The Complexity of the Inertia *

Thanh Minh Hoang[1] and Thomas Thierauf [2, **]

[1] Abt. Theoretische Informatik, Universität Ulm, D-89069 Ulm
[2] FB Elektronik und Informatik, FH Aalen, D-73430 Aalen
{hoang,thierauf}@informatik.uni-ulm.de

Abstract. The *inertia* of a square matrix A is defined as the triple $(i_+(A), i_-(A), i_0(A))$, where $i_+(A)$, $i_-(A)$, and $i_0(A)$ are the number of eigenvalues of A, counting multiplicities, with positive, negative, and zero real part, respectively. A hard problem in Linear Algebra is to compute the inertia. No method is known to get the inertia of a matrix exactly in general. In this paper we show that the inertia is hard for **PL** (*probabilistic logspace*) and in some cases the inertia can be computed in **PL**. We extend our result to some problems related to the inertia. Namely, we show that matrix stability is complete for **PL** and the inertia of symmetric matrices can be computed in **PL**.

1 Introduction

A fundamental topic in linear algebra is the study of equivalence relations between matrices that naturally arise in theory and in applications. In computer science, we are interested in finding efficient algorithms to decide equivalence, or to construct canonical forms of a matrix for the relation under consideration. More general, we are interested in the *computational complexity* of these and related problems.

Most of these problems can be solved within certain logspace counting classes, all of which are contained in the parallel complexity class (uniform) $\mathbf{NC}^2$. In fact, the logspace counting class **GapL** [AO96] seems to capture the complexity of a lot of algebraic problems quite naturally. **GapL** is the extension of **#L** to integers in the same way as **#P** [Val79b,Val79a] can be extended to **GapP** [FFK94] in the polynomial time setting. The break-through result for **GapL** was that it precisely captures the complexity of the determinant of an integer matrix [Ber84,Dam91,Tod91,Vin91,Val92].

The verification of the value of a **GapL**-function defines the complexity class $\mathbf{C_=L}$. For example, the singularity problem (deciding whether a matrix is singular) is complete for $\mathbf{C_=L}$, because this is asking whether the determinant of a matrix is zero.

Inequalities on **GapL**-functions define the complexity class **PL**. For example, the problem to decide whether the determinant of a matrix is positive, is complete for **PL**.

* Supported by the Deutsche Forschungsgemeinschaft
** Part of the work done at Universität Ulm.

The computational problems over matrices like testing similarity, equivalence, and congruence are located in logspace counting classes like $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ (the $\mathbf{AC}^0$-closure of $\mathbf{C}_=\mathbf{L}$) or $\mathbf{PL}$. We describe these results in more detail.

Similarity. Two matrices A and B are *similar*, if there is a nonsingular matrix S such that $A = S^{-1}BS$. Santha and Tan [ST98] observed that testing similarity is in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$. Testing similarity is actually complete for this class [HT00]. A and B are similar iff they have the same invariant factors. The invariant factors can be computed in $\mathbf{AC}^0(\mathbf{GapL})$ and are hard for $\mathbf{GapL}$ [HT01].

Equivalence. Two matrices A and B are *equivalent*, if there exist nonsingular matrices P and Q, such that $A = PBQ$. A simple characterization of equivalence is that A and B have the same rank. Testing the equivalence of two matrices is complete for $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ [ABO99,HT00], as well as verifying one bit of the rank of matrix [ABO99].

Congruence. Two symmetric real matrices A and B are *congruent (via a real matrix)*, if there exists a nonsingular real matrix S such that $A = SBS^T$. By Congruence we denote the problem of testing the congruence. Congruence is hard for $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ [HT00]. An upper bound for Congruence is an open problem in [HT00]. In this paper we show that Congruence $\in \mathbf{PL}$ by considering the inertia of symmetric matrices.

The computational complexity of the inertia and its related problems is the main topic of this paper. In Section 3.2, we use the *Routh-Hurwitz Theorem* to show that the inertia of a matrix (under the restrictions of the Routh-Hurwitz Theorem) can be computed in $\mathbf{PL}$. In Section 3.3 we show that the inertia is hard for $\mathbf{PL}$.

An alternative way to compute the inertia of a matrix could be to determine all the roots of the characteristic polynomial of the given matrix. With the $\mathbf{NC}^2$-algorithm of Neff and Reif [Nef94,NR96] these roots can be approximated to some precision [ABO]. However, it is not clear to what precision we have to approximate a root in order to tell it apart from zero. This result is different from our approach.

We also consider the verification of the inertia. That is, for matrix A and integers $\mathfrak{p}$, $\mathfrak{n}$, and $\mathfrak{z}$, one has to decide whether $(\mathfrak{p},\mathfrak{n},\mathfrak{z})$ is the inertia of A. We show in Section 3 that for certain matrices the verification is complete for $\mathbf{PL}$.

