



Contents lists available at SciVerse ScienceDirect

Journal of Complexity

journal homepage: www.elsevier.com/locate/jco



On the complexity of the multivariate resultant

Bruno Grenet*, Pascal Koiran, Natacha Portier

LIP, UMR 5668 ÉNS Lyon – CNRS – UCBL – INRIA, École Normale Supérieure de Lyon, Université de Lyon, France

ARTICLE INFO

Article history:

Received 1 February 2012

Accepted 3 October 2012

Available online 13 October 2012

Keywords:

Polynomial system solving

Multivariate resultant

Complexity lower bounds

Implicit representations

Determinant

ABSTRACT

The multivariate resultant is a fundamental tool of computational algebraic geometry. It can in particular be used to decide whether a system of n homogeneous equations in n variables is satisfiable (the resultant is a polynomial in the system's coefficients which vanishes if and only if the system is satisfiable). In this paper, we investigate the complexity of computing the multivariate resultant.

First, we study the complexity of testing the multivariate resultant for zero. Our main result is that this problem is NP-hard under deterministic reductions in any characteristic, for systems of low-degree polynomials with coefficients in the ground field (rather than in an extension). In null characteristic, we observe that this problem is in the Arthur–Merlin class AM if the generalized Riemann hypothesis holds true, while the best known upper bound in positive characteristic remains PSPACE.

Second, we study the classical algorithms to compute the resultant. They usually rely on the computation of the determinant of an exponential-size matrix, known as Macaulay matrix. We show that this matrix belongs to a class of *succinctly representable* matrices, for which testing the determinant for zero is proved PSPACE-complete. This means that improving Canny's PSPACE upper bound requires either to look at the fine structure of the Macaulay matrix to find an *ad hoc* algorithm for computing its determinant, or to use altogether different techniques.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Given two univariate polynomials, their Sylvester matrix is a matrix built on the coefficients of the polynomials which is singular iff the polynomials have a common root. The determinant of the

* Corresponding author.

E-mail addresses: Bruno.Grenet@ens-lyon.fr (B. Grenet), Pascal.Koiran@ens-lyon.fr (P. Koiran), Natacha.Portier@ens-lyon.fr (N. Portier).

Sylvester matrix is known as the resultant of the polynomials. This determinant is easy to compute since the size of the Sylvester matrix is the sum of the degrees of the polynomials. The study of the possible generalizations to multivariate systems comes within the scope of elimination theory [49,38,15,47,16]. This theory proves that the only case where a unique polynomial can testify to the existence of a common root to the system is the case of n homogeneous polynomials in n variables: the resultant of a square system of homogeneous polynomials $f_1, \dots, f_n \in \mathbb{K}[x_1, \dots, x_n]$ is a polynomial in the indeterminate coefficients of f_1, \dots, f_n which vanishes if and only if f_1, \dots, f_n have a nonzero common root in the algebraic closure of \mathbb{K} . The resultant of such a system is known as the *multivariate resultant* in the literature. This captures the case of two univariate polynomials *via* their homogenization. Furthermore, in many cases a system of more than n homogeneous polynomials in n variables can be reduced to a system of n homogeneous polynomials, and so the square case is an important one. This result is sometimes known as Bertini's theorem. (As explained toward the end of this section, we will use an effective version of this result in one of our NP-hardness proofs.) In this paper, we focus on the multivariate resultant which we simply refer to as the resultant.

The resultant has been extensively used to solve polynomial systems [36,43,10,12] and for the elimination of quantifiers in algebraically or real-closed fields [44,24]. More recently, the multivariate resultant has been of interest in pure and applied domains. For instance, the problem of robot motion planning is closely related to the multivariate resultant [8,11], and more generally the multivariate resultant is used in real algebraic geometry [9,29]. Finally, in the domain of symbolic computation progress has been made for finding explicit formulations for the resultant [28,14,7,12,25], see also [26].

The aim of this paper is to contribute to the study of the complexity of the resultant. We both study the complexity of testing the resultant for zero, that is the satisfiability of a system of n homogeneous polynomials in n variables, and the complexity of explicitly (and exactly) computing the resultant.

A shorter version of this paper [22] has been published in the Proceedings of MFCS 2010. It contains material from Sections 2 and 3. Material of Section 4 appeared in preliminary form (and in French) in [21], but was not published so far.

1.1. Definitions and notations

In this paper, \mathbb{K} denotes any field and $\mathbb{K}[x_1, \dots, x_n]$ the ring of polynomials in n indeterminates over \mathbb{K} . The algebraic closure of \mathbb{K} is denoted by $\overline{\mathbb{K}}$. For a prime number p , the finite field with p elements is denoted by \mathbb{F}_p . The notation is extended to the characteristic zero by $\mathbb{F}_0 = \mathbb{Q}$.

A tuple (a_1, \dots, a_s) is denoted by a when there is no possible confusion on the range of the index. In particular, for a tuple of integers $\alpha = (\alpha_1, \dots, \alpha_n)$, x^α denotes the monomial $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$, and in the next definition, γ is the tuple of the $\gamma_{i,\alpha}$'s. The *total degree* of the monomial x^α is $|\alpha| = \alpha_1 + \dots + \alpha_n$.

Definition 1. Let \mathbb{K} be a field and f_1, \dots, f_n be n homogeneous polynomials in $\mathbb{K}[x_1, \dots, x_n]$, $f_i(x) = \sum_{|\alpha|=d_i} \gamma_{i,\alpha} x^\alpha$. The *multivariate resultant* R of f_1, \dots, f_n is an irreducible polynomial in $\mathbb{K}[\gamma]$ such that

$$R(\gamma) = 0 \iff \exists x \in \overline{\mathbb{K}}^n, \quad x \neq \bar{0}, \quad f_1(x) = \dots = f_n(x) = 0. \quad (1)$$

The multivariate resultant is unique up to a constant factor.

The existence of the multivariate resultant is not evident. We refer to [49,35] for a proof of this fact. The uniqueness comes from the fact that two irreducible polynomials having the same roots are equal up to a constant factor.

The first problem we are interested in is testing the resultant for zero. This is the same as deciding whether a square system of homogeneous polynomials (that is n polynomials in n variables) has a non-trivial root. This is closely related to the decision problem for the existential theory of an algebraically closed field. This problem is sometimes called the *Hilbert Nullstellensatz* problem:

Definition 2. Let \mathbb{K} be a field. The *Hilbert Nullstellensatz* problem over \mathbb{K} , $\text{HN}(\mathbb{K})$, is the following: Given a system f of s polynomials in $\mathbb{K}[x_0, \dots, x_n]$, does there exist a root of f in $\overline{\mathbb{K}}^{n+1}$?

Let us now assume that the s components of f are homogeneous polynomials. Then the *homogeneous* Hilbert Nullstellensatz problem over \mathbb{K} , $H_2N(\mathbb{K})$, is to decide whether a *non trivial* (that is, nonzero) root exists in $\overline{\mathbb{K}}^{n+1}$.

If f is supposed to contain as many homogeneous polynomials as variables, the problem is called the *Resultant* over \mathbb{K} , $RESULTANT(\mathbb{K})$.

In the case of the field \mathbb{Q} , it is more natural to have coefficients in \mathbb{Z} . We shall use the notations HN , H_2N and $RESULTANT$ for the case where the system is made of integer polynomials.

In the case of polynomials with coefficients in \mathbb{Z} , Canny [8] gave in 1987 a PSPACE algorithm to compute the resultant. To the authors' knowledge, this is the best known upper bound. In this paper we show that testing the resultant for zero is NP-hard in any characteristic. In other words, $RESULTANT(\mathbb{K})$ is NP-hard for any field \mathbb{K} .

Main results and proof techniques

Sections 2 and 3 are devoted to the study of the decision problem, first in null characteristic and then in any characteristic. Section 4 focuses on the evaluation problem.

In Section 2.1 we observe that for polynomials with integer coefficients, testing the resultant for zero is a problem in the Arthur–Merlin (AM) class. This result assumes the Generalized Riemann Hypothesis (GRH), and follows from a simple reduction to the Hilbert Nullstellensatz. For this problem, membership in AM assuming GRH was established in [30].

