Algebraic diagonals and walks: Algorithms, bounds, complexity

CrossMark

Alin Bostan [a], Louis Dumont [a], Bruno Salvy [a,b]

[a] *Inria, France*
[b] *LIP, U. Lyon, CNRS, ENS Lyon, UCBL, France*

A R T I C L E   I N F O

A B S T R A C T

The diagonal of a multivariate power series F is the univariate power series Diag F generated by the diagonal terms of F. Diagonals form an important class of power series; they occur frequently in number theory, theoretical physics and enumerative combinatorics. We study algorithmic questions related to diagonals in the case where F is the Taylor expansion of a bivariate rational function. It is classical that in this case Diag F is an algebraic function. We propose an algorithm that computes an annihilating polynomial for Diag F. We give a precise bound on the size of this polynomial and show that, generically, this polynomial is the minimal polynomial and that its size reaches the bound. The algorithm runs in time quasi-linear in this bound, which grows exponentially with the degree of the input rational function. We then address the related problem of enumerating directed lattice walks. The insight given by our study leads to a new method for expanding the generating power series of bridges, excursions and meanders. We show that their first N terms can be computed in quasi-linear complexity in N, without first computing a very large polynomial equation.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

The *diagonal* of a multivariate power series with coefficients $a_{i_1,\dots,i_k}$ is the univariate power series with coefficients $a_{i,\dots,i}$. Particularly interesting is the class of diagonals of *rational* power series

E-mail addresses: alin.bostan@inria.fr (A. Bostan), louis.dumont@inria.fr (L. Dumont), bruno.salvy@inria.fr (B. Salvy).

(i.e., Taylor expansions of rational functions). In particular, diagonals of *bivariate* rational power series are always roots of non-zero bivariate polynomials (i.e., they are algebraic series) (Pólya, 1921; Furstenberg, 1967). This property persists for multivariate rational power series, but only in positive characteristic, while the converse inclusion – algebraic series being diagonals of rational series – always holds (Furstenberg, 1967; Safonov, 1987; Denef and Lipshitz, 1987). As far as we are aware, the first occurrence of this result in the literature is an article of Pólya's (1921), which deals with a particular class of bivariate rational functions; the proof uses elementary complex analysis. Along the lines of Pólya's approach, Furstenberg (1967) gave a (sketchy) proof of the general result, over the field of complex numbers; the same argument has been enhanced later (Hautus and Klarner, 1971; Stanley, 1999, §6.3). Three more different proofs exist: a purely algebraic one that works over arbitrary fields of characteristic zero (Gessel, 1980, Th. 6.1) (see also Stanley, 1999, Th. 6.3.3), one based on non-commutative power series (Fliess, 1974, Prop. 5), and a combinatorial proof (Bousquet-Mélou, 2006, §3.4.1) that relies on an encoding of the diagonal using unidimensional walks, seen themselves as words of a non-ambiguous context-free language. Various other generalizations are known (Furstenberg, 1967; Deligne, 1984; Haible, 1997; Pochekutov, 2009).

Polynomial equations Despite the richness of the topic and the fact that most proofs are constructive in essence, we were not able to find in the literature any *explicit* algorithm for computing a bivariate polynomial that cancels the diagonal of a general bivariate rational function. We design in Section 5 such an algorithm for computing a polynomial equation for the diagonal of an arbitrary bivariate rational function. We show in Proposition 20 that generically, the size of the minimal polynomial for the diagonal of a rational function is exponential in the degree of the input and that our algorithm computes it in quasi-optimal complexity (Theorem 18).

The algorithm has two main steps that may be of independent interest. The first step is the computation of a polynomial equation for the residues of a bivariate rational function. We propose an efficient algorithm for this task, that is a polynomial-time version of Bronstein's algorithm (Bronstein, 1992); corresponding size and complexity bounds are given in Theorem 8. The second step is the computation of a polynomial equation for the sums of a fixed number of roots of a given polynomial. We design an additive version of the Platypus algorithm (Banderier and Flajolet, 2002, §2.3) and analyze it in Theorem 12.

Recurrences Since it is also classical that algebraic series are differentially finite (i.e., they satisfy linear differential equations with polynomial coefficients), the coefficients of these bivariate diagonals satisfy linear recurrences and this leads to an optimal algorithm for the computation of their first terms (Chudnovsky and Chudnovsky 1986, 1987; Bostan et al., 2007). We show, however, that computing an annihilating polynomial of the diagonal first is usually not the right approach and that a direct computation of the recurrence (Bostan et al., 2010) will be more efficient. For completeness, we mention that in more than two variables, diagonals of rational functions are still differentially finite (Christol, 1985; Lipshitz, 1988) and currently the most efficient algorithm in that situation is that based on the Griffiths–Dwork method (Bostan et al., 2013; Lairez, 2016).

Walks Diagonals of rational functions appear naturally in enumerative combinatorics. In particular, the enumeration of unidimensional walks has been the subject of recent activity, see Banderier and Flajolet (2002) and the references therein. Three generating functions of different types of walks are of interest: the generating series B of bridges, E of excursions and M of meanders (these are defined precisely in Section 6). The algebraicity of these generating functions is classical as well, and related to that of bivariate diagonals. Beyond this structural result, several quantitative and effective results are known. Explicit formulas give the generating functions in terms of implicit algebraic functions attached to the set of allowed steps in the cases of excursions (Bousquet-Mélou and Petkovšek, 2000, §4), (Gessel, 1980), bridges and meanders (Banderier and Flajolet, 2002). Moreover, Bousquet-Mélou gave a tight exponential bound on the degree of the annihilating polynomial in the case of excur-

sions (Bousquet-Mélou, 2008, §2.1), while Banderier and Flajolet designed an algorithm (called the *Platypus Algorithm*) computing it (Banderier and Flajolet, 2002, §2.3).

Our message for these walks is that again, precomputing a polynomial equation is too costly if one is only interested in the enumeration. Instead, we propose to precompute a differential equation for B, that has polynomial size only, to use it for expanding B, and to recover the expansion of E from that of B. For meanders, we compute a polynomial-size differential equation for $\log M$, from which the expansion of M can be computed efficiently. Our algorithms have quasi-linear complexity in the precision of the expansion, while keeping the precomputation step in polynomial complexity (Theorem 24).

Structure of the article After a preliminary section on background and notation, we first discuss several special bivariate resultants of broader general interest in Sections 3 and 4. Next, we consider diagonals, the size of their minimal polynomials and an efficient way of computing annihilating polynomials in Section 5. Finally, we turn to walks in Section 6 and show how to compute the coefficients of the generating functions of excursions and of meanders efficiently.

A preliminary version of this article has appeared at the ISSAC'15 conference (Bostan et al., 2015). In the present version, we give tight bounds in the main results (Theorems 12 and 18), an improved algorithm for the algebraic residues and more detailed proofs throughout.

2. Background and notation

In this section, that might be skipped at first reading, we introduce notation and technical results that will be used throughout the article.

2.1. Notation

In this article, $\mathbb{K}$ denotes a field of characteristic 0, and $\overline{\mathbb{K}}$ an algebraic closure of $\mathbb{K}$. We denote by $\mathbb{K}[x]_n$ the set of polynomials in $\mathbb{K}[x]$ of degree less than n. Similarly, $\mathbb{K}(x)_n$ stands for the set of rational functions in $\mathbb{K}(x)$ with numerator and denominator in $\mathbb{K}[x]_n$, and $\mathbb{K}[[x]]_n$ for the set of power series in $\mathbb{K}[[x]]$ truncated at precision n.

If P is a polynomial in $\mathbb{K}[x, y]$, then its degree with respect to x (resp. y) is denoted $\deg_x P$ (resp. $\deg_y P$). We take the convention that $\deg 0 = -\infty$. The *bidegree* of P is the pair $\text{bideg } P = (\deg_x P, \deg_y P)$. The notation deg without any subscript is used for univariate polynomials. Inequalities between bidegrees are component-wise. The set of polynomials in $\mathbb{K}[x, y]$ of bidegree less than (n, m) is denoted by $\mathbb{K}[x, y]_{n,m}$, and similarly for more variables.

The *valuation* of a polynomial $F \in \mathbb{K}[x]$ or a power series $F \in \mathbb{K}[[x]]$ is its smallest exponent with non-zero coefficient. It is denoted val F, with the convention val $0 = \infty$.

The *reciprocal* of a polynomial $P \in \mathbb{K}[x]$ is the polynomial $\text{rec}(P) = x^{\deg P} P(1/x)$. If $P = c(x - \alpha_1) \cdots (x - \alpha_d)$ with $c \neq 0$ and $\alpha_i \in \overline{\mathbb{K}}$ for all i, the notation $\mathcal{N}(P)$ stands for the generating series of the *Newton sums* of P:

$$\mathcal{N}(P) = \sum_{n \geqslant 0} (\alpha_1^n + \alpha_2^n + \cdots + \alpha_d^n) x^n.$$

A polynomial is called *square-free* when its gcd with its derivative is trivial. A *square-free decomposition* of a non-zero polynomial $Q \in \mathbb{A}[y]$, where $\mathbb{A} = \mathbb{K}$ or $\mathbb{K}[x]$, is a factorization $Q = Q_1^1 \cdots Q_m^m$, with $Q_i \in \mathbb{A}[y]$ square-free, the Q_i's pairwise coprime and $\deg_y(Q_m) > 0$. The corresponding *square-free part* of Q is the polynomial $Q^\star = Q_1 \cdots Q_m$. If Q is square-free then $Q^\star$ does not depend on the square-free decomposition and $Q = Q^\star$.

The coefficient of x^n in a power series $A \in \mathbb{K}[[x]]$ is denoted $[x^n]A$. If $A = \sum_{i=0}^{\infty} a_i x^i$, then $A \bmod x^n$ denotes the polynomial $\sum_{i=0}^{n-1} a_i x^i$. The exponential series $\sum_n x^n/n!$ is denoted $\exp(x)$. The *Hadamard product* of two power series A and B is the power series $A \odot B$ such that $[x^n]A \odot B = [x^n]A \cdot [x^n]B$ for all n.

If $F(x, y) = \sum_{i,j \geqslant 0} f_{i,j} x^i y^j$ is a bivariate power series in $\mathbb{K}[[x, y]]$, the *diagonal* of F, denoted Diag F, is the univariate power series in $\mathbb{K}[[t]]$ defined by Diag $F(t) = \sum_{n \geqslant 0} f_{n,n} t^n$.

2.2. Complexity estimates

We recall classical complexity notation and facts for later use. Let $\mathbb{K}$ be again a field of characteristic zero. Unless otherwise specified, we estimate the cost of our algorithms by counting arithmetic operations in $\mathbb{K}$ (denoted "ops.") at unit cost. The soft-O notation $\tilde{O}(\cdot)$ indicates that polylogarithmic factors are omitted in the complexity estimates (see von zur Gathen and Gerhard, 2003, Def. 25.8 for a precise definition). The *arithmetic size* of an element of $\mathbb{K}$ is 1. That of a univariate polynomial is its degree plus 1 (i.e., we are considering *dense* representations). That of tuples of polynomials is the sum or their sizes, and this defines the size for rational functions and multivariate polynomials. We say that an algorithm has quasi-linear complexity if its complexity is $\tilde{O}(d)$, where d is the maximal arithmetic size of the input and of the output. In that case, the algorithm is said to be *quasi-optimal*.

Univariate operations. Throughout this article we will use the fact that most operations on polynomials, rational functions and power series in one variable can be performed in quasi-linear time. Standard references for these questions are the books by von zur Gathen and Gerhard (2003) and Bürgisser et al. (1997), as well as Schönhage (1982). The needed results are summarized in Fact 1 below.

Fact 1. *The following operations can be performed in $\tilde{O}(n)$ ops. in $\mathbb{K}$:*

(1) *addition, multiplication and differentiation of elements in $\mathbb{K}[x]_n$, $\mathbb{K}(x)_n$ and $\mathbb{K}[[x]]_n$; integration in $\mathbb{K}[x]_n$ and $\mathbb{K}[[x]]_n$;*
(2) *extended gcd, square-free decomposition and resultant in $\mathbb{K}[x]_n$;*
(3) *multipoint evaluation in $\mathbb{K}[x]_n$, $\mathbb{K}(x)_n$ at $O(n)$ points in $\mathbb{K}$; interpolation in $\mathbb{K}[x]_n$ and $\mathbb{K}(x)_n$ from n (resp. $2n - 1$) values at pairwise distinct points in $\mathbb{K}$;*
(4) *inverse, logarithm, exponential in $\mathbb{K}[[x]]_n$ (when defined);*
(5) *conversions between $P \in \mathbb{K}[x]_n$ and $\mathcal{N}(P) \bmod x^n \in \mathbb{K}[x]_n$.*

Multivariate operations. Basic operations on polynomials, rational functions and power series in several variables are hard questions from the algorithmic point of view. For instance, no general quasi-optimal algorithm is currently known for computing resultants of bivariate polynomials, even though in several important cases such algorithms are available (Bostan et al., 2006). Multiplication is the most basic non-trivial operation in this setting. The following result can be proved using Kronecker's substitution; it is quasi-optimal for a fixed number of variables $m = O(1)$. For polynomials with more complicated monomial supports, or when the number of variables grows, more sophisticated techniques apply (Canny et al., 1989; Pan, 1994; Lecerf and Schost, 2003; van der Hoeven and Schost, 2013).

