The Complexity of the Minimal Polynomial ⋆

Thanh Minh Hoang and Thomas Thierauf

Abt. Theoretische Informatik
Universität Ulm
89069 Ulm, Germany
{hoang,thierauf}@informatik.uni-ulm.de

Abstract. We investigate the computational complexity of the minimal polynomial of an integer matrix.
We show that the computation of the minimal polynomial is in $\mathbf{AC}^0(\mathbf{GapL})$, the $\mathbf{AC}^0$-closure of the logspace counting class $\mathbf{GapL}$, which is contained in $\mathbf{NC}^2$. Our main result is that the problem is hard for $\mathbf{GapL}$ (under $\mathbf{AC}^0$ many-one reductions). The result extends to the verification of all invariant factors of an integer matrix.
Furthermore, we consider the complexity to check whether an integer matrix is diagonalizable. We show that this problem lies in $\mathbf{AC}^0(\mathbf{GapL})$ and is hard for $\mathbf{AC}^0(\mathbf{C_=L})$ (under $\mathbf{AC}^0$ many-one reductions).

1 Introduction

The motivation for our work is twofold: 1) we want to understand the computational complexity of some classical problems in linear algebra, 2) by locating such problems in small space complexity classes we want to clarify the inclusion relationship of such classes.

The *minimal polynomial* of a matrix plays an important role in the theory of matrices. Algorithms to compute the minimal polynomial of a matrix have been studied for a long time. The best known deterministic algorithm to compute the minimal polynomial of an $n \times n$ matrix makes $O(n^3)$ field operations [Sto98]. The Smith normal form of a polynomial matrix can be computed by a randomized $\mathbf{NC}^2$-circuit, i.e., in $\mathbf{RNC}^2$. Therefore the rational canonical form of a matrix and the minimal polynomial of a matrix can be computed in $\mathbf{RNC}^2$ as well (see [KS87,vzGG99] for details). In the case of integer matrices there are even $\mathbf{NC}^2$-algorithms [Vil97].

We take a different approach to compute the minimal polynomial of an integer matrix: we show that the problem can be reduced to matrix powering and solving systems of linear equations. Therefore it is in the class $\mathbf{AC}^0(\mathbf{GapL})$, a subclass of $\mathbf{NC}^2$. Our main result is with respect to the hardness of the problem: we show that the computation of the determinant of a matrix can be reduced to the computation of the minimal polynomial of a matrix. Therefore the problem is hard for $\mathbf{GapL}$.

⋆ This work was supported by the Deutsche Forschungsgemeinschaft

The minimal polynomial is the first polynomial of the system of all *invariant factors* of a matrix. This system completely determines the structure of the matrix. Its computation is known to be in $\mathbf{NC}^2$ [Vil97] for integer matrices. We extend our results and techniques to the *verification* of all the invariant factors of a given integer matrix: it is in $\mathbf{AC}^0(\mathbf{C_=L})$ and hard for $\mathbf{C_=L}$.

Using the results about the minimal polynomial, we can classify some more classical problems in linear algebra: a matrix is diagonalizable if it is similar to a diagonal matrix. Testing similarity of two matrices is known to be in $\mathbf{AC}^0(\mathbf{C_=L})$ [HT00]. We show that the problem to decide whether a given integer matrix is diagonalizable is in $\mathbf{AC}^0(\mathbf{GapL})$ and hard for $\mathbf{AC}^0(\mathbf{C_=L})$.

To obtain the latter result, we have to solve a problem that is interesting in its own: decide, whether all eigenvalues of a given integer matrix are pairwise different. This can be done in $\mathbf{AC}^0(\mathbf{C_=L})$.

2 Preliminaries

For a nondeterministic logspace bounded Turing machine M, we denote the number of accepting paths on input x by $acc_M(x)$, and by $rej_M(x)$ the number of rejecting paths. The difference of these two numbers is $gap_M(x) = acc_M(x) - rej_M(x)$.

For the counting classes, we have $\#\mathbf{L}$, the class of functions $acc_M(x)$ for some nondeterministic logspace bounded Turing machine M, and $\mathbf{GapL}$ based analogously on functions gap_M. Based on counting, we consider the language class $\mathbf{C_=L}$: a set A is in $\mathbf{C_=L}$, if there exists a $f \in \mathbf{GapL}$ such that for all x: $x \in A \iff f(x) = 0$.

For sets A and B, A is $\mathbf{AC}^0$*-reducible to* B, if there is a logspace uniform circuit family of polynomial size and constant depth that computes A with unbounded fan-in and-, or-gates and oracle gates for B. In particular, we consider the classes $\mathbf{AC}^0(\mathbf{C_=L})$ and $\mathbf{AC}^0(\mathbf{GapL})$ of sets that are $\mathbf{AC}^0$-reducible to a set in $\mathbf{C_=L}$, respectively a function in $\mathbf{GapL}$.

