On sparse interpolation of rational functions and gcds[*][†]

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 November 11, 2020

In this note, we present a variant of a probabilistic algorithm by Cuyt and Lee for the sparse interpolation of multivariate rational functions. We also present an analogous method for the computation of sparse gcds.

1. INTRODUCTION

Let $f \in \mathbb{K}(x_1,\ldots,x_n)$ be a rational function over an effective field $\mathbb{K}$, presented as a blackbox that can be evaluated at points in $\mathbb{K}^n$. The problem of *sparse interpolation* is to recover f in its usual representation, as a quotient of the form

$$f = \frac{A}{B}, \qquad A = \sum_{1 \leqslant i \leqslant s_A} a_i \boldsymbol{x}^{\boldsymbol{\alpha}_i}, \qquad B = \sum_{1 \leqslant i \leqslant s_B} b_i \boldsymbol{x}^{\boldsymbol{\beta}_i},$$

where $a_i \in \mathbb{K} \setminus \{0\}$ and $\boldsymbol{\alpha}_i \in \mathbb{N}^n$ for $i = 1,\ldots,s_A$, where $b_i \in \mathbb{K} \setminus \{0\}$ and $\boldsymbol{\beta}_i \in \mathbb{N}^n$ for $i = 1,\ldots,s_B$, where $\gcd(A,B) = 1$, and where $\boldsymbol{x}^{\boldsymbol{\gamma}} = x_1^{\gamma_1} \cdots x_n^{\gamma_n}$ for any $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_n) \in \mathbb{N}^n$.

Here, “sparse” means that we use a sparse representation for multivariate polynomials: only the set of the non-zero terms is stored and each term is represented by a pair made of a coefficient and an exponent vector. When using a sparse representation, the natural complexity measures are the number of non-zero terms and the maximum bit size of the exponents. On the contrary, a dense representation of a polynomial $A \in \mathbb{K}[x_1,\ldots,x_n]$, say of total degree d, is the vector of all its coefficients up to degree d for a prescribed monomial ordering.

The sparse interpolation of a blackbox polynomial f has been widely studied in computer algebra, and very efficient algorithms are known. In this paper, we will use this polynomial case as a basic building block, with an abstract specification. We refer to [1, 4, 8, 9, 13, 17] for state of the art algorithms for polynomial sparse interpolation and further historical references.

*. This paper is part of a project that has received funding from the French “Agence de l'innovation de défense”.

†. This article has been written using GNU $\mathrm{T_EX_{MACS}}$ [8].

Our main problem here is to find an efficient reduction of the general sparse interpolation problem for rational functions to the special case of polynomials. We focus on the case when the total degrees d_A and d_B are "modest" with respect to the total number of terms $s_A + s_B$. This is typically the case for applications in computer algebra. We also recall that algorithms in this area are usually probabilistic, of Monte Carlo type; our new algorithms are no exception. In a similar vein, we assume that the cardinality of $\mathbb{K}$ is "sufficiently large", so that random points in $\mathbb{K}$ are "generic" with high probability.

One efficient reduction from sparse rational function interpolation to sparse polynomial interpolation was proposed by Cuyt and Lee [2]. Our new method is a variant of their algorithm; through the use of projective coordinates, we are able to prove a better complexity bound: see section 3.

Computations with rational functions are intimately related to gcd computations. In section 4, we show that our approach also applies to gcd computations: given a blackbox function for the computation of two sparse polynomials P and Q, we present a new complexity bound for the sparse interpolation of $\gcd(P, Q)$. We also present an alternative heuristic algorithm for the case when P and Q are given explicitly, in sparse representation.

2. Preliminaries

For simplicity, all time complexities are measured in terms of the required number of operations in $\mathbb{K}$. Since our algorithms are probabilistic, we assume that a random element in $\mathbb{K}$ can be produced in unit time. Our complexity bounds are valid in several complexity models. For definiteness, one may assume a probabilistic RAM model that supports operations in $\mathbb{K}$.

2.1. On the complexity of sparse polynomial interpolation

The algorithms in this paper all rely on a given base algorithm for the interpolation of sparse polynomials. For a polynomial in $\leqslant n$ variables of total degree $\leqslant d$ with $\leqslant s$ terms, given by a blackbox function f, we assume that this algorithm requires $\mathsf{E}_{\mathbb{K}}(n,d,s)$ evaluations of f, as well as $\mathsf{S}_{\mathbb{K}}(n,d,s)$ additional operations for the actual interpolation from the latter values of f. We recall that this algorithm is allowed to be probabilistic, of Monte Carlo type.

As a particular example of interest, consider the case when $\mathbb{K} = \mathbb{F}_q$, and let $\mathsf{M}_{\mathbb{K}}(s)$ be the cost to multiply two polynomials of degree $\leqslant s$ in $\mathbb{K}[s]$. In a suitable asymptotic region where d and n do not grow too fast, Prony's traditional probabilistic geometric progression technique yields

$$\begin{aligned} \mathsf{E}_{\mathbb{F}_q}(n,d,s) &= 2s \\ \mathsf{S}_{\mathbb{F}_q}(n,d,s) &= O(\mathsf{M}_{\mathbb{F}_q}(s) \log s \log q). \end{aligned}$$

For FFT-primes q, one may also apply the tangent Graeffe method, which yields

$$\mathsf{S}_{\mathbb{F}_q}(n,d,s) = O(\mathsf{M}_{\mathbb{F}_q}(s) \log s).$$

We refer to [9] for a survey of recent complexity bounds for sparse polynomial interpolation. In what follows, we always assume that $\mathsf{E}_{\mathbb{K}}(n,d,s)/s$ and $\mathsf{S}_{\mathbb{K}}(n,d,s)/s$ are nondecreasing in s.

2.2. Genericity and random dilatations

Consider a polynomial or rational blackbox function f. We assume that a blackbox function over $\mathbb{K}$ only involves field operations in $\mathbb{K}$ (which includes equality testing and branching, if needed), and that the number of these operations is uniformly bounded.

Efficient algorithms for the sparse interpolation of f typically evaluate f at a specific sequence of points, such as a geometric progression. For the algorithms in this paper, it is convenient to systematically make the assumption that we only evaluate f at points $(a_1,\ldots,a_n)\in\mathbb{K}^n$ with $a_k\neq 0$ for $k=1,\ldots,n$. This assumption holds in particular for points in geometric progressions.