In characteristic zero, it seems to be a “folklore” result that testing the resultant for zero is NP-hard, and one can find a proof of this fact in [23]. A similar result can be obtained by considering a system of two homogeneous polynomials, but given in lacunary representation (their degree can therefore be exponential in the input size). This result of incomparable strength is a reformulation of a theorem of Plaisted [42]. We give the proofs of these two results to be able to argue about their irrelevance in positive characteristic.

The first proof does not carry over to positive characteristic since it is a reduction from the problem PARTITION [20, Problem SP12] whose NP-hardness relies in an essential way on the fact that the data are integers (in fact, in any finite field the analogue problem can be solved in polynomial time by dynamic programming).

Plaisted's result can be adapted to positive characteristic [50,27] but this requires randomization. By contrast, our ultimate goal is NP-hardness for deterministic reductions and low degree polynomials. We therefore need to use different techniques. Our starting point is a fairly standard encoding of 3-SAT by systems of polynomial equations. Using this encoding we show at the beginning of Section 3 that deciding the existence of a nontrivial solution to a system of homogeneous equations is NP-hard in any characteristic. The resulting system has in general more equations than variables. In order to obtain a square system we explore these two basic strategies:

- (i) Decrease the number of equations.
- (ii) Increase the number of variables.

In Section 3.1 we give a randomized NP-hardness result based on the first strategy. The idea is to replace the initial system by a random linear combinations of the system's equations (the fact this does not change the solution set is sometimes called a “Bertini theorem”).

In Section 3.2 we use the second strategy to obtain two NP-hardness results for deterministic reductions. The main difficulty is to make sure that the introduction of new variables does not create spurious solutions (we do not want to turn an unsatisfiable system into a satisfiable system). Our solution to this problem can be viewed as a derandomization result. Indeed, it can be shown that the coefficients of the monomials where the new variables occur could be chosen at random. It would be interesting to find out whether the proof based on the first strategy can also be derandomized.

Section 4 is devoted to the study of the Macaulay matrices which are used in several formulations of the resultant. These matrices have an exponential-size (in the number of variables and the degrees of the polynomials) but can be efficiently represented by a polynomial-size boolean circuit. This notion of *circuit representation* goes back to [19] and was developed for graphs. We prove that computing the determinant (or testing it for zero) of a matrix which is given by a circuit is PSPACE-complete in any characteristic. This shows that to improve the best known upper bound on the

computation of the resultant in positive characteristic, one needs either to use the fine structure of the Macaulay matrices, or to find an altogether different method. So far, this has been achieved only in characteristic 0 [30].

2. Complexity of the resultant in null characteristic

2.1. The resultant lies in AM

In this section we show that testing the resultant for zero is reducible to $\text{HN}(\mathbb{K})$. In the case $\mathbb{K} = \mathbb{Z}$, this allows us to conclude (under the Generalized Riemann Hypothesis) that our problem is in the polynomial hierarchy, and more precisely in the Arthur–Merlin class. In fact, we show that this applies more generally to the satisfiability problem for homogeneous systems (recall that testing the resultant for zero corresponds to the square case).

Proposition 1. *For any field \mathbb{K} , the problem $\text{H}_2\text{N}(\mathbb{K})$ is polynomial-time many-one reducible to $\text{HN}(\mathbb{K})$.*

Proof. Consider an instance f of $\text{H}_2\text{N}(\mathbb{K})$, that is s homogeneous polynomials f_1, \dots, f_s in $\mathbb{K}[x_1, \dots, x_n]$. The polynomials f_1, \dots, f_s can be viewed as elements of $\mathbb{K}[x_1, \dots, x_n, y_1, \dots, y_n]$ where y_1, \dots, y_n are new variables which do not appear in the f_i 's. Let g be the system containing all the f_i 's and the new (non-homogeneous) polynomial $\sum_{i=1}^n x_i y_i - 1$. This is an instance of the problem $\text{HN}(\mathbb{K})$. It remains to prove that f and g are equivalent.

Given a root $(a_1, \dots, a_n, b_1, \dots, b_n)$ of g , the new polynomial ensures that there is at least one nonzero a_i . So (a_1, \dots, a_n) is a non trivial root of f . Conversely, suppose that f has a non trivial root (a_1, \dots, a_n) , and let i be such that $a_i \neq 0$. Then the tuple $(a_1, \dots, a_n, 0, \dots, 0, a_i^{-1}, 0, \dots, 0)$ where a_i^{-1} corresponds to the variable y_i is a root of g .

Thus $\text{H}_2\text{N}(\mathbb{K})$ is polynomial-time many-one reducible to $\text{HN}(\mathbb{K})$. \square

Koiran [30] proved that $\text{HN} \in \text{AM}$ under the Generalized Riemann Hypothesis. We denote here by AM the Arthur–Merlin class, defined by *interactive proofs with public coins* (see [2]). Thereby,

Corollary 1. *Under the Generalized Riemann Hypothesis, H_2N is in the class AM.*

In positive characteristic, the best upper bound on the complexity of the Hilbert Nullstellensatz known to this day remains PSPACE (in particular it is not known whether the problem lies in the polynomial hierarchy, even assuming some plausible number-theoretic conjecture such as the generalized Riemann hypothesis).

2.2. The resultant is NP-hard in characteristic 0

We now give a first NP-hardness result, for the satisfiability of square systems of homogeneous polynomial equations. The first part of the theorem can be easily deduced from [23, Proposition 10]. The second part shows that the problems remains NP-hard even for systems with small integer coefficients (i.e., coefficients bounded by 2). This is achieved by a standard trick: we introduce new variables in order to “simulate” large integers coefficients. It is interesting to note, however, that a similar trick for reducing degrees does not seem to apply to the resultant problem (more on this after Theorem 2).

Theorem 1. *The problem RESULTANT of deciding whether a square system of homogeneous polynomials with coefficients in \mathbb{Z} has a non trivial root is NP-hard.*

The problem remains NP-hard even if no polynomial has degree greater than 2 and even if the coefficients are bounded by 2.

Proof. The reduction is done from PARTITION which is known to be NP-hard [20, Problem SP12]: Given a finite set S and a non negative integer weight $w(s)$ for each $s \in S$, the problem is to decide the

existence of subset S' such that $\sum_{s \in S'} w(s) = \sum_{s \notin S'} w(s)$. That is, the aim is to cut A into two subsets of same weights.

Given such an instance of PARTITION where $S = \{s_1, \dots, s_n\}$, let us define a system of polynomials. For $1 \leq i \leq n$, $f_i(x) = x_0^2 - x_i^2$. And

$$f_0(x_0, \dots, x_n) = w(s_1)x_1 + w(s_2)x_2 + \dots + w(s_n)x_n. \quad (2)$$

A tuple (a_0, \dots, a_n) has to verify $a_i = \pm a_0$ for each i to be a solution, hence the only case to consider is $a_0 = 1$ and $a_i = \pm 1$ for $i \geq 1$. Then it is clear that the system has a non trivial solution if and only if S may be split into two subsets of equal weights.

For the second part of the theorem, it remains to show that the coefficients in the system can be bounded by 2. As the $w(s_i)$'s may be large integers, they have to be replaced by variables. Let us write $w(s_i) = \sum_{j=0}^p w_{ij}2^j$. For each w_{ij} , a new variable W_{ij} is introduced. For every i , the values of the W_{ij} 's are defined by a descending recurrence:

$$\begin{cases} W_{ip} - w_{ip}x_0 = 0 \\ W_{ij} - (2W_{i,j+1} + w_{ij}x_0) = 0 \quad \text{for all } j < p. \end{cases} \quad (3)$$

These equalities imply that for every i and j we have $W_{ij} = \sum_{l=j}^p w_{il}2^{l-j}x_0$. Then f_0 is replaced by $W_{1,0}x_1 + W_{2,0}x_2 + \dots + W_{n,0}x_n$. Doing so, the number of polynomials remains the same as the number of variables. Hence, this algorithm build a new homogeneous system where the polynomials have their coefficients bounded by 2 and their degrees too. One can readily check that the new system has a non trivial solution if and only if the original one has. In particular, if x_0 is set to zero, then all other variables have to be set to zero too. \square

A related result is Plaisted's [42] on the NP-hardness of deciding whether the gcd of two sparse univariate polynomials has degree greater than one. By homogenization of the polynomials, this is the same problem as in Theorem 1 for only two bivariate polynomials. Note that the polynomials are sparse and can be of very high degree since exponents are written in binary (this polynomial representation is sometimes called "supersparse" or "lacunary" [27]). If both polynomials were dense, the resultant could be computed in polynomial time since it is equal to the determinant of their Sylvester matrix. Plaisted's theorem stated in the same language as Theorem 1 is the following:

Theorem 2. *Given two sparse homogeneous polynomials in $\mathbb{Z}[x, y]$, it is NP-hard to decide whether they share a common root in \mathbb{C}^2 .*

We briefly sketch Plaisted's reduction since it will help understand the discussion at the end of this section. For a full proof (including a correctness proof), see [42, Theorem 5.1].