A is $\mathbf{AC}^0$ *many-one reducible to* B, in symbols: $A \leq_m^{AC^0} B$, if there is a function $f \in \mathbf{AC}^0$ such that for all x we have $x \in A \iff f(x) \in B$. All reductions in this paper are $\mathbf{AC}^0$ many-one reductions.

Let $A \in \mathbf{F^{n\times n}}$ be a matrix over the field $\mathbf{F}$. A nonzero polynomial $p(x)$ over $\mathbf{F}$ is called an *annihilating polynomial of* A if $p(A) = \mathbf{0}$. The Cayley-Hamilton Theorem states that the characteristic polynomial $\chi_A(x)$ of A is an annihilating polynomial. The characteristic polynomial is a *monic polynomial*: its highest coefficient is one. The *minimal polynomial of* A, denoted $\mu_A(x)$, is the unique monic annihilating polynomial of A with minimal degree.

Let polynomial $d_k(x)$ be the greatest common divisor of all sub-determinants of $(xI - A)$ of order k. For example $d_n(x) = \chi_A(x)$. We see that d_k divides d_{k+1} for each index $0 \leq k \leq n$. Define $d_0(x) \equiv 1$. The *invariant factors of* $(xI - A)$ (or A, for short) are defined as the following (monic) polynomials:

$$i_1(x) = \frac{d_n(x)}{d_{n-1}(x)}, \quad i_2(x) = \frac{d_{n-1}(x)}{d_{n-2}(x)}, \quad \ldots, \quad i_n(x) = \frac{d_1(x)}{d_0(x)}.$$

The characteristic polynomial of A is the product of all the invariant factors: $\chi_A(x) = i_1(x) \cdots i_n(x)$. The $n \times n$ polynomial matrix that has the invariant factors of A as its diagonal entries (starting with $i_n(x)$) and zero elsewhere is the *Smith normal form of* $xI - A$, denoted by $diag\{i_n(x), \ldots, i_1(x)\}$.

We decompose the invariant factors into irreducible divisors over the given number field $\mathbf{F}$:

$$\begin{array}{rcl} i_1(x) & = & [e_1(x)]^{c_1}[e_2(x)]^{c_2} \cdots [e_s(x)]^{c_s}, \\ i_2(x) & = & [e_1(x)]^{d_1}[e_2(x)]^{d_2} \cdots [e_s(x)]^{d_s}, \\ \vdots & & \vdots \\ i_n(x) & = & [e_1(x)]^{l_1}[e_2(x)]^{l_2} \cdots [e_s(x)]^{l_s}, \\ \multicolumn{3}{l}{(0 \le l_k \le \ldots \le d_k \le c_k;\ k = 1, 2, \ldots, s).} \end{array}$$

The irreducible divisors $e_1(x), e_2(x), \ldots, e_s(x)$ are distinct (with highest coefficient 1) and occur in $i_1(x), i_2(x), \ldots, i_n(x)$. All powers $[e_1(x)]^{c_1}, \ldots, [e_s(x)]^{l_s}$, which are different from 1, are called the *elementary divisors* of A in $\mathbf{F}$

Note that the coefficients of the characteristic polynomial and the invariant factors of an integer matrix are all integers. Furthermore, the set of eigenvalues of A is the same as the set of all roots of $\chi_A(x)$ which, in turn, is the set of all roots of $\mu_A(x)$.

Next, we define some natural problems in linear algebra we are looking at. If nothing else is said, our domain for the algebraic problems are the integers.

1. PowerElement
   Input: an $n \times n$-matrix A and i, j, and m, $(1 \le i, j, m \le n)$.
   Output: $(A^m)_{i,j}$, the (i, j)-th element of A^m.
2. Determinant
   Input: an $n \times n$-matrix A.
   Output: $\det(A)$, the determinant of A.
3. CharPolynomial
   Input: an $n \times n$-matrix A.
   Output: $(c_0, c_1, \ldots, c_{n-1})$, the coefficients of the characteristic polynomial $\chi_A(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_0$ of the matrix A.
4. MinPolynomial
   Input: an $n \times n$-matrix A.
   Output: $(c_0, c_1, \ldots, c_{m-1})$, the coefficients of the minimal polynomial $\mu_A(x) = x^m + c_{m-1}x^{m-1} + \cdots + c_1x + c_0$ of the matrix A.
5. InvSystem
   Input: an $n \times n$-matrix A.
   Output: the system of invariant factors of the matrix A.

The first three problems are complete for **GapL** [ABO99,HT00,ST98]. MinPolynomial and InvSystem are in $\mathbf{RNC}^2$ [KS87], and in $\mathbf{NC}^2$ for integer matrices [Vil97].

