Fast computation of generic bivariate resultants

Joris van der Hoeven, Grégoire Lecerf

Fast computation of generic bivariate resultants

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

CNRS (UMR 7161, LIX)
Laboratoire d'informatique de l'École polytechnique
Campus de l'École polytechnique
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 March 26, 2019

We prove that the resultant of two "sufficiently generic" bivariate polynomials over a finite field can be computed in quasi-linear time.

KEYWORDS: complexity, algorithm, computer algebra, resultant, elimination, multi-point 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 [9, 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 [14, Theorem 25]. 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 $\mathbb{K}$ contains sufficiently many elements, and if the coefficients of P and Q are chosen at random, 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*

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

using a randomized algorithm of Las Vegas type.

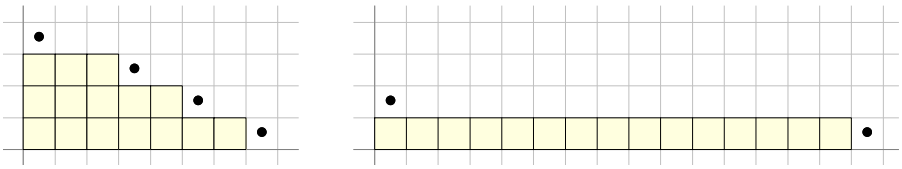

Figure 1. Illustration of the Gröbner stairs for P and Q in generic position with respect to $<_{\text{grevlex}}$ (left) and $<_{\text{lex}}$ (right) in the case when $d=5$ and $e=3$.

A first result in a similar direction has recently been obtained by Villard [17]. 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 in x and degree e in y using $(de^{2-1/\omega})^{1+o(1)}$ 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 [13] that one may take $\omega<2.373$. If $\mathbb{K}=\mathbb{Q}$, then the bit complexity of the bivariate resultant has been studied in [2, 15]. Over finite fields, Poteaux and Schost [16] previously proved Theorem 1 in the very special case when P or Q belongs to $\mathbb{K}[y]$.

Another recent related result [10] concerns the computation of a Gröbner basis for the ideal I generated by P and Q. Before we state it, let us briefly introduce some terminology. Throughout the paper, we assume that the reader is familiar with the basic theory of Gröbner bases, as found in standard text books [4, 9].

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

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

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 (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 for I with respect to $<_{\text{lex}}$ consists of a constant multiple of $\text{Res}_y(P,Q)$ and $y-U(x)$ for some univariate polynomial $U\in\mathbb{K}[x]$ with $\deg U<de$; see section 3.1.

Assuming that P and Q are in grevlex-generic position, the main result from [10] is an algorithm to compute a "terse Gröbner basis" for I with respect to $<_{\text{grevlex}}$ using $\tilde{O}(de)$ operations in $\mathbb{K}$. The actual Gröbner basis may require $\Theta(d^2e)$ storage and its computation is therefore too expensive. This explains why [10] uses a more compact "terse" representation for the Gröbner basis, while conserving its main properties; see section 2.2.

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, known "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, we instead rely on a bivariate counterpart of Kedlaya–Umans' algorithm for modular composition [12]. 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.

Let us briefly outline the structure of this paper and the proof of Theorem 1. In section 2, we introduce further notations and recall the required results about terse Gröbner bases from [10]. Assuming from there on that P and Q are in grevlex-lex-generic position, we recall in section 3 how to reduce the computation of the resultant to the computation of the minimal polynomial of the multiplication endomorphism by $x+I$ in $\mathbb{A} := \mathbb{K}[x,y]/I$, where $I := (P,Q)$. This minimal polynomial can be computed with high probability using the usual Wiedemann algorithm (see for instance [9, chapter 12]), provided that we have an algorithm for the transposed map of evaluating a univariate polynomial at $x+I$ in $\mathbb{A}$. Exploiting the fact that multiplication in $\mathbb{A}$ is fast (thanks to the terse Gröbner basis), we show in section 4 how to adapt Kedlaya–Umans' techniques to reduce this bivariate modular composition problem to multivariate multipoint evaluation. At that point, we can apply Kedlaya–Umans' algorithms for multipoint evaluation and its transpose (also known as power projection); see section 5.