Proof sketch. The idea is to turn a 3-SAT instance into a system of two univariate polynomials which share a common root if and only if the 3-CNF formula is satisfiable.

To every variable X_j is associated a prime p_j , and let $M = \prod_i p_j$ where the product ranges over all the variables that appear in the formula. A formula ϕ is turned into a polynomial P_ϕ according to the following rules. A non negated variable X_j is turned into $P_{X_j}(x) = x^{M/p_j} - 1$ and a negated variable $\neg X_k$ into $P_{\neg X_k}(x) = 1 + x^{M/p_k} + \dots + x^{(p_k-1)M/p_k}$. Then a formula $\phi \vee \psi$ is turned into $P_{\phi \vee \psi} = \text{lcm}(P_\phi, P_\psi)$. A conjunction $\phi = \bigwedge_i \phi_i$ is turned into the polynomial

$$P_\phi(x) = x^M \sum_i P_{\phi_i}(x) P_{\phi_i}(1/x). \quad (4)$$

This defines the first polynomial P . The second polynomial is simply $x^M - 1$. The proof that those two polynomials share a common root if and only if ϕ is satisfiable is omitted.

To obtain the result in the way we stated it, it is sufficient to homogenize $P(x)$ and $x^M - 1$ with the second variable y . \square

Theorems 1 and **2** seem to be incomparable. In particular, it is not clear how to derive **Theorem 1** from **Theorem 2**. A natural idea would be to introduce new variables and use the repeated squaring trick to reduce the degrees of the polynomials occurring in Plaisted's result. However, as we now explain this can lead to the creation of unwanted roots at infinity in the resulting polynomial system.

Assume for instance that we wish to get rid of all occurrences of x^2 in a polynomial. One can add a new variable x_2 , replace the occurrences of x^2 by x_2 and add a new polynomial $x_2 - x^2$. In order to keep the system homogeneous, the idea is to homogenize the latter polynomial: $x_0x_2 - x^2$. The problem with this technique is that it adds some new roots with all variables but x_2 set to 0, and in particular the homogenization variable x_0 .

To give an explicit example of the problem mentioned above, let us consider the formula

$$(X \vee Y) \wedge (\neg X) \wedge (\neg Y). \quad (5)$$

Let us associate the prime number 2 to the variable X , and 3 to Y (M in the previous proof is therefore 6). By Plaisted's construction, X is turned into $x^{M/2} - 1 = x^3 - 1$ and Y into $x^2 - 1$. Their negations $\neg X$ and $\neg Y$ are respectively turned into $1 + x^3$ and $1 + x^2 + x^4$. The disjunction of X and Y is turned into the lcm of $x^3 - 1$ and $x^2 - 1$, that is $(x^2 - 1)(x^2 + x + 1)$. Finally, we have to apply formula (4) with the latter polynomial, $1 + x^3$ and $1 + x^2 + x^4$. Therefore, the two polynomials of Plaisted's construction are $x^M - 1 = x^6 - 1$ and $-x^3 + x^4 + 2x^5 + 9x^6 + 2x^7 + x^8 - x^9$. It can be checked that as expected, those two polynomials do not share any common root.

Applying the repeated squaring trick with homogenization on this example gives the following system where the two first polynomials represent the original ones and the other ones are new ones:

$$\begin{cases} -x_3 + x_4 + 2x_5 + 9x_6 + 2x_7 + x_8 - x_9 = 0 \\ x_6 - x_0 = 0; & x_0x_2 - x^2 = 0; & x_0x_3 - x_2x = 0 \\ x_0x_4 - x_2^2 = 0; & x_0x_5 - x_4x = 0; & x_0x_6 - x_2x_4 = 0 \\ x_0x_7 - x_4x_3 = 0; & x_0x_8 - x_4^2 = 0; & x_0x_9 - x_8x = 0. \end{cases} \quad (6)$$

But in that example, one can easily check that solutions with $x_0 = 0$ exist. Namely if we set x_8 and x_9 to the same nonzero value and all other variables to 0, this defines a solution to the system.

To the authors' knowledge, there is no solution to avoid these unwanted roots. Furthermore, Plaisted's result works well with fields of characteristic 0, but as it uses the fact that a sum of non negative terms is zero if and only if every term is zero, this generalizes not so well to positive characteristic. In particular, generalizations to positive characteristic require randomization (see [27,50]). By contrast, two of the reductions given in the next section are deterministic and they yield systems with polynomials of low degree (i.e., of linear or even constant degree).

3. NP-hardness in arbitrary characteristic

In this section we give three increasingly stronger NP-hardness results for testing the resultant. As explained in the introduction, we first provide in Section 3.1 a NP-hardness proof for randomized reductions. We then give in Section 3.2 two NP-hardness results for deterministic reductions: the first one applies to systems with coefficients in an extension of the ground field, and the second (stronger) result to systems with coefficients in the ground field only. The starting point for these three NP-hardness results is the following easy lemma.

Lemma 1. *Given a field \mathbb{K} of any characteristic, it is NP-hard to decide whether a system of s homogeneous polynomials in $\mathbb{K}[x_0, \dots, x_n]$ has a non trivial root. That is, $H_2N(\mathbb{K})$ is NP-hard.*

In [31], this lemma is proved using a reduction from the language **BOOLSYS**. An input of **BOOLSYS** is a system of boolean equations in the variables X_1, \dots, X_n where each equation is of the form $X_i = \text{True}$, $X_i = \neg X_j$, or $X_i = X_j \vee X_k$. The question is the existence of a valid assignment for the system, that is an assignment of the variables such that each equation is satisfied. This problem is easily shown NP-hard by reduction from 3-SAT. We now give a proof of this lemma since the specific form of the systems that we construct in the reduction will be useful in the sequel. This proof is a slight variation on the proof from [31].

Proof. Let \mathbb{K} be a field of any characteristic p , p being either zero or a prime number. At first, p is supposed to be different from 2. The proof has to be slightly changed in the case $p = 2$ and this case is explained at the end of the proof.

Let \mathcal{B} be an instance of **BOOLSYS**. Let us define a system of homogeneous polynomials from this instance with the property that \mathcal{B} is satisfiable if and only if the polynomial system has a non trivial common root. The variables in the system are x_0, \dots, x_n where x_i , $1 \leq i \leq n$, corresponds to the boolean variable X_i in **BOOLSYS**, and x_0 is a new variable. The system contains four kinds of polynomials:

- $x_0^2 - x_i^2$, for each $i > 0$;
- $x_0 \cdot (x_i + x_0)$, for each equation $X_i = \text{True}$;
- $x_0 \cdot (x_i + x_j)$, for each equation $X_i = \neg X_j$;
- $(x_i + x_0)^2 - (x_j + x_0) \cdot (x_k + x_0)$, for each equation $X_i = X_j \vee X_k$.

Let us denote by f the polynomial system obtained from \mathcal{B} . The first kind of polynomials ensures that if (a_0, \dots, a_n) is a non trivial root of f , then $a_0^2 = a_1^2 = \dots = a_n^2$. Now if f has a non trivial root (a_0, \dots, a_n) , then one can readily check that the assignment $X_i = \text{True}$ if $a_i = -a_0$ and $X_i = \text{false}$ if $a_i = a_0$ satisfies \mathcal{B} . Conversely, if there is a valid assignment X_1, \dots, X_n for \mathcal{B} , any $(n+1)$ -tuple (a_0, \dots, a_n) where $a_0 \neq 0$ and $a_i = -a_0$ if $X_i = \text{True}$ and $a_i = a_0$ if $X_i = \text{false}$ is a non trivial root of f .

