\ & a_{2,k}p_1(X) - a_{2,1}a_{1,k}))) \dots) \end{aligned}$$

Computing $p_k(X)$ from $p_{k-1}(X), p_{k-2}(X), \dots, p_1(X)$ is done at cost $O(k^2)$. The total cost for $p_n(X)$ is $O(n^3)$.

3 Characteristic subspaces and minimal polynomial in $O(n^{3.5})$. Their construction.

In this section, an algorithm with complexity $O(n^3\sqrt{n})$ is presented for computing the minimal polynomial of a matrix **A**, and a block-diagonal matrix **D** similar to **A** and exhibiting its

characteristic subspaces. The inputs are $\mathbf{A}$ and the factored characteristic polynomial. The output are the minimal polynomial, the block-diagonal form $\mathbf{D}$ exhibiting the characteristic subspaces, and an invertible matrix $\mathbf{P}$ such that $\mathbf{D} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$.

3.1 Characteristic subspaces

We recall known facts about characteristic subspaces of a matrix $\mathbf{A}$. The reader may refer to [9].

Theorem 2 *Let $C(X)$ be the characteristic polynomial of a matrix $\mathbf{A} \in M_n(\mathbf{k})$, and assume that $C(X) = P(X)Q(X)$ where $P(X)$ and $Q(X)$ are relatively prime. Then the vector-space $\mathbf{k}^n$ splits as follows*

$$\mathbf{k}^n = V_P \oplus V_Q$$

$$V_P = \ker P(\mathbf{A}) \text{ and } V_Q = \ker Q(\mathbf{A})$$

Furthermore we can construct subspaces V_P and V_Q as follows

Theorem 3 *Let $C(X)$ be the characteristic polynomial of matrix $\mathbf{A}$, and assume $C(X) = P(X)Q(X)$ where $P(X)$ and $Q(X)$ are relatively prime. Let $V_P = \ker P(\mathbf{A})$ and $V_Q = \ker Q(\mathbf{A})$, then*

$$V_P = \text{Im } Q(\mathbf{A}) \text{ and } V_Q = \text{Im } P(\mathbf{A})$$

Definition 2 *Let $C(X)$ be the characteristic polynomial of matrix $\mathbf{A}$, and let $C(X) = f_1(X)^{r_1} \cdots f_k(X)^{r_k}$ be the factorization of $C(X)$ into irreducible polynomials. By definition, the characteristic subspaces of $\mathbf{A}$ are the invariant subspaces $V_i = \ker f_i(A)^{r_i}$, $i = 1, \ldots, k$.*

3.2 Overall strategy

The strategy of the algorithm is as follows. If the characteristic polynomial of $\mathbf{A}$ is $C(X) = p(X)^r$ where $p(X)$ is irreducible, then $\mathbf{k}^n$ is a characteristic subspace, and finding the minimal polynomial of $\mathbf{A}$ reduces to finding the minimal exponent s such that $p(\mathbf{A})^s = 0$.

If the characteristic polynomial is not a power of an irreducible polynomial, we are able to split $C(X)$ into $C(X) = P(X)Q(X)$ with $P(X)$ and $Q(X)$ relatively prime and either $P(X)$ or $Q(X)$ is of degree greater than $\frac{2}{3}n$ and is a power of an irreducible polynomial, or we have that $\deg P(X), \deg Q(X) \leq \frac{2}{3}n$. We recursively apply the procedure on both V_P and V_Q, given by Theorem 3. The new matrices are split in their turn, until all characteristic subspaces of $\mathbf{A}$ are obtained. Finally the minimal polynomial of the restriction of $\mathbf{A}$ to each of those subspaces is computed. The product of those polynomials gives the final result.

3.3 The algorithm

We now describe the algorithm more precisely.
Input Matrix $\mathbf{A}$ and its factored characteristic polynomial $C(X)$,

$$C(X) = f_1(X)^{r_1} \ldots f_k(X)^{r_k},$$

where $f_1(X), \ldots, f_k(X)$ are the irreducible factors of $C_{\mathbf{a}}(X)$.
Output The minimal polynomial of $\mathbf{A}$ and the splitting of $\mathbf{k}^n$ into all characteristic subspaces of $\mathbf{A}$.
Step 1: Find a splitting of $C(X) = P(X)Q(X)$ where $P(X)$ and $Q(X)$ are coprime. Three cases are considered.

- $C(X) = p(X)^r$, $p(X)$ irreducible. Compute the minimal polynomial $p(X)^s$ of $\mathbf{A}$ in $\lceil \log_2 r \rceil$ steps by trial and error on s. This is done with complexity $O(n^3\sqrt{n})$, using the algorithm presented in Section 3.5.

- One factor, $p_i(X)^{r_i}$, has degree larger than $\frac{2}{3}n$. Then $P(X) = p_i(X)^{r_i}$, i.e. $C(X) = p_i(X)^{r_i}Q(X)$, and $Q(\mathbf{A})$ gives a basis for a characteristic subspace.

- All factors $p_i(X)^{r_i}$ have degree $\leq \frac{2}{3}n$. Find a splitting $C(X) = P(X)Q(X)$ where $P(X)$ and $Q(X)$ are relatively prime and where $\deg P(X) \leq \frac{2}{3}n$, $\deg Q(X) \leq \frac{2}{3}n$. This is described in Lemma 2, which follows.

Step 2: Compute $Q(\mathbf{A})$, $P(\mathbf{A})$. This gives generating vectors for subspaces for V_P and V_Q respectively. It is seen in Subsection 3.5 that this is done at cost $O(n^3\sqrt{n})$.
Step 3: Compute bases for V_P and V_Q respectively. This is done with Gauss elimination, at cost $O(n^3)$.
Step 4: Change basis, taking for the new basis the union of the bases just computed, compute the matrices $\mathbf{A}_P$ and $\mathbf{A}_Q$ of the restriction of $\mathbf{A}$ to V_P and V_Q respectively. The cost is again $O(n^3)$.
Recursive Step Recursively apply the procedure to $\mathbf{A}_P$ and $\mathbf{A}_Q$, terminal steps end in basis for all characteristic subspaces by giving the diagonal blocks of $\mathbf{D}$.

Now two main operations are to be performed.

- The splitting. How to do this is detailed in next section.

- Evaluating polynomials $P(X)$ and $Q(X)$ at $\mathbf{A}$ with complexity $O(n^3\sqrt{n})$. This is detailed in Subsection 3.5.

3.4 Splitting the characteristic polynomial

3.4.1 A general procedure for a recursive partitioning

Definition 3 *A <u>multiset</u> is a mapping from* $\mathbf{E}$ *into* $\mathbf{N}$ *where* $\mathbf{E}$ *is a subset of* $\mathbf{N}$.

Thus a multiset yields a sequence of positive integers $n_{i_1}, \ldots, n_{i_k}, \ldots$ For the following definitions we assume that E is finite.

Definition 4 *A partition of* $\mathbf{E}$ *into two subsets I and J, which are the <u>classes</u> of the partition, yields two sequences $(n_i)_{i\in I}$ and $(n_j)_{j\in J}$. A class consisting in a single integer $\{i\}$ is called an <u>atom</u>.*

Definition 5 *Given a multiset, we denote by $n(E)$ the number $\sum_{i\in E} n_i$. A <u>θ-equitable partition</u> for $0 < \theta < 1$ is a partition of a multiset for which the partition of $E = I \bigcup J$ satisfies either*

- I *or* $J = \{i\}$ *and* $n_i > \theta n(E)$

  or

- $\sum_{i \in I} n_i \leq \theta n(E)$ *and* $\sum_{j \in J} n_j \leq \theta n(E)$

Definition 6 *A recursive θ-equitable partition is a θ-equitable partition recursively applied, treating each class of E in succession at each step, until every class is reduced to an atom.*

3.4.2 The problem $P(E)$

Assume that a multiset $S \rightarrow \mathbf{N}$ is given and that all finite multisets considered subsequently yield subsequences of that one. Given a submultiset $E \rightarrow \mathbf{N}$ of $S \rightarrow \mathbf{N}$ (i.e.$E \subset S$) yielding the sequence $(n_i)_{i \in E}$, we denote by $n(E)$ the number $\sum_{i \in E} n_i$. Problem $P(E)$ is a well defined problem on a multiset E and it is assumed that solving problem $P(E)$ reduces to solving problems $P(I)$ and $P(J)$ for any partition $I \bigcup J$ of E and that the cost of solving $P(E)$ is the sum of the costs of $P(I)$ and $P(J)$ augmented with an extra cost bounded from above by αn^e. Moreover the cost of solving problem $P(\{i\})$ at $\{i\}$ is assumed to be βn_i^e where e is the same positive real number as above. We denote by $C(E)$ the best possible cost of solving $P(E)$ for all possible recursive partitions of E. Finally $C(n)$ is the greatest of all $C(E)$ for subsets E such that $n(E) = n$. Consequently for any multiset $E \rightarrow \mathbf{N}$, there exists a recursive partition such that the cost of solving $P(E)$ is at most $C(n)$.

3.4.3 A general lemma

Lemma 1 *Consider $P(E)$ with $n(E) = n$ defined as above. Then, provided that $\theta \geq \frac{2}{3}$, there exists a recursive θ-equitable partition. If $\theta = \frac{2}{3}$, we have that $C(n) \leq \gamma n^e$ with $\gamma = \frac{\alpha+\beta}{1-2\theta^e}$, whenever e is at least $\frac{\log 2}{\log 3 - \log 2} \simeq 1.71$.*

Proof: We first prove that there exists a recursive θ-equitable partition provided that θ is at least $\frac{2}{3}$. Next we show that assuming that there exists a recursive θ-equitable partition then the whole thesis holds. We thus have to show that given any multiset with associated sequence $n_1, n_2, \ldots, n_k$ with $n(E) = n$, then there exists a θ-equitable partition provided $\theta \geq \frac{2}{3}$. From the definition of a θ-equitable partition we can assume that $n_i \leq \theta n, i = 1, \ldots, k$. Now if there is an i such that $n_i > (1-\theta)n$, then the partition $I = \{i\}$ and $J = E \setminus I$ is θ-equitable. We thus assume that $n_i \leq (1-\theta)n, i = 1, \ldots, k$. Let J be the maximal sized subset of E such that $\sum_{j \in J} n_j \leq \theta n$. We show that $\sum_{i \in I} n_i \leq \theta n$, for $I = E \setminus J$. Else adding any element l from I to J, I becoming I', we would have that

$$(1-\theta)n \geq \sum_{i \in I'} n_i > \theta n - (1-\theta)n = (2\theta - 1)n$$

That would entail $2\theta - 1 < 1 - \theta$ which contradicts the hypothesis. Next, under the stated assumption for problem $P(E)$, and with $\theta = \frac{2}{3}$, we show that $C(n)$ is bounded from above by γn^e. The proof is by recurrence on n. The thesis holds for $n = 2$ and, for $n > 2$ we have

that $\theta n \leq n-1$. The following inequalities hold.

$$\begin{aligned} C(n) &\leq \alpha n^e + \max(C((1-\theta)n) + \beta n^e, 2C(\theta n)) \\ &\leq \alpha n^e + \beta n^e + 2C(\theta) \\ &\leq (\alpha+\beta)n^e + 2\gamma\theta^e n^e. \end{aligned}$$

We thus have that $C(n) \leq \gamma n^e$ with $\gamma = \frac{\alpha+\beta}{1-2\theta^e}$, which is positive and finite provided that $e > \frac{\log 2}{\log 3 - \log 2}$. □

3.4.4 Applying the partitioning procedure to the characteristic polynomial

For clarity and with a view toward applying the previous lemma in an algorithm, we state it again since it will be used to show how the proof leads to an algorithm.

Lemma 2 *Let $n = n_1 + n_2 + \ldots + n_k, n_i > 0, i = 1, \ldots, k$, where $n_i \leq \frac{2}{3}n$.*
Then there exists a partition $[1,k] = I \cup J$ such that

$$\sum_{i \in I} n_i \leq \frac{2}{3}n \text{ and } \sum_{j \in J} n_j \leq \frac{2}{3}n.$$

Proof: For every subset J of $[1,k]$, denote by S_J the sum $\sum_{j \in J} n_j$. If there exists $n_i > \frac{n}{3}$, then choose $I = \{i\}$, and $J = [1,k] \setminus I$.

Otherwise choose J as the subset of $[1,k]$ of maximal size such that que $S_J \leq \frac{2}{3}n$. Then $I = [1,k] \setminus J$ necessarily satisfies $S_I \leq \frac{2}{3}n$. Indeed, if $S_I > \frac{2}{3}n$, let I' be the subset of $[1,k]$ obtained by removing any element of I the size of which, by hypothesis, being at most $\frac{n}{3}$. Then $S_{I'} > \frac{2}{3}n - \frac{n}{3} = \frac{n}{3}$, and the complementary J' of I' in $[1,k]$ satisfies $S_{J'} \leq \frac{2}{3}n$ and contains J. This contradicts the maximality of J. □

From an algorithmic point of view, this splitting can be obtained by sorting the integers $n_1, n_2, \ldots, n_k$, then adding them in increasing order until a value $n_1 + n_2 + \cdots n_{t+1}$ greater than $\frac{2}{3}n$ is found. Then take $I = \{1, 2, \ldots, t\}$.

Nicolas Sendrier suggested that a Huffman algorithm presented for example in [8, pp. 75-82] and used in source coding would probably provide a convenient binary tree describing the successive bipartitions. It can indeed be shown a that Huffman tree works, i.e. it gives a recursive θ-equitable partition for $\theta = \frac{2}{3}$. Yet it is not necessarily better than the one obtained by the algorithm described here. For let the set of integers be $\{1,2,3,4,5,6\}$. The first algorithm gives the weighted binary tree

$$21 = \{10 = \{6 = \{3 = \{\{1\},\{2\}\},\{3\}\},\{4\}\}, 11 = \{\{5\},\{6\}\}\}$$

and the Huffman algorithm gives

$$21 = \{12 = \{6 = \{3 = \{\{1\},\{2\}\},\{3\}\},\{6\}\}, 9 = \{\{4\},\{5\}\}\}.$$

Nearly all classes correspond with equal sizes to each other except two of respective sizes 12 and 9 for the Huffman tree and sizes 11 and 10 for the other.

3.5 Computing $P(\mathbf{A})$, $Q(\mathbf{A})$

We now show how $p(\mathbf{A})$ can be computed at cost $\sqrt{t}n^3$, where t is the degree of $p(X)$. A naïve Horner algorithm would lead to $O(tn^3)$.

This is a variant of Shank's procedure "baby step, giant step", and it needs to keep $\sqrt{t}$ matrices in a table.

Theorem 4 *For all* $\mathbf{A}$ *in* $M_n(\mathrm{k})$, *for all* $p(X)$ *with* $\deg p(X)$ *at most* t, *we have that* $p(\mathbf{A})$ *can be computed with complexity* $O(\sqrt{t}n^3)$, *the size of memory space being* $O(\sqrt{t}n^3)$.

Proof: For the sake of simplicity, we describe the algorithm in case $t = d^2 - 1$, for some integer d. We have to evaluate

$$U(\mathbf{A}) = u_0 + u_1\mathbf{A} + u_2\mathbf{A}^2 + \ldots + u_t\mathbf{A}^t. \tag{1}$$

Let $\mathbf{B} = \mathbf{A}^d$, we split the polynomial U into polynomials of size d

$$\begin{aligned} U(\mathbf{A}) &= u_0 + u_1\mathbf{A} + u_2\mathbf{A}^2 + \ldots + u_{d-1}\mathbf{A}^{d-1} \\ &\quad +(u_d + u_{d+1}\mathbf{A} + u_{d+2}\mathbf{A}^2 + \ldots + u_{d+d-1}\mathbf{A}^{d-1})\mathbf{B} \\ &\quad +(u_{2d} + u_{2d+1}\mathbf{A} + u_{2d+2}\mathbf{A}^2 \ldots + u_{2d+d-1}\mathbf{A}^{d-1})\mathbf{B}^2 \\ &\quad \ldots \\ &\quad +(u_{d(d-1)} + u_{d(d-1)+1}\mathbf{A} + \ldots + u_{d(d-1)+(d-1)}\mathbf{A}^{d-1})\mathbf{B}^{d-1} \\ &= U_0(\mathbf{A}) + U_1(\mathbf{A})\mathbf{B} + U_2(\mathbf{A})\mathbf{B}\cdots + U_{d-1}\mathbf{B}^{d-1} \end{aligned}$$

Precomputation is performed to store the following matrices

$\mathbf{A}$	$\mathbf{A}^2$	$\mathbf{A}^3$	...	$\mathbf{A}^{d-1}$	$\mathbf{B} = \mathbf{A}^d$	$\mathbf{B}^2$	...	$\mathbf{B}^{d-1}$

The cost of these precomputation is $2d-3$ matrix multiplications. Computing each $U_i(\mathbf{A})$ does not require any matrix multiplications, since each $\mathbf{A}^i, 0 \leq i \leq d-1$, is in the table. Then each $U_i(\mathbf{A})\mathbf{B}^i$ is left to be computed, this leads to d extra matrix multiplications.

Hence the total cost is $O(dn^3)$. □

3.6 The complexity

Theorem 5 *Using the previous algorithm, it is possible to compute the minimal polynomial of any square matrix over a finite field* k *and a block-diagonal matrix similar to* $\mathbf{A}$ *exhibiting its characteristic subspaces with time complexity* $O(n^3\sqrt{n})$, *and memory size* $O(n^3\sqrt{n})$.

The theorem is proved by making $e = 3.5$ in Lemma 3.4.3.

The result does not hold as it is stated for any field $\mathbf{k}$ because the bit-complexity of elementary arithmetic operations and the cost of factoring the characteristic polynomial cannot be evaluated in general.