A system of differential equations is stable iff its coefficient matrix is stable (matrix whose eigenvalues have negative real parts). Therefore, the study of stable matrices has a long-standing history and it is an important topic in Linear Algebra. We prove in Section 4 that the problem of deciding whether all eigenvalues of a matrix have positive real parts is complete for $\mathbf{PL}$. A matrix has no eigenvalues with negative real part is called *positive semistable*. We show in Section 4 that the problem to decide whether a matrix is positive semistable is in $\mathbf{AC}^0(\mathbf{GapL})$ and is hard for $\mathbf{PL}$.

Finally, in Section 5 we prove that the inertia of a *symmetric* integer matrix can be computed in $\mathbf{PL}$. It follows that the congruence of two matrices can be decided in $\mathbf{PL}$. Note that there are deterministic algorithms for the inertia of symmetric integer matrices see for example [For00].

2 Preliminaries

We assume familiarity with some basic notions of complexity theory and linear algebra. We refer the readers to the papers [ABO99,AO96] for more details and properties of the considered complexity classes, and to the textbooks [Gan77,HJ91,HJ85] for more background in linear algebra.

Complexity Classes. For a nondeterministic Turing machine M, let gap_M denote the difference between the number of accepting and rejecting computation paths of M on input x. The function class **GapL** is defined as the class of all functions $gap_M(x)$ such that M is a nondeterministic logspace bounded Turing machine M.

It is easy to see that **GapL** is closed under addition, subtraction, and multiplication. Allender, Arvind, and Mahajan [AAM99] showed that **GapL** is closed under composition. Even stronger, they showed that the determinant of a matrix A where each entry of A is computed in **GapL** can be computed in **GapL**.

A set S is in $\mathbf{C_=L}$, if there exists a function $f \in \mathbf{GapL}$ such that for all x we have $x \in S \iff f(x) = 0$. A set S is in **PL** if there is a function $f \in \mathbf{GapL}$ such that for all x we have $x \in S \iff f(x) > 0$. Ogihara [Ogi98] showed that **PL** is closed under logspace Turing reductions.

By $\mathbf{AC^0(C_=L)}$, $\mathbf{AC^0(PL)}$, and $\mathbf{AC^0(GapL)}$ we denote the class of sets that are $\mathbf{AC^0}$-reducible to a set in $\mathbf{C_=L}$, **PL**, respectively a function in **GapL**. All these classes are contained in $\mathbf{TC^1}$, a subclass of $\mathbf{NC^2}$. The known relationships among these classes are as follows:

$$\mathbf{C_=L} \subseteq \mathbf{AC^0(C_=L)} \subseteq \mathbf{AC^0(PL)} = \mathbf{PL} \subseteq \mathbf{AC^0(GapL)} \subseteq \mathbf{TC^1} \subseteq \mathbf{NC^2}.$$

Unless otherwise stated, all reductions in this paper are logspace many-one.

Linear Algebra. Let M_n be the set of $n \times n$ integer matrices. For $A \in M_n$ we denote the *characteristic polynomial* of A by $\chi_A(x)$, that is $\chi_A(x) = \det(xI - A)$ is a degree n polynomial, $\deg(\chi_A) = n$. The *companion matrix* of the polynomial $p(x) = x^n + \alpha_1 x^{n-1} + \cdots + \alpha_n$ is the matrix $P \in M_n$, where the last column is $(-\alpha_n, \ldots, -\alpha_1)^T$, all entries on the lower subdiagonal are 1. All the other elements are zero. The property of P we use is that $\chi_P(x) = p(x)$.

The *inertia* of a matrix $A \in M_n$ is defined as the triple $(i_+(A), i_-(A), i_0(A))$, where $i_+(A)$, $i_-(A)$, and $i_0(A)$ are the number of eigenvalues of A, counting multiplicities, with positive, negative, and zero real part, respectively. Note that $i_+(A), i_-(A), i_0(A)$ are nonnegative integers and the sum of these is exactly n.

Matrix A is called *positive stable*, if $i(A) = (n, 0, 0)$, and *negative stable*, if $i(A) = (0, n, 0)$. Furthermore, A is called as *positive semistable* if $i_-(A) = 0$. In case that A is *real symmetric* all eigenvalues of A are real and the word *"stable"* will be replaced by *"definite"*.