For each of them, we define the corresponding *verification problem* as the graph of the corresponding function: for a fixed function $f(x)$, define

V-f as the set all pairs (x, y) such that $f(x) = y$. This yields the verification problems V-PowerElement, V-Determinant, V-CharPolynomial, V-MinPolynomial and V-InvSystem. The first three problems are known to be complete for $\mathbf{C_{=}L}$ [HT00]. We note that a special case of V-Determinant is Singularity where one has to decide whether the determinant of a matrix is zero. Singularity is complete for $\mathbf{C_{=}L}$ as well.

Related problems are computing the rank of a matrix, Rank, or deciding whether a system of linear equations is feasible, FSLE for short. FSLE is many-one complete for $\mathbf{AC}^0(\mathbf{C_{=}L})$ [ABO99].

Similarity is another many-one complete problem for $\mathbf{AC}^0(\mathbf{C_{=}L})$ [HT00]. Two square matrices A and B are *similar*, if there exists a nonsingular matrix P such that $A = P^{-1}BP$. It is well known that A and B are similar iff they have the same invariant factors or, what is the same, the same elementary divisors (see for example [Gan77]). Another characterization of similarity is based on tensor products. This was used by Byrnes and Gauger [BG77] to get the $\mathbf{AC}^0(\mathbf{C_{=}L})$ upper bound on Similarity.

3 The Minimal Polynomial

In this section we show that MinPolynomial is in $\mathbf{AC}^0(\mathbf{GapL})$ and is hard for **GapL**.

3.1 Upper Bound

We mentioned in the previous section that the minimal polynomial of an integer matrix can be computed in $\mathbf{NC}^2$ [Vil97]. We take a different approach and show that MinPolynomial is in $\mathbf{AC}^0(\mathbf{GapL})$, a subclass of $\mathbf{NC}^2$.

Let $m(x) = x^m + c_{m-1}x^{m-1} + \cdots + c_0$ be a monic polynomial. Then $m(x)$ is the minimal polynomial of A iff 1) m is an annihilating polynomial of A, i.e., $m(A) = A^m + c_{m-1}A^{m-1} + \cdots + c_0 I = \mathbf{0}$, and 2) for every monic polynomial $p(x)$ of degree smaller than $m(x)$, we have $p(A) \neq \mathbf{0}$.

Define vectors $\boldsymbol{a}_i = vec(A^i)$ for $i = 0, 1, 2, \ldots, n$, where $vec(A^i)$ is the vector of length n^2 obtained by putting the columns of A^i below each other. The equation $m(A) = \mathbf{0}$ can be rewritten as

$$\boldsymbol{a}_m + c_{m-1}\boldsymbol{a}_{m-1} + \ldots + c_0\boldsymbol{a}_0 = \mathbf{0}. \tag{1}$$

In other words, the vectors $\boldsymbol{a}_m, \ldots, \boldsymbol{a}_0$ are linearly dependent. Consequently, for some polynomial p with degree $k < m$, the inequation $p(A) \neq \mathbf{0}$ means that the vectors $\boldsymbol{a}_k, \ldots, \boldsymbol{a}_0$ are linearly independent.

In summary, the coefficients of $\mu_A(x)$ are the solution $(c_{m-1}, \ldots, c_0)$ of the system (1), for the smallest m where this system has a solution. Hence we have the following algorithm to compute $\mu_A(x)$:

MinPolynomial(A)
1 compute vectors $\boldsymbol{a}_i = vec(A^i)$ for $i = 0, \ldots, n$

2 determine m such that $\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{m-1}$ are linearly independent and $\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$ are linearly dependent
3 solve the linear system $\boldsymbol{a}_m + c_{m-1}\boldsymbol{a}_{m-1} + \cdots + c_0\boldsymbol{a}_0 = \boldsymbol{0}$
4 **return** $(1, c_{m-1}, \ldots, c_0)$, the coefficients of $\mu_A(x)$.

Step 1 and 3 in the above algorithm can be computed in **GapL** (see [ABO99]). In Step 2, checking linear independence of given vectors is in $\mathbf{coC_{=}L}$ and linear dependence is in $\mathbf{C_{=}L}$ [ABO99]. Hence we end up in the $\mathbf{AC}^0$-closure of **GapL**, namely $\mathbf{AC}^0(\mathbf{GapL})$. Recall that $\mathbf{AC}^0(\mathbf{GapL}) \subseteq \mathbf{NC}^2$. We conclude:

Theorem 3.1. MinPolynomial *is in* $\mathbf{AC}^0(\mathbf{GapL})$.

3.2 Lower Bound

Our main result is to show the hardness of the computation of the minimal polynomial of a matrix. Namely, we show that it is hard for **GapL**.