Fact 2. *For fixed m, polynomials in $\mathbb{K}[x_1, \ldots, x_m]_{d_1,\ldots,d_m}$ and power series in $\mathbb{K}[[x_1, \ldots, x_m]]_{d_1,\ldots,d_m}$ can be multiplied using $\tilde{O}(d_1 \cdots d_m)$ ops.*

A related operation is multipoint evaluation and interpolation. The simplest case is when the evaluation points form an m-dimensional tensor product grid $I_1 \times \cdots \times I_m$, where I_j is a set of cardinal d_j; it extends to subgrids of tensor product grids (van der Hoeven and Schost, 2013).

Fact 3. *(Pan, 1994) For fixed m, polynomials in $\mathbb{K}[x_1, \ldots, x_m]_{d_1,\ldots,d_m}$ can be evaluated and interpolated from values that they take on $d_1 \cdots d_m$ points that form an m-dimensional tensor product grid using $\tilde{O}(d_1 \cdots d_m)$ ops.*

Again, the complexity in Fact 3 is quasi-optimal for fixed $m = O(1)$.

A general (although non-optimal) technique to deal with more involved operations on multivariable algebraic objects (e.g., in $\mathbb{K}[x, y]$) is to use (multivariate) evaluation and interpolation on polynomials and to perform operations on the evaluated algebraic objects using Facts 1–3. To put this strategy in practice, the size of the output needs to be well controlled. We illustrate this philosophy

on the example of resultant computation, based on the following easy variation of von zur Gathen and Gerhard (2003, Thm. 6.22).

Fact 4. *Let $P(x, y)$ and $Q(x, y)$ be bivariate polynomials of respective bidegrees (d_x^P, d_y^P) and (d_x^Q, d_y^Q). Then,*

$$\deg \operatorname{Resultant}_y (P(x, y), Q(x, y)) \leqslant d_x^P d_y^Q + d_x^Q d_y^P,$$

and this is an equality whenever one of d_x^Q or d_x^P is zero.

Lemma 5. *Let P and Q be polynomials in $\mathbb{K}[x_1, \ldots, x_m, y]_{d_1, \ldots, d_m, d}$. Then the polynomial*

$$R = \operatorname{Resultant}_y (P, Q)$$

belongs to $\mathbb{K}[x_1, \ldots, x_m]_{D_1, \ldots, D_m}$, where $D_i = 1 + 2(d - 1)(d_i - 1)$. Moreover, the coefficients of R can be computed using $\tilde{O}(2^m d_1 \cdots d_m d^{m+1})$ ops. in $\mathbb{K}$.

Proof. The degrees estimates follow from Fact 4. To compute R, we use an evaluation-interpolation scheme: P and Q are evaluated at $D = D_1 \cdots D_m$ points $(x_1, \ldots, x_m)$ forming an m-dimensional tensor product grid; D univariate resultants in $\mathbb{K}[y]_d$ are computed; R is recovered by interpolation. By Fact 3, the evaluation and interpolation steps are performed in $\tilde{O}(mD)$ ops. The second one has cost $\tilde{O}(dD)$. Using the inequality $D \leqslant 2^m d_1 \cdots d_m d^m$ concludes the proof. □

We conclude this section by recalling two complexity results on bivariate polynomials and rational functions; for proofs, see Lecerf (2008) and Bostan et al. (2010).

Fact 6.

(1) *A square-free decomposition of polynomials in $\mathbb{K}[x, y]_{d_x, d_y}$ can be computed using $\tilde{O}(d_x^2 d_y)$ ops.*
(2) *If $P, Q \in \mathbb{K}[x, y]$ are non-zero coprime polynomials such that $\operatorname{bideg}(P) < \operatorname{bideg}(Q)$ and Q is primitive with respect to y, then a minimal telescoper for P/Q of degree $O(d_x d_y^\star d_y)$ and order at most $d_y^\star$ can be computed using $\tilde{O}(d_x d_y^2 d_y^{\star 3})$ ops., where $(d_x, d_y) = \operatorname{bideg}(Q)$ and $d_y^\star$ is the degree in y of any square-free part of Q.*

Recall that a minimal telescoper for P/Q is a differential operator $L \in \mathbb{K}[x]\langle \partial_x \rangle$ of minimal order such that $L \cdot (P/Q) = \partial_y(g)$ with $g \in \mathbb{K}[x, y]$.

3. Polynomials for residues

3.1. Algorithm

We are interested in a polynomial that vanishes at some or all of the residues of a given rational function. It is a classical result in symbolic integration that in the case of simple poles, there is a resultant formula for such a polynomial, first introduced by Rothstein (1976) and Trager (1976). This was later generalized by Bronstein (1992) to accommodate multiple poles as well. However, as mentioned by Bronstein, the complexity of his method grows exponentially with the multiplicity of the poles. Instead, we develop in this section an algorithm with polynomial complexity.

Let $f = P/Q$ be a non-zero element in $\mathbb{K}(y)$, where P, Q are two coprime polynomials in $\mathbb{K}[y]$. Let also $\hat{Q}$ be a divisor of Q such that $\hat{Q}$ and $Q/\hat{Q}$ are coprime. In our context, $\hat{Q}$ represents the subset of the roots of Q at which we want to compute an annihilating polynomial of the residues. Let $Q_1 Q_2^2 \cdots Q_m^m$ be a square-free decomposition of $\hat{Q}$. For $i \in \{1, \ldots, m\}$, if α is a root of Q_i in an algebraic extension of $\mathbb{K}$, then it is simple and the residue of f at α is the coefficient of t^{-1} in the Laurent expansion of $f(\alpha + t)$ at $t = 0$. Consider the polynomial $V_i(y, t) = (Q_i(y + t) - Q_i(y))/t$. Since α is a simple root of Q_i, V_i satisfies $V_i(\alpha, t) = Q_i(\alpha + t)/t$ and $V_i(\alpha, 0) = Q_i'(\alpha) \neq 0$. Therefore, the rational

---

Algorithm 1 Polynomial canceling the residues.

Algorithm **AlgebraicResidues**$(P/Q, \hat{Q})$

Input   Three polynomials P, Q and $\hat{Q}$ a divisor of Q in $\mathbb{K}[y]$ such that $\hat{Q}$ and $Q/\hat{Q}$ are coprime ($\hat{Q}$ can be Q)
Output   A polynomial in $\mathbb{K}[z]$ canceling the residues of P/Q at the roots of $\hat{Q}$

Compute $Q_1 Q_2^2 \cdots Q_m^m$ a square-free decomposition of $\hat{Q}$;
for $i \leftarrow 1$ to m **do**
  if $\deg_y Q_i = 0$ **then** $R_i \leftarrow 1$
  else
     $U_i(y) \leftarrow Q(y)/Q_i^i(y)$;
     $V_i(y, t) \leftarrow (Q_i(y + t) - Q_i(y))/t$;
     Expand $\frac{P(y+t)}{U_i(y+t)V_i^i(y,t)} = S_0 + \cdots + S_{i-1}t^{i-1} + O(t^i)$;
     Write S_{i-1} as $A_i(y)/B_i(y)$ with A_i and B_i coprime polynomials;
     $R_i(z) \leftarrow \text{Resultant}_y(A_i - zB_i, Q_i)$;
return $R_1 R_2 \cdots R_m$

---

function g defined by $g(y, t) = f(y + t)Q_i^i(y + t)/V_i^i(y, t)$ satisfies $g(\alpha, t) = f(\alpha + t) \cdot t^i$ and has the advantage of being regular at $t = 0$. The residue of f at α may hence be computed as the evaluation at $y = \alpha$ of $[t^{i-1}]g(y, t)$. If this coefficient is denoted $S_{i-1}(y) = A_i(y)/B_i(y)$, with polynomials A_i and B_i, the residue at α is thus a root of $\text{Resultant}_y(A_i - zB_i, Q_i)$. When the multiplicity of the pole $m = 1$, this is exactly the Rothstein–Trager resultant. This computation leads to Algorithm 1, which avoids the exponential blowup of the complexity that would follow from a symbolic precomputation of the Bronstein resultants.

Example 7. Let $d \geqslant 0$ be an integer, and let $G_d(x, y) \in \mathbb{Q}(x)[y]$ be the rational function $y^d/(y - y^2 - x)^{d+1}$. The poles have order $d + 1$. In this example, the algorithm can be performed by hand for arbitrary d: a square-free decomposition has $m = d + 1$ and $Q_m = y - y^2 - x$, the other Q_i's being 1. Then $V_m = 1 - 2y - t$ and the next step is to expand

$$\frac{(y + t)^d}{(1 - 2y - t)^{d+1}} = \frac{(y + t)^d}{(1 - 2y)^{d+1} \left(1 - \frac{t}{1-2y}\right)^{d+1}}.$$

Expanding the binomial series gives the coefficient of t^d as $\frac{A_m}{B_m}$, with

$$A_m = \sum_{k=0}^{d} \binom{d}{k}\binom{d + k}{k} y^k (1 - 2y)^{d-k}, \quad B_m = (1 - 2y)^{2d+1}.$$

The residues are then canceled by $R_m = \text{Resultant}_y(A_m - zB_m, Q_m)$, namely by

$$R_m = (1 - 4x)^{2d+1} z^2 - \left(\sum_{k=0}^{\lfloor d/2 \rfloor} \binom{d}{2k}\binom{2k}{k} x^k\right)^2. \tag{1}$$

(Equality (1) is a consequence of the identity[1] $A_m = \sum_{k=0}^{\lfloor d/2 \rfloor} \binom{d}{2k}\binom{2k}{k}(y - y^2)^k$, which implies A_m mod $Q_m = \sum_{k=0}^{\lfloor d/2 \rfloor} \binom{d}{2k}\binom{2k}{k} x^k$, while B_m mod $Q_m = (1 - 4x)^d (1 - 2y)$.)

In our applications, as in the previous example, the polynomials P and Q have coefficients that are themselves polynomials in another variable x. The rest of this section is devoted to the proof of the following.

---

[1] Both sides of the identity satisfy $(2y - 1)^2(d + 1)u_d - (2d + 3)u_{d+1} + (d + 2)u_{d+2} = 0, u_0 = u_1 = 1$.

Theorem 8. *Let $P(x, y)/Q(x, y) \in \mathbb{K}(x, y)_{d_x+1, d_y+1}$. Let $\hat{Q}$ be a divisor of Q, $\hat{Q}^\star$ be a square-free part of it with respect to y, and denote by m the number of factors in the square-free decompositions of $\hat{Q}$. Let $(d_x^\star, d_y^\star)$ be bounds on the bidegree of $Q^\star$. Then the polynomial computed by Algorithm 1 annihilates the residues of P/Q at the roots of $\hat{Q}$, has degree in z bounded by $\deg_y \hat{Q}^\star$ and degree in x bounded by*

$$2d_x^\star(d_y + 1) + 2(d_y^\star - 1)d_x - 2d_x^\star d_y^\star.$$

It can be computed in $O(m^2 d_x^\star d_y^\star (m^2 + d_y^{\star 2}))$ operations in $\mathbb{K}$.

Note that rewriting the bound in the equivalent form

$$2d_x d_y - 2(d_x - d_x^\star)(d_y - d_y^\star + 1)$$

shows that the degree in x is bounded by $2d_x d_y$, independently of the multiplicities. The complexity is also bounded independently of the multiplicities by $O(d_x^\star d_y^\star d_y^4)$.

3.2. Bounds

By Fact 4, the resultant R_i has degree in z exactly $\deg Q_i$ so that the degree in z of the result is bounded by $\deg_y Q_1 + \cdots + \deg_y Q_m = d_y^\star$.

The degree in x is the sum of the degrees in x of all the R_i's. In order to derive a bound on the degree of R_i using Fact 4, we first consider the degrees in x and y of A_i and B_i. The important point is that these degrees do not depend so much on Q as on its square-free part. In order to quantify this precisely, we first focus on power series expansion of a special type about which we state a few useful lemmas.

