The multivariate resultant lies between NP and AM

Bruno Grenet, Pascal Koiran, Natacha Portier

The multivariate resultant lies between NP and AM

Bruno Grenet, Pascal Koiran and Natacha Portier*

December 11, 2009

LIP†, École Normale Supérieure de Lyon, Université de Lyon
Department of Computer Science, University of Toronto
{Bruno.Grenet,Pascal.Koiran,Natacha.Portier}@ens-lyon.fr

Rapport de Recherche RRLIP2009-34

Abstract

The resultant of a square system of homogeneous polynomials is a polynomial in their coefficients which vanishes whenever the system has a solution. Canny gave an algorithm running in polynomial space to compute it but no lower bound was known.

We investigate the complexity of the associated decision problem and give a hardness result: Testing the resultant for zero lies in the class Arthur − Merlin and is NP-hard. We give a randomized reduction and a deterministic reduction for NP-hardness. The latter can be seen as a derandomization result.

*This work was partially funded by the Fields Institute and the European Community (7th PCRD Contract: PIOF-GA-2009-236197)

†UMR 5668 École Normale Supérieure de Lyon – cnrs – UCBL – INRIA.

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 Sylvester matrix is known as the resultant of the polynomials. The size of this matrix is the sum of the degrees of the polynomials. Hence its determinant is easy to compute given the coefficients of the polynomials. Generalizations of this notion to multivariate polynomials is not always possible. The study of the possible generalizations comes within the scope of the theory of elimination [37, 30, 14, 31, 34, 15]. 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 $f_1, \dots, f_n \in K[x_1, \dots, x_n]$ is a polynomial in the indeterminate coefficients of $f_1, \dots, f_n$ which vanishes iff $f_1, \dots, f_n$ have a nonzero common root. The resultant of such a system is known as the *multivariate resultant* in the literature. This captures the case of two univariate polynomials *modulo* their homogeneization. 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, so that the square case is an important one. This result is sometimes known as Bertini's theorem. 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 [28, 32, 8, 10] and for the elimination of quantifiers in algebraically or real-closed fields [33, 18]. 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 [4, 5, 9], and more generally the multivariate resultant is used in real algebraic geometry [6, 22]. Finally, in the domain of symbolic computation progress has been made for finding explicit formulations for the resultant [11, 7, 21, 13, 3, 10, 19], see also [20].

This paper deals with the complexity of the multivariate resultant. Thus, we study systems of n homogeneous multivariate polynomials in n variables for which the roots are in projective space. That is, only *non trivial* (*i.e.* nonzero) roots are considered. Canny [5] gave in 1987 a PSPACE algorithm to compute the resultant in that case. To the authors' knowledge, this is the best known upper bound and no lower bound has ever been given[1]. The main result of this paper is a proof of the NP-hardness of the associated decision problem. A tighter upper bound (the decision problems lies in AM) is given as well.

The associated decision problem consist in deciding whether the resultant vanishes, and this actually is the problem of deciding whether the polynomials have a common root. Thus this is a variant of the *Hilbert's Nullstellensatz* problem.

Definition 1. The problem $\mathtt{HN}_{\mathbb{C}}$ is given $f_1, \dots, f_s \in \mathbb{C}[x_1, \dots, x_n]$, does there exist a tuple $\bar{a} \in \mathbb{C}^n$ such that $f_1, \dots, f_s$ vanish on $\bar{a}$?

If $f_1, \dots, f_s$ are homogeneous polynomials and $\bar{a}$ is required to be *non trivial* (*i.e.*, nonzero), this defines the problem $\mathtt{H_2N}_{\mathbb{C}}$. Further, if $s = n$, that is if there are as many homogeneous polynomials as variables, this defines the problem $\mathtt{H_2N}^{\square}_{\mathbb{C}}$.

Boolean versions of this problems, respectively denoted by $\mathtt{HN}$, $\mathtt{H_2N}$ and $\mathtt{H_2N}^{\square}$, are

[1] Canny states in his PhD thesis [5] that the problem is NP-hard but no proof seems available in the literature.

defined by considering integer polynomials as inputs and asking for a complex (non trivial in the last two cases) common root.

Testing the resultant for zero is the problem $\mathtt{H_2N}^\square$, namely the *square homogeneous Hilbert's Nullstellensatz*. We first show that the upper bound for the *Hilbert's Nullstellensatz* still holds for $\mathtt{H_2N}^\square$.

We denote here by AM the class $\mathsf{Arthur} - \mathsf{Merlin}$, defined by *interactive proofs with public coins* (see [2]).

Proposition 1. *Under the Generalized Riemann Hypothesis,* $\mathtt{H_2N}^\square$ *is in the class* AM.

Proof. Koiran [23] proved that $\mathtt{HN} \in \mathsf{AM}$ under the same hypothesis. Consider an instance $\mathcal{S}$ of $\mathtt{H_2N}^\square$, that is n homogeneous polynomials $f_1, \ldots, f_n \in \mathbb{Z}[x_1, \ldots, x_n]$. The polynomials $f_1, \ldots, f_n$ can be viewed as elements of $\mathbb{Z}[x_1, \ldots, x_n, y_1, \ldots, y_n]$ where $y_1, \ldots, y_n$ are new variables which do not appear in the f_i. Let $\mathcal{T}$ be the system containing all the f_i and the new (non-homogeneous) polynomial $\sum_{i=1}^n x_i y_i - 1$. This is an instance of the problem HN. It remains to prove that $\mathcal{S}$ and $\mathcal{T}$ are equivalent.

