Parallel computations with algebraic numbers a case study: Jordan normal form of matrices

J.L. Roch and G. Villard

Institut IMAG, Laboratoire LMC, 46 Av. F. Viallet,
F38031 Grenoble cedex
Email: Jean-Louis.Roch@imag.fr, Gilles.Villard@imag.fr

Abstract

Proposing a new method for parallel computations on algebraic numbers, we establish that computing the Jordan normal form of matrices over any commutative field F is in $\mathcal{NC}_F$.

1 Introduction

Computing normal forms of matrices is a basic problem in linear algebra. In this paper we are concerned with the parallel computation of the *Jordan normal form* of matrices over a commutative field F. This form has many applications such as computing matrix functions, solving matrix equations, solving differential equations and systems...The reader will refer to [7] to get an insight into the subject.

The Jordan form has been widely studied from a theoretical point of view [7], and sequential polynomial time algorithms are known [19, 16, 10, 11]. From a parallel point of view, few algorithms are reported in the litterature. Fast parallel randomized algorithms may be found in [14, 15]: Kaltofen, Krishnamoorthy and Saunders have shown that computing the Jordan form is in $\mathcal{RNC}$. In [11] Giesbrecht gives a processor efficient randomized algorithm for the same problem. Those results have been improved in [20], partially answering to a question in [9], we have established that random choices may be avoided for the Jordan form and that the problem is in $\mathcal{NC}_F$, unfortunately, since the algorithm uses the squarefree decomposition of polynomials, we were restricted to fields of characteristic zero or finite fields. We refer to [3] for the definitions of the boolean complexity classes $\mathcal{NC}$ and $\mathcal{RNC}$ of problems deterministically and probabilistically solvable by boolean circuits. In analogy with these classes von zur Gathen [9] has defined the classes $\mathcal{NC}_F$ and $\mathcal{RNC}_F$ of problems solvable by arithmetic circuits.

This paper is devoted to a new proof for the fact that computing the Jordan form is in $\mathcal{NC}_F$. The method we propose, based on parallel computations on algebraic numbers, may be useful in many other situations. Concerning the Jordan form, it leads to a much simpler algorithm than in [20] and avoids restrictive assumptions on the ground field since this new algotihm runs over any field F. The same approach is used in [12] from a sequential point if view.

For the purpose of giving the organization of the rest of the paper, let us recall the key idea that we have proposed in [20]. Let A be a matrix of $F^{n\times n}$ having l distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_l$ with respective multiplicities in the characteristic polynomial $m_1, m_2, \ldots, m_l$. The Jordan form may be easily computed from the dimensions of the nullspaces of the successive powers:

$$A - \lambda_i I, (A - \lambda_i I)^2, \ldots, (A - \lambda_i I)^{m_i},\ 1 \leq i \leq l. \tag{1}$$

If F is not algebraically closed, the eigenvalues of A lie in an algebraic extension of F and an appropriate arithmetic has to be used. After some basic reminders on matrix normal forms at section 2, we thus show at section 3, how parallel computations on algebraic numbers may be done in $\mathcal{NC}_F$ in a $D5$ arithmetic manner [4]. $D5$ is a powerful system that has been designed for computing in algebraic closure of fields. Even if only sequential versions of the system are available [5], its working is intrinsically parallel [6], and the results presented in this paper essentially rely on it. As examples, this arithmetic on algebraic numbers is studied for computing the *rank* and the *nullspace* of matrices. Then, at section 4, from the dimensions of the nullspaces of the matrices given by (1), the Jordan normal form will be computed in a natural way in $\mathcal{NC}_F$.

2 Basic concepts

We recall some main definitions. In the following, for a commutative field F, A is a matrix of dimension n whose entries are in F, having l distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_l$.

2.1 Jordan Normal Form

Any matrix A is similar to a unique (up to permutation) block-diagonal matrix J whose diagonal blocks are matrices of the form:

$$J_k(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & 0 & \ldots & 0 \\ 0 & \lambda_i & 1 & & 0 \\ 0 & 0 & \lambda_i & & \vdots \\ \vdots & & & & 1 \\ 0 & \ldots & \ldots & 0 & \lambda_i \end{bmatrix} \in F^{k \times k}$$

where λ_i is an eigenvalue of A; J_k is a $k \times k$ banded matrix, which is called a k-Jordan block associated with λ_i. We refer to [7] for the proof; J is the Jordan normal form of A. Each block $J_k(\lambda_i)$ corresponds to an *elementary divisor* $(\lambda - \lambda_i)^k$ of A. Two similar matrices have the same Jordan normal form.

When the field F is not algebraically closed, the eigenvalues of A lie in an algebraic extension of F. If these are given our algorithm will compute the Jordan form. In general, since we do not know how to factor polynomials fast in parallel [9] we will compute a variant of the Jordan form consisting of blocks corresponding to *generalized eigenvalues*, *i.e.* to eigenvalues belonging to the same factors in a partial factorization of the characteristic polynomial of A [15, 9]. This form is the *symbolic Jordan form*, it gives the structure of J using symbols that take the place of the eigenvalues.