4 The Shift-Hessenberg form and the centralizer of a matrix

We now use the same algorithm on a particular form of the Hessenberg matrix, which will be called the Shift-Hessenberg form. The main point is that evaluating a polynomial at a matrix is less expensive when that matrix has the Shift-Hessenberg form. The average improvement is, as will be seen, considerable. Before going to the use of the Shift-Hessenberg form for our algorithmic purposes, we show how Shift-Hessenberg forms shed light on the subgroup of $GL(n, \mathbf{k})$ commuting with a given fixed linear operator on $\mathbf{k}^n$. In fact, properties arising from our investigation lead to an algorithm to actually construct any matrix commuting with a given matrix. We write "operator" for linear operator $\mathbf{T}$ and use the notation $\mathbf{T}$ for the matrix representing $\mathbf{T}$ in the canonical basis of $\mathbf{k}^n$.

4.1 Shift-basis

Definition 7 *For* $\mathbf{A}$ *in* $M_n(\mathbf{k})$ *and* v *in* $\mathbf{k}^n$, *the minimal polynomial of* $\mathbf{A}$ *restricted to* v *is the lowest degree monic polynomial* $\pi_v(X)$ *such that* $\pi_v(\mathbf{A})v = 0$.

Notice that $\pi_v(X) \mid \pi(X)$.

Definition 8 *Let* $\mathbf{T}$ *be an operator on* $\mathbf{k}^n$. *A shift-basis for* $\mathbf{T}$ *is a basis which has the form*

$$\left[v_1, \mathbf{T}v_1, \ldots, \mathbf{T}^{n_1-1}v_1, v_2, \mathbf{T}v_2, \ldots, \mathbf{T}^{n_2-1}v_2, \ldots, v_m, \mathbf{T}v_m, \ldots, \mathbf{T}^{n_m-1}v_m\right] \qquad (2)$$

It is understood that a shift-basis is actually an *ordered basis.* Given $\mathbf{T}$, a shift-basis for $\mathbf{T}$ can be obtained as follows. First select any v_1, and introduce the linear independent set $\mathbf{T}^i v_1$ for $i = 0, \ldots, n_1 - 1$ where n_1 is the smallest value of i for which $\mathbf{T}^i v_1$ linearly depends on all previous vectors. Then select v_2 independent of the previous vectors up to v_1 and proceed with $\mathbf{T}^i v_2$, $i = 0, \ldots, n_2 - 1$ as for v_1. The process ends in a shift-basis with $n_1 + n_2 + \ldots + n_m = n$

Definition 9 *We call a matrix which represents an operator* $\mathbf{T}$ *in a shift-basis a Shift-Hessenberg matrix.*

It is important to observe that the Frobenius form of a matrix, also known as the Rational Canonical Form, is a particular Shift-Hessenberg form. Notice that if $\mathbf{T}$ is the zero operator, then any basis of $\mathbf{k}^n$ is a shift-basis for $\mathbf{T}$. The other extreme situation is when the characteristic polynomial of $\mathbf{T}$ is irreducible: we have that $m = 1$ for whatever v_1.
Clearly, to every shift-basis there corresponds an increasing sequence $\mathbf{V}_1, \ldots, \mathbf{V}_m$ of invariant subspaces of $\mathbf{T}$. We have that $\mathbf{V}_i$ is a $\mathbf{k}[\mathbf{T}]$-module, $i = 1, \ldots, m$ and consequently $\mathbf{V}_i/\mathbf{V}_{i-1}$ is a module . Such a sequence of modules is known as a *finite filtration* (see for example Serge Lang [13, page 126]). In the present particular situation, we have that each of those quotient modules is generated by a single element $\overline{v}_i$ which is the class of v_i in $\mathbf{V}_i/\mathbf{V}_{i-1}$. Denote by $f_i(X)$ the minimal polynomial of $\overline{v}_i$, and observe that the *ith* diagonal block in the Shift-Hessenberg matrix is the companion matrix of $f_i(X)$. Notice incidentally that it

can be seen that $h(\mathbf{T})\overline{v}_i$ generates the k[T]-module $\mathbf{V}_i/\mathbf{V}_{i-1}$ as well as $\overline{v}_i$ if and only if $h(X)$ is prime to $f_i(X)$. Since the annihilating ideal of $\mathbf{V}_i/\mathbf{V}_{i-1}$ contains the one of $\mathbf{V}_i$ and since $f_i(v_i)$ is in $\mathbf{V}_{i-1}$ then $f_i(X)$ divides the minimal polynomial of v_i. We just have pointed out a structure induced by any shift-basis which leads to a converse statement. If $\mathbf{V}_1, \ldots, \mathbf{V}_m$ is a finite filtration of k[T]-modules such that $\mathbf{V}_i/\mathbf{V}_{i-1}$ is a module generated by the single element $\overline{v}_i, i = 1, \ldots, m$, then $v_1, v_2, \ldots, v_m$ yield a shift-basis for $\mathbf{T}$.

As we will next see, a unique Shift-Hessenberg matrix will represent the operator $\mathbf{T}$ in diverse shift-basis obtained from the same increasing sequence $\mathbf{V}_1, \ldots, \mathbf{V}_m$. There exists a partition of all shift-bases with respect to the Shift-Hessenberg form to which they correspond. We will see that each class yields a unique subgroup af $GL(n, \mathbf{k})$ that will be investigated. The following property clearly follows from the above definitions.

Property 1 *Given any two shift-bases* $\mathbf{B}_1$ *and* $\mathbf{B}_2$*:*

$$\left[v_1, \mathbf{T}v_1, \ldots, \mathbf{T}^{n_1-1}v_1, v_2, \mathbf{T}v_2, \ldots, \mathbf{T}^{n_2-1}v_2, \ldots v_m, \mathbf{T}v_m, \ldots, \mathbf{T}^{n_m-1}v_m\right] \tag{3}$$

$$\left[v'_1, \mathbf{T}v'_1, \ldots, \mathbf{T}^{n_1-1}v'_1, v'_2, \mathbf{T}v'_2, \ldots, \mathbf{T}^{n_2-1}v'_2, \ldots v'_m, \mathbf{T}v'_m, \ldots, \mathbf{T}^{n_m-1}v_m\right] \tag{4}$$

such that

$$[v_1, v'_1 \in V_1 = \mathbf{T}V_1; v_2, v'_2 \in V_2 \setminus V_1, V_2 = \mathbf{T}V_2; v_m, v'_m \in V_m \setminus V_{m-1}, V_m = TV_m], \tag{5}$$

with

$$V_1 \subset V_2 \subset \ldots \subset V_m. \tag{6}$$

If $\mathbf{T}^{n_i}v_i$ *depends on the preceding vectors with the same coefficients as* $\mathbf{T}^{n_i}v'_i$, $i = 1, \ldots, m$ *respectively, then to* $\mathbf{B}_1$ *and* $\mathbf{B}_2$ *there corresponds a unique Shift-Hessenberg form.*

Here we use the same notation for a basis $\mathbf{B}$ and for the matrix whose columns are formed with the elements of $\mathbf{B}$ represented in the canonical basis of $\mathbf{k}^n$. We now start with an example illustrating Property 1 to introduce a construction for the group of matrices commuting with a given matrix.

Example: Consider the extension F_{16} over F_2. For this example, we thus have $n = 4$. Let $\mathbf{T}$ be the Frobenius map represented by the matrix $\mathbf{F}$ in the basis $\mathbf{B}_0 = \{1, X, X^2, X^3\}$, the presentation of $\mathbf{T}$ being given by the irreducible polynomial $1 + X + X^4$. A natural Shift-Hessenberg matrix similar to $\mathbf{F}$ is the Rational Canonical Form (Frobenius form) of $\mathbf{F}$ which here is the permutation matrix $\mathbf{P}$

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

whose minimal polynomial is $\mathbf{Z}^n - 1$, $n = 4$. Now let $\mathbf{B}_1$ be the matrix whose columns are the successive elements $\{X^3, X^2 + X^3, 1 + X + X^2 + X^3, X + X^3\}$ of a normal basis expressed in the basis $\mathbf{B}_0$. Then $\mathbf{B}_1$ is a shift-basis for $\mathbf{T}$ in which $\mathbf{T}$ is represented by the Shift-Hessenberg

form $\mathbf{P}$. Here $v_1 = X^3$ and $m = 1$. Taking $v'_1 = 1 + X^3$, then Property 1 applies, since $1 + X^3$ generates another normal basis whose vectors will form the columns of $\mathbf{B'}_1$. We thus have that $\mathbf{B}_1^{-1}\mathbf{F}\mathbf{B}_1 = \mathbf{P} = \mathbf{B'}_1^{-1}\mathbf{F}\mathbf{B'}_1$ which shows that $\mathbf{G} = \mathbf{B}_1\mathbf{B'}_1^{-1}$ commutes with $\mathbf{F}$. The integer n being a power of 2, there are exactly $2^n - 2^{n-1} = 8$ polynomials of degree less than 4 and relatively prime to $\mathbf{Z}^n - 1$ in $\mathbf{F}_2[\mathbf{Z}]$. There are consequently 8 cyclic vectors such as v_1. As above, 8 distinct matrices which commute with $\mathbf{F}$ are obtained. After computation we observe that they form the abelian group $\mathbf{F}, \mathbf{F^2}, \mathbf{F^3}, \mathbf{I}$, together with the coset containing $\mathbf{G}$. We notice that $\mathbf{F}$ is the cube of $\mathbf{B}_1\mathbf{P}^{-1}\mathbf{B}_1^{-1}$. We will soon prove Theorem 6 which shows that this is the whole centralizer of $\mathbf{F}$ in $GL(n, 2)$ for $n = 4$. Its proof moreover gives an algorithm for constructing any matrix in the centralizer of a given matrix $\mathbf{T}$.

4.2 An algorithm for the centralizer of a matrix

We here give an algorithm for constructing any element to be selected in the centralizer $\mathcal{Z}(\mathbf{T})$ of a matrix $\mathbf{T}$ over a field $\mathbf{k}$ and for enumerating $\mathcal{Z}(\mathbf{T})$ in the case where $\mathbf{k} = \mathbf{F}_q$

4.3 The k[X]-module induced by a matrix

Definition 10 *The Expanded-Frobenius form of* $\mathbf{T}$ *in* $M_n(\mathbf{k})$ *is the following matrix* $\mathbf{D}$ *similar to* $\mathbf{T}$,

$$\mathbf{D} = \begin{bmatrix} \mathbf{F}_{B_1,B_1} & 0 & \cdots & 0 \\ 0 & \mathbf{F}_{B_2,B_2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{F}_{B_d,B_d} \end{bmatrix}$$

where each matrix $\mathbf{F}_{B_i,B_i}$ *is a Frobenius matrix*

$$\begin{bmatrix} \mathbf{C}_{p_i^{s_{i,1}}} & 0 & \cdots & 0 \\ 0 & \mathbf{C}_{p_i^{s_{i,2}}} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{C}_{p_i^{s_{i,m_i}}} \end{bmatrix}$$

with $s_{i,1} \leq s_{i,2} \leq \ldots \leq s_{i,m_i}$ *and the polynomials* p_i *are such that* $\gcd(p_i, p_j) = 1$ *if* $i \neq j$.

We thus have that $p_i^{s_{i,m_i}}$ is the minimal polynomial of $\mathbf{F}_{B_i,B_i}$. The subspaces for which the matrix is a companion matrix are denoted by $V_{p_i^{s_{i,1}}}, V_{p_i^{s_{i,2}}} \ldots V_{p_i^{s_{i,m_i}}}$. respectively.

Now $\mathbf{k}^n$ being viewed as the direct sum

$$\bigoplus_{i=1}^{d} \bigoplus_{j=1}^{m_i} V_{p_i^{s_{i,j}}},$$

we consider $\mathbf{k}^n$ equipped with the natural structure of k[X]-module induced by T. Then the module $\mathbf{k}^n$ can be represented as the product of rings

$$R = R_{1,1} \times R_{1,2} \times \cdots \times R_{1,m_1} \times R_{2,1} \times R_{2,2} \times \cdots \times R_{2,m_2} \times \cdots \times R_{d,1} \times R_{d,2} \times \cdots \times R_{d,m_d}$$

considered as $\mathrm{k}[X]$-modules where

$$R_{i,j} = \mathrm{k}[X]/p_i^{s_{i,j}}.$$

For any vector u, we denote by $u_{|R_{i,j}}$ the component of u in the ring $R_{i,j}$. Thus from now on $u_{|R_{i,j}}$ is considered indiscriminately a vector or a polynomial of degree less than $s_{i,j}deg(p_i)$.

4.4 Shift-bases for the Expanded-Frobenius form

By Property 1 every shift-basis for the Expanded-Frobenius form is defined by a sequence of vectors

$$v'_{1,1}, v'_{1,2}, \ldots, v'_{1,m_1}, v'_{2,1}, \ldots, v'_{2,m_2}, \ldots, v'_{d,1}, \ldots, v'_{d,m_d}$$

such that for every couple i, j the polynomial with minimum degree cancelling $u'_{i,j}$ is $p_i^{s_{i,j}}$. Notice that in the $\mathrm{k}[X]$-module decomposition of $\mathbf{k}^n$, the vector $u'_{i,j}$ may have non zero coefficients in other rings than $R_{i,j}$.

However we can state more precisely the following

Lemma 3 *Let u be a vector in $\mathbf{k}^n$, such that $p_i^{s_{i,j}} u = 0$. Then the components of u viewed in the $\mathrm{k}[X]$-module decomposition of $\mathbf{k}^n$ satisfy*

$$u_{|R_{k,l}} = 0 \text{ if } k \neq i.$$

Proof: Suppose there exists k, l, $k \neq i$ such that $u_{|R_{k,l}} \neq 0$. Then $p_i^{s_{i,j}} u_{|R_{k,l}}$ cannot be zero, since $u_{|R_{k,l}}$ is not zero and $p_i^{s_{i,j}}$ is a unit of $R_{k,l} = \mathrm{k}[X]/p_k^{s_{k,l}}$. This contradicts the assumption on u. □

We characterize all components in $R_{i,l}$, $l \neq j$ of a vector u whose minimal polynomial is $p_i^{s_{i,j}}$.

Lemma 4 *Let u be a vector in $\mathbf{k}^n$, whose minimal polynomial is $p_i^{s_{i,j}}$. Then the components of u in $R_{i,l}$ are described as follows.*

- *$l < j$; $u_{|R_{i,l}}$ can be any element of $R_{i,l}$,*
- *$l = j$; $u_{|R_{i,l}}$, considered as a polynomial, is prime to p_i,*
- *$l > j$; $u_{|R_{i,l}}$ is a multiple of $p_i^{s_{i,u} - s_{i,j}}$.*

Proof: Since the minimal polynomial of u is $p_i^{s_{i,j}}$, then we have that $p_i^{s_{i,j}} v = 0$ for any vector v in $R_{i,l}$, whenever $l < j$, since $p_i^{s_{i,l}}$, which divides $p_i^{s_{i,j}}$, is the minimal polynomial of $\mathbf{T}$ restricted to $R_{i,l}$. This establishes the result for the case $l < j$.

In case $l = j$, we have seen that a vector is cyclic for a companion matrix if, considered as a polynomial, it is relatively prime to the minimal polynomial of that matrix.

In case $l > j$ we must have that

$$p_i^{s_{i,j}} u = 0$$

and this implies in $R_{i,j}$ that

$$p_i^{s_{i,j}} u_{|R_{i,l}} = 0 \bmod p_i^{s_{i,l}},$$

and thus we must have that $p_i^{s_{i,l} - s_{i,j}}$ divides $u_{|R_{i,l}}$. □

Property 2 *All shift-bases for an expanded Frobenius matrix* $\mathbf{D}$ *described as in definition 10 have the form*

$$v'_{1,1}, \mathbf{D}v'_{1,1}, \ldots, \mathbf{D}^{n_{1,1}-1}v'_{1,1}, v'_{1,2}, \mathbf{D}v'_{1,2}, \ldots, \mathbf{D}^{n_{1,2}-1}v'_{1,2}, \ldots, v'_{d,m_d}, \mathbf{D}v'_{d,m_d}, \ldots, \mathbf{D}^{n_{d,m_d}-1}v'_{d,m_d}$$

where $n_{i,j} = s_{i,j} \deg p_i$ *and where each* $v'_{i,j}$ *is any element in* $R_{i,l}$ *such that* $p_i^{s_{i,j}}$ *is its minimal polynomial.*

The proof follows straightforwards from Property 1. Moreover Lemma 4 gives an explicit construction of all $v'_{i,j}$ and consequently of all such shift-bases.

4.5 From shift-bases to the centralizer of a matrix

Theorem 6 *Given an operator* $\mathbf{T}$*, then every Shift-Hessenberg matrix similar to* $\mathbf{T}$ *yields the group* $\mathcal{Z}(\mathbf{T})$ *of operators commuting with* $\mathbf{T}$*, i.e. the centralizer of* $\mathbf{T}$ *in* $GL(n, \mathrm{k})$*. A one-to-one mapping from the set of shift-bases onto* $\mathcal{Z}(\mathbf{T})$ *can be constructed from the data of the Expanded-Frobenius form of* $\mathbf{T}$*.*

Proof: We first show how to any Shift-Hessenberg matrix similar to $\mathbf{T}$ there corresponds the centralizer of $\mathbf{T}$. Notice that we use the word *centralizer* for the group of matrices commuting with $\mathbf{T}$ even if $\mathbf{T}$ is not invertible. Let $\mathbf{B_1}$ and $\mathbf{B_2}$ be two basis satisfying the assumption of Property 1 and denote by $\mathbf{H_1}$ the Shift-Hessenberg matrix $\mathbf{B_1}^{-1}\mathbf{T}\mathbf{B_1} = \mathbf{B_2}^{-1}\mathbf{T}\mathbf{B_2}$. Then $\mathbf{G} = \mathbf{B_2}\mathbf{B_1}^{-1}$ which commutes with $\mathbf{T}$ can be constructed from the data of $\mathbf{B_1}$ and $\mathbf{B_2}$. Clearly any two Shift-Hessenberg matrices similar to $\mathbf{T}$ are conjugate. If $\mathbf{H_2} = \mathbf{C}^{-1}\mathbf{H_1}\mathbf{C}$, then $\mathbf{B_1C} = \mathbf{B'_1}$ and $\mathbf{B_2C} = \mathbf{B'_2}$ are shift-bases in which $\mathbf{T}$ is represented by $\mathbf{H_2}$. We thus have that $\mathbf{B'_2}\mathbf{B'_1}^{-1} = \mathbf{B_2}\mathbf{B_1}^{-1}$ which shows that the group obtained from the bases corresponding to $\mathbf{H_2}$ is the same as the one obtained by considering the bases corresponding to $\mathbf{H_1}$.

