On the Computation of Minimal Polynomials, Cyclic Vectors, and Frobenius Forms

Daniel Augot and Paul Camion*

Projet Codes
INRIA Rocquencourt
Domaine de Voluceau, BP105
78153, Le Chesnay Cedex, France

Submitted by Richard A. Brualdi

ABSTRACT

Various algorithms connected with the computation of the minimal polynomial of an $n \times n$ matrix over a field $K$ are presented here. 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 of a matrix of size $n$. Furthermore, an iterative algorithm for the minimal polynomial is presented with complexity $O(n^3 + mn^2)$, where $m$ is a parameter of the shift Hessenberg matrix used. It does not require knowledge of the characteristic polynomial. Important here is the fact that the average value of $m$ or $mA$ is $O(\log n)$. Next we are concerned with the topic of finding a cyclic vector for a matrix. We first consider the case where its characteristic polynomial is square-free. Using the shift Hessenberg form leads to an algorithm at cost $O(n^3 + mn^2)$. A more sophisticated recurrent procedure gives the result in $O(n^3)$ steps. In particular, a normal basis for an extended finite field of size $q^n$ will be obtained with deterministic complexity $O(n^3 + n^2 \log q)$. Finally, the Frobenius form is obtained with asymptotic average complexity $O(n^3 \log n)$. All algorithms are deterministic. In all four cases, the complexity obtained is better than for the heretofore best known deterministic algorithm. © Elsevier Science Inc., 1997

* E-mail: Daniel.Augot@inria.fr, Paul.Camion@inria.fr.

1. INTRODUCTION

We present various low complexity algorithms for computing the objects in the title, for an $n \times n$ square matrix over a commutative field $K$.

Section 2 is concerned with the problem of obtaining the minimal polynomial of a square matrix $A$. The introduced algorithm requires the factorization of the characteristic polynomial of $A$ as input, and produces the minimal polynomial and all characteristic subspaces at cost $O(\sqrt{n}n^3)$. It appeals to a recurrent divide-and-conquer procedure.

Section 3 is of a theoretical nature. We introduce the shift Hessenberg form of a matrix, whose algebraic properties are studied. Asymptotics are also derived, from the results of R. Stong [16]. Basic algorithmic with that form is detailed.

In Section 4, using this form, we obtain an iterative algorithm ending in the minimal polynomial of $A$ in $O(n^3 + n^2 m^2)$ arithmetic operations. It does not need any knowledge of the characteristic polynomial. The number $m$ is a parameter of the shift-Hessenberg form of $A$, and we have that $m \leq m_A$, where $m_A$ is the number of factors of the characteristic polynomial of $A$, counted with multiplicities.

We next are concerned with the topic of finding a cyclic vector. We will construct an 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 5, and to a more sophisticated recurrent procedure with complexity $O(n^3)$ presented in Section 6. Special attention is given to cyclic vectors for the Frobenius automorphism of a finite field. This ends in a deterministic algorithm for computing a normal basis for $\mathbb{F}_q^n$ in $O(n^3 + n^2 \log q)$ elementary operations.

In Section 7, the Frobenius form is obtained with asymptotic average complexity $O(n^3 \log n)$.

The results are summarized in Table 1. The best known algorithms are obtained by Giesbrecht in [8, 6], whose algorithms have complexities very close to the optimal complexities. Giesbrecht uses randomness, probabilistic methods, and fast algorithms for polynomial arithmetic. When deterministic algorithms are considered, the best previously known bound is $O(n^4)$ (mainly Özello's thesis [13] and Lüneburg [12], but other references are in [8]), which we also get here for most algorithms, except for the cyclic vector problem and the normal basis problem, where we obtain better complexities (see [3], but note that an algorithm of $O(n^3 + n \log n \log \log \log n \log q)$ is given in [15], using, as we did before [2], shift Hessenberg matrices).

But we want to emphasize that we are able to design deterministic algorithms well adapted for "easy matrices," that is, matrices which have only a few factors for the characteristic polynomial. Over finite fields, this happens to be the general case, as we show here in Theorem 6.
### Table 1

**Summary of Complexities**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
<th>Average complexity</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal polynomial(^b)</td>
<td>(O(n^{3.5}))</td>
<td>(O(n^{3.5}))</td>
<td>2</td>
</tr>
<tr>
<td>Minimal polynomial</td>
<td>(O(n^3 + n^2m_A^2))</td>
<td>(O(n^3))</td>
<td>4</td>
</tr>
<tr>
<td>Cyclic vector(^c)</td>
<td>(O(n^3 + n^2m_A^2))</td>
<td>(O(n^3))</td>
<td>5</td>
</tr>
<tr>
<td>Cyclic vector(^c)</td>
<td>(O(n^3))</td>
<td>(O(n^3))</td>
<td>6</td>
</tr>
<tr>
<td>Normal basis(^d)</td>
<td>(O(n^3 + n^2\log q))</td>
<td>(O(n^3 + n^2\log q))</td>
<td>6</td>
</tr>
<tr>
<td>Frobenius form(^e)</td>
<td>(O(n^3m_A))</td>
<td>(O(n^3\log n))</td>
<td>7</td>
</tr>
</tbody>
</table>

\(^a\) n is the size of \(A\), \(m_A\) is the number of factors of its characteristic polynomial. Average complexity is over a finite field. 

\(^b\) With the knowledge of the factorization of the characteristic polynomial. Also produces bases for the characteristic subspaces. 

\(^c\) Assuming that the characteristic polynomial is square free. 

\(^d\) Normal basis of \(\mathbb{F}_{q^n}\) over \(\mathbb{F}_q\). 

\(^e\) With the knowledge of the factorization of the characteristic polynomial. 

---

**NOTE 1.** Our complexity assessments are given in terms of elementary operations over \(K\). All algorithms presented here may be applied to matrices over any field \(K\) and in particular over \(\mathbb{Q}\), but we don’t give any evaluation of the bit complexity.

The bit complexity when matrices over integers are considered in [13, 7], and in [10, 9] with experimental studies. We have not investigated parallel algorithms for these problems. These are discussed most recently in [8], and previously in [17].

---

**2. CHARACTERISTIC SUBSPACES AND MINIMAL POLYNOMIAL IN \(O(n^{3.5})\)**

In this section, an algorithm with complexity \(O(n^3\sqrt{n})\) is presented for computing the minimal polynomial of a matrix \(A\), computing a block-diagonal matrix \(D\) similar to \(A\), and exhibiting its characteristic subspaces. The inputs are \(A\) and the factorization of its characteristic polynomial. The outputs are the minimal polynomial, a block-diagonal matrix \(D\) exhibiting the restriction of \(A\) at its characteristic subspaces, and an invertible matrix \(P\) such that \(D = P^{-1}AP\).

Note that the characteristic polynomial of a matrix \(A\) can be computed in \(O(n^3)\) elementary operations, as described in [20, 4], by computing a Hessenberg form of \(A\). The characteristic polynomial can be factored at low cost, for instance \(O(n^3 + n^3\log q)\) [19], although it is not a deterministic algorithm.
2.1. Characteristic Subspaces

We recall known facts about characteristic subspaces of a matrix $A$. The reader may refer to [5]. Let $K$ be a field. We denote by $M_n(K)$ the algebra of $n \times n$ matrices, and we denote by $\text{GL}_n(K)$ the group of invertible $n \times n$ matrices.

**Theorem 1.** Let $C(X)$ be the characteristic polynomial of matrix $A \in M_n(K)$, and assume $C(X) = P(X)Q(X)$ where $P(X)$ and $Q(X)$ are relatively prime. Let $V_p = \ker P(A)$ and $V_Q = \ker Q(A)$. Then

$$K^n = V_p \oplus V_Q,$$

and $V_p = \text{Im } Q(A)$ and $V_Q = \text{Im } P(A)$.

**Definition 1.** Let $C(X)$ be the characteristic polynomial of a matrix $A$, and let $C(X)$ factor into $f_1(X)^{a_1} \cdots f_k(X)^{a_k}$, where the polynomials $f_i$ are irreducible. The characteristic subspaces of $A$ are the invariant subspaces $V_i = \ker f_i(A)^{a_i}$, $i = 1, \ldots, k$.

2.2. The Algorithm

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

If the characteristic polynomial is not a power of an irreducible polynomial, we are able to split $C(X)$ into $C(X) = P(X)Q(X)$ where $P(X)$ and $Q(X)$ are 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 given by Theorem 1 on both $V_p$ and $V_Q$. The new matrices are split in their turn, until all characteristic subspaces of $A$ are obtained. Finally the minimal polynomial of the restriction of $A$ to each of those subspaces is computed. The product of those polynomials gives the final result.

**Input:** The matrix $A$ and the factorization of its characteristic polynomial, $C(X) = f_1(X)^{a_1} \cdots f_k(X)^{a_k}$, where $f_1(X), \ldots, f_k(X)$ are the irreducible factors of $C(X)$.

**Output:** The minimal polynomial of $A$, the splitting of $K^n$ into all characteristic subspaces of $A$, and the matrix of the restriction $A$ to the characteristic subspaces.

**Step 1:** Find a splitting of $C(X) = P(X)Q(X)$ where $P(X)$ and $Q(X)$ are coprime. Three cases are considered.
(1) $C(X) = p(X)^r$, $p(X)$ irreducible. Compute the minimal polynomial $p(X)^r$ of $A$ in $\lceil \log_2 r \rceil$ steps by trial and error on $s$. This is done with complexity $O(n^{3\sqrt{n}})$, using Theorem 2, which follows.