2.2 Symbolic Jordan Normal Form

With any matrix A we may associate its symbolic Jordan form $\tilde{J}$. The structure of $\tilde{J}$ is the same as the structure of J with l distinct symbols $\tilde{\lambda_i}$ taking the place of the eigenvalues. Each symbol $\tilde{\lambda_i}$ is associated with a polynomial $\Lambda_i(\lambda)$ in $F[\lambda]$, with the understanding that Λ_i is a representation of λ_i, *i.e.* $\Lambda_i(\lambda_i) = 0$. The Λ_i are divisors of the characteristic polynomial of A.

Clearly, the symbolic Jordan form is not unique, since different choices are possible for the Λ_i. It coincides with the Jordan form if the eigenvalues are known, *i.e.* if the Λ_i are the linear factors $(\lambda - \lambda_i)$.

But the Λ_i need not be irreducible, otherwise polynomial factorization would be required. In the same way, the previous studies on the subject [15, 20], were assuming the Λ_i to be squarefree and consequently, algorithms were running only for selected fields. This assumption is not necessary to obtain the structure of the Jordan form, that is the degrees of the elementary divisors, and will be removed in the rest of the paper. To distinguish between eigenvalues having Jordan blocks with different structures, we only need to consider symbolic Jordan forms corresponding to Λ_i satisfying:

(i) if there exists a dimension k such that λ_i and λ_j do not have the same number of k-Jordan blocks, then Λ_i and Λ_j are relatively prime.

When the eigenvalues are not known, such a symbolic matrix will be improperly called *Jordan form* of A. The main fact is that it can be computed by polynomial gcd operations only and does not require polynomial factorization [15].

3 Parallel rank and nullspace over algebraic numbers

As said in the introduction and as it will be developped at section 4, computing the rank of matrices over algebraic numbers, is the basic operation of our algorithm for the computation of the Jordan form.

In the following, F is a field, $\Lambda(\lambda)$ is a univariate polynomial over F with $l \leq n$ distinct roots $\lambda_1, \lambda_2, \ldots, \lambda_l$. For any root $\tilde{\lambda}$ of $\Lambda(\lambda)$, we denote by $F(\tilde{\lambda})$ the simple algebraic extension of F by $\tilde{\lambda}$. As usually, the elements of $F(\tilde{\lambda})$ will be represented as polynomials in $F[\lambda]/(\Lambda(\lambda))$. This is common way algebraic extensions appear in computer algebra [6].

Now, let A be a square matrix of dimension n over $F(\tilde{\lambda})$, we assume that the entries of A are polynomial expressions of degree n in $F[\lambda]$ taken modulo $\Lambda(\lambda)$. If A is not square, it is extended by zero rows or columns. Clearly, because the rank of A depends on the chosen root $\tilde{\lambda}$ of $\Lambda(\lambda)$, computing the rank or a basis of the nullspace of A leads to an "automatic discussion". In a more formal way, we will adress the two following problems:

$\mathcal{RANK}_{F(\tilde{\lambda})}$: Computes polynomials $R_r(\lambda)$, $0 \leq r \leq n$, over F, such that:

- The rank of A evaluated at any root of R_i is equal to i.
- $\mathrm{Gcd}(R_r, R_{r'}) = 1$ if $r \neq r'$, and $\prod_{i=1}^{l}(\lambda - \lambda_i)$ divides $\prod_{r=0}^{n} R_r(\lambda)$.

$\mathcal{NULLSPACE}_{F(\tilde{\lambda})}$: Computes the above polynomials R_r, $0 \leq r \leq n$, and computes non singular squares matrices N_r, $0 \leq r \leq n$, of dimension n over $F(\tilde{\lambda})$ such that:

- The first r columns of the matrix AN_r evaluated at any root of R_r are linearly independent and its last $n - r$ columns are zero.

In sequential, those computations may be achieved using a system like $D5$ [4, 5]. For instance, with $D5$ the rank function would return:

" Either λ is a root of R_0 and the rank is 0,
...or λ is a root of R_r and the rank is r, ...
or λ is a root of R_n and the rank is n."

Notice that, the splitting of the initial polynomial Λ into the polynomials R_i does not require polynomial factorization but only polynomial gcd computations.

We begin at section 3.1 by studying the two basic polynomial operations needed for this splitting. Then section 3.2 will give some reminders about rank and nullspace parallel computations over a field, especially we will see that computing the rank and a basis of the

nullspace over $F(\tilde{\lambda})$ are in $\mathcal{NC}^2_{F(\tilde{\lambda})}$. Finally, our purpose at sections 3.3 and 3.4 will be to extend those results and to show that $\mathcal{RANK}_{F(\tilde{\lambda})}$ and $\mathcal{NULLSPACE}_{F(\tilde{\lambda})}$ are in $\mathcal{NC}^2_F$.

3.1 Preliminary results

In the lemmas below we introduce the two standard basic polynomial operations that will be needed for the splitting of the polynomial $A(\lambda)$ during the discussion to generate the different cases. Those operations are the computation of the greatest divisor of a polynomial relatively prime to another, and the multiple multiplicity free decomposition of a polynomial, they are important tools when the ground field does not allow to compute the squarefree decomposition.

Lemma 1. Given two polynomials $P(x)$ and $Q(x)$ of degree n in $F[x]$, to compute $r(P,Q)$, the greatest divisor of P relatively prime to Q, is in $\mathcal{NC}^2_F$.