For square matrices $A = (a_{i,j}) \in M_n$ and $B \in M_m$, the *Kronecker product* $A \otimes B$ is defined as the matrix $(a_{i,j}B) \in M_{nm}$. The *Kronecker sum* $A \oplus B$ is defined as the matrix $A \otimes I_m + I_n \otimes B \in M_{nm}$, where $I_n \in M_n$ and $I_m \in M_m$ are identity matrices. If $\lambda_1, \ldots, \lambda_n$ and $\mu_1, \ldots, \mu_m$ are the eigenvalues of A and B, respectively, then the eigenvalues of $A \otimes B$ are $\lambda_k \mu_l$, and the eigenvalues of $A \oplus B$ are $\lambda_k + \mu_l$, for all $1 \leq k \leq n$ and $1 \leq l \leq m$.

Problems We define some natural problems in linear algebra that we are considering. Unless otherwise specified, our domain for the algebraic problems are the integers. The two following functions are complete for **GapL** [ABO99,ST98].

PowerElement: given $A \in M_n$ and m, compute $(A^m)_{1,n}$.

Determinant: given $A \in M_n$, compute $\det(A)$.

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 of all pairs (x, y) such that $f(x) = y$. This yields v-PowerElement and v-Determinant. They are known to be complete for $\mathbf{C_{=}L}$.

We denote by PosPowerElement the problem of deciding whether one element of the power of a matrix is positive, and by PosDeterminant the problem of deciding whether the determinant of a matrix is positive. These problems are complete for **PL**.

We define Inertia to be the problem of computing one bit of $i(A)$ (with respect to some fixed coding). That is,

$$\text{Inertia} = \{(A, k, b) | \text{ the } k\text{-th bit of } i(A) \text{ is } b\}.$$

By v-Inertia we denote the problem of verifying the value of $i(A)$.

PosStable and PosSemistable are the sets of all positive stable, semistable matrices, respectively. PosDefinite and PosSemidefinite are the sets of all positive definite, semidefinite matrices, respectively.

3 The Inertia

3.1 The Routh-Hurwitz Theorem

The Routh-Hurwitz Theorem (see [Gan77], Volume II, Chapter XV) provides a method for determining the number of roots in the right half-plane of a given real polynomial. Since the roots of the characteristic polynomial $\chi_A(x)$ are the eigenvalues of the matrix A, we can compute the inertia of A by applying the Routh-Hurwitz method to $\chi_A(x)$.

Let $A \in M_n$. Consider the characteristic polynomial of A

$$\chi_A(x) = x^n + c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_n.$$

Define $c_0 = 1$. The *Routh-Hurwitz matrix* $\Omega(A) = (\omega_{i,j}) \in M_n$ is defined as

$$\Omega(A) = \begin{pmatrix} c_1 & c_3 & c_5 & c_7 & \cdots & 0 \\ c_0 & c_2 & c_4 & c_6 & \cdots & 0 \\ 0 & c_1 & c_3 & c_5 & \cdots & 0 \\ 0 & c_0 & c_2 & c_4 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & c_n \end{pmatrix}.$$

That is, the diagonal elements of $\Omega(A)$ are $\omega_{i,i} = c_i$. In the i-th column, the elements above the diagonal are $\omega_{i-1,i} = c_{i+1}$, $\omega_{i-2,i} = c_{i+2}$, $\ldots$ until we reach either the first row $\omega_{1,i}$ or c_n. In the latter case, the remaining entries are filled with zeros. The elements below $\omega_{i,i}$ are $\omega_{i+1,i} = c_{i-1}$, $\omega_{i+2,i} = c_{i-2}$, ..., c_1, c_0, 0, 0, ... down to the last row $\omega_{n,i}$.

The successive leading principal minors D_i of $\Omega(A)$ are called the *Routh-Hurwitz determinants*, they are

$$D_1 = c_1, \quad D_2 = \det \begin{pmatrix} c_1 & c_3 \\ c_0 & c_2 \end{pmatrix}, \quad \cdots, \quad D_n = \det(\Omega(A)).$$

Theorem 3.1 (Routh-Hurwitz). *If $D_n \neq 0$, then the number of roots of the polynomial $\chi_A(x)$ in the right half-plane is determined by the formula*

$$i_+(A) = V(1, D_1, \frac{D_2}{D_1}, \ldots, \frac{D_n}{D_{n-1}}),$$

where $V(x_1, x_2, \ldots)$ computes the number of sign alternations in the sequence of numbers $x_1, x_2, \ldots$. For the calculation of the values of V, for every group of p successive zero Routh-Hurwitz determinants (p is always odd!)

$$D_s \neq 0, \; D_{s+1} = \cdots = D_{s+p} = 0, \; D_{s+p+1} \neq 0$$

we have to set $V(\frac{D_s}{D_{s-1}}, \frac{D_{s+1}}{D_s}, \ldots, \frac{D_{s+p+2}}{D_{s+p+1}}) = h + \frac{1-(-1)^h \varepsilon}{2}$ *, where $p = 2h - 1$ and $\varepsilon = \mathrm{sign}(\frac{D_s}{D_{s-1}} \frac{D_{s+p+2}}{D_{s+p+1}})$. For $s = 1$, $\frac{D_s}{D_{s-1}}$ is to be replaced by D_1; and for $s = 0$, by c_0.*