2. TERSE DESCRIPTION OF THE QUOTIENT ALGEBRA

2.1. Notations and conventions

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 5, 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 $\tilde{O}(\log q)$.

Given indeterminates $z_1,\ldots,z_\nu$ and positive integers $n_1,\ldots,n_\nu$, we define

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

Consider a Gröbner basis G of an ideal $I \subseteq \mathbb{K}[z_1,\ldots,z_\nu]$ for some term ordering on the set of monomials $z_1^{\mathbb{N}}\cdots z_\nu^{\mathbb{N}} := \{z_1^{i_1}\cdots z_\nu^{i_\nu} : i_1,\ldots,i_\nu \in \mathbb{N}\}$. We write $\mathbb{K}[z_1,\ldots,z_\nu]_G$ for the $\mathbb{K}$-vector space of polynomials $f \in \mathbb{K}[z_1,\ldots,z_\nu]$ that are reduced with respect to G. The reduced monomials in $B_G := \mathbb{K}[z_1,\ldots,z_\nu]_G \cap z_1^{\mathbb{N}}\cdots z_\nu^{\mathbb{N}}$ form a basis for $\mathbb{K}[z_1,\ldots,z_\nu]_G$ and correspond to the monomials "under the Gröbner stairs". We also write $\rho_G : \mathbb{K}[z_1,\ldots,z_\nu] \to \mathbb{K}[z_1,\ldots,z_\nu]_G$ for the map that computes the normal form of a polynomial $f \in \mathbb{K}[z_1,\ldots,z_\nu]$ with respect to G. In particular, $f - \rho_G(f) \in I$ for all $f \in \mathbb{K}[z_1,\ldots,z_\nu]$.

Given a finite dimensional $\mathbb{K}$-vector space V of $\mathbb{K}[z_1,\ldots,z_\nu]$ 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" can also be applied to more general programs [1]. In particular, in order to transpose a program that computes $L \circ K$, where $K : U \to V$ is another $\mathbb{K}$-linear map, it suffices to transpose the programs for K and L, and then apply the usual formula $(L \circ K)^* = K^* \circ L^*$.

2.2. Terse Gröbner bases

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 [10].

Gröbner basis. The reduced Gröbner basis G^* of I with respect to the grevlex monomial ordering consists of polynomials $G_0^*, G_1^*, \ldots, G_e^* \in \mathbb{K}[x,y]$ with leading monomials y^e, $x^{d-e+1}y^{e-1}, x^{d-e+3}y^{e-2}, \ldots, x^{d+e-1}$; see [8], [10, section 2], and Figure 1.

Terse Gröbner basis. [10, section 4 and Theorem 28] Using $\tilde{O}(d^2)$ operations in $\mathbb{K}$, one may compute a terse Gröbner basis $G = \{G_0, G_1, \ldots, G_e\}$ of I with respect to the grevlex monomial ordering. The leading monomials of G_i and G_i^* coincide for $i = 0, \ldots, e$, but $G_0, \ldots, G_e$ are not necessarily reduced. Furthermore, $G_0, \ldots, G_e$ are not explicitly written out (since this typically requires $\Theta(d^2 e)$ coefficients in $\mathbb{K}$); we rather represent G in a "terse" way that is sufficient for our computational purposes.

Normal form. [10, 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 $\rho_G(\varphi) := \varphi \operatorname{rem} G \in \mathbb{K}[x,y]_G$ 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. [10, 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 $\mathbb{K}$.

Multiplication in the quotient algebra. [10, section 6.2 and Theorem 33] We represent elements in the quotient algebra $\mathbb{A}$ by normal forms in $\mathbb{K}[x,y]_G$. Given $\varphi, \psi \in \mathbb{K}[x,y]_G$, we may 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.