Proof. In sequential $r(P,Q)$ is usually obtained by repeated gcd computations [16,6]. In fact this can be done with only one gcd computation: if $G = \gcd(P, Q^n)$ then $r(P,Q) = P/G$. Using the algorithm in [2] for the gcd, the computation is in $\mathcal{NC}^2_F$. □

Lemma 2. Given a polynomial $P(x)$ of degree n in $F[x]$, the multiple multiplicity free decomposition (up to a constant) of P consists of n polynomials $P_1, P_2, \ldots, P_n$ such that for all i, the roots of multiplicity i in P are roots of multiplicity i in P_i (in a splitting field), and $\prod_{i=1}^n P_i = cP$, $c \in F$. To compute the decomposition is in $\mathcal{NC}^2_F$.

Proof. For any root λ of $P(x)$, the multiplicity of λ in $P(x)$ is the valuation of $P(x+\lambda)$. Now consider λ as an indeterminate and let

$$P(x+\lambda) = \sum_{i=0}^{n} a_i(\lambda) x^i.$$

For any fixed i, the roots of P which are roots of $a_0(\lambda), a_1(\lambda), \ldots, a_{i-1}(\lambda)$ but not of $a_i(\lambda)$ are the roots of multiplicity i in P. Viewing P as a polynomial in λ instead of x, define

$$q_i(\lambda) = \gcd\left(P(\lambda), a_0(\lambda), a_1(\lambda), \ldots, a_i(\lambda)\right), \ 0 \le i \le n.$$

By construction, the roots of q_i are the roots of P whose multiplicity in P is strictly greater than i. For $1 \le i \le n$, let $r_i = r(q_{i-1}, q_i)$ be the greatest divisor of q_{i-1} relatively prime to q_i: the roots of r_i are the roots of P whose multiplicity in P is precisely i. In other words, if we denote by $\lambda_1^{(i)}, \ldots, \lambda_{k_i}^{(i)}$ the roots of P whose multiplicity is i, we may write for a certain power d_i, $d_i < i$, and a constant c_i in F:

$$r_i(\lambda) = c_i \left((\lambda - \lambda_1^{(i)}) \ldots (\lambda - \lambda_{k_i}^{(i)})\right)^{d_i}.$$

By definition we have

$$P(\lambda) = T_i(\lambda) \left((\lambda - \lambda_1^{(i)}) \ldots (\lambda - \lambda_{k_i}^{(i)})\right)^{i},$$

we compute

$$r(P, r_i) = c_i' T_i(\lambda), \quad c_i' \in F,$$

then we have the target polynomials: $P_i = P/r(P, r_i)$, $1 \le i \le n$. Using the algorithm for the gcd of many polynomials in [8] the q_i are computed in $\mathcal{NC}^2_F$, and applying lemma 1 to compute the r_i and the $r(P, r_i)$ we conclude the proof: the P_i are computed in $\mathcal{NC}^2_F$. □

3.2 Parallel rank and nullspace over a field

This section is intended to remind fundamentals results on rank and nullspace basis computations.

• **Computing the rank.** We briefly describe the algorithm of Mulmuley [18] for the rank or the dimension of the nullspace of a matrix A over an arbitrary field. A is assumed to be square: if not, A is extended by zero rows or columns. Consider the symmetric matrix

$$\tilde{A} = \begin{bmatrix} 0 & A \\ A^t & 0 \end{bmatrix}.$$

Let Z be a diagonal matrix in an indeterminate z such that $Z_{ii} = z^{i-1}$, $1 \leq i \leq 2n$. The highest degree d such that x^d divides the characteristic polynomial

$$\tilde{\chi}_A(x) = \det(xI - Z\tilde{A}) \tag{2}$$

is twice the dimension of the nullspace of A. We now point out an elementary property that will be useful in the following of the paper. Since for any matrices A_1, A_2, A_3 and A_4 of equal dimensions, we have the following identity on determinants:

$$\begin{vmatrix} A_1 & A_2 \\ A_3 & A_4 \end{vmatrix} = \begin{vmatrix} -A_1 & A_2 \\ A_3 & -A_4 \end{vmatrix},$$

we deduce that

$$\tilde{\chi}_A(x) = \det(xI - Z\tilde{A}) = \begin{vmatrix} xI & -Z_1 A \\ -Z_2 A^t & xI \end{vmatrix} = \begin{vmatrix} -xI & -Z_1 A \\ -Z_2 A^t & -xI \end{vmatrix} = \tilde{\chi}_A(-x).$$

In other words, $\tilde{\chi}_A(x)$ is of degree $2n$ and is an even function of x. Using the algorithm in [1] for the computation of the characteristic polynomial, the algorithm of Mulmuley computes the rank in time $O(\log^2 n)$ using a polynomial number of processors.

• **Computing a nullspace basis.** We shortly describe the algorithm in [2]. The first step consists in computing a maximal linearly independent subset S of the set $(C_1, C_2, \ldots, C_n)$ of the columns of the

$$rank\ (C_1, C_2, \ldots, C_{k-1}) < rank\ (C_1, C_2, \ldots, C_k).$$

A maximal non singular minor M of A is then computed by applying the same process to the rows of the selected columns. Up to permutations, let M be the $r \times r$ principal minor of A. Let y_k be the first r rows of the k-th column of A, $r+1 \leq k \leq n$, and consider the solutions x_k of the systems $Mx_k = y_k$, $r+1 \leq k \leq n$. A basis of the nullspace of A then consists of the vectors ${}^t(x_k, 0, 0, \ldots, -1, 0, \ldots, 0)$, $r+1 \leq k \leq n$, where -1 is in the k-th position. In addition we obtain a non singular matrix N such that the first r columns of the matrix AN are linearly independent, and its last $n-r$ columns are zero. The first r columns of N corresponds to the permutations used to obtain the first r columns of A linearly independent, its last $n-r$ columns are the computed basis of the nullspace. Using the previous algorithm for the rank, and the algorithm in [2] for matrix inversion, the matrix N is computed in time $O(\log^2 n)$ using a polynomial number of processors.

