Fast computation of generic bivariate resultants*

JORIS VAN DER HOEVEN[a], GRÉGOIRE LECERF[b]

CNRS, École polytechnique, Institut Polytechnique de Paris
Laboratoire d'informatique de l'École polytechnique (LIX, UMR 7161)
1, rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing, CS35003
91120 Palaiseau, France

a. Email: vdhoeven@lix.polytechnique.fr
b. Email: lecerf@lix.polytechnique.fr

Preliminary version of May 21, 2020

We prove that the resultant of two "sufficiently generic" bivariate polynomials over a finite field can be computed in quasi-linear expected time, using a randomized algorithm of Las Vegas type. A similar complexity bound is proved for the computation of the lexicographical Gröbner basis for the ideal generated by the two polynomials.

KEYWORDS: complexity, algorithm, computer algebra, resultant, elimination, multipoint evaluation

1. INTRODUCTION

The efficient computation of resultants is a fundamental problem in elimination theory and for the algebraic resolution of systems of polynomial equations. Given an effective field $\mathbb{K}$, it is well known [10, chapter 11] that the resultant of two univariate polynomials $P, Q \in \mathbb{K}[x]$ of respective degrees $d \geqslant e$ can be computed using $\tilde{O}(d)$ field operations in $\mathbb{K}$. Here the *soft-Oh* notation $\tilde{O}(E)$ is an abbreviation for $E(\log E)^{O(1)}$, for any expression E.

Given two bivariate polynomials $P, Q \in \mathbb{K}[x, y]$ of respective total degrees $d \geqslant e$, their resultant $\mathrm{Res}_y(P, Q)$ in y can be computed in time $\tilde{O}(d^2 e)$; e.g. see [20, Theorem 25] and references in that paper. If $d = e$, then this corresponds to a complexity exponent of $3/2$ in terms of input/output size. An important open question in algebraic complexity theory is whether this exponent can be lowered.

In the present paper, we consider the case when P and Q are "sufficiently generic". If the coefficients of P and Q are chosen at random in a finite field $\mathbb{K} = \mathbb{F}_q$ with sufficiently many elements, then this will be the case with high probability. Under a suitable hypothesis of "grevlex-lex-generic position" (defined below) and assuming the *random access memory* (RAM) bit complexity model, our main result is the following theorem:

THEOREM 1. *Let $\epsilon > 0$ be a fixed rational number. Let $P, Q \in \mathbb{K}[x, y]$ be two polynomials of respective total degrees $d \geqslant e$ over a finite field $\mathbb{K} = \mathbb{F}_q$. If P and Q are in grevlex-lex-generic position, then $\mathrm{Res}_y(P, Q)$ can be computed in expected time*

$$O((d e \log q)^{1+\epsilon}) + \tilde{O}(d^2 \log q),$$

using a randomized algorithm of Las Vegas type.

*. This paper is part of a project that has received funding from the French "Agence de l'Innovation de Défense".

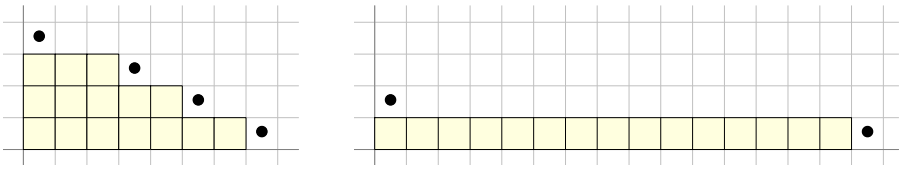

Figure 1. Illustration of the Gröbner stairs for P and Q in generic position with respect to $<_{\text{grevlex}}$ (left) and $<_{\text{lex}}$ (right) in the case when $d=5$ and $e=3$.

A major result in a similar direction has recently been obtained by Villard [25]. For a general effective field $\mathbb{K}$, and under different genericity assumptions, he proposed an algorithm that computes the resultant in y of two polynomials $P,Q\in\mathbb{K}[x,y]$ of degree d_x in x and degree d_y in y using $(d_x d_y^{2-1/\omega})^{1+o(1)}$ operations in $\mathbb{K}$. Here ω is the usual exponent for matrix multiplication (such that two $n\times n$ matrices over $\mathbb{K}$ can be multiplied using $O(n^\omega)$ operations in $\mathbb{K}$). Le Gall has shown in [19] that one may take $\omega<2.373$.

Relationship to Gröbner bases Resultants are related to elimination theory. Throughout the paper, we assume that the reader is familiar with the basic theory of Gröbner bases, as found in standard text books [4, 10], so we only briefly recall basic terminology.

Let $\mathbb{K}$ still be a general effective field. Two common monomial orderings on the polynomial ring $\mathbb{K}[x,y]$ are the lexicographical ordering $<_{\text{lex}}$ and the reverse graded lexicographical ordering $<_{\text{grevlex}}$ defined by

$$\begin{aligned} x^i y^j <_{\text{lex}} x^k y^l &\iff j<l \vee (j=l \wedge i<k)\\ x^i y^j <_{\text{grevlex}} x^k y^l &\iff (i+j<k+l \vee (i+j=k+l \wedge j<l)). \end{aligned}$$

We say that P and Q are in *lex-generic* (resp. *grevlex-generic*) position if the leading monomials of the reduced Gröbner basis of I with respect to $<_{\text{lex}}$ (resp. $<_{\text{grevlex}}$) coincide with the ones that we would obtain when taking symbolic parameters for the coefficients of P and Q; see Figure 1. We say that P and Q are in *grevlex-lex-generic position* when they are both in lex-generic and grevlex-generic position. Notice that we do not require the ideal $I:=(P,Q)$ to be radical over the algebraic closure of $\mathbb{K}$.

The relationship between resultants and Gröbner bases is the following: if P and Q are in lex-generic position, then the reduced Gröbner basis of I with respect to $<_{\text{lex}}$ consists of the minimal polynomial of x, which is a constant multiple of $\text{Res}_y(P,Q)$, and the polynomial $y-U(x)$ for some $U\in\mathbb{K}[x]$ with $\deg U<de$; see section 4.1.