One problem with the evaluation of a blackbox function that involves divisions is that the evaluation may fail whenever a division by zero occurs. An evaluation point for which this happens is colloquially called a *bad point*. Such bad points must be discarded and the same holds for sequences of evaluation points that contain at least one bad point. Of course, it would in principle suffice to choose another sequence of evaluation points. However, in general, sufficiently generic sequences are not readily at our disposal with high probability. This problem may be overcome using randomization, as follows.

Since our blackbox function f only involves a bounded number of field operations in $\mathbb{K}$, the set of bad evaluation points is a constructible subset of $\mathbb{K}^n$. We assume that this subset is not Zariski dense. Consequently, there exists a polynomial $Z\in\mathbb{K}[x_1,\ldots,x_n]$ such that $Z(a_1,\ldots,a_n)=0$ whenever $(a_1,\ldots,a_n)$ is a bad point. Given $(a_1,\ldots,a_n)\in(\mathbb{K}\setminus\{0\})^n$, the probability that a point $(\lambda_1,\ldots,\lambda_n)\in(\mathbb{K}\setminus\{0\})^n$ satisfies

$$Z(a_1\lambda_1,\ldots,a_n\lambda_n)\neq 0$$

is at least

$$1-\frac{\deg Z}{|\mathbb{K}|-1},$$

by the Schwartz–Zippel lemma [5, Lemma 6.44], where $|\mathbb{K}|$ denotes the cardinality of $\mathbb{K}$.

Now assume that we are given a method for the sparse interpolation of any blackbox function with the same support as f. The evaluation points for such a method form a fixed finite subset E of $(\mathbb{K}\setminus\{0\})^n$. If $|\mathbb{K}|$ is sufficiently large, then it follows with high probability that $Z(a_1\lambda_1,\ldots,a_n\lambda_n)\neq 0$ for all $(a_1,\ldots,a_n)\in E$, when taking $(\lambda_1,\ldots,\lambda_n)\in(\mathbb{K}\setminus\{0\})^n$ at random. We may thus interpolate

$$\tilde{f}(x_1,\ldots,x_n):=f(\lambda_1x_1,\ldots,\lambda_nx_n)$$

instead of f, with high probability. Since $\lambda_1,\ldots,\lambda_n$ are non-zero, we may next recover the sparse interpolation of $f(x_1,\ldots,x_n)=\tilde{f}(\lambda_1^{-1}x_1,\ldots,\lambda_n^{-1}x_n)$ itself.

3. RATIONAL FUNCTIONS

This section concerns the sparse interpolation of a rational function $f\in\mathbb{K}(x_1,\ldots,x_n)$ given as a blackbox.

3.1. Related work

The natural idea to interpolate a univariate rational function $f\in\mathbb{K}(x)$ given as a blackbox is to evaluate it at many points and then use Cauchy interpolation [5, Corollary 5.18] in combination with the fast Euclidean algorithm [5, chapter 11]; this idea appears in early works on the topic [14]. If the numerator and denominator of f have respective degrees d_A and d_B, then the total cost decomposes into d_A+d_B+2 evaluations of f and

$$O(\mathsf{M}_{\mathbb{K}}(d_A+d_B)\log(d_A+d_B))=\tilde{O}(d_A+d_B)$$

operations in $\mathbb{K}$ for the Cauchy interpolation. Here $\tilde{O}(\varphi(n))$ is the usual shorthand for $O(\varphi(n)\log^{O(1)}(\varphi(n)))$, as in [5].

Interpolating rational functions seems more difficult than polynomials. In high degree, it is still a challenging problem to design efficient algorithms in terms of the sparse size s_A+s_B of the function. An algorithm with exponential time complexity is known from [6].

In the multivariate case, the interpolation problem can be handled as a dense one with respect to one of the variables and as a sparse one with respect to the other variables. Although no general polynomial complexity bound is known in terms of the sparse size, this approach is extremely useful in practice, as soon as one of the variables has small degree. Roughly speaking, one proceeds as follows: for a fixed point $(a_1,...,a_{n-1})$ we reconstruct the univariate function $f(a_1,\ldots,a_{n-1},x_n)$ with the dense representation in x_n. Then we regard the numerator A and the denominator B as polynomials in $\mathbb{K}[x_1,\ldots,x_{n-1}][x_n]$, and it remains to interpolate their sparse coefficients using sufficiently many evaluation points $(a_1,\ldots,a_{n-1})$.

The difficulty with the latter interpolation is that the values of $A(a_1,\ldots,a_{n-1},x_n)$ and $B(a_1,\ldots,a_{n-1},x_n)$ are only determined up to a scalar in $\mathbb{K}$. In order to compute this scalar, a normalization assumption is necessary. For instance we may require A to be monic in x_n, which can be achieved through a random linear change of coordinates [14]. Another kind of normalization has been proposed by Cuyt and Lee [2]: they introduce a new auxiliary variable to decompose A and B into their homogeneous components. Their approach is sketched below. Let us finally mention [12], which addresses the special case $\mathbb{K}=\mathbb{Q}$ using a specific strategy.

3.2. The algorithm by Cuyt and Lee

An efficient reduction of sparse interpolation from rational functions to polynomials was proposed by Cuyt and Lee in [2]. Their approach starts by introducing a homogenizing variable u and

$$f(x_1u,\ldots,x_nu)=\frac{A_0+A_1(x_1,\ldots,x_n)\,u+\cdots+A_{d_A}(x_1,\ldots,x_n)\,u^{d_A}}{B_0+B_1(x_1,\ldots,x_n)\,u+\cdots+B_{d_B}(x_1,\ldots,x_n)\,u^{d_B}}.$$

With high probability, they next reduce to the case when B_0 is non-zero, by choosing a random point $(\sigma_1,\ldots,\sigma_n)\in\mathbb{K}^n$ and performing a shift:

$$\begin{aligned}\tilde{f}(x_1,\ldots,x_n,u) &:= f(x_1u+\sigma_1,\ldots,x_nu+\sigma_n)\\ &= \frac{\tilde{A}_0+\tilde{A}_1(x_1,\ldots,x_n)\,u+\cdots+\tilde{A}_{d_A}(x_1,\ldots,x_n)\,u^{d_A}}{\tilde{B}_0+\tilde{B}_1(x_1,\ldots,x_n)\,u+\cdots+\tilde{B}_{d_B}(x_1,\ldots,x_n)\,u^{d_B}}.\end{aligned}$$

For a fixed point $(a_1,\ldots,a_n)\in\mathbb{K}^n$, we may consider $\tilde{f}(a_1,\ldots,a_n,u)$ as a univariate rational function in u, which can generically be interpolated from d_A+d_B+2 evaluations for different values of u.