(2) 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 $Q(A)$ gives a basis for a characteristic subspace.

(3) 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 $\deg P(X) \leq \frac{2}{3}n$ and $\deg Q(X) \leq \frac{2}{3}n$. This is described in Lemma 1, which follows.

Step 2: Compute $Q(A)$ [and $P(A)$]. This gives generating vectors for the subspace for $V_P [V_Q]$. This is done at cost $O(n^{3\sqrt{n}})$ by using Theorem 2.

Step 3: Compute a basis for $V_P [V_Q]$. This is done with Gaussian elimination, at cost $O(n^3)$.

Step 4: Change basis, taking as a new basis for $K^n$ the union of the bases just computed. This produces the matrices $A_P$ and $A_Q$ of the restriction of $A$ to $V_P$ and $V_Q$ respectively. The cost is again $O(n^3)$.

Recursive step: Recursively apply the procedure to $A_P$ and $A_Q$. The terminal steps end in a basis for each characteristic subspace by giving the diagonal blocks of $D$.

Now two main operations are to be performed: the splitting and the evaluation of the polynomials $P(X)$ and $Q(X)$ at $A$. The algorithm for that evaluation with complexity $O(n^{3\sqrt{n}})$ is detailed in next section.

2.3. Splitting the Factors, and Evaluation

We state the following useful lemma.

**Lemma 1.** Let $n$ and $n_i$, $i \in [1, k]$, be positive integers. For a subset $T \subset [1, k]$ we denote by $n_T$ the integer $\sum_{i \in T} n_i$. If we have that $n_1 + \cdots + n_k = n$, and $n_i \leq \frac{2}{3}n$, for $1 \leq i \leq k$, then there exists a partition $[1, k] = I \cup J$ such that

$$n_I \leq \frac{2}{3}n \quad \text{and} \quad n_J \leq \frac{2}{3}n.$$

**Proof.** If there exists $n_i > \frac{1}{3}n$, choose $I = \{i\}$ and $J = [1, k] \setminus I$.

Otherwise, choose $J$ as the subset of $[1, k]$ of maximal size such that $n_J \leq \frac{2}{3}n$. Then $I = [1, k] \setminus J$ necessarily satisfies $n_I \leq \frac{2}{3}n$. Indeed, if $n_I > \frac{2}{3}n$, let $I'$ be constructed from $I$ by removing any of its element. Then $n_I' > \frac{2}{3}n - \frac{1}{3}n$, since $n_i \leq \frac{2}{3}n$, $i \in [1, k]$. Then the complementary $J'$ of $I'$ in $[1, k]$ satisfies $n_J' \leq \frac{2}{3}n$ and contains $J$. This contradicts the maximality of $J$. \qed
Thus, given $k$ integers $n_1, \ldots, n_k$ summing up to $n$, we are able to find a splitting $[1, k] = I \cup J$ such that either $I = \{i\}$, and $i > \frac{2}{3}n$, or both sets $I$ and $J$ are such that $n_i \leq \frac{2}{3}n$ and $n_j \leq \frac{2}{3}n$.

It is known that $p(A)$ 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 result has been published in [14], and we recall it for completeness.

**Theorem 2.** For all $A$ in $M_n(K)$, and for all $p(X)$ with $\deg p(X)$ at most $t$, $p(A)$ can be computed with complexity $O(\sqrt{t}n^3)$ and memory space $O(\sqrt{t}n^2)$.

**2.4. The Complexity**

**Theorem 3.** Given the factorization of the characteristic polynomial of any square matrix over a finite field $K$, then its minimal polynomial and a block-diagonal matrix similar to it, exhibiting its characteristic subspaces, can be computed with time complexity $O(n^3\sqrt{n})$ and memory size $O(n^2\sqrt{n})$.

**Proof.** We can assume that the intermediate computations, in one step, for splitting the factors, reevaluating matrices, and computing bases for subspaces are all bounded by $\gamma n^{3.5}$. We have to show that the whole recursive algorithm has complexity bounded by $O(n^{3.5})$. We prove it by induction, assuming that the cost $C(m)$ of the algorithm is bounded by $\beta m^{3.5}$, for $m < n$. Then

$$C(n) \leq \gamma n^{3.5} + 2C(\frac{2}{3}n) \leq \gamma n^{3.5} + 2\beta(\frac{2}{3})^{3.5}n^{3.5}.$$  

Thus $C(n) \leq \beta n^{3.5}$, with $\beta = \gamma/[1 - 2(\frac{2}{3})^{3.5}]$. The memory size $O(n^2\sqrt{n})$ is established similarly.

**3. THE SHIFT HESSENEBERG FORM AND THE CENTRALIZER OF A MATRIX**

We now use the shift Hessenberg form of a matrix. 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 $GL_n(K)$ commuting with a given fixed linear operator on $K^n$. 
3.1. **Shift Basis**

Let $A$ be a square matrix, we denote $\pi(A)$ the minimal polynomial of $A$.

**Definition 2.** For $A$ in $M_n(K)$ and $v$ in $K^n$, the *minimal polynomial of $A$ restricted to $v$* is the lowest degree monic nonzero polynomial $\pi_v(X)$ such that $\pi_v(A)v = 0$.

**Definition 3.** Let $A$ be an operator on $K^n$. A *shift basis* for $A$ is a basis which has the form

$$[v_1, Av_1, \ldots, A^{n_1-1}v_1, v_2, Av_2, \ldots, A^{n_2-1}v_2, \ldots, v_m, Av_m, \ldots, A^{n_m-1}v_m]$$

(1)

such that $A^{n_k}v_k$ is a linear combination of vectors with the form $A^jv_j$, $j < k$, and $A^iv_k$, $i < l$, but for $l < n_k$, $A^iv_k$ is linearly independent of the $A^jv_k$, $i < l$, and of the $A^jv_j$, $j < k$.

It is understood that a shift basis is actually an ordered basis. Given $A$, a shift basis for $A$ can be obtained as follows. First select any $v_1$, and introduce the linear independent vectors $A^iv_1$, for $i = 0, \ldots, n_1 - 1$, where $n_1$ is the smallest value of $i$ such that $A^iv_1$ linearly depends on all previous vectors. Then select $v_2$ independent of all previous vectors, and proceed with $A^iv_2$, $i = 0, \ldots, n_2 - 1$ as for $v_1$. The process ends in a shift basis with $n_1 + n_2 + \cdots + n_m = n$.

**Definition 4.** We call a matrix which represents an operator $A$ in a shift basis a *shift Hessenberg* matrix $A$ shift Hessenberg matrix has the following form:

$$H = \begin{bmatrix}
1 & \times & \times & \times \\
1 & \times & \times & \times \\
0 & 1 & \times & \times \\
& & & & \\
& & 1 & \times & \times \\
& & 0 & 1 & \times \\
& & & & & & & \ddots & \ddots & \ddots & \ddots \\
& & & & & & & & 1 & \times & \times \\
& & & & & & & & 0 & 1 & \times \\
& & & & & & & & & & & \ddots & \ddots & \ddots \\
& & & & & & & & & & & & 1 & \times & \times \\
& & & & & & & & & & & & 0 & 1 & \times \\
& & & & & & & & & & & & & & & \ddots & \ddots \\
& & & & & & & & & & & & & & & & 1 & \times & \times \\
& & & & & & & & & & & & & & & & 0 & 1 & \times \\
& & & & & & & & & & & & & & & & \ddots & \ddots \end{bmatrix}$$
The parameter \( m \) of a shift Hessenberg matrix is defined to be the number of zeros on the first subdiagonal, plus one.

**Remark 1.** We observe that if \( v_1 \) is the first element of a basis in which a matrix \( M \) represents an operator \( A \), then there exists a shift Hessenberg matrix \( H \) that represents \( A \) and that can be written as \( Q^{-1}MQ \) where \( v_1 \) indexes the first column of \( Q \).

**Note 2.** The number \( m \) is the number of diagonal blocks, each block being a companion matrix, i.e. a matrix of the form

\[
\begin{bmatrix}
1 & c_0 \\
. & . \\
. & . \\
. & . \\
1 & c_{n-1}
\end{bmatrix}
\]

The characteristic polynomial of such a matrix equals its minimal polynomial and is equal to \( X^n - c_{n-1}X^{n-1} - c_{n-2}X^{n-2} - \cdots - c_1X - c_0 \). In the case where the parameter \( m = 1 \), the shift Hessenberg matrix is itself a companion matrix. We shall denote by \( C_p \) the companion matrix of the polynomial \( p \). The other extreme situation is for \( m = n \), where we have an upper triangular matrix.

To every shift basis corresponds an increasing sequence \( V_1 \subset \cdots \subset V_m \) of invariant subspaces of \( A \), defined by \( V_1 = \text{Vect}(Av_1; \ i \geq 0) \), \( V_2 = \text{Vect}(Av_1, Av_2; \ i, j \geq 0), \ldots, V_m = K^n \).

We have that \( V_i \) is a \( K[A] \)-module, \( i = 1, \ldots, m \), and consequently \( V_i/V_{i-1} \) is a module. Each of those quotient modules is generated by a single element \( \bar{v}_i \) which is the class of \( v_i \) in \( V_i/V_{i-1} \). If the minimal polynomial of \( \bar{v}_i \) is denoted by \( f_i(X) \), the \( i \)th diagonal block in the shift Hessenberg matrix is the companion matrix of \( f_i(X) \). We also note that \( f_i(X) \) divides the minimal polynomial of \( v_i \).

