Square–Free decomposition in Finite Characteristic: an application to Jordan Form computation*

Elisabetta Fortuna and Patrizia Gianni
Dipartimento di Matematica, Università di Pisa
Via Buonarroti 2, I - 56127 Pisa, Italy
email: fortuna@dm.unipi.it, gianni@dm.unipi.it

Introduction

In ([GT]) has been addressed the problem of the computation of the square–free decomposition for univariate polynomials with coefficients in arbitrary fields. The complete square–free decomposition can be computed over arbitrary fields of finite characteristic solely assuming that the field satisfies the **Condition P** of Seidenberg ([Se]), which has been proven equivalent to the ability of computing such decompositions (see also [MRR]). If we assume that the field is only an effective field (i.e. a field $\mathbf{K}$ where there are constructive procedures for performing rational operations in $\mathbf{K}$ and for deciding whether or not two elements in $\mathbf{K}$ are equal), it is possible to obtain a weaker decomposition into powers of relatively prime factors, not necessarily square–free, but such that within each factor the roots have constant multiplicity. Although this is a partial decomposition, much useful information can be gathered from this result. As an application we present an algorithm to compute the Jordan form of a matrix over an arbitrary effective field. In particular we show how to handle problems of inseparability while splitting invariant factors and constructing a symbolic Jordan form.

The computation of normal forms of a matrix, in particular of the Jordan form, is a very important task and has many useful applications, so it has been widely studied for many years and many efficient algorithms, sequential and parallel ([O], [L], [Gi1], [Gi2], [Ol], [KKS], [RV]), are already available for its computation. There are already algorithms which compute the Jordan form of a matrix over general fields ([GD], [RV]), but they are based on dynamic evaluation ([D5]) and we want to avoid the use of such a scheme, that requires a special computational environment. Storjohann ([St]) has given a new algorithm for computing the rational canonical form which has a deterministic complexity of $\mathcal{O}(n^3)$ but he does not compute the transition matrix with the same complexity. Steel's ([S]) algorithm for computing generalized Jordan form has a complexity of $\mathcal{O}(n^4)$ but requires factoring polynomials into irreducibles. Kaltofen et. al. ([KKS]) give fast parallel algorithms for canonical forms and make the observation that one could compute a symbolic Jordan form from a rational canonical form by splitting the invariant factors using gcd's and square–free decompositions. They require the computation of complete square–free decompositions and thus also require that $\mathbf{K}$ be a perfect field with the ability to compute p^{th} roots. They also don't compute the transition matrix. Ozello ([O]) presents an algorithm for computing the rational canonical

*This work was partially supported by CNR and MURST

form which is deterministic with complexity $\mathcal{O}(n^4)$, and leaves the question of faster probabilitic approaches for future work. Giesbrecht ([Gi2]) gives a probabilistic algorithm whose complexity is essentially the same as matrix multiplication but requires choosing n "good" random vectors simultaneously thus giving only a probability of $1/4$ of making a successful choice.

Our aim is to obtain a general sequential algorithm, of a complexity comparable with most of the existing algorithms, that works in the widest possible setting, without requiring particular computing resources and hence of easy and straightforward implementation. Because of our hypothesis, in general, our algorithm will produce a *symbolic Jordan form* ([K], [RV]), but the main difference with the other available algorithms based on dynamic evaluation is that our algorithm is a rational algorithm, since all the computations take place in the given field, except for the output and eventually the computation of the inverse of the transition matrix. To obtain all the information on the symbolic roots of the characteristic polynomial (multiplicities and recognition) we, at first, transform the given matrix A into a *pseudo–rational* form, i.e. a block diagonal matrix, similar to A, with companion matrices on the diagonal without requiring any kind of divisibility of the associated polynomials. Then we refine the factorization of the characteristic polynomial, given by the polynomials whose companion matrices are on the diagonal of the pseudo–rational form, using partial square–free decomposition and gcd computations, so that we can identify the same roots in different blocks and also we reduce, as much as possible without factorization, the degree of the defining polynomials for the eigenvalues.

The pseudo–rational form is computed with a probabilistic algorithm of complexity $\mathcal{O}(n^3)$ such that each independent random choice is verifiable with probability better than $1 - 1/n$ of success. We derive this probabilistic algorithm from one for the computation of the rational form, which has a complexity of $\mathcal{O}(n^4)$, and is obtained via a straightforward analysis of the properties of the minimal polynomial that leads to a natural way to construct invariant subspaces.

1. Definitions and Preliminaries

In this section we recall the main definitions and many results, classical ([G], [W]) and more recent ([O]), that will be used in the following. We will omit most of the proofs and we will give only the ones that seem to be new or that we think will be useful to clarify the exposition.

The *Jordan Form* of a matrix $A \in M(n, \mathbf{K})$ is a matrix $J(A)$, similar to A, of the form

$$J(A) = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_r \end{pmatrix}$$

where

$$J_i = J(\alpha_i, s_i) = \begin{pmatrix} \alpha_i & 1 & & \\ & \ddots & \ddots & \\ & & \alpha_i & 1 \\ & & & \alpha_i \end{pmatrix} \in M(s_i, \overline{\mathbf{K}})$$

($\overline{\mathbf{K}}$ is the algebraic closure of the field $\mathbf{K}$). Each block $J(\alpha_i, s_i)$ is called the *Jordan block* associated with the eigenvalue α_i of multiplicity s_i.

The α_i can also be symbolically represented as roots of factors of the characteristic polynomial, but in this case we need the ability to identify symbols associated with different polynomials and to compute their multiplicities.

We will use the following result ([MRR], [GT]):

Theorem 1.1. *Let* $\mathbf{K}$ *be an effective field of characteristic* $p > 0$ *and* $f \in \mathbf{K}[x]$. *Then it is possible to compute* $q_1, \ldots, q_t \in \mathbf{K}[x]$ *such that :*

(i) $f = \prod_j q_j{}^{s_j}$

(ii) the q_j*'s are pairwise relatively prime*

(iii) for each j *there exists a separable polynomial* $\tilde{q}_j(x)$ *and an integer* $e_j \geq 0$ *such that* $q_j(x) = \tilde{q}_j(x^{p^{e_j}})$.

(We recall that a polynomial q *is separable iff* $gcd(q, q') = 1$*).*

Definition 1.2. *We call* p**–separable polynomial** *any polynomial* $q(x)$ *that can be expressed as a separable polynomial evaluated in* p*–th powers of* x*, (i.e.* $q(x) = \tilde{q}(x^{p^e})$*, with* $\tilde{q}(x)$ *a separable polynomial). We call the decomposition of a polynomial* f *as a product of powers of coprime* p*–separable polynomials described in the previous theorem a* **partial square–free decomposition** *of* f.