This proof works for any field of characteristic different from 2. The problem in characteristic 2 is the implementation of **BOOLSYS** in terms of a system of polynomials. Indeed, for the other characteristics, the truth is represented by $-a_0$ and the falseness by a_0 . In characteristic 2, those values are equal. Yet, one can just change the polynomials and define in the case of characteristic 2 the following system:

- $x_0 x_i - x_i^2$, for each $i > 0$;
- $x_0(x_i + x_0)$, for each equation $X_i = \text{True}$;
- $x_0(x_i + x_j + x_0)$, for each equation $X_i = \neg X_j$;
- $x_i^2 + x_j x_k + x_0 \cdot (x_j + x_k)$, for each equation $X_i = X_j \vee X_k$.

Now, given any nonzero value a_0 for x_0 , the truth of a variable X_i is represented by $x_i = a_0$ whence the falseness is represented by $x_i = 0$. A root of the system is in particular a root of the polynomials defined by the first item. Therefore each x_i has to be set either to a_0 or to 0. The system has a non trivial root if and only if the instance of **BOOLSYS** is satisfiable. \square

3.1. A randomized reduction

We now give the first of our three NP-completeness results in positive characteristic. The proof also applies to characteristic zero, but in this case [Theorem 1](#) is preferable (its proof is simpler and the NP-hardness result stronger since it relies on deterministic reductions). For more on randomized reductions, see [\[2\]](#).

We begin with a result on algebraic varieties in algebraic closed fields. This is a classical result in algebraic geometry [\[45\]](#), see also [\[32, Proposition 1\]](#) for a proof in our language.

Theorem 3. Let \mathbb{K} be an algebraically closed field and V an algebraic variety of \mathbb{K}^{n+1} defined by a set of homogeneous degree- d polynomials $f_1, \dots, f_s \in \mathbb{K}[x_0, \dots, x_n]$. This variety can be defined by $(n+1)$ homogeneous degree- d polynomials $g_1, \dots, g_{n+1} \in \mathbb{K}[x_0, \dots, x_n]$. Moreover, suitable g_i 's can be obtained by taking generic linear combinations of the f_i . That is, we can take $g_i = \sum_{j=1}^s \alpha_{ij} f_j$ where (α_{ij}) is a matrix of elements of \mathbb{K} , and the set of suitable matrices is Zariski-dense in $\mathbb{K}^{s(n+1)}$.

This result leads us to our first NP-hardness result in arbitrary characteristic.

Theorem 4. Let p be either zero or a prime number. The following problem is NP-hard under randomized reductions:

- Input: A square system of homogeneous equations with coefficients in a finite extension of \mathbb{F}_p .
- Question: Is the system satisfiable in the algebraic closure of \mathbb{F}_p ?

In the case $p = 0$, the results also holds for systems with coefficients in \mathbb{Z} .

Proof. Lemma 1 shows that it is NP-hard to decide whether a non square polynomial system f with coefficients in \mathbb{F}_p has a non trivial root. From f , a square system g is built in randomized polynomial time.

Let us denote by f_j , $1 \leq j \leq s$, the components of f . They are homogeneous polynomials in $\mathbb{F}_p[x_0, \dots, x_n]$. The components of g are defined by

$$g_i = \sum_{j=1}^s \alpha_{ij} f_j \quad (7)$$

for $0 \leq i \leq n$. In the sequel, we explain how to choose the α_{ij} 's for f and g to be equivalent. For any choice of the α_{ij} 's a root of f is a root of g . Thus it is sufficient to show how to choose them so that g has no non trivial root if the same is true for f .

The property the α_{ij} 's have to satisfy is expressed by the first-order formula

$$\Phi(\alpha) \equiv \forall x_0 \dots \forall x_n \left(\bigwedge_{j=1}^s f_j(x) = 0 \right) \vee \left(\bigvee_{i=0}^n \sum_{j=1}^s \alpha_{ij} f_j(x) \neq 0 \right). \quad (8)$$

The formula Φ belongs to the language of the first-order theory of the algebraically closed field $\overline{\mathbb{F}_p}$. This theory eliminates quantifiers and $\Phi(\alpha)$ is therefore equivalent to a quantifier-free formula of the form

$$\Psi(\alpha) \equiv \bigvee_k \left(\bigwedge_l P_{kl}(\alpha) = 0 \wedge \bigwedge_m Q_{km}(\alpha) \neq 0 \right), \quad (9)$$

where $P_{kl}, Q_{km} \in \mathbb{F}_p[\alpha]$. As a special case of [18, Theorem 2], one can bound the number of polynomials in Ψ as well as their degrees by $2^{\text{poly}(n, \log(s+n))}$ where poly represents some polynomial independent from Φ .

Theorem 3 shows that the set A of tuples satisfying Φ is Zariski-dense in $\overline{\mathbb{F}_p}^{s(n+1)}$. Since A is dense, and A is also defined by Ψ , one of the clauses of Ψ must define a Zariski dense subset of $\overline{\mathbb{F}_p}^{s(n+1)}$. This clause is of the form $\bigwedge_m Q_m(\alpha) \neq 0$.

To satisfy Φ , it is sufficient for the α_{ij} 's to avoid the roots of a polynomial $Q = \prod_m Q_m$. As mentioned before, it is known that Ψ contains at most $2^{\text{poly}(n, \log(s+n))}$ polynomials of degree at most $2^{\text{poly}(n, \log(s+n))}$. Thus, Q is a polynomial of degree at most $2^{2\text{poly}(n, \log(s+n))}$. Consider now a finite extension \mathbb{K} of \mathbb{F}_p with at least $2^{2+2\text{poly}(n, \log(s+n))}$ elements (that is, of polynomial degree). If we choose the α_{ij} 's uniformly at random in \mathbb{K} , then with probability at least $3/4$ they are not a root of Q (by the Schwartz–Zippel Lemma). Thus with the same probability, they satisfy Φ . Note that \mathbb{K} can be built in polynomial-time with Shoup's algorithm [46] when p is prime (for $p = 0$, we take of course $\mathbb{K} = \mathbb{Q}$).

To sum up, we build from f a square system g defined by random linear combinations of the components of f . If f has a non trivial root, then it is a root of g too. Conversely, if f has no non trivial root, then with probability at least $3/4$ it is also the case that g has no nontrivial root. \square

In characteristic zero the bounds in the above proof can be sharpened: instead of appealing to the general-purpose quantifier elimination result of [18] we can use a result of [34]. Indeed, it follows from Section 4.1 of [34] that there exists a polynomial F of degree at most 3^{n+1} such that $F(\alpha) \neq 0$ implies that g has no non trivial root as soon as it is true for f . This polynomial plays the same role as Q in the previous proof but the bound on its degree is sharper.

3.2. Deterministic reductions

We now improve the NP-hardness result of Section 3.1: we show that the same problem is NP-hard for deterministic reductions. This result is not only stronger, but also the proof is more elementary (there is no appeal to effective quantifier elimination).

Recall from the introduction that for a field \mathbb{K} , $\text{RESULTANT}(\mathbb{K})$ is the following problem:

- *Input:* A square system of homogeneous equations with coefficients in \mathbb{K} .
- *Question:* Is the system satisfiable in the algebraic closure of \mathbb{K} ?

Theorem 5. *Let p be either zero or a prime number. There exists a finite extension \mathbb{K} of \mathbb{F}_p such that $\text{RESULTANT}(\mathbb{K})$ is NP-hard.*

In the case $p = 0$, this result also holds for systems with coefficients in \mathbb{Z} .