Assuming that P and Q are in grevlex-generic position, the recent result from [12] is an algorithm to compute a *concise representation* [12, Definition 14] of the Gröbner basis of I with respect to $<_{\text{grevlex}}$ using $\tilde{O}(d^2)$ operations in $\mathbb{K}$, while conserving its main properties; see section 3. Note that the Gröbner basis in its usual representation generically requires $\Theta(d^2e)$ storage, so its computation is too expensive for our purposes.

If we were able to rapidly convert a Gröbner basis for $<_{\text{grevlex}}$ into a new one for $<_{\text{lex}}$, then this would allow us to compute resultants in softly linear time. Unfortunately, traditional "change-of-ordering" algorithms such as the FGLM algorithm [6, 7] rely on linear algebra, and do not run in softly linear time. For our proof of Theorem 1 in section 6, we will instead rely on a bivariate counterpart of Kedlaya–Umans' algorithm for modular composition [16]. This technique does not work for general effective fields $\mathbb{K}$, which explains the restriction to the case when $\mathbb{K}=\mathbb{F}_q$ is a finite field in Theorem 1.

Specific fields For $\mathbb{K} = \mathbb{Q}$, Mehrabi and Schost [21] showed how to compute a basis of $I := (P, Q)$ with respect to $<_{\text{lex}}$ by means of a probabilistic algorithm of Monte Carlo type with a nearly optimal bit complexity bound. Their bound is optimal when the coefficients grow as expected in the worst case and their method depends on Kedlaya–Umans as well. This approach has been generalized to higher dimensions in [14]. The probabilistic aspect comes from the need of a sufficiently generic linear change of the variables. The best known deterministic complexity bound can be found in [2].

Over finite fields, Poteaux and Schost [22, Theorem 1.2] achieved the computation of bivariate lexicographical bases with bit complexity $(d e)^{1+\epsilon}\, \tilde{O}(\log q)$ in the special case when P or Q belongs to $\mathbb{F}_q[y]$, provided that the underlying characteristic is greater than $d e$, and that I is radical, yet without genericity assumption. Their method extends to an arbitrary number of variables.

Outline of our contribution The proof of Theorem 1 relies on a sequence of reductions, using a novel combination of classical and more recent techniques. In sections 2 and 3, we first recall basic notations and the required results from [12] about concise Gröbner bases.

Assuming from there on that P and Q are in grevlex-lex-generic position, we recall in section 4 how to reduce the computation of the resultant to the computation of the minimal polynomial of the multiplication endomorphism by $x + I$ in $\mathbb{A} := \mathbb{K}[x, y]/I$. This minimal polynomial can be computed with high probability using Wiedemann's algorithm, provided that we have an algorithm for the transposed map of evaluating a univariate polynomial at $x + I$ in $\mathbb{A}$. This kind of strategy has been used several times before in computer algebra [15, 22, 23, 24]; see also [10, chapter 12, section 4].

The evaluation of a univariate polynomial at $x + I$ in $\mathbb{A}$ can be regarded as a bivariate modular composition problem. Exploiting the fact that multiplication in $\mathbb{A}$ is fast (thanks to the concise Gröbner basis representation), we show in section 5 how to reduce this problem to multivariate multipoint evaluation. For this reduction, we mostly follow Kedlaya and Umans, along the same lines as in the proof of [16, Theorem 3.1] for univariate modular composition; refinements can be found in [13].

At that point, we restrict ourselves to the case when $\mathbb{K}$ is a finite field, so as to benefit from Kedlaya and Umans' fast algorithm for multipoint evaluation and its transpose [16]. In section 6, this allows us to conclude the proof of Theorem 1.

The final section 7 contains a few further notes and directions for future research. In particular, we show that the full lexicographical Gröbner basis can be computed in a similar time as the resultant, with ideas similar to [22] and [8, Algorithm 2]. Finally we quantify the genericity hypotheses in a more precise manner.

2. PRELIMINARIES

Computational model Throughout this paper, $\mathbb{K}$ is an effective field. Most of our algorithms work in the algebraic complexity models of straight-line programs (SLPs) or computation trees [3], in which execution times correspond to the required number of field operations in $\mathbb{K}$. The genericity assumptions imply that non-trivial zero tests always fail, so the straight-line program framework actually suffices.

In section 6, where we prove Theorem 1, we specialize $\mathbb{K}$ to become a finite field $\mathbb{F}_q$. From that point on, we assume a RAM bit complexity model and recall that field operations in $\mathbb{F}_q$ can be performed in softly linear time $\tilde{O}(\log q)$.

Transposition principle Given a finite dimensional $\mathbb{K}$-vector space V of $\mathbb{K}[z_1,\dots,z_\nu]$ that admits $V \cap z_1^{\mathbb{N}} \cdots z_\nu^{\mathbb{N}}$ as a basis, it is convenient to mentally represent elements of V as column vectors with respect to this basis and linear forms $\lambda: V \to \mathbb{K}$ as row vectors. Linear maps between two vector spaces V, W of this type correspond to matrices.

Writing V^* for the set of linear forms $\lambda: V \to \mathbb{K}$, the *transpose* of a linear map $L: V \to W$ is the linear map $L^*: W^* \to V^*$ such that $L^*(\lambda)(f) = \lambda(L(f))$ for all $f \in V$. If L can be computed by a linear SLP over $\mathbb{K}$ of length ℓ, then it is well-known [3, Theorem 13.20] that L^* can be computed by an SLP of length $\ell + O(\dim_{\mathbb{K}} V + \dim_{\mathbb{K}} W)$. This "transposition principle" is in general easy to be put into practice on concrete programs, as exemplified in [1]. Roughly speaking, the program is regarded as a composition of individual steps that are easy to transpose. For the transposition of a composition $L \circ K$, where $K: U \to V$ is another $\mathbb{K}$-linear map, we next apply the usual formula $(L \circ K)^* = K^* \circ L^*$.