The first main idea from [2] is to normalize the resulting rational function in such a way that the constant coefficient of the denominator is equal to one:

$$\tilde{f}(x_1,\ldots,x_n,u)=\frac{\frac{\tilde{A}_0}{\tilde{B}_0}+\frac{\tilde{A}_1(x_1,\ldots,x_n)}{\tilde{B}_0}u+\cdots+\frac{\tilde{A}_{d_A}(x_1,\ldots,x_n)}{\tilde{B}_0}u^{d_A}}{1+\frac{\tilde{B}_1(x_1,\ldots,x_n)}{\tilde{B}_0}u+\cdots+\frac{\tilde{B}_{d_B}(x_1,\ldots,x_n)}{\tilde{B}_0}u^{d_B}}.$$

This yields blackbox functions that take $a_1, \ldots, a_n$ as input and produce

$$\frac{\tilde{A}_0}{\tilde{B}_0}, \frac{\tilde{A}_1(a_1, \ldots, a_n)}{\tilde{B}_0}, \ldots, \frac{\tilde{A}_{d_A}(a_1, \ldots, a_n)}{\tilde{B}_0}, \frac{\tilde{B}_1(a_1, \ldots, a_n)}{\tilde{B}_0}, \ldots, \frac{\tilde{B}_{d_B}(a_1, \ldots, a_n)}{\tilde{B}_0}$$

as output. We may now use sparse polynomial interpolation for each of these blackbox functions and determine the sparse expressions for the above polynomials. We finally obtain $f(x_1, \ldots, x_n) = \tilde{f}(x_1 - \sigma_1, \ldots, x_n - \sigma_n, 1)$ using a simple shift.

One complication here is that shifts

$$(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n) + (\sigma_1, \ldots, \sigma_n)$$

destroy the sparsity. The second main observation by Cuyt and Lee is that this drawback only occurs for the non-leading terms: the coefficient $\tilde{A}_{d_A}$ (resp. $\tilde{B}_{d_B}$) coincides with A_{d_A} (resp. B_{d_B}). Using this observation recursively in a clever way, it is possible to obtain the non-leading coefficients as well, using at most

$$(d_A + d_B + 2) \max(\mathsf{E}_{\mathbb{K}}(n, d_A, s_A), \mathsf{E}_{\mathbb{K}}(n, d_B, s_B))$$

blackbox evaluations [2, section 2.2]. However, even though sparse interpolations of shifted polynomials can be avoided, the resulting algorithm still requires evaluations of shifted polynomials in their interpolated form: see the remarks at the end of [2, section 2.2]. In fact, the focus of [2] is on minimizing the number of blackbox evaluations; the overall complexity of the algorithm is not analyzed in detail.

3.3. A variant based on projective coordinates

Instead of using the recursive method from [2, section 2.2] for determining the non-leading coefficients, we propose to avoid the mere existence of such terms altogether. At the start of our algorithm, we assume that the total degrees d_A and d_B of the numerator A and the denominator B of f are known. If we are only given a rough upper bound D on these degrees, then we may reduce to this case as follows: take a point $(a_1, \ldots, a_n) \in \mathbb{K}^n$ at random, and interpolate $\tilde{f}(a_1, \ldots, a_n, u)$ as a univariate rational function in u, from $2(D+1)$ evaluations at different values of u. With high probability, d_A and d_B coincide with the degrees of the numerator and denominator of $\tilde{f}(a_1, \ldots, a_n, u)$.

Our key observation is the following: if A and B are homogeneous polynomials then $\tilde{A}_{d_A} = A$ and $\tilde{B}_{d_B} = B$, so Cuyt and Lee's method mostly reduces to two sparse polynomial interpolations. Our method is therefore to force the polynomials A and B to become homogeneous by introducing a new projective coordinate x_0.

More precisely, we introduce the rational function

$$\hat{f}(x_0, \ldots, x_n) := x_0^{d_A - d_B} f(x_1 / x_0, \ldots, x_n / x_0),$$

and we write

$$\hat{f}(x_0, \ldots, x_n) = \frac{\hat{A}(x_0, \ldots, x_n)}{\hat{B}(x_0, \ldots, x_n)},$$

where

$$\hat{A}(x_0, \ldots, x_n) := x_0^{d_A} A(x_1 / x_0, \ldots, x_n / x_0) \text{ and } \hat{B}(x_0, \ldots, x_n) := x_0^{d_B} B(x_1 / x_0, \ldots, x_n / x_0).$$

The numerator $\hat{A}(x_0, \ldots, x_n)$ and denominator $\hat{B}(x_0, \ldots, x_n)$ of $\hat{f}(x_0, \ldots, x_n)$ are both homogeneous and coprime. One evaluation of $\hat{f}$ requires one evaluation of f plus $n + O(\log |d_A - d_B|)$ operations in $\mathbb{K}$.

For some random shift $(\sigma_0,\ldots,\sigma_n)\in\mathbb{K}^{n+1}$, we next consider

$$\begin{aligned}\tilde{A}(x_0,\ldots,x_n,u) &:= \hat{A}(x_0u+\sigma_0,\ldots,x_nu+\sigma_n)\\ &= \tilde{A}_0+\tilde{A}_1(x_0,\ldots,x_n)\,u+\cdots+\tilde{A}_{d_A}(x_0,\ldots,x_n)\,u^{d_A},\\ \tilde{B}(x_0,\ldots,x_n,u) &:= \hat{B}(x_0u+\sigma_0,\ldots,x_nu+\sigma_n)\\ &= \tilde{B}_0+\tilde{B}_1(x_0,\ldots,x_n)\,u+\cdots+\tilde{B}_{d_B}(x_0,\ldots,x_n)\,u^{d_B},\\ \tilde{f}(x_0,\ldots,x_n,u) &:= \hat{f}(x_0u+\sigma_0,\ldots,x_nu+\sigma_n).\end{aligned}$$

One evaluation of $\tilde{f}$ requires one evaluation of $\hat{f}$ plus $2\,(n+1)$ operations in $\mathbb{K}$. Before addressing the interpolation algorithm for f, let us show that $\tilde{A}/\tilde{B}$ is the canonical representative of $\tilde{f}$.