### 3.2. The \( K[X] \)-Module Induced by a Matrix

**Notation 1.** Let \( p \) be a polynomial and \( v \) a vector. We use the module notation \( pv \) for \( p(A)v \). We use the same notation \( pv \) when \( p \) is a polynomial \( p(A_V) \) evaluated at the restriction \( A_V \) of \( A \) to an invariant subspace \( V \), and when \( v \) is a vector in \( V \).
DEFINITION 5. The expanded Frobenius form of $A$ in $M_n(K)$ is the following matrix $D$ similar to $A$:

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

where each matrix $F_{B_i,B_i}$ is a Frobenius matrix

$$\begin{bmatrix}
C_{p_{i,1}} & 0 & \cdots & 0 \\
0 & C_{p_{i,2}} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & C_{p_{i,m_i}}
\end{bmatrix},$$

with $s_{i,1} < s_{i,2} < \cdots < s_{i,m_i}$, and with $\gcd(p_i, p_j) = 1$ if $i \neq j$.

We thus have that $p_{i,m_i}$ is the minimal polynomial of $F_{B_i,B_i}$. The subspaces for which the matrix is a companion matrix are denoted by $V_{p_{i,1}}, \ldots, V_{p_{i,m_i}}$. The whole space $K^n$ can be viewed as the direct sum $\bigoplus_{i=1}^{d} \bigoplus_{j=1}^{m_i} V_{p_{i,j}}$.

We consider $K^n$ equipped with the natural structure of a $K[X]$-module induced by $A$. It is readily seen that the module $K^n$ can be represented as the product of rings:

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

considered as $K[X]$-modules where $R_{i,j} = K[X]/p_{i,j}^{\deg p_i}$. For any vector $u$, we denote by $u|_{R_{i,j}}$ the component of $u$ in the ring $R_{i,j}$. Thus $u|_{R_{i,j}}$ may be considered indiscriminately as a vector or a polynomial of degree less than $s_{i,j} \deg p_i$.

3.3. Shift Bases for the Expanded Frobenius Form

We observe that an expanded Frobenius form is an instance of a shift Hessenberg form, and we will study the nature of shift bases which yield the expanded Frobenius form of an operator $A$. For brevity we call such a shift-basis a Frobenius shift basis. Thus for a fixed expanded Frobenius form $D$ of $A$, all matrices $B$ such that $B^{-1}AB = D$ are to be considered.
Lemmas 2. Let $u$ be a vector in $K^n$, such that $p_i^{s_{i,j}} u = 0$. Then the components of $u$ viewed in the $K[X]$-module decomposition of $K^n$ satisfy $u_{R_{k,l}} = 0$ if $k \neq i$.

Proof. Let us assume that there exist $k, l$, $k \neq i$, 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 since $p_i^{s_{i,j}}$ is a unit of $R_{k,l} = K[X]/p_i^{s_{i,j}}$. This contradicts the assumption on $u$.

Lemma 3. Let $u$ be a vector in $K^n$ whose minimal polynomial is $p_i^{s_{i,j}}$. Then the components of $u$ in $R_{i,l}$ are described as follows:

1. $l < j$: $u_{R_{i,l}}$ can be any element of $R_{i,l}$.
2. $l = j$: $u_{R_{i,l}}$, considered as a polynomial, is prime to $p_i$.
3. $l > j$: $u_{R_{i,l}}$ is a multiple of $p_i^{s_{i,j} - s_{i,l}}$.

Proof. Since the minimal polynomial of $u$ is $p_i^{s_{i,j}}$, then we have that $p_i^{s_{i,j}+1} v = 0$ for any vector $v$ in $R_{i,l}$ whenever $l < j$, since $p_i^{s_{i,j}}$, which divides $p_i^{s_{i,j}+1}$, is the minimal polynomial of $A$ restricted to $R_{i,l}$. This establishes the result for the case $l < j$.

In the case $l = j$, 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 the case $l > j$ we must have that $p_i^{s_{i,j}+1} u = 0$. This implies in $R_{i,l}$ that $p_i^{s_{i,j}+1} u_{R_{i,l}} = 0$ mod $p_i^{s_{i,j}}$, and thus we must have that $p_i^{s_{i,j} - s_{i,l}}$ divides $u_{R_{i,l}}$.

Property 1. From the previous lemmas, the following property holds. All shift bases for $A$ which yield the expanded Frobenius matrix $D$ as described in Definition 5 have the form

$v_{1,1}, Dv_{1,1}, \ldots, D^{n_{i,l}-1}v_{1,1}, \ldots, v_{d,m_d}, Dv_{d,m_d}, \ldots, D^{n_{i,d}-1}v_{d,m_d}$,

where $n_{i,j} = s_{i,j} \deg p_i$, and where each $u_{i,j}$ is such that $p_i^{s_{i,j}}$ is its minimal polynomial.

3.4. From Shift Bases to the Centralizer of a Matrix

Let $A \in M_n(K)$ be similar to the matrix $S$, with $S = P^{-1}AP$. Suppose that $P'$ is another basis yielding the same form $S$ of $A$, that is, $S = P'^{-1}AP'$. From the equality $P^{-1}AP = P'^{-1}AP'$, we get $(P'P^{-1})A = A(P'P^{-1})$. Thus $P'P^{-1}$ belong to the centralizer of $A$. Conversely, let $C$ belong to the centralizer of $A$, and let $S \sim P^{-1}AP$. Since $AC = CA$, we get $S = (CP)^{-1}A(CP)$, thus $CP$ is another basis yielding the matrix $S$ for $A$.
In particular, if $S$ is an expanded Frobenius form of $A$, then the shift bases yielding $S$ correspond to the elements of the centralizer of $A$. Let us consider such a matrix $S$ with the form

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

where $s$ is the restriction of $A$ to $V_s$, and $t$ the restriction of $A$ to $V_t$, with $V_s \oplus V_t = K^n$. In addition let the minimal polynomials of $s$ and $t$ be relatively prime.

Let $v_{1,1}, \ldots, v_{k_r,m_k}$ be vectors producing a Frobenius shift basis for $s$, as in Property 1, and let $p_{i,j}^s$ be the minimal polynomial of $v_{i,j}$. Let $v_{i,j}^*$ denote the vector $(v_{i,j}, 0) \in K^n$, then the minimal polynomial of $v_{i,j}^*$ is $p_{i,j}^s$. Similarly and independently, let $w_{1,1}, \ldots, w_{k_r,m_k}$ yield a Frobenius shift basis for $t$, with minimal polynomials $q_{i,j}^t$, and let $w_{i,j}^*$ be the vector $(0, w_{i,j})$, whose minimal polynomial is also $q_{i,j}^t$. Then, by Property 1, the sequence $v_{1,1}, \ldots, v_{k_r,m_k}, w_{1,1}, \ldots, w_{k_r,m_k}$ yields a Frobenius shift basis for $A$, and every expanded Frobenius form for $A$ is obtained that way. We have proved:

**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.

Indeed, by the argument above, the size of the centralizer is the number of distinct shift bases for $A$ yielding a fixed expanded Frobenius form for $A$. In the case where $K = \mathbb{F}_q$, we can derive the enumeration of the centralizer of any given matrix $A$.

**Theorem 4.** Let $A$ be an operator whose expanded Frobenius form is as in Definition 5. The number of shift bases for $A$ which yield the above Frobenius form is

$$\prod_{i=1}^d \prod_{j=1}^{m_i} q^{(\deg p_i)(\sum_{l=1}^{i-1} z_{i,l} + (m_i - j) z_{i,j})} \phi(p_{i,j}^t),$$

(2)

where $\phi(p_{i,j}^t) = q^{z_{i,j} \deg p_i (1 - q^{-\deg p_i})}$ is the number of polynomials of degree less than $\deg p_{i,j}^t$ prime to $p_{i,j}^t$. 


Proof. Each such shift basis is given by a sequence such as
\[ v_{1,1}, v_{1,2}, \ldots, v_{1,m}, v_{2,1}, \ldots, v_{2,m}, \ldots, v_{d,1}, \ldots, v_{d,m} \]
in which the minimum degree polynomial canceling \( v_{i,j} \) is \( p_{l}^{i,j} \).

In (2), the outermost product is due to Lemma 2. The innermost product enumerates for each \( p_{l}^{i,j} \) the vectors \( v \) such that \( p_{l}^{i,j}v = 0 \). The sum \( \Sigma_{j=1}^{n} s_{i,j} \) stands for the rings \( R_{i,j}, l < j \), in which any vector \( v \) satisfies \( p_{l}^{i,j}v = 0 \). 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 \( p_{l}^{i,j-s_{i,j}} \) in \( K[X]/p_{l}^{i,j} \) is \( q^{(deg p_{l})s_{i,j}} \).

Finally, \( \phi(p_{l}^{i,j}) \) is the number of polynomials prime to \( p_{l}^{i,j} \), i.e. the number of units in \( R_{i,j} \).

3.5. The Average Number of Factors of a Characteristic Polynomial

R. Stong gives in [16] the following result.

THEOREM 5. Let \( X_n \) be the random variable assuming as values the numbers of factors of the characteristic polynomials of matrices in \( GL_n(F_q) \), counted with multiplicities. Then the expectation \( EX_n \) of \( X_n \) is asymptotically equivalent to \( \log n \).