For a polynomial $Q \in \mathbb{K}[x]$ and a real number α, we denote by $\mathcal{E}_\alpha(Q)$ the subset of $\mathbb{K}(x)[[t]]$ formed of power series that can be written

$$c_0 + c_1 \frac{t}{Q} + \cdots + c_n \frac{t^n}{Q^n} + \cdots,$$

with $c_n \in \mathbb{K}[x]$ and $\deg c_n \leqslant n\alpha$, for all n (recall that $\deg 0 = -\infty$, which makes it convenient to allow negative α). This notation extends to the case when x is a tuple of variables, with α replaced by a tuple of real numbers. The main properties of $\mathcal{E}_\alpha(Q)$ are summarized as follows.

Lemma 9. *Let $Q, R \in \mathbb{K}[x]$, $\alpha, \beta \in \mathbb{R}$ and $f \in \mathbb{K}[[t]]$.*

(1) *The set $\mathcal{E}_\alpha(Q)$ is a subring of $\mathbb{K}(x)[[t]]$.*
(2) *Let $S \in \mathcal{E}_\alpha(Q)$ with $S(0) = 0$, then $f(S) \in \mathcal{E}_\alpha(Q)$.*
(3) *The products obey*

$$\mathcal{E}_\alpha(Q) \cdot \mathcal{E}_\beta(R) \subset \mathcal{E}_{\max(\alpha + \deg R, \, \beta + \deg Q)}(Q R).$$

Proof. For *(3)*, if $A = \sum_n a_n t^n / Q^n$ and $B = \sum_n b_n t^n / R^n$ belong respectively to $\mathcal{E}_\alpha(Q)$ and $\mathcal{E}_\beta(R)$, then the nth coefficient of their product is a sum of terms of the form $a_i(x) Q^{n-i} b_{n-i}(x) R^i / (Q R)^n$. Therefore, the degree of the numerator is bounded by $i(\alpha + \deg R) + (n - i)(\beta + \deg Q)$, whence *(3)* is proved. Property *(1)* is proved similarly, the nth coefficient of the product being a sum of terms $a_i(x) b_{n-i}(x) t^n / Q^n$. In Property *(2)*, the condition on $S(0)$ makes $f(S)$ well-defined. The result then follows from *(1)*. $\quad\square$

Corollary 10. *Let $Q \in \mathbb{K}[x, t]$ be such that $Q(0, 0) \neq 0$. Let $Q^\star$ be a square-free part of Q and $\delta(Q^\star)$ its total degree in (x, t). Then*

$$\frac{1}{Q(x, t)} \in \frac{1}{Q(x, 0)} \mathcal{E}_{\min(\deg_x(Q^\star), \delta(Q^\star) - 1)}(Q^\star(x, 0)).$$

Proof. For all i, the coefficient of t^i in Q has degree at most $\min(\deg_x(Q), \delta(Q) - i)$. Thus $R :=$ $(Q(x,t) - Q(x,0))/Q(x,0) \in \mathcal{E}_{\min(\deg_x(Q), \delta(Q)-1)}(Q(x,0))$. Writing $Q(x,t) = Q(x,0)(1+R)$ and using Part *(2)* of Lemma 9 with $f = 1/(1+y)$ then gives the result when Q is square-free. Using $f = 1/(1+y)^i$ gives the result for a pure power by Part *(1)* of the lemma. The general case then follows from Part *(3)* by induction on the number of parts in the square-free decomposition of Q, using additivity of degree and total degree. □

Now, we turn to the fraction $F_i := P(x, y+t)/U_i(x, y+t)/V_i(x, y, t)^i$, with $U_i(x, y) = Q(x,y)/Q_i(x,y)^i$ and $V_i(x, y, t) = (Q_i(x, y+t) - Q_i(x,y))/t$. We use bidegrees with respect to (x, y) and observe that

$$\text{bideg } U_i^\star = \text{bideg } Q^\star - \text{bideg } Q_i, \quad \text{bideg } V_i \leqslant \text{bideg } Q_i - (0, 1).$$

The total degrees in (y, t) behave similarly: that of $U_i^\star(x, y+t)$ is $\deg_y Q - \deg_y Q_i$, while that of $V_i(x, y, t)$ is $\deg_y Q_i - 1$. Corollary 10 gives

$$\frac{1}{U_i(x, y+t)} \in \frac{1}{U_i(x, y)} \mathcal{E}_{\text{bideg } Q^\star - \text{bideg } Q_i - (0,1)}(U_i^\star(x, y)), \tag{2}$$

$$\frac{1}{V_i(x, y, t)^i} \in \frac{1}{V_i(x, y, 0)^i} \mathcal{E}_{\text{bideg } Q_i - (0,2)}(V_i(x, y, 0)). \tag{3}$$

From there, Part *(3)* of Lemma 9 shows that the product of these series belongs to

$$\frac{1}{U_i(x, y) V_i(x, y, 0)^i} \mathcal{E}_{\text{bideg } Q^\star - (0,2)}(U_i^\star(x, y) V_i(x, y, 0)).$$

Thus the coefficient S_{i-1} of t^{i-1} in the power series expansion of F_i can be written as A_i/B_i with

$$B_i = U_i(x, y) V_i(x, y, 0)^i U_i^\star(x, y)^{i-1} V_i(x, y, 0)^{i-1},$$

and finally

$$\begin{aligned}
\text{bideg } A_i &\leqslant \text{bideg } P + (i-1)\text{bideg } Q^\star - 2(i-1)(0,1), \\
\text{bideg } B_i &\leqslant \text{bideg } Q + (i-1)\text{bideg } Q^\star - (2i-1)(0,1),
\end{aligned} \tag{4}$$

whence

$$\text{bideg}(A_i - zB_i) \leqslant \max(\text{bideg } P, \text{bideg } Q - (0,1)) + (i-1)(\text{bideg } Q^\star - (0,2)).$$

Fact 4 can now be exploited, leading to a bound on the degree of the resultant:

$$\begin{aligned}
\deg_x R_i &\leqslant \deg_y Q_i \left(\max(\deg_x P, \deg_x Q) + (i-1)\deg_x Q^\star\right) \\
&\quad + \deg_x Q_i \left(\max(\deg_y P, \deg_y Q - 1) + (i-1)(\deg_y Q^\star - 2)\right).
\end{aligned}$$

Next, we sum over the indices i corresponding to factors of $\hat{Q}$. This leads to the following bound for the degree in x of the result

$$\begin{aligned}
\deg_y \hat{Q}^\star &\max(\deg_x P, \deg_x Q) + (\deg_y \hat{Q} - \deg_y \hat{Q}^\star)\deg_x Q^\star \\
&+ \deg_x \hat{Q}^\star \max(\deg_y P, \deg_y Q - 1) + (\deg_x \hat{Q} - \deg_x \hat{Q}^\star)(\deg_y Q^\star - 2).
\end{aligned}$$

This bound being an increasing function of each of the degrees that appear, it is itself upper bounded by replacing any of those degrees by an upper bound.

In the context of Theorem 8, the bidegrees of P, Q and $\hat{Q}$ are bounded by (d_x, d_y), while those of $Q^\star$ and $\hat{Q}^\star$ are bounded by $(d_x^\star, d_y^\star)$. This leads to the bound

$$d_y^\star d_x + (d_y - d_y^\star)d_x^\star + d_x^\star d_y + (d_x - d_x^\star)(d_y^\star - 2),$$

which rewrites as the bound in the Theorem and completes that part of the proof.

3.3. Complexity

By Fact 6, a square-free decomposition of $\hat{Q}$ can be computed using $\tilde{O}(d_x^2 d_y)$ ops. We now focus on the computations performed inside the ith iteration of the loop and write $(d_x^{(i)}, d_y^{(i)})$ for the bidegree of Q_i. Computing U_i requires an exact division of polynomials of bidegrees at most (d_x, d_y); this division can be performed by evaluation-interpolation in $\tilde{O}(d_x d_y)$ ops. Similarly, the trivariate polynomial V_i can be computed by evaluation-interpolation with respect to (x, y) in time $\tilde{O}(d_x^{(i)} (d_y^{(i)})^2)$. By Eq. (4), both $A_i(x, y)$ and $B_i(x, y)$ have bidegrees at most $(D_x^{(i)}, D_y^{(i)})$, where $D_x^{(i)} = d_x + i d_x^\star$ and $D_y^{(i)} = d_y + i d_y^\star$. They can be computed by evaluation-interpolation in $\tilde{O}(i D_x^{(i)} D_y^{(i)})$ ops. Finally, the resultant $R_i(x, z)$ has bidegree at most $(d_x^{(i)} D_y^{(i)} + d_y^{(i)} D_x^{(i)}, d_y^{(i)})$, and since the degree in y of $A_i - z B_i$ and Q_i is at most $D_y^{(i)}$, it can be computed by evaluation-interpolation in $\tilde{O}((d_x^{(i)} D_y^{(i)} + d_y^{(i)} D_x^{(i)}) d_y^{(i)} D_y^{(i)})$ ops. by Lemma 5. The total cost of the loop is thus $\tilde{O}(L)$, where

$$L = \sum_{i=1}^{m} \left((i + (d_y^{(i)})^2) D_x^{(i)} D_y^{(i)} + d_x^{(i)} d_y^{(i)} (D_y^{(i)})^2 \right).$$

Using the (crude) bounds $D_x^{(i)} \leqslant D_x^{(m)}$, $D_y^{(i)} \leqslant D_y^{(m)}$, $\sum_{i=1}^{m} (d_y^{(i)})^2 \leqslant d_y^{\star 2}$ and $\sum_{i=1}^{m} d_x^{(i)} d_y^{(i)} \leqslant d_x^\star d_y^\star$ shows that L is bounded by

$$D_x^{(m)} D_y^{(m)} \sum_{i=1}^{m} (i + (d_y^{(i)})^2) + (D_y^{(m)})^2 \sum_{i=1}^{m} d_x^{(i)} d_y^{(i)} \leqslant D_x^{(m)} D_y^{(m)} (m^2 + d_y^{\star 2}) + (D_y^{(m)})^2 d_x^\star d_y^\star,$$

which, by using the inequalities $D_x^{(m)} \leqslant 2 m d_x^\star$ and $D_y^{(m)} \leqslant 2 m d_y^\star$, is seen to belong to $O(m^2 d_x^\star d_y^\star (m^2 + d_y^{\star 2}))$, as was to be proved. This completes the proof of the theorem. $\quad \square$

Remark. Note that one could also use Hermite reduction combined with the usual Rothstein–Trager resultant in order to compute a polynomial $\tilde{R}(x, z)$ that annihilates the residues. Indeed, Hermite reduction computes an auxiliary rational function that admits the same residues as the input, while only having simple poles. A close inspection of this approach provides the same bound $d_y^\star$ for the degree in y of $\tilde{R}(x, z)$, but a less tight bound for its degree in x, namely worse by a factor of $d_y^\star$. The complexity of this alternative approach appears to be $\tilde{O}(d_x d_y (d_y + d_y^{\star 3}))$ (using results from Bostan et al., 2010), to be compared with the complexity bound from Theorem 8.

4. Sums of roots of a polynomial

4.1. Algorithm

Given a polynomial $P \in \mathbb{K}[y]$ of degree d with coefficients in a field $\mathbb{K}$ of characteristic 0, let $\alpha_1, \ldots, \alpha_d$ be its (not necessarily distinct) roots in the algebraic closure of $\mathbb{K}$. For any positive integer $c \leqslant d$, the polynomial of degree $\binom{d}{c}$ defined by

$$\Sigma_c P = \prod_{i_1 < \cdots < i_c} \left(y - (\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_c}) \right) \tag{5}$$

has coefficients in $\mathbb{K}$. This section discusses the computation of $\Sigma_c P$ summarized in Algorithm 2, which can be seen as an additive analogue of the *Platypus algorithm* of Banderier and Flajolet (2002).

We recall two classical formulas for the generating function of the Newton sums (see, e.g., Bostan et al., 2006, §2), the second one being valid for monic P only:

$$\mathcal{N}(P) = \frac{\mathrm{rec}(P')}{\mathrm{rec}(P)}, \qquad \mathrm{rec}(P) = \exp\left(\int \frac{d - \mathcal{N}(P)}{y} \, dy \right). \tag{6}$$

Algorithm 2 Polynomial canceling the sums of c roots.