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

The remaining of the paper is devoted to prove the NP-hardness of $\mathtt{H_2N}^\square$. The reduction is done in several steps, from $\mathtt{3-SAT}$ to $\mathtt{H_2N}^\square$. The first steps actually prove that $\mathtt{H_2N}$ is NP-hard, which is an already known result [24]. The proof is nevertheless given in Section 2 as the special form of the obtained system will be useful in the last step of the reduction. Then Section 3 is dedicated to a fairly simple randomized reduction between $\mathtt{H_2N}$ and $\mathtt{H_2N}^\square$. Section 4 is a deterministic reduction, but the proof is more complex. This reduction can be viewed as a derandomization result. Furthermore, those two completely different reductions adopt opposite viewpoints: In the randomized one, the instance of $\mathtt{H_2N}$ is transformed into a square system by decreasing the number of polynomials, while the deterministic one proceeds by adding some new variables. In the last section, we present some results on computing *succintly represented* determinants (that is, given by circuits), namely these are PSPACE-complete to compute. This gives clues that Canny's approach cannot be significantly improved without a very careful examination of the structure of the matrices involved.

2 Preliminary work

In order to transform a $\mathtt{3-SAT}$ instance into a system of polynomials, a formula is expressed as a system of boolean equations to begin with. Toward this end, let us define the problem `Boolsys`. The input is a system of boolean equations in the variables $X_1, \ldots, X_n$, each equation being on 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 a assignment of the variables such that each equation is satisfied.

Lemma 1. `Boolsys` *is* NP*-hard.*

Proof. There is an easy reduction from $3-\mathtt{SAT}$. Each clause $l_1 \vee l_2 \vee l_3$ is replaced by at most six equations. For example, a clause $x \vee \bar{y} \vee z$ where x, y, z are variables is turned into the four following equations:

$$\begin{cases} y' &= \neg y \\ X &= x \vee y' \\ Y &= X \vee z \\ Y &= \text{True} \end{cases}$$

where y', X and Y are fresh variables. There can be up to six equations if all variables in the clause are negated.

It is straightforward to verify that the `Boolsys` instance that is created when each clause is replaced by the corresponding equations is equivalent to the original formula. □

Lemma 2. $\mathtt{H_2N}$ *is* NP*-hard.*

Proof. We consider an instance I of `Boolsys` with n variables $X_1, \ldots, X_n$ and create an instance J of $\mathtt{H_2N}$ with $n+1$ variables $x_0, x_1, \ldots, x_n$. The variable x_0 has to be seen as a fresh one while for each $i > 0$, x_i corresponds to X_i.

The new instance J contains the polynomial $x_0^2 - x_i^2$ for each i, so that the x_i can only have the values x_0 and $-x_0$. Now every equation of the `Boolsys` system is turned into a polynomial in $\mathtt{H_2N}$ in the following manner:

- $X_i = \text{True}$ is turned into $(x_i + x_0)^2 = 0$;
- $X_i = \neg X_j$ is turned into $(x_i + x_j)^2 = 0$;
- $X_i = X_j \vee X_k$ is turned into $(x_i + x_0)^2 - (x_j + x_0) \cdot (x_k + x_0) = 0$.

The equivalence between I and J is now proved. If $(a_0, \ldots, a_n)$ is a root of J, it must verify $a_0 = \pm a_i$ for each i. Furthermore, by homogeneity, the value of a_0 can be fixed for the sequel. To prove the equivalence, it is sufficient to prove that every equation of `Boolsys` is satisfied by an assignment of its variables iff the corresponding polynomial in $\mathtt{H_2N}$ vanishes at $(a_0, \ldots, a_n)$ where $a_i = -a_0$ if X_i is true and $a_i = a_0$ if X_i is false.

This property is clear for the equation $X_i = \text{True}$. The equation $X_i = \neg X_j$ is satisfied iff X_i and X_j have not the same value, so iff $(a_i + a_j) = 0$. Now $X_i = X_j \vee X_k$ is satisfied iff X_i is true and at least one amongst X_j and X_k also, or X_i, X_j and X_k are false. And the corresponding equation holds iff $(a_i + a_0) = 0$ and one among $(a_j + a_0)$ and $(a_k + a_0)$ is zero, or those three sums equal $2a_0$. □

The next two sections are devoted to two different reductions from $\mathtt{H_2N}$ to $\mathtt{H_2N}^{\square}$. The first one is a randomized reduction, while the second one is deterministic. In the deterministic reduction, the starting problem is not exactly $\mathtt{H_2N}$. One must actually use the form of the system that was built in the previous proof. In the sequel, such a system where the polynomials $x_0^2 - x_i^2$ appear for every i, and where the other polynomials are of one of the three forms $(x_i + x_j)^2$, $(x_i + x_j)^2$ or $(x_i + x_j)^2 - (x_i + x_0) \cdot (x_k + x_0)$ is denoted by $\mathbb{C}-\mathtt{Boolsys}$. This problem actually is a translation of `Boolsys` in the language of the field $\mathbb{C}$.