In conclusion, over $F[\lambda]$, rank and nullspace can be computed in $O(\log^2 n)$ operations in $F[\lambda]$. We show in the two next sections that they can also be computed in $O(\log^2 n)$ operations in F.

3.3 Parallel rank over algebraic numbers

We are given a square matrix A of dimension n over $F(\tilde{\lambda})$ whose entries are polynomials of degree n in λ modulo $\Lambda(\lambda)$: λ is a symbol representing any root of $\Lambda(\lambda)$. $\Lambda(\lambda)$ is a univariate polynomial over F with $l \leq n$ distinct roots $\lambda_1, \lambda_2, \ldots, \lambda_l$. We show that a "parallel discussion" on the rank of A may be achieved in $O(\log^2 n)$ operations in F.

Proposition 3. *$\mathcal{RANK}_{F(\tilde{\lambda})}$ is in $\mathcal{NC}^2_F$.*

Proof. For $1 \leq i \leq l$, let $\tilde{\chi}_i(x)$ be the characteristic polynomial (given by identity (2)) introduced by Mulmuley to compute the rank of A evaluated at $\lambda = \lambda_i$. In the same way, let $\tilde{\chi}(x, \lambda)$ be the corresponding characteristic polynomial associated with A viewing λ as an indeterminate. These polynomials are of degree $2n$ in x and are even functions, so we can write:

$$\begin{aligned} \tilde{\chi}_i(x) &= \textstyle\sum_{j=0}^{n} a_{i,j}\, x^{2j},\ 1 \leq i \leq l, \\ \tilde{\chi}(x, \lambda) &= \textstyle\sum_{j=0}^{n} a_j(\lambda)\, x^{2j}. \end{aligned}$$

Furthermore, since these polynomials are determinants, we know by homomorphism that:

$$\begin{aligned} \tilde{\chi}(x, \lambda_i) &= \tilde{\chi}_i(x),\ 1 \leq i \leq l, \\ a_j(\lambda_i) &= a_{i,j},\ 0 \leq j \leq n,\ 1 \leq i \leq l. \end{aligned}$$

From Mulmuley, the highest degree d such that x^d divides $\tilde{\chi}_i(x)$ is twice d_i, the dimension of the kernel of A evaluated at $\lambda = \lambda_i$. In other words, d_i is half the valuation of $\tilde{\chi}_i(x)$. Such a computation has been done previously for the proof of lemma 2: we define

$$\begin{aligned} q_{-1}(\lambda) &= \Lambda(\lambda), \\ q_j(\lambda) &= \gcd\left(\Lambda(\lambda), a_0(\lambda), a_1(\lambda), \ldots, a_j(\lambda)\right),\ 0 \leq j \leq n, \end{aligned}$$

and, for $0 \leq j \leq n$, we compute the greatest divisor of q_{j-1} relatively prime to q_j: $K_j = r(q_{j-1}, q_j)$. The roots of K_j are exactly the λ_i for which the dimension of the kernel of A is j. In other words, the wanted polynomials R_r are:

$$R_r = r(q_{n-r-1}, q_{n-r}), 0 \leq r \leq n.$$

The roots of R_r are exactly the λ_i for which the rank of A is r.

It remains to establish that the computation can be done in $O(\log^2 n)$ time using a polynomial number of processors. Using the algorithm in [1] the computation of the characteristic polynomial $\tilde{\chi}$ is done in $\mathcal{NC}^2_F$. The computation of the q_j consists in calculating the gcd of Λ and of the polynomials a_j which are the coefficients of the previous polynomial $\tilde{\chi}$. These coefficients are polynomials of degree $O(n^2)$ in z and in λ. The wanted gcd is a univariate polynomial in λ so we may use the algorithm in [8] for the gcd of many polynomials, since the polynomials Λ and a_j are in $F[z, \lambda]$, the computations are done over $F(z)$ the fraction field of $F[z]$: in [8] the gcd is computed using rank and linear system solutions. From [18] for the rank and from [1] for the determinant computations and the systems solutions when entries are rational functions of a variable, the computation of the q_j is in $\mathcal{NC}^2_F$. Finally applying lemma 1 the R_r are obtained in $\mathcal{NC}^2_F$. □

• ***"Parallel $D5$ arithmetic".*** The polynomials R_r may be viewed as generalized algebraic numbers: as the representations of the algebraic numbers λ_i satisfying the property "the rank of A evaluated at $\lambda = \lambda_i$ is r". From this point of view, the above parallel computation of the rank simply consists in running $O(n)$ $D5$ systems simultaneously, each computing "λ such that the rank of A is r", $0 \leq r \leq n$, and using the algorithm of Mulmuley for the rank.