We next prove that *all matrices commuting with* $\mathbf{T}$ can be obtained in the form $\mathbf{B_2}\mathbf{B_1}^{-1}$ where $\mathbf{B_1}^{-1}\mathbf{T}\mathbf{B_1} = \mathbf{B_2}^{-1}\mathbf{T}\mathbf{B_2}$. Therefore let $\mathbf{Q}$ be any invertible matrix commuting with $\mathbf{T}$. We see that $\mathbf{QB_1}$ is a shift-basis corresponding to the vectors $\mathbf{Q}v_1, \mathbf{Q}v_2, \ldots, \mathbf{Q}v_m$ since $\mathbf{TQ}v_i = \mathbf{QT}v_i, i = 1, \ldots, m$. In addition Property 1 entails that $\mathbf{T}$ is represented by the same Shift-Hessenberg matrix $\mathbf{H_1}$ in that basis. Thus $\mathbf{B}_1(\mathbf{QB}_1)^{-1} = \mathbf{Q}$ belongs to the group constructed from the data of the set of all shift-bases in which $\mathbf{T}$ is represented by $\mathbf{H_1}$.

We are left with identifying a Shift-Hessenberg form for which all bases in which it represents $\mathbf{T}$ can be easily constructed. We choose the Expanded-Frobenius form $\mathbf{F}$ of $\mathbf{T}$. It is actually easier to construct the centralizer of $\mathbf{F}$ and afterwards go back to the one of the given $\mathbf{T}$ by conjugation. Indeed, a straightforward shift-basis for $\mathbf{F}$ is given by the identity matrix. In addition all other shift-bases may be constructed by using Lemma 4. □

Corollary 1 *The centralizer of the direct sum of two matrices* s *and* t *whose minimal polynomials are relatively prime is the direct product of the centralizers of* s *and* t *respectively.*

The Corrollary follows from Theorem 6 and Lemma 4.

Note 3 *The proof of Theorem 6 together with Lemma 4 clearly describes an algorithm to construct the whole of* $\mathcal{Z}(\mathbf{T})$. *To be more specific, let us observe that the first step of the algorithm consists in reducing the given matrix to its Expanded-Frobenius form. How to do this is dealt with in Section 9. The obtained basis* $\mathbf{B}$ *in which the given matrix is represented by the Frobenius form, i.e.* $\mathbf{B}^{-1}\mathbf{T}\mathbf{B} = \mathbf{F}$ *allows the final computation:* $\mathbf{B}\mathcal{Z}(\mathbf{F})\mathbf{B}^{-1} = \mathcal{Z}(\mathbf{T})$.

4.6 The size of the centralizer of a matrix over a finite field

In the case where $\mathbf{k} = \mathsf{F}_q$ we can derive from the previous results the enumeration of the centralizer of any given matrix.

Theorem 7 *Let* T *be an operator whose Expanded-Frobenius form is as in definition 10.*
The number of shift-basis for T *which yield the above Frobenius form is*

$$\prod_{i=1}^{d}\prod_{j=1}^{m_i} q^{\deg(p_i)\left(\sum_{w=1}^{j-1} s_{i,w}+(m_i-j)s_{i,j}\right)}\phi(p_i^{s_{i,j}}). \tag{7}$$

where $\phi(g)$ *is the number of polynomials of degree less than* $\deg(g)$ *prime to* g.

Proof: Each such shift basis is given by a sequence as

$$v'_{1,1}, v'_{1,2}, \ldots, v'_{1,m_1}, v'_{2,1}, \ldots, v'_{2,m_2}, \ldots, v'_{d,1} \ldots v'_{d,m_d}$$

in which for every couple i, j the polynomial with minimum degree canceling $v'_{i,j}$ is $p_i^{s_{i,j}}$.

In formula (7), the outermost product is due to lemma 3. The innermost product enumerates for each $p_i^{s_{i,j}}$ the number of vectors v such that $p_i^{s_{i,j}}v = 0$. The sum

$$\sum_{w=1}^{j-1} s_{i,w}$$

is for the rings $R_{i,l}$, $l < j$, in which any vector v satisfies $p_i^{s_{i,j}}v = 0$. The term

$$(m_i - j)s_{i,j}$$

is a result of the fact that for every $l > j$ the number of polynomials multiple of $p_i^{s_{i,l}-s_{i,j}}$ in $\mathbf{k}[X]/p_i^{s_{i,l}}$ is $q^{\deg(p_i)s_{i,j}}$.

Finally, $\phi(p_i^{s_{i,j}}) = q^{s_{i,j}\deg(p_i)}(1 - q^{-\deg(p_i)})$ is the number of polynomials prime to $p_i^{s_{i,j}}$, i.e. the number of units in $R_{i,j}$. □

4.7 The centralizer of a matrix

By Theorem 6, any shift-basis for the Expanded-Frobenius form yields a matrix commuting with T, and any commutator of the Expanded-Frobenius form gives a shift basis for the Expanded-Frobenius form. Since the group of commutators of T and the group of commutators of its Expanded-Frobenius form are conjugates, we have proved the following

Theorem 8 *The size of the centralizer of a matrix T in $M_n(q)$, whose Expanded-Frobenius form is given as in definition 10 is given by formula (7).*

Corollary 2 *Let $\mathbf{F}$ be the Frobenius map from F_q^n over F_q and let $f_1, \ldots, f_r \in \mathsf{F}_q[X]$ be the distinct irreducible factors of $X^n - 1$ in $\mathsf{F}_q[X]$ and $n_i = deg f_i$. Then the size of $\mathcal{Z}(\mathbf{F})$ is*

$$\nu(n,q) = q^n(1 - q^{-n_1}) \ldots (1 - q^{-n_r})$$

Proof: This is an immediate consequence of the previous corollary and of the fact that the given size is exactly the number of polynomials relatively prime to $X^n - 1$, since Lidl & Niederreiter[15, (1983),Theorem3.73] give $\nu(n,q)/n$ as the number of normal polynomials. □

Corollary 3 *Let $\mathbf{F}$ be the Frobenius map from F_q^n over F_q where n is some power of the characteristic of F_q. Then the size of $\mathcal{Z}(\mathbf{F})$ is $(q-1)q^{n-1}$.*

4.8 The average number of factors of a characteristic polynomial

R. Stong gives in [20] the following result.

Theorem 9 *Let X_n be the random variable assuming as values the number of factors of the characteristic polynomials of matrices in $GL(n,q)$ counted with multiplicities, and let EX_n be the expectation of X_n. Then EX_n is asymptotically equivalent to* $\log n$.

We shall prove the following

Theorem 10 *Let Y_n be the random variable assuming as values the number of factors counted with multiplicities of characteristic polynomials of matrices in $M_n(q)$, and let EY_n be the expectation of Y_n. Then for every $\epsilon > 0$ there exits n_0 such that $EY_n \leq 2(1+\epsilon)\log n$ for $n \geq n_0$.*

The proof of the Theorem needs two lemmas that will be first established. For any matrix $A \in M_n(q)$ we consider its Expanded-Frobenius form as follows

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix}$$

where s is a Frobenius form with characteristic polynomial X^{n_1} for some n_1, and t is an invertible matrix of size $n_2 = n - n_1$.

The following holds true.

Lemma 5 *The average number EZ_n of factors counted with multiplicities of the characteristic polynomial of t , for matrices A in $M_n(q)$, satisfies: $\forall \epsilon > 0, \exists n_0 \mid n \geq n_0 \Rightarrow EZ_n \leq (1+\epsilon)\log n$.*

Proof:

Let S_{n_1} be the set of Frobenius matrices with characteristic polynomial X^{n_1} and let S_{n_2} be the set of invertible Frobenius matrices whose characteristic polynomial has degree n_2.

We denote by z_{s,n_1} the size of the centraliser of $s \in S_{n_1}$ and by z_{t,n_2} the size of the centralizer of $t \in S_{n_2}$.

Given s and t in S_{n_1} and S_{n_2} respectively, then by Corollary 1 the number of matrices having Frobenius form

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \tag{8}$$

is

$$\frac{|GL(n,q)|}{z_{s,n_1} z_{t,n_2}}.$$

Then the number of matrices having X^{n_1} in the decomposition of their characteristic polynomial and a fixed matrix t in their second diagonal block in their Expanded-Frobenius form presented as in (8) is

$$\begin{aligned} \sum_{s \in S_{n_1}} \frac{|GL(n,q)|}{z_{s,n_1} z_{t,n_2}} &= \frac{|GL(n,q)|}{z_{t,n_2}} \sum_{s \in S_{n_1}} \frac{1}{z_{s,n_1}} \\ &= \frac{1}{z_{t,n_2}} \chi(n_1,n,q) \end{aligned}$$

where

$$\chi(n_1,n,q) = |GL(n,q)| \sum_{s \in S_{n_1}} \frac{1}{z_{s_{n_1}}}.$$

Now let $C_{n_2,k}$ be the set of polynomials $C(X)$, $C(0) \neq 0$ of degree n_2 that split into k factors counted with multiplicities and $S_{n_2,k}$ be the set Frobenius matrices of size n_2 whose characteristic polynomial belongs to $C_{n_2,k}$. The number of matrices in $M_n(q)$ whose characteristic polynomial is $X^{n_1}C(X)$, for $C(X)$ in $C_{n_2,k}$, is

$$\chi(n_1,n,q) \sum_{t \in S_{n_2,k}} \frac{1}{z_{t,n_2}}.$$

Denote by θ the random variable assuming as value the size of the non-singular part of a matrix, and denote by η the random variable assuming as value the number of factors of the characteristic polynomial of the non-singular part. The conditional probability $P_n\{\eta = k \mid \theta = n_2\}$ that $C(X)$ belongs to $C_{n_2,k}$ for a matrix in $M_n(q)$ whose characteristic polynomial is $X^{n_1}C(X)$, is thus

$$\begin{aligned} \frac{\chi(n_1,n,q) \sum_{t \in S_{n_2,k}} \frac{1}{z_{t,n_2}}}{\chi(n_1,n,q) \sum_{t \in S_{n_2}} \frac{1}{z_{t,n_2}}} &= \frac{\sum_{t \in S_{n_2,k}} \frac{|GL(n_2,q)|}{z_{t,n_2}}}{\sum_{t \in S_{n_2}} \frac{|GL(n_2,q)|}{z_{t,n_2}}} \\ &= P_{n_2}\{\eta = k\} \end{aligned}$$

where $P_n\{\eta = k\}$ is the probability that an invertible matrix in $GL(n,q)$ has a characteristic polynomial which splits into k factors.

Now we can conclude: the expected number of factors of the invertible block of any matrix in $M_n(q)$ is given by

$$\sum_{k=1}^{n} k \sum_{n_2=1}^{n} P\{\theta = n_2\} P_{n_2}\{\eta = k\} = \sum_{n_2=1}^{n} P\{\theta = n_2\} \sum_{k=1}^{n} k P_{n_2}\{\eta = k\} \quad (9)$$

$$= \sum_{n_2=1}^{n} P\{\theta = n_2\} EX_{n_2}. \quad (10)$$

This is an average over the EX_{n_2}, $n_2 = 1 \ldots n$.

Let ϵ be given. Since $EX_n \sim \log n$, there exists n_1 such that $n \geq n_1 \Rightarrow EX_n / \log n \leq 1 + \epsilon/2$. Thus

$$\frac{\sum_{n_2=1}^{n} P\{\theta = n_2\} EX_{n_2}}{\log n} = \frac{\sum_{n_2=1}^{n_1} P\{\theta = n_2\} EX_{n_2}}{\log n} + \sum_{n_2=n_1+1}^{n} P\{\theta = n_2\} \frac{EX_{n_2}}{\log n} \quad (11)$$

$$\leq \frac{\sum_{n_2=1}^{n_1} EX_{n_2}}{\log n} + \sum_{n_2=n_1+1}^{n} P\{\theta = n_2\}(1 + \frac{\epsilon}{2}) \quad (12)$$

$$\leq \frac{\sum_{n_2=1}^{n_1} EX_{n_2}}{\log n} + 1 + \frac{\epsilon}{2} \quad (13)$$

And since $\lim_{n \to \infty} 1/\log n = 0$, we can choose n_0 such that, for all $n \geq n_0$, we have that

$$\frac{EZ_n}{\log n} \leq 1 + \frac{\epsilon}{2}.$$

□

The proof of Theorem 10 will be completed by the following Lemma.

Lemma 6 *Let Z_n be the random variable assuming as values the number of factors X of characteristic polynomials of matrices in $M_n(q)$. Then the expectation EZ_n is asymptotically bounded by* $\log n$.

Proof: Let us consider the translation $M \mapsto M + I_n$. The factor X^{n_1} of a matrix M becomes $(X-1)^{n_1}$ in the factorization of the characteristic polynomial of $M' = M + I_n$. Consider the Frobenius form of M'

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix}$$

where s is nilpotent and t is invertible, then $(X-1)^{n_1}$ is the largest power of $X-1$ which is a factor of the characteristic polynomial $C(X)$ of t. By Lemma 5 the expected number of factors of $C(X)$ is asymptotically $\log n$, thus n_1 is asymptotically bounded by $\log n$. □

Theorem 10 now follows from Lemma 5 and Lemma 6.

Example of construction of a subgroup of $\mathcal{Z}(\mathbf{T})$.

The construction that we have described in the proof of Theorem 6 consists in constructing the matrices like $\mathbf{G}$ which commute with $\mathbf{T}$ by first obtaining matrices as $\mathbf{B_1}$ and $\mathbf{B_2}$.

We now show on an example how this can be systematically done when restricting the selection of such matrices in a particular prescribed subgroup of $GL(n,q)$. The example is concerned with a subgroup of a centralizer in which every matrix is taken from the group of automorphisms of the Hamming $(8,4,4)$ binary code.

We consider the subgroup $GL(n,q)_1$ of $GL(n,q)$ consisting of all matrices in which the sum of entries in every column is 1. Let us recall that the general linear affine group $GA(n,q)$ is the group of all transformations of the form $x \mapsto \mathbf{G}x + b$, $(\mathbf{G}, b) \in GL(n,q) \times \mathbf{k}^n$. We first prove in a few lines a property which is a particular case of the well known theorem of Kasami-Lin-Peterson [17].

Property 3 *The group of non-projective automorphisms of the extended first order Reed-Muller code over F_q , i.e. the group of permutations on the set of positions of the codewords preserving the code, is the general affine group $GA(n-1,q)$. Moreover $GL(n,q)_1$ is isomorphic to $GA(n-1,q)$.*

Proof: Let us first prove the last assertion. Let $\mathbf{Q}$ be the $n \times n$ matrix in which the first $n-1$ rows are the transposed of the first $n-1$ unit vectors and the last row is the all-one vector. Then it is easy to see that $\mathbf{Q}GL(n,q)_1\mathbf{Q}^{-1}$ represents $GA(n-1,q)$ as a subgroup of $GL(n,q)$. Next let $\mathbf{M}$ be the matrix formed by all columns summing up to 1. Then $GL(n,q)_1$ is seen to be the whole group of non-projective automorphisms of the linear code $(q^{n-1}, k=n, q^{n-2}(q-1))$ whose generator matrix is $\mathbf{M}$, the code being known as the extended first order Reed-Muller code over F_q (Properties of that code are described in [17]). □

For $n=4$, $q=2$ we get an instance of such a matrix $\mathbf{M}$ which also is the generator matrix of a Hamming code. We have that

$$\mathbf{M} = [\mathbf{T}, \mathbf{I}] = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Denote by $\mathcal{Z}(\mathbf{T})_1$ the subgroup of matrices of $GL(4,2)_1$ commuting with $\mathbf{T}$. By theorem 6, we are able to construct any matrix $\mathbf{G}$ commuting with $\mathbf{T}$. Many matrices of $GL(4,2)_1$ do not commute with $\mathbf{T}$. Thus we are able to constuct an increasing sequence of subgroups of $\mathcal{Z}(\mathbf{T})_1$ by adjoining new commutating matrices to the group just obtained, those subgroups being properly contained in $GL(4,2)_1$.

For instance:

$$\mathbf{B_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{B_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

are, by Property 1, two shift-basis associated with the Shift-Hessenberg matrix

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

Clearly there is no difficulty in constructing shift-basis with all vectors with odd weights.

Then by theorem 6, we have that

$$\mathbf{G_1} = \mathbf{B_2}\mathbf{B_1}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

commutes with $\mathbf{T}$. Constructing an other shift-basis, say

$$\mathbf{B_3} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix},$$

we have that

$$\mathbf{G_2} = \mathbf{B_3}\mathbf{B_2}^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

is another matrix in $\mathcal{Z}(\mathbf{T})_1$ which does not commute with $\mathbf{G_1}$. That could have been expected since $\mathbf{T}$ is not diagonizable: indeed the characteristic polynomial of $\mathbf{T}$ is $(X+1)^4$ and its minimal polynomial is $(X+1)^2$.