3 Randomized reduction

The aim of this section is to give a quite simple reduction from $\mathtt{H_2N}$ to $\mathtt{H_2N}^{\square}$. In view of the previous reduction, a reduction from $\mathtt{H_2N}$ restricted to degree-2 polynomials is sufficient to prove the NP-hardness of $\mathtt{H_2N}^{\square}$. In the sequel, we give a reduction in this restricted case, although it could be easily extended to the more general case. This reduction is a randomized one: There exists a *probabilistic* Turing Machine which turns any instance of $\mathtt{H_2N}$ into an instance of $\mathtt{H_2N}^{\square}$ which is equivalent with probability at least 2/3. For more on this kind of reduction, see [2, §7.6].

In the problem $\mathtt{H_2N}$, the instance consists in s homogeneous polynomials in $n+1$ variables. If $s < n+1$, an equivalent square system is obtained by duplicating the last polynomial $n+1-s$ times. In the sequel, we assume $s > n+1$. The natural idea in order to decrease the number of polynomials is to define the instance of $\mathtt{H_2N}^{\square}$ as a set of $n+1$ linear combinations of the s polynomials of $\mathtt{H_2N}$. In [26, §4.4], a condition on the existence of (possibly trivial) roots for non necessarily homogeneous polynomials is studied. The following special case of this result will be useful for the reduction:

Lemma 3. *Let $f_1, \ldots, f_s \in \mathbb{Z}[x_0, \ldots, x_n]$ be polynomials of degree at most 2 without a common root. There exists a non zero polynomial $F \in \mathbb{C}[Z_{1,1}, \ldots, Z_{n+1,s}]$ of degree at most 3^{n+1} such that $F(\bar{\alpha}) \neq 0$ for $\bar{\alpha} = (\alpha_{1,1}, \ldots, \alpha_{n+1,s})$ implies that the polynomials $g_i = \sum_{j=1}^{s} \alpha_{ij} f_j$ $(1 \leq i \leq n+1)$ have no common root.*

The following lemma is the core of the randomized reduction:

Lemma 4. *Let $f_1, \ldots, f_s \in \mathbb{Z}[x_0, \ldots, x_n]$ be homogeneous polynomials of degree 2, and let $f = (f_1, \ldots, f_s)$. Let $g = (g_1, \ldots, g_{n+1})$ be a random function defined in the following manner: Pick $s \cdot (n+1)$ integers α_{ij} $(1 \leq i \leq n+1,\ 1 \leq j \leq s)$ independently at random with uniform distribution in $\{0, \ldots, 3^{n+2}\}$, and for all i, let $g_i = \sum_{j=1}^{s} \alpha_{ij} f_j$. Then,*

(i) if f has a non trivial root, then $\Pr[g \text{ has a non trivial root}] = 1$;

(ii) if f has no non trivial root, then $\Pr[g \text{ has no non trivial root}] \geq 2/3$.

Proof. The first point is clear, every root of f being a root of g. Let us prove the second point.

Suppose that f has no non trivial root, and consider a non trivial tuple $(a_0, \ldots, a_n)$. One at least of the a_i is non zero, say a_0. Consider the n-tuple $(\tilde{a}_1, \ldots, \tilde{a}_n)$ defined by $\tilde{a}_i = a_i/a_0$. As the f_j and the g_i are homogeneous, $(a_0, \ldots, a_n)$ is a root iff $(1, \tilde{a}_1, \ldots, \tilde{a}_n)$ is. One can also define $\tilde{f}_j \in \mathbb{Z}[x_1, \ldots, x_n]$ by $\tilde{f}_j(x_1, \ldots, x_n) = f_j(1, x_1, \ldots, x_n)$, and the same for the g_i. Then, $(\tilde{a}_1, \ldots, \tilde{a}_n)$ is a root of $\tilde{f}_j$ iff $(a_0, \ldots, a_n)$ is a (non trivial) root of f_j, and the same holds for the g_i. Furthermore, $\tilde{g}_i = \sum_i \alpha_{ij} \tilde{f}_j$. Thus, Lemma 3 proves the existence of a polynomial $F \in \mathbb{C}[\bar{\alpha}]$ with degree at most 3^{n+1} such that $F(\bar{\alpha}) \neq 0$ implies that the $\tilde{g}_i$ have no common root. In particular, if $F(\bar{\alpha}) \neq 0$, then $(\tilde{a}_1, \ldots, \tilde{a}_n)$ is not a common root of the $\tilde{g}_i$, and $(a_0, \ldots, a_n)$ is not a common root of the g_i. As this is valid for all non trivial $(a_0, \ldots, a_n)$, then $F(\bar{\alpha}) \neq 0$ implies that the g_i have no non trivial common root.