Let us discuss the case when $D_n = 0$. It is known that $D_n = 0$ iff $\chi_A(x)$ has a pair of opposite roots x_0 and $-x_0$ (see [Gan77]). Define

$$p_1(x) = x^n + c_2 x^{n-2} + c_4 x^{n-4} + \cdots \quad \text{and} \quad p_2(x) = c_1 x^{n-1} + c_3 x^{n-3} + \cdots .$$

Then $\chi_A(x) = p_1(x) + p_2(x)$ and $p_1(x_0) = p_2(x_0) = 0$. Therefore, x_0 is also a root of the greatest common divisor $g(x)$ of $p_1(x)$ and $p_2(x)$. We can write $\chi_A(x) = g(x)\chi_A^*(x)$, where the polynomial $\chi_A^*(x)$ has no pair of opposite roots, i.e. the Routh-Hurwitz matrix of $\chi_A^*(x)$ is nonsingular. Let B be the companion matrix of $g(x)$ and C be the companion matrix of $\chi_A^*(x)$. Then we have

$$i(A) = i(B) + i(C).$$

Note that all nonzero-eigenvalues of B are pairs of opposite values. The Routh-Hurwitz method does not work in the case where B has some opposite eigenvalues *on* the imaginary axis, and no method is known to get the exact number of roots of a polynomial on an axis (to the best of our knowledge).

However, there are methods to determine the number of *distinct* roots of a polynomial on an axis, and we will show below how to use these methods to solve at least some cases where $D_n = 0$.

Let P be the companion matrix of a polynomial $p(x)$, where $\deg(p(x)) = n$. The *Hankel matrix* $H = (h_{i,j}) \in M_n$ associated with $p(x)$ is defined as $h_{i,j} = \mathrm{trace}(P^{i+j-2})$, for $i, j = 1, \ldots, n$, where $\mathrm{trace}(P^{i+j-2})$ is the sum of all diagonal elements of P^{i+j-2}. Note that H is symmetric. By $\mathrm{sig}(H)$ we denote the *signature* of H, that is the difference between $i_+(H)$ and $i_-(H)$. The following Theorem can be found in Volume II, Chapter XV of [Gan77].

Theorem 3.2. *1) The number of distinct* real *roots of $p(x)$ is equal* $\mathrm{sig}(H)$.
2) The number of all *distinct roots of $p(x)$ is equal to the rank of H.*

3.2 Upper Bounds

We consider the complexity to compute the inertia via Theorem 3.1. The first step is to compute all the coefficients c_i of $\chi_A(x)$ and from these all Routh-Hurwitz determinants D_i, for $i = 1, \dots, n$. Since the coefficients $c_1, \dots, c_n$ are computable in **GapL**, each of the determinants $D_1, \dots, D_n$ can be computed in **GapL** as well [AAM99].

If $D_n \neq 0$, i.e. $\Omega(A)$ is nonsingular, we can compute $i_+(A)$ by using the formulas from Theorem 3.1: a logspace machine with a **PL** oracle can ask, for each of the determinants $D_1, \dots, D_n$, if it is positive, negative, or zero. Because $i_-(A) = i_+(-A)$, we can apply the same method to compute $i_-(A)$ and get $i_0(A) = n - i_+(A) - i_-(A)$. Hence all three values of $i(A)$ can be computed in **PL**.

Theorem 3.3. *The inertia of a matrix A with the property that $\Omega(A)$ is nonsingular can be computed in* **PL**.

Let us consider the case when $D_n = 0$, i.e. when $\Omega(A)$ is singular. We decompose $\chi_A(x) = g(x)\chi^*_A(x)$, as described in the previous section. Recall that $g(x)$ is the greatest common divisor of two polynomials $p_1(x)$ and $p_2(x)$. Therefore the coefficients of $g(x)$ can be computed as the solution of a system of linear equations (see [Koz91]), which can be done in $\mathbf{AC}^0(\mathbf{GapL})$ (see [ABO99]). It follows that we can compute the polynomial $\chi^*_A(x)$ in $\mathbf{AC}^0(\mathbf{GapL})$ as well. In other words, each of the elements of the companion matrices B (of $g(x)$) and C (of $\chi^*_A(x)$) can be computed in $\mathbf{AC}^0(\mathbf{GapL})$.