3. 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 $\mathbb{A} := \mathbb{K}[x,y]/I$.

3.1. 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 $r=1$. (Notice that I is not assumed to be radical over the algebraic closure of $\mathbb{K}$, so a direct comparison of the sets of roots of χ and of $\operatorname{Res}_y(P(t,y), Q(t,y))$ cannot replace the use of [5, Proposition 2.7].) On the other hand, since P and Q are in grevlex-generic position, we have

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

Once χ is known and assuming that the cardinality of $\mathbb{K}$ satisfies $|\mathbb{K}| > de$, we may find a $\lambda \in \mathbb{K}$ such that $\chi(\lambda) \neq 0$ using $\tilde{O}(de)$ operations in $\mathbb{K}$, by means of fast multi-point evaluation. Then α can be computed using $\tilde{O}(d^2)$ further operations, as

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

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

The minimal polynomial $\mu \in \mathbb{K}[t]$ of ξ is 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 lexicographical Gröbner basis for I that belongs to $\mathbb{K}[x]$. 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.

Assuming that P and Q are in grevlex-lex-generic position, the above discussion shows that the computation of $\mathrm{Res}_y(P, Q)$ reduces to the determination of μ.

3.2. Wiedemann's algorithm

We use Wiedemann's algorithm for the computation of μ (see for instance [9, chapter 12]), 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 $N := 2de - 1$, we define the map

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

  and 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. We will explain how to evaluate $E_{x,G}$ efficiently in section 4. Then, in section 5, the latter sequence will be obtained by transposing this evaluation algorithm.
- Using the fast variant of the Berlekamp–Massey algorithm [9, 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 $\mu^* = \mu = \chi = \alpha \,\mathrm{Res}_y(P(t, y), Q(t, y))$. We conclude this section with the reminder why this happens with high probability.

3.3. 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

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

4. REDUCTION TO MULTIPOINT EVALUATION

In this section, we show how to efficiently reduce the evaluation of $E_{x,G}$ to multivariate multipoint evaluation.

4.1. 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 $D_x := d+e-2$, $D_y := e-1$, and $g_i := \rho_G(x^{\delta^{i-1}})$ for $i=1,\ldots,\nu$. We may compute $g_1,\ldots,g_\nu \in \mathbb{K}[x,y]_{D_x+1,D_y+1}$ using binary powering. By what has been said in section 2, this requires $\tilde{O}(d^2\nu\log\delta)$ 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

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

It follows that

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

4.2. 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 [9, 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 $\Gamma_i := E_{\mathrm{A}\times\mathrm{B}}(g_i) \in \mathbb{K}^{\mathrm{A}\times\mathrm{B}}$ for $i=1,\ldots,\nu$ 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

$$E_{x,G} \;=\; \rho_G \circ E_{\mathrm{A}\times\mathrm{B}}^{-1} \circ E_\Gamma \circ K^{-1}. \tag{3}$$

5. FAST COMPUTATION OF RESULTANTS

We have reduced the computation of bivariate resultants to the evaluation of $E^*_{x,G}$. In view of (3), it remains to be shown how to compute E_Γ and E^*_Γ. We first recall how to do this using algorithms by Kedlaya and Umans. We then prove our main result.