Algorithm **PureComposedSum**(P, c)

Input	A polynomial P of degree d in $\mathbb{K}[y]$, a positive integer $c \leqslant d$
Output	The polynomial $\Sigma_c P$ from Eq. (5)

$D \leftarrow \binom{d}{c}$

$\mathcal{N}(P) \leftarrow \text{rec}(P')/\text{rec}(P) \bmod y^{D+1}$

$S \leftarrow \mathcal{N}(P) \odot \exp(y) \bmod y^{D+1}$

$F \leftarrow \exp\left(\sum_{n=1}^{c} (-1)^{n-1} \frac{S(ny)}{n} z^n\right) \bmod (y^{D+1}, z^{c+1})$

$\mathcal{N}(\Sigma_c P) \leftarrow ([z^c]F) \odot \sum n! y^n \bmod y^{D+1}$

return $\text{rec}\left(\exp\left(\int \frac{D - \mathcal{N}(\Sigma_c P)}{y} \, dy\right) \bmod y^{D+1}\right)$

Truncating these formulas at order $d + 1$ makes $\mathcal{N}(P)$ a representation of the polynomial P (up to normalization), since both conversions above can be performed quasi-optimally by Newton iteration (Schönhage, 1982; Pan, 2000; Bostan et al., 2006). The key for Algorithm 2 is the following variant of Banderier and Flajolet (2002, § 2.3).

Proposition 11. *Let $P \in \mathbb{K}[y]$ be a polynomial of degree d, let $\mathcal{N}(P)$ denote the generating series of its Newton sums and let S be the series $\mathcal{N}(P) \odot \exp(y)$. Let Ψ_c be the polynomial in $\mathbb{K}[t_1, \ldots, t_c]$ defined by*

$$\Psi_c(t_1, \ldots, t_c) = [z^c] \exp\left(\sum_{n \geqslant 1} (-1)^{n-1} t_n \frac{z^n}{n}\right).$$

Then the following equality holds

$$\mathcal{N}(\Sigma_c P) \odot \exp(y) = \Psi_c(S(y), S(2y), \ldots, S(cy)).$$

Proof. By construction, the series S is

$$S(y) = \sum_{n \geqslant 0} (\alpha_1^n + \alpha_2^n + \cdots + \alpha_d^n) \frac{y^n}{n!} = \sum_{i=1}^{d} \exp(\alpha_i y).$$

When applied to the polynomial $\Sigma_c P$, this becomes

$$\mathcal{N}(\Sigma_c P) \odot \exp(y) = \sum_{i_1 < \cdots < i_c} \exp\left((\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_c})y\right)$$

$$= [z^c] \prod_{i=1}^{d} (1 + z \exp(\alpha_i y)).$$

This expression rewrites:

$$[z^c] \exp\left(\sum_{i=1}^{d} \log(1 + z \exp(\alpha_i y))\right) = [z^c] \exp\left(\sum_{i=1}^{d} \sum_{m \geqslant 1} (-1)^{m-1} \exp(\alpha_i my) \frac{z^m}{m}\right)$$

$$= [z^c] \exp\left(\sum_{m \geqslant 1} (-1)^{m-1} S(my) \frac{z^m}{m}\right),$$

and the last expression equals $\Psi_c(S(y), S(2y), \ldots, S(cy))$. $\square$

The correctness of Algorithm 2 follows from observing that the truncation orders $D + 1$ in y and $c + 1$ in z of the power series involved in the algorithm are sufficient to enable the reconstruction of $\Sigma_c P$ from its first Newton sums by (6).

We will be interested in the case where P is a polynomial in $\mathbb{K}[x, y]$. Then, the coefficients of $\Sigma_c P$ with respect to y may have denominators. We analyze the structure of the coefficients of $\Sigma_c P$ as elementary symmetric functions of the roots of P in order to compute bounds on the bidegree of the polynomial obtained by clearing out these denominators. The rest of this section proves the following result.

Theorem 12. *Let $P \in \mathbb{K}[x, y]_{d_x+1,d_y+1}$, and let $c \leqslant d_y$ be a positive integer. Let $a \in \mathbb{K}[x]$ denote the leading coefficient of P with respect to y and let $\Sigma_c P$ be defined as in Eq.* (5). *We also denote*

$$D_x := \binom{d_y - 1}{c - 1}, \qquad D_y := \binom{d_y}{c}.$$

Then $a^{D_x} \cdot \Sigma_c P$ is a polynomial in $\mathbb{K}[x, y]$ that cancels all sums $\alpha_{i_1} + \cdots + \alpha_{i_c}$ of c roots $\alpha_i(x)$ of P, with $i_1 < \cdots < i_c$, and satisfies

$$\deg_x(a^{D_x} \cdot \Sigma_c P) \leqslant d_x D_x, \quad \deg_y(a^{D_x} \cdot \Sigma_c P) = D_y.$$

Moreover, this polynomial can be computed in $\tilde{O}(c d_x D_x D_y)$ ops.

These bounds are sharp. Experiments suggest that for generic P of bidegree (d_x, d_y) the minimal polynomial of $\alpha_{i_1} + \cdots + \alpha_{i_c}$ has bidegree precisely $(d_x D_x, D_y)$. Similarly, the complexity result is quasi-optimal up to a factor of c only.

4.2. Bounds

We start with the following effective version of a very classical result on symmetric functions (Yap, 2000, Theorem 6.21).

Lemma 13. *Let $\alpha_1, \ldots, \alpha_n$ be indeterminates, and $\sigma_1, \ldots, \sigma_n$ be the associated elementary symmetric functions. Let $P \in \mathbb{K}[\alpha_1, \ldots, \alpha_n]$ be a symmetric polynomial satisfying*

$$\deg_{\alpha_i} P \leqslant d \text{ for all } 1 \leqslant i \leqslant n$$

Then P can be expressed as a polynomial in $\sigma_1, \ldots, \sigma_n$ of total degree at most d.

Proof. This is a consequence of the form of the matrix of the change of bases from the elementary symmetric functions to the monomial symmetric functions as described for instance in the proof of Stanley (1999, Theorem 7.4.4). Since P is symmetric and has degree at most d with respect to each variable, it can be written as a linear combination of monomial symmetric functions of the form $\sum_{i_1 < \cdots < i_k} \alpha_{i_1}^{\lambda_1} \cdots \alpha_{i_k}^{\lambda_k}$, where $\lambda_i \leqslant d$ for all i. These monomial symmetric functions can in turn be written as linear combinations of elementary symmetric functions of the form $\sigma_{\mu_1} \cdots \sigma_{\mu_l}$ where $l \leqslant d$, which is exactly the result of the lemma. $\square$

For the proof of the bounds in Theorem 12, we write

$$P = a(x) y^{d_y} + \sum_{i=0}^{d_y - 1} a_i(x) y^i = a(x) \prod_{i=1}^{d_y} (y - \alpha_i(x)).$$

Let $\sigma_1(x), \ldots, \sigma_{d_y}(x)$ denote the elementary symmetric functions of the α_i's. Then, the elementary symmetric functions of the roots $\alpha_{i_1} + \cdots + \alpha_{i_c}$ of $\Sigma_c P$ have degree $\binom{d_y - 1}{c - 1}$ in each α_i. Therefore, by Lemma 13, the coefficients of $\Sigma_c P$ are polynomials of total degree at most $\binom{d_y - 1}{c - 1}$ in $\sigma_1, \ldots, \sigma_{d_y}$. From there, the bound on $\deg_x(a^{D_x} \cdot \Sigma_c P)$ is immediately derived from the classical relations $(-1)^i \sigma_i = a_{d_y - i}/a$.

4.3. Complexity

The computation is performed by evaluation and interpolation at $1 + d_x D_x$ values of x. By Fact 1, at each of these values, the computation of the truncated series expansions $\mathcal{N}(P)$ and S in $\mathbb{K}[[y]]_{1+D_y}$ have complexity $\tilde{O}(D_y)$; so do the computations of $\mathcal{N}(\Sigma_c P)$ and the last step; the most expensive step is the computation of F, which costs $\tilde{O}(cD_y)$ ops. in $\mathbb{K}$. Since this is executed $O(d_x D_x)$ times, the total cost is $\tilde{O}(cd_x D_x D_y)$.

5. Diagonals

In this section we turn to our main topic, namely the computation of annihilating polynomials for diagonals of bivariate rational functions. The algorithm relies on a classical expression of the diagonal as a sum of residues (see Lemma 14), and on the results of Sections 3 and 4. The conclusions of the analysis of Algorithm 3 can be found in Theorem 18 and Proposition 20.

5.1. Algebraic equations for diagonals

Let $F(x, y) = \sum_{i,j \geqslant 0} a_{i,j} x^i y^j$ be a rational function in $\mathbb{K}(x, y)$, whose denominator does not vanish at $(0, 0)$. Then the diagonal of F is defined as $\operatorname{Diag} F(t) = \sum_{i \geqslant 0} a_{i,i} t^i$. A first basic, but very important, remark is that

$$\operatorname{Diag} F(t) = [y^{-1}] \frac{1}{y} F\left(\frac{t}{y}, y\right).$$

When $\mathbb{K} = \mathbb{C}$, this coefficient can be viewed as a Cauchy integral and computed by the residue formula (Furstenberg, 1967). For general $\mathbb{K}$ (of characteristic 0), we proceed similarly with a purely algebraic approach, adapted from Gessel (1980, Theorem 6.1). (The reader who is not interested in the general proof may also skip directly to Lemma 14.) The starting point is the partial fraction decomposition of $G(t, y) := \frac{1}{y} F(\frac{t}{y}, y)$ considered as a rational function in $\mathbb{K}(t)(y)$:

$$G(t, y) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} f_{i,j}(t, y), \tag{7}$$

where

$$f_{i,j}(t, y) = \frac{r_{i,j}(t)}{(y - y_i(t))^j}, \qquad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant m_i.$$

In particular, $r_{i,1}(t)$ is the residue of G at $y_i(t)$ for all $i \in \{1, 2, \ldots, n\}$. By Puiseux's theorem, there exists $N \in \mathbb{N}^\star$ such that the y_i's and $r_{i,j}$'s all lie in the field $\overline{\mathbb{K}}((t^{1/N}))$. In order to apply the operator $[y^{-1}]$ on both sides of Equation (7), it is necessary to find a ring where both the equality and the operator $[y^{-1}]$ make sense. We are going to check that $\mathbb{A} = \mathbb{K}((y))((t^{1/N}))$ and $[y^{-1}]$ computed coefficient-wise are suitable for this.

First, as a rational function, it is immediate that $G(t, y)$ belongs to $\mathbb{A}$. In order to expand the right-hand side, we consider each term separately and distinguish between the cases $\operatorname{val}_t(y_i) \leqslant 0$ and $\operatorname{val}_t(y_i) > 0$. If $\operatorname{val}_t(y_i) \leqslant 0$, $f_{i,j}$ can be written as follows:

$$f_{i,j} = \frac{r_{i,j}}{(-y_i)^j} \cdot \frac{1}{(1 - y/y_i)^j} = \frac{r_{i,j}}{(-y_i)^j} \sum_{k \geqslant 0} \binom{-j}{k} \frac{y^k}{y_i^k} \in \overline{\mathbb{K}}((t^{1/N}))[[y]].$$

Since $\operatorname{val}_t(1/y_i) \geqslant 0$, the series $f_{i,j}/r_{i,j}$ actually belongs to $\overline{\mathbb{K}}[[t^{1/N}]][[y]] \cong \overline{\mathbb{K}}[[y]][[t^{1/N}]]$. Hence $f_{i,j} \in \overline{\mathbb{K}}[[y]]((t^{1/N})) \subset \mathbb{A}$, and in particular $[y^{-1}] f_{i,j} = 0$. On the other hand, if $\operatorname{val}_t(y_i) > 0$ then $f_{i,j}$ can be expanded directly in $\mathbb{A}$ as:

Algorithm 3 Polynomial canceling the diagonal of a rational function. The notation ddeg is defined in Eq. (9); numer denotes the numerator of the irreducible form of a fraction.

Algorithm **AlgebraicDiagonal**(A/B)

Input   Two polynomials A and B in $\mathbb{K}(x, y)$, with $B(0, 0) \neq 0$
Output   A polynomial $\Phi \in \mathbb{K}[t, \Delta]$ such that $\Phi(t, \mathrm{Diag}\, A/B) = 0$

$P, Q, \alpha \leftarrow y^{\mathrm{ddeg}^-(A)} A(\frac{t}{y}, y),\ y^{\mathrm{ddeg}^-(B)} B(\frac{t}{y}, y),\ \mathrm{ddeg}^-(B) - \mathrm{ddeg}^-(A) - 1$
if $\alpha < 0$ **then**
    $r \leftarrow$ **AlgebraicResidues**$(y^\alpha P/Q, y)$
$R \leftarrow$ **AlgebraicResidues**$(y^\alpha P/Q, Q)$
$c \leftarrow$ number of small branches of Q
$\Phi(t, \Delta) \leftarrow$ numer(**PureComposedSum**(R, c))
if $\alpha < 0$ **then**
    $\Phi(t, \Delta) \leftarrow$ numer$(\Phi(t, \Delta - r))$
return $\Phi(t, \Delta)$

$$f_{i,j} = \frac{r_{i,j}}{y^j} \cdot \frac{1}{(1 - y_i/y)^j} = \frac{r_{i,j}}{y^j} \sum_{k \geq 0} \binom{-j}{k} \frac{y_i^k}{y^k}.$$

Since $y_i/y \in \mathbb{A}$ and $\mathrm{val}_t(y_i/y) > 0$, this last quantity is the sum of a convergent series (in the sense of formal Laurent series) of elements of $\mathbb{A}$, hence belongs to $\mathbb{A}$. In this case we obtain $[y^{-1}] f_{i,j} = r_{i,1}$.