There is no method to compute $i(B)$ in general. However, in some cases, when $g(x)$ is easy, we can do so anyway. Suppose for example that

$$g(x) = x^t, \text{ for some } t \geq 0.$$

Equivalently we can say that B (and hence A) has *no opposite nonzero-eigenvalues*. Then it is clear that $i(B) = (0, 0, t)$, and hence $i(A) = (0, 0, t) + i(C)$.

Note that the decision whether A has no opposite nonzero-eigenvalues is in $\mathbf{coC_=L}$: with the greatest t such that x^t is a divisor of $\chi_A(x)$ (it is possible that $t = 0$) we can decide whether the Routh-Hurwitz matrix associated with the polynomial $\frac{\chi_A(x)}{x^t}$ is nonsingular.

Corollary 3.4. *The inertia of a matrix with no opposite nonzero-eigenvalues can be computed in* **PL**.

We can considerably extend Corollary 3.4 to the following theorem.

Theorem 3.5. *The inertia of a matrix A with the property that*

1) A has all *opposite eigenvalues on the imaginary axis,* or

2) A has no *opposite eigenvalues on the imaginary axis,*

can be computed in $\mathbf{AC}^0(\mathbf{GapL})$.

Proof. Assume that the condition on A is fulfilled. Let B be again the companion matrix of $g(x)$. The triple $i(B)$ can be easily computed. In the case 1) we have $i(B) = (0, 0, \deg(g(x))$ and in the case 2) we have $i(B) = (\frac{1}{2}\deg(g(x)), 0, \frac{1}{2}\deg(g(x)))$. Thus we can compute $i(A)$ by adding $i(B)$ to $i(C)$.

We show how to check the condition on A by using Theorem 3.2.

Since Theorem 3.2 deals with the real axis instead of the imaginary axis, we first turn $g(x)$ by 90°: consider the matrix $E = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$. Its eigenvalues are $+i$ and $-i$. Define $D = B \otimes E$. The eigenvalues of D are $i\lambda_k(B)$ and $-i\lambda_k(B)$ where $\lambda_k(B)$ runs through all eigenvalues of B. It follows that the number of distinct purely imaginary eigenvalues of B is the same as the number of distinct real eigenvalues of D.

Finally, let H be the Hankel matrix of $\chi_D(x)$. From Theorem 3.2 we have

$$\begin{aligned} i_+(B) = 0 &\iff \mathrm{rank}(H) = \mathrm{sig}(H), \\ i_0(B) = 0 &\iff \mathrm{sig}(H) = 0. \end{aligned}$$

The conditions on the right-hand side can be decided in $\mathbf{AC}^0(\mathbf{GapL})$. This proves the theorem. □

Because of the closure properties of **PL** and $\mathbf{AC}^0(\mathbf{GapL})$, we get the same upper bounds for the verification of the inertia.

3.3 Lower Bounds

Theorem 3.6. Inertia *and* v-Inertia *are hard for* **PL**.

Proof. We reduce PosPowerElement, a complete problem for **PL**, to Inertia and v-Inertia. Let $A \in M_n$ be an input for PosPowerElement. One has to decide whether $(A^m)_{1,n} > 0$ for a given $m > 1$.

There is a reduction from matrix powering to the characteristic polynomial which is shown in [HT00] (see also [HT01] and [HT02]): given A, m and a, one can construct a matrix B in $\mathbf{AC}^0$ such that

$$(A^m)_{1,n} = a \iff \chi_B(x) = x^{N-2m-1}\left(x^{2m+1} - a\right),$$

where $N = m(n + d) + n$, and d is the number of nonzero-elements in A.

The eigenvalues of B are the roots of $\chi_B(x)$. We consider the case when $a = (A^m)_{1,n} \neq 0$. The roots of $x^{2m+1} - a$ are the corners of a regular $(2m + 1)$-gon inscribed in a circle of radius $a^{\frac{1}{2m+1}}$ with its center at the origin. Since $2m + 1$ is odd, none of these roots lies on the imaginary axis. This implies that

$i_0(B) = N - (2m+1)$, and one of $i_+(B)$ and $i_-(B)$ is m and the other is $m+1$. Moreover, these values depend on the sign of a. Namely, if $a > 0$, we have

$$i_+(B) = \begin{cases} m+1 & \text{if } 2m+1 \equiv 1 \pmod 4, \\ m & \text{if } 2m+1 \equiv 3 \pmod 4. \end{cases} \tag{1}$$

Note in particular that $i_+(B)$ in (1) is always odd. Analogously, $i_+(B)$ is even if $a < 0$. In the case where $(A^m)_{1,n} = 0$, we have $i(B) = (0, 0, N)$.