Then, applying Schwartz-Zippel lemma [29] to F shows that with probability at least 2/3, the g_i have no non trivial common root. □

In summary, if a new system of $(n+1)$ homogeneous polynomials $g_1, \ldots, g_{n+1}$ in $(n+1)$ variables is defined by random linear combinations of the s original polynomials $f_1, \ldots, f_s$, then with probability at least $2/3$, the f_j have a non trivial common root iff the g_i have one. This proves the following:

Theorem 1. *The problem* $\mathtt{H_2N}^{\square}$ *is* NP*-hard under randomized reduction.*

4 Deterministic reduction

In this section, we give a deterministic reduction from $\mathbb{C}-\mathtt{Boolsys}$ to $\mathtt{H_2N}^{\square}$, the problem of deciding whether a square system of homogeneous polynomials has a non trivial root.

The idea of the reduction is to add some new variables and to slightly change the existing polynomials without adding any new one in order to get a square system. This has to be done carefully in order to avoid the situation where the new system has a non trivial root consisting in a trivial part for the old variables and a non trivial one for the new variables. The method is based on the *Jacobian matrix* of the system. In the following definition, as in the sequel of the section, $\bar{x}$ represents a vector $(x_0, \ldots, x_n)$.

Definition 2. Let f be a function from $\mathbb{C}^{n+1}$ to $\mathbb{C}^s$ defined by

$$f : \begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix} \mapsto \begin{pmatrix} f_1(\bar{x}) \\ f_2(\bar{x}) \\ \vdots \\ f_s(\bar{x}) \end{pmatrix}.$$

The *Jacobian matrix* of f is defined by

$$(J_f)_{ij} = \frac{\partial f_i}{\partial x_j}$$

for $1 \leq i \leq s$ and $0 \leq j \leq n$.

Furthermore, $J_f(\bar{\alpha})$ represents the Jacobian matrix of f taken at the point $\bar{\alpha}$.

The Jacobian matrix has the following interesting property:

Lemma 5. *Let f be a homogeneous polynomial function with $n+1$ variables and s components. If $(a_0, \ldots, a_n)$ is a non trivial root of f, then the rank of $J_f(\bar{a})$ is at most n.*

Proof. As the system is homogeneous, there exists a root iff there exists a line of roots. For every i, this line lies in the algebraic set E_i defined by $f_i(\bar{x}) = 0$. So for every i the line lies in the tangent space to E_i. Now if a vector X belongs to this line, then $J_f(\bar{a}) \cdot X = (0, \ldots, 0)^t$. As a result, $J_f(\bar{a})$ has rank not greater than n. □

For the purpose of the reduction, a kind of converse of that lemma is necessary. Unfortunately, this is not true in a general case. Nevertheless, in the case of the problem $\mathbb{C}-\mathtt{Boolsys}$, the systems that appear have some nice properties. Let us recall the polynomials in such a system:

- $x_0^2 - x_i^2$, for each $i > 0$;

- $(x_i + x_0)^2$, coming from $X_i = \text{True}$ in `Boolsys`;
- $(x_i + x_j)^2$, coming from $X_i = \neg X_j$;
- $(x_i + x_0)^2 - (x_j + x_0) \cdot (x_k + x_0)$, coming from $X_i = X_j \vee X_k$.

In the sequel, f denotes a instance of the problem $\mathbb{C}-$`Boolsys`, that is a system with those four kinds of polynomials. A converse of the previous lemma is now proved:

Lemma 6. *Let f be an instance of $\mathbb{C}-$`Boolsys`, and let $\bar{a}$ be a non zero $(n+1)$-tuple such that for each $i > 0$, $a_i^2 = a_0^2$. Then*

(i) the rank of $J_f(\bar{a})$ is at least n;

(ii) the rank of $J_f(\bar{a})$ is equal to n iff $\bar{a}$ is a non trivial root of f.

Proof. The first point is trivial. The system contains n polynomials of the form $x_0^2 - x_i^2$, and the condition on the a_i implies that every a_i is nonzero. So the $n \times (n+1)$ matrix

$$\begin{pmatrix} a_0 & -a_1 & 0 & \dots & 0 \\ a_0 & 0 & -a_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ a_0 & 0 & \dots & 0 & -a_n \end{pmatrix}$$

is a submatrix of $\frac{1}{2}J_f(\bar{a})$, and is clearly of rank n.

Without loss of generality, the n rows above are considered as the n first rows of $\frac{1}{2}J_f(\bar{a})$. The second point may be expressed as follows: $\bar{a}$ is a non trivial root of f iff every row among the last ones is a linear combination of the n first ones. To prove that, something stronger is actually proved, namely that each polynomial of f (but the n first ones) vanishes at $\bar{a}$ iff the corresponding row in $J_f(\bar{a})$ is a linear combination of the n first rows. There exist three kinds of rows among the lowest ones, corresponding to the three kinds of polynomials different from $x_0^2 - x_i^2$. Each one is now separately studied to prove the desired property.