We now construt two shift-basis which are not in $GL(4,2)_1$.

$$\mathbf{B_1'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \mathbf{B_2'} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

they both correspond to the Shift-Hessenberg matrix

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

Then

$$\mathbf{G_1'} = \mathbf{B_2'}\mathbf{B_1'}^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

is in $\mathcal{Z}(\mathbf{T})$.

We observe that $\mathbf{G_1'}$ is in $GL(4,2)_1$ as well.

In fact we have as in the proof of the theorem 6 that

$$\mathbf{G}_1'\mathbf{B}_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

is a shift-basis associated to $\mathbf{H}$. We incidentally observe that $\mathbf{G}_1'^3 = \mathbf{T}$ and that $\mathbf{G}_1'\mathbf{G}_1 \neq \mathbf{G}_1\mathbf{G}_1'$.

There unexpectedly exists an algorithm derived from the well known Hessenberg algorithm presented in Subsection 2.1 to directly derive a shift-basis for any operator $\mathbf{T}$ without the need of selecting a vector at random. This is done in next section.

4.9 The Shift-Hessenberg form

The Shift-Hessenberg form for a matrix is a particular Hessenberg form. The cost for the Shift-Hessenberg form is still $O(n^3)$. The following definition clearly is equivalent to the previous one. It was used in [16], as an intermediate matrix for computing the Frobenius form of a matrix.

Definition 11 *A matrix* $\mathbf{H}$ *in* $M_n(\mathrm{k})$ *is Shift-Hessenberg if it has the following form*

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

i.e. $\mathbf{H}$ *is a Hessenberg matrix such that* $(h_{i+1,i} \neq 0) \Rightarrow (h_{i+1,i} = 1 \text{ et } \forall j \leq i \; h_{j,i} = 0)$.

The parameter m *of a Shift-Hessenberg matrix is defined to be the number of zeros on the first subdiagonal, plus one.*

Note 4 *The number* m *is the number of diagonal blocks, each block being a companion matrix, i.e. a matrix of the form*

$$\begin{bmatrix} & & & & c_0 \\ 1 & & & & c_1 \\ & 1 & & & c_2 \\ & & \ddots & & c_i \\ & & & 1 & c_{n-2} \\ & & & & c_{n-1} \end{bmatrix}$$

The characteristic polynomial of such a matrix equals its minimal polynomial and is

$$X^n - c_{n-1}X^{n-1} - c_{n-2}X^{n-1} \cdots - c_1X^1 - c_0$$

In the case where the parameter $m = 1$, the Shift-Hessenberg matrix is itself a companion matrix. The other extreme situation is for $m = n$ where we have an upper triangular matrix.

An algorithm for obtaining a Shift-Hessenberg form similar to a given matrix
We have a theorem analogous to the one concerning Hessenberg matrices.

Theorem 11 *For all* $\mathbf{A}$ *in* $M_n(\mathrm{k})$, *there exists a Shift-Hessenberg matrix* $\mathbf{H}$ *and an invertible matrix* $\mathbf{P}$ *such that* $\mathbf{H} = \mathbf{PAP}^{-1}$. *The matrices* $\mathbf{H}$ *and* $\mathbf{P}$ *can be obtained in* $O(n^3)$ *elementary operations.*

Proof: Again, we prove the theorem by giving an algorithm.
Input $\mathbf{A} \in M_n(\mathbf{k})$, where $\mathbf{k}$ is a field.
Output $\mathbf{H}$, a Shift-Hessenberg form for $\mathbf{A}$, and $\mathbf{P}$ such that $\mathbf{H} = \mathbf{PAP}^{-1}$.

$\mathbf{H}$:=$\mathbf{A}$; $\mathbf{P}$:=I_n; i:=1{*ith row is denoted by* L_i, *ith column by* C_i}
while i < n **do** {*treat each column, do not treat the last one.*}
    Search for the first non zero element in column i
    starting at row index $i + 1$. *If such an element exists, let* j *be that position*
    if *no such element has been found* **then** i:=i+1
{*that column remains unchanged up to the end*}
    else
        with H:
            swap rows $i + 1$ and j
            swap columns $i + 1$ and j
        with P: swap rows $i + 1$ and j
    c := 1/**H**[i+1,i] {*pivoting element*}
    with H
        $L_{i+1} \leftarrow L_{i+1} \times c$; $C_{i+1} \leftarrow C_{i+1}/c$ {$h[i+1, i]$ *is now a* 1}
    with P $L_{i+1} \leftarrow L_{i+1} \times c$
    for l **from** 1 **to** n **suchthat** $l \neq i + 1$ **do**
        h:= **H**[l,i]
        with H: $L_l \leftarrow L_l - h \times L_{i+1}$; $C_l \leftarrow h \times C_l + C_{i+1}$
        with P: $L_l \leftarrow L_l - h \times L_{i+1}$
    i:=i+1
return(H,P)

□

We now investigate more precisely the number m of diagonal blocks of the Shift-Hessenberg form for a matrix $\mathbf{A}$. Let us introduce another parameter $m_{\mathbf{A}}$ associated with a matrix $\mathbf{A}$, which is involved in the complexity assessments.

Definition 12 *Let* $\mathbf{A}$ *be a square matrix in* $M_n(\mathrm{k})$. *We denote by* $\underline{m_{\mathbf{A}}}$ *the maximum size of an increasing sequence of invariant subspaces of* k^n *under* $\mathbf{A}$*:*

$$V_1 \subset V_2 \subset \ldots V_{m_{\mathbf{A}}}$$

It follows that for any Shift-Hessenberg form of a matrix $\mathbf{A}$ with parameter $m_{\mathbf{A}}$, we have that parameter m is bounded from above by $m_{\mathbf{A}}$, since each zero on the first subdiagonal yields an invariant subspace. The invariant subspaces are nested.

Property 4 *The number* $m_{\mathbf{A}}$ *equals the number of irreducible factors of the characteristic polynomial of* $\mathbf{A}$*, counted with multiplicities.*

We recall without proof the following theorem by Richard Stong. [20][1]

Theorem 12 ([20], Proposition 12) *The asymptotic value of the average number of irreducible factors, counted with multiplicity, of the characteristic polynomial of an invertible matrix of size* n *is equivalent to* $\log n$*, with an average deviation of* $\log n$.

That result, which is completed by our Theorem 10 leads to the corollary

Corollary 4 *The expected value of* $m_{\mathbf{A}}$ *is* $O(\log n)$.

For clarity, the complexity of some algorithms will be given in terms of n and $m_{\mathbf{A}}$. This will lead to complexities in terms of n and $\log n$. Notice that the algorithms here presented all are deterministic. However the complexity is a random variable bounded from above.

4.10 Evaluating a polynomial at a Shift-Hessenberg matrix

In this subsection we introduce some results about the complexity of computations with a Shift-Hessenberg matrix. The next subsection is concerned with solving some problems concerning companion matrices.

First observe that a Shift-Hessenberg is a sparse matrix, with at most $m+1$ non-zero entries in each row. This leads to the following lemma.

Lemma 7 *Let* $\mathbf{H}$ *be a Shift-Hessenberg matrix of size* n*, and let* $\mathbf{M}$ *be any matrix of size* $n \times n'$. *Then product* $\mathbf{HM}$ *can be computed at cost* $O(mnn')$.

Furthermore a Shift-Hessenberg matrix has some properties regarding cyclicity, as already seen in definition 9, which can be exploited for reducing costs. A new definition is introduced.

Definition 13 *Let* $\mathbf{H}$ *in* $M_n(\mathrm{k})$ *be a Shift-Hessenberg matrix. The matrix* $\mathbf{A}$ *is* <u>polycyclic</u> *for* $\mathbf{H}$ *if its columns* C_i *have the form*

$$\left[v_1, \mathbf{H}v_1, \ldots \mathbf{H}^{n_1-1}v_1, v_2, \mathbf{H}v_2, \ldots, \mathbf{H}^{n_2-1}v_2, \ldots, v_m, \mathbf{H}v_m, \ldots, \mathbf{H}^{n_m-1}v_m\right] \qquad (14)$$

where $n_1, n_2, \ldots n_m$ *are the sizes of the diagonal blocks of* $\mathbf{H}$*, and* $v_1, v_2, \ldots, v_m$ *are vectors of* k^n.

[1]This result was brought to the attention of one of the authors by Jeremy Johnson, from Drexel University, Philadelphia, USA at a meeting in Oberwolfach organized by Thomas Beth in February 1993.

Notice that linear independance , required in the definition of shift-bases, does not show in this definition.

Proposition 1 *Let* $\mathbf{H}$ *be a Shift-Hessenberg matrix, let* $\mathbf{A}$,$\mathbf{B}$ *be two matrices which are polycyclic for* $\mathbf{H}$. *Let* α, β *be any field elements, then* $\alpha\mathbf{A} + \beta\mathbf{B}$, I_n, $\mathbf{H}$, $\mathbf{HA}$ *and* $\mathbf{HB}$ *all are polycyclic for* $\mathbf{H}$.

In other words, the matrix $\mathbf{H}$ defines a k[$\mathbf{H}$]-module of polycyclic matrices which is a k[$\mathbf{H}$]-submodule of $M_n(\mathrm{k})$.

Proposition 2 *Let* $\mathbf{H}$ *be a Shift-Hessenberg matrix of parameter* m. *Then the product* $\mathbf{HA}$ *can be obtained with complexity* $O(mn^2)$ *for any matrix* $\mathbf{A}$ *in* $M_n(\mathrm{k})$ *and with complexity* $O(m^2n)$ *whenever* $\mathbf{A}$ *is polycyclic for* $\mathbf{H}$.

Proof: For a polycyclic $\mathbf{A}$, the product $\mathbf{HA}$ is performed by modifying $\mathbf{A}$ as follows. Delete v_1, shift all vectors to the left. Then replace $v_2, \ldots, v_m$ by $\mathbf{H}\mathbf{H}^{n_1-1}v_1, \ldots, \mathbf{H}\mathbf{H}^{n_{m-1}-1}v_{m-1}$ respectively.

Finally, put $\mathbf{H}\mathbf{H}^{n_m-1}v_m$ as nth column. The whole cost is $m(mn)$. □

Corollary 5 *A polynomial* $p(X)$ *of degree at most* t *can be evaluated at* $\mathbf{H}$ *with complexity* $O(tm^2n)$.

Proof: We apply Horner's rule for evaluating a polynomial $p(\mathbf{H}) = p_t\mathbf{H}^t + p_{t-1}\mathbf{H}^{t-1} + \cdots + p_1\mathbf{H} + p_0\mathbf{I}$. We compute $\mathbf{h}_1 = p_t\mathbf{H} + p_{t-1}\mathbf{I}$, $\mathbf{h}_2 = \mathbf{H}\mathbf{h}_1 + p_{t-2}\mathbf{I}, \ldots, \mathbf{h}_t = \mathbf{H}\mathbf{h}_{t-1} + t_0\mathbf{I}$. From proposition 2, $\mathbf{h}_i$ is computed from $\mathbf{h}_{i-1}$ at a cost $O(m^2n)$, thus a total cost of $O(tm^2n)$ for $p(\mathbf{H})$. □

We now can refine our algorithm for Shift-Hessenberg matrices, using the fact that evaluating a polynomial at a Shift-Hessenberg matrix is cheap.

Input A Shift-Hessenberg matrix $\mathbf{H}$ and its factored characteristic polynomial $C(X)$.
Output The minimal polynomial of $\mathbf{H}$ and the splitting of k^n into all characteristic subspaces of $\mathbf{H}$.

Step 1: Find a splitting of $C(X) = P(X)Q(X)$, as in step 1 of 3.3 for the general algorithm.
Step 2: Compute $Q(\mathbf{H})$, $P(\mathbf{H})$ as in the proof of Corollary 5. This gives generating vectors for subspaces for V_P and V_Q respectively.
Step 3: Compute a basis for V_P and V_Q respectively. This is done with gaussian elimination, at cost $O(n^3)$.
Step 4: Change basis, as in step 4 of 3.3. Next compute the matrices $\mathbf{H}_P$ and $\mathbf{H}_Q$ of the restriction of $\mathbf{H}$ to V_P and V_Q respectively. The cost is again $O(n^3)$. Compute the Shift-Hessenberg forms $\mathbf{H}'_P$ and $\mathbf{H}'_Q$ for both matrices $\mathbf{H}_P$ and $\mathbf{H}_Q$.
Recursive Step Recursively apply the same procedure to $\mathbf{H}'_P$ and $\mathbf{H}'_Q$. This ends in exhibiting all characteristic subspaces.

Corollary 6 *Using the evaluation rule given in the proof of Corollary 5, and the algorithm described in Section 3 and modified for Shift-Hessenberg matrices , it is possible, given the factored characteristic polynomial of* $\mathbf{H}$ *to compute the minimal polynomial of a Shift-Hessenberg matrix and a block-diagonal matrix* $\mathbf{D}$ *similar to* $\mathbf{H}$ *in* $O(n^3 + m_{\mathbf{H}}^2 n^2)$ *elementary operations.*

Remark 1 *The term in* n^3 *in the above evaluation is due to the computation of the Hessenberg form of the given matrix and to construct bases for invariant subspaces.*

A deterministic algorithm giving the minimal polynomial in $O(n^3 m_A)$ steps has been obtained by Patrick Ozello ([16]). That algorithm is thus slightly more expensive than the present algorithm, but it does not require knowledge of the characteristic polynomial. It however does not produce the characteristic susbspaces.

Remark 2 *A Shift-Hessenberg form for the given matrix* $\mathbf{A}$ *being computed first, then all results of Corollary 6 are obtained at total cost* $O(n^3 + m_{\mathbf{A}}^2 n^2)$ *by Theorem 11.*

Remark 3 *Note that the worst case complexity is* $O(n^4)$*, when the parameter* $m_{\mathbf{A}}$ *of matrix* $\mathbf{A}$ *is* n*. However, since* $m_{\mathbf{A}}$ *is known when the factorization of* $C_{\mathbf{A}}(X)$ *is known, the algorithm descibed in section 3 could be used for a worst-case complexity of* $O(n^3\sqrt{n})$*.*

4.11 Linear algebra with a companion matrix

Since the diagonal blocks of a Shift-Hessenberg matrix are companion matrices, we will use the fact that companion matrices appear in linear representations of algebras of polynomials. This subsection is dedicated to describing very simple and efficient procedures for solving relations involving a companion matrix. This will lead to low complexity algorithms.

From now on, given a companion matrix C with minimal polynomial $\pi(X)$ of degree n, and a vector $v = (v_0, \ldots v_{n-1})$, the vector is identifyed with a polynomial

$$v = (v_0, v_1, \ldots, v_{n-1}) \Leftrightarrow v(X) = v_0 + v_1 X + v_2 X^2 + \cdots + v_{n-1} X^{n-1} \tag{15}$$

We first consider the computation of Cv for any vector $v \in \mathbf{k}^n$. Observe that $Cv = Xv(X) \bmod \pi(X)$. This means that computing Cv is only a shift-add on the vector v, modulo $\pi(X)$. We state this in a lemma.

Lemma 8 *For a companion matrix* C*,* $\forall v \in \mathbf{k}^n$*,* Cv *is computed with complexity* $2n$*.*

This entails the following lemma.

Lemma 9 *For a companion matrix* C *with minimal polynomial* $\pi(X)$*, for all* v *in* $\mathbf{k}^n$*, for all* $P(X)$ *of degree at most* n*, then* $P(C)v$ *can be computed at cost* $O(n^2)$*.*

Proof: Computing $P(C)v$ reduces to computing $P(X)v(X) \bmod \pi(X)$. □

The solution for specific systems of equations is obtained as shown in the proof of the next statement.

Lemma 10 *For a companion matrix C with minimal polynomial $\pi(X)$, for all v in $\mathbf{k}^n$, for $P(X)$ prime to $\pi(X)$ and of degree at most n, solving the following system at u*

$$P(C)u = v \tag{16}$$

can be done at cost $O(n^2)$.

Proof: Since $P(X)$ is prime to $\pi(X)$, there exists $Q(X)$ such that
$P(X)Q(X) = 1$
(mod $\pi(X)$). The solution u is given by $u = Q(C)v$. Computing $Q(X)$ can be done in $O(n^2)$ by the extended euclidean algorithm, and computing $Q(C)v$ is done in $O(n^2)$ by Lemma 9. □

Let us state a lemma for computing the minimal polynomial of a vector for a companion matrix.

Lemma 11 *The minimal polynomial $\pi_v(X)$ for of a companion matrix C restricted to a vector v is given by*

$$\pi_v(X) = \frac{\pi(X)}{\gcd(\pi(X), v(X))} \tag{17}$$

where $\pi(X)$ is the minimal polynomial of C

Proof: Let $\pi_v(X)$ be the polynomial of smallest degree such that $\pi_v(C)v = 0$. From previous remarks, we have to find $\pi_x(X)$ of smallest degree such that

$$\pi_x(X)v(X) = 0 \pmod{\pi(X)}. \tag{18}$$

$\pi_u(X) = \frac{\pi(X)}{\gcd(\pi(X),v(X))}$ is a solution. Let $\pi_w(X)$ be any solution for $\pi_x(X)v(X) = 0 \bmod \pi(X)$, then

$$\pi(X) \mid \pi_w(X)v(X),$$

and thus

$$\frac{\pi(X)}{\gcd(\pi(X), v(X))} \mid \pi_w(X)$$

Hence $\pi_u(X)$ is the polynomial with smallest degree that we are looking for. □

5 A direct algorithm for the minimal polynomial

We now give another algorithm for computing the minimal polynomial of a matrix **A**, given a Shift-Hessenberg form for **A**. This algorithm is a direct algorithm, it does not appeal to any "divide-and-conquer" process, and it does not require any previous knowledge on the characteristic polynomial. The drawback is that is does not produce a diagonal-block decomposition of $\mathbf{k}^n$ into the characteristic subspaces of **A**.

Assume that we are given a Shift-Hessenberg form **H** for matrix **A**. Then **H** is described by blocks as follows.