In summary, we can compute values $\mathfrak{p}$, $\mathfrak{n}$, $\mathfrak{z}$ in logspace such that

$$\begin{aligned} (A^m)_{1,n} > 0 &\iff i(B) = (\mathfrak{p}, \mathfrak{n}, \mathfrak{z}) \\ &\iff i_+(B) = \text{odd} \end{aligned}$$

This proves the theorem. ☐

Note also that B in the above proof has no pair of opposite nonzero-eigenvalues. Therefore B fulfills the condition of Corollary 3.4.

Corollary 3.7. *The computation and the verification of the inertia of a matrix with no opposite nonzero-eigenvalues are complete for* **PL**.

4 Stability

Theorem 4.1. PosStable $\in$ **PL** *and* PosSemistable $\in$ $\mathbf{AC}^0(\mathbf{GapL})$

Proof. A is positive stable iff all the Routh-Hurwitz determinants of the matrix $\Omega(-A)$ are positive. Hence, positive stability of A can be decided in **PL**.

If $\Omega(A)$ is nonsingular, then PosSemistable $\in$ **PL** by Theorem 3.3. So assume that $\Omega(A)$ is singular. As described in Section 3.1, we decompose $\chi_A(x) = g(x)\chi_A^*(x)$ in $\mathbf{AC}^0(\mathbf{GapL})$. Let B and C be the companion matrices of $g(x)$ and $\chi_A^*(x)$, respectively. Then A is positive semistable iff B is positive semistable and C is positive stable. Matrix B is positive semistable iff all eigenvalues of B are on the imaginary axis. Now the result follows from Theorem 3.5. ☐

Now we consider the hardness of the stability problems. A matrix A is nonsingular iff AA^T is positive definite. AA^T can be computed in $\mathbf{NC}^1$. Therefore, PosDefinite is hard for $\mathbf{coC_=L}$ under $\mathbf{NC}^1$ many-one reductions.

Corollary 4.2. PosDefinite *is hard for* $\mathbf{coC_=L}$.

Theorem 4.3. PosStable *and* PosSemistable *are hard for* **PL**.

Proof. By NegDeterminant we denote the set of all matrices with negative determinant. Note that NegDeterminant is complete for **PL**. We construct a reduction from NegDeterminant to PosStable as follows.

Let $A \in M_n$. Let $d_{1,1}, \ldots, d_{n,n}$ be the diagonal elements of A^TA. The *Hadamard Inequality* states that $|\det(A)| \leq (d_{1,1} \cdots d_{n,n})^{1/2}$. W.l.o.g. we can assume that the input matrix A is a 0-1 matrix and that no row or column of A

has more than two 1's in it [All02]. Therefore $d_{i,i} \leq 2$ for all i. By the Hadamard Inequality we get the following bound for $\det(A)$:

$$-2^n < \det(A) < 2^n.$$

Define $t = \lceil \frac{n}{2m+1} \rceil (2m+1)$, for an integer $m \geq 1$. Since $n \leq t$, we have $\det(A) + 2^t > 0$ and

$$\det(A) < 0 \iff \det(A) + 2^t < 2^t. \tag{2}$$

Lemma 4.4. *We can construct a matrix $B \in M_k$ and m such that*

$$(B^m)_{1,k} = \det(A) + 2^t. \tag{3}$$

Note that m depends on t, and we defined t in terms of m. This makes the construction a bit tricky. We prove the lemma below.

Define $b = (B^m)_{1,k}$. We further reduce B to a matrix C such that

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

where $N = m(k+d) + k$, and d is the number of elements different from zero of B [HT00] (see also [HT01] and [HT02]). This is an $\mathbf{AC}^0$-reduction.

As explained in Theorem 3.6, matrix C has $N - 2m - 1$ eigenvalues zero and $2m+1$ eigenvalues as the roots of $x^{2m+1} - b$. The latter $2m+1$ eigenvalues of C are complex and lie on the circle of radius $r = b^{\frac{1}{2m+1}}$ (with the origin as center). Since $b > 0$, the eigenvalue with the largest real part is $\lambda_{\max}(C) = r$.

We shift the eigenvalues of C by $s = 2^{\frac{t}{2m+1}} = 2^{\lceil \frac{n}{2m+1} \rceil}$. That is, define the matrix $D = -C + sI$. The eigenvalue of C with the largest real part becomes the eigenvalue of D with the smallest real part: $\lambda_{\min}(D) = -r + s$. So we get

$$b < 2^t \iff r < s \iff \lambda_{\min}(D) > 0. \tag{4}$$

By (2) and (4) we have $A \in$ NegDeterminant $\iff D \in$ PosStable.

An analogous argument reduces the set of matrices with nonpositive determinants (a **PL**-complete set) to PosSemistable □

Proof of Lemma 4.4. Since PowerElement is complete for **GapL**, we can compute $B_0 \in M_l$ and an exponent m in logspace such that $(B_0^m)_{1,l} = \det(A)$.