The first kind of row corresponds to a polynomial $(x_i + x_0)^2$. The row in $J_f(\bar{a})$ has the form

$$(2(a_0 + a_i), 0, \dots, 0, 2(a_0 + a_i), 0, \dots, 0),$$

where the nonzero coefficients are the first one and the $(i+1)$-th one. Given the form of the n first rows, this row can be a linear combination of them iff it is proportional to the i-th one. Hence it is even sufficient to know whether the submatrix of $\frac{1}{2}J_f(\bar{a})$

$$\begin{pmatrix} a_0 & -a_i \\ a_0 + a_i & a_0 + a_i \end{pmatrix}$$

is singular. As its determinant is $(a_0 + a_i)^2$, it is clear that this matrix is singular iff the polynomial vanishes at $\bar{a}$.

The same method works for the second kind of polynomials, $(x_i + x_j)^2$. It is sufficient to study the submatrix of $\frac{1}{2}J_f(\bar{a})$

$$\begin{pmatrix} a_0 & -a_i & 0 \\ a_0 & 0 & -a_j \\ 0 & a_i + a_j & a_i + a_j \end{pmatrix}.$$

Its determinant is equal to $a_0 \cdot (a_i + a_j)^2$, and as a_0 is nonzero, the equivalence is clear.

The last kind of polynomials is of the form $(x_i + x_0)^2 - (x_j + x_0) \cdot (x_k + x_0)$. The studied matrix in this case is the submatrix of $J_f(\bar{a})$

$$\begin{pmatrix} 2a_0 & -2a_i & 0 & 0 \\ 2a_0 & 0 & -2a_j & 0 \\ 2a_0 & 0 & 0 & -2a_k \\ 2a_i - a_j - a_k & 2(a_i + a_0) & -a_k - a_0 & -a_j - a_0 \end{pmatrix}.$$

Its determinant is equal to

$$8\left(2a_j a_k \cdot (a_i + a_0)^2 - a_i \cdot (a_j + a_k)(a_j + a_0)(a_k + a_0)\right).$$

Hence, the matrix is singular iff

$$2a_j a_k \cdot (a_i + a_0)^2 = a_i \cdot (a_j + a_k)(a_j + a_0)(a_k + a_0). \tag{1}$$

The left-hand side of (1) is zero iff $a_i = -a_0$. Moreover, the sum $a_j + a_k$ is zero iff $a_j = -a_k$, and as $a_j^2 = a_k^2 = a_0^2$ the sum is zero iff $a_j + a_0$ or $a_k + a_0$ is also zero. So, the right-hand side of (1) is zero iff $a_j = -a_0$ or $a_k = -a_0$.

Furthermore, both sides are nonzero iff $a_0 = a_i = a_j = a_k$, and in that case, $2a_j a_k = a_i \cdot (a_j + a_k)$. Hence, (1) holds iff $(a_i + a_0)^2 = (a_j + a_0) \cdot (a_k + a_0)$, *i.e.* iff (a_0, a_i, a_j, a_k) is a root of the polynomial.

This proves the lemma. □

All the ingredients which are needed for the reduction have been proved. We now state our main theorem and prove it:

Theorem 2. $\mathtt{H_2N}^{\square}$ *is* NP*-hard.*

Proof. The problem $\mathbb{C}-$`Boolsys` defined in Section 2 is reduced to $\mathtt{H_2N}^{\square}$. Let f be an instance of $\mathbb{C}-$`Boolsys`. The vector-valued function f has $n+1$ variables and s components. The n first components are still $f_i(\bar{x}) = x_0^2 - x_i^2$, $i = 1, \ldots, n$. Each remaining component is of the form $(x_i + x_0)^2$, $(x_i + x_j)^2$ or $(x_i + x_0)^2 - (x_j + x_0) \cdot (x_k + x_0)$. The instance f is reduced to an instance $g : \mathbb{C}^s \to \mathbb{C}^s$ of $\mathtt{H_2N}^{\square}$. The variables are denoted by $x_0, \ldots, x_n$ and $y_1, \ldots, y_{s-n-1}$. The components of g are defined as follows:

$$g(\bar{x}, \bar{y}) = \begin{pmatrix} f_1(\bar{x}) & & \\ \vdots & & \\ f_n(\bar{x}) & & \\ f_{n+1}(\bar{x}) & & +5y_1^2 \\ f_{n+2}(\bar{x}) & -y_1^2 & +5y_2^2 \\ & \vdots & \\ f_{n+i}(\bar{x}) & -y_{i-1}^2 & +5y_i^2 \\ & \vdots & \\ f_{s-1}(\bar{x}) & -y_{s-n-2}^2 & +5y_{s-n-1}^2 \\ f_s(\bar{x}) & -y_{s-n-1}^2 & \end{pmatrix}$$

It remains to prove that the reduction is valid, that is f has a non trivial root iff g does. Clearly, if $f(a_0, \ldots, a_n) = \bar{0}$, then $g(a_0, \ldots, a_n, 0, \ldots, 0) = \bar{0}$. Let us prove the converse: if f has no non trivial root, then neither does g. Let $(\bar{a}, \bar{b})$ be a non zero tuple and let us prove that $g(\bar{a}, \bar{b}) \neq \bar{0}$.