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{B_1,B_1} & \mathbf{H}_{B_1,B_2} & \cdots & \mathbf{H}_{B_1,B_m} \\ 0 & \mathbf{H}_{B_2,B_2} & \cdots & \mathbf{H}_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{H}_{B_m,B_m} \end{bmatrix}.$$

Notation 1 *We denote by B_k the set of indices of block k. We also denote by $B_{\geq k}$ the set of indices $B_k \cup B_{k+1} \ldots \cup B_m$. For any matrix $\mathbf{A} \in M_n(\mathrm{k})$ we denote by $\mathbf{A}_{B_i,B_j}$ the matrix obtained from rows in B_i and columns in B_j. We denote by $\mathbf{A}_{B_{\geq k}}$ the square matrix obtained from all rows and columns from the k^{th} block up to the end.*

5.1 Nested ideals related to H

Property 5 *Let I_k denote the set of polynomials $g(X)$ such that*

$$g(\mathbf{H})_{B_i,B_i} = 0,\ i = 1,\ldots,m,\ \text{and}\ g(\mathbf{H})_{B_i,B_j} = 0,\ i < j,\ i,j = k,\ldots,m, \tag{19}$$

We have that I_k is an ideal of $\mathrm{k}[X]$. Moreover the following chain of inclusions holds,

$$I_1 \subseteq I_2 \cdots \subseteq I_m. \tag{20}$$

We have that

$$I_k = (p_k(X)), k = 1,\ldots,m\ \text{and}\ p_k(X) \mid p_{k-1}(X), k = 2,\ldots,m \tag{21}$$

Finally $p_1(X)$ is the minimal polynomial of $\mathbf{H}$.

Before giving the proof let us introduce some notation.

Notation 2 *Since we will have that $p_{k+1}(X) \mid p_k(X)$, we will denote by $\underline{\phi_k(X)}$ the polynomial such that $p_k(X) = \phi_k(X)p_{k+1}(X)$. The minimal polynomial of the companion matrix $\mathbf{H}_{B_i,B_i}$, which lives on the last column of that matrix is denoted by $\underline{f_i(X), i = 1,\ldots,m}$.*

Proof: Consider the case where $k = 1$. We have that

$$g(\mathbf{H})_{B_i,B_j} = 0, i \leq j;\ i,j = k,\ldots,m \Leftrightarrow g(\mathbf{H}) = 0$$

and the ideal $I_1 = (p_1(X))$ is the ideal annihilating the matrix $\mathbf{H}$, i.e. the ideal defining the minimal polynomial $p_1(X)$ of $\mathbf{H}$.

Now let I_k be the set of polynomials such that (19) holds.

Then I_k is an ideal of polynomials and $\mathrm{k}[X]$ being a principal ideal domain, we have that $I_k = (p_k) \subseteq I_{k+1}$ and thus $p_{k+1}(X) \mid p_k(X)$. From now on $\phi_k(X)$ is well defined.

Let $g(X) \in I_{k+1}$, then, focusing on blocks with row index set B_k and column index set B_j, $k \leq j \leq m$, we consider the result of computations with $\mathbf{H}$ and we obtain the relation

$$(\mathbf{H}g(\mathbf{H}))_{B_k,B_j} = \mathbf{H}_{B_k,B_k}(g(\mathbf{H}))_{B_k,B_j}$$

Let now $p(X)$ be a polynomial of the form $q(X)p_{k+1}(X)$, which is the general form for polynomials in I_{k+1}. We then have that

$$p(\mathbf{H})_{B_k,B_j} = q(\mathbf{H})_{B_k,B_k}(p_{k+1}(\mathbf{H}))_{B_k,B_j} = q(\mathbf{H}_{B_k,B_k})(p_{k+1}(\mathbf{H}))_{B_k,B_j}, k \leq j \leq n$$

Thus $p(\mathbf{H})_{B_k,B_j} = 0,\ k \leq j \leq n$ if and only if $q(\mathbf{H}_{B_k,B_k})(p_{k+1}(\mathbf{H}))_{B_k,B_j} = 0,\ k \leq j \leq n$, i.e. iff $q(\mathbf{H}_{B_k,B_k})$ annihilates the space generated by columns of all matrices $(p_{k+1}(\mathbf{H}))_{B_k,B_j}$, $j = k,\ldots,m$.

We conclude that $\phi_k(X)$ is the minimal polynomial of $\mathbf{H}_{B_k,B_k}$ restricted to the subspace generated by the columns of all $(p_{k+1}(\mathbf{H}))_{B_k,B_j}, j = k,\ldots,m$. Now $p_{k+1}(X)$ is, by the definition of I_{k+1}, the polynomial with smallest degree such that all $p_{k+1}(\mathbf{H})_{B_{k+1},B_j} = 0, j = k,\ldots,m$. Consequently we have that $I_k = (\phi_k(X)p_{k+1}(X))$ □

Notice that, since $f_k(X)$ is the minimal polynomial of $\mathbf{H}_{B_k,B_k}$ on $\mathbf{k}^{B_k}$, we have that $\phi_k(X) \mid f_k(X)$.

5.2 The algorithm for the minimal polynomial of H

From Property 5, the algorithm consists in constructing $p_m(X)$, $p_{m-1}(X)\ldots p_1(X)$, step by step.

First step

Polynomial $p_m(X)$ is to be computed. Since all diagonal blocks of $p_m(\mathbf{H})$ vanish, then $p_m(X)$ is the least commun multiple of all $f_i(X), i = 1,\ldots,m$.

Iterative step: computing $p_k(X)$ from the data of $p_{k+1}(X)$

Assume that $p_{k+1}(X)$ is already computed. We have that

$$p_{k+1}(\mathbf{H}) = \begin{bmatrix} 0 & p_{k+1}(\mathbf{H})_{B_1,B_2} & \cdots & \cdots & p_{k+1}(\mathbf{H})_{B_1,B_m} \\ & \ddots & & & \\ & & 0 & p_{k+1}(\mathbf{H})_{B_k,B_{k+1}} & p_{k+1}(\mathbf{H})_{B_k,B_m} \\ & & 0 & 0 & 0 \\ & & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

From the proof of Property 5, we have to find $\phi_k(X)$, the minimal polynomial of $\mathbf{H}_{B_k,B_k}$ restricted to the subspace generated by the columns of matrices $p_{k+1}(\mathbf{H})_{B_k,B_j}$, $k \le j \le m$. We then shall have $p_k(X) = \phi_k(X)p_{k+1}(X)$. We compute $\phi_k(X)$ as follows.

Let ${}^t a^1 = (a_1^1, a_2^1, \ldots, a_{m_k}^1)$ be the first non zero-column of the array formed by all matrices $p_{k+1}(\mathbf{H})_{B_k,B_j}$, $j \ge k$. We compute the minimal polynomial $\phi_{k,a^1}(X)$ of $\mathbf{H}_{B_k,B_k}$ restricted to a^1. Thus $\phi_{k,a^1}(X)$ is a factor of $\phi_k(X)$ and $\mathbf{H}_{k,a^1} = \phi_{k,a^1}(\mathbf{H})p_{k+1}(\mathbf{H})$ is then computed.

Next the process is repeated on the first non-zero column a^2 of column of $\mathbf{H}_{k,a^1}$, to get a new factor $\phi_{k,a^2}(X)$ de $\phi_k(X)$. We compute again $\mathbf{H}_{k,a^2} = \phi_{k,a^2}(\mathbf{H})\mathbf{H}_{k,a^1}$, and proceed with the first non-zero column of the array $\{(\mathbf{H}_{k,a^2})_{B_k,B_j}\}, j \ge k$. The process is stopped when all columns are canceled. We then have that $\phi_k(X) = \phi_{k,a^1}(X)\phi_{k,a^2}(X)\cdots\phi_{k,a^l}(X)$ where a^l is the last non-zero column which was met.

The key is that computing the minimal polynomial of $\mathbf{H}_{B_k,B_k}$ restricted to a column is easily performed by the use of Lemma 11. It reduces to a gcd computation on polynomials. That cost is negligeable, and the most expensive computations are the evaluations of $\mathbf{H}_k, \mathbf{H}_{k,a^1}, \mathbf{H}_{k,a^2}$. Fortunately the cost is much reduced by the use of Corollary 5.

5.3 Complexity bounds

The most expensive computations lie in the computation of matrices $p_m(\mathbf{H})$, $p_{m-1}p_m(\mathbf{H})$, $p_{m-2}p_{m-1}p_m(\mathbf{H})\ldots$ At each step the polynomial obtained divides $p_1(X)$. The total cost is thus bounded by the cost of evaluating a polynomial of degree n at a Shift-Hessenberg

matrix, which is, by Corollary 5, $O(m^2n^2)$ a number of times which is bounded by $m_{\mathbf{A}}$, the number of factors of the characteristic polynomials.

Each computation of a minimal polynomial over a vector is done at cost $O(n_k^2)$. The number of such computations is also bounded by $m_{\mathbf{A}}$. This results in $O(m_{\mathbf{A}}n^2)$ elementary operations for all those gcd computations.

Theorem 13 *Given a Shift-Hessenberg form of a matrix, its minimal polynomial can be obtained in $O(m_{\mathbf{A}}m^2n^2)$ elementary operations without any previous knowledge the characteristic polynomial.*

This compares well with Ozello's procedure, which is $O(n^3m)$.

Remark 4 *Note that the worst case complexity, when m is n, is $O(n^5)$, which is bad. When m is large, one can use the following technique for computing the matrices $p_m(\mathbf{H})$, $p_{m-1}p_m(\mathbf{H})$, $p_{m-2}p_{m-1}p_m(\mathbf{H})\ldots$*

Let $d_1, d_2, \ldots, d_m$ be the degrees of the polynomials $p_1, p_2, \ldots, p_m$. First note that $p_m(\mathbf{H})$ is computed at cost $d_m m^2 n$ by corollary 5. Let $\mathbf{C}_{k+1}$ be the matrix $p_{k+1}p_k \cdots p_m(\mathbf{H})$ which is a polycyclic matrix for $\mathbf{H}$, and let $p_k(X) = X^{d_k} + a_{d_k-1}X^{d_k-1} + \cdots + a_1X + a_0$. We compute $p_k(\mathbf{H})\mathbf{C}_{k+1}$ as follows:

$$(\mathbf{H}^{d_k} + a_{d_k-1}\mathbf{H}^{d_k-1} + \cdots + a_1\mathbf{H} + a_0)\mathbf{C}_{k+1} = (\mathbf{H}^{d_k-1} + a_{d_k-1}\mathbf{H}^{d_k-2} + \cdots + a_1)\mathbf{H}\mathbf{C}_{k+1} + a_0\mathbf{C}_{k+1}$$

Now the product $\mathbf{H}\mathbf{C}_{k+1}$ is computed at cost $O(m^2n)$ by lemma 2, and the product $a_0\mathbf{C}_{k+1}$ at cost $O(n^2)$, and the sum of these two matrices is computed at cost $O(n^2)$. Thus computing $p_k(\mathbf{H})\mathbf{C}_{k+1}$ is performed at cost $O(d_k(m^2n+n^2))$, and the final cost is $O((d_1+\cdots+d_m)(m^2n+n^2)) = O(m^2n^2+n^3)$. This method is thus better when m is large, and leads to a worst case complexity of $O(n^4)$.

Corollary 7 *The minimal polynomial of any matrix $\mathbf{A}$ can be obtained in $O(n^3+m_{\mathbf{A}}^3n^2)$ elementary operations without any previous knowledge on the characteristic polynomial, where the term in n^3 is only due to computing a Shift-Hessenberg form of the given matrix.*

Proof: First compute a Shift-Hessenberg matrix $\mathbf{H}$ for $\mathbf{A}$. Since parameter m of $\mathbf{H}$ is not greater than $m_{\mathbf{A}}$, then the statement is entailed by Theorem 13. □

6 Searching for a cyclic vector

6.1 Normal basis

6.1.1 Introduction

We now are going to find a cyclic vector for a matrix $\mathbf{A}$. Though a construction will come to light as a by-product of the Rational Canonical form obtained in last section without any assumption on the matrix $\mathbf{A}$, this section and the next one are dedicated to this goal, and both assume that the characteristic polynomial of $\mathbf{A}$ is square-free. The reason is that a better algorithm is constructed under that assumption. Moreover, factoring the

characteristic polinomial is not needed for these algorithms. Checking that a polynomial is square-free is done by derivating and then computing a g.c.d.. This is especially attractive for the zero-characteristic. Now the characteristic polynomial is square-free for instance when **A** represents the Frobenius operator σ of F_{q^n} over F_q, when n is prime to the characteristic of F_q.

Next we generalize the construction for the Frobenius map to any n. Therefore we write $n = n_1 n_2$ where $n_2 = p^t$, where p, which does not divide n_1, is the characteristic of the field and we give a very simple proof of how a cyclic vector of F_{q^n} over F_q is merely the product in F_{q^n} of a cyclic vector of $\mathsf{F}_{q^{n_1}}$ over F_q by one of $\mathsf{F}_{q^{n_2}}$ over F_q. Let us first observe that a straightforward probabilistic search could easily give the result for $\mathsf{F}_{q^{n_2}}$ since the number of normal elements in that field is $(1 - \frac{1}{q})q^{n_2}$. This is because if γ is cyclic, then $f(\sigma)(\gamma)$ is cyclic in its turn if and only if $f(1) \neq 0$. Thus the straighforward probabilistic algorithm succeeds in $O(n^3)$ steps on the average. That is the cost of verifying whether a random element in the field is cyclic. However a cyclic vector can be deterministically obtained in $O(n^3)$ elementary operations as will be shown at the end of Section 9. The whole procedure ends in a *deterministic algorithm* for computing a cyclic vector of F_{q^n} over F_q and from there a normal basis for *any* n *in* $O(n^3)$ operations in F_q.

6.1.2 A cyclic element for the composite of two fields with relatively prime degrees over F_q

For this subsection, we put $n = n_1 n_2$ where $gcd(n_1, n_2) = 1$.

Notation 3 *In this section,* $\mathsf{F}_{q^{n_1}}$ *is denoted by* K_1 *and* $\mathsf{F}_{q^{n_2}}$ *is denoted by* K_2. *Also* F_{q^n} *is denoted by* K *and* F_q *is denoted by* k.

From our assumption, K is the composite of K_1 and K_2. Recall that every element in K can be written in the form

$$\sum_{i \in [0, n_2[} \alpha_{n_i} \beta_{n_i}, \tag{23}$$

where all α_{n_i} belong to K_1 and all β_{n_i} belong to K_2. The reader may for example refer to A.Albert [4, p. 101, Theorem 10]. But this is straightforward from the fact that if $K_2 = \mathsf{k}(\theta_2)$ then $K_2 \subset K_1(\theta_2)$. We thus have that $K_1(\theta_2) \supset K$ and since $\theta_2 \in K$, then $K = K_1(\theta_2)$. We now recall a basic fact among the properties of Galois extensions.

Notation 4 *We write* $[\sigma]$ *for the group generated by* σ *which is the Galois group of* K *over* k.

Property 6 *The Galois group of* K *over* K_1 *is the subgroup* $[\sigma_2]$ *of* $[\sigma]$ *of order* n_2 *and the Galois group of* K_1 *over* k *is the subgroup* $[\sigma_1]$ *of* $[\sigma]$ *of order* n_1.

The corresponding property for K_2 is obtained by interchanging 1 and 2 in the statement.

We now state the property which allows to obtain a normal basis for any **K** in two steps. In step i a normal basis for $\mathbf{K_i}$ in constructed, $i = 1, 2$. The algorithm for the first basis is completely different from the one for the other. But the algorithms in both cases have

complexity $O(n^3)$. That property is well known and we recall the few lines of its proof. The reader will find more on the topic in [5].

Property 7 *Let θ_i be a cyclic element of K_i over* k, $i = 1, 2$. *Then $\theta = \theta_1\theta_2$ is a cyclic element of K over* k.

Proof: Under our assumption, we can write

$$d_1 n_1 + d_2 n_2 = 1.$$

Hence $\sigma = \sigma_1\sigma_2$ where $\sigma_2 = \sigma^{d_1 n_1}$ is a generator of the subgroup of order n_2 of $[\sigma]$ since $gcd(d_1, n_2) = 1$. Similarly $\sigma_1 = \sigma^{d_2 n_2}$ generates the subgroup of order n_1 of $[\sigma]$. By (23) and under our assumption then we know that $\{\sigma_1^i\theta_1.\sigma_2^j\theta_2\}_{i\in[0,n_1[,j\in[0,n_2[}$ span the whole K over k. We are thus left with verifying that $\sigma_1^i\theta_1.\sigma_2^j\theta_2$ is $\sigma^l\theta_1\theta_2$ for some integer l. For let $i' \equiv n_2^{-1} i \bmod n_1$ and $j' \equiv n_1^{-1} j \bmod n_2$. Let then l be the integer $i'n_2 + j'n_1$. We have that

$$\sigma^l = (\sigma_1\sigma_2)^l = \sigma_1^{i'n_2}\sigma_2^{j'n_1} = \sigma_1^i\sigma_2^j.$$

Then

$$\sigma^l\theta = \sigma^l\theta_1\theta_2 = \sigma_1^i\sigma_2^j\theta_1.\sigma_1^i\sigma_2^j\theta_2 = \sigma_1^i\theta_1.\sigma_2^j\theta_2.$$

□

For practical implementation it is important to note that θ can be computed with no *presentation* of $\mathbb{F}_{q^n}$. In fact we are able to compute the minimal polynomial of θ. Therefore we here describe an algorithm which is suggested by M.Mignotte [19, p. 137]

Let P_1 denote the minimal polynomial of θ_1, P_2 the minimal polynomial of θ_2 and finally $\tilde{P}_2(X, Y) = X^{n_2} P_2(\frac{Y}{X})$. Consider the following resultant:

$$\begin{aligned} R(Y) &= Res_X(\tilde{P}_2(X, Y), P_1(X)) \\ &= \prod_{\beta, P_1(\beta)=0} \tilde{P}_2(\beta, Y). \end{aligned}$$

We have that $R(Y) = 0$ if and only if there exists β such that

$$\begin{cases} P_1(\beta) = 0 \\ \tilde{P}_2(\beta, Y) = 0 \end{cases} \Leftrightarrow \begin{cases} P_1(\beta) = 0 \\ \beta^{n_2} P_2(\frac{Y}{\beta}) = 0 \end{cases}$$

that is, $R(Y) = 0$ if and only if Y is a product of a root of P_1 and a root of P_2. Thus the polynomial $R(Y)$ has $n_1 n_2$ roots, which is the number of conjugates of $\theta_1\theta_2$. Since $\theta = \theta_1\theta_2$ is a root of $R(Y)$, the polynomial $R(Y)$ is the minimal polynomial of θ.