Define a $(m+1) \times (m+1)$ block matrix F with m times B_0 on the first upper subdiagonal, all other blocks are zero.

Define $S = \begin{pmatrix} s^2 & s^3 \\ 0 & 0 \end{pmatrix}$, where $s = 2^{\lceil \frac{n}{2m+1} \rceil}$. It is easily to get $S^m = \begin{pmatrix} s^{2m} & s^{2m+1} \\ 0 & 0 \end{pmatrix}$ and $s^{2m+1} = 2^t$.

We define an $l(m+1) \times 2$ matrix T whose elements at the position $(1,1)$ and $(l(m+1), 2)$ are 1 and all the others are zero.

Finally, for $k = l(m+1) + 2$ we define

$$B = \begin{pmatrix} F & FT + TS \\ \mathbf{0} & S \end{pmatrix},$$

and claim that the matrix B fulfills (3). From the powers of B we get

$$B^m = \begin{pmatrix} F^m & F^mT + 2F^{m-1}TS + 2F^{m-2}TS^2 + \cdots + 2FTS^{m-1} + TS^m \\ \mathbf{0} & S^m \end{pmatrix}.$$

In particular, for each $1 \leq i \leq m$, F^i has a very simple form: on its i-th upper subdiagonal are purely B^i and all the other block-elements are zeromatrix. Furthermore, it is not hard to see that $F^{m-i}TS^i = \mathbf{0}$ for all $i < m$. Thus

$$B^m = \begin{pmatrix} F^m & F^mT + TS^m \\ \mathbf{0} & S^m \end{pmatrix}.$$

Now, it is not hard to see that $(B^m)_{1,k} = \det(A) + 2^t$. This proves the lemma.

5 The Congruence of Symmetric Matrices

Recall that all the eigenvalues of a symmetric (real) matrix A are real. Therefore, if we decompose $\chi_A(x) = g(x)\chi_A^*(x)$ as explained in Section 3.1, $g(x)$ has only real roots. Let t be the multiplicity of the zero-eigenvalue of the companion matrix B of $g(x)$. Then we have $i(B) = (\frac{1}{2}(\deg(g(x)) - t), \frac{1}{2}(\deg(g(x)) - t), t)$. It follows that $i(A)$ can be computed in $\mathbf{AC}^0(\mathbf{GapL})$. Actually, we can show a better upper bound for this problem. By $\textsc{Inertia}_{sym}$ we denote the restriction of $\textsc{Inertia}$, where the input matrix A is a symmetric integer matrix.

Theorem 5.1. $\textsc{Inertia}_{sym}$ *is in* $\mathbf{PL}$.

Proof. Let $A \in M_n$. If A is singular, we can compute $\chi_A(x)$ and determine the multiplicity t of eigenvalues 0 (in $\mathbf{GapL}$). Then it suffices to compute the inertia of the companion matrix of polynomial $\chi_A(x)/x^t$. The latter matrix is nonsingular. Therefore we may as well assume that A is nonsingular.

If A has no pair of opposite nonzero-eigenvalues, then $\textsc{Inertia}_{sym}$ is in $\mathbf{PL}$, as explained in Section 3.2. Therefore we consider the case when A has some pair of opposite nonzero eigenvalues. The idea is to determine a positive rational number ε such that the value $i_+(M)$ of the matrix $M = A - \varepsilon I$ is equal to $i_+(A)$ and the Routh-Hurwitz matrix $\Omega(M)$ is regular. Then we can apply Theorem 3.3 to compute $i_+(M)$.

The *spectral radius* of A is a bound on the distance of the eigenvalues of A from the origin. Furthermore, if $\| \cdot \|$ is any matrix norm, then $\rho(A) \leq \| A \|$ (see [HJ85]). We choose $\| \cdot \|$ as the *maximum column sum matrix norm*, i.e., $\nu(A) = \| A \| = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$, and we have $\rho(A) \leq \nu(A)$. Because A is nonsingular, we can define $r_1 = (\nu(A^{-1}) + 1)^{-1}$. Let $\lambda_1(A), \ldots, \lambda_n(A)$ be the eigenvalues of A. Then $\lambda_i(A) > r_1$, for $i = 1, \ldots, n$. Now let $0 < \varepsilon \leq r_1$ and define matrix $M = A - \varepsilon I$. The eigenvalues of M are $\lambda_1(A) - \varepsilon, \ldots, \lambda_n(A) - \varepsilon$ and we have $i_+(M) = i_+(A)$.