Gröbner bases Given indeterminates $z_1,\dots,z_\nu$ and positive integers $n_1,\dots,n_\nu$, we define

$$\mathbb{K}[z_1,\dots,z_\nu]_{n_1,\dots,n_\nu} := \{A \in \mathbb{K}[z_1,\dots,z_\nu] : \deg_{z_1} A < n_1, \dots, \deg_{z_\nu} A < n_\nu\}.$$

Consider a Gröbner basis G of an ideal $I \subseteq \mathbb{K}[z_1,\dots,z_\nu]$ for some term ordering on the set of monomials $z_1^{\mathbb{N}} \cdots z_\nu^{\mathbb{N}} := \{z_1^{i_1} \cdots z_\nu^{i_\nu} : i_1,\dots,i_\nu \in \mathbb{N}\}$. We write $\mathbb{K}[z_1,\dots,z_\nu]_G$ for the $\mathbb{K}$-vector space of polynomials $f \in \mathbb{K}[z_1,\dots,z_\nu]$ that are reduced with respect to G. The reduced monomials in $B_G := \mathbb{K}[z_1,\dots,z_\nu]_G \cap z_1^{\mathbb{N}} \cdots z_\nu^{\mathbb{N}}$ form a basis for $\mathbb{K}[z_1,\dots,z_\nu]_G$ and correspond to the monomials "under the Gröbner stairs". In other words, B_G consists of the monomials that are not divisible by the leading monomial of an element in G. We also write

$$\rho_G: \mathbb{K}[z_1,\dots,z_\nu] \to \mathbb{K}[z_1,\dots,z_\nu]_G$$

for the map that computes the normal form of a polynomial $f \in \mathbb{K}[z_1,\dots,z_\nu]$ with respect to G, i.e. the unique polynomial $\rho_G(f)$ in $\mathbb{K}[z_1,\dots,z_\nu]_G$ such that $f - \rho_G(f) \in I$.

3. CONCISE REPRESENTATION OF THE QUOTIENT ALGEBRA

In the remainder of this paper, let $P, Q \in \mathbb{K}[x,y]$ be two polynomials of total degrees $d \geqslant e$, in grevlex-lex-generic position. We write $I := (P,Q)$ for the ideal generated by P and Q, and $\mathbb{A} := \mathbb{K}[x,y]/I$ for the corresponding quotient algebra. Let us start by recalling several facts from [12].

Gröbner basis The reduced Gröbner basis G^* of I with respect to $<_{\text{grevlex}}$ consists of polynomials $G_0^*, G_1^*, \dots, G_e^* \in \mathbb{K}[x,y]$ with leading monomials $y^e, x^{d-e+1}y^{e-1}, x^{d-e+3}y^{e-2}, \dots, x^{d+e-1}$; see [9], [12, section 2], and Figure 1.

Concise Gröbner bases [12, section 4 and Theorem 28] Using $\tilde{O}(d^2)$ operations in $\mathbb{K}$, one may compute the concise representation of the Gröbner basis $G = \{G_0, G_1, \dots, G_e\}$ of I with respect to $<_{\text{grevlex}}$. The leading monomials of G_i and G_i^* coincide for $i = 0,\dots,e$, but $G_0,\dots,G_e$ are not necessarily reduced. Furthermore, $G_0,\dots,G_e$ are not explicitly written out (since this typically requires $\Theta(d^2 e)$ coefficients in $\mathbb{K}$); this is why we need to represent G in a concise way, while ensuring that no essential information is lost.

Normal form [12, section 5 and Proposition 31] Given a polynomial $\varphi \in \mathbb{K}[x,y]$ with $\deg_x \varphi \leqslant s$ and $\deg_y \varphi \leqslant t \leqslant s$, we may compute its normal form $\rho_G(\varphi) \in \mathbb{K}[x,y]_G$ with respect to G using $\tilde{O}((s+d)(t+e))$ operations in $\mathbb{K}$. Recall that $\rho_G(\varphi)$ is the unique element in $K[x,y]_G = K[x,y]_{G^*}$ with $\varphi - \rho_x(\varphi) \in I$; in particular, ρ_G and ρ_{G^*} coincide.

Checking the genericity assumption By means of [12, Remark 4, Theorem 28, and Proposition 31], the condition that P and Q are indeed in grevlex-generic position can be checked using $\tilde{O}(d^2)$ operations in $\mathbb{K}$ by running the basis computation and aborting when an irregularity occurs.

Multiplication in the quotient algebra [12, section 6.2 and Theorem 33] Assume that the concise Gröbner basis of I has been computed. We represent elements in the quotient algebra $\mathbb{A} = \mathbb{K}[x,y]/I$ by normal forms in $\mathbb{K}[x,y]_G$. Given $\varphi, \psi \in \mathbb{K}[x,y]_G$, we may now compute $\varphi\,\psi \in \mathbb{K}[x,y]$ using $\tilde{O}(de)$ operations in $\mathbb{K}$, since $\deg_x(\varphi\,\psi) \leqslant 2\,(d+e-2)$ and $\deg_y(\varphi\,\psi) \leqslant 2\,(e-1)$. By what precedes, we may therefore compute $\rho_G(\varphi\,\psi) \in \mathbb{K}[x,y]_G$ in time $\tilde{O}(de)$. In other words, products in $\mathbb{A}$ can be computed in softly linear time.

4. REDUCTION TO BIVARIATE MODULAR COMPOSITION

As above, P and Q are polynomials in $\mathbb{K}[x,y]$ in grevlex-lex-generic position, $I := (P,Q)$, and $\mathbb{A} := \mathbb{K}[x,y]/I$.

4.1. Resultants and minimal polynomials

Consider the $\mathbb{K}$-linear multiplication map $\xi\colon \mathbb{A} \to \mathbb{A}; a \mapsto (x+I)\,a$. It is known that the characteristic polynomial $\chi \in \mathbb{K}[t]$ of this map equals $\alpha \operatorname{Res}_y(P(t,y), Q(t,y))$ for some $\alpha \in \mathbb{K}$; see for instance [5, Proposition 2.7] applied with $n=1$ and $r=1$. When all the zeros of I are regular, χ is separable and the latter property follows more straightforwardly by comparing the sets of roots of $\chi(t)$ and of $\operatorname{Res}_y(P(t,y), Q(t,y))$ via the Stickelberger eigenvalue theorem [18].