6.1.3 The case where the characteristic polynomial is square-free

Let us introduce some definitions.

Theorem 14 *[9, Ch. VII §3 th. 2] For all* **A** *in M_n*(k), *there exists a vector v in* $\mathbf{k}^n$ *such that $\pi_v(X) = \pi(X)$ where $\pi(X)$ is the minimal polynomial of* **A**.

Definition 14 *Let* $\mathbf{A}$ *be a matrix in* $M_n(\mathrm{k})$. *A vector* v *in* k^n *such that* $\pi_v(X) = \pi(X)$, *where* $\pi(X)$ *is the minimal polynomial of* $\mathbf{A}$, *is called a cyclic vector for* $\mathbf{A}$.

First we here show how to compute a cyclic vector at cost $O(m^3+m^2n^2)$ for a square matrix $\mathbf{A}$ whose characteristic polynomial is square-free. This implies that the minimal polynomial of $\mathbf{A}$ equals its characteristic polynomial. Also the minimal polynomials $f_k(X)$ of the diagonal companion matrices of a Shift-Hessenberg form for $\mathbf{A}$ are pairwise relatively prime.

6.2 Technical lemmas

Notation 5 *Given a vector* v *in* k^n, *the vector of size* n_I, *projection of* v *into* k^{B_I}, *is denoted by* v_{B_I}. *We denote by* $v^\star_{B_I}$ *the unique vector of* k^n *such that its projection into* k^{B_I} *equals* v_{B_I} *and such that its projection into* k^{B_J} *is 0, where* J *is the complementary set of* I *in* $[1,n]$: $(v^\star_{B_I})_{B_J} = 0$.

The following lemma sets up the recurrence which ends in the sougth for cyclic vector. We state the lemma for a general matrix $\mathbf{A}$ although we aim to finally exploit the Shift-Hessenberg form for the computations. In particular the recurrence needs the matrix to be split into blocks as it is the case for the Shift-Hessenberg form.

Lemma 12 *Let* $\mathbf{A}$ *be a block matrix with the form*

$$\begin{bmatrix} \mathbf{A}_{B_1,B_1} & \mathbf{A}_{B_1,B_2} \\ 0 & \mathbf{A}_{B_2,B_2} \end{bmatrix}$$

and v_{B_1}, v_{B_2} *be cyclic vectors for* $\mathbf{A}_{B_1,B_1}$ *and* $\mathbf{A}_{B_2,B_2}$ *respectively, matrices with respective minimal polynomials* $f_1(X)$ *and* $f_2(X)$. *If* $f_1(X)$ *and* $f_2(X)$ *are relatively prime, then the relations*

$$v_{B_2} = u_{B_2} \tag{24}$$
$$v_{B_1} = f_2(\mathbf{A}_{B_1,B_1})u_{B_1} + (f_2(\mathbf{A})u^\star_{B_2})_{B_1}. \tag{25}$$

can be solved at $u = (u_{B_1}, u_{B_2})$ *and the unique solution is a cyclic vector for* $\mathbf{A}$.

Proof: The leading idea is that if u is cyclic for $\mathbf{A}$ then u_{B_2} should be cyclic for $\mathbf{A}_{B_2,B_2}$. Moreover we then have that $f_2(\mathbf{A})u_{B_2} = 0$ and $f_2(\mathbf{A})u$ will next be annihilated by the action of $f_1(\mathbf{A}_{B_1,B_1})$. We have to develop that argument formally. Since by hypothesis $f_1(X)f_2(X)$ is the minimal polynomial of $\mathbf{A}$, we then have to prove that $f_1(X)f_2(X)$ is the minimal polynomial of the restriction of $\mathbf{A}$ to u. Assume that $p(\mathbf{A})u = 0$ for a non-zero polynomial $p(X)$ with minimal degree. Then $p(X)$ is a divisor of $f_1(X)f_2(X)$ and we must have that $p(X) = p_1(X)p_2(X)$ with the condition that $p_1(X) \mid f_1(X)$, $p_2(X) \mid f_2(X)$ and $(p_1(X), p_2(X)) = 1$. We then have the following implications.

$$p(\mathbf{A})u = 0 \Rightarrow (p(\mathbf{A})u)_{B_2} = 0 \tag{26}$$
$$\Rightarrow p(\mathbf{A}_{B_2,B_2})u_{B_2} = 0 \tag{27}$$
$$\Rightarrow p_1(\mathbf{A}_{B_2,B_2})p_2(\mathbf{A}_{B_2,B_2})u_{B_2} = 0. \tag{28}$$

Since $(p_1(X), f_2(X)) = 1$, there exists $h_1(X)$ such that $p_1(X)h_1(X) = 1 \pmod{f_2(X)}$, i.e.

$$h_1(\mathbf{A}_{B_2,B_2})p_1(\mathbf{A}_{B_2,B_2}) = I_n$$

Applying $h_1(\mathbf{A}_{B_2,B_2})$ on both sides of (28) we get

$$p_2(\mathbf{A}_{B_2,B_2})u_{B_2} = 0.$$

This implies that $f_2(X) \mid p_2(X)$ because u_{B_2} is a cyclic vector for $\mathbf{A}_{B_2,B_2}$. Thus $p_2(X) = f_2(X)$.

For the first block of coordinates, we have the following implications

$$p(\mathbf{A})u = 0 \Rightarrow (p(\mathbf{A})u)_{B_1} = 0 \tag{29}$$

$$\Rightarrow (p(\mathbf{A})u^*_{B_1})_{B_1} + (p(\mathbf{A})u^*_{B_2})_{B_1} = 0. \tag{30}$$

But

$$p(\mathbf{A}) = p_1(\mathbf{A})p_2(\mathbf{A}) = p_1(\mathbf{A})f_2(\mathbf{A}) \tag{31}$$

and

$$f_2(\mathbf{A})_{B_2} = 0. \tag{32}$$

Thus

$$\forall x \in \mathrm{k}^n, (f_2(\mathbf{A})x)_{B_2} = 0 \tag{33}$$

and

$$p_1(\mathbf{A})p_2(\mathbf{A})x = p_1(\mathbf{A}_{B_1,B_1})f_2(\mathbf{A})_{B_1}x. \tag{34}$$

We finally have that $p(\mathbf{A})u = 0$ writes

$$p_1(\mathbf{A}_{B_1,B_1})(f_2(\mathbf{A}_{B_1,B_1})u_{B_1} + (f_2(\mathbf{A})u^*_{B_2})_{B_1}) = 0 \tag{35}$$

By hypothesis, $f_2(\mathbf{A}_{B_1,B_1})u_{B_1} + (f_2(\mathbf{A})u^*_{B_2})_{B_1}$ is cyclic for $\mathbf{A}_{B_1,B_1}$. Then by (35), $f_1(X) \mid p_1(X)$. Thus $p_1(X) = f_1(X)$ which gives the proof. □

Remark 5 *Solving equations (24) (25) needs three main computations. Matrix $f_2(\mathbf{A}_{B_1,B_1})$ is to be computed, then $w_{B_1} = (f_2(\mathbf{A})u^*_{B_2})_{B_1}$ is a vector to be computed, and finally the system $f_2(\mathbf{A}_{B_1,B_1})u_{B_1} = v_{B_1} - w_{B_1}$ is to be solved.*

We however observe the striking fact that those computations can be performed at low cost. Therefore we point out how equation (25) can be solved very cheaply.

Lemma 13 *A solution u to equations (24) and (25) may be computed in $O(n^3)$ elementary operations.*

Proof: Proceed as follows. First compute $w_{B_1} = (f_2(\mathbf{A})u^*_{B_2})_{B_1}$. This done at cost $O(n^3)$. Then solve equation (25) by finding an inverse $h_2(X)$ of $f_2(X)$ mod $f_1(X)$. Then the solution u_{B_1} is given by

$$u_{B_1} = h_2(\mathbf{A}_{B_1,B_1})(v_{B_1} - w_{B_1})$$

which is evaluated with complexity $O(n^3)$. □

6.3 The naïve recurrence

Let us first recall that we denoted by $\mathbf{H}_{B_{\geq k}}$ the square submatrix obtained from matrix $\mathbf{H}$ using blocks starting at k^{th} block

$$\mathbf{H}_{B_{\geq k}} = \begin{bmatrix} \mathbf{H}_{B_k,B_k} & \mathbf{H}_{B_k,B_{k+1}} & \cdots & \mathbf{H}_{B_k,B_m} \\ & \mathbf{H}_{B_{k+1},k+1} & \cdots & \mathbf{H}_{B_{k+1},n} \\ & & \ddots & \\ & & & \mathbf{H}_{B_m,B_m} \end{bmatrix}$$

Notation 6 *We denote by* $u_{B_{\geq k}}$ *a cyclic vector for* $\mathbf{H}_{B_{\geq k}}$. *We moreover assume that all diagonal blocks have pairwise relatively prime characteristic polynomials.*

First step: Compute u_{B_m}. $\mathbf{H}_{B_m,B_m}$ is a companion matrix, the vector ${}^t(1,0,\ldots,0)$ is a cyclic vector for $\mathbf{H}_{B_m,B_m}$.
Iterative step: Suppose that the problem has been solved for $\mathbf{H}_{B_{\geq k+1}}$, i.e. we have a vector $u_{B_{\geq k+1}}$ which is cyclic for $\mathbf{H}_{B_{\geq k+1}}$.

Using Lemma 12, we will construct $u_{B_{\geq k}} = (u_{B_k}, u_{B_{\geq k+1}})$ which is cyclic for $\mathbf{H}_{B_{\geq k}}$ as follows.

The minimal polynomial $f_k(X)$ of $\mathbf{H}_{B_k,B_k}$, and $u_{B_{\geq k+1}}$ are now at disposal.

We have that

$$((f_{k+1}f_{k+2}\cdots f_m)(\mathbf{H})(u^\star_{B_{\geq k+1}}))_{B_{\geq k+1}} = 0 \tag{36}$$

Now denote by w_{B_k} the vector

$$w_{B_k} = ((f_{k+1}f_{k+2}\cdots f_m(\mathbf{H}))(u^\star_{B_{\geq k+1}}))_{B_k}$$

By Lemma 12, the following relation is to be solved at u_{B_k} .

$$(f_{k+1}f_{k+2}\cdots f_m)(\mathbf{H}_{B_k,B_k})u_{B_k} + w_{B_k} = v^\dagger \tag{37}$$

with a given $v^\dagger$ cyclic for $\mathbf{H}_{B_k,B_k}$. For instance $v^\dagger$ can be chosen as ${}^t(1,0\ldots0)$.

The polynomial $f_{k+1}(X)\cdots f_m(X)$ is prime to $f_k(X)$ and has as inverse $h_k(X)$ mod $f_k(X)$. We thus have that

$$u_{B_k} = h_k(\mathbf{H}_{B_k,B_k})(v^\dagger - w_{B_k})$$

6.4 An upper bound on the complexity

We now evaluate the number of operations to be performed to achieve the recurrence presented in 6.3. The most expensive calculations lie in computing the vectors $w_{B_{m-1}}, w_{B_{m-2}}, \ldots, w_{B_1}$, which are obtained in succession, starting with w_{B_m} which can be chosen as $v^\dagger$ in the previous subsection.

$$\begin{aligned} w_{B_{m-1}} &= (f_m(\mathbf{H})u^\star_{B_m})_{B_{m-1}} \\ w_{B_{m-2}} &= ((f_{m-1}f_m)(\mathbf{H})u^\star_{B_{\geq m-1}})_{B_{m-2}} \\ &\vdots \\ w_{B_k} &= ((f_{k+1}f_{k+2}\cdots f_m)(\mathbf{H})u^\star_{B_{\geq k+1}})_{B_k} \end{aligned}$$

Computing each vector w_{B_k} consists mainly in applying at most n times matrix $\mathbf{H}$ at vectors with n components. The cost is $n.mn$ for each of the m values of k. Moreover each u_{B_k} needs $O(n^2 m)$ steps and a separate cost of $O(n_k^2)$ is required for computing each of m gcd's. Taking into account the construction of $\mathbf{H}$ itself, this amounts to $O(m^2n^2 + n^3)$ elementary operations.

Theorem 15 *If the characteristic polynomial of the matrix* $\mathbf{A}$ *is square-free, the recurrence described in 6.3 ends in a cyclic vector for* $\mathbf{A}$ *on the data of a Shift-Hessenberg form* $\mathbf{H}$ *of* $\mathbf{A}$ *at cost* $O(m^2n^2 + n^3)$.

Corollary 8 *If the characteristic polynomial of the matrix* $\mathbf{A}$ *is square-free, a cyclic vector for* $\mathbf{A}$ *can be obtained in* $O(n^3 + m_{\mathbf{A}}^2 n^2)$ *steps.*

Remark 6 *Note again that the worst case complexity is* $O(n^4)$ *for* $m = n$.

7 Obtaining a cyclic vector in $O(n^3)$ elementary operations

The previous procedure is not efficient for large m. We thus develop a more sophisticated procedure, whose complexity is $O(n^3)$, for any value of m.

The present algorithm computes a cyclic vector for a matrix whose minimal polynomial is square-free. The algorithm is rather sophisticated and uses a "divide-and-conquer" approach as in Section 3, we then first present its global structure, before going into details. We also set out separately a technique of splitting, and finally give the complete description.

7.1 Overall strategy

First a Shift-Hessenberg form for the given matrix is to be computed. Then our strategy is some kind of "divide-and conquer" method on the Shift-Hessenberg matrix, by splitting it into two parts, whose sizes remain under control. We use the same notations as in the previous section. The matrix $\mathbf{H}$ has the following form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{B_1,B_1} & \mathbf{H}_{B_1,B_2} & \cdots & \mathbf{H}_{B_1,B_m} \\ 0 & \mathbf{H}_{B_2,B_2} & \cdots & \mathbf{H}_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{H}_{B_m,B_m} \end{bmatrix}$$

We introduce some notation.

Notation 7 *For every* $I \subset [1,n], J \subset [1,n]$, *we denote by* $\mathbf{H}_{I,J}$ *the sub-matrix formed with the rows of* $\mathbf{H}$ *in* I *and the columns of* $\mathbf{H}$ *in* J. *The size of* I *is denoted by* n_I. *Whenever* I *is reduced to a block* B_k *then the size of* I *is denoted by* n_k.

Roughly, the splitting consists in finding a matrix $\mathbf{H}_{split}$ equivalent to $\mathbf{H}$ with the form

$$\mathbf{H}_{split} = \begin{bmatrix} \mathbf{H}'_{B_I,B_I} & \mathbf{H}'_{B_I,B_J} \\ 0 & \mathbf{H}'_{B_J,B_J} \end{bmatrix}, \tag{38}$$

which moreover is a Shift-Hessenberg matrix, such that $n_I \leq \frac{2}{3}n, n_J \leq \frac{2}{3}n$. We recursively apply the algorithm on both matrices $\mathbf{H}'_{B_I,B_I}$ and $\mathbf{H}'_{B_J,B_J}$, in order to find v_{B_I}, v_{B_J} which are cyclic vectors of $\mathbf{H}'_{B_I,B_I}$ and $\mathbf{H}'_{B_J,B_J}$ respectively.

It remains to compute a vector u' cyclic for $\mathbf{H}_{split}$, v_{B_I} and v_{B_J} being known. Changing the current basis for the original one , we finally transform u' into a cyclic vector u for $\mathbf{H}$.

7.2 The splitting

We give a lemma for splitting the matrix into two submatrices. Before stating this lemma, we explain a technical but important phenomenom that appears when permuting rows and columns of Shift-Hessenberg matrices in order to move the blocks.

Consider the following Shift-Hessenberg matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{B_1,B_1} & \mathbf{H}_{B_1,B_2} & \cdots & \mathbf{H}_{B_1,B_k} & \cdots & \mathbf{H}_{B_1,B_m} \\ 0 & \mathbf{H}_{B_2,B_2} & \cdots & \mathbf{H}_{B_2,B_k} & \cdots & \mathbf{H}_{B_2,B_m} \\ \vdots & & \ddots & \vdots & \cdots & \vdots \\ \vdots & & & \mathbf{H}_{B_k,B_k} & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \mathbf{H}_{B_m,B_m} \end{bmatrix}$$

Let us perform the permutation of rows and columns which places $\mathbf{H}_{B_k,B_k}$ in the upper-left corner. This leads to the matrix $\mathbf{H}_{swap}$:

$$\mathbf{H}_{swap} = \begin{bmatrix} \mathbf{H}_{B_k,B_k} & 0 \cdots 0 & 0 & \mathbf{H}_{B_k,B_{>k}} & \\ \mathbf{H}_{B_{[2,k-1]},B_k} & \mathbf{H}_{B_{[2,k-1]},B_{[2,k-1]}} & \mathbf{H}_{B_{[2,k-1]},B_1} & \mathbf{H}_{B_{[2,k-1]},B_{>k}} & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{B_1,B_k} & \mathbf{H}_{B_1,B_{[2,k-1]}} & \mathbf{H}_{B_1,B_1} & \mathbf{H}_{B_1,B_{>k}} & \\ \mathbf{H}_{B_{>k},B_k} & \mathbf{H}_{B_{>k},B_{[2,k-1]}} & \mathbf{H}_{B_{>k},B_1} & \mathbf{H}_{B_{>k},B_{>k}} & \end{bmatrix}$$

We now use the algorithm for computing a Shift-Hessenberg form for $\mathbf{H}_{split}$. This leads to the matrix

$$\mathbf{H}' = \begin{bmatrix} \mathbf{H}'_{B_1,B_1} & \mathbf{H}'_{B_1,B_2} & \cdots & \mathbf{H}'_{B_1,B_k} & \cdots & \mathbf{H}'_{B_1,B_m} \\ 0 & \mathbf{H}'_{B_2,B_2} & \cdots & \mathbf{H}'_{B_2,B_k} & \cdots & \mathbf{H}'_{B_2,B_m} \\ \vdots & & \ddots & \vdots & \cdots & \vdots \\ \vdots & & & \mathbf{H}'_{B_k,B_k} & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \mathbf{H}'_{B_m,B_m} \end{bmatrix}.$$

The next lemma establishes a relation between the companion polynomial of block $\mathbf{H}'_{B_1,B_1}$ and the companion polynomial of block $\mathbf{H}_{B_k,B_k}$.