Proof. The proof of Lemma 1 gives a method to implement an instance of Boolsys with a system f of s homogeneous polynomials in $n + 1$ variables with coefficients in \mathbb{F}_p . It remains to explain how to construct a square system g that has a non trivial root if and only if f does. Let us denote by f_1, \dots, f_s the components of f , with for each $i = 1, \dots, n$, $f_i = x_0^2 - x_i^2$ if $p \neq 2$ and $f_i = x_0 x_i - x_i^2$ if $p = 2$. A new system g of s polynomials in s variables is built. The s variables are x_0, \dots, x_n and y_1, \dots, y_{s-n-1} , that is $(s - n - 1)$ new variables are added. The system g is the following:

$$g(x, y) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \\ f_{n+1}(x) & + \lambda y_1^2 \\ f_{n+2}(x) & - y_1^2 & + \lambda y_2^2 \\ \vdots \\ f_{n+i}(x) & - y_{i-1}^2 & + \lambda y_i^2 \\ \vdots \\ f_{s-1}(x) & - y_{s-n-2}^2 & + \lambda y_{s-n-1}^2 \\ f_s(x) & - y_{s-n-1}^2 \end{pmatrix}. \quad (10)$$

The parameter λ is to be defined later. Clearly, if f has a non trivial root a , then $(a, 0, \dots, 0)$ is a non trivial root of g . Let us now prove that the converse also holds true for some λ : if g has a non trivial root, then so does f . Note that a suitable λ has to be found in polynomial time.

Let $(a_0, \dots, a_n, b_1, \dots, b_{s-n-1})$ be any non trivial root of g . In particular, $f_1(a) = \dots = f_n(a) = 0$. Hence $a_0^2 = \dots = a_n^2$ if $p \neq 2$, and $a_i \in \{0, a_0\}$ for every i if $p = 2$. Now, either $a_0 = 0$ and $f_i(a) = 0$ for every i , or a_0 can be supposed to equal 1. Therefore, if $p \neq 2$ either $a = \bar{0}$ or $a_i = \pm 1$ for every i , and if $p = 2$ either $a = \bar{0}$ or $a_i \in \{0, 1\}$ for every i . Let us define $\epsilon_i = f_{n+i}(a) \in \mathbb{F}_p$. As (a, b) is a root of g , the b_i^2 's satisfy the linear system

$$\begin{cases} \epsilon_1 & & + \lambda Y_1 & = 0, \\ \epsilon_2 & - Y_1 & + \lambda Y_2 & = 0, \\ & & \vdots & \\ \epsilon_{s-n-1} & - Y_{s-n-2} & + \lambda Y_{s-n-1} & = 0, \\ \epsilon_{s-n} & - Y_{s-n-1} & & = 0. \end{cases} \quad (11)$$

This system can be homogenized by replacing each ϵ_i by $\epsilon_i Y_0$ where Y_0 is a fresh variable. This gives a square homogeneous linear system. The determinant of the matrix of this system is equal to

$$(-1)^{s-n-1} (\epsilon_1 + \epsilon_2 \lambda + \dots + \epsilon_{s-n} \lambda^{s-n-1}).$$

Let us consider this determinant as a polynomial in λ . This polynomial vanishes identically if and only if all the ϵ_i 's are zero. In that case, the only solutions satisfy $Y_i = 0$ for $i > 0$, that is $(a, \bar{0})$ is a

root of g and therefore a is a root of f . If some ϵ_i 's are nonzero, this is a nonzero polynomial of degree $(s - n - 1)$. If λ can be chosen such that it is not a root of this polynomial (for any possible nonzero value of $\bar{\epsilon}$), then the only solution to the linear system is the trivial one. This means that the only non trivial root of g is $(a, \bar{0})$ where a is a root of f .

If the polynomials have coefficients in \mathbb{Z} , $\lambda = 3$ (or any other integer $\lambda > 2$) satisfies the condition. Indeed, one can check that $\epsilon_i = f_{n+i}(a) \in \{-4, 0, 2, 4\}$ when $a_0 = 1$. The determinant is zero if and only if $\epsilon'_1 + \epsilon'_2\lambda + \dots + \epsilon'_{s-n}\lambda^{s-n-1} = 0$ where $\epsilon'_i = \epsilon_i/2 \in \{-2, 0, 1, 2\}$. For each i , let $\epsilon_i^+ = \max\{\epsilon'_i, 0\}$ and $\epsilon_i^- = \max\{-\epsilon'_i, 0\}$. Then $\epsilon'_i = \epsilon_i^+ - \epsilon_i^-$, and $0 \leq \epsilon_i^+, \epsilon_i^- \leq 2$. Now the determinant is zero if and only if $\sum_i \epsilon_i^+ 3^i = \sum_i \epsilon_i^- 3^i$. By the unicity of base-3 representation, this means that for all i , $\epsilon_i^+ = \epsilon_i^-$, and so $\epsilon'_i = 0$.

For a field of positive characteristic, this argument cannot be applied. The idea is to find a λ that is not a root of any polynomial of degree $(s - n - 1)$. Nothing else can be supposed on the polynomial because if $p = 3$ for example, any polynomial of $\mathbb{F}_3[\lambda]$ can appear. This also shows that λ cannot be found in the ground field. Suppose an extension of degree $(s - n)$ is given as $\mathbb{F}_p[X]/(P)$ where P is an irreducible degree- $(s - n)$ polynomial with coefficients in \mathbb{F}_p . Then a root of P in $\mathbb{F}_p[X]/(P)$ cannot be a root of a degree- $(s - n - 1)$ polynomial with coefficients in \mathbb{F}_p . Thus, if one can find such a P , taking for λ the indeterminate X is sufficient. For any fixed characteristic p , Shoup gives a deterministic polynomial-time algorithm [46] that given an integer N outputs a degree- N irreducible polynomial P in $\mathbb{F}_p[X]$. Thus, the system g is now a square system of polynomials in $(\mathbb{F}_p[X]/(P)) [x, y]$ and this system has a non trivial root if and only if f has a non trivial root. And Shoup's algorithm allows us to build g in polynomial time from f .

For any field \mathbb{F}_p , it has been shown that from an instance \mathcal{B} of **BOOLSYS** a square system g of polynomials with coefficients in an extension of \mathbb{F}_p (in \mathbb{Z} for integer polynomials) can be built in deterministic polynomial time such that g has a non trivial root if and only if \mathcal{B} is satisfiable. This shows that the problem is NP-hard. \square

The previous result is somewhat unsatisfactory as it requires, in the case of positive characteristic, to work with coefficients in an extension field rather than in the ground field. A way to get rid of this limitation is now shown. Yet, a property of the previous result is lost. Instead of having constant-degree (even degree-2) polynomials, our next result uses linear-degree polynomials. It is not clear whether the same result can be obtained for degree-2 polynomials (for instance, as explained at the end of Section 2.2 reducing the degree by introducing new variables can create unwanted solutions at infinity).

The basic idea behind Theorem 6 is quite simple (we put the irreducible polynomial used to build the extension field into the system), but some care is required in order to obtain an equivalent homogeneous system.

Theorem 6. For any prime p , $\text{RESULTANT}(\mathbb{F}_p)$ is NP-hard under deterministic reductions.

Proof. The idea for this result is to turn coefficient λ in the previous proof into a variable and to add the polynomial P as a component of the system. Of course, considering λ as a variable implies that the polynomials are not homogeneous anymore. Thus, it remains to explain how to keep the system homogeneous.

First, the polynomial P needs to be homogenized. This is done through the variable x_0 in the canonical way. As $P(\lambda)$ is irreducible, it is in particular not divisible by λ . Hence, the homogenized polynomial $P(\lambda, x_0)$ contains a monomial $\alpha\lambda^d$ and another one βx_0^d where d is the degree of P . Hence x_0 is zero if and only if λ is.

The other polynomials have the form $f_{n+i}(x) - y_{i-1}^2 + \lambda y_i^2$. It is impossible to homogenize those polynomials by multiplying f_{n+i} and y_{i-1}^2 by x_0 (or any other variable) because then the variable y_{i-1} never appears alone in a monomial, and a s -tuple with all variables set to 0 but y_{i-1} would be a non trivial solution. Moreover, in the previous proof, the fact that the y_i 's all appear with degree 2 is used to consider the system as a linear system in the y_i^2 . Thus replacing the monomial λy_i^2 by λy_i does not

work either. Instead, we construct the slightly more complicated homogeneous system:

$$g_h(x, y, \lambda) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \\ x_0^{s-n-1}f_{n+1}(x) & & +\lambda y_1^{s-n} \\ x_0^{s-n-2}f_{n+2}(x) & -y_1^{s-n} & +\lambda y_2^{s-n-1} \\ & \vdots & \\ x_0^{s-n-i}f_{n+i}(x) & -y_{i-1}^{s-n-i+2} & +\lambda y_i^{s-n-i+1} \\ & \vdots & \\ x_0f_{s-1}(x) & -y_{s-n-2}^3 & +\lambda y_{s-n-1}^2 \\ f_s(x) & -y_{s-n-1}^2 & \\ P(\lambda, x_0) & & \end{pmatrix}. \quad (12)$$

Contrary to the previous proof, the y_i 's do not appear all at the same power. Yet, all the occurrences of each y_i have the same degree, and we shall prove that this is sufficient.