We shall generalize the result to all matrices by proving the following:

THEOREM 6. Let \( Y_n \) be the random variable assuming as values the number of factors of characteristic polynomials of matrices in \( M_n(F_q) \), counted with multiplicities, and let \( EY_n \) be the expectation of \( Y_n \). Then, for every \( \epsilon > 0 \), there exists \( n_0 \) 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 established first. For any matrix \( A \in M_n(F_q) \), we consider its expanded Frobenius form as follows:
\[ \begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \]  \hspace{1cm} (3)

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

LEMMA 4. The average number \( EZ_n \) of factors, counted with multiplicities, of the characteristic polynomial of \( t \), as in (3), for matrices \( A \) in \( M_n(F_q) \), satisfies \( \forall \epsilon > 0, \exists n_0 : n \geq n_0 \Rightarrow EZ_n \leq (1 + \epsilon) \log n \).
Proof. Let $S_n$ 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 centralizer of $s \in S_n$, and by $z_{t, n_2}$ the size of the centralizer of $t \in S_{n_2}$. Given $s$ in $S_n$ and $t$ in $S_{n_2}$, then by Corollary 1, the number of matrices having as Frobenius form the matrix (3) is

$$\frac{|\text{GL}_n(F_q)|}{z_{s, n_1} z_{t, n_2}}.$$ 

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 as in (3) is

$$\sum_{s \in S_n} \frac{|\text{GL}_n(F_q)|}{z_{s, n_1}} = \frac{|\text{GL}_n(F_q)|}{z_{t, n_2}} \sum_{s \in S_n} \frac{1}{z_{s, n_1}} = \frac{1}{z_{t, n_2}} \chi(n_1, n, q),$$

where $\chi(n_1, n, q) = |\text{GL}_n(F_q)| \sum_{s \in S_n} (1/z_{s, n_1})$.

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 let $S_{n_2, k}$ be the set of Frobenius matrices of size $n_2$ each of whose characteristic polynomial belongs to $C_{n_2, k}$. The number of matrices in $M_n(F_q)$ whose characteristic polynomial is $X^{n_1}C(X)$, for some $C(X)$ 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 $\theta$ the random variable assuming as value the size of the nonsingular part of a matrix, and denote by $\eta$ the random variable assuming as value the number of factors of the characteristic polynomial of the nonsingular part. The conditional probability $P_{\eta|\theta}(\eta = k | \theta = n_2)$ that $C(X)$ belongs to $C_{n_2, k}$ for a matrix in $M_n(F_q)$ whose characteristic polynomial is $X^{n_1}C(X)$ is thus

$$\chi(n_1, n, q) \sum_{t \in S_{n_2, k}} \frac{1}{z_{t, n_2}} = \frac{\sum_{t \in S_{n_2, k}} |\text{GL}_{n_2}(F_q)|}{\sum_{t \in S_{n_2}} |\text{GL}_{n_2}(F_q)|} = P_{\eta}(\eta = k),$$

where $P_{\eta}(\eta - k)$ denotes the probability that an invertible matrix $\text{GL}(n, F_q)$ 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 $M_n(F_q)$ is given by

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

$$= \sum_{n_k=1}^{n} P(\theta = n_k) EX_{n_k}.$$ 

Let $\varepsilon$ be given. Since $EX_n \sim \log n$, there exists $n_1$ such that for $n \geq n_1$ then $EX_n/\log n < 1 + \varepsilon/2$. Thus

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

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

$$\leq \frac{\sum_{n_2=n_1}^{n} EX_{n_2}}{\log n} + \sum_{n_2=n_1+1}^{n} P(\theta = n_2) \left(1 + \frac{\varepsilon}{2}\right)$$

$$\leq \frac{\sum_{n_2=n_1}^{n} EX_{n_2}}{\log n} + 1 + \frac{\varepsilon}{2}.$$ 

We thus can choose $n_0$ such that, for all $n \geq n_0$, $(EZ_n)/\log n < 1 + \varepsilon$. 

The proof of Theorem 6 will be completed by the following lemma.

**Lemma 5.** Let $Z_n$ be the random variable assuming as values the number of factors $X$ of characteristic polynomials of matrices in $M_n(F_q)$. Then the expectation $EZ_n$ is asymptotically bounded by $\log n$.

**Proof.** Let us consider the translation on $M_n(F_q)$ defined by $M \mapsto M + I_n$. 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 $M'$:

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix}.$$
with \( s \) nilpotent and \( t \) 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 4 the expected number of factors of \( C(X) \) is asymptotically \( \log n \); thus \( n_1 \) is asymptotically bounded by \( \log n \).

Theorem 6 now follows from Lemma 4 and Lemma 5.

3.6. Computing with the Shift Hessenberg Form

We recall the following theorem, whose proof and corresponding algorithm may be found in Ozello's thesis [13]. The algorithm is an adaptation of the algorithm for computing the Hessenberg form [20, 4].

**Theorem 7.** For all \( A \) in \( M_n(K) \), there exists a shift Hessenberg matrix \( H \) and an invertible matrix \( P \) such that \( H = PAP^{-1} \). The matrices \( H \) and \( P \) can be obtained in \( O(n^3) \) elementary operations.

We now investigate more precisely the number \( m \) of diagonal blocks of the shift Hessenberg form for a matrix \( A \).

**Definition 6.** Let \( A \) be a square matrix in \( M_n(K) \). We denote by \( m_A \) the maximum size of an increasing sequence \( V_1 \subset \cdots \subset V_{m_A} \) of invariant subspaces of \( K^n \) under \( A \).

Since any shift Hessenberg form of a matrix \( A \) yields \( m \) invariant subspaces \( V_1 \subset \cdots \subset V_m \), the parameter \( m \) is bounded from above by \( m_A \). The number \( m_A \) equals the number of irreducible factors of the characteristic polynomial of \( A \), counted with multiplicities. Thus

**Corollary 2.** The expected value of \( m_A \) is \( O(\log n) \).

For clarity, the complexity of some algorithms will be given in terms of \( n \) and \( m_A \). This will lead to average case complexities in terms of \( n \) and \( \log n \). Although the algorithms here presented all are deterministic, the complexity is a random variable (for an average distribution of matrices) whose expectation is bounded from above.

We show some results about the complexity of some computations with a shift Hessenberg matrix. We also recall how some problems concerning companion matrices can be solved fast.

First observe that a shift Hessenberg matrix is a sparse matrix, with at most \( m + 1 \) nonzero entries in each row. This leads to the observation stated as follows.
Lemma 6. Let $H$ be a shift Hessenberg matrix of size $n$, and let $M$ be any matrix of size $n \times n'$. Then product $HM$ can be computed at cost $O(mnn')$.

Furthermore a shift Hessenberg matrix has some properties regarding cyclicity, as already seen in Definition 4, which can be exploited for reducing costs.

Definition 7. Let $H$ from $M_n(K)$ be a shift Hessenberg matrix. A matrix $A$ is polycyclic for $H$ if there exist column vectors $v_1, v_2, \ldots, v_m$ in $K^n$ such that $A$ has the form

$$[v_1, Hv_1, \ldots, H^{n_1-1}v_1, v_2, Hv_2, \ldots, H^{n_2-1}v_2, \ldots, v_m, Hv_m, \ldots, H^{n_m-1}v_m],$$

where $n_1, n_2, \ldots, n_m$ are the sizes of the diagonal blocks of $H$.

Proposition 1. Let $H$ be a shift Hessenberg matrix, and let $A, B$ be two matrices which are polycyclic for $H$. Let $\alpha, \beta$ be any field elements, then $\alpha A + \beta B$, $I_n$, $H$, $HA$, and $HB$ all are polycyclic for $H$.

In other words, polycyclic matrices for $H$ form a $K[H]$-submodule of $M_n(K)$. We observe that $H$ itself is polycyclic for $H$ and so is every matrix in $K[H]$.

Proposition 2. Let $H$ be a shift Hessenberg matrix of parameter $m$. Then the product $HA$ can be obtained with complexity $O(mn^2)$ for any matrix $A$ in $M_n(K)$ and with complexity $O(m^2n)$ whenever $A$ is polycyclic for $H$. A polynomial $p(X)$ of degree at most $t$ can be evaluated at $H$ with complexity $O(tm^2n)$.

Proof. Let $A$ be a polycyclic matrix for $A$. The product $HA$ is performed by modifying $A$ as follows. Delete $v_1$; shift all vectors to the left. Then replace $v_2$ by $HH^{n_1-1}v_1$, and $v_m$ by $HH^{n_m-1}v_{m-1}$. Finally, put $HH^{n_m-1}v_m$ as $n$th column. The whole cost is $m(mn)$.

Let $p$ be a polynomial of degree less than or equal to $t$. We apply Horner's rule for evaluating a polynomial $p(H) = p_1 H^t + p_{t-1} H^{t-1} + \cdots + p_1 H + p_0 I$. We compute $h_1 = p_1 H + p_{t-1} I$, $h_2 = H h_1 + p_{t-2} I$, \ldots, $h_t = H h_{t-1} + t_0 I$. Each matrix $h_t$ is computed from $h_{t-1}$ at a cost $O(m^2n)$, thus a total cost of $O(tm^2n)$ for $p(H)$. ■
We recall very simple and efficient procedures for solving relations involving a companion matrix. This can be found in [6]. From now on, given a companion matrix \( C \) with minimal polynomial \( \pi(X) \) of degree \( n \), the vector \( (v_0, \ldots, v_{n-1}) \) is identified with the polynomial \( u(X) = v_0 + v_1 X + v_2 X^2 + \cdots + v_{n-1} X^{n-1} \).