Lemma 14 *Let f_k be the companion polynomial of block $\mathbf{H}_{B_k,B_k}$ of the matrix $\mathbf{H}$, and let f'_1 be the companion polynomial block $\mathbf{H}'_{B_1,B_1}$ of the matrix $\mathbf{H}'$ obtained in the previous transformation. We have that f_k divides f'_1.*

Let us introduce the following

Notation 8 *We denote by* ϵ_k *the vector from the basis of* k^n *such that* $(\epsilon_k)_{B_k} = (1, 0, \ldots, 0)$.

Proof: We have that f_k divides the minimal polynomial of $\mathbf{H}$ relatively to ϵ_k. Swapping from $\mathbf{H}$ to $\mathbf{H}_{swap}$ is placing vector ϵ_k as the first vector of the new basis.

The Shift-Hessenberg reduction algorithm computes a matrix whose first block is a companion matrix whose companion polynomial is the minimal polynomial of the first vector. Thus f'_1 is the minimal polynomial of ϵ_k, which is a multiple of f_k. □

Now we can state our important lemma for splitting Shift-Hessenberg matrices:

Lemma 15 (Splitting the matrix) *Let* $\mathbf{H}$ *be a Shift-Hessenberg matrix. It is always possible to find a Shift-Hessenberg matrix* $\mathbf{H}_{split}$ *and an invertible matrix* $\mathbf{P}$ *such that* $\mathbf{H} = \mathbf{P}\mathbf{H}_{split}\mathbf{P}^{-1}$ *with* $\mathbf{H}_{split}$ *of the form*

$$\mathbf{H}_{split} = \begin{bmatrix} \mathbf{H}'_{B_I,B_I} & \mathbf{H}'_{B_I,B_J} \\ 0 & \mathbf{H}'_{B_J,B_J} \end{bmatrix} \tag{39}$$

and such that one of those three possibilities occur

1. *either* $\mathbf{H}'_{B_I,B_I}$ *is a companion block with size* $\geq \frac{2}{3}n$, *and* $\mathbf{H}'_{B_J,B_J}$ *has size* $\leq \frac{1}{3}n$.
2. *or* $\mathbf{H}'_{B_I,B_I}$ *is a companion block with size* $\leq \frac{2}{3}n$, *and* $\mathbf{H}'_{B_J,B_J}$ *has size* $\leq \frac{2}{3}n$.
3. *or both blocks* $\mathbf{H}'_{B_I,B_I}$ *and* $\mathbf{H}'_{B_J,B_J}$ *are nothing else that Shift-Hessenberg matrices with size smaller than* $\frac{2}{3}n$.

The computation of $\mathbf{H}_{split}$ *and* $\mathbf{P}$ *can be performed in* $O(n^3)$ *operations.*

Proof: Two main distinct cases are first considered.

$\exists k \in [1, m] \mid n_k \geq \frac{2}{3}n$. Choose $I = B_k$, $J = [1, m] \setminus I$. We have that $n_J \leq \frac{1}{3}n$ but the block B_k may not be the first block. By permutations of rows and colums, block B_k is put in the first place. This gives a matrix $\mathbf{H}_{swap}$ which is not Shift-Hessenberg. We now can clean up matrix $\mathbf{H}_{swap}$ by applying the reduction algorithm producing a Shift-Hessenberg matrix. The size of the first block can only grow, by Lemma 14, and then remains larger than $\frac{2}{3}n$. This gives matrix $\mathbf{H}_{split}$ shaped as in Case 1 at cost $O(n^3)$.

$\forall j \in [1, m], n_j < \frac{2}{3}n$. Suppose first that all n_i are smaller than $\frac{1}{3}n$. In the sequence of sets $I_i = \{1, 2, \ldots i\}$, we choose the largest, I_{i_0} with the condition that $\sum_{j \in I_i} n_j < \frac{2}{3}n$. Then $I = B_1 \cup B_2 \cdots B_{i_0}$ and $J = B_{i_0+1} \cup B_{i_0+2} \cdots \cup B_m$ both satisfy $n_I \leq \frac{2}{3}n$ and $n_J \leq \frac{2}{3}n$. Indeed since $n_{J \setminus B_{i_0+1}} < \frac{1}{3}n$, we have that $n_J < \frac{1}{3}n + n_{i_0+1} \leq \frac{2}{3}n$. Then the matrix $\mathbf{H}_{split}$ is the unchanged matrix $\mathbf{H}$. This is case 3.

If there exists $n_k \geq \frac{1}{3}n$, we choose $I = B_k$, $J = [1, m] \setminus I$. We have $n_I \leq \frac{2}{3}n$, $n_J \leq \frac{2}{3}n$. By swapping rows and columnmns, we put the block $\mathbf{H}_{I,I}$ in the first place, then clean up the resulting matrix by the Shift-Hessenberg reduction algorithm in $O(n^3)$ steps. The first block can only grow. As a result the size of the remaining block remains lower than $\frac{2}{3}n$; if the size of the first block is larger than $\frac{2}{3}n$, then we are in Case 1, else we are in Case 2. □

7.3 The algorithm itself

We now present the complete algorithm for computing a cyclic vector for a matrix **A** such that its minimal polynomial is square-free.

Step 1*: computation of a Shift-Hessenberg form of A. As stated in Theorem 11, this is done in $O(n^3)$ operations. This step needs only to be performed once, in the first decomposition, and is not needed in the remaining recursive applications of the algorithm.

Step 2: splitting the matrix. We perform the splitting indicated by Lemma 15, and obtain two submatrices $\mathbf{H'}_{B_I,B_I}$ and $\mathbf{H'}_{B_J,B_J}$.

We recursively apply the algorithm on all submatrices which occur with size $\leq \frac{2}{3}n$.

Step 3: reconstruction of a cyclic element in a new basis. We get the two vectors u_{B_1} and u_{B_2} for (24) and (25) from applying the algorithm at $\mathbf{H'}_{B_I,B_I}$ and $\mathbf{H'}_{B_J,B_J}$. By Lemma 13 we can construct a cyclic element for $\mathbf{H}_{split}$ at cost $O(n^3)$.

Step 4: reconstruction of the cyclic element in the original basis. From a cyclic vector of $\mathbf{H}_{split}$, changing basis gives a cyclic vector for **H** at cost $O(n^3)$.

Step 5*: reverting to the original basis. From a cyclic vector for **H**, we compute a cyclic vector for **A** by changing basis. This costs $O(n^3)$, and is performed only once, at the end of the algorithm.

7.4 The complexity

The cost of each step is here evaluated.

Step 1*: This is done at a cost $O(n^3)$, only once.

Step 2: The splitting costs $O(n^3) = a_1 n^3$.

Step 3: The reconstruction costs $O(n^3) = a_2 n^3$.

Step 4: Changing basis is done in $O(n^3)$ steps.

Step 5*: Changing basis from **H** to **A** for obtaining the cyclic vector in the original basis. This is done in $a_3 n^3$ steps only once.

Notice that obtaining a cyclic vector for a companion matrix is done at negligeable cost.

Only steps 2,3 and 4 are applied recursively . The total cost for those steps is $(a_1+a_2)n^3 = an^3$. The important point is that those steps are recursively applied at matrices whose sizes are reduced by a factor $\frac{2}{3}$.

Let us assume that the cost $C(m)$ of the algorithm is at most γm^3 for all $m < n$, we then have that

$$\begin{aligned} C(n) &\leq an^3 + 2C(\tfrac{2}{3}n) \\ &\leq an^3 + 2\left(\tfrac{2}{3}\right)^3 \gamma n^3 \end{aligned}$$

This leads to $C(n) \leq \gamma n^3$ with:

$$\gamma = \frac{a}{1 - 2(\frac{2}{3})^3} \approx 2.45a$$

Let us now conclude.

Theorem 16 *Given a matrix* $\mathbf{A} \in M_n(k)$ *whose minimal polynomial is square-free, a cyclic vector for* $\mathbf{A}$ *can be computed in* $O(n^3)$ *elementary operations. This is a worst-case complexity.*

Proof: The whole cost is indeed $C(n) + 2a_3n^3$ i.e. the cost of the recursive algorithm plus the cost of obtaining a Shift-Hessenberg form plus the cost of reverting to the original basis. □

Corollary 9 *When* n *is prime to* p*, it is possible to compute a normal basis of* F_{q^n} *in* $O(n^3)$ *elementary operations on the data of a matrix representing the Frobenius map.*

Proof: The minimal polynomial of the Frobenius map is $X^n - 1$, which is square-free when $\gcd(n, q) = 1$. Given the matrix $\mathbf{F}_n$ of the Frobenius map (computed in $O(n^3)$), we are able to compute a cyclic vector for the Frobenius map in $O(n^3) \log q$. This vector is a normal element. □

Theorem 17 *For all* n*, a normal basis of* F_{q^n} *can be computed deterministically in* $O(n^3)$ *elementary operations.*

Proof: Merging Property 7 and the algorithm in Section 9.5 yields the result. □

8 An easy probabilistic algorithm

The investigation in [10] establishes that the expected number of operations in F_q for obtaining an element in F_{q^n} whose conjugates are linearly independent is essentially the cost of computing the conjugates and then verifying if they are independent. Given the matrix representing the Frobenius map in the given basis, this can be done very simply in $O(n^3)$ elementary operations in F_q as follows.

1. The matrix representing the Frobenius map $\mathbf{F}$ in the given bases of F_{q^n} over F_q is known.

2. An element α from F_{q^n} is taken at random. A new basis is formed by substituting that element to the first one in the previous basis and the corresponding representation $\mathbf{F}'$ of $\mathbf{F}$ is computed.

3. The Sparse-Hessenberg algorithm is applyied to $\mathbf{F}'$ which gives $\mathbf{F}''$. Remember that the first basis element is the same in the basis in which $\mathbf{F}''$ or $\mathbf{F}'$ is represented, i.e. α. If $\mathbf{F}''$ is a companion matrix, then the columns of that matrix are α and its conjugates in the basis for which $\mathbf{F}$ is represented by $\mathbf{F}''$. If not, a new α is selected from F_{q^n}.

9 Computation of the Frobenius Form

9.1 Definitions and Notations

Let $\mathbf{T}$ be an operator. We will consider $\mathbf{k}^n$ equipped with the natural structure of $\mathbf{k}[X]$-module induced by $\mathbf{T}$.

Notation 9 *Let p be a polynomial and v a vector, we use the module notation pv for $p(\mathbf{T})v$.*

It is sometime convenient to use the same notation pv when p is a polynomial $p(\mathbf{T}_{B_i})$ evaluated at the restriction of $\mathbf{T}$ to the subspace k^{B_i} and when v is a vector in k^{B_i}. We do so without mentioning it.

Furthermore we need a specific notation for the columns of a Shift-Hessenberg matrix.

Notation 10 *Let* $\mathbf{H}$ *be a Shift-Hessenberg matrix*

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{B_1,B_1} & \mathbf{H}_{B_1,B_2} & \cdots & \mathbf{H}_{B_1,B_m} \\ 0 & \mathbf{H}_{B_2,B_2} & \cdots & \mathbf{H}_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{H}_{B_m,B_m} \end{bmatrix}.$$

We denote by ϵ_i the unit vector from k^n such that $(\epsilon_i)_{B_i} = {}^t(1,\ldots,0)$. All columns of $\mathbf{H}$ *have the form $\mathbf{T}^i(\epsilon_j)$ for suitable i,j.*

We also set $e_i = f_i\epsilon_i$. Informally, e_i is seen to be the vector "above" block i in the Shift-Hessenberg form complemented to n components with zeros.

We now recall the definition of the Frobenius form, which is known as Rational Canonical form, briefly RCF , as well.

Definition 15 *The Frobenius form has the following structure*

$$\mathbf{F} = \begin{bmatrix} C_{p_1} & 0 & \cdots & 0 \\ 0 & C_{p_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{p_t} \end{bmatrix}$$

where $p_1 \mid p_2 \mid \cdots \mid p_t$. The polynomials p_i, $i = 1,\ldots,t$ are the elementary divisors of the matrix.

Theorem 18 *Elementary divisors are invariants for similiraty classes, and a set of elementary divisors characterizes such a class.*

9.2 Preliminary computation

We assume that a block-diagonal matrix $\mathbf{A}$ exhibiting the characteristic subspaces of a given matrix has been computed. From Section 3, this can be obtained in $O(n^{3.5})$ elementary operations, or by using the Shift-Hessenberg form with average complexity $O(n^3)$.

Our goal is to find a Frobenius form for each characteristic subspace. This is because it is easy to recover the Frobenius form from the Frobenius forms of the restrictions to characteristic subspaces. Indeed let us be given a block-diagonal matrix similar to $\mathbf{A}$:

$$\mathbf{D} = \begin{bmatrix} \mathbf{F}_{B_1,B_1} & 0 & \cdots & 0 \\ 0 & \mathbf{F}_{B_2,B_2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{F}_{B_d,B_d} \end{bmatrix}$$

where each matrix $\mathbf{F}_{B_i,B_i}$ is a Frobenius matrix

$$\begin{bmatrix} \mathbf{C}_{p_i^{s_{i,1}}} & 0 & \cdots & 0 \\ 0 & \mathbf{C}_{p_i^{s_{i,2}}} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{C}_{p_i^{s_{i,m_i}}} \end{bmatrix}$$

and where $s_{i,1} \leq s_{i,2} \leq \cdots \leq s_{i,m_i}$, $i = 1, \ldots, d$. We thus have that $p_i^{s_{i,m_i}}$ is the minimal polynomial of $\mathbf{F}_{B_i,B_i}$. Such a matrix $\mathbf{D}$ is called Expanded-Frobenius form. Extensive use of that form is made in Section 4. The subspaces for which the matrix is a companion matrix are denoted by $V_{p_i^{s_{i,1}}}, V_{p_i^{s_{i,2}}} \ldots V_{p_i^{s_{i,m_i}}}$ and a cyclic vector for each of those subspaces is a unit vector denoted by $\epsilon_{p_i^{s_{i,j}}}$.

We consider the subpaces

$$W_1 = V_{p_1^{s_{1,m_1}}} \oplus V_{p_2^{s_{2,m_2}}} \oplus \cdots \oplus V_{p_d^{s_{d,m_d}}}$$
$$W_2 = V_{p_1^{s_{1,m_1-1}}} \oplus V_{p_2^{s_{2,m_2-1}}} \oplus \cdots \oplus V_{p_d^{s_{d,m_d-1}}}$$
$$\vdots$$

formed by taking the exponents of each irreducible polynomial in decreasing order.

Let us now consider the vectors

$$E_1 = \epsilon_{p_1^{s_{1,m_1}}} + \epsilon_{p_2^{s_{2,m_2}}} + \cdots + \epsilon_{p_d^{s_{d,m_d}}}$$
$$E_2 = \epsilon_{p_1^{s_{1,m_1-1}}} + \epsilon_{p_2^{s_{2,m_2-1}}} + \cdots + \epsilon_{p_d^{s_{d,m_d-1}}}$$
$$\vdots$$

It is easily checked that E_i is a cyclic vector for W_i, for each value of i. Each cyclic vector thus defines an invariant subspace such that the minimal polynomial f_{i+1} of $\mathbf{A}$ restricted to the minimal invariant subspace containing E_{i+1} divides f_i, the minimal polynomial of $\mathbf{A}$ restricted to the minimal invariant subspace containing E_i. The matrix representing the restriction of $\mathbf{A}$ to the minimal invariant subspace containing E_i in the basis $\{E_i, \mathbf{A}E_i, \mathbf{A}^2E_i, \ldots\}$ is a companion matrix.

9.3 Computing the Frobenius form for the restriction of the given operator to characteristic subspaces

From now on, we confine ourselve to the case where the characteristic polynomial of the considered matrix is $C(X) = p(X)^r$, with $r \geq 1$, for some polynomial $p(X)$ irreducible.

We apply the reduction process to get a Shift-Hessenberg form $\mathbf{H}$ for the matrix. The matrix $\mathbf{H}$ can be written in the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{B_1,B_1} & \mathbf{H}_{B_1,B_2} & \cdots & \mathbf{H}_{B_1,B_m} \\ 0 & \mathbf{H}_{B_2,B_2} & \cdots & \mathbf{H}_{B_2,B_m} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{H}_{B_m,B_m} \end{bmatrix}.$$

The minimal polynomial f_i of the companion matix $\mathbf{H}_{B_i,B_i}$ is p^{s_i}. Step by step the submatrix of $\mathbf{H}$ with row indices in B_1 will be cleaned up in order to get a matrix similar to $\mathbf{H}$ with two diagonal blocks, the first block being a companion matrix. Once this has been done, the procedure is applied to the remaining block.