Let us prove that if f does not have any non trivial root, then neither does g_h . Some of the observations made for g in the previous proof remain valid. Hence, it is sufficient to prove that a non trivial $(s+1)$ -tuple (a, b, ℓ) cannot be solution of g_h whenever $a_0 = 1$, $b \neq \bar{0}$ and $a_0^2 = \dots = a_n^2$ if $p \neq 2$ or $a_i \in \{0, a_0\}$ if $p = 2$. By a previous remark on the polynomial P , ℓ can also be supposed to be nonzero.

So, similarly as in the previous proof, let us define $\epsilon_i = a_0^{s-n-i}f_{n+i}(a) \in \mathbb{F}_p$. In the system g_h , the variable y_i only appears at the power $(s-n-i+1)$. Therefore, given a value of a and ℓ , the tuple (a, b, ℓ) is a root of g_h if and only if the $b_i^{s-n-i+1}$'s satisfy the linear system

$$\begin{cases} \epsilon_1 & & + \ell Y_1 & = 0 \\ \epsilon_2 & - Y_1 & + \ell Y_2 & = 0 \\ & & \vdots & \\ \epsilon_{s-n-1} & - Y_{s-n-2} & + \ell Y_{s-n-1} & = 0 \\ \epsilon_{s-n} & - Y_{s-n-1} & & = 0. \end{cases} \quad (13)$$

This is the same system as in the previous proof. Now if $(\ell, 1)$ is supposed to be a root of P , as P is an irreducible polynomial of degree $(s-n)$, ℓ cannot be a root of a univariate polynomial of degree less than $(s-n)$ with coefficient in \mathbb{F}_p . But the determinant of the linear system is such a polynomial, and thus cannot be zero. This determinant is then 0 if and only if all the $\epsilon_i = 0$. The same arguments as in the previous proof can be used to conclude that (a, b, ℓ) can be a root of g_h if and only if a is a root of f .

Thus, from an instance \mathcal{B} of **Boolsys**, a square homogeneous system g_h of polynomials with coefficients in the ground field \mathbb{F}_p is built in deterministic polynomial time. This system has a non trivial root if and only if \mathcal{B} is satisfiable. The result is proved. \square

4. Macaulay matrices

There exist several formulations of the multivariate resultant. More precisely, there are quite a lot of determinant-based formulations but they give in certain cases only multiple of the resultant: It can happen that the determinant that is computed vanishes even though the polynomial system has no root [28, 14, 7, 12, 25]. Some exact formulations exist based on Macaulay matrices. The resultant can be computed as the gcd of the determinants of several Macaulay matrices (viewed as polynomials) [8], or as the quotient of two determinants of Macaulay matrices [49]. Note also that this quotient can actually be turned into a single determinant using Strassen's method to eliminate divisions [26]. But in all cases, a computation of the determinant of one or several Macaulay matrices is needed. Canny [8] gives an exact algorithm to compute the resultant based on determinants of Macaulay

matrices that runs in polynomial space. In Valiant's algebraic model of computation (see [5]), the resultant polynomials belongs to the class VPSPACE [33], which can be seen as an equivalent to the boolean class PSPACE in this setting.

In this section, we shall prove that an improvement of Canny's algorithm for the computation of the resultant polynomial is likely to require some new techniques, in particular techniques not based on Macaulay matrices. More precisely, we show that despite being of exponential size, Macaulay matrices are efficiently representable by circuits. One could thus wish to use this small representations to get an efficient algorithm. Yet, we prove that computing the determinant of such matrices is in general PSPACE-complete (and even testing it for zero). This holds in any characteristic. Therefore, there is an exponential blow-up of the complexity when the matrix is given as a circuit. Indeed, when the matrix is given in a standard (non succinct) way, the determinant is very closely related to the complexity class #L which is the counting version of L (see [1] for more on this).

This proves that improving the PSPACE upper bound requires either to look at the fine structure of Macaulay matrices and use an *ad hoc* algorithm to compute their determinants, or to go through completely different methods. The complexity of computing the determinant of succinctly represented matrices has recently been studied by Malod [40] in the settings of Valiant's algebraic model of computation. He shows the VPSPACE-completeness of this problem.

4.1. Representation of Macaulay matrices

We first define the Macaulay matrices. This presentation follows [39,8]. We consider a system f of n homogeneous polynomials in $\mathbb{K}[x_1, \dots, x_n]$. Let d_1, \dots, d_n be the respective degrees of the f_i 's, and $d = 1 + \sum_{i=1}^n (d_i - 1)$ the *degree of the system*. Let also $\text{Mon}_d = \{x^\alpha : \alpha_1 + \dots + \alpha_n = d\}$ the set of degree- d monomials. Note that $|\text{Mon}_d| = \binom{n+d-1}{d}$.

The Macaulay matrices depend on an ordering of the variables. Let us consider the ordering $x_1 < x_2 < \dots < x_n$. The matrix Mac is defined as follows: The rows and columns of Mac are indexed by the elements of Mon_d , ordered by the reverse lexicographic order on the tuples $\alpha: \alpha < \alpha'$ if there exists i such that $\alpha_i < \alpha'_i$ and for all $j > i$, $\alpha_j = \alpha'_j$. (There are thus $|\text{Mon}_d|$ rows and columns.) The row of index x^α represents the polynomial

$$\frac{x^\alpha}{x_i^{d_i}} f_i, \quad \text{where } i = \min \left\{ j : x_j^{d_j} \text{ divides } x^\alpha \right\}. \quad (14)$$

Note that $\{j : x_j^{d_j} \text{ divides } x^\alpha\} \neq \emptyset$ thanks to the definition of d . The other Macaulay matrices are similarly defined but with the minimum in the formula depending on another ordering of the variables. In particular, we can consider the n orderings satisfying all but one of the n inequalities $x_i < x_{i+1}$ ($1 \leq i < n$) and $x_n < x_1$. That is, all possible ordering induced by the cycle (x_1, \dots, x_n) , broken at one place. Then one can show that the gcd of the n Macaulay matrices obtained in this way is the resultant of the polynomial system [39].

We now aim to prove that the matrix Mac is easily representable by a circuit. The following definition is the straightforward adaptation of the notion of *Small Circuit Representation* for graphs [19], also known as *succinct representation*. Several authors have studied the transfers of complexity results between a (classical) graph problem and its succinct version, that is the same problem but where the input is a circuit describing the graph [19,41,37,3]. Note that some variants of this notion have also been studied [51,4,17], as well as a counting version [48] and version for BSS machines [6].

Definition 3. A *circuit representation* of an integer matrix M of dimensions $(n \times m)$ is a multiple-output circuit C_M with two inputs of $\lceil \log n \rceil$ and $\lceil \log m \rceil$ bits respectively, that on input (i, j) (written in binary) evaluates to (the binary representation of) M_{ij} .

A circuit representation of a graph G is a circuit representation of its adjacency matrix.

Proposition 2. The matrix Mac has a circuit representation of polynomial size (in n and d).

Proof. We actually give a polynomial time algorithm that on input n, d, i and j outputs the entry Mac_{ij} .

The first step of the algorithm is to find the monomials x^α and x^β corresponding to the i -th row and the j -th column respectively. Let us write $A_d = \{(\alpha_1, \dots, \alpha_n) : \alpha_1 + \dots + \alpha_n = d\}$, so that $\text{Mon}_d = \{x^\alpha : \alpha \in A_d\}$. We need to find the i -th j -th elements of A_d in reverse lexicographic order.