We begin with a few remarks. If $\bar{b} = \bar{0}$, then $g(\bar{a}, \bar{0}) = f(\bar{a})$, so $g(\bar{a}, \bar{0}) = \bar{0}$ implies $\bar{a} = \bar{0}$, and $(\bar{a}, \bar{b})$ is the trivial tuple. Moreover, if $\bar{a} = \bar{0}$, it is easy to see that $g(\bar{0}, \bar{b}) = \bar{0}$ implies $\bar{b} = \bar{0}$. Furthermore, if $\bar{a}$ does not satisfy $a_0^2 = \cdots = a_n^2$, then $g(\bar{a}, \bar{b}) \neq \bar{0}$. In the sequel, $(\bar{a}, \bar{b})$ is supposed to verify $a_0^2 = \cdots = a_n^2 \neq 0$, and $\bar{b} \neq \bar{0}$. To begin with, all the components of b are supposed to be nonzero.

Consider the Jacobian matrix of g at the point $(\bar{a}, \bar{b})$, denoted by $J_g(\bar{a}, \bar{b})$. The first $(n+1)$ columns of $J_g(\bar{a}, \bar{b})$ form $J_f(\bar{a})$, the Jacobian matrix of f at $\bar{a}$. As $f(\bar{a}) \neq \bar{0}$, Lemma 6 states that $J_f(\bar{a})$ has maximal rank. It is now proved that $J_g(\bar{a}, \bar{b})$ has also maximal rank, and hence Lemma 5 ensures that $g(\bar{a}, \bar{b}) \neq \bar{0}$. The Jacobian matrix of g is

$$J_g(\bar{a}, \bar{b}) = \left(\begin{array}{ccc|ccc} 2a_0 & -2a_1 & & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots & & \vdots \\ 2a_0 & & -2a_n & 0 & \cdots & 0 \\ \hline & & & 10b_1 & & \\ & M & & -2b_1 & \ddots & \\ & & & & \ddots & 10b_{s-n-1} \\ & & & & & -2b_{s-n-1} \end{array}\right).$$

The submatrix M contains three kinds of rows R, R' and R'' for the three kinds of polynomials $(x_i + x_0)^2$, $(x_i + x_j)^2$ and $(x_i + x_0)^2 - (x_j + x_0) \cdot (x_k + x_0)$ appearing as components of f. Most coefficients of the rows are equal to 0. As we have seen in the proof of Lemma 6, R and R' have each two nonzero coefficients, while R'' has four:

- $R_1 = 2(x_0 + x_i)$ and $R_{i+1} = 2(x_0 + x_i)$;
- $R'_{i+1} = R'_{j+1} = 2(x_i + x_j)$;
- $R''_1 = 2x_i - x_j - x_k$, $R''_{i+1} = 2(x_i + x_0)$, $R''_{j+1} = -x_k - x_0$ and $R''_{k+1} = -x_j - x_0$.

Each of the $(n+1)$ first columns of $J_g(\bar{a}, \bar{b})$ can be seen as an integer column multiplied by $2a_0$. Similarly, by dividing the last $s - n - 1$ columns of $J_g(\bar{a}, \bar{b})$ respectively by $2b_1, \ldots, 2b_{s-n-1}$ integer columns are obtained. So the rank of $J_g(\bar{a}, \bar{b})$ is the same as the rank of the integer matrix obtained by dividing each column by an appropriate value. The rank does not change either if we apply a circular permutation to the $(n+1)$ first columns so that the the first column becomes the $(n+1)$-th, the second becomes the first, and so on. Hence, the rank of $J_g(\bar{a}, \bar{b})$ is

equal to the rank of the matrix

$$J = \left(\begin{array}{ccc|ccc} \pm 1 & & & 1 & 0 & \cdots & 0 \\ & \ddots & & \vdots & \vdots & & \vdots \\ & & \pm 1 & 1 & 0 & \cdots & 0 \\ \hline & & & & 5 & & \\ & M_1 & & & -1 & \ddots & \\ & & & & & \ddots & 5 \\ & & & & & & -1 \end{array}\right),$$

where the rows of M_1 have 1-norm bounded by 4.

To prove that J has maximal rank, its determinant is computed. The following operation does not change the determinant:

$$C_{n+1} \leftarrow C_{n+1} + \sum_{i=1}^{n} \pm C_i, \tag{2}$$

where C_i is the i-th column of J. If the ± 1 in the sum are well chosen, the n first coefficients of C_{n+1} vanish. The matrix we obtain is partitioned into four blocks where the top left block is a diagonal made of ± 1 and the top right block is zero. The last column of M_1, modified by the operation (2), is denoted by $(c_0, \ldots, c_{s-n-1})^t$. The determinant of J is then equal, up to sign, to the determinant of the matrix

$$\begin{pmatrix} c_0 & 5 & & \\ \vdots & -1 & \ddots & \\ \vdots & & \ddots & 5 \\ c_{s-n-1} & & & -1 \end{pmatrix}.$$

For $0 \leq i \leq s-n-1$, $|c_i| \leq 4$ as the 1-norms of the rows of M_1 are also bounded by 4. Furthermore, if all c_i vanish, then $J_F(\bar{a})$ is rank-deficient, which contradicts the hypothesis. The determinant of the above matrix can be shown to be equal to $(-1)^{s-n-1} \cdot (c_0 + 5c_1 + 5^2 c_2 + \cdots + 5^{s-n-1} c_{s-n-1})$. For each i, let $c_i^+ = \max\{c_i, 0\}$ and $c_i^- = \max\{-c_i, 0\}$. Then $c_i = c_i^+ - c_i^-$, and $0 \leq c_i^+, c_i^- \leq 4$. Now the determinant is zero iff $\sum_i 5^i c_i^+ = \sum_i 5^i c_i^-$. By the unicity of base-5 representation, this means that for all i, $c_i^+ = c_i^-$, and so $c_i = 0$, which is a contradiction.