On the other hand, since P and Q are in lex-generic position, we have

$$\deg \chi = \dim_{\mathbb{K}} \mathbb{A} = de.$$

We introduce the minimal polynomial $\mu \in \mathbb{K}[t]$ of ξ as the monic polynomial of minimal degree such that $\mu(\xi) = 0$, or, equivalently, $\mu(x) \in I$. In particular, $\mu(x)$ coincides with the unique element of the reduced Gröbner basis of I for $<_{\text{lex}}$ that belongs to $\mathbb{K}[x]$.

LEMMA 2. *With P and Q in grevlex-lex-generic position, we have $\chi = \mu$. In addition, if $|\mathbb{K}| > de$, then $\operatorname{Res}_y(P,Q)$ can be recovered from μ using $\tilde{O}(d^2)$ operations in $\mathbb{K}$.*

Proof. We always have $\mu \mid \chi$. The polynomials μ and χ coincide whenever $\deg \mu = \deg \chi = de$; this is the case if and only if P and Q are in lex-generic position.

Once χ is known, since $|\mathbb{K}| > de$, we may find a $\beta \in \mathbb{K}$ such that $\chi(\beta) \neq 0$ using $\tilde{O}(de)$ operations in $\mathbb{K}$, by means of fast multipoint evaluation. Then the above value α can be computed using $\tilde{O}(d^2)$ further operations, as

$$\alpha = \frac{\chi(\beta)}{\operatorname{Res}_y(P(\beta,y), Q(\beta,y))}.$$

From χ and α, we deduce $\operatorname{Res}_y(P(t,y), Q(t,y)) = \chi(t)/\alpha$. □

The above discussion shows that the computation of $\operatorname{Res}_y(P,Q)$ reduces to the determination of μ.

4.2. Wiedemann's algorithm

We use Wiedemann's algorithm and the transposition principle for the computation of μ, as follows:

- We first select a random linear form $\lambda: \mathbb{K}[x,y]_G \to \mathbb{K}$. More precisely, assuming that $|\mathbb{K}| \geqslant 4de$, we select a finite subset $S \subseteq \mathbb{K}$ of size $|S| \geqslant 4de$ and take λ to be a row vector with random entries from S.
- Taking $N := 2de$, we define the map

$$\begin{aligned} E_{x,G}: \mathbb{K}[t]_N &\longrightarrow \mathbb{K}[x,y]_G \\ \varphi &\longmapsto \rho_G(\varphi(x)). \end{aligned}$$

  We will explain how to evaluate $E_{x,G}$ efficiently in section 5.
- Then we compute the sequence

$$(\lambda \circ E_{x,G})(1), (\lambda \circ E_{x,G})(t), \ldots, (\lambda \circ E_{x,G})(t^{N-1}). \tag{1}$$

  This task is an extension of the usual "power projection" problem to the bivariate case, since it corresponds to one evaluation of the transposed map of $E_{x,G}$:

$$\begin{aligned} E^*_{x,G}: \mathbb{K}[x,y]^*_G &\longrightarrow \mathbb{K}[t]^*_N \\ \lambda &\longmapsto ((\lambda \circ E_{x,G})(1), (\lambda \circ E_{x,G})(t), \ldots, (\lambda \circ E_{x,G})(t^{N-1})). \end{aligned}$$

- Using the fast variant of the Berlekamp–Massey algorithm [10, chapter 12, Algorithm 12.9 combined with the extended half-gcd algorithm], we determine the linear recurrence relation of smallest order $m \leqslant de$ satisfied by the sequence (1). Stated otherwise, this means that we compute the monic polynomial μ^* of minimal degree $m \leqslant de$ such that

$$(\lambda \circ E_{x,G})(\mu^*) = (\lambda \circ E_{x,G})(t\,\mu^*) = \cdots = (\lambda \circ E_{x,G})(t^{N-1-m}\,\mu^*) = 0.$$

- The set of polynomials φ for which $(\lambda \circ E_{x,G})(t^i\,\varphi) = 0$ for $i = 0, \ldots, N-1-\deg\varphi$ is closed under gcds and clearly contains μ. This implies that we always have $\mu^* \mid \mu$. If $\deg\mu^* = de$, then we are sure that $\mu^* = \mu = \chi = \alpha\,\mathrm{Res}_y(P(t,y), Q(t,y))$. The next subsection reminds why this happens with high probability.

4.3. Probability analysis

The above polynomials μ and μ^* coincide if, and only if, $\lambda(E_{x,G}(\mu/\psi)) \neq 0$ for any irreducible factor ψ of μ. Now given an irreducible factor ψ of μ, we have $E_{x,G}(\mu/\psi) \neq 0$. A random linear form $\lambda: \mathbb{K}[x,y]_G \to \mathbb{K}$ as above annihilates a fixed non-zero element of $\mathbb{K}[x,y]_G$ with probability at most $1/|S|$. The probability that λ annihilates $E_{x,G}(\mu/\psi)$ is therefore bounded by $1/|S|$. We conclude that the probability $\mathcal{P}_{\text{success}}$ that none of the $\leqslant de$ irreducible factors ψ of μ annihilates $E_{x,G}(\mu/\psi)$ is at least

$$\mathcal{P}_{\text{success}} \geqslant \left(1 - \tfrac{1}{|S|}\right)^{de} \geqslant \left(1 - \tfrac{1}{4de}\right)^{de} > \frac{3}{4}.$$

The algorithm of the previous subsection and the present probability analysis are summarized in the following lemma.

LEMMA 3. *Assume that P and Q are in grevlex-lex-generic position and that $|\mathbb{K}| \geqslant 4de$. Then the computation of μ takes an expected number of $\tilde{O}(de)$ operations in $\mathbb{K}$ plus an expected number of $O(1)$ computations of sequences* (1) *for different values of λ.*

5. REDUCTION TO MULTIPOINT EVALUATION

In this section, we show how to efficiently reduce the evaluation of $E_{x,G}$ to multivariate multipoint evaluation. We mostly follow the same Kronecker segmentation strategy as in [13, 16, 22] for modular composition.