A problem known to be complete for **GapL** is PowerElement where one has to compute the entry (i, j) of A^m, for a $n \times n$ integer matrix A. W.l.o.g. we can focus on entry $(1, n)$ of A, i.e. $(A^m)_{1,n}$.

In order to reduce PowerElement to MinPolynomial, we construct a matrix C such that the value $(A^m)_{1,n}$ occurs as one of the coefficients of the minimal polynomial of C.

The reduction build on the techniques from Toda [Tod91], Valiant [Val92], and Hoang and Thierauf [HT00] to reduce matrix powering to the determinant, and the latter to the characteristic polynomial. We give the proof of this result here because we need the matrices constructed there. We follow the presentation from [ABO99] and [HT00].

Theorem 3.2. [HT00] PowerElement $\leq_m^{AC^0}$ CharPolynomial.

Proof. Let A be an $n \times n$ matrix and $1 \leq m \leq n$. W.l.o.g. we fix $i = 1$ and $j = n$ in the definition of PowerElement. In $\mathbf{AC}^0$ we construct a matrix C such that all the coefficients of its characteristic polynomial can be easily computed from the value $(A^m)_{1,n}$.

Interpret A as representing a directed bipartite graph on $2n$ nodes and e edges. That is, the nodes are arranged in two columns of n nodes each. In both columns, nodes are numbered from 1 to n. If entry $a_{k,l}$ of A is not zero, then there is an edge labeled $a_{k,l}$ from node k in the first column to node l in the second column. The number of non-zero entries in A is exactly e. Now, take m copies of this graph, put them in a sequence and identify each second column of nodes with the first column of the next graph in the sequence. Call the resulting graph G'.

Graph G' has $m + 1$ columns of nodes. The *weight* of a path in G' is the product of all labels on the edges of the path. The crucial observation now is that the entry at position $(1, n)$ in A^m is the sum of the weights of all paths in G' from node 1 in the first column to node n in the last column. Call these two nodes s and t, respectively.

Graph G' is further modified: for each edge (k,l) with label $a_{k,l}$, introduce a new node u and replace the edge by two edges, (k,u) with label 1 and (u,l) with label $a_{k,l}$. Now all paths from s to t have *even* length, but still the same weight. Add an edge labeled 1 from t to s. Call the resulting graph G. Let C be the adjacency matrix of G. Graph G has $N = m(n+e)+n$ nodes and therefore C is a $N \times N$ matrix.

From combinatorial matrix theory we know that the coefficient c_i in $\chi_C(x)$ equals the sum of the disjoint weighted cycles that cover $N-i$ nodes in G, with appropriate sign (see [BR91] or [CDS80] for more details). In the graph G, all edges go from a layer to the next layer. The only exception is the edge (t,s). So any cycle in G must use precisely this edge (t,s), and then trace out a path from s to t. Therefore each cycle in G have exactly the length $2m+1$, and the weighted sum of all these cycles is precisely $(A^m)_{1,n}$ with the sign -1. Hence $c_{N-(2m+1)} = -(A^m)_{1,n}$ and all other coefficients must be zero. That is,

$$\chi_C(x) = x^N - a x^{N-(2m+1)},$$

is the characteristic polynomial of C, where $a = (A^m)_{1,n}$. □

Theorem 3.3. PowerElement $\leq_m^{AC^0}$ MinPolynomial.

Proof. We consider the $N \times N$ matrix C from the previous proof in more detail.

Except for the edge from t to s, graph G is acyclic. Thus we can put the nodes of G in such an order, that adjacency matrix C is upper triangular for the first $N-1$ rows with zeros along the main diagonal. The last row of C has a one in the first position (representing edge (t,s)), and the rest is zero.

We also consider the upper triangle in C. Each column of graph G' was split in our construction into two columns and we got a new node on every edge. The first part we describe by the $n \times e$ matrix F:

$$F = \begin{pmatrix} 1\cdots1 & 0\cdots0 & \cdots & 0\cdots0 \\ 0\cdots0 & 1\cdots1 & \cdots & 0\cdots0 \\ \vdots & \vdots & \ddots & \vdots \\ 0\cdots0 & 0\cdots0 & \cdots & 1\cdots1 \end{pmatrix}$$

The number of ones in the k-th row of F is the number of edges leaving node k in the first column of G'.

From each of the newly introduced nodes there is one edge going out. Hence this second part we can describe by the $e \times n$-matrix S, which has precisely one non-zero entry in each row. The value of the non-zero entry is the weight of the corresponding edge in G'. With the construction of graph G it is not hard to see that $FS = A$. Now we can write C as a block matrix as follows:

$$C = \left(\begin{array}{c|ccccc} & F & & & & \\ & & S & & & \\ & & & \ddots & & \\ & & & & F & \\ & & & & & S \\ \hline L & & & & & \end{array}\right)$$

There is m-times matrix F, alternating with m-times matrix S. L is the $n \times n$ matrix with a one at position $(n, 1)$ and zero elsewhere. Hence C is a $(2m+1) \times (2m+1)$ block matrix. The empty places in C are all zero matrix.