The representation in Theorem 1.1 can be obtained as follows:

• **Algorithm SQFREE–PART : Partial Square–Free**
Input : $f \in K[x]$
Output : $(q_1, e_1, s_1), \ldots, (q_t, e_t, s_t)$ s.t. $f = \prod q_j(x^{p^{e_j}})^{s_j}$, $q_j(x)$ separable, $\gcd(q_i, q_j) = 1$, for $i \neq j$.
Initialize Result:= empty
Step 1.
$(P_1, \ldots, P_k, Q)$:= basicSquareFree(f)
($P_1, \ldots, P_k$ are separable, Q s.t. Q'=0, i.e. $Q \in K$ or $Q(x) = Q_1(x^p)$)
Result:=(($P_i, 0, i$) for i in $1 \ldots k$)
if degree(Q) = 0 then return Result

Step 2
Q_1:= divideExponents(Q, p) (Divide by p the exponents of Q)
Step 3
$((q_1, e_1, s_1), \ldots, (q_r, e_r, s_r))$:= SQFREE–PART($Q_1$)
Result := append((($q_i, e_i + 1, s_i$) for i in $1 \ldots r$),Result)

• **Algorithm basicSquareFree(f)**
Input : $f \in K[x]$
Output : $(P_1, \ldots, P_k, Q)$ s.t. $f = Q \prod P_i{}^i$, $P_1, \ldots, P_k$ separable and Q s.t. Q'=0, i.e. $Q \in K$ or $Q(x) = Q_1(x^p)$.

$C_1 := gcd(f, f')$	$C_1 := Q\ P_2 P_3{}^2 \ldots P_k{}^{k-1}$
$B_1 := f/C_1$	$B_1 := P_1 P_2 \ldots P_k$
for i in 1.. while $B_i \neq 1$ repeat	
	$C_i := Q\ P_{i+1} P_{i+2}{}^2 \ldots P_k{}^{k-1}$
	$B_i := P_i P_{i+1} \ldots P_k$
$B_{i+1} := gcd(C_i, B_i)$	$B_{i+1} := P_{i+1} \ldots P_k$
$C_{i+1} := C_i/B_{i+1}$	$C_{i+1} := Q\ P_{i+2} P_{i+3}{}^2 \ldots P_k{}^{k-i-1}$
$P_i := B_i/B_{i+1}$	

Remark 1.3. All of the roots in $\mathbf{K}$ of a p–separable polynomial have the same multiplicity, so the partial square–free decomposition allows us to distinguish the multiplicities of the roots of the given polynomial: if α is a root of a factor of f, say $q_j(x)^{s_j} = \tilde{q}_j(x^{p^{e_j}})^{s_j}$, then α is a root of f of multiplicity $p^{e_j}s_j$.

Remark 1.4. In the case of perfect fields, since we can compute p–th roots of the coefficients, we can substitute $q(x^p)$ with $(\tilde{q}(x))^p$, and hence we can obtain a decomposition with $e_j = 0$.

So any polynomial, with coefficients in *any* effective field, can be expressed as the product of powers of pairwise relatively prime p–separable polynomials. For simplicity of notation, we will denote by m_j the multiplicity of the roots of each p–separable factor q_j, meaning that

(i) $m_j = p^{e_j}s_j$, if $f = \prod_j q_j(x^{p^{e_j}})^{s_j}$ with $e_j \geq 0$ (non-perfect field)

(ii) $m_j = s_j$, if $f = \prod_j q_j(x)^{s_j}$ (perfect field).

Let us now fix our notations and recall some classical results (see, for instance, [G] and [W]).

Notation 1.5. *Given a matrix $A \in M(n, \mathbf{K})$ and a vector $v \in \mathbf{K}^n$, we will denote by*

(i) $m_A \in \mathbf{K}[x]$ *the minimal polynomial of A, i.e. the monic polynomial of minimum degree such that $m_A(A) = 0$*

(ii) $m_{v,A} \in \mathbf{K}[x]$ *the minimal polynomial of the vector v with respect to the matrix A, i.e. the monic polynomial of minimum degree such that $m_{v,A}(A)v = 0$.*

Proposition 1.6. *For any $A \in M(n, \mathbf{K})$, there exists $v \in \mathbf{K}^n$ such that $m_{v,A} = m_A$.*

If $v \in \mathbf{K}^n$ is a vector such that $m_{v,A} = m_A$ and $d = deg\ m_A$, then the set $\mathcal{S} = \{v, Av, \ldots, A^{d-1}v\}$ consists of linearly independent vectors, which generate an A–invariant subspace of $\mathbf{K}^n$ (also called a *cyclic* subspace). A classical result, that in Section 4 we will examine again from an algorithmic point of view, assures that it is possible to complete $\mathcal{S}$ to a basis $\mathcal{B}$ of $\mathbf{K}^n$ in such a way that, if N denotes the transition matrix from the canonical basis $\mathcal{E}$ to the basis $\mathcal{B}$, we have

$$NAN^{-1} = \begin{pmatrix} C_{m_A} & 0 \\ 0 & A_1 \end{pmatrix}$$

where

$$C_{m_A} = \begin{pmatrix} 0 & 0 & & -a_0 \\ 1 & \ddots & \ddots & \vdots \\ & \ddots & 0 & -a_{d-2} \\ & & 1 & -a_{d-1} \end{pmatrix}$$

is the *companion matrix* of $m_A = \sum_{i=0}^{d} a_i x^i$ and A_1 is a square matrix of order $n - d$. Moreover the minimal polynomial of the matrix A_1 is a divisor of m_A.

By iterating this process, we eventually find a matrix $R(A)$, similar to A, such that

$$R(A) = \begin{pmatrix} C_{m_A} & & & \\ & C_{m_{A_1}} & & \\ & & \ddots & \\ & & & C_{m_{A_k}} \end{pmatrix} \in M(n, \mathbf{K})$$

where $C_{m_A}, C_{m_{A_1}}, \ldots, C_{m_{A_k}}$ are companion matrices of polynomials m_{A_i} such that $m_{A_{i+1}} \mid m_{A_i}$ for any $i = 0, \ldots, k-1$ (we let $m_A = m_{A_0}$).

Definition 1.7. *The matrix $R(A)$ is called the rational canonical form of A and the polynomials $m_A, m_{A_1}, \ldots, m_{A_k}$ are called the invariant factors of A.*

It is well known that the rational canonical form of a matrix is unique.

If we drop the divisibility condition on the polynomials m_i, we get the weaker notion of *pseudo–rational* form:

Definition 1.8. *A block diagonal matrix*

$$B = \begin{pmatrix} B_1 & & \\ & \ddots & \\ & & B_r \end{pmatrix}$$

is a pseudo–rational form of a matrix A if B is similar to A and each block B_i on the diagonal is a companion matrix.

The Jordan form of a companion matrix C_q can be immediately deduced from the partial square–free decomposition of q, since the structure and the number of the Jordan blocks of $J(C_q)$ are determined by the number and the multiplicity of the roots of q. However, if we tried to compute the Jordan form of a matrix in pseudo–rational form working separately on each of its cyclic blocks C_{q_i}, since we don't have an explicit representation of the roots of the q_i's, we would not be able to recognize when two symbolic roots correspond to the same root. So we need to improve our strategy.

We will show that the ability to perform partial square–free decompositions and gcd's is sufficient to overcome this difficulty and to compute the Jordan form and a transition matrix.

The paper is organized as follows:

- In Section 2 we describe a way to compute the Jordan form and a transition matrix for the companion matrix of a polynomial q, if a decomposition of q in powers of relatively prime p–separable polynomials of has been previously computed.
- Section 3 is the core of the algorithm: we transform a matrix already in pseudo–rational form into a block diagonal matrix such that the blocks are companion matrices of powers of p–separable polynomials, with different symbolic roots recognized, so that the algorithm described in Section 2 can be applied to each block separately.
- In Section 4 we present, as preparatory material for the next section, an algorithm to compute the rational canonical form and a transition matrix, whose complexity is $\mathcal{O}(n^4)$. Many algorithms for the computation of the rational canonical form are already known, but the one we present here will be used in Section 5 and adapted to obtain an algorithm for the computation of a pseudo–rational form.