3.4 Parallel nullspace over algebraic numbers

We keep the same notations than above. Let r be the rank of A, the problem is to transform A into a matrix $AN_r = [A', 0]$, such A' is a $n \times r$ matrix of rank r. We are going to adapt the algorithm of section 3.2, a difficulty is to show that the same non singular matrix N_r may be chosen for all the roots of a given R_r. Consider for instance

$$A = \begin{bmatrix} \lambda - 1 & 0 \\ 0 & \lambda - 2 \end{bmatrix} \text{ with } \Lambda(\lambda) = \lambda^2 - 3\lambda + 2.$$

Computing the rank of A yield $R_1(\lambda) = \Lambda(\lambda)$: both for $\lambda = 1$ and for $\lambda = 2$ the rank of A is equal to 1. A maximal linearly independant subset S_r of the columns of A is $\{C_2\}$, the second column of A for $\lambda = 1$, and is $\{C_1\}$, the first column for $\lambda = 2$. Consequently we may consider the vector $(\lambda - 1)C_1 + (\lambda - 2)C_2$: it is non zero both for $\lambda = 1$ and for $\lambda = 2$. And for N_1 one may take:

$$N_1 = \begin{bmatrix} \lambda - 1 & \lambda - 2 \\ \lambda - 2 & \lambda - 1 \end{bmatrix}.$$

The next proposition apply this method in the general case.

Proposition 4. *$NULLSPACE_{F(\tilde{\lambda})}$ is in $\mathcal{NC}^2_F$.*

Proof. Using proposition 3 we compute the rank of A, and the polynomials R_r. For all r, $0 \leq r \leq n$, we simultaneously compute a matrix N_r: we assume r to be fixed.

Following the method presented at section 3.2, we begin by computing a set of r linearly independent columns of A, *i.e.* a non singular matrix M_r such that the first r columns of $A_1 = AM_r$ are linearly independent. The same argument will be then applied on the rows of A_1 to get a matrix L_r so that the principal $r \times r$ minor of $A_2 = L_r A_1$ is non singular. Solving $n - r$ linear systems will finally lead to a matrix N_r such that $AN_r = [A', 0]$.

As said above we firstly focus on the computation of a maximal set of linearly independent columns of A. Let C_k denote the k-th column of A, and, for all k, $1 \leq k \leq n$, and apply proposition 3 to the matrices $[C_1, C_2, \ldots, C_k]$ with $\Lambda(\lambda) = R_r(\lambda)$. This yields polynomials $R_j^{(k)}$, for $1 \leq k \leq n$ and $0 \leq j \leq r$, with the understanding that for all the roots of $R_j^{(k)}$, the rank of $[C_1, C_2, \ldots, C_k]$ is equal to j. In addition, we take $R_0^{(0)}(\lambda) = R_r(\lambda)$, and $R_j^{(0)}(\lambda) = 1$, $1 \leq j \leq r$. The generalization of the previous example leads to consider polynomials $Q_j^{(k)}$ whose roots satisfy

$$\begin{cases} rank([C_1, C_2, \ldots, C_k]) & = j \\ rank([C_1, C_2, \ldots, C_{k-1}]) & = j - 1. \end{cases} \tag{3}$$

Those polynomials can be computed simultaneously for all j, $1 \leq j \leq r$, and for all k, $1 \leq k \leq n$. From the definition of the $R_j^{(k)}$ we have

$$Q_j^{(k)} = \gcd(R_j^{(k)}, R_{j-1}^{(k-1)}),\ 1 \leq j \leq r,\ 1 \leq k \leq n.$$

In other words, $Q_j^{(k)}(\lambda) = 0$, represents a generalized algebraic number for which the column k may be taken to be the j-th column of a maximal linearly independent set of the columns of A: the column k may be taken to be C_j^*, the j-th column of A_1. Conversely, for

$r(R_r(\lambda), Q_j^{(k)}(\lambda)) = 0$ (greatest divisor of R_r relatively prime to $Q_j^{(k)}$), the column k do not have to be taken, consequently we may construct C_j^* as follows:

$$C_j^* = \sum_{k=1}^{n} r\left(R_r(\lambda), Q_j^{(k)}(\lambda)\right) C_k,\ 1 \le j \le r.$$

By duality, let $P_j^{(k)}$, be the polynomials whose roots satisfy:

$$\begin{cases} \dim\ker\left([C_1, C_2, \ldots, C_k]\right) &= j \\ \dim\ker\left([C_1, C_2, \ldots, C_{k-1}]\right) &= j-1, \end{cases} \tag{4}$$

they are computed simultaneously for all j, $1 \le j \le n-r$, and for all k, $1 \le k \le n$ using polynomials $K_j^{(k)}$: for all the roots of $K_j^{(k)}$, the dimension of the kernel of $[C_1, C_2, \ldots, C_k]$ is equal to j (see polynomials K_j computed in the proof of proposition 3). If $\bar{C}_{r+j}$ denote the last $n-r$ columns of A_1, we may take,

$$\bar{C}_{r+j} = \sum_{k=1}^{n} r\left(R_r(\lambda), P_j^{(k)}(\lambda)\right) C_k,\ 1 \le j \le n-r.$$

And M_r is given by:

$$\begin{cases} (M_r)_{k,j} &= r\left(R_r, Q_j^{(k)}\right),\ 1 \le j \le r,\ 1 \le k \le n, \\ (M_r)_{k,r+j} &= r\left(R_r, P_j^{(k)}\right),\ 1 \le j \le n-r,\ 1 \le k \le n. \end{cases}$$