Let a denote the element $(A^m)_{1,n}$. We claim that the minimal polynomial of C is $\mu_C(x) = x^{4m+2} - ax^{2m+1}$.

First, we observe that $d_{N-1}(x) = x^l$ for some l, because the minor of order $N-1$ of the matrix $xI - C$ at the position $(1, 1)$ is x^{N-1}. Therefore the minimal polynomial must have the form

$$\mu_C(x) = \chi_C(x)/d_{N-1}(x) = x^{N-l} - ax^{N-(2m+1)-l}.$$

Define polynomials $p_k(x) = x^{(2m+1)+k} - ax^k$ for $0 \leq k \leq N - (2m+1)$. To prove our claim, we have to show that $p_{2m+1}(C) = \mathbf{0}$ and $p_k(C) \neq \mathbf{0}$ for $k < 2m + 1$. To do so, we explicitly construct all the powers of C. The general form of C^i for $i \leq 2m$ is as follows:

$$C^i = \begin{array}{c} \begin{array}{cc} i & i+1 \\ \downarrow & \downarrow \end{array} \\ \left(\begin{array}{ccc|ccc} & & & * & & \\ & & & & \ddots & \\ & & & & & * \\ \hline + & & & & & \\ & \ddots & & & & \\ & & + & & & \end{array}\right) \end{array} \begin{array}{l} \leftarrow 1 \\ \vdots \\ \leftarrow 2m+1-i \\ \leftarrow 2m+2-i \\ \vdots \\ \leftarrow 2m+1 \end{array}.$$

The entry $(C^i)_{j,i+j}$ for $1 \leq j \leq 2m - i + 1$ and $i \leq 2m$ lies on the sub-diagonal $(* \cdots *)$ and has the following form:

$$(C^i)_{j,i+j} = \begin{cases} S^{(j-1) \bmod 2}(FS)^{\frac{i-1}{2}}F^{j \bmod 2}, & \text{for odd } i, \\ (FS)^{j \bmod 2}S^{(j-1) \bmod 2}(FS)^{\frac{i-2}{2}}F^{(j-1) \bmod 2}, & \text{otherwise.} \end{cases}$$

The entry $(C^i)_{2m+1-i+k,k}$ for $1 \leq k \leq i$ and $i \leq 2m$ lies on the sub-diagonal $(+ \cdots +)$ and has the following form:

$$(C^i)_{2m+1-i+k,k} = S^{(i+k) \bmod 2}(FS)^{\lfloor \frac{i-k}{2} \rfloor}L(FS)^{\lfloor \frac{k-1}{2} \rfloor}F^{(j-1) \bmod 2}.$$

From this we get in particular

$$\begin{aligned} C^{2m+1} &= diag\{A^mL,\ SA^{m-1}LF,\ A^{m-1}LA,\ \ldots,\ LA^m\}, \\ C^{4m+2} &= diag\{A^mLA^mL,\ SA^{m-1}LA^mLF,\ A^{m-1}LA^mLA,\ \ldots,\ LA^mLA^m\} \end{aligned}$$

Since $LA^mL = aL$, we have $p_{2m+1}(C) = C^{4m+2} - aC^{2m+1} = \mathbf{0}$. It remains to prove that $p_k(C) = C^{2m+1+k} - aC^k \neq \mathbf{0}$ for all $k \leq 2m$. Note that it suffices to prove this for $k = 2m$, because $p_k(C) = \mathbf{0}$ for some k implies $p_{k+1}(C) = \mathbf{0}$.

For technical reasons we assume that the input matrix A is a nonsingular upper triangular matrix. The following lemma says that we can w.l.o.g. make this assumption.

Lemma 3.4. *Suppose A is an $n \times n$ matrix. Then there is a nonsingular upper triangular $p \times p$ matrix B such that $(B^m)_{1,p} = (A^m)_{1,n}$.*

Proof. We define B as an $(m+1) \times (m+1)$ block matrix in which all the elements of the principal diagonal are $n \times n$ identity matrices, all the elements of the first super-diagonal are matrices A and all the the other elements are zero-matrices. For $p = (m+1)n$ we have $(B^m)_{1,p} = (A^m)_{1,n}$ as claimed. $\square$

We compute C^{4m+1} as the product $C^{2m+1}C^{2m}$. Now we have $p_{2m}(C) = \mathbf{0}$ iff $C^{4m+1} = aC^{2m}$ iff $A^m L A^m = aA^m$. However, the latter equation cannot hold: by Lemma 3.4 we can assume that A is nonsingular. Therefore $rank(A^m L A^m) = rank(L) = 1$, whereas $rank(aA^m) \neq 1$. We conclude that $p_{2m}(C) \neq \mathbf{0}$.