5.1. Kronecker segmentation

Given an integer ν that will be specified later, let $\delta := \lceil (2de)^{1/\nu} \rceil$ be the smallest integer such that $\delta^\nu \geqslant 2de$. We define the Kronecker map

$$\begin{aligned} K\colon \mathbb{K}[z_1,\ldots,z_\nu]_{\delta,\ldots,\delta} &\longrightarrow \mathbb{K}[t]_{\delta^\nu} \\ z_i &\longmapsto t^{\delta^{i-1}}, \qquad i=1,\ldots,\nu, \end{aligned}$$

as the restriction to $\mathbb{K}[z_1,\ldots,z_\nu]_{\delta,\ldots,\delta}$ of the unique morphism $\check{K}\colon \mathbb{K}[z_1,\ldots,z_\nu] \to \mathbb{K}[t]$ of $\mathbb{K}$-algebras that sends z_i to $t^{\delta^{i-1}}$ for $i=1,\ldots,\nu$. Notice that K is bijective and that both K and its inverse can be computed in linear time with respect to the monomial bases.

Let $D_x := d+e-2$ and $D_y := e-1$ be upper bounds for the degrees in x and y of elements in $\mathbb{K}[x,y]_G$. We may compute

$$g_i := \rho_G(x^{\delta^{i-1}}) \in \mathbb{K}[x,y]_{D_x+1,D_y+1} \text{ for } i=1,\ldots,\nu$$

using binary powering. By what has been said in section 3, this requires $\tilde{O}(de\,\nu \log\delta)$ operations in $\mathbb{K}$. For any $\phi \in \mathbb{K}[t]_{\delta^\nu}$ and $f = K^{-1}(\phi)$, we notice that

$$\rho_G(\phi(x)) \;=\; \rho_G(f(g_1(x,y),\ldots,g_\nu(x,y))).$$

Let $N_x := \nu(\delta-1)D_x + 1$, $N_y := \nu(\delta-1)D_y + 1$, and

$$\begin{aligned} E_g\colon \mathbb{K}[z_1,\ldots,z_\nu]_{\delta,\ldots,\delta} &\longrightarrow \mathbb{K}[x,y]_{N_x,N_y} \\ f &\longmapsto f(g_1(x,y),\ldots,g_\nu(x,y)). \end{aligned}$$

Note that $\deg_x(f(g_1(x,y),\ldots,g_\nu(x,y))) < N_x$ and $\deg_y(f(g_1(x,y),\ldots,g_\nu(x,y))) < N_y$ indeed hold for $f \in \mathbb{K}[z_1,\ldots,z_\nu]_{\delta,\ldots,\delta}$. It follows that

$$E_{x,G} \;=\; \rho_G \circ E_g \circ K^{-1}. \tag{2}$$

5.2. Evaluation-interpolation

We will compute the map E_g using evaluation-interpolation. Assume for the time being that $|\mathbb{K}| \geqslant N_x$ and let $\alpha_1,\ldots,\alpha_{N_x} \in \mathbb{K}$ be pairwise distinct points. Define $\beta_i := \alpha_i$ for $i=1,\ldots,N_y$. Setting $\mathrm{A} := \{\alpha_1,\ldots,\alpha_{N_x}\}$, $\mathrm{B} := \{\beta_1,\ldots,\beta_{N_y}\}$, consider the evaluation map

$$\begin{aligned} E_{\mathrm{A}\times\mathrm{B}}\colon \mathbb{K}[x,y]_{N_x,N_y} &\longrightarrow \mathbb{K}^{\mathrm{A}\times\mathrm{B}} \\ h &\longmapsto (h(\alpha_i,\beta_j))_{(\alpha_i,\beta_j)\in\mathrm{A}\times\mathrm{B}}, \end{aligned}$$

which is a $\mathbb{K}$-linear bijection. Using traditional univariate evaluation-interpolation in each coordinate [10, chapter 10], both $E_{\mathrm{A}\times\mathrm{B}}$ and its inverse $E_{\mathrm{A}\times\mathrm{B}}^{-1}$ can be evaluated using SLPs of length $\tilde{O}(N_xN_y)$ over $\mathbb{K}$. In particular, we can compute

$$\Gamma_i := E_{\mathrm{A}\times\mathrm{B}}(g_i) \in \mathbb{K}^{\mathrm{A}\times\mathrm{B}} \text{ for } i=1,\ldots,\nu \tag{3}$$

in time $\tilde{O}(\nu N_xN_y)$. We next define the map

$$\begin{aligned} E_\Gamma\colon \mathbb{K}[z_1,\ldots,z_\nu]_{\delta,\ldots,\delta} &\longrightarrow \mathbb{K}^{\mathrm{A}\times\mathrm{B}} \\ f &\longmapsto (f((\Gamma_1)_{(\alpha_i,\beta_j)},\ldots,(\Gamma_\nu)_{(\alpha_i,\beta_j)}))_{(\alpha_i,\beta_j)\in\mathrm{A}\times\mathrm{B}}. \end{aligned}$$

Then we have

$$E_g \;=\; E_{\mathrm{A}\times\mathrm{B}}^{-1} \circ E_\Gamma.$$

Combined with (2), this yields

$$E_{x,G} \;=\; \rho_G \circ E_{\mathrm{A}\times\mathrm{B}}^{-1} \circ E_\Gamma \circ K^{-1}. \tag{4}$$

6. FAST COMPUTATION OF RESULTANTS

In section 4, we have reduced the computation of bivariate resultants to the evaluation of the transposed map $E_{x,G}^*$ of $E_{x,G}$. In section 5 the evaluation of $E_{x,G}$ has been reduced to multivariate multipoint evaluation. We first recall how to perform the latter evaluation using algorithms by Kedlaya and Umans. We next combine the above reductions with the transposition principle and prove our main result.