LEMMA 1. *If $\tilde{B}_0\neq 0$, then $\tilde{A}$ and $\tilde{B}$ are coprime.*

Proof. We consider the following weighted degree:

$$\operatorname{wdeg}(x_0^{\alpha_0}\cdots x_n^{\alpha_n}u^e)=\alpha_0+\cdots+\alpha_n-e.$$

For this weight, $\tilde{A}$ and $\tilde{B}$ are quasi-homogeneous, whence so are their respective irreducible factors and therefore their gcd, written $\tilde{H}$. Consequently, there exists a non-negative integer e such that

$$\tilde{H}(x_0,\ldots,x_n,u)=u^e\tilde{H}(x_0u,\ldots,x_nu,1).$$

By construction, $\tilde{A}(x_0,\ldots,x_n,1)$ and $\tilde{B}(x_0,\ldots,x_n,1)$ are coprime, whence $c:=\tilde{H}(x_0,\ldots,x_n,1)$ is a non-zero constant and $\tilde{H}(x_0,\ldots,x_n,u)=c\,u^e$. Since $\tilde{B}_0\neq 0$, we conclude that $e=0$. □

THEOREM 2. *Let $f=A/B\in\mathbb{K}(x_1,\ldots,x_n)$ be a rational blackbox function whose evaluation requires L operations in $\mathbb{K}$. Let $s=s_A+s_B$, where s_A and s_B are bounds for the number of terms of A and B. Let $d=d_A+d_B$, where d_A and d_B are the total degrees of A and B. Using a probabilistic algorithm of Monte Carlo type, the sparse interpolation of f takes at most*

$$(L+4n+\tilde{O}(\log^2 d))\,(d+2)\,\mathsf{E}_{\mathbb{K}}(n+1,d,s)+\mathsf{S}_{\mathbb{K}}(n+1,d,s)+O(ns\log d)$$

operations in $\mathbb{K}$, provided that the cardinality of $\mathbb{K}$ is sufficiently large.

Proof. First of all, a point $(\sigma_0,\ldots,\sigma_n)$ is taken at random, so $\tilde{B}_0=\hat{B}(\sigma_0,\ldots,\sigma_n)\neq 0$ holds with high probability. Let us describe a blackbox for evaluating $\tilde{A}_{d_A}/\tilde{B}_0$ and $\tilde{B}_{d_B}/\tilde{B}_0$ at a point $(a_0,\ldots,a_n)$:

- We perform d_A+d_B+2 evaluations of $\tilde{f}(a_0,\ldots,a_n,u)$ for different values of u, which give rise to at most

$$(L+3n+2+O(\log|d_A-d_B|))\,(d_A+d_B+2)=(L+3n+O(\log d))\,(d+2)$$

  operations in $\mathbb{K}$.
- If the latter evaluations do not fail, then we perform the univariate Cauchy interpolation of $\tilde{f}(a_0,\ldots,a_n,u)$, which takes $O(\mathsf{M}_{\mathbb{K}}(d)\log d)=d\,\tilde{O}(\log^2 d)$ operations in $\mathbb{K}$. If this Cauchy interpolation fails, then the evaluation of $\tilde{A}_{d_A}/\tilde{B}_0$ and $\tilde{B}_{d_B}/\tilde{B}_0$ aborts.

The failure of the Cauchy interpolation of $\tilde{f}(a_0,\ldots,a_n,u)$ depends on the actual procedure being used, but we may design it to succeed on a Zariski dense subset of points $(a_0,\ldots,a_n)$ in $\mathbb{K}^{n+1}$. For instance, with the aforementioned method of [5, Corollary 5.18], the reconstruction succeeds whenever the degrees of the reconstructed numerator and denominator are d_A and d_B, respectively. This holds when $R(a_0,\ldots,a_n)\neq 0$, where

$$R(x_0,\ldots,x_n):=\operatorname{Res}_u(\tilde{A}(x_0,\ldots,x_n,u),\tilde{B}(x_0,\ldots,x_n,u)).$$

Since $\tilde{B}$ is primitive as an element of $\mathbb{K}[x_0,\ldots,x_n][u]$, the gcd of $\tilde{A}$ and $\tilde{B}$ in $\mathbb{K}[x_0,\ldots,x_n,u]$ can be obtained as the primitive representative in $\mathbb{K}[x_0,\ldots,x_n][u]$ of the gcd of $\tilde{A}$ and $\tilde{B}$ in $\mathbb{K}(x_0,\ldots,x_n)[u]$; see [15, chapter IV, Theorem 2.3], for instance. Using Lemma 1, we deduce that R is not identically zero. Consequently, the evaluation of $\tilde{A}_{d_A}/\tilde{B}_0$ and $\tilde{B}_{d_B}/\tilde{B}_0$ at a point $(a_0,\ldots,a_n)$ does not fail on a Zariski dense subset of $\mathbb{K}^{n+1}$.

In order to ensure the sparse polynomial interpolation to succeed with high probability, we appeal to the discussion from section 2.2 for the evaluations of $\tilde{A}_{d_A}/\tilde{B}_0$ and $\tilde{B}_{d_B}/\tilde{B}_0$. More precisely, by taking $(\lambda_0,\ldots,\lambda_n)\in(\mathbb{K}\setminus\{0\})^{n+1}$ at random, we can interpolate $\tilde{A}_{d_A}(\lambda_0 x_0,\ldots,\lambda_n x_n)/\tilde{B}_0$ and $\tilde{B}_{d_B}(\lambda_0 x_0,\ldots,\lambda_n x_n)/\tilde{B}_0$ with high probability. This random dilatation further increases the cost of each blackbox evaluation by $n+1$ operations in $\mathbb{K}$.