- In Section 5, in order to complete the procedure, we present an algorithm that, with $\mathcal{O}(n^3)$ field operations, transforms a matrix into pseudo–rational form and computes a transition matrix.

2. Jordan Form of a companion matrix $A = C_q$

In this section we will find the Jordan form and a transition matrix for C_q, the companion matrix of a polynomial q, in three steps:
Case 1: Companion matrix of a power of a p–separable polynomial, with only one eigenvalue
Case 2: Companion matrix of a power of a p–separable polynomial
Case 3: Companion matrix of a general polynomial.

The only work required in Case 1 and 2 is the computation of the transition matrix and, even though the construction is known ([O],[W]), we prefer to recall it for completness. We are not aware of any descriptions in the literature of the construction of the transition matrix in Case 3.

Case 1. Assume that q is a power of a p–separable polynomial and that $A = C_q$ has only one eigenvalue. In this case $q = (x-\alpha)^s$ or $q = (x^{p^e} - \alpha)^s$, and its only root $a = \alpha$ (resp. $a = \alpha^{1/p^e}$) has multiplicity $m = s$ (resp. $m = p^e s$). Then the Jordan form of A is

$$J = J(a,m) = \begin{pmatrix} a & 1 & & \\ & \ddots & \ddots & \\ & & a & 1 \\ & & & a \end{pmatrix} \in M(m, \mathbf{K}).$$

If $e_1, \ldots, e_m$ are the vectors of the canonical basis $\mathcal{E}$ of $\mathbf{K}^m$, then q is the minimal polynomial of e_m w.r.t. J, the vectors $e_m, Je_m, J^2e_m, \ldots, J^{m-1}e_m$ are linearly independent and form a cyclic basis $\mathcal{S}$ of $\mathbf{K}^m$. An easy induction on $k \in \mathbf{N}$ proves that

$$J^k e_m = \sum_{i=1}^{m} \binom{k}{i+k-m} a^{i+k-m} e_i \tag{1}$$

(here the binomial coefficient $\binom{k}{j}$ is 0 when $j < 0$). Hence, the matrix

$$M = M(a,m) = \begin{pmatrix} 0 & \cdots & & 1 \\ & \cdots & & \\ \vdots & & 1 & \\ & 1 & 2a & \\ 1 & a & a^2 & \cdots & a^{m-1} \end{pmatrix}.$$

is such that $MAM^{-1} = J$, since its columns are precisely the coordinates of the vectors e_m, Je_m, $\ldots$, $J^{m-1}e_m$ w.r.t $\mathcal{E}$ (hence it represents the transition matrix from the basis $\mathcal{S}$ to the basis $\mathcal{E}$ of $\mathbf{K}^m$).

Case 2. Let $A = C_q$ and assume that q is a power of a p–separable polynomial having r distinct roots, $\alpha_1, \ldots, \alpha_r$, all with the same multiplicity, say m. The roots α_i can either be given as explicit elements of $\mathbf{K}$ or as new symbols representing the distinct roots of q. Then J is formed by r Jordan blocks $J_1 = J(\alpha_1, m), \ldots, J_r = J(\alpha_r, m)$ of order m associated respectively to the roots of q

$$J = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_r \end{pmatrix} \in M(rm, \mathbf{K}).$$

As for a transition matrix from A to J, one can easily see that the minimal polynomial of the vector

$$y = \begin{pmatrix} \frac{e_m}{e_m} \\ \hline \vdots \\ \hline e_m \end{pmatrix} \in \mathbf{K}^{rm}$$

with respect to J is q, so that the vectors $y, Jy, \ldots, J^{rm-1}y$ form a cyclic basis $\mathcal{S}$ of $\mathbf{K}^{rm}$. As before the transition matrix M is precisely the matrix expressing the vectors of the basis $\mathcal{S}$ in terms of $\mathcal{E}$. Since we have

$$J^k y = \begin{pmatrix} J_1^k & & \\ & \ddots & \\ & & J_r^k \end{pmatrix} \begin{pmatrix} \frac{e_m}{\vdots} \\ \hline e_m \end{pmatrix} = \begin{pmatrix} \frac{J_1^k e_m}{\vdots} \\ \hline J_r^k e_m \end{pmatrix},$$

it follows from (1) that the matrix M is formed by r strips $m \times rm$, one for each root a_i, where as before $a_i = \alpha_i$ (resp. $a_i = \alpha_i^{1/p^e}$):

$$M = \begin{pmatrix} \frac{S(a_1)}{\vdots} \\ \hline S(a_r) \end{pmatrix}.$$

To be precise, the $(i,j) - th$ element of M is given by

$$M(i,j) = \binom{j-1}{r(i-1,m)+j-m} (a_{q(i-1,m)+1})^{r(i-1,m)+j-m},$$

where $r(i-1,m)$ and $q(i-1,m)$ denote respectively the remainder and the quotient of the division of $i-1$ by m.

Case 3. Consider now the case when A is the companion matrix of a general polynomial q and assume that a decomposition $q = \prod_j q_j(x)^{s_j}$ of q into p–separable and coprime factors is known (for instance a partial square–free decomposition). Then the matrix A is similar to a block diagonal matrix of the form

$$B = \begin{pmatrix} C_{q_1^{s_1}} & & \\ & \ddots & \\ & & C_{q_r^{s_r}} \end{pmatrix},$$

as both A and B have only one invariant factor, that is their minimal polynomial q. So

$$J(A) = J(B) = \begin{pmatrix} J(C_{q_1^{s_1}}) & & \\ & \ddots & \\ & & J(C_{q_r^{s_r}}) \end{pmatrix},$$

which can be computed as in Case 2.

As for the transition matrix from A to B, we have the following:

Lemma 2.1. *Let A be a matrix in $M(n,\mathbf{K})$ and let $w \in \mathbf{K}^n$ be a vector such that $m_{w,A} = m_A$. Suppose that $m_A = gh$ and let $d_1 = \deg g$, $d_2 = \deg h$. If we set $w_1 = h(A)w$ and $w_2 = g(A)w$, then*

$m_{w_1,A} = g$ *and* $m_{w_2,A} = h$

if $(g,h) = 1$, the vectors $\{w_1, Aw_1, \dots, A^{d_1-1}w_1, w_2, Aw_2, \dots, A^{d_2-1}w_2\}$ are linearly independent.