It remains to determine ε such that $\Omega(M)$ is nonsingular. Observe that M has this property iff for all $i \neq j$

$$\lambda_i(A) - \varepsilon \neq -(\lambda_j(A) - \varepsilon) \iff \lambda_i(A) + \lambda_j(A) \neq 2\varepsilon. \qquad (5)$$

The eigenvalues of $S = A \oplus A$ are $\lambda_i(A) + \lambda_j(A)$ for all $1 \le i, j \le n$. Thus equivalent to condition on the right-hand side of (5) is that 2ε is not an eigenvalue of S. Matrix S is singular, because A has some pair of opposite nonzero-eigenvalues. If we decompose $\chi_S(x) = x^k \chi_S^*(x)$ such that $\chi_S^*(0) \neq 0$, then the companion matrix S^* of $\chi_S^*(x)$ is nonsingular and k is exactly the multiplicity of the eigenvalue 0 of S. Since S^* is nonsingular, we can define $r_2 = (\nu({S^*}^{-1}) + 1)^{-1}$. Each of the eigenvalues of S^* has absolute value greater than r_2. Hence $\Omega(M)$ is nonsingular if $0 < 2\varepsilon \le r_2$. In summary, we can choose $\varepsilon = \min\{r_1, r_2/2\}$.

The value ε, each element of M, and each of the Routh-Hurwitz determinants of $\Omega(M)$ can be computed in **GapL**, because the elements of A^{-1} and ${S^*}^{-1}$ are computable in **GapL** ([AAM99]). Therefore Inertia_{sym} is in **PL** by Theorem 3.3. □

Since each bit of $i_+(A)$ can be verified in **PL**, the values of $i(A)$ can be verified in **PL**, too. This implies that the problem of testing whether two given symmetric matrices have the same inertia (that is Congruence) is in **PL**. This solves an open problem in [HT00].

Corollary 5.2. *Congruence, PosDefinite, PosSemidefinite* $\in$ **PL**.

Summary and Open Questions

The table summarizes the lower and upper bounds for some of the problems considered in this paper. An obvious task for further research is to close the gap between the lower and the upper bound where it doesn't match.

Problem	hard for	contained in
PosStable	**PL**	**PL**
PosSemistable	**PL**	$\mathbf{AC}^0(\mathbf{GapL})$
Congruence	$\mathbf{AC}^0(\mathbf{C_=L})$	**PL**
PosDefinite	$\mathbf{coC_=L}$	**PL**

A major challenge remains to compute the inertia in general.

Acknowledgments

We thank Eric Allender and the referees for many helpful comments on the paper. In particular, Eric pointed us to the results in [AAM99] from which we got that all Routh-Hurwitz determinants are computable in **GapL**.

References

[AAM99] E. Allender, V Arvind, and M. Mahajan. Arithmetic complexity, Kleene closure, and formal power series, 1999.

[ABO] E. Allender and M. Ben-Or. Electronic Communication, 2001.

[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.

[All02] E. Allender. Personal Communication, 2002.

[AO96] E. Allender and M. Ogihara. Relationship among PL, #L, and the determinant. *RAIRO-Theoretical Informatics and Applications*, 30:1–21, 1996.

[Ber84] S. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18:147–150, 1984.

[Dam91] C. Damm. DET = $L^{(\#L)}$. Technical Report Informatik-Preprint 8, Fachbereich Informatik der Humboldt Universitaet zu Berlin, 1991.

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48:116–148, 1994.

[For00] S. Fortune. Exact computation of the inertia of symmetric integer matrices. In *32th Symposium on Theory of Computing, STOC 2000*, pages 556–564. ACM Press, 2000.

[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.

[HJ85] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[HJ91] R. Horn and C. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991.

[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.

[HT01] T. M. Hoang and T. Thierauf. The complexity of the minimal polynomial. In *26th International Symposium, MFCS 2001*, pages 408–420. Springer, 2001.

[HT02] T. M. Hoang and T. Thierauf. The complexity of the characteristic and the minimal polynomial. Invited paper to the special issue in *Theoretical Computer Science* of the 26th MFCS conference 2001, to appear, 2002.

[Koz91] D. Kozen. *The Design and Analysis of Algorithms.* Springer-Verlag, 1991.

[Nef94] C. A. Neff. Specified precision root isolation is in NC. *Journal of Computer and System Science*, 48:429–463, 1994.

[NR96] C. A. Neff and J. H. Reif. An efficient algorithm for the complex roots problem. *Journal of Complexity*, 12:81–115, 1996.

[Ogi98] M. Ogihara. The PL hierarchy collapses. *SIAM Journal on Computing*, 27:1430–1437, 1998.

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

[Val79a] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[Val79b] L. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8:410–421, 1979.

[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.

[Vin91] V Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *6th IEEE Conference on Structure in Complexity Theory*, pages 270–284, 1991.