For any root of $R_r(\lambda)$, M_r is non singular since it is nearly a permutation matrix : with constant entries instead of 1's. Applying the same process on the rows of $A_1 = AM_r$ we obtain a matrix $A_2 = L_r A M_r$ whose principal minor is non singular for any root of $R_r(\lambda)$. As explained at section 3.2, using matrix inversion, we now compute a basis $X_1(\lambda), X_2(\lambda), \ldots, X_{n-r}(\lambda)$, of the kernel of A, and:

$$AM_r \times \left[\underbrace{\begin{bmatrix} Id \\ 0 \end{bmatrix}}_{r}, X_1, X_2, \ldots, X_{n-r}\right] = \left[C_1^*, C_2^*, \ldots, C_r^*, \underbrace{0, 0, \ldots, 0}_{n-r}\right] = [A', 0],$$

so we can take:

$$N_r = M_r \times \left[\begin{bmatrix} Id \\ 0 \end{bmatrix}, X_1, X_2, \ldots, X_{n-r}\right].$$

To conclude, we have to show that N_r can be computed in $\mathcal{NC}_F^2$. From proposition 3 the polynomials $R_j^{(k)}$ and $K_j^{(k)}$ are computed simultaneously for all j and all k, using the algorithm in [2] for the gcd computations and lemma 1 the matrix M_r is computed in $\mathcal{NC}_F^2$. Finally, using the algorithm in [1] for the inversion of a matrix in one indeterminate, the vectors $X_1(\lambda), X_2(\lambda), \ldots, X_{n-r}$ and the matrix N_r are computed in $\mathcal{NC}_F^2$. □

• ***"Parallel $D5$ arithmetic"***. From a $D5$ point of view, the polynomials $Q_j^{(k)}$ and $P_j^{(k)}$ are generalized algebraic numbers satisfying relations (3) and (4). They are computed in parallel by running $O(n^2)$ $D5$ systems simultaneously.

In the next section, using $\mathcal{RANK}_{F(\lambda)}$ as a key routine, we prove in a more natural way than in [20], that computing the Jordan normal form is in $\mathcal{NC}_F^2$.

4 Parallel Jordan normal form

We now give a fast parallel deterministic algorithm for computing the Jordan form over any field F. We begin with a standard lemma giving a method of computing the number of Jordan blocks. Its proof is omitted, the reader may refer to [17]. From this lemma we will then develop the algorithm computing also the representations Λ_i of the eigenvalues defined at section 2.2. Computations are over algebraic numbers, using the results presented above, they will be done in $\mathcal{NC}_F$.

For F a field, we consider a matrix A of dimension n whose entries are in F, having l distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_l$. For any eigenvalue λ_i of A, let us consider the kernels of the successive powers of $A - \lambda_i I$. It is widely known that

$$\ker(A - \lambda_i I) \subset \ker(A - \lambda_i I)^2 \subset \ldots \subset \ker(A - \lambda_i I)^{\mu_i} = \ldots = \ker(A - \lambda_i I)^{m_i},$$

where μ_i and m_i are respectively the multiplicities of λ_i in the minimal polynomial μ_A and in the characteristic polynomial χ_A of A.

Lemma 5. If $d_i^{(k)}$ is the dimension of the kernel of $(A - \lambda_i I)^k$, then the number $\delta_i^{(k)}$ of blocks $J_k(\lambda_i)$ of dimension k associated with λ_i in the Jordan normal form of A is given by:

$$\delta_i^{(k)} = 2d_i^{(k)} - d_i^{(k-1)} - d_i^{(k+1)}, \ 1 \leq i \leq l \text{ and } 1 \leq k \leq n.$$

The problem consequently reduces to rank, or equivalently, to nullity (dimension of the kernel) computations [20]. In the following we adress two slightly different problems: to compute the structure of the Jordan normal form, this will be in $\mathcal{NC}_F^2$, and to compute a symbolic normal form with representations Λ_i that distinguish between eigenvalues associated with different block structures, this will be in $\mathcal{NC}_F^3$.

Theorem 6. *For F a field and $A \in F^{n \times n}$, to compute the structure of the Jordan form of A, that is the degrees of the elementary divisors of A, is in $\mathcal{NC}_F^2$.*

Proof. For any fixed d and k, $1 \leq d, k \leq n$, we compute polynomials $J_d^{(k)}(\lambda)$ whose roots λ_i are the eigenvalues of A which are associated with exactly d Jordan blocks of dimension k: which are associated with d elementary divisors $(x - \lambda_i)^k$.

To compute the $J_d^{(k)}(\lambda)$, we first use proposition 3 and calculate polynomials $K_j^{(k)}(\lambda)$ whose roots are the eigenvalues λ_i such that $\dim(\ker(A - \lambda_i I)^k) = j$. From lemma 5, an eigenvalue λ_i is associated with exactly d Jordan blocks of dimension k if and only if there exist d_1, d_2 and d_3 such that $2d_1 - d_2 - d_3 = d$ and such that λ_i is a root of $K_{d_1}^{(k)}$, $K_{d_2}^{(k-1)}$ and $K_{d_3}^{(k+1)}$. Consequently,

$$J_d^{(k)} = \prod_{\substack{0 \leq d_1, d_2, d_3 \leq n \\ 2d_1 - d_2 - d_3 = d}} \gcd\left(K_{d_1}^{(k)}, K_{d_2}^{(k-1)}, K_{d_3}^{(k+1)}\right), \quad 1 \leq d, k \leq n.$$

Using $O(n^3)$ parallel processes, the above gcds can be simultaneously computed for all d and (d_1, d_2, d_3) such that $2d_1 - d_2 - d_3 = d$. Then the $J_d^{(k)}$ are obtained by multiplications.