Proof. (1) Since $g(A)w_1 = g(A)h(A)w = 0$ and $m_A = gh$, it is clear that g is the minimal polynomial of w_1 with respect to A, so that $\{w_1, Aw_1, \dots, A^{d_1-1}w_1\}$ is a set of independent vectors. The same holds for w_2 and h.
(2) Let $g = \sum_{i=0}^{d_1} a_i x^i$ and $h = \sum_{i=0}^{d_2} b_i x^i$. Since by hypothesis $m_{w,A}$ has degree $d_1 + d_2$, the set $\mathcal{S} = \{w, Aw, \dots, A^{d_1+d_2-1}w\}$ consists of linearly independent vectors which generate an A–invariant subspace $< \mathcal{S} >$. It is easy to check that the vectors $\{w_1, Aw_1, \dots, A^{d_1-1}w_1, w_2, Aw_2, \dots, A^{d_2-1}w_2\}$ belong to the subspace $< \mathcal{S} >$ and that the square matrix of order $d_1 + d_2$ having as columns the coordinates of such vectors w.r.t the basis $\mathcal{S}$ is

$$S = \begin{pmatrix}
b_0 & 0 & & & & a_0 & 0 & & \\
b_1 & b_0 & \ddots & & & a_1 & a_0 & \ddots & \\
b_2 & b_1 & \ddots & \ddots & & a_2 & a_1 & \ddots & 0 \\
\vdots & b_2 & \ddots & \ddots & 0 & \vdots & a_2 & \ddots & a_0 \\
b_{d_2} & \vdots & \ddots & \ddots & b_0 & \vdots & \vdots & \ddots & a_1 \\
0 & b_{d_2} & \ddots & \ddots & b_1 & a_{d_1} & \vdots & \ddots & a_2 \\
 & \ddots & \ddots & \ddots & b_2 & 0 & a_{d_1} & \ddots & \vdots \\
 & & \ddots & \ddots & \vdots & & \ddots & \ddots & \vdots \\
 & & & 0 & b_{d_2} & & & 0 & a_{d_1}
\end{pmatrix}.$$

So $S = S(h,g)$ is the Sylvester matrix of the polynomials h and g; since they are coprime and consequently without common roots, S is invertible, which proves the thesis. □

From the proof of the previous lemma it follows immediately :

Corollary 2.2. *Let $A = C_q$ be a companion matrix in $M(n,\mathbf{K})$ and suppose that $q = gh$ where $g = \sum_{i=0}^{d_1} a_i x^i$ and $h = \sum_{i=0}^{d_2} b_i x^i$, and $(g,h) = 1$. Let S=S(h,g) the Sylvester matrix of h and g. Then*

$$S^{-1}AS = \begin{pmatrix} C_g & 0 \\ 0 & C_h \end{pmatrix}$$

where C_g and C_h are the companion matrices of g and h respectively.

Proof. Since $A = C_q$, we have that $m_A = q = m_{e_1,A}$ (where e_1 is the first vector of the canonical basis). With respect to the proof of Lemma 2.1, we have the additional information that $n = d_1 + d_2$, so that $\mathbf{K}^n$ is the direct sum of the two cyclic A–invariant subspaces generated respectively by $w_1 = h(A)e_1$ and $w_2 = g(A)e_1$. □

The previous corollary, applied recursively to the factors of q, easily yields a transition matrix from A to the block diagonal matrix B introduced above. Finally, a transition matrix from B to $J = J(B)$ is evidently given by a block diagonal matrix having on the diagonal the transition matrices from $C_{q_i^{s_i}}$ to $J(C_{q_i^{s_i}})$ determined in Case 2.

3. Jordan form of a matrix in pseudo–rational form

As we remarked earlier, the results of the previous section cannot be directly applied to the blocks C_{q_i} of a pseudo–rational form, unless all the roots are explicitly known, because in this way we would not be able to identify roots that appear in different blocks. In this section we give algorithms that will transform the given pseudo–rational matrix, splitting the blocks so that each corresponds to a power of a p–separable polynomial and if two blocks have a root in common then they have exactly the same roots, i.e. the corresponding p–separable polynomials are coprime or identical.

Example 3.1. Consider the matrix

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \in M(3, \mathbf{Q}).$$

This matrix is in rational form and has $x^2 - x, x - 1$ as invariant factors. The invariant factors are square–free, so if we apply the algorithm described in Section 2 we obtain

$$J = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & a & 0 \\ 1 & b & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where a, b are roots of $x^2 - x$.

It is clear from this example that the form obtained is unsatisfactory: we introduce two "unnecessary" symbols, and we don't explicitely "identify" one of the roots with one of the eigenvalues already found. But, if we take into account the fact that $x^2 - x, x - 1$ are the invariant factors and hence that $x - 1$ is a factor of $x^2 - x$, by division we obtain $x^2 - x = x(x - 1)$ and so $a = 0$ and $b = 1$.

The following procedure refines a partial square–free decomposition of the q_i's, splitting as much as possible the p–separable factors into distinct and relatively prime factors, via gcd computations.

• **refinePolySep**(h, q)
Input :

- *h a separable polynomial*
- *q any polynomial*

Output :

- $([g_1, \ldots, g_k], \overline{q})$ *where: $[g_1, \ldots, g_k]$ is a list of pairwise relatively prime and separable polynomials such that $h = \prod_j g_j$ and $\overline{q}$ is a polynomial such that $q = \overline{q} \prod g_i^{r_i}$, with $r_i \geq 0$ and $(g_i, \overline{q}) = 1$ for each i.*

begin
$h = 1$ => return ([],q)
$q = 1$ => return ([h],q)
$d := \gcd(h, q)$
(*finalList*, $\overline{q}$) := **refinePolySep**(d, $\frac{q}{d}$)
if $\frac{h}{d} \neq 1$ then *finalList* := [$\frac{h}{d}$, *finalList*]
return (*finalList*, $\overline{q}$)
end

Proof of the algorithm. First of all we remark that all the recursive calls are legitimate since the first argument is always a factor of a separable polynomial and hence separable itself.
If $h = 1$ the thesis is true vacuously.
We assume inductively that the recursive call to **refinePolySep**(d, $\frac{q}{d}$) is correct: this means that *finalList* = $[g_1, \ldots, g_s]$ is a list of pairwise relatively prime and separable polynomials s.t. $d = \prod_j g_j$ and $\overline{q}$ is a polynomial s.t. $\frac{q}{d} = \overline{q} \prod g_i^{r_i}$ and $(g_i, \overline{q}) = 1$ for each i. We have to show that:

(i) $\frac{h}{d}$ is coprime with each element of *finalList*. This follows from the fact that h is separable and all the elements in finalList are divisors of d.

(ii) $(\overline{q}, \frac{h}{d}) = 1$. If $f = (\overline{q}, \frac{h}{d})$ then by inductive hypothesis $f \mid \overline{q} \mid \frac{q}{d}$ and hence $f \mid (\frac{q}{d}, \frac{h}{d}) = 1$.

(iii) $h = \frac{h}{d} \prod_j g_j$ and $q = \overline{q} \prod g_i^{s_i}$. We have $h = \frac{h}{d}\, d = \frac{h}{d} \prod_j g_j$, since we assumed that $d = \prod_j g_j$, and also $q = d\, \frac{q}{d} = \overline{q} \prod g_i^{r_i+1}$, since $\frac{q}{d} = \overline{q} \prod g_i^{r_i}$.

□

Remark 3.2. In practice all the non–trivial quotients $(\frac{h}{d}, \frac{d}{d_1}, \ldots, \frac{d_{s-1}}{d_s}, d_s)$ appear in *finalList*, where $d_{i+1} = (d_i, \frac{q}{\prod_0^i d_i})$ and $d_0 = d$. Hence it is clear that, when h is separable, any common factor (not necessarily irreducible) that appears in h and q with different multiplicities will be discovered by the algorithm.

Example 3.3. Suppose $h = f_0 f_1 f_3$ and $q = (f_1 f_2)^2 {f_3}^3 f_4$ (the $f_i's$ are not necessarily irreducible). Then **refinePolySep**$(h, q) = ([f_0, f_1, f_3], {f_2}^2 f_4)$.