The required number of evaluation points $(a_0,\ldots,a_n)$ is

$$\max(\mathsf{E}_{\mathbb{K}}(n+1,d_A,s_A),\mathsf{E}_{\mathbb{K}}(n+1,d_B,s_B))\leqslant\mathsf{E}_{\mathbb{K}}(n+1,d,s).$$

Once all evaluations are done, we need

$$\mathsf{S}_{\mathbb{K}}(n+1,d_A,s_A)+\mathsf{S}_{\mathbb{K}}(n+1,d_B,s_B)\leqslant\mathsf{S}_{\mathbb{K}}(n+1,d,s)$$

further operations to reconstruct the sparse representations of $\tilde{A}_{d_A}(\lambda_0 x_0,\ldots,\lambda_n x_n)/\tilde{B}_0$ and $\tilde{B}_{d_B}(\lambda_0 x_0,\ldots,\lambda_n x_n)/\tilde{B}_0$. The backward dilatation to recover $\tilde{A}_{d_A}(x_0,\ldots,x_n)/\tilde{B}_0$ and $\tilde{B}_{d_B}(x_0,\ldots,x_n)/\tilde{B}_0$ takes

$$O(n(s_A\log d_A+s_B\log d_B))=O(n s\log d)$$

further operations in $\mathbb{K}$.

To conclude the proof, we observe that $\tilde{A}_{d_A}$ and $\tilde{B}_{d_B}$ coincide with $\hat{A}_{d_A}$ and $\hat{B}_{d_B}$. But since $\hat{A}$ are $\hat{B}$ are homogeneous, this means that $\hat{A}/\tilde{B}_0$ and $\hat{B}/\tilde{B}_0$ actually coincide with $\tilde{A}_{d_A}/\tilde{B}_0$ and $\tilde{B}_{d_B}/\tilde{B}_0$. □

Remark 3. One nice aspect of Cuyt and Lee's algorithm is that the number of blackbox evaluations can be of the order of

$$(O(L)+\tilde{O}(d))\,\mathsf{E}_{\mathbb{K}}(n+1,d,s)$$

if the terms of A and B are more or less equally distributed over their homogeneous components. With our method, we always pay the maximal possible overhead $\Theta(d)$ with respect to sparse interpolation of polynomials.

4. SPARSE GCDS

Let $P,Q\in\mathbb{K}[x_1,\ldots,x_n]$ be polynomials that are given through a common blackbox. In this section we show how to adapt the technique from the previous section to the computation of $G:=\gcd(P,Q)$.

4.1. Related work

A classical approach for multivariate gcds relies on evaluation/interpolation for all but one of the variables, say x_n, and thus on the computation of many univariate gcds; see for instance [5, chapter 6]. This turns out to be similar to rational function interpolation: the univariate case is handled using dense representations and a suitable normalization is necessary to recombine the specialized gcds.

In general and without generic coordinates, the gcd G does not admit a natural normalization. This issue has been studied in [3]. Basically one piece of their solution is to consider P and Q as polynomials in $\mathbb{K}(x_1,\ldots,x_{n-1})[x_n]$ and to force G to be monic in x_n. This leads to interpolate rational functions instead of polynomials, and the technique of the previous section may be used. But it is even more efficient to adapt our main idea for rational functions to gcds.

The sparse gcd problem was first investigated by Zippel, who proposed a variable by variable approach [18]. Then, subresultant polynomials turned out to be useful to avoid rational function interpolations. For historical references we refer the reader to [3, 14]. Software implementations and a recent state of the art can be found in [11].

Let us mention that probabilistic sparse gcd algorithms may be turned into Las Vegas type algorithms by interpolating the full Bézout relation, at the price of an increased computational cost; see for instance [5, chapter 6]. In this paper we content ourselves with Monte Carlo algorithms and we focus on gcd computations only.

4.2. A variant based on projective coordinates

Let $P,Q\in\mathbb{K}[x_1,\ldots,x_n]$ be polynomials that are given through a common blackbox, and assume that $G:=\gcd(P,Q)$ contains s terms. We let $d_P:=\deg P$, $d_Q:=\deg Q$, $d:=\deg G$. Using the technique from the previous section we first reduce to the case when P and Q are homogeneous (without increasing their total degrees), so we let:

$$\begin{aligned}\hat{P}(x_0,\ldots,x_n) &:= x_0^{d_P}P(x_1/x_0,\ldots,x_n/x_0)\\ \hat{Q}(x_0,\ldots,x_n) &:= x_0^{d_Q}Q(x_1/x_0,\ldots,x_n/x_0)\\ \hat{G}(x_0,\ldots,x_n) &:= x_0^{d}G(x_1/x_0,\ldots,x_n/x_0).\end{aligned}$$

For $(\sigma_0,\ldots,\sigma_n)\in\mathbb{K}^{n+1}$, we let

$$\begin{aligned}\tilde{P}(x_0,\ldots,x_n,u) &:= \hat{P}(x_0u+\sigma_0,\ldots,x_nu+\sigma_n)\\ \tilde{Q}(x_0,\ldots,x_n,u) &:= \hat{Q}(x_0u+\sigma_0,\ldots,x_nu+\sigma_n)\\ \tilde{G}(x_0,\ldots,x_n,u) &:= \hat{G}(x_0u+\sigma_0,\ldots,x_nu+\sigma_n)\\ &= \tilde{G}_0+\tilde{G}_1(x_0,\ldots,x_n)\,u+\cdots+\tilde{G}_d(x_0,\ldots,x_n)\,u^d.\end{aligned}$$

Example 4. With $n=2$, $P=x_1^2$, $Q=x_1x_2^2$, $\sigma_0=\cdots=\sigma_n=0$, we have $\tilde{P}=x_1^2u^2$ and $\tilde{Q}=x_1x_2^2u^3$. We note that $G=x_1$, and then that $\tilde{G}(x_0,x_1,x_2,u)=x_1u$. However $\gcd(\tilde{P},\tilde{Q})=x_1u^2$.

The latter example shows that we need to take care of the relationship between $\tilde{G}$ and $\gcd(\tilde{P},\tilde{Q})$. This is the purpose of the following lemma, which extends Lemma 1.