The favourable case occurs when each e_i is such that $(e_i)_{B_1} = e_i^\dagger p^{r_i}$, $r_i \geq s_i$, where $e_i^\dagger$ is prime to p. We then introduce the vectors

$$\begin{aligned}
\epsilon_2' &= \epsilon_2 - e_2^\dagger p^{r_2 - s_2}\epsilon_1 \\
\epsilon_3' &= \epsilon_3 - e_3^\dagger p^{r_3 - s_3}\epsilon_1 \\
&\vdots \\
\epsilon_m' &= \epsilon_m - e_m^\dagger p^{r_m - s_m}\epsilon_1
\end{aligned}$$

For those vectors we have that $(p^{s_i}\epsilon_i')_{B_1} = 0$ since $p^{s_i}\epsilon_i = e_i$. The first basis-vector ϵ_1 remains unchanged. As a result they yield a basis in which the matrix has the following form

$$\begin{bmatrix} \mathbf{C}_{p^{s_1}} & 0 \\ 0 & \mathbf{H}' \end{bmatrix},$$

The process is next applied to $\mathbf{H}'$.

Otherwise, there exists i such that $r_i < s_i$. Let ϵ_j be the vector such that $s_j - r_j$ is the largest. We permute the basis vectors in order to have ϵ_j in the first position. By applying the reduction algorithm, we compute a new Shift-Hessenberg form, the first block being a companion matrix, for which the companion polynomial is the minimal polynomial of ϵ_j as proved for Lemma 14. We claim that the exponent of the minimal polynomial of ϵ_j is larger than s_1. The size of the first block has then grown and as a result the sizes of the other blocks had to decrease. The process stops when we have $s_i \leq r_i$ for all i. The first rows can then be cleaned up as above.

We now prove the claim. Assume that $r_j < s_j$. We then have to prove that the minimal polynomial of ϵ_j has degree larger than the minimal polynomial of ϵ_1.
Proof: We first compute $p^{s_j}\epsilon_j$. We do so because $p^{s_j}(\mathbf{H})_{B_j,B_j} = 0$ and consequently $(p^{s_j}\epsilon_j)_{B_j}$ vanishes. This leads to

$$p^{s_j}\epsilon_j = (\epsilon_j)^*_{B_1} + (e_j)^*_{B_2} + \cdots + (e_j)^*_{B_{j-1}},$$

where Notation 5 is used. The coordinates on block B_{j-1} will vanish in their turn, when applying the minimal polynomial $p^{\lambda_{j-1}}$ of $(e_{j-1})^*_{B_{j-1}}$. We have that

$$p^{\lambda_{j-1}}p^{s_j}\epsilon_j = p^{\lambda_{j-1}}(e_j)^*_{B_1} + v(j-2).$$

Here $v(j-2)$ is a vector with support in blocks $B_1 \cup B_2 \cdots \cup B_{j-2}$. We proceed in this way and at each step we get a new relation

$$p^{\lambda_k}\cdots p^{\lambda_{j-1}}p^{s_j}\epsilon_j = p^{\lambda_k}\cdots p^{\lambda_{j-1}}(e_j)_{B_1} + v(k-1).$$

This ends after all other coordinates vanished except those in the first block. We then have that

$$\begin{aligned} p^{\lambda_2}\cdots p^{\lambda_{j-1}}p^{s_j}\epsilon_j &= p^{\lambda_2}\cdots p^{\lambda_{j-1}}(e_j)_{B_1} + v(1) \\ &= p^{\lambda_2}\cdots p^{\lambda_{j-1}}e_j^{\dagger}p^{r_j} + v(1). \end{aligned}$$

Recall that applying the polynomials $g(\mathbf{H})$ to vectors with supports in the first block reduces to computing in $\mathrm{k}[X]/(p^{s_1}(X))$. We thus are left with determining the minimum exponent l such that

$$p^l(p^{\lambda_2}\cdots p^{\lambda_{j-1}}e_j^{\dagger}p^{r_j} + v(1)) = 0 \bmod p^{s_1}$$

and we write $v(1) = p^{r_0}v(1)^{\dagger}$ where $\gcd(p, v(1)^{\dagger}) = 1$.

Two cases are to be considered.

- $r_0 \geq \lambda_2 + \cdots + \lambda_{j-1} + r_j$.

  The exponent l is

  $$l = s_1 - (\lambda_2 + \cdots + \lambda_{j-1} + r_j)$$

  and the exponent of the minimal polynomial of ϵ_j is

  $$l + \lambda_2 + \cdots + \lambda_{j-1} + s_j = s_1 - r_j + s_j > s_1$$

  since $r_j < s_j$.

- $r_0 < \lambda_2 + \cdots + \lambda_{j-1} + r_j$.

  The exponent l is

  $$l = s_1 - r_0 > s_1 - (\lambda_2 + \cdots + \lambda_{j-1} + r_j)$$

  and the exponent of the minimal polynomial of ϵ_j is

  $$\begin{aligned} s_1 - r_0 + \lambda_2 + \cdots + \lambda_{j-1} + s_j &> s_1 - (\lambda_2 + \cdots + \lambda_{j-1} + r_j) + \lambda_2 + \cdots + \lambda_{j-1} + s_j \\ &> s_1 - r_j + s_j > s_1 \end{aligned}$$

□

Remark 7 *The algorithm for the Frobenius form presented here requires the factored characteristic polynomial of the given matrix. It however can be adapted in order to get rid of that requirement. Indeed the companion matrices which are diagonal blocks of a Shift-Hessenberg matrix yields factors of the characteristic polynomial. If the polynomials which show in that way are not powers of a unique polynomial, then by g.c.d. operations the characteristic polynomial can be split into relatively prime factors and a direct sum of invariant subspace is then obtained as in Section 3. Next the algorithm just described can be applied to the restriction of the linear operator to each subspace. Shift-Hessenberg forms are then computed for each restriction and we can proceed until companion matrices exhibit polynomials which all are powers of a unique one, say $q(X)$. The algorithm described is then applied as if $q(X)$ were irreducible and if it should fail, we then would get new factors and then new invariant subspaces and we would proceed as above. This leads to an algorithm which is attractive for fields with zero-characteristic for which factoring polynomials is expensive.*

9.4 Complexity

Either cleaning up the matrix when it is possible, otherwise augmenting the size of the first block is done at cost $O(n^3)$. The number of times those processes are performed is bounded by r. Notice that matrices for changing bases are also obtained. Thus the complexity in the case of a characteristic subspace is bounded by $O(n^3 r)$.

The complexity for all characteristic subspaces is bounded by

$$O(n_1^3 r_1) + O(n_2^3 r_2) + \cdots + O(n_d^3 r_d) \leq O(n^3(r_1 + r_2 + \cdots + r_d))$$

The number $r_1 + r_2 + \cdots + r_d$ is the number of factors of the characteristic polynomial counted with multiplicities. This number is $\log n$ on the average.

Theorem 19 *The Frobenius form of a matrix* $\mathbf{A}$ *and the matrix for changing basis can be computed in* $O(n^3 m_{\mathbf{A}})$*, where* $m_{\mathbf{A}}$ *is the number of factors of the characteristic polynomial of* $\mathbf{A}$*, counted with multiplicities. The asymptotic average complexity over a finite field is* $O(n^3 \log n)$.

Patrick Ozello gives a bound for his algorithm, which is $8n^4 + 2n^3$. The number of irreducible polynomials, counted with multiplicities, was not taken into account for computing this bound, which may be rough.

Remark 8 *Note that the worst case complexity of our algorithm is* $O(n^4)$ *when* $m = n$.

9.5 Normal basis and the Rational Canonical form of a Frobenius map

It easily follows from Definition 15 that computing the RCF leads to exhibiting a cyclic vector. In the case where the considered operator $\mathbf{T}$ is a Frobenius map, then a normal basis is obtained. However, this is not a wise procedure for this purpose in general since the number of factors of $X^n - 1$ over $\mathbf{F}_q$ is far from $\log n$ on the average. Yet, in the particular case where $n = p^t$ where p is the characteristic of the field, then the algorithm presented above is the most efficient. For, we have that $X^n - 1 = (X - 1)^n$ in that case and the minimal polynomial of $\mathbf{T}$ restricted to any single vector is a power of $X - 1$. Considering the matrix $\mathbf{H}$ of Section 9.3, it is easily seen that the minimal polynomial of ϵ_m is $X^n - 1$. For we first observe that $X^n - 1$ is the minimal polynomial of one of the ϵ_i's. Now if ϵ_l were cyclic with $l < m$, we would permute the basis vectors in order to have ϵ_l in the first position. Applying next the reduction algorithm, it is seen that the last rows and columns would remain unchanged and in particular the zero in the subdiagonal located in the column preceding ϵ_m would remain unchanged. This contradicts the fact that ϵ_l is cyclic, since putting it in the first position would lead to a new Shift-Hessenberg form which has to be a companion matrix.

To sum up, a reduction of any representation of the Frobenius map into a Shift-Hessenberg form exhibits ϵ_m which necessarily is cyclic and thus yields a normal basis for $\mathbf{F}_{q^{p^t}}$ over $\mathbf{F}_q$. All other cyclic elements are obtained here by applying polynomial $f(\sigma)$ to a fixed cyclic element, where $f(X)$ runs over all polynomials of degree less than p^t such that $f(1) \neq 0$.

9.6 A distributed algorithm for the minimal polynomial

While proving that the presented algorithm for producing the Frobenius form of any linear operator is well-founded, we had to go through the computation of the minimal polynomials of vectors $\epsilon_i, i = 1, \ldots, m$ of a Shift-Hessenberg $\mathbf{H}$ form in the particular case where the minimal polynomial of $\mathbf{H}$ is some power of an irreducible polynomial. However the technique is applicable to any Shift-Hessenberg form and computing next the least commun multiple of all minimal polynomials produces the minimal polynomial of the given matrix. That algorithm is less efficient than others presented here. However it can be distributed for a huge matrix on several computers, each one dealing with a single ϵ_i. Moreover that algorithm applies to any field $\mathbf{k}$ since knowledge of the characteristic polynomial is not required.

10 Conclusion

The efficiency of the presented algorithms is due to two major procedures here introduced.

The first one is the use of a divide-and-conquer algorithm which splits matrices of size n into submatrices of size $\leq \frac{2}{3}n$. Therefore we obtain a general result: such an algorithm has the same global bounded cost as the one of "dividing" and "recombining"only once. Since we applied that procedure in view of diverse aims, there probably are other opportunities to be found where the same procedure could be exploited.

The second is the use of the Shift-Hessenberg form which is very sparse on the average, and very well reflects algebraic properties of the matrix. It can be computed at low cost and above all it allows one to make the most of the isomorphism from the algebra generated by the given matrix onto an algebra of polynomials by converting operations on matrices into operations on polynomials. What is more, the ring of polynomials dealt with in the last section is a local ring. Advantage is taken of that as it was done in solving equations over a ring in [6].

Considering the results of this paper, a natural question raises. Does there exist a deterministic algorithm for obtaining the Frobenius form of any matrix in $O(n^3)$ elementary operations on the average?

The moral of the present endeavour to obtain good deterministic algorithm is that one should not grow slack after very satisfactory probabilistic algorithms have been conceived. Indeed, the way to deterministic algorithm sheds light on structures and leads to new results as for example the algorithm for the centralizer of a matrix which is presented here.

Acknowledgement.

Daniel Lazard read the first draft of this paper and among very constructive criticisms drew our attention on the important results of Patrick Ozello. His remarks were encouragements to carry on with this venture. We also had a nice opportunity to get informed of the background of the present topic by Joachim von zur Gathen. We had an encouraging conversation with Arnold Schönhage and Jeremy Johnson informed us of a result by Richard Stong which was here generalized for the need of evaluating complexities.

Contents

1 Introduction **3**

2 Computing the characteristic polynomial **6**
2.1 The Hessenberg form of a matrix 6
2.2 Obtaining the characteristic polynomial from a Hessenberg form 7

3 Characteristic subspaces and minimal polynomial in $O(n^{3.5})$. Their construction. **7**
3.1 Characteristic subspaces 8
3.2 Overall strategy 8
3.3 The algorithm 8
3.4 Splitting the characteristic polynomial 9
3.4.1 A general procedure for a recursive partitioning 9
3.4.2 The problem $P(E)$ 10
3.4.3 A general lemma 10
3.4.4 Applying the partitioning procedure to the characteristic polynomial 11
3.5 Computing $P(\mathbf{A})$, $Q(\mathbf{A})$ 12
3.6 The complexity 12

4 The Shift-Hessenberg form and the centralizer of a matrix **13**
4.1 Shift-basis 13
4.2 An algorithm for the centralizer of a matrix 15
4.3 The $\mathbf{k}[X]$-module induced by a matrix 15
4.4 Shift-bases for the Expanded-Frobenius form 16
4.5 From shift-bases to the centralizer of a matrix 17
4.6 The size of the centralizer of a matrix over a finite field 18
4.7 The centralizer of a matrix 18
4.8 The average number of factors of a characteristic polynomial 19
4.9 The Shift-Hessenberg form 24
4.10 Evaluating a polynomial at a Shift-Hessenberg matrix 26
4.11 Linear algebra with a companion matrix 28

5 A direct algorithm for the minimal polynomial **29**
5.1 Nested ideals related to $\mathbf{H}$ 30
5.2 The algorithm for the minimal polynomial of $\mathbf{H}$ 31
5.3 Complexity bounds 31

6 Searching for a cyclic vector **32**
6.1 Normal basis 32
6.1.1 Introduction 32
6.1.2 A cyclic element for the composite of two fields with relatively prime degrees over $\mathbf{F}_q$ 33
6.1.3 The case where the characteristic polynomial is square-free 34

6.2 Technical lemmas 35
6.3 The naïve recurrence 37
6.4 An upper bound on the complexity 37

7 Obtaining a cyclic vector in $O(n^3)$ elementary operations **38**
7.1 Overall strategy 38
7.2 The splitting 39
7.3 The algorithm itself 41
7.4 The complexity 41

8 An easy probabilistic algorithm **42**

9 Computation of the Frobenius Form **42**
9.1 Definitions and Notations 42
9.2 Preliminary computation 43
9.3 Computing the Frobenius form for the restriction of the given operator to characteristic subspaces 44
9.4 Complexity 47
9.5 Normal basis and the Rational Canonical form of a Frobenius map 47
9.6 A distributed algorithm for the minimal polynomial 48

10 Conclusion **48**

References

[1] G. B. Agnew, R. C. Mullin, S. A. Vanstone *Fast exponentiation in $GF(2^n)$*, Advances in Cryptology-Eurocrypt '88, Lecture Notes in Computer Science, Vol 330, Springer-Verlag 1988.

[2] B. Agnew, R. C. Mullin, I. M. Onyszuchuk, and S. A. Vanstone *An implementation of a fast public-key cryptosystem*, J. of Cryptology, Vol 3, N0 2, 1991.

[3] B. Agnew, T. Beth, R. C. Mullin, S. A. Vanstone *Arithmetic operations in $GF(2^n)$*, J.of Cryptology, Vol 6, N0 1, 1993.

[4] A.A. Albert *Fundamental Concept of Higher Algebra* The University of Chicago Press, 1959.

[5] I.F. Blake, X.H. Gao, R.C. Mullin, S.A.Vanstone, T. Yaghoobian *Applications of finite fields* Kluwer Academic Publishers, Boston, Dordrecht, London.

[6] P. Camion, H.B. Mann, L. Levy *Linear Equations over a Commutative Ring* Journal of Algebra Vol **18** 432-446 1971 MR 43-4835

[7] D. Coppersmith and S. Winograd *Matrix multiplication via arithmetic progressions*, J. Symbolic Computation 1990 **9**, 251-280.

[8] R.M. Fano *Transmission of Information*, The M.I.T.Press, Massachusetts Institute of Technology Cambridge, Massachusetts, 1961.

[9] F. R. Gantmacher *The Theory of Matrices*, Vol 1, Chelsea, 1977.

[10] J. von Zur Gathen, M. Giesbrecht *Constructing normal bases in finite fields*, J.Symbolic Computation 1990 **10**, 547-570.

[11] J. von Zur Gathen and Victor Shoup, *Computing Frobenius maps and Factoring Polynomials* Computational Complexity, Vol. 2 1992, pp 187-224.

[12] D. E. Knuth *The Art of Computer Programming*, Vol. 2/ Seminumerical Algorithms, Addison Wesley.

[13] S.Lang *Algebra* Addison Wesley, 1984.

[14] H. W. Lenstra, R. J. Schoof *Primitive normal bases for finite fields*, Mathematics of computation, Vol 48, 1987, pp. 217-232.

[15] R. Lidl and H. Niederreiter *Finite Fields*, vol.**20** of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley (Reading MA), 1983.

[16] P. Ozello *Calcul exact des formes de Jordan et de Frobenius d'une matrice*, Thèse de 3ème cycle, Université Scientifique Technologique et Médicale de Grenoble, 1987.

[17] F. J. Mac Williams and N. J. A. Sloane *The Theory of Correcting Codes*, North-Holland, 1977.

[18] T. Matsumoto, H. Imai *Public quadratic polynomial-tuples for efficient signature-verification and message encryption* Advances in Cryptology-Eurocrypt '88, Lecture Notes in Computer Science, Vol 330, Springer-Verlag 1988.

[19] M. Mignotte, *Mathématiques pour le calcul formel*, Presses Universitaires de France 108, boulevard Saint-Germain, 75006 Paris.

[20] Richard Stong *Some Asymptotic Results on Finite Vector Spaces*, Advances in Applied Mathematics, **9**, pp. 167-199 (1988).

[21] J. H. Wilkinson *The Algebraic Eigenvalue Problem* Clarendon Press Oxford.

Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399