As soon as no b_i vanishes, $J_g(\bar{a}, \bar{b})$ has maximal rank and by Lemma 5, $g(\bar{a}, \bar{b}) \neq \bar{0}$. Suppose now that some b_i vanish. Without loss of generality, one can suppose that the non-zero components of $\bar{b}$ are $b_1, \ldots, b_k$. Consider the function $\tilde{g}(\bar{x}, y_1, \ldots, y_k) = g(\bar{x}, y_1, \ldots, y_k, \bar{0})$. The Jacobian matrix of $\tilde{g}$ at the point $(\bar{a}, b_1, \ldots, b_k)$ is formed by the $(n+k+1)$ first rows of $J_g(\bar{a}, \bar{b})$ which have been proved linearly independent. Lemma 5 also applies to $\tilde{g}$, and $\tilde{g}(\bar{a}, b_1, \ldots, b_k) = g(\bar{a}, \bar{b}) \neq 0$. □

This result can be seen as a derandomization result as a simpler randomized reduction exists. In the previous proof, the function g is defined by adding to each f_i, $n+1 \leq i \leq s$, a term of the form $\sum_{j=1}^{s-n-1} \alpha_{ij} y_j^2$, where the matrix $(\alpha_{ij})_{ij}$ is a bidiagonal matrix ($\alpha_{ii} = 5$ and $\alpha_{i,i+1} = -1$ for $1 \leq i \leq s-n-1$). If we replace it by the matrix of indeterminates $(\alpha_{ij})_{ij}$, it is easy to check that the determinant is a nonzero polynomial in the α_{ij}. Hence it is sufficient to use Schartz-Zippel Lemma to conclude that if the α_{ij} are randomly chosen (in an appropriate interval), the determinant does not vanish.

5 Final remarks

The upper and lower bounds on $\mathtt{H}_2\mathtt{N}^{\square}$ are in a sense "close" to each other. Indeed, $\mathsf{NP} = \Sigma_1\mathsf{P} \subseteq \mathsf{AM} \subseteq \Pi_2\mathsf{P}$, that is AM lies between the first and the second level of the polynomial hierarchy. Furthermore, "under plausible complexity conjectures, AM = NP" [2, p157]. Improving the lower bound may be challenging as the proof of Proposition 1 shows that this would imply the same bound for the *Hilbert's Nullstellensatz*.

Computing a multivariate resultant is as least as hard as testing it for zero. Therefore, the lower bound applies for computing the resultant. Nevertheless, there remains a big gap between Canny's PSPACE upper bound and our NP lower bound. We decreased the gap for the decision problem, and it may be possible to decrease it for the function. Indeed, Canny's algorithm uses determinant of Macaulay matrices (which are generalizations of the Sylvester matrix to multivariate polynomials). Those matrices have an exponential dimension but admit a succinct representation (in the sense of [16]). One can prove that computing the determinant of a general succinctly represented matrix is FPSPACE-complete (and even testing for zero is PSPACE-complete) [17]. This corresponds to an exponential blow up of the complexity with regards to the classically described determinant. The decision problem is indeed $\mathsf{C}_=\mathsf{L}$-complete [1] and computing its value is GapL-complete [35, 38, 36, 12]. Moreover, at the "PSPACE level", #PSPACE = FPSPACE [27], and therefore $\mathsf{C}_=\mathsf{PSPACE}$ = PSPACE and GapPSPACE = FPSPACE (with obvious definitions of these classes with analogy to $\mathsf{C}_=\mathsf{L}$ and GapL).

This may indicate that in order to improve Canny's algorithm one has to use some very specific properties of the Macaulay matrices. Another interesting question is to characterize the complexity of the resultant within Valiant's algebraic framework. Indeed, it is proved in [25] that the Macaulay matrices form a VPSPACE family of polynomials. Finally, it would be interesting to know whether the randomized reduction of Theorem 1 can be derandomized.