Remark 3.4. *Let $h(x)$ and $q(x) \in \mathbf{K}[x]$, then*

$$(h(x), q(x)) = 1 \iff (h(x^k), q(x^k)) = 1$$

Lemma 3.5. *Let $q(x) = f(x)g(x)$ be a p–separable polynomial and $(f, g) = 1$. Then f and g are p–separable.*

Proof. We have $q(x) = \tilde{q}(x^{p^e})$, with $\tilde{q}(x)$ a separable polynomial, because $q(x)$ is p–separable; we proceed by induction on e.
If $e = 0$, then $q(x)$ is separable and the thesis is trivially satisfied.
If $e > 0$, write $q(x) = \tilde{q}(x^{p^e}) = q_1(x^p) = f(x)g(x)$. By differentiating we obtain $f'(x)g(x) + f(x)g'(x) = 0$ and so, since $(f, g) = 1$, $f' = g' = 0$ and there exist f_1 and g_1 such that $f(x) = f_1(x^p)$ and $g(x) = g_1(x^p)$ and $q_1(x) = f_1(x)g_1(x)$. By the previous remark $(f_1, g_1) = 1$; moreover $q_1(x) = \tilde{q}(x^{p^{e-1}})$ is p–separable, hence the thesis follows from the inductive hypothesis. □

Corollary 3.6. *Let* $([g_1, \ldots, g_k], \bar{q}) =$ **refinePolySep**(h, q). *If* q *is a* p*–separable polynomial, then* $\bar{q}$ *is* p*–separable.*

Proof. Since $(g_i, \bar{q}) = 1$ for each i, $q = \bar{q}(\prod g_i^{r_i})$ is a decomposition of a p–separable polynomial which satisfies the hypothesis of the previous lemma. □

• **refinePolyP–Sep**(h, q)
Input :

- h *and* q *which are* p*–separable polynomials*

Output :

- $([g_1, \ldots, g_k], \bar{q})$ *where:* $[g_1, \ldots, g_k]$ *is a list of pairwise relatively prime and* p*–separable polynomials such that* $h = \prod_j g_j^{s_j}$, *and* $\bar{q}$ *is a* p*–separable polynomial s.t.* $q = \bar{q} \prod g_i^{r_i}$, *with* $r_i \geq 0$ *and* $(g_i, \bar{q}) = 1$ *for each* i.

begin
$(h_1, e) :=$ **insDeg**$^{(*)}(h)$
$(q_1, r) :=$ **insDeg**(q)
if $e \leq r$ then

$$H := h_1(x)$$
$$Q := q_1(x^{p^{r-e}})$$

else

$$H := q_1(x)$$
$$Q := h_1(x^{p^{e-r}})$$

$(rlist, \overline{Q}) :=$ **refinePolySep**(H, Q)
$e \leq r \Rightarrow$ return(**power** $^{(**)}(e, rlist)$, **power** $(e, \overline{Q})$)
$hlist := \{g \in rlist \ : \ g \mid Q\}$
$hlist := [\overline{Q}, hlist]$
$\bar{q} := \prod_{g \in rlist - hlist} g$
$hlist :=$ **power**$(r, hlist)$
$\bar{q} :=$ **power**$(r, \bar{q})$
return $(hlist, \bar{q})$
end

$(*)$ The function **insDeg**(h) returns the separable polynomial $h_1(x)$ and the exponent e such that $h(x) = h_1(x^{p^e})$ in characteristic $p > 0$, $h(x)$ and 0 otherwise.

$(**)$ The function **power**(e, pol) (resp. **power**$(e, listPol)$) substitutes x^{p^e} for x in the polynomial pol (resp. in all the polynomials of the list $listPol$).

Proof of the algorithm. By definition of *insDeg*, H is separable and hence the arguments for the call of **refinePolySep** are valid. Let $(rlist, \overline{Q}) = ([g_1, \ldots, g_k], \overline{Q})$ be the returned result from

that call. Then we have that the g_i's are separable and pairwise relatively prime and $H = \prod g_i$ and $Q = \overline{Q} \prod {g_i}^{r_i}$.
Case $e \leq r$: In this case $h(x) = H(x^{p^e}) = \prod g_i(x^{p^e})$ and $q(x) = \overline{Q}(x^{p^e}) \prod g_i(x^{p^e})^{r_i}$. Since the function *power* transforms a separable polynomial into a p–separable one, and by the previous remark it preserves relative primality, the result satisfies the required properties.
Case $e > r$: In this case *rlist* contains divisors of both $h(x) = Q(x^{p^r})$ and $q(x) = H(x^{p^r})$ while $\overline{Q}(x^{p^r})$ is a divisor of $h(x)$. As the function *power* transforms separable polynomials into p–separable ones and preserves relative primeness, the $g_i(x^{p^r})$ are p–separable and relatively prime and also relatively prime with $\overline{Q}(x^{p^r})$. Moreover $\overline{Q}(x^{p^r})$ is p–separable by the previous corollary. Hence it is enough to select the right factors among the elements of *rlist* in order to obtain the thesis. □

• **refine**$([h_1, \ldots, h_s], q)$
Input :

- $[h_1, \ldots, h_s]$ *a list of pairwise relatively prime and p–separable polynomials*
- q *any polynomial*

Output :

- $[g_1, \ldots, g_k]$ *a list of pairwise relatively prime and p–separable polynomials s.t.* $h_i = \prod_j {g_j}^{s_{i,j}}$ and $q = \prod_j {g_j}^{r_j}$.

begin
hlist := $[h_1, \ldots, h_s]$
factq := []
sqlist := **partialSqfrFactors** $^{(*)}(q)$
finalList := []
for *gg* in *sqlist* repeat
  $\overline{g}$:= *gg*
  for *hh* in *hlist* repeat
    (*newList*, $\overline{g}$) := **refinePolyP–Sep**(*hh*, $\overline{g}$)
    finalList := [*finalList*, *newList*]
  if $\overline{g} \neq 1$ then *factq* := $[\overline{g}, factq]$
  hlist := *finalList*
return [*finalList*, *factq*]
end

(∗) The function **partialSqfrFactors**(q) returns the list of the p–separable factors of q, i.e. a list $[q_1, \ldots, q_k]$ of pairwise coprime and p–separable polynomials s.t. $q = \prod_i {q_i}^{s_i}$, where $q_i(x) = \tilde{q}_i(x^{p^{e_i}})$ with $\tilde{q}_i$ separable polynomial in the case of non perfect fields or q_i square–free otherwise.

Proof of the algorithm. Clearly the g_i's are pairwise relatively prime and p–separable; the last two conditions follow from the properties of the output of the previous algorithms. □

Remark 3.7. In the case of a perfect field the partial square–free factorization in **refine** can be deleted, so only one *for–loop* is required. At the end the square–free factors of the remaining $\overline{g}$ must be added to the list. Moreover in this case the function **refinePolyP–Sep** can be eliminated and **refinePolySep** can be directly called.