From proposition 3 the $K_j^{(k)}$ are computed in $\mathcal{NC}_F^2$, and we use the algorithm in [2] for the gcd of polynomials to compute the $J_d^{(k)}$ in $\mathcal{NC}_F^2$. □

• *"Parallel $D5$ arithmetic"*. The polynomials $J_d^{(k)}$ may be viewed as generalized algebraic numbers satisfying:

$$\begin{cases} \dim\left(\ker(A-\lambda I)^k\right) & = d_1 \\ \dim\left(\ker(A-\lambda I)^{k-1}\right) & = d_2 \\ \dim\left(\ker(A-\lambda I)^{k+1}\right) & = d_3,\ 2d_1 - d_2 - d_3 = d. \end{cases}$$

They are computed in parallel by running $O(n^3)$ $D5$ systems simultaneously.

Now the problem is to refine the polynomials $J_d^{(k)}$ to obtain target representations Λ_i of the eigenvalues, as introduced at section 2.2 and satisfying property:

(i) if there exists a dimension k such that the two eigenvalues λ_i and λ_j do not have the same number of k-Jordan blocks then their representations Λ_i and Λ_j are relatively prime.

When the field F is such that squafree decomposition of polynomials can be computed in $F[\lambda]$, one may use a result in [15]: computing the representations Λ_i consists in computing the *standard squarefree relatively prime basis of polynomials* $\{J_d^{(k)}\}_{1\leq k,d\leq n}$, this can be done in $\mathcal{NC}_F^3$ [15]. In the general case, we have to use the more general definition of a *gcd-free basis* [13]: such a basis exists over any field. A gcd-free basis of polynomials $\{P_1, P_2, \ldots, P_n\}$ in $F[\lambda]$ consists of non unit polynomials $\{I_1, I_2, \ldots, I_m\}$ in $F[\lambda]$ that satisfy:

- I_i and I_j are relatively prime for $1 \leq i < j \leq m$.
- For all $1 \leq j \leq n$ there exist positive integers $e_{i,j}$ such that $P_j = \prod_{i=1}^{m} I_i^{e_{i,j}}$.

Next theorem consists in computing a gcd-free basis of $\{J_d^{(k)}\}_{k,d}$.

Theorem 7. *For F a field, to compute a symbolic Jordan form of $A \in F^{n\times n}$ is in $\mathcal{NC}_F^3$.*

Proof. By construction, a gcd-free basis of $\{J_d^{(k)}\}_{k,d}$ will yield desired representations $\Lambda_i(\lambda)$ of the eigenvalues (without repetition). Since they are divisors of the characteristic polynomial of A, the $J_d^{(k)}$ are of degrees less than n.

We now give an algorithm to compute rapidly in parallel a gcd-free basis of a set $\{P_1, P_2, \ldots, P_n\}$ of polynomials of degrees less than n. Without loss of generality, by lemma 2 we may assume the P_j are multiple multiplicity free (all the roots of P_j have the same multiplicity in a splitting field) for all $1 \leq j \leq n$. The algorithm is a generalization of the algorithm in [13] in the case where squafree decomposition cannot be computed. We proceed in three steps: we compute a gcd-free basis of two polynomials, then merge two gcd-free basis and finally compute a basis for the whole given set.

Let $P(\lambda)$ and $Q(\lambda)$ be two multiple multiplicity free polynomials, and $G(\lambda)$ be their gcd. Let g be the greatest squarefree divisor of G in a splitting field. We may write for some $P', Q' \in F[\lambda]$ and p, q positive integers:

$$P = P'g^p,\ Q = Q'g^q.$$

These decompositions are computed as follow:

$$G_p = g^p = P/r(P,G),\ G_q = g^q = Q/r(Q,G).$$

If $\gamma = \gcd(p,q)$ and α and β are such that $\alpha p + \beta q = \gamma$, let $b(P,Q) = G_p^\alpha G_q^\beta = g^\gamma \in F[\lambda]$. A gcd-free basis of $\{P,Q\}$ consists of the polynomials $\{b(P,Q), r(P,G), r(Q,G)\}$. Using

lemma 1, $r(P,G), r(Q,G)$ then G_p and G_q are computed in $\mathcal{NC}^2_F$. Applying lemma 2, we calculate p and q, since polynomials we consider are of degrees bounded by n, α, β and γ are then computed in time $O(\log^2 n)$. From there, $b(P,Q)$ is computed in $\mathcal{NC}^2_F$.

Let $\{P_i\}_i$ and $\{Q_j\}_j$ be two gcd-free basis. The above scheme is easily extended to compute a merged basis. The entries of a merged basis are $B_{i,j} = b(P_i, Q_j)$, $P_i^* = r(P_i, \prod_j B_{i,j})$ and $Q_j^* = r(Q_j, \prod_i B_{i,j})$. Units entries may be discarded. The $B_{i,j}$ are computed simultaneously in $\mathcal{NC}^2_F$, then the P_i^* and Q_j^* are obtained in $\mathcal{NC}^2_F$ by multiplications and lemma 1.