With the reverse lexicographic order on A_d , the first elements are tuples of the form $(\alpha_1, \dots, \alpha_{n-1}, 0)$, then $(\alpha_1, \dots, \alpha_{n-1}, 1)$, and this up to $(0, \dots, 0, d)$ which is the largest element. Given an index i , we first want to find the value of α_n of the i -th element. To this end consider, for $1 \leq k \leq n$, $A_d^k = \{\alpha : \alpha_1 + \dots + \alpha_k = d \text{ and } \alpha_{k+1} = \dots = \alpha_n = 0\}$. In particular, $A_d = A_d^n$, and $A_d^k \subset A_d^{k+1}$ for all k . Moreover, for all $\alpha \in A_d^k$ and $\alpha' \in A_d \setminus A_d^k$, then $\alpha < \alpha'$. The elements of A_d of the form $(\alpha_1, \dots, \alpha_{n-1}, v)$ for some value v of α_n are in bijection with A_{d-v}^{n-1} . Thus, if we know that $\sum_{l=0}^{d-v-1} |A_l^{n-1}| < i \leq \sum_{l=0}^{d-v} |A_l^{n-1}|$, we know that $\alpha_n = v$. To continue, we remark that if the i -th element of A_d^n satisfies $\alpha_n = v$, then it is also the element of index $(i - \sum_{l=0}^{d-v-1} |A_l^{n-1}|)$ in A_{d-v}^{n-1} . This allows us to find recursively all the values of the α_k 's. This gives us an algorithm, and we shall prove it runs in polynomial time. Since

$$|A_d^k| = \binom{d+k-1}{d} = |A_{d-1}^k| \cdot \frac{d+k-1}{d},$$

we can compute the sums $\sum_{l=0}^{d-v-1} |A_l^{n-1}|$ for all values of v in a linear (in n and d) number of operations. Thus, we only need a quadratic number of operations to find the i -th and the j -th elements of A_d .

Once x^α and x^β are found, we can compute Mac_{ij} using Eq. (14). We aim to find the coefficient of x^β in the polynomial $(x^\alpha/x_i^{d_i})f_i$. Note that $\min\{j : x_j^{d_j} \text{ divides } x^\alpha\} = \min\{j : d_j \leq \alpha_j\}$. Therefore, it is easy to compute $x^\alpha/x_i^{d_i}$. Now, the coefficient x^β in $(x^\alpha/x_i^{d_i})f_i$ equals the coefficient of $x^\beta/(x^\alpha/x_i^{d_i})$ in f_i . This proves that we can find the coefficient we need in polynomial time.

Since we have a polynomial-time algorithm that given n, d, i and j computes Mac_{ij} , there exists a polynomial-size circuit representing Mac . \square

4.2. Determinant of a matrix given by a circuit

In this subsection, we show that testing for zero the determinant of a matrix given by a circuit is PSPACE-complete. This result holds in any characteristic. The proof is based on the PSPACE-completeness of testing the existence of a unique s - t -path in a graph given by a circuit.

The PSPACE-completeness of the s - t -connectivity (without the unicity condition) in a graph given by a circuit was proved via a reduction from the problem QBF [37]. The idea was to turn a quantified formula into a graph. Since the proof of PSPACE-completeness of QBF actually uses a formula to express the s - t -connectivity in a graph, one may wonder whether their proof can be simplified by removing the use of a quantified formula. A by-product of our proof is a positive answer to this question.

Lemma 2. *Let C be a circuit representing a directed graph G with two distinguished vertices s and t , with the promise that there is at most one path from s to t in G . It is PSPACE-complete to decide if such a path exists.*

Proof. The problem can be decided in polynomial space since the existence of a s - t -path in a (classical) digraph can be decided in nondeterministic logarithmic space. The classical algorithm is used as if the graph were given as input, and each time the algorithm needs to know if there is an arc from a vertex u to another vertex v , it evaluates the circuit on these inputs. This nondeterministic algorithm uses logarithmic space in the number of vertices, that is, polynomial space in the size of the input circuit.

Now let $L \in \text{PSPACE}$, decided by a deterministic Turing Machine M in polynomial space. Consider, on input x , the graph of configurations G_x^M of the machine M . This graph is described by a circuit that given as inputs two configurations c and c' of M , tells if c' can be reached from c in one step of computation. It is known that such a circuit of polynomial size exists, and that on input x , the circuit

can be built in polynomial time. Now, there is a path from the start configuration to the accepting one in G_x^M if and only if $x \in L$.

To conclude, it remains to remark that there is at most one path from the start configuration to the accepting one in a deterministic machine. \square

Corollary 2. *Let C be a circuit representing a (directed) forest, that is a directed graph F such that each vertex has out-degree at most 1, and two distinguished vertices s and t . It is PSPACE-complete to decide if there is a path from s to t .*

Proof. The configuration graph of a deterministic machine is a forest. This shows that the previous proof actually implies this stronger statement. \square

Let G be a digraph and M its adjacency matrix. A cycle cover of G is a subset of the arcs which form a set of cycles such that each vertex belongs to exactly one cycle. Therefore, if the vertices are numbered from 1 to n , a cycle cover corresponds to a permutation of $\{1, \dots, n\}$. The signature of the cycle cover is then defined as the signature of the corresponding permutation. It follows that the determinant of M equals the sum of the signatures of all the cycle covers of G .

Theorem 7. *Let C be a circuit describing a $(0, 1)$ -matrix M whose determinant is promised to be either 0, 1 or -1 . Then it is PSPACE-complete to decide if this determinant vanishes.*

Proof. A PSPACE algorithm for this problem follows from the (uniform) NC algorithm for the determinant [13].

Suppose we are given a circuit C describing a forest F , with two distinguished vertices s and t . Consider the graph G obtained by adding an arc from t to s and adding loops on all remaining vertices. Then the only cycles in G are the loops and a cycle through s and t which exists if and only if there was a path from s to t in F . Therefore, a cycle cover of G exists if and only if there is a path from s to t in F .

If we consider the adjacency matrix of G , its determinant equals the sum of the signatures of the cycle covers of G . Then the determinant is nonzero if and only if there is a s - t -path in F , in which case its value is ± 1 . \square

Corollary 3. *The problem of deciding if the determinant of a square $(0, 1)$ -matrix vanishes modulo n is PSPACE-complete for all $n \geq 2$.*

Proof. For any $n \geq 2$, $1 \not\equiv 0 \pmod n$ and $-1 \not\equiv 0 \pmod n$. Therefore, the theorem directly implies this corollary. \square

5. Final remarks

In characteristic zero, the upper and lower bounds on RESULTANT are in a sense close to each other. Indeed, $\text{NP} = \Sigma_1\text{P} \subseteq \text{AM} \subseteq \Pi_2\text{P}$, that is, AM lies between the first and the second level of the polynomial hierarchy. Furthermore, “under plausible complexity conjectures, $\text{AM} = \text{NP}$ ” [2, p. 157]. In particular, $\text{AM} = \text{BP} \cdot \text{NP}$ by definition, that is AM is a randomized version of NP, and randomization is often believed not to add any power to computation models. Improving the NP lower bound may be challenging as the proof of Proposition 1 shows that this would imply the same lower bound for Hilbert’s Nullstellensatz.

In positive characteristic, the situation is quite different. Indeed, the best known upper bound for Hilbert’s Nullstellensatz as well as for the resultant is PSPACE. As in characteristic zero, the known upper and lower bounds are therefore the same for both problems. But as the gap between the NP lower bound and the PSPACE upper bound is rather big, these problems might be of widely different complexity (more precisely, testing the resultant for zero could in principle be much easier than deciding whether a general polynomial system is satisfiable). Canny’s algorithm for computing the resultant [8] involves the computation of the determinants of exponential-size matrices, known as Macaulay matrices, in polynomial space. Those matrices admit a succinct representation. We proved that computing the determinant of a general succinctly represented matrix is PSPACE-hard (and testing it for zero is PSPACE-complete). It follows that Canny’s polynomial-space upper bound could

be improved only by exploiting the specific structure of the Macaulay matrices in an essential way, or by finding an altogether different (non Macaulay-based) approach to this problem. As pointed out in Section 2.1, in characteristic zero a different approach is indeed possible for *testing whether the resultant vanishes* (rather than for computing it). This problem is wide open in positive characteristic.

Acknowledgments

This work was partially funded by the European Community (7th PCRD Contract: PIOF-GA-2009-236197). We thank Bernard Mourrain and Maurice Rojas for sharing their insights on the complexity of the resultant in characteristic 0.