We have everything we need to apply $[y^{-1}]$ on both sides of Equation (7), leading to the generalization to any base field of characteristic 0 of Furstenberg's classical result (Furstenberg, 1967, §2).

Lemma 14. *If $F(x, y)$ is a rational function in $\mathbb{K}(x, y)$ whose denominator does not vanish at $(0, 0)$, then*

$$\mathrm{Diag}\, F(t) = \sum_{\substack{y(t) \in \mathcal{P} \\ \mathrm{val}_t(y(t)) > 0}} \mathrm{Residue}\left(\frac{1}{y} F\left(\frac{t}{y}, y \right), y = y(t) \right), \tag{8}$$

where $\mathcal{P}$ is the set of poles of $\frac{1}{y} F(\frac{t}{y}, y)$.

Since the elements of $\mathcal{P}$ are algebraic and finite in number and residues are obtained by series expansion, which entails only rational operations, it follows that the diagonal is algebraic too.

When $Q \in \mathbb{K}(t)[y]$ is a polynomial, the roots of Q with positive valuation are called the *small branches* of Q. Their number is denoted Nsmall(Q). Combining the algorithms of the previous section gives Algorithm 3 that produces a polynomial equation for Diag F.

Example 15. Let $d \geq 0$ be an integer, and let $F_d(x, y)$ be the rational function $1/(1 - x - y)^{d+1}$. The diagonal of F_d is equal to

$$\sum_{n \geq 0} \binom{2n + d}{n} \binom{n + d}{d} t^n.$$

By the previous argument, it is an algebraic series, which is the sum of the residues of the rational function G_d of Example 7 over the small branches of its denominator (with x replaced by t). In this case, the denominator is $y - t - y^2$. It has one solution tending to 0 with t; the other one tends to 1. Thus the diagonal is canceled by the quadratic polynomial (1).

Example 16. For an integer $d > 0$, we consider the rational function

$$F_d(x, y) = \frac{x^{d-1}}{1 - x^d - y^{d+1}},$$

of bidegree $(d, d + 1)$. The first step of Algorithm 3 produces

$$G_d(t, y) = y^\alpha \frac{P}{Q} = \frac{t^{d-1}}{y^d - t^d - y^{2d+1}},$$

of bidegree $(d, 2d + 1)$, whose denominator is irreducible with d small branches. From there, Algorithm 3 computes a polynomial Φ_d annihilating $\text{Diag}\, F_d$, which is experimentally irreducible and whose bidegrees for $d = 1, 2, 3, 4$ are $(2, 3), (18, 10), (120, 35), (700, 126)$. From these values, it is easy to conjecture that the bidegree is given by

$$\left(d(d+1) \binom{2d-1}{d-1}, \binom{2d+1}{d} \right),$$

of exponential growth in the bidegree of F_d. In general, these bidegrees do not grow faster than in this example. In Theorem 18 below, we prove bounds that are barely larger than the values above.

Sloped diagonals. If p and q are relatively prime positive integers and $F(x, y) = \sum_{i,j \geqslant 0} f_{i,j} x^i y^j$, then the *sloped diagonal* of F, $\text{Diag}_{p,q} F(t)$ is $\sum_{n \geqslant 0} f_{pn,qn} t^n$. Direct manipulations show that

$$\text{Diag}_{p,q} F(t^{pq}) = \text{Diag}(F(x^q, y^p))(t),$$

so that our bounds and algorithm apply almost directly to these more general diagonals.

5.2. Degree bounds and complexity

The rest of this section is devoted to the derivation of bounds on the complexity of Algorithm 3 and on the size of the polynomial it computes, which are given in Theorem 18.

Degrees. A bound on the bidegree of Φ will be obtained from the bounds successively given by Theorems 8 and 12.

In order to follow the impact of the change of variables in the first step, we define the *lower diagonal degree* and *upper diagonal degree* of a polynomial $P(x, y) = \sum_{i,j} a_{i,j} x^i y^j$ respectively as the integers

$$\begin{aligned}
\text{ddeg}^-(P) &= \sup \left\{ i - j \mid a_{i,j} \neq 0 \right\} \\
\text{ddeg}^+(P) &= \sup \left\{ j - i \mid a_{i,j} \neq 0 \right\}
\end{aligned} \tag{9}$$

We collect the properties of interest in the following.

Lemma 17. *For any P and Q in $\mathbb{K}[x, y]$,*

(1) $\text{ddeg}^-(P) \leqslant \deg_x P$ *and* $\text{ddeg}^+(P) \leqslant \deg_y P$;
(2) $\text{ddeg}^\pm(PQ) = \text{ddeg}^\pm(P) + \text{ddeg}^\pm(Q)$;
(3) *there exists a polynomial $\tilde{P} \in \mathbb{K}[x, y]$, such that $P(x/y, y) = y^{-\text{ddeg}^-(P)} \tilde{P}(x, y)$, with $\tilde{P}(x, 0) \neq 0$ and*

$$\text{bideg}(\tilde{P}) = (\deg_x P, \text{ddeg}^-(P) + \text{ddeg}^+(P));$$

(4) $\text{bideg}((\tilde{P})^\star) = (\deg_x P^\star, \text{ddeg}^-(P^\star) + \text{ddeg}^+(P^\star))$.

Proof. Part (1) is immediate. The quantities $\text{ddeg}^-(P)$ and $\text{ddeg}^+(P)$ are nothing else than $-\text{val}_y P(x/y, y)$ and $\deg_y P(x/y, y)$, which makes Parts (2) and (3) clear too. From there, we get the identity $\widetilde{PQ} = \tilde{P}\tilde{Q}$ for arbitrary P and Q, whence $(\tilde{P})^\star = \widetilde{P^\star}$ and Part (4) is a consequence of Parts (1) and (3). $\square$

Thus, starting with a rational function $F = A/B \in \mathbb{K}(x, y)$, with (d_x, d_y) a bound on the bidegrees of A and B, and $(d_x^\star, d_y^\star)$ a bound on the bidegree of a square-free part $B^\star$ of B, the first step of the algorithm constructs $G(t, y) = y^\alpha \frac{P}{Q}$, with polynomials P and Q and

$$\alpha = \mathrm{ddeg}^-(B) - \mathrm{ddeg}^-(A) - 1 \tag{10}$$

$$\mathrm{bideg}\, P \leqslant (d_x, \mathrm{ddeg}^-(A) + \mathrm{ddeg}^+(A)), \quad \mathrm{bideg}\, Q \leqslant (d_x, \mathrm{ddeg}^-(B) + \mathrm{ddeg}^+(B)),$$

$$\mathrm{bideg}\, Q^\star = (d_x^\star, \mathrm{ddeg}^-(B^\star) + \mathrm{ddeg}^+(B^\star)).$$

We first explain how to compute the number c of small branches of Q.

Small branches. It is classical that for a polynomial $P = \sum a_{i,j} x^i y^j \in \mathbb{K}[x, y]$, the number of its solutions tending to 0 can be read off its Newton polygon (see, e.g. Walker, 1978). This polygon is the lower convex hull of the union of $(i, j) + \mathbb{N}^2$ for (i, j) such that $a_{i,j} \neq 0$. The number of solutions tending to 0 is given by the minimal y-coordinate of its leftmost points. Since the number of small branches counts only distinct solutions, it is thus given by

$$\mathrm{Nsmall}(P) = \mathrm{Nsmall}(P^\star) = \mathrm{val}_y([x^{\mathrm{val}_x\, P^\star}] P^\star). \tag{11}$$

The change of variables $x \mapsto x/y$ changes the coordinates of the point corresponding to $a_{i,j}$ into $(i, j - i)$. This transformation maps the vertices of the original Newton polygon to the vertices of the Newton polygon of the Laurent polynomial $P(x/y, y)$. Multiplying by $y^{\mathrm{ddeg}^-(P)}$ yields a polynomial and shifts the Newton polygon up by $\mathrm{ddeg}^-(P)$, thus

$$\mathrm{Nsmall}\left(y^{\mathrm{ddeg}^-(P)} P(x/y, y)\right) = \mathrm{Nsmall}(P^\star) + \mathrm{ddeg}^-(P^\star).$$

The number of small branches of the polynomial Q constructed above is then given by

$$c := \mathrm{Nsmall}(B^\star) + \mathrm{ddeg}^-(B^\star). \tag{12}$$

Degree in $\triangle$. At this point, there is a slight difference between the cases $\alpha \geqslant 0$ and $\alpha < 0$. Indeed, in the latter case we have to take the additional small branch at 0 into account. To do this, we denote by r the residue of G at 0. Since r is rational, we may compute a polynomial R that vanishes only on the residues at non-zero small branches of the denominator of G. If $\tilde{\Phi}(t, \triangle)$ is the polynomial produced by applying Algorithm 2 to (R, c), then the polynomial $\Phi(t, \triangle) = \tilde{\Phi}(t, \triangle - r)$ cancels $\mathrm{Diag}\, F$. Thus we apply Algorithm 1 to $((y^\alpha P)/Q, Q)$ if $\alpha \geqslant 0$, and to $(P/(y^{-\alpha} Q), Q)$ otherwise. By Theorem 8, in both cases we obtain a polynomial R of degree D_y, with

$$D_y := \mathrm{ddeg}^-(B^\star) + \mathrm{ddeg}^+(B^\star), \tag{13}$$

and applying Algorithm 2 gives a polynomial Φ with $\deg_\triangle \Phi = \binom{D_y}{c}$.

Degree in t. To bound the degree of Φ in t, we can neglect our optimization and apply Algorithm 1 to $(y^\alpha P, Q, Q)$ or $(P, y^{-\alpha} Q, y^{-\alpha} Q)$ depending on whether $\alpha \geqslant 0$ or $\alpha < 0$. Indeed, the polynomial Φ obtained this way is clearly a multiple of the one computed by the algorithm. By Theorem 8, since the bidegrees of P, $y^\alpha P$, Q and $y^{-\alpha} Q$ are all bounded by $(d_x, d_x + d_y + 1)$, we compute a polynomial R of degree bounded by D_x, where

$$D_x := 2d_x(d_x + d_y + 1) + d_x - 2(d_x - d_x^\star)(d_x - d_x^\star + d_y - d_y^\star + 1). \tag{14}$$

Applying Theorem 12 to (R, c) or $(R, c + 1)$ depending on the sign of α yields in both cases $\deg_t \Phi \leqslant D_x \binom{D_y}{c}$.

Complexity. We now analyze the cost of Algorithm 3. The computation of P and Q does not require any arithmetic operation. Next, the computation of R and r takes $\tilde{O}((d_x + d_y)^6)$ ops. (see the comment after Theorem 8). The number of small branches is obtained with no arithmetic operation from a square-free decomposition computed in Algorithm 1. The bounds of the discussion above and Theorem 12 show that Algorithm 2 uses $\tilde{O}(c D_x \binom{D_y}{c}^2)$ ops. Finally, if a translation of the variable is needed, it can be performed by evaluation-interpolation in $\tilde{O}(D_x \binom{D_y}{c}^2)$ ops. (One may as well evaluate and interpolate with respect to x and apply better algorithms for univariate translation (Bostan et al., 2006, §5).)

We summarize all the results of this section in the following theorem.

Theorem 18. *Let $F = A/B$ be a rational function in $\mathbb{K}(x, y)$ with $B(0,0) \neq 0$. Let (d_x, d_y) (resp. $(d_x^\star, d_y^\star)$) be a bound on the bidegrees of A and B (resp. a square-free part of B). Let D_x, D_y, c be defined as in Eqs. (14), (13), (12). Then there exists a polynomial $\Phi \in \mathbb{K}[t, \Delta]$ such that $\Phi(t, \mathrm{Diag}\, F(t)) = 0$ and*

$$\deg_\Delta \Phi = \binom{D_y}{c}, \qquad \deg_t \Phi \leqslant D_x \binom{D_y}{c}.$$

Algorithm 3 computes it in $\tilde{O}\left(cD_x\binom{D_y}{c}^2 + (d_x + d_y)^6 \right)$ ops.