References

1. E. Allender and M. Ogihara. Relationships among PL,#L, and the determinant. *RAIRO-Inf. Théor. Appl.*, 30(1):1–22, 1996.
2. S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
3. L. Busé and C. D'Andrea. On the irreducibility of multivariate subresultants. *CR Math.*, 338(4):287–290, 2004.
4. J. F. Canny. A new algebraic method for robot motion planning and real geometry. In *Proc. FOCS'87*, pages 39–48, 1987.
5. J. F. Canny. *The complexity of robot motion planning*, volume 1987 of *ACM Doctoral Dissertation Award*. MIT Press, 1988.
6. J. F. Canny. Some algebraic and geometric computations in PSPACE. In *Proc. STOC'88*, pages 460–469, 1988.
7. J. F. Canny. Generalized characteristic polynomials. In *Proc. ISSAC'88*, 1989.
8. J. F. Canny, E. Kaltofen, and L. Yagati. Solving systems of nonlinear polynomial equations faster. In *Proc. SIGSAM'89*, pages 121–128, 1989.
9. J. F. Canny and J. H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. FOCS'87*, pages 49–60, 1987.
10. E. Cattani and A. Dickenstein. Introduction to residues and resultants. In A. Dickenstein and I. Emiris, editors, *Solving polynomial equations*, pages 1–61. Springer, 2005.
11. G. Collins. The calculation of multivariate polynomial resultants. In *Proc. SYMSAC'71*, pages 212–222, 1971.
12. C. Damm. $\mathsf{DET} = \mathsf{L}^{\#\mathsf{L}}$?, Informatik-Preprint 8. *Fachbereich Informatik der Humboldt-Universität zu Berlin*, 1991.
13. C. D'Andrea and A. Dickenstein. Explicit formulas for the multivariate resultant. *J. Pure Appl. Algebra*, 164(1-2):59–86, 2001.
14. A. Dixon. The eliminant of three quantics in two independent variables. *Proc. Lond. Math. Soc.*, 6:468–478, 1908.
15. I. Emiris and B. Mourrain. Matrices in elimination theory. *J. Symb. Comput.*, 28(1-2):3–43, 1999.
16. H. Galperin and A. Wigderson. Succinct representations of graphs. *Inform. Control*, 56(3):183–198, 1984.
17. B. Grenet. Difficulté du résultant et des grands déterminants. Technical report, RRLIP2009-32, LIP, 2009. `http://prunel.ccsd.cnrs.fr/ensl-00431714/`.
18. D. Ierardi. Quantifier elimination in the theory of an algebraically-closed field. In *Proc. STOC'89*, pages 138–147, 1989.
19. G. Jeronimo and J. Sabia. Computing multihomogeneous resultants using straight-line programs. *J. Symb. Comput.*, 42(1-2):218–235, 2007.
20. E. Kaltofen and P. Koiran. Expressing a fraction of two determinants as a determinant. In *Proc. ISSAC'08*, pages 141–146, 2008.
21. D. Kapur and T. Saxena. Comparison of various multivariate resultant formulations. In *Proc. ISSAC'95*, pages 187–194, 1995.

22. D. Kapur, T. Saxena, and L. Yang. Algebraic and geometric reasoning using Dixon resultants. In *Proc. ISSAC'94*, pages 99–107, 1994.
23. P. Koiran. Hilbert's Nullstellensatz is in the polynomial hierarchy. *J. Complexity*, 12(4):273–286, 1996.
24. P. Koiran. The complexity of local dimensions for constructible sets. *J. Complexity*, 16(1):311–323, 2000.
25. P. Koiran and S. Perifel. VPSPACE and a transfer theorem over the reals. In *Proc. STACS'07*, pages 417–258, 2007.
26. T. Krick, L. Pardo, and M. Sombra. Sharp estimates for the arithmetic Nullstellensatz. *Duke Math. J.*, 109(3):521–598, 2001.
27. R. Ladner. Polynomial space counting problems. *SIAM J. Comput.*, 18:1087, 1989.
28. D. Lazard. Résolution des systèmes d'équations algébriques. *Theor. Comput. Sci.*, 15(1):77 – 110, 1981.
29. R. J. Lipton. The Curious History of the Schwartz-Zippel Lemma, 2009. `http://rjlipton.wordpress.com/2009/11/30/the-curious-history-of-the-schwartz-zippel-l`
30. F. Macaulay. Some formulae in elimination. *Proc. Lond. Math. Soc.*, 1(1):3, 1902.
31. F. Macaulay. *The algebraic theory of modular systems*. Cambridge University Press, Cambridge, 1916.
32. J. Renegar. On the worst-case arithmetic complexity of approximating zeros of systems of polynomials. *SIAM J. Comput.*, 18:350, 1989.
33. A. Seidenberg. A new decision method for elementary algebra. *Ann. Math.*, 60(2):365–374, 1954.
34. B. Sturmfels. Sparse elimination theory. In *Proc. Comput. Algebr. Geom. Commut. Algebra*. D. Eisenbud and L. Robbiano, eds., 1991.
35. S. Toda. Counting problems computationally equivalent to computing the determinant. Technical report, Tech. Rep. CSIM 91-07, Dept C.S., U. of Electro-Communications, Tokyo, Japan, 1991.
36. L. Valiant. Why is Boolean complexity theory difficult? In *Proc. Lond. Math. Soc. Symp. on Boolean Function Complexity*, pages 84–94. Cambridge University Press New York, NY, USA, 1992.
37. B. L. van der Waerden. *Modern Algebra*. (third ed.) F. Ungar Publishing Co., New York, 1950.
38. V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proc. Struct. in Compl. Th. Conf.*, page 284, 1991.