The following lemma will be used in next section.

**Lemma 7.** Let \( C \) be a companion matrix with minimal polynomial \( \pi(X) \), let \( v \) in \( K^n \), let \( P(X) \) be a polynomial of degree at most \( n \). Then:

1. \( Cv \) is computed at cost \( 2n \).
2. \( P(C) \) is computed at cost \( O(n^2) \).
3. If \( P(X) \) is prime to \( \pi(X) \), then the equation at \( u \), \( P(C)u = v \), is solved at cost \( O(n^2) \).
4. The minimal polynomial \( \pi_v(X) \) of \( C \) restricted to \( v \) is computed at cost \( O(n^2) \).

Using Proposition 2, and computing a shift Hessenberg form at each step of recursion, the algorithm described in Section 2 can be modified to get the following result.

**Theorem 8.** Given the factorized characteristic of a matrix \( A \), the minimal polynomial of matrix \( A \) and a block-diagonal matrix similar to \( A \) exhibiting its characteristic subspaces can be computed with time complexity \( O(n^3 + m_A^2 n^2) \).

The term in \( m_A^2 n^2 \) is due to evaluating polynomials at a shift Hessenberg matrix.

4. **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 straightforward, and it does not require any previous knowledge of the characteristic polynomial. The drawback is that it does not produce a diagonal block decomposition of \( K^n \) into the characteristic subspaces of \( A \).
Let there be given a shift Hessenberg form $H$ for a matrix $A$. Then $H$ has the following block structure:

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

**Notation 2.** We denote by $B_k$ the set of indices of block $k$, and by $K^{B_k}$ the corresponding vector space. We denote by $B_{>k}$ the set of indices $B_k \cup B_{k+1} \cup \cdots \cup B_m$. For any matrix $A \in M_n(K)$ we denote by $A_{B_i, B_j}$ the matrix obtained from rows in $B_i$ and columns in $B_j$. We denote by $A_{B_{>k}}$ the square matrix obtained from rows and columns from the $k$th block up to the end. For a shift Hessenberg matrix, the minimal polynomial of the companion matrix $H_{B_i, B_j}$ is denoted by $f_i(X)$, $i = 1, \ldots, m$. Finally we observe that for any polynomial $g(x)$, then $g(H)$ has the same block structure as $H$.

### 4.1. Nested Ideals Related to $H$

Let $I_k$ denote the set of polynomials $g(X) \in K[X]$ such that

$$
g(H)_{B_i, B_i} = 0, \quad i = 1, \ldots, m, \quad \text{and} \quad g(H)_{B_i, B_j} = 0, \quad k \leq i < j \leq m.
$$

It is in fact an ideal, and the inclusions $I_1 \subseteq I_2 \subseteq \cdots \subseteq I_m$ hold. Let $p_k(X)$ be themonic generator of $I_k$; then $p_k \mid p_{k-1}$. We denote by $\phi_k(X)$ the polynomial such that $p_k(X) = \phi_k(X)p_{k-1}(X)$. Consider the case where $k = 1$. For $g(X) \in I_1$, we have that $g(H)_{B_i, B_j} = 0$, $1 \leq i \leq j \leq m$, and thus $g(H) = 0$. The ideal $I_1 = (p_1(X))$ is the ideal annihilating the matrix $H$, and $p_1$ is the minimal polynomial of $H$.

Now 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 $H$, and we obtain the relation

$$
(Hg(H))_{B_k, B_j} = H_{B_k, B_j}(g(H))_{B_k, B_j}.
$$

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

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

**Lemma 8.** The polynomial \( \phi_k(X) \) is the minimal polynomial of the restriction of \( H_{B_k, B_j} \) to the subspace generated by the columns of the matrices \( p_{k+1}(H)_{B_k, B_j}, j = k, \ldots, m. \)

Notice that, since \( f_k(X) \) is the minimal polynomial of \( H_{B_k, B_j} \), then \( \phi_k(X) \) divides \( f_k(X) \).

### 4.2. The Algorithm for the Minimal Polynomial of \( H \)

The algorithm consists in constructing \( p_m(X), p_{m-1}(X), \ldots, p_1(X) \), step by step, by actually computing the polynomials \( \phi_k(X) \), using Lemma 8.

**First step:** Polynomial \( p_m(X) \) is to be computed. Since all diagonal blocks of \( p_m(H) \) vanish, then \( p_m(X) \) is the least common multiple of the \( f_i(X), i = 1, \ldots, m. \)

**Iterative step:** Computing \( \phi_k(X) \) from the data of \( p_{k+1}(X) \). We have

\[
\begin{bmatrix}
0 & p_{k+1}(H)_{B_k, B_1} & \cdots & \cdots & p_{k+1}(H)_{B_1, B_m} \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
0 & p_{k+1}(H)_{B_k, B_{k+1}} & \cdots & \cdots & 0 \\
0 & 0 & \cdots & \cdots & 0 \\
0 & 0 & \cdots & \cdots & 0
\end{bmatrix}
\]

From Lemma 8, \( \phi_k(X) \) is the minimal polynomial of \( H_{B_k, B_j} \) restricted to the subspace generated by the columns of matrices \( p_{k+1}(H)_{B_k, B_j}, k \leq j \leq m. \) Let \( a^1 = (a^1_1, a^1_2, \ldots, a^1_m) \) be the first nonzero column of the array formed by all matrices \( p_{k+1}(H)_{B_k, B_j}, j \geq k. \) Using Lemma 7, we compute the minimal polynomial \( \phi_{k, a^1}(X) \) of \( H_{B_k, B_k} \) restricted to \( a^1. \) Thus \( \phi_{k, a^1}(X) \) is a factor of \( \phi_k(X) \) and we then compute \( H_{k, a^1} = \phi_{k, a^1}(H)p_{k+1}(H). \) The process is repeated on the first nonzero column \( a^2 \) of the array \( ((H_{k, a^1})_{B_k, B_j}), j \geq k, \) to get a new factor \( \phi_{k, a^2}(X) \) of \( \phi_k(X). \) We compute again \( H_{k, a^2} = \phi_{k, a^2}(H)H_{k, a^1}, \) and proceed with the first nonzero column of the array \( ((H_{k, a^2})_{B_k, B_j}), j \geq k. \) 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 nonzero column which was met.
4.3. Complexity Bounds

The most expensive computations lie in the computation of the matrices $p_m(H), p_{m-1}(H), \ldots$ Let $d_1, \ldots, d_m$ be the degrees of the polynomials $\phi_1, \phi_2, \ldots, \phi_{m-1}, \phi_m$. First $p_m(H)$ is computed at cost $d_m m^2 n$ by Proposition 2. We show how to compute $p_k(H)$ on the data of $p_{k+1}(H)$, which is a polycyclic matrix for $H$. Let $\phi_k(H) = X^{d_k} + a_{d_k-1} X^{d_k-1} + \cdots + a_1 X + a_0$; compute $\phi_k(H)p_{k+1}(H)$ as follows:

$$p_k(H) = \phi_k(H)p_{k+1}(H)$$

$$= (H^{d_k} + a_{d_k-1} H^{d_k-1} + \cdots + a_1 H + a_0)p_{k+1}(H)$$

$$= (H^{d_k-1} + a_{d_k-1} H^{d_k-2} + \cdots + a_1)Hp_{k+1}(H) + a_0 p_{k+1}(H).$$

Now the product $Hp_{k+1}(H)$ is computed at cost $O(m^2 n)$ by Proposition 2, and the product $a_0 p_{k+1}(H)$ at cost $O(n^2)$, and the sum of these two matrices is computed at cost $O(n^2)$. Thus computing $p_k(H)$ is performed at cost $O(d_k(m^2 n + n^2))$, and the final cost is $O((d_1 + \cdots + d_m)(m^2 n + n^2)) - O(m^2 n^2 + n^3)$.

Each computation of a minimal polynomial $\phi_{k,d}$ is done at cost $O(n^2)$. The number of such computations is also bounded by $m_A$. This results in $O(m_A n^3)$ elementary operations for all those gcd computations.

Theorem 9. The minimal polynomial of any matrix $A$ can be obtained in $O(n^3 + m_A n^2)$ elementary operations without any previous knowledge of the characteristic polynomial.

5. SEARCHING FOR A CYCLIC VECTOR

Let us recall some definitions.

Theorem 10 [5, Chapter VII, §3, Theorem 2]. For all $A$ in $M_n(K)$, there exists a vector $v$ in $K^n$ such that $\pi_v(X) = \pi(X)$, where $\pi(X)$ is the minimal polynomial of $A$.

Definition 8. Let $A$ be a matrix in $M_n(K)$. A vector $v$ in $K^n$ such that $\pi_v(X) = \pi(X)$, where $\pi(X)$ is the minimal polynomial of $A$, is called a cyclic vector for $A$. 
First we show how to compute a cyclic vector at cost $O(n^3 + m_A^2 n^2)$ for a square matrix $A$ whose characteristic polynomial is square-free. This implies that the minimal polynomial of $A$ equals its characteristic polynomial. Also the minimal polynomials $f_k(X)$ of the diagonal companion matrices of a shift Hessenberg form for $A$ are pairwise relatively prime.