A general bound on $\mathrm{bideg}\, \Phi$ depending only on a bound (d, d) on the bidegree the input can be deduced from the above as

$$\mathrm{bideg}\, \Phi \leqslant (d(4d+3), 1) \times \binom{2d}{d}.$$

5.3. Optimization

Assume that the denominator of $F(x/y)/y$ is already partially factored as $Q(y) = \tilde{Q}(y) \prod_{i=1}^{k} (y - y_i(x))^{\alpha_i}$, where the y_i's are k distinct *rational* branches among the c small branches of Q. Then their corresponding (rational) residues r_i contribute to the diagonal. The special case where $k = 1$ and $y_1 = 0$ is exactly the situation that occurred in the discussion on $\deg_\Delta \Phi$ before Theorem 18, when $\alpha < 0$. The trick that we used extends directly to the general case: it suffices to apply Algorithm 1 to $\tilde{Q}$, Algorithm 2 with $c - k$ roots, and Φ is then recovered through a change of variable.

5.4. Generic case

The bounds from Theorem 18 on the bidegree of Φ are slightly pessimistic with respect to the variable t, but generically tight with respect to the variable Δ, as will be proved in Proposition 20 below. We first need a lemma.

Lemma 19. *Let $\mathbb{K}$ be a field of characteristic 0, and let $P \in \mathbb{K}[y]$ be a polynomial of degree d, with Galois group $\mathfrak{S}_d$ over $\mathbb{K}$. Assume that the roots $\alpha_1, \ldots, \alpha_d$ of P are algebraically independent over $\mathbb{Q}$. Then, for any $c \leqslant d$, the degree $\binom{d}{c}$ polynomial $\Sigma_c P$ is irreducible in $\mathbb{K}[y]$.*

Proof. Since $\Sigma = \alpha_1 + \cdots + \alpha_c$ is a root of $\Sigma_c P$, it suffices to prove that $\mathbb{K}(\Sigma)$ has degree $\binom{d}{c}$ over $\mathbb{K}$. The α_i's being algebraically independent, any permutation $\sigma \in \mathfrak{S}_d$ of all the α_i's that leaves Σ unchanged has to preserve the sets $\{\alpha_1, \ldots, \alpha_c\}$ and $\{\alpha_{c+1}, \ldots, \alpha_d\}$. Conversely, any such permutation induces an automorphism of $\mathbb{K}(\alpha_1, \ldots, \alpha_d)$ that leaves Σ invariant. In other words, the Galois group of $\mathbb{K}(\alpha_1, \ldots, \alpha_d)$ over $\mathbb{K}(\Sigma)$ is equal to $\mathfrak{S}_c \times \mathfrak{S}_{d-c}$. It follows that $\mathbb{K}(\alpha_1, \ldots, \alpha_d)$ has degree $c!(d-c)!$ over $\mathbb{K}(\Sigma)$ and degree $d!$ over $\mathbb{K}$, so that $\mathbb{K}(\Sigma)$ has degree $\binom{d}{c}$ over $\mathbb{K}$. $\quad\square$

Proposition 20. *Let A be a polynomial in $\mathbb{Q}[x, y]$. Let d_x, d_y be non-negative integers, $s^- \leqslant d_x$, $s^+ \leqslant d_y$, and*

$$B(x, y) = \sum_{i=0}^{s^-} b_i^{(x)} x^i + \sum_{j=1}^{s^+} b_j^{(y)} y^j + \sum_{\substack{i \leqslant d_x, j \leqslant d_y \\ -s^- \leqslant j-i \leqslant s^+}} b_{i,j} x^i y^j \in \mathbb{Q}[(b_i^{(x)}), (b_j^{(y)}), x, y],$$

where the $b_i^{(x)}$ and $b_j^{(y)}$ are indeterminates and $b_{i,j} \in \mathbb{Q}$.

Then the polynomial computed by Algorithm 3 with input A/B is irreducible of degree $\binom{s^- + s^+}{s^-}$ over $\mathbb{K} = \mathbb{Q}((b_i^{(x)}), (b_j^{(y)}))(x)$.

Proof. First apply the change of variables to obtain $G = y^\alpha P/Q$, with

$$Q(x, y) = \sum_{i=0}^{s^-} b_i^{(x)} x^i y^{s^- - i} + \sum_{j=1}^{s^+} b_j^{(y)} y^{s^+ + j} + \sum_{i,j} b_{i,j} x^i y^{s^- - i + j}.$$

Denote $d = s^- + s^+$. Then, the polynomial $Q(1, y)$ has the form $\sum_{j \leqslant d} t_j y^j$ where each of the t_j's is the sum of one of the indeterminates and rational constants. This implies that the t_j's are algebraically independent over $\mathbb{Q}$. Therefore, $Q(1, y)$ has Galois group $\mathfrak{S}_d$ over $\mathbb{Q}(t_0, \ldots, t_d)$ and its roots are algebraically independent over $\mathbb{Q}$ (van der Waerden, 1949, §57). This property lifts to $Q(x, y)$ (van der Waerden, 1949, §61), which thus has Galois group $\mathfrak{S}_d$ and algebraically independent roots, denoted $y_1, \ldots, y_d$.

Now define the polynomial $R(x, y) = \prod_i (y - \tilde{P}(x, y_i)/\partial_y Q(x, y_i))$, where $\tilde{P} = y^\alpha P$ if $\alpha \geqslant 0$ and $\tilde{P} = P$ otherwise. Since Q has simple roots, this is exactly the polynomial that is computed by Algorithm 1. The family $\{P(x, y_i)/\partial_y Q(x, y_i)\}$ is algebraically independent, since any algebraic relation between them would induce one for the y_i's by clearing out denominators. In particular, the natural morphism $\mathrm{Gal}(Q/\mathbb{K}) = \mathfrak{S}_d \to \mathrm{Gal}(R/\mathbb{K})$ is injective, whence an isomorphism. (Here, $\mathrm{Gal}(P/\mathbb{K})$ denotes the Galois group of $P \in \mathbb{K}[y]$ over $\mathbb{K}$.) Since an immediate investigation of the Newton polygon of Q shows that it has s^- small branches, we conclude using Lemma 19 and the fact that the translation of the variable doesn't change the irreducible character of Φ. $\square$

Proposition 20 should be viewed as an optimality result. Indeed, for a generic rational function A/B as in the proposition, we have $B = B^\star$, $\mathrm{ddeg}^-(B) = s^-$, $\mathrm{ddeg}^+(B) = s^+$ and B has s^- small branches. This implies that the bound of Theorem 18 for $\deg_\Delta \Phi$ is optimal in this (generic) case.

If one believes that random examples should behave like the generic case, then the proposition means that the polynomial computed by Algorithm 3 will be irreducible most of the time.

As an example, we consider the special case of Proposition 20 where $s^- = s^+ = d_x = d_y = d$. In this case, $\deg_\Delta \Phi$ is $\binom{2d}{d}$. We compare this to the following experiment on random examples.

Example 21. We consider a rational function $F(x, y) = 1/B(x, y)$, where $B(x, y)$ is a dense polynomial of bidegree (d, d) chosen at random. For $d = 1, 2, 3, 4$, algorithm **AlgebraicDiagonal**(F) produces *irreducible* outputs with bidegrees $(2, 2), (16, 6), (108, 20), (640, 70)$, that are matched by the formulas

$$\left(2d^2 \binom{2d-2}{d-1}, \binom{2d}{d} \right), \tag{15}$$

so that the bound on $\deg_\Delta \Phi$ is tight in this case and the irreducibility of the output shows that Theorem 18 cannot be improved further.

6. Walks

The key ingredient in the fact that diagonals may have a big minimal polynomial was the possibility to write them as a sum of residues. The same exponential growth as in Proposition 20 therefore occurs for other functions bearing this same structure. For instance, constant terms of rational functions in $\mathbb{C}(x)[[y]]$ can also be written as contour integrals of rational functions around the origin and thus by the residue theorem be expressed as a sum of residues.

By contrast, such sums of residues of rational functions always satisfy a differential equation of only polynomial size (Bostan et al., 2010). Thus, when an algebraic function appears to be connected to a sum of residues of a rational function, the use of this differential structure is much more adapted to the computation of series expansions, instead of going through a potentially large polynomial.

As an example where this phenomenon occurs naturally, we consider here the enumeration of unidimensional lattice walks, following Banderier and Flajolet (2002) and Bousquet-Mélou (2008). Our goal in this section is to study, from the algorithmic perspective, the series expansions of various generating functions (for bridges, excursions, meanders) that have been identified as algebraic (Banderier and Flajolet, 2002). One of our contributions is to point out that although algebraic
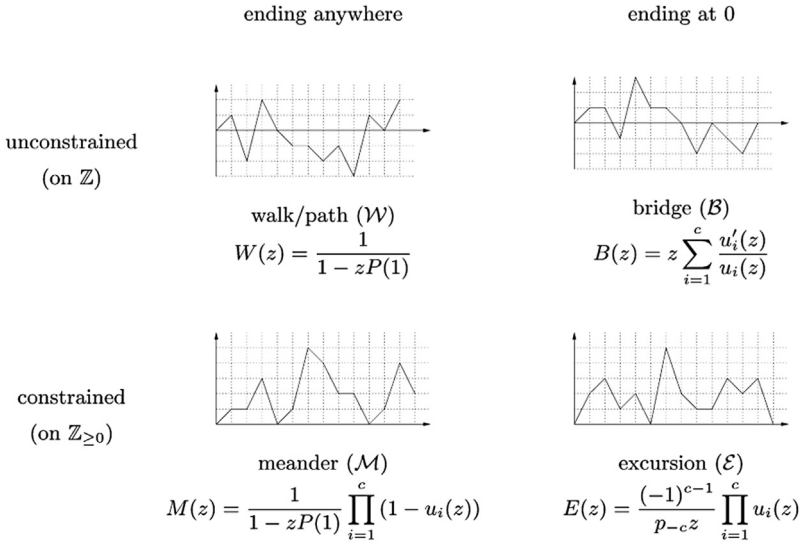
ending anywhere   ending at 0

unconstrained
(on $\mathbb{Z}$)

walk/path $(\mathcal{W})$
$$W(z) = \frac{1}{1 - zP(1)}$$

bridge $(\mathcal{B})$
$$B(z) = z \sum_{i=1}^{c} \frac{u_i'(z)}{u_i(z)}$$

constrained
(on $\mathbb{Z}_{\geq 0}$)

meander $(\mathcal{M})$
$$M(z) = \frac{1}{1 - zP(1)} \prod_{i=1}^{c} (1 - u_i(z))$$

excursion $(\mathcal{E})$
$$E(z) = \frac{(-1)^{c-1}}{p_{-c} z} \prod_{i=1}^{c} u_i(z)$$

Fig. 1. (Banderier and Flajolet, 2002) The four types of paths: walks, bridges, meanders and excursions and the corresponding generating functions.

series can be expanded fast (Chudnovsky and Chudnovsky 1986, 1987; Bostan et al., 2007), the pre-computation of a polynomial equation could have prohibitive cost. We overcome this difficulty by precomputing differential (instead of polynomial) equations that have polynomial size only, and using them to compute series expansions to precision N for bridges, excursions and meanders in time quasi-linear in N.

6.1. Preliminaries

We start with some vocabulary on lattice walks. A *simple step* is a vector $(1, u)$ with $u \in \mathbb{Z}$. A *step set S* is a finite set of simple steps. A *unidimensional walk* in the plane $\mathbb{Z}^2$ built from S is a finite sequence $(A_0, A_1, \ldots, A_n)$ of points in $\mathbb{Z}^2$, such that $A_0 = (0, 0)$ and $\overrightarrow{A_{k-1}A_k} = (1, u_k)$ with $(1, u_k) \in S$. In this case n is called the *length* of the walk, and S is the *step set* of the walk. The y-coordinate of the endpoint A_n, namely $\sum_{i=1}^{n} y_i$, is called the *final altitude* of the walk. The characteristic polynomial of the step set S is

$$\Gamma_S(y) = \sum_{(1,u) \in S} y^u.$$

Following Banderier and Flajolet, we consider three specific families of walks: bridges, excursions and meanders (Banderier and Flajolet, 2002). *Bridges* are walks with final altitude 0, *meanders* are walks confined to the upper half plane, and *excursions* are bridges that are also meanders. Fig. 1, taken from Banderier and Flajolet (2002), summarizes these definitions graphically.