• **rootsBasis**$([m_1, \ldots, m_k])$
Input :

– $[m_1, \ldots, m_k]$ *a list of polynomials*

Output :

– $[[f_i, lroots(i), [n_{i1}, \ldots, n_{ik}]], i = 1, \ldots s]$ *where for every i:*
f_i *is a p–separable polynomial with* $\prod_j f_j^{n_{ij}} = m_i$ *and* $(f_i, f_j) = 1$ *for all* $i \neq j$
$lroots(i)$ *is the list of the distinct roots of* f_i,
$[n_{i1}, \ldots, n_{ik}]$ *are integers representing the multiplicity (possibly zero) of each* f_i *as a factor of* m_j.

begin
$factors := []$
for i in $1 \ldots k$ repeat $factors :=$ **refine**$(factors, m_i)$
$result := []$
for $fact$ in $factors$ repeat
$lroots :=$ **allRoots**$^{(*)}(fact)$
$lexp :=$ **exponents**$^{(**)}(fact, [m_1, \ldots, m_k])$
$result := ([fact, lroots, lexp], result)$
return $result$
end

$(*)$ The function **allRoots**(f), with f p–separable, returns the list of all the distinct roots of f. More precisely if $f(x) = f_1(x^{p^e})$ with f_1 separable polynomial of degree s, **allRoots**$(f) = [\alpha_1, \ldots, \alpha_s]$, where the α_i's either belong to $\mathbf{K}$ or are symbols representing the roots.

$(**)$ The function **exponents**$(f, [m_1, \ldots, m_k])$ returns the list of the multiplicities of f as a factor of m_i.

At this point if we apply the function **rootsBasis** to the list of the polynomials m_i's whose companion matrices are on the diagonal of a pseudo–rational B form of a matrix A (to the invariant factors, in the case of the rational form), the information returned allows us to apply to each block of B the procedure described in Case 3 of the previous section. Using the factorizations $\prod_j f_j^{n_{ij}} = m_i$ we construct a block diagonal matrix similar to B, and hence to A, (and a transition matrix) such that the blocks are the companion matrices of the powers of p–separable factors f_i's returned by **rootsBasis**. In this way we avoid the possible double–naming of the roots and reduce as much as possible the degree of the defining polynomials for the eigenvalues. In order to complete the process, compute the Jordan form and a transition matrix, we can apply the algorithm described in Case 2 of Section 2 to each block separately.

4. Computation of the rational canonical form

The procedure described so far applies to a matrix already in pseudo–rational form, so what is lacking is a procedure to transform by similarity a matrix into a pseudo–rational form. This will

be done in Section 5 by adapting the algorithm for the computation of the rational canonical form that we are going to present in this section.

The first step is to find an A–invariant subspace: this will be done constructing a vector v whose minimal polynomial with respect to A is equal to the minimal polynomial of A. This construction is based on the following propositions.

Proposition 4.1. *Let $A \in M(n, \mathbf{K})$. Given $v \in \mathbf{K}^n$, it is possible to compute the minimal polynomial $m_{v,A}$ of v with respect to A. If we let $d = \deg m_{v,A}$, then this computation requires at most $\mathcal{O}(dn^2)$ arithmetic operations over $\mathbf{K}$.*

Proof. For any given v vector in $\mathbf{K}^n$, the minimal polynomial $m_{v,A}$ can be found by iteratively computing the vectors $v, Av, A^2v, \ldots, A^kv$ and looking for a monic relation of linear dependence among them. The first time such a relation

$$b_0v + b_1Av + b_2A^2v + \ldots + A^kv = 0$$

is found by means of a Gaussian elimination, the polynomial $b_0 + b_1x + b_2x^2 + \ldots + x^k$ is the minimal polynomial of v w.r.t A. □

At this point if the polynomial $m_{v,A}$ is such that $m_{v,A}(A) = 0$ then $m_{v,A} = m_A$, otherwise we propose to use the vector v and the polynomial $m_{v,A}$ to complete the construction. If $m_{v,A}(A) \neq 0$ there exists a vector $w \in \mathbf{K}^n$ such that $m_{v,A}(A)w \neq 0$ (for instance any non zero row of $m_{v,A}(A)$). Also, if w is such a vector , $m_{w,A}$ does not divide $m_{v,A}$, (otherwise $m_{v,A}(A)w = 0$), hence the polynomial $lcm(m_{v,A}, m_{w,A})$ has degree strictly bigger that the degree of $m_{v,A}$. Moreover we will show how to construct a vector $y \in \mathbf{K}^n$ such that $m_{y,A} = lcm(m_{v,A}, m_{w,A})$ and hence, since $deg\ m_{y,A} > deg\ m_{v,A}$, a vector z such that $m_{z,A} = m_A$, after at most $d = deg(m_A)$ steps.

Lemma 4.2. *Let $f, g \in \mathbf{K}[x]$. It is possible to construct polynomials $p_1, p_2, q_1, q_2 \in \mathbf{K}[x]$ such that:*

(i) $f = p_1p_2, \quad g = q_1q_2,$

(ii) $(p_1, p_2) = 1, \quad (q_1, q_2) = 1, \quad (p_2, q_2) = 1$

(iii) $lcm(f, g) = p_2q_2.$

Proof. Let $d = (f, g)$ and let h, k be polynomials such that $f = dh$ and $g = dk$. Evidently $(h, k) = 1$, while d and h may not be coprime. Set $\delta = (d, h)$, we can write $f = \frac{d}{\delta}(\delta h)$ with $(\delta h, k) = 1$. Iterating this procedure, after a finite number of steps we find a polynomial $a(x) \in K[x]$ dividing d such that $(\frac{d}{a}, ah) = 1$ and every prime dividing a divides h.
Define $p_1 = \frac{d}{a}$, $p_2 = ah$, $q_1 = a$ and $q_2 = \frac{d}{a}k$. We have:

- $(p_1, p_2) = (\frac{d}{a}, ah) = 1,$
- $(q_1, q_2) = (a, \frac{d}{a}k) = (a, k) = 1$, since each prime dividing a divides h, which is relatively prime with k,
- $(p_2, q_2) = (ah, \frac{d}{a}k) = (ah, k) = (h, k) = 1,$
- $\mathrm{lcm}(f, g) = \frac{fg}{(f,g)} = \frac{(dh)(dk)}{d} = dhk = (ah)(\frac{d}{a}k) = p_2q_2.$

□

Corollary 4.3. *Let $A \in M(n, \mathbf{K})$ and $v, w \in \mathbf{K}^n$. If $m_{v,A} = f$ and $m_{w,A} = g$, then there exists $y \in \mathbf{K}^n$ such that $m_{y,A} = lcm(f, g)$.*

Proof. If f and g are coprime, one can easily check that it is enough to take $y = v + w$. Otherwise, let $p_1, p_2, q_1, q_2 \in \mathbf{K}[x]$ be the polynomials obtained by applying Lemma 4.2 to f and g. If we set $v_1 = p_1(A)v$, by Lemma 2.1 we have $m_{v_1,A} = p_2$; similarily, if $w_1 = q_1(A)w$, then $m_{w_1,A} = q_2$. Since $(p_2, q_2) = 1$, by the same argument used at the beginning of the proof we have that $m_{v_1+w_1,A} = p_2 q_2 = \text{lcm}(f, g)$. It is so enough to take $y = v_1 + w_1$. □

Proposition 4.4. *It is possible to construct a vector $z \in \mathbf{K}^n$ such that $m_A = m_{z,A}$. If we let $d = \deg m_A$, then this computation requires at most $\mathcal{O}(dn^3)$ arithmetic operations over $\mathbf{K}$.*

Proof. Let v be a vector in $\mathbf{K}^n$. If $m_{v,A}(A) = 0$, then $m_{v,A} = m_A$; otherwise consider a vector w such that $m_{v,A}(A)w \neq 0$, (any non zero row of $m_{v,A}(A)$ will suffice). By iterating the construction of the previous corollary we complete the proof. The complexity is dominated by the cost of computing $m_{v,A}(A)$ which is $\mathcal{O}(dn^3)$. □

Assume therefore that we have computed a vector v such that $m_A = m_{v,A}$ and let *deg* $m_A = d$. So the set $\mathcal{S} = \{v, Av, \ldots, A^{d-1}v\}$ is linearly independent and generates an A–invariant subspace $< \mathcal{S} >$. It is clear that completing the set $\mathcal{S}$ to a basis $\mathcal{B}$ allows to convert A by similarity to a block–upper–triangular form.