To conclude the proof it remains to construct a gcd-free basis of a given set $\{P_1, P_2, \ldots, P_n\}$ (by lemma 2 the polynomials are assumed to be multiple multiplicity free). Each P_i is a gcd-free basis of $\{P_i\}$. The n bases P_i may be merged by pairs, repeating this process iteratively: the target basis is computed in $\log(n)$ steps. The whole computation is done in $\mathcal{NC}^3_F$. □

If the eigenvalues of A are known, in other words, if the characteristic polynomial of A can be completely factorized, the computation of the Jordan form is clearly $\mathcal{NC}_F$-reducible to linear system solutions and can be computed in $\mathcal{NC}^2_F$. The next corollary follows immediately, its proof is omitted. Given a field F, we refer to [9] for the definition of the complexity class $\mathcal{DET}_F$ of the problems easier than the determinant.

Corollary 8. For F a field, A in $F^{n\times n}$ and given the l distinct eigenvalues $\lambda_i \in F$ of A, to compute the Jordan normal form of A is complete for $\mathcal{DET}_F$.

If the factorization of the characteristic polynomial of A is not known, the problem consists in obtaining "good" representations of the eigenvalues. In this case, as seen above, the complexity is dominated by the computation of a gcd-free basis.

5 Conclusion

To deal with algebraic numbers is a central problem in computer algebra. We have shown how this can be done rapidly in parallel in a $D5$ philosophy, and how some crucial questions of linear algebra can be solved. An interesting problem will be to develop and generalize our model to still simplify its use. Those results have various applications especially in linear algebra [7, 10, 21].

Acknowledgements

Many thanks go to Dominique Duval for useful discussions; in particular she suggested the use of the operations introduced in lemma 1 and lemma 2 and of the successive kernels.

References

1. A. Borodin, S.A. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space bounded probalistic machines. *Information and Control*, 58:113–136, 1983.
2. A. Borodin, J. von zur Gathen, and J. Hopcroft. Fast parallel matrix and gcd computations. *Information and Control*, 52:241–256, 1982.
3. S.A. Cook. A taxonomy of problems with fast parallel algorithms. *Inf. Control*, 64:2–22, 1985.

4. J. Della Dora, C. Dicrescenzo, and D. Duval. About a new method for computing in algebraic number fields. In *Proc. EUROCAL'85*, LNCS 204, Springer Verlag, pages 289–290, 1985.
5. C. Dicrescenzo and D. Duval. Algebraic extensions and algebraic closure in Scratchpad II. In *Proc. ISSAC'88*, LNCS 358, Springer Verlag, pages 440–446, 1988.
6. D. Duval. *Diverses questions relatives au calcul formel avec des nombres algébriques*. Thèse de Doctorat d'Etat, Université de Grenoble, France, 1987.
7. F.R. Gantmacher. *Théorie des matrices*. Dunod, Paris, France, 1966.
8. J. von zur Gathen. Parallel algorithms for algebraic problems. *SIAM J. Comp.*, 13:802–824, 1984.
9. J. von zur Gathen. Parallel arithmetic computations: a survey. In *Proc. 12th Int. Symp. Math. Found. Comput. Sci., Bratislava*, pages 93–112. LNCS 233, Springer Verlag, 1986.
10. M. Giesbrecht. *Nearly optimal algorithms for canonical matrix forms* . PhD thesis, Department of Computer Science, University of Toronto, 1993.
11. M. Giesbrecht. Nearly optimal algorithms for canonical matrix forms. *SIAM Journal on Computing*, 1994. To appear.
12. T. Gómez-Díaz. *Quelques applications de l'évaluation dynamique* . PhD thesis, Université de Limoges, France, 1994.
13. E Kaltofen. Sparse Hensel lifting. In *Proc. EUROCAL'85, vol. 2*, pages 4–17. LNCS 204, Springer Verlag, 1985.
14. E. Kaltofen, M.S. Krishnamoorthy, and B.D. Saunders. Fast parallel computation of Hermite and Smith forms of polynomials matrices. *SIAM J. Alg. Disc. Meth.*, 8 4, pp 683-690, 1987.
15. E. Kaltofen, M.S. Krishnamoorthy, and B.D. Saunders. Parallel algorithms for matrix normal forms. *Linear Algebra and its Applications*, 136:189–208, 1990.
16. H. Lüneburg. *On rational form of endomorphims : a primer to constructive algebra*. Wissenschaftsverlag, Mannheim, 1987.
17. C.C. MacDuffee. *The theory of matrices*. Chelsea, New-York, 1956.
18. K. Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7(1):101–104, 1987.
19. P. Ozello. *Calcul exact des formes de Jordan et de Frobenius d'une matrice*. PhD thesis, Université Scientifique et Médicale de Grenoble, France, 1987.
20. J.L. Roch and G. Villard. Fast parallel computation of the Jordan normal form of matrices. *Parallel Processing Letters*, 1994. To appear.
21. G. Villard. Fast parallel computation of the Smith normal form of polynomial matrices. In *International Symposium on Symbolic and Algebraic Computation, Oxford, UK*. ACM Press, July 1994.