LEMMA 5. *If $\hat{P}(\sigma_0,\ldots,\sigma_n)\neq 0$ or $\hat{Q}(\sigma_0,\ldots,\sigma_n)\neq 0$, then $\tilde{G}=\gcd(\tilde{P},\tilde{Q})$.*

Proof. Let $\tilde{H}:=\gcd(\tilde{P},\tilde{Q})$. In the proof of Lemma 1 we have shown that

$$\tilde{H}(x_0,\ldots,x_n,u)=u^e\tilde{H}(x_0u,\ldots,x_nu,1),$$

for some $e\in\mathbb{N}$. By construction, we have $\hat{G}=\gcd(\hat{P},\hat{Q})$, so $\tilde{G}$ divides $\tilde{H}$. Conversely, $\tilde{H}(x_0,\ldots,x_n,1)$ divides $\tilde{G}(x_0,\ldots,x_n,1)$, so there exists a non-zero constant $c\in\mathbb{K}$ with

$$\tilde{G}(x_0,\ldots,x_n,1)=c\tilde{H}(x_0,\ldots,x_n,1).$$

It follows that

$$\tilde{G}(x_0,\ldots,x_n,u)=\tilde{G}(x_0u,\ldots,x_nu,1)=c\tilde{H}(x_0u,\ldots,x_nu,1)=cu^{-e}\tilde{H}(x_0,\ldots,x_n,u).$$

Since $\hat{P}(\sigma_0,\ldots,\sigma_n)\neq 0$ or $\hat{Q}(\sigma_0,\ldots,\sigma_n)\neq 0$, we finally have $\tilde{G}_0=G(\sigma_0,\ldots,\sigma_n)\neq 0$, so that $e=0$. □

From now on we assume that $\hat{P}(\sigma_0,\ldots,\sigma_n)\neq 0$ or $\hat{Q}(\sigma_0,\ldots,\sigma_n)\neq 0$, so that $\tilde{P}$ or $\tilde{Q}$ is primitive as an element in $\mathbb{K}[x_0,\ldots,x_n][u]$. By Lemma 5 and [15, chapter IV, Theorem 2.3], their gcd $\tilde{G}$ in $\mathbb{K}[x_0,\ldots,x_n,u]$ can be obtained as the primitive representative $\tilde{E}$ in $\mathbb{K}[x_0,\ldots,x_n][u]$ of the gcd of $\tilde{P}$ and $\tilde{Q}$ in $\mathbb{K}(x_0,\ldots,x_n)[u]$. We further observe that $\tilde{E}(x_0,\ldots,x_n,0)\in\mathbb{K}\setminus\{0\}$, whence

$$\frac{\tilde{G}(x_0,\ldots,x_n,u)}{\tilde{G}_0}=\frac{\tilde{E}(x_0,\ldots,x_n,u)}{\tilde{E}(x_0,\ldots,x_n,0)}. \qquad (1)$$

This allows us to obtain

$$\frac{\tilde{G}(x_0,\ldots,x_n,u)}{\tilde{G}_0}=1+\frac{\tilde{G}_1(x_0,\ldots,x_n)}{\tilde{G}_0}u+\cdots+\frac{\tilde{G}_d(x_0,\ldots,x_n)}{\tilde{G}_0}u^d$$

as the gcd of $\tilde{P}$ and $\tilde{Q}$ in $\mathbb{K}(x_0,\ldots,x_n)[u]$, and to recover $\hat{G}$ via

$$\frac{\hat{G}(x_0,\ldots,x_n)}{\tilde{G}_0}=\frac{\tilde{G}_d(x_0,\ldots,x_n)}{\tilde{G}_0}.$$

This approach yields the following complexity bound for the sparse interpolation of $\hat{G}(x_0,\ldots,x_n)/\tilde{G}_0$.

THEOREM 6. *Consider a blackbox function that computes the two polynomials P and Q in $\mathbb{K}[x_1,\ldots,x_n]$ using L operations in $\mathbb{K}$. Let d_P, d_Q, and $d\leqslant\min(d_P,d_Q)$ be the respective total degrees of P, Q, and $G:=\gcd(P,Q)$. Let $D:=\max(d_P,d_Q)$. Using a probabilistic algorithm of Monte Carlo type, we may interpolate G using*

$$(L+4n+\tilde{O}(\log^2 D))\,(D+1)\,\mathsf{E}_{\mathbb{K}}(n+1,d,s)+\mathsf{S}_{\mathbb{K}}(n+1,d,s)+O(ns\log d)$$

operations in $\mathbb{K}$, provided that the cardinality of $\mathbb{K}$ is sufficiently large.

Proof. We first take $(\sigma_0,\ldots,\sigma_n)$ at random, so $\hat{P}(\sigma_0,\ldots,\sigma_n)\neq 0$ or $\hat{Q}(\sigma_0,\ldots,\sigma_n)\neq 0$ with high probability. In order to evaluate $\tilde{G}_d(x_0,\ldots,x_n)/\tilde{G}_0$ at a point $(a_0,\ldots,a_n)$ we proceed as follows:

- We evaluate $\tilde{P}(a_0,\ldots,a_n,u)$ and $\tilde{Q}(a_0,\ldots,a_n,u)$ for $D+1$ different values of u. The evaluation of $x_0^{d_P}, x_0^{d_Q}, x_1/x_0,\ldots,x_n/x_0$ takes

  $$n+O(\log d_P+\log d_Q)$$

  operations in $\mathbb{K}$. The evaluation of $x_0u+\sigma_0,\ldots,x_nu+\sigma_n$ involves $2(n+1)$ more operations. Therefore, one evaluation of $\tilde{P}$ and $\tilde{Q}$ amounts to

  $$L+3n+2+O(\log d_P+\log d_Q)=L+3n+O(\log D)$$

  operations in $\mathbb{K}$.
- We interpolate $\tilde{P}(a_0,\ldots,a_n,u)$ and $\tilde{Q}(a_0,\ldots,a_n,u)$, and then compute

  $$g(u):=\gcd(\tilde{P}(a_0,\ldots,a_n,u),\tilde{Q}(a_0,\ldots,a_n,u)).$$

  This requires $O(\mathsf{M}_{\mathbb{K}}(D)\log D)=D\,\tilde{O}(\log^2 D)$ operations in $\mathbb{K}$. By (1), the normalization $g(u)/g(0)$ coincides with $\tilde{G}(a_0,\ldots,a_n,u)/\tilde{G}_0$ whenever $\deg(\tilde{P}(a_0,\ldots,a_n,u))=d_P$, $\deg(\tilde{Q}(a_0,\ldots,a_n,u))=d_Q$, and the subresultant coefficient of degree d of $\tilde{P}(a_0,\ldots,a_n,u)$ and $\tilde{Q}(a_0,\ldots,a_n,u)$ is non-zero: see [5, chapter 6] or [16], for instance.