6.1. Fast multipoint evaluation

Kedlaya and Umans designed various algorithms for modular composition and multipoint evaluation [16]. They also gave algorithms for the transposed operations. For the computation of $E_{x,G}$ and its transpose, we will rely on the following result, which is a direct consequence of [16, Corollary 4.5 and Theorem 7.6]:

THEOREM 4. *Let $\epsilon > 0$ be a fixed rational number. Given $f \in \mathbb{F}_q[z_1, \ldots, z_\nu]_{\delta,\ldots,\delta}$ and evaluation points $\gamma_1,\ldots,\gamma_\ell \in \mathbb{F}_q^\nu$ such that $\nu = \delta^{o(1)}$, there exists an algorithm that outputs $f(\gamma_i)$ for $i = 1,\ldots,\ell$, and that runs in time*

$$O(((\delta^\nu + \ell)\log q)^{1+\epsilon}).$$

The transpose of the linear map $f \mapsto (f(\gamma_i))_{1\leqslant i\leqslant \ell}$ can be computed with the same complexity.

Note that [16, Corollary 4.5 and Theorem 7.6] are actually stated in an more precise manner and that we voluntarily simplified the presentation. We also refer to [13] for some recent refinements.

6.2. Evaluating $E_{x,G}$ and $E_{x,G}^*$

Assume from now on that $\mathbb{K} = \mathbb{F}_q$. Before we prove our main result, let us first study the complexity of evaluating the transpose $E_{x,G}^*$ of $E_{x,G}$. Recall that the computation of a sequence as in (1) reduces to one such evaluation.

PROPOSITION 5. *Let $\epsilon > 0$ be a constant, thought to be small. Assume that $q \geqslant \max(4de, N_x)$ and that the concise representation of G and the sets $\Gamma_1,\ldots,\Gamma_\nu$ of (3) have been precomputed. Then one evaluation of $E_{x,G}$ or of its transpose $E_{x,G}^*$ takes time $O((de\log q)^{1+\epsilon})$.*

Proof. We let

$$\nu := \lceil \log\log(d+3) \rceil$$

and verify the following bounds:

$$\begin{aligned}
\delta \;&=\; O((de)^{1/\log\log d}) \\
\delta^\nu \;&\leqslant\; (2(2de)^{1/\nu})^\nu \;=\; 2^{\nu+1}de \;=\; (de)^{1+o(1)} \\
\ell = |\mathrm{A}||\mathrm{B}| \;&\leqslant\; \nu^2\delta^2 D_x D_y \;=\; O((\log\log d)^2 (de)^{1+2/\log\log d}) \;=\; (de)^{1+o(1)}.
\end{aligned}$$

Theorem 4 therefore implies that E_Γ and E_Γ^* can be computed in time $O(de\log q)^{1+\epsilon}$.

In the previous sections, we have already shown that ρ_G, $E_{\mathrm{A}\times\mathrm{B}}^{-1}$ and K^{-1} can be computed using $O((de)^{1+\epsilon})$ operations over $\mathbb{F}_q$. Combining this with (4), it follows that one evaluation of $E_{x,G}$ takes time $O((de\log q)^{1+\epsilon})$.

Using our genericity assumptions, we also observed that these computations can be carried out by linear SLPs over $\mathbb{F}_q$. In view of the transposition principle (see section 2), it follows that ρ_G^*, $(E_{\mathrm{A}\times\mathrm{B}}^{-1})^*$ and $(K^{-1})^*$ can also be computed using $O((de)^{1+\epsilon})$ operations over $\mathbb{F}_q$. Combining this with (4), we conclude that

$$E_{x,G}^* \;=\; (K^{-1})^* \circ E_\Gamma^* \circ (E_{\mathrm{A}\times\mathrm{B}}^{-1})^* \circ \rho_G^*$$

can be computed in time $O((de\log q)^{1+\epsilon})$ as well. □

6.3. Proof of Theorem 1

Let us first reduce to the case when $q \geqslant 4de$ and $q \geqslant N_x$. Let $q' = O(qd^2)$ be the smallest power of q such that $q' \geqslant 4de$ and $q' \geqslant N_x$. Whenever $q' > q$, we replace $\mathbb{F}_q$ by the extension field $\mathbb{F}_{q'}$. Since $\log q' = O(\log q + \log d)$, the construction of $\mathbb{F}_{q'}$ takes bit complexity $(\log d)^{O(1)}\,\tilde{O}(\log q)$, e.g. by using [10, Corollary 14.39], and the overhead involved by this extension only concerns hidden logarithmic factors in the complexity bounds.

Consequently from now $q \geqslant 4de$ and $q \geqslant N_x$ are satisfied. The concise representation of the Gröbner basis of I for $<_{\mathrm{grevlex}}$ takes $\tilde{O}(d^2)$ operations in $\mathbb{F}_q$, as recalled in section 3. With ν as in the proof of Proposition 5, we compute $g_1,\ldots,g_\nu$ in time $\tilde{O}(de\log q)$, and then $\Gamma_1,\ldots,\Gamma_\nu$ in time $O((de\log q)^{1+\epsilon})$, as seen in section 5.2. The minimal polynomial μ of x can therefore be obtained in expected time $O((de\log q)^{1+\epsilon})$, thanks to the combination of Lemma 3 and Proposition 5. We finally deduce $\mathrm{Res}_y(P,Q)$ from μ using Lemma 2 and $\tilde{O}(d^2)$ further operations in $\mathbb{F}_q$.

7. FURTHER NOTES

7.1. Parametrization

Recall from section 4.1 that μ coincides with the unique element in $\mathbb{K}[x]$ of the Gröbner basis G^{lex} of I with respect to $<_{\mathrm{lex}}$ and up to a constant multiple with the resultant $\mathrm{Res}_y(P,Q)$. It is an interesting question whether the complete basis G^{lex} can actually be computed fast.

Now if P and Q are in lex-generic position, then G^{lex} contains exactly one other element besides μ, which is of the form $y - U(x)$ with $U(x) = U_0 + U_1 x + \cdots + U_{de-1}x^{de-1}$. Let us show how to recover this parametrization $y = U(x)$ in expected time

$$O((de\log q)^{1+\epsilon}) + \tilde{O}(d^2\log q).$$

One technique for doing this goes back to Kronecker [17]: it performs a first order deformation in order to compute the minimal polynomial of $x + ty + O(t^2)$ modulo I; see for instance in [11, section 2]. In the present paper, we appeal to an other known method relying once more on power projections; as in [8, Algorithm 2], for instance.