We define the full generating power series of walks

$$W_S(x, y) = \sum_{n \geq 0, k \in \mathbb{Z}} w_{n,k} x^n y^k \in \mathbb{Z}[y, y^{-1}][[x]],$$

where $w_{n,k}$ is the number of walks with step set S, of length n and final altitude k. We denote by $B_S(x)$ (resp. $E_S(x)$, and $M_S(x)$) the power series $\sum_{n \geq 0} u_n x^n$, where u_n is the number of bridges (resp. excursions, and meanders) of length n with step set S.

We omit the step set S as a subscript when there is no ambiguity. Several properties of the power series W, B, E and M are classical:

Fact 22. *(Banderier and Flajolet, 2002, §2.1–2.2) The power series W, B, E and M satisfy*

(1) $W(x, y)$ is rational and $W(x, y) = 1/(1 - x\Gamma(y))$;
(2) $B(x)$, $E(x)$ and $M(x)$ are algebraic;
(3) $B(x) = [y^0]W(x, y)$;
(4) $E(x) = \exp\left(\int (B(x) - 1)/x \, dx\right)$.

In what follows, we describe and analyze three methods to compute the power series expansions of B, E and M. In the next two sections, we first study two previously known methods, then we introduce a new one.

6.2. Expanding the generating power series

From now on, we fix a step set S, and we denote by u^- (resp. u^+) the largest u such that $(1, -u) \in S$ (resp. $(1, u) \in S$). We also define $d = u^- + u^+$. The integer d measures the vertical amplitude of S; this makes d a good scale for measuring the complexity of the algorithms that will follow. We assume that both u^- and u^+ are positive, since otherwise the study of the bridges, excursions and meanders becomes trivial.

The direct method. The combinatorial definition of walks yields a recurrence relation for $w_{n,k}$:

$$w_{n,k} = \sum_{(1,u)\in S} w_{n-1,k-u}, \tag{16}$$

with initial conditions $w_{n,k} = 0$ if $n, k \leqslant 0$ with $(n, k) \neq (0, 0)$, and $w_{0,0} = 1$. If $\tilde{w}_{n,k}$ denotes the number of walks of length n and final altitude k that never exit the upper half plane, then $\tilde{w}_{n,k}$ also satisfies recurrence (16), but with the additional initial conditions $\tilde{w}_{n,k} = 0$ for all $k < 0$. Then the bridges (resp. excursions, meanders) are counted by the numbers $w_{n,0}$ (resp. $\tilde{w}_{n,0}$, $\sum_k \tilde{w}_{n,k}$).

One can compute these numbers by unrolling the recurrence relation (16). Each use of the recurrence costs $O(d)$ ops., and in the worst case one has to compute $O(dN^2)$ terms of the sequence (for example, if the step set is $S = \{(1, 1), \ldots, (1, d)\}$). This leads to the computation of each of the generating series in $O(d^2N^2)$ ops.

This quadratic complexity in N is unsatisfactory, and any method that requires the complete expansion of the generating series $W(x, y)$ is bound to be quadratic in N. The two other methods that we are going to present are designed to achieve linear or quasi-linear complexity in N. As will be explained, this comes at the cost of a precomputation that must be taken into account in the analysis.

Using algebraic equations. In Banderier and Flajolet (2002, §2.3), a method relying on the algebraicity of B, E and M (Fact 22(2))) is suggested. The series E and M can be expressed as products in terms of the small branches of the characteristic polynomial Γ_S (see Banderier and Flajolet, 2002, Th. 1, Cor. 1). From there, a polynomial equation can be obtained using the Platypus algorithm (Banderier and Flajolet, 2002, §2.3), which computes a polynomial canceling the products of a fixed number of roots of a given polynomial. Given a polynomial equation $P(z, E) = 0$, another one for B can be deduced from the relation $B = zE'/E + 1$ as $\text{Resultant}_E((B - 1)EP_E + zP_z, P)$.

Once a polynomial equation is known for one of these three series, it can be used to compute a linear recurrence with polynomial coefficients satisfied by its coefficients (Chudnovsky and Chudnovsky 1986, 1987; Bostan et al., 2007). The naive algorithm introduced above provides a way to compute a sufficiently large number of initial conditions to unroll this recurrence. (For a quantitative result on the required number of initial conditions, see Corollary 28 below.) This method produces an algorithm that computes the first N terms of B, E and M in $O(N)$ ops. For this to be an improvement over the naive method for large N, the dependence on d of the constant in the $O()$ should not be too large and the precomputation not too costly.

Indeed, the cost of the precomputation of an algebraic equation is not negligible. The bound $\binom{d}{u^-}$ on the degrees of equations for excursions has been obtained by Bousquet-Mélou, and showed to be tight for a specific family of step sets, as well as generically (Bousquet-Mélou, 2008, §2.1). This bound

may be exponentially large with respect to d. Empirically, the polynomials for B and M are similarly large.

The situation for differential equations and recurrences is different: B satisfies a differential equation of only polynomial size (see below), whereas (empirically), those for E and M have a potentially exponential size. These sizes then transfer to the corresponding recurrences and thereby to the constant in the complexity of unrolling them. The purpose of Theorem 24 below is to give explicitly the polynomial dependence in d when using this method, showing at the same time that a true improvement over the naive method can be achieved.

Example 23. With the step set $S = \{(1, d), (1, 1), (1, -d)\}$ and $d \geqslant 2$, the counting series W_S equals

$$W_S(x, y) = \frac{y^d}{y^d - x(1 + y^{d+1} + y^{2d})}.$$

Experiments indicate that the minimal polynomial of $B_S(x)$ has bidegree $(2d\binom{2d-2}{d-1}, \binom{2d}{d})$, exhibiting an exponential growth in d. On the other hand, they show that $B_S(x)$ satisfies a linear differential equation of order $2d - 1$ with coefficients of degree $d^2 + 3d - 2$ for even d, and $d^2 + 3d - 4$ for odd d.

New method. We now give a method that runs in quasi-linear time (with respect to N) and avoids the computation of an algebraic equation. Our method relies on the fact that periods of rational functions such as the one in Part *(3)* of Fact 22 satisfy differential equations of polynomial size in the degree of the input rational function (Bostan et al., 2010). We summarize our results in the following theorem, and then go over the proof in each case individually.

Theorem 24. *Let S be a finite set of simple steps and $d = u^- + u^+$. The series B_S (resp. E_S and M_S) can be expanded at order N in $O(d^2 N)$ ops. (resp. $\tilde{O}(d^2 N)$ ops.), after a precomputation in $\tilde{O}(d^5)$ ops.*

6.3. Fast algorithms

Bridges. To expand $B(x)$, we rely on Fact 22(3). The formula can be written $B = (1/2\pi i) \oint W(x, y)\frac{dy}{y}$, the integration path being a circle inside a small annulus around the origin (Banderier and Flajolet, 2002, proof of Th. 1). Moreover, $W(x, y)/y$ is of the form P/Q, where bideg $Q \leqslant (1, d)$ and bideg $P \leqslant (0, d - 1)$. Since P and Q are relatively prime and Q is primitive with respect to y, Algorithm **HermiteTelescoping** (Bostan et al., 2010, Fig. 3) computes a telescoper for P/Q, which is also a differential equation satisfied by B. By Fact 6*(2)*, the resulting differential equation has order at most d and degree $O(d^2)$, and is computed using $\tilde{O}(d^5)$ ops. This differential equation can be turned into a recurrence of order $r = O(d^2)$ in quasi-optimal time (see the discussion after Bostan and Schost, 2005, Cor. 2). We may use it to expand $B(x)$ mod x^N in $O(d^2 N)$ ops., once enough initial conditions are known. Again, the initial conditions are computed by means of the direct method. The only remaining question is the number of initial conditions needed. Indeed, the recurrence may be singular, i.e., its leading coefficient may have positive integer roots. If we denote by α the largest such root, then we need to compute the first terms of the recurrence up to $\max(r - 1, \alpha)$. In order not to break the flow of reading, we postpone the discussion on the size of α to the next section. For now, we only state the result.

Proposition 25. *Let S be a set of simple steps, and $d = \max_{(1,u),(1,v)\in S} |u - v|$. Then the largest integer root of the leading term of the recurrence computed by Algorithm 4 is at most $O(d^3)$.*

Proof. See Section 6.4. □

Thus, a sufficient number of initial conditions is computed with $O(d^5)$ ops. by the direct method, and the total cost of the precomputation is $\tilde{O}(d^5)$, as announced.

Algorithm 4 Expanding the generating functions of bridges, excursions and meanders.

Algorithm **Walks**(S, N)

Input	A set S of simple steps and an integer N
Output	B_S, E_S, M_S mod x^{N+1}

$F \leftarrow W(x, y)/y$ [case B, E] or $W(x, y)/(1 - y)$ [case M]
$D \leftarrow$ **HermiteTelescoping**(F) (Bostan et al., 2010, Fig. 3)
$R \leftarrow$ the recurrence of order r associated with D
$I \leftarrow [y^0]W(x, y)$ mod x^{r+1} [case B, E]
    $[y^0]yW(x, y)/(1 - y)$ mod x^{r+1} [case M]
$B \leftarrow [y^0]W(x, y)$ mod x^{N+1} (from R, I)
$A \leftarrow [y^0]yW(x, y)/(1 - y)$ mod x^{N+1} (from R, I)
$E \leftarrow \exp\left(\int (B(x) - 1)/x\,dx\right)$ mod x^{N+1}
$M \leftarrow \exp\left(-\int (A(x)/x)/(1 - \Gamma(1)x)\,dx\right)$ mod x^{N+1}
return B, E, M

Excursions. If $B(x)$ mod x^{N+1} is known, it is then possible to recover $E(x)$ mod x^{N+1} thanks to Fact 22(4). Expanding $E(x)$ comes down to the computation of the exponential of a series, which can be performed using $\tilde{O}(N)$ ops. (Fact 1(4)).

Meanders. As in the case of excursions, the logarithmic derivative of $M(x)$ is recovered from a sum of residues by the following.

Proposition 26. *The series W and M are related through*

$$A(x) = [y^0]\frac{y}{1 - y}W(x, y), \quad M(x) = \frac{\exp\left(-\int \frac{A(x)}{x}\,dx\right)}{1 - x\Gamma(1)}.$$

Proof. Denote by $y_1, \ldots, y_{u^-}$ the small branches of the polynomial $y^{u^-} - xy^{u^-}\Gamma(y)$. Then M is given as (Banderier and Flajolet, 2002, Cor. 1):

$$M(x) = \frac{1}{1 - x\Gamma(1)}\prod_{i=1}^{u^-}(1 - y_i).$$

On the other hand,

$$A(x) = \frac{1}{2\pi i}\oint \frac{W(x, y)}{1 - y}\,dy$$

$$= \sum_{i=1}^{u^-}\text{Residue}_{y=y_i(x)}\left(\frac{1}{(1 - y)(1 - x\Gamma(y))}\right) = -\sum_{i=1}^{u^-}\frac{1}{(1 - y_i)x\Gamma'(y_i)},$$

where the integral has been taken over a circle around the origin and the small branches. Differentiating the equation $1 - x\Gamma(y) = 0$ with respect to x leads to $-x\Gamma'(y_i) = 1/(xy_i')$, whence $A(x) = x\sum_{i=1}^{u^-}y_i'/(1 - y_i)$. Therefore, $\prod(1 - y_i) = \exp(-\int A/x\,dx)$, finishing the proof. □

Thus we apply the same method as in the case of the excursions. We first compute a differential equation for $A(x)$ using the method of Bostan et al. (2010). The computation of the initial conditions for A can also be performed naively from its definition as a constant term, by simply expanding $yW(x, y)/(1 - y)$. The formula of the proposition then recovers $M(x)$. The complexity analysis goes exactly as in the previous case, giving a global cost of $\tilde{O}(d^5)$ ops.

6.4. Singular recurrences

We now come back to the problem of singular recurrences. In our context, the recurrences that we come across have a very specific structure: they are associated with differential resolvents of polyno-

mials. (The differential resolvent of a polynomial is the least order differential operator canceling all of its roots.) This structure can be exploited to derive bounds on the singularities of our recurrences.

If $P \in \mathbb{K}[x][y]$ is a polynomial, consider the recurrence associated with its differential resolvent L. The leading coefficient of this recurrence is called the indicial polynomial of L at 0. Its largest integer root will be denoted α. The fundamental idea is that there exists a Laurent series solution of L which has valuation α (Ince, 1956, §15.31) Therefore, it is sufficient to find bounds on the valuations of the solutions of L. This is done in the following theorem.