References

- [1] E. Allender, M. Ogihara, Relationships among PL, #L, and the determinant, *RAIRO-Inf. Théor. Appl.* 30 (1) (1996) 1–22.
- [2] S. Arora, B. Barak, *Computational Complexity: A Modern Approach*, first ed., Cambridge University Press, 2009.
- [3] J. Balcázar, The complexity of searching implicit graphs, *Artificial Intelligence* 86 (1) (1996) 171–188.
- [4] J. Balcázar, A. Lozano, J. Torán, The complexity of algorithmic problems on succinct instances, *Computer Science: Research and Applications* R. Baeza-Yates, U. Manber, (Eds.), pp. 351–377, 1992.
- [5] P. Bürgisser, Completeness and Reduction in Algebraic Complexity Theory, in: *Algorithms and Computation in Mathematics*, Springer, 2000.
- [6] P. Bürgisser, F. Cucker, P. De Naurois, The complexity of semilinear problems in succinct representation, *Comput. Complexity* 15 (3) (2006) 197–235.
- [7] L. Busé, C. D'Andrea, On the irreducibility of multivariate subresultants, *CR Math.* 338 (4) (2004) 287–290.
- [8] J.F. Canny, The Complexity of Robot Motion Planning, in: *ACM Doctoral Dissertation Award*, vol. 1987, MIT Press, 1988.
- [9] J.F. Canny, Some algebraic and geometric computations in PSPACE, in: *Proc. STOC'88*, pp. 460–469, 1988.
- [10] J.F. Canny, E. Kaltofen, L. Yagati, Solving systems of nonlinear polynomial equations faster, in: *Proc. SIGSAM'89*, pp. 121–128, 1989.
- [11] J.F. Canny, J.H. Reif, New lower bound techniques for robot motion planning problems, in: *Proc. FOCS'87*, pp. 49–60, 1987.
- [12] E. Cattani, A. Dickstein, Introduction to residues and resultants, in: A. Dickstein, I. Emiris (Eds.), *Solving Polynomial Equations*, Springer, 2005, pp. 1–61.
- [13] L. Csanky, Fast parallel matrix inversion algorithms, *SIAM J. Comput.* 5 (1976) 618.
- [14] C. D'Andrea, A. Dickstein, Explicit formulas for the multivariate resultant, *J. Pure Appl. Algebra* 164 (1–2) (2001) 59–86.
- [15] A. Dixon, The eliminant of three quantics in two independent variables, *Proc. Lond. Math. Soc.* 6 (1908) 468–478.
- [16] I. Emiris, B. Mourrain, Matrices in elimination theory, *J. Symbolic. Comput.* 28 (1–2) (1999) 3–43.
- [17] J. Feigenbaum, S. Kannan, M. Vardi, M. Viswanathan, Complexity of problems on graphs represented as OBDDs (extended abstract), in: *Proc. STACS'98*, p. 216, 1998.
- [18] N. Fitchas, A. Galligo, J. Morgenstern, Precise sequential and parallel complexity bounds for quantifier elimination over algebraically closed fields, *J. Pure Appl. Algebra* 67 (1) (1990) 1–14.
- [19] H. Galperin, A. Wigderson, Succinct representations of graphs, *Inform. Control* 56 (3) (1983) 183–198.
- [20] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, in: *Series of Books in the Mathematical Sciences*, W.H. Freeman, 1979.
- [21] B. Grenet, Complexité du résultant et des grands déterminants, M.Sc. Thesis, École Normale Supérieure de Lyon, 2009.
- [22] B. Grenet, P. Koiran, N. Portier, The multivariate resultant is NP-hard in any characteristic, in: P. Hliněný, A. Kucera (Eds.), *35th International Conference on Mathematical Foundations of Computer Science*, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 477–488.
- [23] J. Heintz, J. Morgenstern, On the intrinsic complexity of elimination theory, *J. Complexity* 9 (1993) 471–498.
- [24] D. Jerardi, Quantifier elimination in the theory of an algebraically-closed field, in: *Proc. STOC'89*, pp. 138–147, 1989.
- [25] G. Jeronimo, J. Sabia, Computing multihomogeneous resultants using straight-line programs, *J. Symbolic. Comput.* 42 (1–2) (2007) 218–235.
- [26] E. Kaltofen, P. Koiran, Expressing a fraction of two determinants as a determinant, in: *Proc. ISSAC'08*, pp. 141–146, 2008.
- [27] E. Kaltofen, P. Koiran, On the complexity of factoring bivariate supersparse (lacunary) polynomials, in: *Proc. ISSAC'05*, ACM, 2005, pp. 208–215.
- [28] D. Kapur, T. Saxena, Comparison of various multivariate resultant formulations, in: *Proc. ISSAC'95*, pp. 187–194, 1995.
- [29] D. Kapur, T. Saxena, L. Yang, Algebraic and geometric reasoning using Dixon resultants, in: *Proc. ISSAC'94*, pp. 99–107, 1994.
- [30] P. Koiran, Hilbert's Nullstellensatz is in the polynomial hierarchy, *J. Complexity* 12 (4) (1996) 273–286.
- [31] P. Koiran, The complexity of local dimensions for constructible sets, *J. Complexity* 16 (1) (2000) 311–323.
- [32] P. Koiran, Circuits versus trees in algebraic complexity, in: *Proc. STACS'00*, pp. 35–54, 2000.
- [33] P. Koiran, S. Perifel, VPSPACE and a transfer theorem over the reals, *Comput. Complexity* 18 (4) (2009) 551–575. Preliminary version in *Proc. STACS'07*.
- [34] T. Krick, L. Pardo, M. Sombra, Sharp estimates for the arithmetic Nullstellensatz, *Duke Math. J.* 109 (3) (2001) 521–598.
- [35] S. Lang, *Algebra*, third ed., Springer, 2002.
- [36] D. Lazard, Résolution des systèmes d'équations algébriques, *Theoret. Comput. Sci.* 15 (1) (1981) 77–110.
- [37] A. Lozano, J. Balcázar, The complexity of graph problems for succinctly represented graphs, in: *Proc. 15th Int. Workshop WG'89*, p. 277, 1989.
- [38] F. Macaulay, Some formulae in elimination, *Proc. Lond. Math. Soc.* 1 (1) (1902) 3.

- [39] F.S. Macaulay, *The Algebraic Theory of Modular Systems*, in: Cambridge Mathematical Library, Cambridge University Press, 1994.
- [40] G. Malod, Succinct algebraic branching programs characterizing non-uniform complexity classes, in: O. Owe, M. Steffen, J.A. Telle (Eds.), *Proc. FCT'11*, in: Lecture Notes in Computer Science, vol. 6914, Springer, 2011, pp. 205–216.
- [41] C. Papadimitriou, M. Yannakakis, A note on succinct representations of graphs, *Inf. Control* 71 (3) (1986) 181–185.
- [42] D.A. Plaisted, New NP-hard and NP-complete polynomial and integer divisibility problems, *Theoret. Comput. Sci.* 31 (1–2) (1984) 125–138.
- [43] J. Renegar, On the worst-case arithmetic complexity of approximating zeros of systems of polynomials, *SIAM J. Comput.* 18 (1989) 350.
- [44] A. Seidenberg, A new decision method for elementary algebra, *Ann. of Math.* 60 (2) (1954) 365–374.
- [45] I.R. Shafarevich, *Basic Algebraic Geometry 1: Varieties in Projective Space*, Springer, 1994.
- [46] V. Shoup, New algorithms for finding irreducible polynomials over finite fields, *Math. Comp.* 54 (189) (1990) 435–447.
- [47] B. Sturmfels, Sparse elimination theory, in: *Proc. Comput. Algebr. Geom. Commut. Algebra*, D. Eisenbud, L. Robbiano, (Eds.), 1991.
- [48] J. Torán, Succinct representations of counting problems, in: *Proc. 6th Int. Conf. Appl. Algebra, Algebraic Algo, and Error-Correcting Codes*, pp. 415–426, 1988.
- [49] B.L. van der Waerden, *Modern Algebra*, third ed., F. Ungar Publishing Co., New York, 1950.
- [50] J. von zur Gathen, M. Karpinski, I. Shparlinski, Counting curves and their projections, *Comput. Complexity* 6 (1) (1996) 64–99.
- [51] K. Wagner, The complexity of combinatorial problems with succinct input representation, *Acta Inform.* 23 (3) (1986) 325–356.