For this purpose, let λ be the linear form that has successfully led to the computation of μ and let $\tilde{\mu}(z) := z^{de}\mu(z^{-1})$ be the reciprocal of μ. We compute the polynomial $\vartheta \in \mathbb{K}[z]$ of degree $< de$ such that ϑ and $\tilde{\mu}$ are coprime and

$$\sum_{i\geqslant 0} (\lambda \circ E_{x,G})(t^i)\, z^i = \frac{\vartheta(z)}{\tilde{\mu}(z)}.$$

Now let $\lambda_y\colon \mathbb{K}[x,y]_G \to \mathbb{K}$ be the linear form that sends v to $\lambda(\rho_G(y\,v))$. This form λ_y can be computed in softly linear time by the transposition principle. Note that the sequence

$$(\lambda_y \circ E_{x,G})(1), (\lambda_y \circ E_{x,G})(t), \ldots, (\lambda_y \circ E_{x,G})(t^{de-1}) \tag{5}$$

may be obtained in the same way as (1), in time $O((de\log q)^{1+\epsilon})$, thanks to Proposition 5. The sequences (1) and (5) are related by the identity

$$\begin{aligned}\sum_{i\geqslant 0} (\lambda_y \circ E_{x,G})(t^i)\, z^i &= \sum_{i\geqslant 0}\sum_{j=0}^{de-1} U_j\, (\lambda \circ E_{x,G})(t^{i+j})\, z^i \\ &= U(z^{-1}) \sum_{i\geqslant 0} (\lambda \circ E_{x,G})(t^i)\, z^i + z^{-1} V(z^{-1}),\end{aligned}$$

where $V \in \mathbb{K}[z]_{de-1}$. It follows that

$$W(z) := z^{de-1}\left(\sum_{i\geqslant 0} (\lambda_y \circ E_{x,G})(t^i)\, z^i\right) \tilde{\mu}(z) = z^{de-1} U(z^{-1})\, \vartheta(z) + z^{de-2} V(z^{-1})\, \tilde{\mu}(z)$$

is a polynomial of degree $<2\,d\,e-1$. Since ϑ and $\tilde{\mu}$ are coprime, we finally obtain the reciprocal $\tilde{U}(z) := z^{de-1} U(z^{-1})$ of U using

$$\tilde{U}(z) = \vartheta(z)^{-1}\, W(z) \bmod \tilde{\mu}(z).$$

This takes $\tilde{O}(d\,e)$ further operations in $\mathbb{K}$.

7.2. On the genericity assumptions

We already noted in section 3 that the grevlex-generic position can be checked using $\tilde{O}(d^2)$ operations in $\mathbb{K}$. Let us now quantify the genericity of the grevlex-generic position. Write $P = \sum_{i+j\leqslant d} P_{i,j}\, x^i y^j$ and $Q = \sum_{i+j\leqslant e} Q_{i,j}\, x^i y^j$. As in [12], we define $\operatorname{Diag} P := \sum_{i=0}^{d} P_{d-i,i}\, z^i$ and $\operatorname{Diag} Q := \sum_{i=0}^{e} Q_{e-i,i}\, z^i$, which both belong to $\mathbb{K}[z]$. By [12, Remark 4] and thanks to the relationship between the Euclidean remainder sequence and the subresultant polynomials (see for instance [20, Corollary 3]), P and Q are in grevlex-generic position if and only if the following conditions are satisfied:

1. $P_{0,d} \neq 0$ and $Q_{0,e} \neq 0$,
2. The subresultant coefficient S_i in degree i of $\operatorname{Diag} P$ and $\operatorname{Diag} Q$ is non-zero for $i = 0,\ldots, e-1$. This subresultant is the determinant of a square matrix of size $d+e-2\,i$ whose entries are coefficients of $\operatorname{Diag} P$ and $\operatorname{Diag} Q$; see for instance [10, chapter 6].
3. The system $P = Q = 0$ admits exactly $d\,e$ solutions counting multiplicities, or equivalently the coefficient R_{de} of x^{de} in $R(x) := \operatorname{Res}_y(P(x,y), Q(x,y))$ is non-zero; see for instance [5, Proposition 2.7] applied with $n=1$ and $r=1$. This coefficient R_{de} has total degree $\leqslant d+e$ in the coefficients of P and Q.

Overall, there exists a polynomial $\mathcal{G}$ in $\mathbb{Z}[(X_{i,j})_{i+j\leqslant d}, (Y_{i,j})_{i+j\leqslant e}]$ of total degree at most

$$2 + \sum_{i=0}^{e-1} (d+e-2\,i) + d + e = 2 + e\,(d+e) - 2\,\frac{e\,(e-1)}{2} = d\,e + e + 2,$$

such that $\mathcal{G}((P_{i,j})_{i+j\leqslant d}, (Q_{i,j})_{i+j\leqslant e}) \neq 0$ implies the grevlex-generic position of P and Q.

In order to show that $\mathcal{G}$ is not the zero polynomial, we construct the auxiliary sequence of polynomials recursively by $A_{i+1} = z\,A_i + A_{i-1}$ and starting with with $A_{-1} := 0$ and $A_0 := 1$, so $A_1 = z$, $A_2 = z^2+1$, $A_3 = z^3+2\,z$, etc. We verify by induction that $\deg A_i = i$ for $i \geqslant -1$. Taking

$$Q := x^e\, A_e(y/x) \text{ and } P := y^{d-e} Q + x^d\, A_{e-1}(y/x), \tag{6}$$

P contains y^d, Q contains y^e, Diag $P = z^{d-e}$ Diag $Q + A_{e-1}(z)$, and Diag $Q = A_e(z)$. In this way $A_{e-1}, \ldots, A_0$ is the Euclidean remainder sequence of Diag P and Diag Q. By [20, Corollary 3] all the subresultant coefficients of Diag P and Diag Q are non-zero. In addition, since P and Q are homogeneous, it is easy to verify that R is a non-zero multiple of x^{de}. Consequently $\mathcal{G}$ is not identically zero.