5.1. Technical Lemmas

The following lemma sets up the recurrence which ends in the sought for cyclic vector.

**Lemma 9.** Let $A$ be a block matrix with the form

\[
\begin{bmatrix}
  A_{B_1, B_1} & A_{B_1, B_2} \\
  0 & A_{B_2, B_2}
\end{bmatrix}
\]

and let $v_{B_1}, v_{B_2}$ be cyclic vectors for $A_{B_1, B_1}$ and $A_{B_2, B_2}$ respectively, which are 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 equations

\[
v_{B_2} = u_{B_2},
\]

(4)

\[
v_{B_1} = f_2(A_{B_1, B_1}) u_{B_1} + (f_2(A) u_{B_2}^*)_{B_1}
\]

(5)

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

**Proof.** The solution $u$ is obtained by finding $u_{B_1}$. Since $f_1(X)$ and $f_2(X)$ are coprime, there exists $h_2(X)$ such that $f_2(X) h_2(X) = 1 \mod f_1(X)$. Thus the matrix $h_2(A_{B_2, B_2})$ is the inverse of $f_2(A_{B_1, B_1})$, and the existence and uniqueness of $u_{B_1}$ is guaranteed. Now we have to prove that $f_1(X) f_2(X)$ is the minimal polynomial of $u = (u_{B_1}, u_{B_2})$. Assume that $p(A)u = 0$ for a
nonzero 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 \( \gcd(p_1(X), p_2(X)) = 1 \). The relation \( p(A)u = 0 \) gives

\[
p_1(A_{B_2, u_2})p_2(A_{B_2, u_3})u_{B_2} = 0. \quad (6)
\]

Since \( \gcd(p_1(X), f_2(X)) = 1 \), there exists \( h_1(X) \) such that \( p_1(X)h_1(X) = 1 \mod f_2(X) \). Applying \( h_1(A_{B_2, B_2}) \) on both sides of (6), we get \( p_2(A_{B_2, B_2})u_{B_2} = 0 \). Since \( u_{B_2} \) is a cyclic vector for \( A_{B_2, B_2} \), \( f_2(X) \) divides \( p_2(X) \) and \( p_2(X) = f_2(X) \).

On the first block of coordinates, the equation \( p(A)u = 0 \) becomes

\[
p_1(A_{B_1, B_1})[f_2(A_{B_1, B_1})u_{B_1} + (f_2(A)u_{B_2}^*)_{B_1}] = 0. \quad (7)
\]

By hypothesis, \( f_2(A_{B_1, B_1})u_{B_1} + (f_2(A)u_{B_2}^*)_{B_1} = u_{B_1} \) is cyclic for \( A_{B_1, B_1} \). Then by (7), \( f_1(X) \mid p_1(X) \) and \( p_1(X) = f_1(X) \). \( \square \)

We observe the striking fact that those computations can be performed at low cost.

**Lemma 10.** A solution \( u \) to the equations (4) and (5) may be computed in \( O(n^3) \) elementary operations.

**Proof.** First compute \( w_{B_1} = (f_2(A)u_{B_2}^*)_{B_1} \), at cost \( O(n^3) \). Then solve (5) by finding an inverse \( h_2(X) \) of \( f_2(X) \mod f_1(X) \). Then the solution \( u_{B_1} \) is given by \( u_{B_1} = h_2(A_{B_1, B_1})(v_{B_1} - w_{B_1}) \), calculated with complexity \( O(n^3) \). \( \square \)

5.2. The Naive Recurrence

Using Notation 2, we denote by \( u_{B \geq k} \) a cyclic vector for \( H_{B \geq k} \). The aim is to find \( u_{B \geq 1} \).

**First step:** The last block \( H_{B \geq k} \) is a companion matrix; the vector \((1, 0, \ldots, 0)\) is a cyclic vector for \( H_{B \geq k} \) and is chosen for \( u_{B \geq k} \).

**Iterative step:** Suppose that the problem has been solved for \( H_{B \geq k+1} \), i.e., we have a vector \( u_{B \geq k+1} \) which is cyclic for \( H_{B \geq k+1} \). The minimal polynomial of \( H_{B \geq k+1} \) is \( f_{k+1}f_{k+2} \cdots f_m \), and the minimal polynomial of \( H_{B \geq k} \) is \( f_k(X) \). These polynomials are coprime, and Lemma 9 can be used to construct \( u_{B \geq k} = (u_{B \geq k}, u_{B \geq k+1}) \) which is cyclic for \( H_{B \geq k} \).

**End:** The result is \( u_{B \geq 1} \).
We now evaluate the number of operations to be performed to achieve the recurrence. The most expensive calculations lie in computing the vectors

\[ w_{B_{m-1}} = (f_m(H)u_{B_m}^*)_{B_{m-1}}, \]

\[ w_{B_{m-2}} = (f_{m-1}f_m(H)u_{B_{m-2}}^*)_{B_{m-2}}, \]

\[ \vdots \]

\[ w_{B_k} = (f_{k+1}f_{k+2} \cdots f_m(H)u_{B_{k+1}}^*)_{B_k}. \]

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

**Theorem 11.** If the characteristic polynomial of the matrix \( A \) is square-free, a cyclic vector for \( A \) can be obtained in \( O(n^3 + m \cdot n^2) \) steps.

6. **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 uses a divide-and-conquer approach as in Section 2. We first present its global structure, before going into details. We also set out separately a technique of splitting, and finally give the complete description.

6.1. **Overall Strategy**

First a shift Hessenberg form for the given matrix is to be computed. Then our strategy is to split it into two parts, whose sizes remain under
control. The matrix $H$ has the following form:

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

**Notation 4.** For every $I \subset [1, n]$, $J \subset [1, n]$, we denote by $H_{i, j}$ the submatrix formed with the rows of $H$ in $I$ and the columns of $H$ 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 $n_k$.

The splitting consists in finding a matrix $H_{\text{split}}$ similar to $H$ with the form

$$
H_{\text{split}} = \begin{bmatrix}
H'_{B_1, B_1} & H'_{B_1, B_j} \\
0 & H'_{B_j, B_j}
\end{bmatrix}
$$

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 $H'_{B_1, B_j}$ and $H'_{B_j, B_j}$, in order to find $v_{B_1}, v_{B_j}$ which are cyclic vectors of $H'_{B_1, B_1}$ and $H'_{B_j, B_j}$ respectively.

It remains to compute a vector $u'$ cyclic for $H_{\text{split}} - v_{B_1}$ and $v_{B_j}$ being known. Changing the current basis for the original one, we finally transform $u'$ into a cyclic vector $u$ for $H$.

### 6.2. The Splitting

We give a lemma for splitting the matrix into two submatrices. Before stating this lemma, we bring to light a technical but important phenomenon that appears when permuting rows and columns of shift Hessenberg matrices in order to move the blocks. Consider the following shift Hessenberg matrix:

$$
H = \begin{bmatrix}
H_{B_1, B_1} & H_{B_1, B_2} & \cdots & H_{B_1, B_m} \\
0 & H_{B_2, B_2} & \cdots & H_{B_2, B_m} \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & H_{B_m, B_m}
\end{bmatrix}
$$
Let us perform the permutation of rows and columns which exchanges \( H_{B_1, B_1} H_{B_k, B_k} \). This leads to the matrix \( H_{\text{swap}} \):

\[
H_{\text{swap}} = \begin{bmatrix}
H_{B_k, B_k} & 0 & 0 & H_{B_k, B > k} \\
H_{B_{[2, k-1]}, B_k} & H_{B_{[2, k-1]}, B_{[2, k-1]}} & H_{B_{[2, k-1]}, B_1} & H_{B_{[2, k-1]}, B > k} \\
\vdots & \vdots & \vdots & \vdots \\
H_{B_1, B_k} & H_{B_1, B_{[2, k-1]}} & H_{B_1, B_1} & H_{B_1, B > k} \\
H_{B > k, B_k} & H_{B > k, B_{[2, k-1]}} & H_{B > k, B_1} & H_{B > k, B > k}
\end{bmatrix}.
\]

We now use the algorithm for computing a shift Hessenberg form of \( H_{\text{swap}} \). This leads to the matrix \( H' \):

\[
H' = \begin{bmatrix}
H_{B_1, B_1}' & H_{B_1, B_2}' & \cdots & H_{B_1, B_k}' & \cdots & H_{B_1, B_m}' \\
0 & H_{B_2, B_2}' & \cdots & H_{B_2, B_k}' & \cdots & H_{B_2, B_m}' \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \cdots & H_{B_k, B_k}' & \cdots & \vdots \\
0 & 0 & \cdots & 0 & \cdots & H_{B_m, B_m}'
\end{bmatrix}.
\]

The next lemma establishes a relation between the minimal polynomial of the block \( H_{B_1, B_1} \) and the minimal polynomial of the block \( H_{B_k, B_k} \).

**Notation 5.** We denote by \( \varepsilon_k \) the vector from the basis of \( K^n \) such that \( (\varepsilon_k)_{B_k} = (1, 0, \ldots, 0) \).

**Lemma 11.** Let \( f_k \) be the minimal polynomial of the block \( H_{B_k, B_k} \) of the matrix \( H \), and let \( f'_1 \) be the minimal polynomial block \( H_{B_1, B_1}' \) of the matrix \( H' \) obtained in the previous transformations. We have that \( f_k \) divides \( f'_1 \).