Using the properties of the *dual space* $(\mathbf{K}^n)^*$ consisting of all linear transformations from $\mathbf{K}^n$ to $\mathbf{K}$, it is however possible to complete $\mathcal{B}$ to a basis in such a way that the subspace generated by the added vectors is itself A–invariant. In this way the matrix A will be transformed by similarity into a block diagonal matrix.

For any basis $\mathcal{B} = \{v_1, \ldots, v_n\}$ of $\mathbf{K}^n$, we will denote by $\mathcal{B}^* = \{v_1{}^*, \ldots, v_n{}^*\}$ the basis of $(\mathbf{K}^n)^*$, called the *dual basis* of $\mathcal{B}$, consisting of the functionals defined by $w_i{}^*(w_j) = \delta_{i,j}$ (where $\delta_{i,j}$ denotes the Kronecker delta).

Let us finally recall that:
– if M denotes the transition matrix from the canonical basis $\mathcal{E}$ to a basis $\mathcal{B}$ of $\mathbf{K}^n$, then the transition matrix from the basis $\mathcal{B}^*$ to basis $\mathcal{E}^*$ in $(\mathbf{K}^n)^*$ is ${}^tM$
– if $A^* : V^* \to V^*$ is the linear map defined by $A^*(T) = T \circ A$, then the matrix associated to A^* with respect to the basis $\mathcal{E}^*$ is ${}^tA$.
Our use of dual basis is explained by the next:

Proposition 4.5. *([J]) Let $A \in M(n, \mathbf{K})$ and v be a vector such that $m_A = m_{v,A}$ with deg $m_A = d$. Let $\mathcal{B} = \{w_1 = v, w_2 = Av, \ldots, w_d = A^{d-1}v, w_{d+1}, \ldots, w_n\}$ be a basis of $\mathbf{K}^n$ obtained completing to a basis the independent set $\{w_1 = v, w_2 = Av, \ldots, w_d = A^{d-1}v\}$. Then the functionals $w_d{}^*, A^*w_d{}^*, \ldots, (A^*)^{d-1}w_d{}^*$ are linearly independent and the set*

$$W = \{x \in \mathbf{K}^n \mid w_d{}^*(x) = 0, A^*w_d{}^*(x) = 0, \ldots, (A^*)^{d-1}w_d{}^*(x) = 0\}$$

is a vectorial subspace which is A–invariant and such that

$$\mathbf{K}^n = Span(v, Av, \ldots, A^{d-1}v) \oplus W.$$

In order to construct the set W of the previous proposition we prove the following results:

Proposition 4.6. *Let $A \in M(n, \mathbf{K})$ and v be a vector such that $m_A = m_{v,A}$ with $\deg m_A = d$. It is possible to complete the independent set $\mathcal{S} = \{w_1 = v, w_2 = Av, \ldots, w_d = A^{d-1}v\}$ to a basis $\mathcal{B}$ of $\mathbf{K}^n$ and compute the transition matrix M from the canonical basis $\mathcal{E}$ to the new basis $\mathcal{B}$ with $\mathcal{O}(n^2 d)$ field operations.*

Proof. The construction we describe will complete $\mathcal{S}$ to a basis and simultaneously compute the corresponding transition matrix M, i.e. the inverse of the matrix whose columns are the coordinates, with respect to $\mathcal{E}$, of the vectors of the basis $\mathcal{B}$.

We consider the $2n \times d$ matrix R obtained by stacking the vectors of the coordinates of the vectors $w_i \in \mathcal{S}$, with respect to $\mathcal{E}$, and the first d columns of the $n \times n$ identity matrix. Call R_1 the matrix obtained from R via a complete stepwise Gaussian elimination by columns. R_1 is such that each column contains a pivot element $r_{k_i,i} = 1$, with $1 \leq i \leq d$ and $1 \leq k_i \leq n$, and $r_{k_i,j} = 0$ if $j \neq i$. Call $P = \{k_1, \ldots, k_d\}$ the set of indexes corresponding to the rows which contain a pivot element. With this information we can complete the independent set $\mathcal{S}$ to a basis of $\mathbf{K}^n$: we add those vectors $e_i \in \mathcal{E}$ such that $i \notin P$, call them $e_{i_1}, \ldots, e_{i_{(n-d)}}$.

At this point we concatenate the matrix R_1 and the $2n \times (n-d)$ matrix whose columns are the vectors ${}^t(e_{i_j}, e_{d+j})$, $1 \leq j \leq (n-d)$ and we obtain a $2n \times n$ matrix R_2. We can now continue our stepwise Gaussian elimination by columns on R_2, however due to the zero structure of the added columns, no additional multiplications are required and we can complete the work with the same overall complexity. Everytime we use a pivot element to zero out an element in the top half of R_2, the only required operation is to negate the element and place it in the corresponding position of the bottom half of R_2, i.e. given an element $r_{i_j s}$ in R_2 such that $r_{i_j s} \neq 0$ with $1 \leq i_j \leq n, 1 \leq j \leq n-d, 1 \leq s \leq d$ and $i_j \notin P$,ments we set $r_{(n+d+j)s} = -r_{i_j s}$ and then we set $r_{i_j s} = 0$. We make a permutation of the columns of R_2 in order to obtain an identity matrix in the first n rows and then extract the submatrix consisting of the last n rows as our resulting transition matrix M. □

We use the matrix M to complete our construction. We have that the d–th row of M is precisely the vector C of the coordinates of $w_d{}^*$ with respect to $\mathcal{E}^*$ (the transition matrix from $\mathcal{B}^*$ to $\mathcal{E}^*$ is ${}^tM$). Moreover the coordinates of the functionals $A^* w_d{}^*, \ldots, (A^*)^{d-1} w_d{}^*$ are given by the vectors ${}^tAC, ({}^tA)^2 C, \ldots, ({}^tA)^{d-1} C$, and we have:

Corollary 4.7. *In order to compute the A–invariant subspace W of Proposition 4.5 it is enough to solve the $d \times n$ linear system*

$$\begin{pmatrix} C \\ {}^tAC \\ ({}^tA)^2 C \\ \vdots \\ ({}^tA)^{d-1} C \end{pmatrix} X = 0$$

whose complexity is $\mathcal{O}(dn^2)$.

Corollary 4.8. *If the set of vectors $v, Av, \ldots, A^{d-1}v$ is completed to a basis of $\mathbf{K}^n$ by means of a basis of W, N is the matrix having the vectors of this basis of $\mathbf{K}^n$ as columns and $M = N^{-1}$ is the matrix computed in Proposition 4.6, then we have*

$$N^{-1}AN = \begin{pmatrix} C_{m_A} & 0 \\ 0 & B \end{pmatrix}$$

with B a square matrix of order $n-d$.

It is evident that the procedure described here above, applied recursively, after a finite number of steps leads to compute the rational canonical form $R(A)$ of a matrix A and a transition matrix.

The complexity is dominated by the cost of finding cyclic vectors associated with the invariant factors. As explained above the cost of this step is $\mathcal{O}(d_i n^3)$, where $d_i = \deg m_i$ (m_i are the invariant factors of A). Since $\sum_i d_i = n$ the overall complexity is $\mathcal{O}(n^4)$.