5.1. Fast multipoint evaluation

Kedlaya and Umans designed various algorithms for modular composition and multipoint evaluation [12]; see also [11]. They also gave algorithms for the transposed operation, called *power projection*. For the computation of $E_{x,G}$ and its transpose, we will rely on the following result, which is a direct consequence of [12, Corollary 4.5 and Theorem 7.6]:

THEOREM 2. *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*

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

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

5.2. Proof of the main theorem

We are now in a position to prove our main result. Let $\epsilon > 0$ be a constant, thought to be small, and take

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

Let $q' = O(q d^2)$ be the smallest power of q such that $q' \geqslant 4de$ and $q' \geqslant N_x$. If $q < q'$, then we replace $\mathbb{K}$ by the extension field $\mathbb{F}_{q'}$ in order to ensure that both the hypotheses $|\mathbb{K}| \geqslant 4de$ and $|\mathbb{K}| \geqslant N_x$ from sections 3.2 and 4.2 are satisfied. We next verify the following bounds:

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

The construction of $\mathbb{F}_{q'}$ takes bit complexity $(\log d)^{O(1)} \tilde{O}(\log q)$, e.g. by using [9, Corollary 14.39]. Altogether, Theorem 2 therefore implies that E_Γ and E^*_Γ can be computed in time $(de \log q)^{1+\epsilon}$.

In the previous sections, we have already shown that ρ_G, $E^{-1}_{\mathrm{A}\times\mathrm{B}}$ and K^{-1} can be computed using $(de)^{1+o(1)}$ operations over $\mathbb{F}_{q'}$, provided that the terse Gröbner basis G and $\Gamma_1, \ldots, \Gamma_\nu$ have been precomputed. 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 aforementioned transposition principle, it follows that ρ^*_G, $(E^{-1}_{\mathrm{A}\times\mathrm{B}})^*$ and $(K^{-1})^*$ can also be computed using $(de)^{1+o(1)}$ operations over $\mathbb{F}_{q'}$. Combining this with (3), it follows that

$$E^*_{x,G} = (K^{-1})^* \circ E^*_\Gamma \circ (E^{-1}_{\mathrm{A}\times\mathrm{B}})^* \circ \rho^*_G$$

can be computed in time $(de \log q)^{1+\epsilon}$. We use this algorithm for the computation of the sequence (1). With probability $> 3/4$, this allows us to recover the resultant of P and Q. Altogether, this gives a probabilistic Las Vegas algorithm to compute $\mathrm{Res}_y(P, Q)$ in expected time $(de \log q)^{1+\epsilon}$. Adding the cost $\tilde{O}(d^2 \log q)$ of the precomputation of G and $\Gamma_1, \ldots, \Gamma_\nu$, this completes the proof of Theorem 1.

5.3. Variants

Our method admits variants that we briefly outline now.

- With more work, it should be possible to relax the lex-genericity assumption somewhat, e.g. to the case when the Gröbner basis for $<_{\mathrm{lex}}$ consists of polynomials of degree $O(\log d)$ in y. Indeed, using linear algebra techniques inspired by Wiedemann's algorithm, the idea is to recover the characteristic polynomial from the minimal polynomial by determining the multiplicities of the square-free factors.
- Similarly, and as already noticed in [10], it should be possible to relax the grevlex-genericity assumption somewhat, e.g. to the case when the Gröbner basis for $<_{\mathrm{grevlex}}$ consists of Q and polynomials with leading monomials of the form $x^{d-e+i+O(\log d)}y^{e-i}$.
- In [12, 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 2. 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.
- Through appropriate use of the Chinese remainder theorem, Theorem 1 can also be used for the quasi-optimal computation of generic bivariate resultants with integer coefficients. Using the technique of rational number reconstruction [9, chapter 5], a similar remark holds for coefficients in $\mathbb{Q}$.
- Unfortunately, we are not aware of any efficient implementations of Kedlaya–Umans' algorithms; see [11] for a discussion. 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.

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] 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.

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

[10] J. van der Hoeven and R. Larrieu. Fast Gröbner basis computation and polynomial reduction for generic bivariate ideals. Technical report, HAL, 2018. `http://hal.archives-ouvertes.fr/hal-01770408`, version 2.

[11] J. van der Hoeven and G. Lecerf. Fast multivariate multi-point evaluation revisited. Technical report, HAL, 2018. `http://hal.archives-ouvertes.fr/hal-01848571`.

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

[13] 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.

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

[15] 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.

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

[17] 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.