A sufficient (but non necessary condition) to ensure that P and Q are in lex-generic position is that R is separable. The coefficients of R have degree $\leqslant d+e$ in the coefficients of P and Q. Consequently the discriminant of R, written $\mathcal{L} \in \mathbb{Z}[(X_{i,j})_{i+j\leqslant d}, (Y_{i,j})_{i+j\leqslant e}]$, has total degree $\leqslant (d+e)\,(2\,d\,e-1)$. When $\mathcal{L}((P_{i,j})_{i+j\leqslant d}, (Q_{i,j})_{i+j\leqslant e}) \neq 0$ the lex-generic position holds. To see that $\mathcal{L}$ is not identically zero it suffices to take $P := (x-\beta y)^d - 1$ and $Q := y^e - 1$: then R is separable for sufficiently generic values of β.

7.3. Possible extensions

Our method can be extended in several directions, as we will briefly outline now.

- With more work, we expect that the lex-genericity assumption can be relaxed somewhat, e.g. to the case when the Gröbner basis for $<_{\text{lex}}$ consists of polynomials of degree $O(\log d)$ in y. Indeed, using linear algebra techniques inspired by Wiedemann's algorithm, the idea would be to recover the characteristic polynomial from the minimal polynomial by determining the multiplicities of the square-free factors.
- Similarly, and as already noticed in [12], it might be possible to relax the grevlex-genericity assumption somewhat, e.g. to the case when the Gröbner basis for $<_{\text{grevlex}}$ consists of Q and polynomials with leading monomials of the form $x^{d-e+i+O(\log d)} y^{e-i}$.
- In [16, section 6], Kedlaya and Umans proposed an algebraic algorithm for multivariate multipoint evaluation (and its transpose, in virtue of the transposition principle). These algorithms are mainly interesting in small characteristic, in which case they can be used instead of the ones from Theorem 4. These algebraic algorithms can also be generalized to more general finite fields, provided that one has an operation for the Frobenius map and its inverse.
- Unfortunately, we are not aware of any efficient implementations of Kedlaya–Umans' algorithms; see [13] for a discussion about the (very large) input sizes for which the complexity bounds would become competitive. For the time being, we therefore do not expect Theorem 1 to induce faster practical implementations of bivariate resultants. Nevertheless, it should be noticed that the maps $E_{x,G}$ and E_Γ can both be regarded as black boxes for our algorithm: whenever a faster algorithm for one of these maps does become available, our method might be relevant for practical applications.
- Using Chinese remaindering and rational reconstruction, the approach of this paper to compute lexicographical Gröbner bases should extend to the case when P and Q have coefficients in $\mathbb{Q}$. In the generic case, this would provide an interesting, more direct alternative of Las Vegas type for [21] and [14].

Acknowledgments. We thank the anonymous referees for their useful comments.

BIBLIOGRAPHY

[1] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In Hoon Hong, editor, *ISSAC '03: Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 37–44. New York, NY, USA, 2003. ACM.

[2] Y. Bouzidi, S. Lazard, G. Moroz, M. Pouget, F. Rouillier, and M. Sagraloff. Solving bivariate systems using rational univariate representations. *J. Complexity*, 2016.

[3] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1997.

[4] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2nd edition, 2013.

[5] C. Durvye and G. Lecerf. A concise proof of the Kronecker polynomial system solver from scratch. *Expo. Math.*, 26(2), 2007.

[6] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Sub-cubic change of ordering for Gröbner basis: a probabilistic approach. In K. Nabeshima, editor, *ISSAC '14: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 170–177. New York, NY, USA, 2014. ACM.

[7] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symbolic Comput.*, 16(4):329–344, 1993.

[8] J.-C. Faugère and C. Mou. Sparse FGLM algorithms. *J. Symbolic Comput.*, 80:538–569, 2017.

[9] A. Galligo. A propos du théorème de préparation de Weierstrass. In F. Norguet, editor, *Fonctions de plusieurs variables complexes*, volume 409 of *Lecture Notes in Math.*, pages 543–579. Springer, Berlin, Heidelberg, 1974.

[10] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.

[11] B. Grenet, J. van der Hoeven, and G. Lecerf. Deterministic root finding over finite fields using Graeffe transforms. *Appl. Algebra Engrg. Comm. Comput.*, 27(3):237–257, 2016.

[12] J. van der Hoeven and R. Larrieu. Fast Gröbner basis computation and polynomial reduction for generic bivariate ideals. *Appl. Algebra Engrg. Comm. Comput.*, 30:509–539, 2019.

[13] J. van der Hoeven and G. Lecerf. Fast multivariate multi-point evaluation revisited. *J. Complexity*, 56:101405, 2020.

[14] J. van der Hoeven and G. Lecerf. On the complexity exponent of polynomial system solving. *Found. Comput. Math.*, 2020. `https://doi.org/10.1007/s10208-020-09453-0`.

[15] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.*, 67(223):1179–1197, 1998.

[16] K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.

[17] L. Kronecker. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *J. reine angew. Math.*, 92:1–122, 1882.

[18] D. Lazard. Résolution des systèmes d'équations algébriques. *Theoret. Comput. Sci.*, 15(1):77–110, 1981.

[19] F. Le Gall. Powers of tensors and fast matrix multiplication. In K. Nabeshima, editor, *ISSAC '14: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 296–303. New York, NY, USA, 2014. ACM.

[20] G. Lecerf. On the complexity of the Lickteig–Roy subresultant algorithm. *J. Symbolic Comput.*, 2019.

[21] E. Mehrabi and É. Schost. A softly optimal Monte Carlo algorithm for solving bivariate polynomial systems over the integers. *J. Complexity*, 34:78–128, 2016.

[22] A. Poteaux and É. Schost. Modular composition modulo triangular sets and applications. *Comput. Complex.*, 22(3):463–516, 2013.

[23] V. Shoup. Fast construction of irreducible polynomials over finite fields. *J. Symbolic Comput.*, 17(5):371–391, 1994.

[24] V. Shoup. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation*, ISSAC '99, pages 53–58. New York, NY, USA, 1999. ACM.

[25] G. Villard. On computing the resultant of generic bivariate polynomials. In C. Arreche, editor, *ISSAC '18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, pages 391–398. New York, NY, USA, 2018. ACM.