In this way, we have built a blackbox that evaluates $\tilde{G}_d / \tilde{G}_0$ on a Zariski dense subset of $\mathbb{K}^{n+1}$. In order to ensure that the sparse interpolation of $\tilde{G}_d / \tilde{G}_0$ succeeds with high probability, we appeal to the discussion from section 2.2. Precisely, we take $(\lambda_0,\ldots,\lambda_n) \in (\mathbb{K} \setminus \{0\})^{n+1}$ at random and we interpolate $\tilde{G}_d(\lambda_0 x_0,\ldots,\lambda_n x_n) / \tilde{G}_0$. This random dilatation further increases the cost of each blackbox evaluation by $n+1$ operations in $\mathbb{K}$.

We need $\mathsf{E}_{\mathbb{K}}(n+1,d,s)$ evaluation points, together with $\mathsf{S}_{\mathbb{K}}(n+1,d,s)$ further operations in $\mathbb{K}$, for the sparse interpolation of $\tilde{G}_d / \tilde{G}_0$. Finally, we recover the requested gcd of P and Q from $\hat{G}(x_0,\ldots,x_n) / \tilde{G}_0 = \tilde{G}_d(\lambda_0^{-1} x_0,\ldots,\lambda_n^{-1} x_n, 1) / \tilde{G}_0$, using $O(n s \log d)$ extra operations in $\mathbb{K}$. □

4.3. Another heuristic approach

Let us now focus on the case when P and Q are presented as sparse polynomials instead of blackbox functions (more generally, one may consider the model from [7, section 5.3]). The number of terms of P (resp. of Q) is written s_P (resp. s_Q). The above method has the disadvantage that we need to evaluate P and Q at many shifted points that are generally not (e.g.) in a geometric progression. Such evaluations are typically more expensive, when using a general algorithm for multi-point evaluation, so it is better to avoid shifting altogether in this case.

Now consider the particular case when

$$G(x_1 u,\ldots,x_n u) = G_v(x_1,\ldots,x_n)\, u^v + \cdots + G_d(x_1,\ldots,x_n)\, u^d$$

and G_v contains a single term $c\boldsymbol{x}^{\boldsymbol{\alpha}}$. Then we may use an alternative normalization where we divide by $c\boldsymbol{x}^{\boldsymbol{\alpha}}$ instead of G_0, and multiply by $(x_1 \cdots x_n)^d$ to keep things polynomial. For a random point $(a_1,\ldots,a_n) \in (\mathbb{K} \setminus \{0\})^n$, we may check whether we are in this case (with high probability) by testing whether

$$G_v(a_1,\ldots,a_n)\, G_v(a_1^3,\ldots,a_n^3) = G_v(a_1^2,\ldots,a_n^2)^2.$$

Once $H := (x_1 \cdots x_n)^d G(x_1 u,\ldots,x_n u) / (c\boldsymbol{x}^{\boldsymbol{\alpha}})$ has been interpolated, we recover G as

$$G = \boldsymbol{x}^{\inf(\boldsymbol{\nu}_P,\boldsymbol{\nu}_Q) - \boldsymbol{\nu}_H} H,$$

where $\boldsymbol{x}^{\boldsymbol{\nu}_P}$ stands for the gcd of all monomials occurring in P and similarly for $\boldsymbol{\nu}_Q$ and $\boldsymbol{\nu}_H$. With these modifications, the complexity bound of Theorem 6 becomes

$$(L + 2n + \tilde{O}(\log^2 D))\,(D+1)\,\mathsf{E}_{\mathbb{K}}(n+1,d,s) + \mathsf{S}_{\mathbb{K}}(n+1,d,s) + O(n s \log d).$$

The same approach can be used if G_d contains a single term $c\boldsymbol{x}^{\boldsymbol{\alpha}}$.

In the case when G_v contains several terms, we consider

$$\tilde{G}(x_1,\ldots,x_n) := G(x_1^{w_1} u,\ldots,x_n^{w_n} u) = \tilde{G}_{\tilde{v}}(x_1,\ldots,x_n)\, u^{\tilde{v}} + \cdots + \tilde{G}_{\tilde{d}}(x_1,\ldots,x_n)\, u^{\tilde{d}}$$

for a suitable vector $\boldsymbol{w} = (w_1,\ldots,w_n) \in \mathbb{N}_{>0}^n$ of weights. Whenever $\tilde{G}_v(x_1,\ldots,x_n)$ contains a single term, we may use the above method to compute $\tilde{G}$ and then G. However, the degree $\tilde{d}$ of $\tilde{G}$ is potentially $|\boldsymbol{w}|$ times larger than the degree d of G, where $|\boldsymbol{w}| = \max(w_1,\ldots, w_n)$. This leads to the question how small we can take $|\boldsymbol{w}|$ while ensuring that $\tilde{G}_{\tilde{w}}$ contains a single term with probability at least $1/2$. We conjecture that this is the case when taking $\boldsymbol{w}$ at random with $|\boldsymbol{w}| \geqslant 2d$. In practice, the idea is to try a sequence of random weights $\boldsymbol{w}$ for which $|\boldsymbol{w}|$ follows a slow geometric progression.