5. Computation of a pseudo–rational form In this section we propose a probabilistic algorithm to construct a pseudo–rational form of a matrix A, which requires $\mathcal{O}(n^3)$ field operations.

The basic idea is to find a vector v such that A can be transformed by similarity into the form

$$\begin{pmatrix} C_{m_{v,A}} & 0 \\ 0 & A_1 \end{pmatrix}$$

and therefore, after a finite number of steps, into a pseudo–rational form.

Instead of constructing a vector whose minimal polynomial coincides with the minimal polynomial of the matrix, we are able to decide if a randomly chosen vector can be used to split $\mathbf{K}^n$ into the direct sum of two A–invariant subspaces. This construction is based on the following result, obtained by modifying the hypothesis of Proposition 4.5.

Proposition 5.1. *Let $A \in M(n, \mathbf{K})$ and $v \in \mathbf{K}^n$; let $d = \deg m_{v,A}$. Assume $\mathcal{B} = \{w_1 = v, w_2 = Av, \ldots, w_d = A^{d-1}v, w_{d+1}, \ldots, w_n\}$ is a basis of $\mathbf{K}^n$ obtained completing to a basis the independent set $\{w_1 = v, w_2 = Av, \ldots, w_d = A^{d-1}v\}$. Let F be the subspace of $(\mathbf{K}^n)^*$ generated by the functionals $w_d{}^*, A^*w_d{}^*, \ldots, (A^*)^{d-1}w_d{}^*$ and let*

$$W = \{x \in \mathbf{K}^n \mid w_d{}^*(x) = 0, A^*w_d{}^*(x) = 0, \ldots, (A^*)^{d-1}w_d{}^*(x) = 0\}.$$

If F is A^*–invariant, then W is an A–invariant subspace of $\mathbf{K}^n$ such that:

$$\mathbf{K}^n = Span(v, Av, \ldots, A^{d-1}v) \oplus W.$$

This result can be proved exactly as Proposition 4.5 (see [J]), observing only that the A^*–invariance of F holds here by hypothesis, not as a consequence of the dropped hypothesis that $m_{v,A} = m_A$.

Thus a random vector v can be used to split the space if F as above is A^*–invariant, which can be easily tested as follows.

If we denote with M the transition matrix from the canonical basis $\mathcal{E}$ to the basis $\mathcal{B}$ in Proposition 5.1 and if C is the d–th row of M, then we have:

Proposition 5.2. *With the notations of the proposition above, F is A^*–invariant if and only if*

$$rank \begin{pmatrix} C \\ {}^tAC \\ ({}^tA)^2 C \\ \vdots \\ ({}^tA)^d C \end{pmatrix} = d.$$

If this is the case, in order to compute the subspace W it is enough to solve the linear system

$$\begin{pmatrix} C \\ {}^tAC \\ ({}^tA)^2 C \\ \vdots \\ ({}^tA)^{d-1} C \end{pmatrix} X = 0.$$

This proposition gives us a procedure for testing whether or not a particular vector allows us to split the space. We need to examine the probability that a randomly chosen vector will pass the test. If in fact the randomly chosen vector happens to have the same minimal polynomial as the matrix, then by Proposition 4.5 F as above will be A^*–invariant and the vector will yield a direct sum decomposition into A–invariant subspaces. This condition is stronger than we need, but it allows us to use the following result of Giesbrecht:

Lemma 5.3. *([Gi2]). Let L be a subset of $\mathbf{K}$ containing at least n^2 elements. Then*

$$Prob_{v \in L^n} \{m_A = m_{v,A}\} \geq 1 - 1/n$$

If $\mathbf{K}$ has fewer than n^2 elements then we can make a small algebraic extension of $\mathbf{K}$. Giesbrecht needs to simultaneously find a complete family of successful vectors and thus arrives at an overall probability of success of $1/4$. The previous corollary allows us to check each vector separately with probability greater than $(1 - 1/n)$ of success.

The complexity of testing the random vector and then using it to generate a direct sum splitting $\mathbf{K}^n$ is $\mathcal{O}(dn^2)$ field operations where $d = \deg m_{v,A}$. By repeating this construction on the complementary matrix A_1, we eventually arrive at a pseudo–rational form of A and a transition matrix. Since the sum of the degrees of the minimal polynomials of the blocks in the pseudo–rational form is n, we arrive at the overall complexity of $\mathcal{O}(n^3)$.

References

[D5] J.Della Dora, C.Di Crescenzo, and D.Duval. About a new method for computing in algebraic number fields. In *Proc.EUROCAL'85*, volume LNCS 204, pages 289–290, 1985.

[G] F. R. Gantmacher. *The theory of matrices.* Chelsea Pub. Co., New York, 1959.

[Gi1] M. Giesbrecht. *Nearly optimal algorithms for canonical matrix forms.* PhD thesis, University of Toronto, Toronto, Canada, 1993.

[Gi2] M. Giesbrecht. Nearly optimal algorithms for canonical matrix forms. *SIAM J. Comp.*, 24(5), 1995.

[GT] P. Gianni, and B. Trager. Square-free algorithms in finite characteristic. *Journal of AAECC*, 7(1), 1996.

[GD] T. Gómez Diaz. *Quelques applications de l'évaluation dynamique.* PhD thesis, Université de Limoges, Limoges, France, 1994.

[K] E. Kaltofen. Sparse hensel lifting. In *Proc. EUROCAL'85*, volume 2 of *LNCS*, pages 4–17, 1985.

[J] H. G. Jacob. Another proof of the rational decomposition theorem. *Amer. Math. Monthly*, 80(10):1131–1134, 1973.

[KKS] E. Kaltofen, M. S. Krishnamoorthy, and B. D. Saunders. Parallel algorithms for matrix normal forms. *Linear Algebra and its Appl.*, 136:189–208, 1990.

[L] H.Lüneburg. *On rational form of endomorphisms: a primer to constructive algebra.* Wissenschaftsverlag, Mannheim, Germany, 1987.

[MRR] R. Mines, F. Richman, and W. Ruitenberg. *A course in constructive algebra.* Springer-Verlag, 1988.

[O] P. Ozello. *Calcul exact des formes de Jordan et de Frobenius d'une matrice.* PhD thesis, Université Scientifique et Médicale de Grenoble, Grenoble, France, 1987.

[O1] J.M. de Olazábal. Unified method for determining canonical forms of a matrix. *SIGSAM Bull.*, 33(1):6–20, 1999.

[RV] G.Villard, and J.L.Roch. Parallel computations with algebraic numbers a case of study: Jordan normal form of matrices. In *PARLE '94, Parallel Architectures and Languages Europe*, volume 817 of *LNCS*, pages 701–712. Springer–Verlag, 1994.

[Se] A. Seidenberg. Constructions in algebra. *Trans. Amer. Math. Soc.*, 197:273–313, 1974.

[S] A. Steel. A new algorithm for the computation of canonical forms of matrices over fields. *J. Symb. Comp.*, 24:409–432, 1997.

[St] A. Storjohann. An $O(n^3)$ algorithm for the Frobenius normal form. In *Proceedings of ISSAC'98*, pages 101–104. ACM Press, 1998.

[W] J.H.Wilkinson. *The algebraic eigenvalue problem.* Clarendon Press, Oxford, 1965.