In summary, we have $\mu_C(x) = x^{4m+2} - ax^{2m+1}$, where $a = (A^m)_{1,n}$. Since the construction of graph G can be done in $\mathbf{AC}^0$, we have PowerElement $\leq_m^{AC^0}$ MinPolynomial as claimed. $\square$

3.3 The Invariant Factors

The system of all invariant factors of a matrix can be computed in $\mathbf{NC}^2$ [Vil97]. Since the minimal polynomial is one of the invariant factors, it follows from Theorem 3.3 that these are hard for **GapL** as well.

In the *verification versions* of the above problems we have given A and coefficients of one, respectively several polynomials and have to decide whether these coefficients represent in fact the minimal polynomial, respectively the invariant factors of A.

Note that in the case of the invariant factors we get potentially more information with the input than in the case of the minimal polynomial. Therefore, it could be that the invariant factors are easier to verify than the minimal polynomial. Interestingly we locate in fact the verification of the invariant factors in a seemingly smaller complexity class.

To verify the minimal polynomial we can simplify the above algorithm for MinPolynomial as follows:

v-MinPolynomial$(A, c_{m-1}, \ldots, c_0)$
compute vectors $\boldsymbol{a}_i = vec(A^i)$ for $i = 0, \ldots, m$
if $\boldsymbol{a}_m + c_{m-1}\boldsymbol{a}_{m-1} + \cdots + c_0\boldsymbol{a}_0 = \mathbf{0}$ **and**
  $\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{m-1}$ are linearly independent
then accept **else** reject.

Hence we get the same upper bound as for MinPolynomial, namely $\mathbf{AC}^0(\mathbf{GapL})$. Since MinPolynomial is hard for **GapL**, v-MinPolynomial must be hard for $\mathbf{C_{=}L}$. We summarize:

Corollary 3.5. v-MinPolynomial *is in* $\mathbf{AC}^0(\mathbf{GapL})$ *and hard for* $\mathbf{C_{=}L}$.

Next we show that the verification of the invariant factors is hard for $\mathbf{C_{=}L}$ as well. However, as an upper bound we get the seemingly smaller class $\mathbf{AC}^0(\mathbf{C_{=}L})$.

Theorem 3.6. v-InvSystem *is in* $\mathbf{AC}^0(\mathbf{C_{=}L})$ *and hard for* $\mathbf{C_{=}L}$.

Proof. **Inclusion.** Let $\mathcal{S} = \{i_1(x), \ldots, i_n(x)\}$ be the system of n given monic polynomials and let A be an $n \times n$ matrix. We construct the companion matrices that correspond to the non-constant polynomials in $\mathcal{S}$. Let B denote the diagonal block matrix of all these companion matrices. Recall that $\mathcal{S}$ is the system of invariant factors of A iff A is similar to B. Testing similarity can be done in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ [HT00], therefore v-InvSystem is in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ too.

Hardness. We continue with the setting from the proof of Theorem 3.3, in particular with matrix C. Our goal is to determine the system of all invariant factors of C. We have already shown that $i_1(x) = \mu_C(x) = x^{4m+2} - ax^{2m+1}$, where $(A^m)_{1,n} = a$. Next, we compute the invariant factors $i_2(x), \ldots, i_N(x)$.

It follows from the proof of Theorem 3.3 that $d_{N-1}(x) = x^{N-(4m+2)}$. Since $d_{N-1}(x) = i_2(x) \ \cdots \ i_N(x)$, each of the invariant factors must have the form x^l for some number l. Note that all non-constant invariant factors of the form x^l are already elementary divisors.

Define g_l to be the *number of occurrences of the elementary divisor* x^l. Clearly, if we have all numbers g_l, we can deduce the invariant factors. Numbers g_l can be determined from the ranks of matrices C^j (see [Gan77]). More precisely, let r_j denote *the rank of* C^j. The following formula relates the ranks to numbers g_j:

$$g_j = r_{j-1} + r_{j+1} - 2r_j, \tag{2}$$

for $j = 1, \ldots, t$, where $r_0 = N$ and t is the smallest index such that $r_{t-1} > r_t = r_{t+1}$. We can actually compute all the ranks r_j from the expressions we already have for matrices C^j.