Let us provide some partial evidence for our conjecture. In dimension $n=2$, consider the Newton polygon at $(0,0)$ of a polynomial of degree d made of the maximal number of slopes p/q with $p,q\leqslant k$. For each integer $k>0$, let N_k be the number of slopes p/q with $p,q\in\{1,\ldots,k\}$, $\gcd(p,q)=1$, and $\max(p,q)=k$. We have

$$N_1+2N_2+\cdots+dN_d\leqslant 2d.$$

Under this constraint, the sum $N_1+\cdots+N_d$ is maximized by considering Newton polygons that first exhaust all slopes p/q for which $\gcd(p,q)$ is minimal. Now $N_k\leqslant 2k-1$ for all k. Taking m minimal with $1+2\cdot 3+\cdots+m(2m-1)\geqslant 2d$, it follows that

$$N_1+\cdots+N_d\leqslant 1+3+\cdots+(2m-1)=m^2.$$

Now $1+2\cdot 3+\cdots+m(2m-1)\geqslant {}^2\!/_3 m^3$, so $m\leqslant(3d)^{1/3}$ and

$$N_1+\cdots+N_d\leqslant(3d)^{2/3}.$$

On the other hand, for any $k>0$, the number of rational numbers $p/q\in\mathbb{Q}$ with $p,q\in\{1,...,k\}$ is at least

$$k^2-\left\lfloor\frac{k}{2}\right\rfloor^2-\left\lfloor\frac{k}{3}\right\rfloor^2-\cdots-\left\lfloor\frac{k}{k}\right\rfloor^2\geqslant\left(2-\frac{\pi^2}{6}\right)k^2\geqslant\frac{k^2}{3}.$$

Now let $k=\lfloor\sqrt{6}(3d)^{1/3}+1\rfloor$ be the smallest integer with $k^2/3>2(3d)^{2/3}$. For a random weight $(w_1,w_2)\in\{1,\ldots,k\}^2$ with $\gcd(w_1,w_2)=1$, we then conclude that $\tilde{G}_{\tilde{v}}$ has a unique term with probability at least ${}^1\!/_2$.

As a second piece of evidence, let us show that there always *exists* a weight vector $\boldsymbol{w}\in\{0,\ldots,d\}^n$ for which $\tilde{G}_{\tilde{d}}$ contains a unique term. Let $\boldsymbol{\kappa}$ be an element of the support $\operatorname{supp}G:=\{\sigma\in\mathbb{N}^n:G_\sigma\neq 0\}$ of G for which $r:=\sqrt{\kappa_1^2+\cdots+\kappa_n^2}$ is maximal. Then $\operatorname{supp}G$ is included in the ball with center $\mathbf{0}$ and radius r, so the tangent space to this ball at $\boldsymbol{\kappa}$ intersects $\operatorname{supp}G$ only at $\boldsymbol{\kappa}$. Taking $\boldsymbol{w}:=\boldsymbol{\kappa}$ to be the normal vector to this tangent space, it follows that $\tilde{G}_{\tilde{d}}=c\,x_1^{\kappa_1^2}\cdots x_n^{\kappa_n^2}$ for some $c\in\mathbb{K}\setminus\{0\}$.

BIBLIOGRAPHY

[1] A. Arnold, M. Giesbrecht, and D. S. Roche. Faster sparse multivariate polynomial interpolation of straight-line programs. *J. Symbolic Comput.*, 75:4–24, 2016.

[2] A. Cuyt and W.-S. Lee. Sparse interpolation of multivariate rational functions. *Theor. Comput. Sci.*, 412:1445–1456, 2011.

[3] J. de Kleine, M. Monagan, and A. Wittkopf. Algorithms for the non-monic case of the sparse modular gcd algorithm. In *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation*, ISSAC '05, pages 124–131. New York, NY, USA, 2005. ACM.

[4] S. Garg and É. Schost. Interpolation of polynomials given by straight-line programs. *Theor. Comput. Sci.*, 410(27-29):2659–2662, 2009.

[5] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, 3rd edition, 2013.

[6] D. Grigoriev, M. Karpinski, and M. F. Singer. Computational complexity of sparse rational interpolation. *SIAM J. Comput.*, 23(1):1–11, 1994.

[7] J. van der Hoeven. Probably faster multiplication of sparse polynomials. Technical Report, HAL, 2020. `http://hal.archives-ouvertes.fr/hal-02473830`.

[8] J. van der Hoeven. *The Jolly Writer. Your Guide to GNU TeXmacs*. Scypress, 2020.

[8] J. van der Hoeven and G. Lecerf. Sparse polynomial interpolation in practice. *ACM Commun. Comput. Algebra*, 48(3/4):187–191, 2015.

[9] J. van der Hoeven and G. Lecerf. Sparse polynomial interpolation. Exploring fast heuristic algorithms over finite fields. Technical Report, HAL, 2019. `http://hal.archives-ouvertes.fr/hal-02382117`.

[11] J. Hu and M. Monagan. A fast parallel sparse polynomial GCD algorithm. *J. Symbolic Comput.*, 2020. `https://doi.org/10.1016/j.jsc.2020.06.001`.

[12] Q.-L. Huang and X.-S. Gao. Sparse rational function interpolation with finitely many values for the coefficients. In J. Blömer, I. S. Kotsireas, T. Kutsia, and D. E. Simos, editors, *Mathematical Aspects of Computer and Information Sciences. 7th International Conference, MACIS 2017, Vienna, Austria, November 15-17, 2017, Proceedings*, number 10693 in Theoretical Computer Science and General Issues, pages 227–242. Cham, 2017. Springer International Publishing.

[13] M. Javadi and M. Monagan. Parallel sparse polynomial interpolation over finite fields. In M. Moreno Maza and J.-L. Roch, editors, *PASCO '10: Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*, pages 160–168. New York, NY, USA, 2010. ACM.

[14] E. Kaltofen and B. M. Trager. Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators. *J. Symbolic Comput.*, 9(3):301–320, 1990.

[15] S. Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 3rd edition, 2002.

[16] G. Lecerf. On the complexity of the Lickteig–Roy subresultant algorithm. *J. Symbolic Comput.*, 92:243–268, 2019.

[17] D. S. Roche. What can (and can't) we do with sparse polynomials? In C. Arreche, editor, *ISSAC '18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, pages 25–30. New York, NY, USA, 2018. ACM.

[18] R. Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation. EUROSAM '79, An International Symposium on Symbolic and Algebraic Manipulation, Marseille, France, June 1979*, volume 72 of *Lect. Notes Comput. Sci.*, pages 216–226. Springer Berlin Heidelberg, 1979.