**Proof.** We have that \( f_k \) divides the minimal polynomial of \( H \) restricted to \( \varepsilon_k \). While swapping from \( H \) to \( H_{\text{swap}} \), the vector \( \varepsilon_k \) becomes the first vector of the new basis.

The shift Hessenberg reduction algorithm computes a matrix whose first block is a companion matrix whose minimal polynomial is the minimal polynomial for the first vector of the new basis. That vector is \( \varepsilon_k \), by Remark 1. Thus \( f'_1 \) is the minimal polynomial of \( \varepsilon_k \), which is a multiple of \( f_k \).
Now we can state our crucial lemma for splitting shift Hessenberg matrices. The results in Section 2.4 will be reused in a decisive way.

**Lemma 12 (Splitting the matrix).** Let $H$ be a shift Hessenberg matrix. It is always possible to find a shift Hessenberg matrix $H_{\text{split}}$ and an invertible matrix $Q$ such that $H = Q^{-1} H_{\text{split}} Q$ with $H_{\text{split}}$ of the form

$$H_{\text{split}} = \begin{bmatrix} H'_{B_1, B_1} & H'_{B_1, B_j} \\ 0 & H'_{B_j, B_j} \end{bmatrix},$$

and such that one of those three possibilities occurs: either

1. $H'_{B_1, B_1}$ is a companion block with size $\geq \frac{2}{3}n$, and $H'_{B_j, B_j}$ has size $\leq \frac{1}{3}n$, or
2. $H'_{B_1, B_1}$ is a companion block with size $\leq \frac{2}{3}n$, and $H'_{B_j, B_j}$ has size $\leq \frac{2}{3}n$, or
3. both blocks $H'_{B_1, B_1}$ and $H'_{B_j, B_j}$ are just shift Hessenberg matrices with size smaller than $\frac{2}{3}n$.

The computation of $H_{\text{split}}$ and $Q$ can be performed in $O(n^3)$ operations.

**Proof.** Two main cases are considered.

First case: there exists $k \in [1, m]$ such that $n_k \geq \frac{2}{3}n$. 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 columns, the block $B_k$ is put in the first place. This gives a matrix $H_{\text{swap}}$ which is not shift Hessenberg. We now can clean up the matrix $H_{\text{swap}}$ by applying the reduction algorithm producing a shift Hessenberg matrix. The size of the first block can only grow, by Lemma 11, and then remains not smaller than $\frac{2}{3}n$. This gives matrix $H_{\text{split}}$ shaped as in (1) at cost $O(n^3)$.

Second case: for each $j \in [1, m]$, $n_j < \frac{2}{3}n$. Let us first assume 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 $\sum_{j=1}^{i_0} n_j < \frac{2}{3}n$. Then $I = B_1 \cup B_2 \cup \cdots \cup B_{i_0}$ and $J = B_{i_0+1} \cup B_{i_0+2} \cup \cdots \cup B_m$ both satisfy $n_I \leq \frac{2}{3}n$ and $n_J \leq \frac{2}{3}n$. Indeed, since $n_I \setminus B_{i_0+1} < \frac{1}{3}n$, we have that $n_I \leq \frac{1}{3}n + n_{i_0+1} \leq \frac{2}{3}n$. Then the matrix $H_{\text{split}}$ is the unchanged matrix $H$. 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 columns, we put the block $H_{I, I}$ in the first place, then clean up the resulting matrix by the shift Hessenberg reduction algorithm in $O(n^3)$ steps. The first block cannot decrease in size. As a result the size of the remaining block stays 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).
6.3. 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 7, this is done in $O(n^3)$ operations. This operation only needs to be performed once, and is not needed in the recursive steps.

**Step 2**: Splitting the matrix. We perform the splitting resulting from Lemma 12, and obtain two submatrices $H'_{B_1, B_1}$ and $H'_{B_2, B_2}$. We recursively apply the algorithm to 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 the equations (4) and (5) from the results of the algorithm applied at $H'_{B_1, B_1}$ and $H'_{B_2, B_2}$. By Lemma 10 we can construct a cyclic element for $H_{\text{split}}$, at cost $O(n^3)$.

**Step 4**: Reconstruction of the cyclic element in the original basis. From a cyclic vector of $H_{\text{split}}$, 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.

**Theorem 12.** Given a matrix $A \in M_n(K)$ whose minimal polynomial is square-free, a cyclic vector for $A$ can be computed in $O(n^3)$ elementary operations.

**Proof.** The proof is easily done by induction, as in Section 2.4, by observing in addition that the cost of all steps before and after recursion are bound by $O(n^3)$.

6.4. Applications to Normal Bases

Before stating our result, we recall that the best known complexity for deterministic algorithms for finding a normal basis is $O((n^2 + \log q)n^2)$. Those algorithms are due to E. Bach, J. Driscoll, and J. Sallit and independently to H. W. Lenstra [3]. Considering probabilistic algorithms, J. von zur Gathen [18] presents an algorithm with expected time $O^{-}(n^2 \log q)$, where $g = O^{-}(h)$ means that there exists $k$ such that $g = O(h(\log h)^k)$ ("soft $O$" notation). Fast algorithms for polynomial multiplication and for gcd's are used. When using classical arithmetic, the complexity of this algorithm turns to $O(n^3 + n^2 \log q)$ or $O(n^3 \log n)$, depending on the relative size of $q$ and $n$ [6].
COROLLARY 3. When $n$ is prime to the characteristic $p$ of $\mathbb{F}_q$, a normal basis of $\mathbb{F}_{q^n}$ can be computed in $O(n^3)$ elementary operations on the data of a matrix representing the Frobenius map, that matrix being computed at cost $O(n^3 + n^2 \log q)$.

Proof. The minimal polynomial of the Frobenius map is $X^n - 1$, which is square-free when $\gcd(n, q) = 1$. Given the matrix $F_n$ of the Frobenius map [computed in $O(n^3 + n^2 \log q)$], we are able to compute a cyclic vector for the Frobenius map in $O(n^3)$. This vector is a normal element. 

We now consider the case where $n = p^t$, where $p$ is the characteristic of the field. In that case, $X^n - 1 = (X - 1)^n$. Let $H$ be a shift Hessenberg matrix for the Frobenius automorphism, and let $\varepsilon_1, \ldots, \varepsilon_m$ be basis vectors as in Notation 5. Let the minimal polynomial of $H$ be $X^n - 1$. It is also the least common multiple of the minimum polynomial of the $\varepsilon_i$'s, thus $X^n - 1$ is the minimal polynomial of one of the $\varepsilon_i$ which is thus cyclic. Now if $\varepsilon_l$ were cyclic with $l < m$, we would permute the basis vectors in order to have $\varepsilon_l$ in the first position. After reduction to shift Hessenberg form, 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 $\varepsilon_m$ would remain unchanged. This contradicts the fact that $\varepsilon_l$ is cyclic, since putting it in the first position would lead to a companion matrix.

To sum up, a reduction of any representation of the Frobenius map into a shift Hessenberg form exhibits $\varepsilon_m$ which necessarily is cyclic. Knowing normal bases for $\mathbb{F}_{q^1}$ and $\mathbb{F}_{q^2}$, one can construct a normal element for $\mathbb{F}_{q^1q^2}$ when $\gcd(n_1, n_2) = 1$ [1, 3].

COROLLARY 4. For all $n$, a normal basis of $\mathbb{F}_{q^n}$ can be computed deterministically in $O(n^3 + n^2 \log q)$ elementary operations.

7. COMPUTATION OF THE FROBENIUS FORM

7.1. Definitions and Notation

In the sequel we shall need a specific notation for the columns of a shift Hessenberg matrix.
NOTATION 6. Let $H$ be a shift Hessenberg matrix:

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

(10)

We recall that $e_i$, the unit vector from $K^n$ such that $(e_i)_{B_i} = ^t(1, \ldots, 0)$ (Notation 5). We set $e_i = f_i(H)e_i$. Informally, $e_i$ is seen to be the vector "above" the last column of the $i$th block in $H$.

We here shall describe how to compute the expanded Frobenius form of a matrix. A preliminary computation is done, to obtain a basis for the characteristic subspaces of $A$, using the algorithm presented in Section 2 (and thus the factorization of the characteristic polynomial of $A$ is required). Then the expanded Frobenius form for each characteristic subspace is computed, using the following method.

7.2. Computing the Frobenius Form for Characteristic Subspaces

In view of Theorem 8, we only need to consider the case where the characteristic polynomial of the considered matrix is $C(X) = p(X)^r$, with $r > 1$ and $p(X)$ irreducible.

We apply the reduction process to get a shift Hessenberg form $H$ for the matrix, as in (10). The minimal polynomial of the block $H_{B_i, B_i}$ is thus $p^s$ for some $s_i$.

The vector $(e_i)_{B_i}$ is seen as a polynomial, as is any projection on a block $B_j$, since as a rule $B_j^{K}$ is identified with $K[X]/p^{r_j}K[X]$. A power of $p$ can appear as a factor of $(e_i)_{B_i}$, and using gcd computations we can write $(e_i)_{B_i} = e_i^\dagger p^{r_i}$, with $e_i^\dagger$ prime to $p$. A favorable situation appears when each $e_i$ is such that $r_i > s_i$. In that case, we introduce the vectors

$$
\begin{align*}
\varepsilon'_2 &= e_2 - e_2^\dagger p^{r_2-s_2}(H)e_1, \\
& \quad \vdots \\
\varepsilon'_m &= e_m - e_m^\dagger p^{r_m-s_m}(H)e_1.
\end{align*}
$$

For those vectors we have that $(p^s(H)e'_i)_{B_i} = 0$, since $p^s(H)e_i = e_i$. The first vector $\varepsilon_1$ is left unchanged. The vectors $\varepsilon_1, \varepsilon'_2, \ldots, \varepsilon'_m$ yield a basis in
which the matrix has the following form:

\[
\begin{bmatrix}
  C_{p^{s_1}} & 0 \\
  0 & H'
\end{bmatrix}.
\]

The process is next applied to \( H' \).