Let us consider the blocks of C^j. By Lemma 3.4 we may assume that A is nonsingular, that is $\operatorname{rank}(F) = \operatorname{rank}(S) = \operatorname{rank}(A) = n$. Therefore $\operatorname{rank}(A^k) = \operatorname{rank}(A^k F) = \operatorname{rank}(A^k S) = n$ for any k. Hence blocks in C^j of the form $(FS)^k$, $(FS)^k F$, $(SF)^k$, or $(SF)^k S$ all have rank n (recall that $FS = A$). In all other blocks occurs matrix L. Recall that matrix L is all-zero except for the entry at the lower left corner, which is 1. Therefore, for any matrix M, we have $\operatorname{rank}(ML) = 1$ iff the n-th column of M is a non-zero column. Analogously, $\operatorname{rank}(LM) = 1$ iff the first row of M is a non-zero row. We conclude that all blocks that contain matrix L have rank 1.

Since the non-zero blocks of C^j are in pairwise different lines and columns, we can simply add up their ranks to obtain the rank of C^j. That way we get

$$r_j = \begin{cases} (2m+1-j)n + j, & \text{for } j = 1, \ldots, 2m, \\ 2m+1, & \text{for } 2m+1 \leq j. \end{cases}$$

The ranks don't change any more from $j = 2m+1$ on. Hence $t = 2m+1$. Plugged into the formula (2) we get

$$g_j = \begin{cases} N - n(2m+1), & \text{for } j = 1, \\ 0, & \text{for } j = 2, \ldots, 2m, \\ n-1, & \text{for } j = 2m-1. \end{cases} \tag{3}$$

From equations (3) we can deduce the invariant factors:

$$i_k(x) = \begin{cases} x^{2m+1}, & \text{for } k = 2, \ldots, n, \\ x, & \text{for } k = n+1, \ldots, N-2nm, \\ 1, & \text{for } k = N-2nm+1, \ldots, N. \end{cases} \tag{4}$$

In summary, $(A^m)_{1,n} = a$ iff $i_1(x) = x^{4m+2} + ax^{2m+1}$, and $i_2(x), \ldots, i_N(x)$ are as in (4). This completes the proof of Theorem 3.6. □

With the proof for the hardness result of v-InvSystem we remark that computing the system of invariant factors is hard for **GapL**.

4 Diagonalization

If a matrix A is similar to a diagonal matrix then we say for short that A is *diagonalizable*. That is, the Jordan normal form of A is a diagonal matrix, called J, where all the entries on the diagonal of J are the eigenvalues of A. We ask for the complexity to check whether a given matrix is diagonalizable.

An obvious way is to compute the Jordan normal form of A and then decide whether it is in diagonal form. However, in general, the eigenvalues of an integer matrix are in the complex field. That is, we run into the problem of dealing with real-arithmetic.

We use another characterization: matrix A is diagonalizable iff the minimal polynomial of A can be factored into pairwise different linear factors.

Theorem 4.1. Diagonalizable *is in* $\mathbf{AC}^0(\mathbf{GapL})$ *and hard for* $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$.

Proof. To decide whether a matrix A is diagonalizable we use the following algorithm:

Diagonalizable(A)
1 compute the minimal polynomial $m(x)$ of A
2 construct from $m(x)$ the companion matrix B
3 **if** B has pairwise different eigenvalues
4 **then** accept **else** reject.

We have already seen that step 1 is in $\mathbf{AC}^0(\mathbf{GapL})$. We argue below (see Corollary 4.3) that the condition in Step 3 can be decided in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$. Therefore Diagonalizable $\in \mathbf{AC}^0(\mathbf{GapL})$.

For the hardness result provide a reduction from FSLE, the set of *feasible linear equations*. That is FSLE is the set of pairs (A, b) such that the linear system $Ax = b$ has a solution $x \in \mathbf{Q}^n$, where A is $m \times n$ integer matrix and b a integer vector of length m. FSLE is complete for $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ [ABO99].

Define the symmetric matrix $B = \begin{pmatrix} \mathbf{0} & A \\ A^T & \mathbf{0} \end{pmatrix}$ and vector $c = (b^T, \mathbf{0})^T$ of length $m+n$. The reduction goes as follows:

$$(A, b) \in \text{FSLE} \iff (B, c) \in \text{FSLE} \tag{5}$$

$$\iff C = \begin{pmatrix} B & \mathbf{0} \\ 0\cdots 0 \end{pmatrix} \text{ is similar to } D = \begin{pmatrix} B & c \\ 0\cdots 0 \end{pmatrix} \tag{6}$$

$$\iff D \in \text{DIAGONALIZABLE}. \tag{7}$$

Equivalence (5) holds, since the system $A^T y = \mathbf{0}$ is always feasible.

To show equivalence (6), let x_0 be a solution of the system $Bx = c$. Define the nonsingular matrix $T = \begin{pmatrix} I & x_0 \\ \mathbf{0} & -1 \end{pmatrix}$. It is easy to check that $CT = TD$, therefore C is similar to D. Conversely, if the above system is not feasible, then C and D have different ranks and can therefore not be similar.