Otherwise, there exists \( \varepsilon_j \) such that \( r_j < s_j \). We then observe the following property:

**Lemma 13.** Let \( \varepsilon_j \) be a vector such that \( r_j < s_j \) in the above situation. Then the minimal polynomial of the restriction of \( H \) to \( \varepsilon_j \) is \( p^t \) with \( t > s_1 \).

**Proof.** To determine the minimal polynomial of \( \varepsilon_j \), we have to find the lowest \( t \) such that \( p^t(H)\varepsilon_j = 0 \). We first have to compute \( p^s(H)\varepsilon_j \) in order to have that \( (p^s(H)\varepsilon_j)_{B_j} \) vanishes. This leads to \( p^s(H)\varepsilon_j = \varepsilon_j \). The coordinates of \( \varepsilon_j \) with indexes in block \( B_{j-1} \) must vanish in their turn. Therefore, having in mind that \( (H^s)_{B_{j-1}} = H_{B_{j-1}} \), we will apply the minimal polynomial of \( (\varepsilon_j)_{B_{j-1}} \), which is \( p^\lambda \) for some \( \lambda \). We get \( p^\lambda \cdot p^s(H)\varepsilon_j = p^\lambda \cdot (H(\varepsilon_j)_{B_1}^* + v(j - 2)) \), where \( v(j - 2) \) is a vector with support in blocks \( B_1 \cup \cdots \cup B_{j-2} \). We proceed this way, and at each step we get a new relation \( p^\lambda \cdot v(k - 1) = p^\lambda \cdot (H(\varepsilon_j)_{B_1}^* + v(k - 1)), \) where the support of \( v(k - 1) \) lies in \( B_1 \cup \cdots \cup B_{k-1} \).

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

\[
p^\lambda \cdot v(k - 1) = p^\lambda \cdot (H(\varepsilon_j)_{B_1}^* + v(k - 1)).
\]

We thus are left with determining the minimum exponent \( l \) such that

\[
p^l[p^\lambda \cdot v(k - 1)] \equiv 0 \mod p^{s_1},
\]

and we write \( v(k - 1)_{B_1} = p^{r_0}v(1)^r \), where \( \gcd(p, v(1)^r) = 1 \). Two cases are to be considered.

If \( r_0 > \lambda_2 + \cdots + \lambda_{j-1} + r_j \), then \( l = s_1 - (\lambda_2 + \cdots + \lambda_{j-1} + r_j) \) and the exponent of the minimal polynomial of \( \varepsilon_j \) is

\[
t = l + \lambda_2 + \cdots + \lambda_{j-1} + s_j = s_1 - r_j + s_j > s_1.
\]
In the other case, \( r_0 < \lambda_2 + \cdots + \lambda_{j-1} + r_j \). Then \( l = s_1 - r_0 \), and the exponent of the minimal polynomial of \( \varepsilon_j \) is

\[
t = s_1 - r_0 + \lambda_2 + \cdots + \lambda_{j-1} + s_j > s_1 - r_j + s_j > s_1.
\]

To complete the algorithm in that case, we permute the basis vectors in order to have \( \varepsilon_j \) in the first position. By applying the reduction algorithm, we compute a new shift Hessenberg form, whose first block is a companion matrix whose minimal polynomial is the minimal polynomial of \( \varepsilon_j \). By Lemmas 13 and 11, the size of the first block has 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 \( i \), and we apply the above method for the favorable case, or when we get a companion matrix.

### 7.3. Complexity

Either cleaning up the matrix, when it is possible, or augmenting the size of the first block is done at cost \( O(n^3) \). The number of times these processes are performed is bounded by \( r \). Notice that matrices for changing of 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:

\[
O(n^3 r_1) + O(n^3 r_2) + \cdots + O(n^3 r_d) \leq O(n^3 (r_1 + r_2 + \cdots + r_d)).
\]

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 13.** Knowing the factorization of its characteristic polynomial, the Frobenius form of a matrix \( A \) and the matrix for changing basis can be computed in \( O(n^3 m_A) \), where \( m_A \) is the number of factors of the characteristic polynomial of \( A \), counted with multiplicities. The asymptotic average complexity over a finite field is \( O(n^3 \log n) \).

### 7.4. Without the Factorization of the Characteristic Polynomial

We show how to perform the computation of the Frobenius form without the knowledge of the factorization of the characteristic polynomial. This is of particular interest when the field has characteristic zero. The idea is the following: the computation of the shift Hessenberg form of the matrix \( A \)
yields a partial factorization of the characteristic polynomial $C$ of $A$. Using a "factor refinement" process, the characteristic polynomial $C$ can be factorized into $C = P_1^{r_1} \cdots P_k^{r_k}$, with $\gcd(P_i, P_j) = 1$ when $i \neq j$. The algorithm for Theorem 8 can be applied to compute the restriction of the matrix to the subspaces $\ker P_i(A)^{r_i}$, $i = 1, \ldots, k$. Then the previous algorithm can be applied, making the (eventually false) assumption that the $P_i$'s are irreducible. If that assumption leads to polynomials which are not powers of $P$, then new factors are obtained, and new subspaces are computed.

Here are the details of the algorithm:

**Input:** Matrix $A$ (whose characteristic polynomial is denoted $C$).

**Step 1:** Computation of a shift Hessenberg form $H$ for $A$. If a companion matrix is obtained, then the algorithm stops returning $H$.

**Step 2:** "Factor refinement." The diagonal companion blocks of the matrix $H$ yield factors $f_1, \ldots, f_m$ such that $f_1 \cdots f_m = C$. The factor refinement is to extract pairwise gcd's from that list recursively, until we get a list $P_1^{r_1}, \ldots, P_k^{r_k} = C$, with $\gcd(P_i, P_j) = 1$ for $i \neq j$. The subspaces $V_i = \ker P_i(A)^{r_i}$ are here called pseudocharacteristic subspaces.

**Step 3:** Computing the restriction of $H$ to the pseudocharacteristic subspaces. This is done using the algorithm from Theorem 8. This algorithm is recursive, and at each step a shift Hessenberg form for each $V_i$ is computed. Diagonal blocks appearing in shift Hessenberg forms along this process may show new factors $P_i$. In that case, the refinement process is applied, and new pseudocharacteristic subspaces are computed. This ends in the knowledge of the shift Hessenberg form of the restriction of $H$ to new pseudocharacteristic subspaces $V_i' = \ker P_i'(A)^{r_i'}$, with $C = P_1^{r_1'} \cdots P_k^{r_k'}$.

**Step 4:** Computing the Frobenius form for each pseudocharacteristic subspace. We apply the algorithm described in Section 7.2, making the (possibly wrong) assumption that we are faced with a matrix $H$ whose characteristic polynomial is $p^r$ with $p$ irreducible. As previously, the vectors $(e_i)_{B_i}$ being seen as polynomials, then powers of $p$ are extracted from $e_i$, using repeated gcd computations. If all gcd's are powers of $p$, then we end with $(e_i)_{B_i} = e_i^r p^{r_i'}$, with $e_i^r$ prime to $p$. If some gcd is now a power of $p$, then a factor of $p$ has been found, and refinement for new subspaces is done, go to step 3. If that is not the case, then the same favorable case may appear as previously, and the process goes on (cleaning the first block). If an unfavorable case occurs, then we put in first position a vector $e_i$ such that $r_i > s_i$, and apply the shift Hessenberg reduction. Then the minimal polynomial of $e_i$ appears on the first block: if it is a power of $p$, then the size of the first block has grown (same argument as for Lemma 13); if not, a new factor of $p$ appears, and refinement is done; go to step 3.
We give the following crude estimate for the complexity. The most expensive computations are the shift Hessenberg reductions and the evaluation of polynomials at matrices for computing the restriction at characteristic subspaces. These operations have complexities bounded by $O(n^3 + n^2 m_A^2)$. Since there are at most $m_A$ characteristic subspaces, these steps occur at most $m_A$ times. Thus the complexity is bounded by $O(m_A n^3 + m_A^3 n^2)$, for an average of $O(n^3 \log n)$ over $\mathbb{F}_q$.

8. CONCLUSION

The algorithm from Section 6 has been implemented using Axiom [11], and tested for computing normal bases of $\mathbb{F}_q^n$ when $n$ is prime to $q$. It gives better results than the algorithm already implemented in Axiom. 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 make the following remark: the cost of such an algorithm is the same as the cost for "dividing" and for "recombining" only once. The second is the use of the shift Hessenberg form, which is very sparse on the average, and which reflects some 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.

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

An expanded version of this exposition is available as [2].

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 about 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 evaluating complexities. We are indebted to the referee who helped us in rewriting the manuscript more concisely.
REFERENCES


15. A. Poli, A deterministic construction of normal bases with complexity $O(n^3 + n \log(n) \log \log(n) \log q)$, *J. Symbolic Comput.* 19:305–319 (1995).


Received 27 June 1994; final manuscript accepted 8 April 1996