To show equivalence (7), observe that matrix C is symmetric. Therefore C is always diagonalizable, i.e., similar to a diagonal matrix, say C'. Now, if C is similar to D, then D is similar to C' as well, because the similarity relation is transitive. Hence D is diagonalizable as well.

Conversely, if D is diagonalizable then all of its elementary divisors are linear of the form $(\lambda - \lambda_i)$ where λ_i is any of its eigenvalues. Since C is diagonalizable, its elementary divisors are linear too. Note furthermore that C and D have the same characteristic polynomial. Therefore they must have the same system of elementary divisors. This implies that they are similar. □

To complete the proof of Theorem 4.1, we show how to test whether all eigenvalues of a given matrix are pairwise different.

Lemma 4.2. *All eigenvalues of the matrix A are pairwise different iff the matrix $B = A \otimes I - I \otimes A$ has 0 as an eigenvalue of multiplicity n (here, $\otimes$ denotes the* tensor product *(see [Gra81])).*

Proof. Just note that if $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of the $n \times n$ matrix A, then $(\lambda_i - \lambda_j)$, for all $1 \leq i, j \leq n$, are the eigenvalues of matrix B. □

Corollary 4.3. *Whether all eigenvalues of a matrix A are pairwise different can be decided in* $\mathbf{AC}^0(\mathbf{C_=L})$.

Proof. Let $B = A \otimes I - I \otimes A$. The matrix B has 0 as an eigenvalue of multiplicity n iff $\chi_B(x) = x^{n^2} + c_{n^2-1}x^{n^2-1} + \cdots + c_n x^n$ such that $c_n \neq 0$. Recall that the coefficients of the characteristic polynomial can be computed in **GapL**. Therefore the test whether $c_0 = c_1 = \cdots = c_{n-1} = 0$ and $c_n \neq 0$ is in $\mathbf{AC}^0(\mathbf{C_=L})$. □

Open Problems

The coefficients of the characteristic polynomial of a matrix can be computed in **GapL**. We do not know whether the minimal polynomial of a matrix can be computed in **GapL** as well. In other words, we want to close the gap between the upper bound (Theorem 3.1) and the lower bound (Theorem 3.3) we have for the minimal polynomial.

Analogously, we ask to close the gaps for the verification of the minimal polynomial (Corollary 3.5), the invariant factors (Theorem 3.6), and the diagonalization problem (Theorem 4.1). Note that *if one could show* that the minimal polynomial can be computed in $\mathbf{GapL}$ (or just in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$) then the minimal polynomial can be verified in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ and it follows that DIAGONALIZABLE is complete for $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$.

An important question not directly addressed here is whether $\mathbf{C}_=\mathbf{L}$ is closed under complement. An affirmative answer would solve many open problems in this area.

References

[ABO99] E. Allender, R. Beals, and M. Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8:99 –126, 1999.

[BG77] C. Byrnes and M. Gauger. Characteristic free, improved decidability criteria for the similarity problem. *Linear and Multilinear Algebra*, 5:153–158, 1977.

[BR91] R. Brualdi and H. Ryser. *Combinatorial Matrix Theory*, volume 39 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1991.

[CDS80] D. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs, Theory and Application*. Academic Press, 1980.

[Gan77] F. Gantmacher. *The Theory of Matrices*, volume 1 and 2. AMS Chelsea Publishing, 1977.

[Gra81] A. Graham. *Kronnecker Products and Matrix Calculus With Applications*. Ellis Horwood Ltd., 1981.

[HT00] T. M. Hoang and T. Thierauf. The complexity of verifying the characteristic polynomial and testing similarity. In *15th IEEE Conference on Computational Complexity (CCC)*, pages 87–95. IEEE Computer Society Press, 2000.

[KS87] E. Kaltofen and B. Saunders. Fast parallel computation of hermite and smith forms of polynomial matrices. *SIAM Algebraic and Discrete Methods*, 8:683–690, 1987.

[ST98] M. Santha and S. Tan. Verifying the determinant in parallel. *Computational Complexity*, 7:128–151, 1998.

[Sto98] A. Storjohann. An $O(n^3)$ algorithm for frobenius normal form. In *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 1998.

[Tod91] S. Toda. Counting problems computationally equivalent to the determinant. Technical Report CSIM 91-07, Dept. of Computer Science and Information Mathematics, University of Electro-Communications, Chofu-shi, Tokyo 182, Japan, 1991.

[Val92] L. Valiant. Why is boolean complexity theory difficult. In M.S. Paterson, editor, *Boolean Function Complexity*, London Mathematical Society Lecture Notes Series 169. Cambridge University Press, 1992.

[Vil97] G. Villard. Fast parallel algorithms for matrix reduction to normal forms. *Applicable Algebra in Engineering Communication and Computing (AAECC)*, 8:511–537, 1997.

[vzGG99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.