Theorem 27. *Let P be a polynomial in $\mathbb{K}[x][y]$, of bidegree at most (d_x, d_y), and L be the differential resolvent of P. Then all the Laurent series solutions $y(x)$ of L uniformly satisfy*

$$\mathrm{val}_x(y(x)) = O(d_x d_y^2).$$

Proof. Choose a subfamily $y_1, y_2, \ldots, y_n$ of the Puiseux series roots of P that constitutes a basis of the solution space of the resolvent (in particular, $n \leqslant d_y$). Let $y = \sum_{i=1}^n \lambda_i y_i$ be a Laurent series solution of the differential resolvent of P.

Then the fact that $\mathrm{val}(f') \geqslant \mathrm{val}(f) - 1$ for any Laurent series $f \in \mathbb{K}((x))$ implies that

$$\mathrm{val}(\mathrm{Wr}(y, y_2, \ldots, y_n)) \geqslant \mathrm{val}(y) + \sum_{i=2}^n \mathrm{val}(y_i) - \binom{n}{2}.$$

By the multilinearity of the Wronskian, the left-hand side of this inequality is nothing more than $\mathrm{val}(\mathrm{Wr}(y_1, y_2, \ldots, y_n))$. On the other hand, the absolute values of the valuations of the y_i's are bounded by $\max(d_x, d_y)$ (because they are slopes of edges in the Newton polygon of P). A bound for $\mathrm{val}(y)$ is thus obtained:

$$\mathrm{val}(y) \leqslant \mathrm{val}(\mathrm{Wr}(y_1, y_2, \ldots, y_n)) + (d_y - 1)\max(d_x, d_y) + \frac{d_y(d_y - 1)}{2}.$$

The proof is then reduced to showing that $\mathrm{val}(\mathrm{Wr}(y_1, y_2, \ldots, y_n)) = O(d_x d_y^2)$. This is very similar to the computations conducted in Bostan et al. (2007, §2.2). We start by recalling some facts that are proved there. There exist polynomials $W_k \in \mathbb{K}[x, y]$ such that for all $i \in \{1, 2, \ldots, n\}$ and all $k \geqslant 1$, the derivative $y_i^{(k)}$ can be expressed as

$$y_i^{(k)} = \frac{W_k(x, y_i)}{P_y(x, y_i)^{2k-1}}.$$

Moreover, the polynomials W_k satisfy

$$\deg_x W_k \leqslant (2d_x - 1)k - d_x, \quad \deg_y W_k \leqslant 2(d_y - 1)k - d_y + 2. \tag{17}$$

It follows that $D = \prod_{i=1}^n P_y(x, y_i)^{2n-3} \in \mathbb{K}[x, y_1, y_2, \ldots, y_n]$ is a polynomial such that $\mathrm{Wr}(y_1, y_2, \ldots, y_n) \cdot D \in \mathbb{K}[x, y_1, y_2, \ldots, y_n]$. We will denote by R this last polynomial. R is the determinant of the matrix

$$\mathcal{N} = \begin{pmatrix} y_1 P_y(x, y_1)^{2n-3} & \cdots & y_n P_y(x, y_n)^{2n-3} \\ W_1(x, y_1) P_y(x, y_1)^{2n-4} & \cdots & W_1(x, y_n) P_y(x, y_n)^{2n-4} \\ W_2(x, y_1) P_y(x, y_1)^{2n-6} & \cdots & W_2(x, y_n) P_y(x, y_n)^{2n-6} \\ \vdots & \vdots & \vdots \\ W_{n-1}(x, y_1) & \cdots & W_{n-1}(x, y_n) \end{pmatrix}.$$

R is an anti-symmetric polynomial in $y_1, y_2, \ldots, y_n$, but R^2 is symmetric, as well as D, so we can apply Lemma 13 to see that R^2 and D belong to $\mathbb{K}(x)$. Therefore, the equality

$$\mathrm{Wr}(y_1, y_2, \ldots, y_n) = \frac{R}{D}$$

shows that $\mathrm{Wr}(y_1, y_2, \ldots, y_n)$ is the square root of a rational function in x. We are going to use this structure and Lemma 13 to derive the desired bound on the valuation of the Wronskian determinant.

If $\det(\mathcal{N})$ is viewed as a polynomial in $\mathbb{K}[x, y_1, y_2, \ldots, y_n]$, then

$$\deg_x \det(\mathcal{N})^2 \leqslant 2 \sum_{k=0}^{n-1} \left((2n-3)d_x - k \right)$$

$$\leqslant 2n(2n-3)d_x + n(n-1),$$

and for all $i \in \{1, 2, \ldots, n\}$,

$$\deg_{y_i} \det(\mathcal{N}) \leqslant 2(2n-3)d_y - 2(2n-4).$$

Similarly, when D is viewed as a polynomial in $\mathbb{K}[x, y_1, y_2, \ldots, y_n]$, we have:

$$\deg_x D = n(2n-3)d_x, \quad \deg_{y_i} D = (2n-3)(d_y - 1).$$

Applying Lemma 13, we deduce that, denoting by $p(x)$ the leading coefficient of $P(x, y)$,

$$\mathrm{Wr}(y_1, y_2, \ldots, y_n) = \frac{U(x)}{p(x)V(x)},$$

where

$$\deg_x U^2 \leqslant 2n(2n-3)d_x + n(n-1) + 2(2n-3)d_x d_y - 2(2n-4)d_x.$$

Finally, the inequalities $\mathrm{val}(\mathrm{Wr}(y_1, y_2, \ldots, y_n)) \leqslant \frac{1}{2}\mathrm{val}(U^2) \leqslant \frac{1}{2}\deg_x(U^2)$ and $n \leqslant d_y$ yield

$$\mathrm{val}(\mathrm{Wr}(y_1, y_2, \ldots, y_n)) = O(d_x d_y^2),$$

which concludes the proof. □

We immediately deduce the following corollary on the number of initial conditions required to expand an algebraic power series.

Corollary 28. *Let $P \in \mathbb{K}[x, y]$ be a polynomial of bidegree bounded by (d_x, d_y). Let R be the recurrence associated with the differential resolvent of P. Then the largest integer root of the leading coefficient of R is at most $O(d_x d_y^2)$.*

Proof. Immediate from the theorem and the discussion that precedes it. □

We are now able to prove Proposition 25.

Proof of Proposition 25. We only treat the case where the recurrence is computed for B, and the proof transposes directly to the case of A. Let S and d be as in the Proposition, and denote by P the minimal polynomial of B. Then the recurrence computed by Algorithm 4 is associated with the minimal annihilating differential operator for B, which is also the differential resolvent of P. We denote it by L_P. Now since $B = [y^{-1}]W(x, y)/y$, it can be written as a sum of residues similar to formula (8). If we denote by R the polynomial that cancels these residues, then P divides $\Sigma_c R$ for some c. This implies in particular that all the solutions of L_P are linear combinations of the roots of R. Thus, if L_R is the differential resolvent of R, then all the solutions of L_P are solutions of L_R. Since W has bidegree $(1, d)$, Theorem 8 and Theorem 27 show that all the roots of P have valuation at most $O(d^3)$, and the result follows. □

7. Conclusion

We gave a complete and efficient algorithm that calculates a polynomial equation satisfied by the diagonal of a bivariate rational function in characteristic 0. Generically, the degree in Δ of the polynomial $P(t, \Delta)$ output by the algorithm is optimal. The bound on the degree in t is not tight. The gap between this bound and the actual degrees is not yet fully understood: it is already present for the Rothstein–Trager and Bronstein resultants. Our complexity results are given in the arithmetic complexity model. The corresponding study in the binary model remains to be done.

The case of positive characteristic requires different methods and algorithms. In that case, diagonals are algebraic even for rational functions with more than two variables. To the best of our knowledge, these questions have never been studied from the complexity viewpoint. One possible direction is to try and make effective the proof by Furstenberg that these diagonals are algebraic (Furstenberg, 1967). Some work has also been done by Adamczewski and Bell (2013) who among other things studied how the sizes of the polynomial equations satisfied by diagonals vary with the characteristic of the base field.

Acknowledgements

References

Adamczewski, B., Bell, J.P., 2013. Diagonalization and rationalization of algebraic Laurent series. Ann. Sci. Éc. Norm. Supér. (4) 46 (6), 963–1004.

Banderier, C., Flajolet, P., 2002. Basic analytic combinatorics of directed lattice paths. Theor. Comput. Sci. 281 (1–2), 37–80.

Bostan, A., Chen, S., Chyzak, F., Li, Z., 2010. Complexity of creative telescoping for bivariate rational functions. In: ISSAC'10. ACM Press, pp. 203–210.

Bostan, A., Chyzak, F., Lecerf, G., Salvy, B., Schost, É., 2007. Differential equations for algebraic functions. In: ISSAC'07. ACM Press, pp. 25–32.

Bostan, A., Dumont, L., Salvy, B., 2015. Algebraic diagonals and walks. In: ISSAC'15. ACM Press, pp. 77–84.

Bostan, A., Flajolet, P., Salvy, B., Schost, É., 2006. Fast computation of special resultants. J. Symb. Comput. 41 (1), 1–29.

Bostan, A., Lairez, P., Salvy, B., 2013. Creative telescoping for rational functions using the Griffiths–Dwork method. In: ISSAC'13. ACM Press, pp. 93–100.

Bostan, A., Schost, É., 2005. Polynomial evaluation and interpolation on special sets of points. J. Complex. 21 (4), 420–446.

Bousquet-Mélou, M., 2006. Rational and algebraic series in combinatorial enumeration. In: International Congress of Mathematicians. EMS, pp. 789–826.

Bousquet-Mélou, M., 2008. Discrete excursions. Sémin. Lothar. Comb. 57, B57d. 23 pp.

Bousquet-Mélou, M., Petkovšek, M., 2000. Linear recurrences with constant coefficients: the multivariate case. Discrete Math. 225 (1–3), 51–75.

Bronstein, M., 1992. Formulas for series computations. Appl. Algebra Eng. Commun. Comput. 2 (3), 195–206.

Bürgisser, P., Clausen, M., Shokrollahi, M.A., 1997. Algebraic Complexity Theory. Grundlehren der Mathematischen Wissenschaften, vol. 315. Springer.

Canny, J.F., Kaltofen, E., Lakshman, Y.N., 1989. Solving systems of nonlinear polynomial equations faster. In: ISSAC'89, pp. 121–128.

Christol, G., 1985. Diagonales de fractions rationnelles et équations de Picard–Fuchs. In: Study Group on Ultrametric Analysis, 12th year, 1984/85, No. 1 (Exp. No. 13). Paris, pp. 1–12.

Chudnovsky, D.V., Chudnovsky, G.V., 1986. On expansion of algebraic functions in power and Puiseux series, I. J. Complex. 2 (4), 271–294.

Chudnovsky, D.V., Chudnovsky, G.V., 1987. On expansion of algebraic functions in power and Puiseux series, II. J. Complex. 3 (1), 1–25.

Deligne, P., 1984. Intégration sur un cycle évanescent. Invent. Math. 76 (1), 129–143.

Denef, J., Lipshitz, L., 1987. Algebraic power series and diagonals. J. Number Theory 26 (1), 46–67.

Fliess, M., 1974. Sur divers produits de séries formelles. Bull. Soc. Math. Fr. 102, 181–191.

Furstenberg, H., 1967. Algebraic functions over finite fields. J. Algebra 7 (2), 271–277.

von zur Gathen, J., Gerhard, J., 2003. Modern Computer Algebra, 2nd ed.. Cambridge Univ. Press.

Gessel, I.M., 1980. A factorization for formal Laurent series and lattice path enumeration. J. Comb. Theory, Ser. A 28 (3), 321–337.

Haible, B., 1997. The diagonal of a rational function, preprint.

Hautus, L.J., Klarner, D.A., 1971. The diagonal of a double power series. Duke Math. J. 38, 229–235.

van der Hoeven, J., Schost, É., 2013. Multi-point evaluation in higher dimensions. Appl. Algebra Eng. Commun. Comput. 24 (1), 37–52.

Ince, E.L., 1956. Ordinary Differential Equations. Dover Publications, New York. Reprint of the 1926 edition.

Lairez, P., 2016. Computing periods of rational integrals. Math. Comput. 85 (300), 1719–1752.

Lecerf, G., 2008. Fast separable factorization and applications. Appl. Algebra Eng. Commun. Comput. 19 (2), 135–160.

Lecerf, G., Schost, É., 2003. Fast multivariate power series multiplication in characteristic zero. SADIO Electron. J. Inform. Oper. Res. 5, 1–10.

Lipshitz, L., 1988. The diagonal of a D-finite power series is D-finite. J. Algebra 113 (2), 373–378.

Pan, V.Y., 1994. Simple multivariate polynomial multiplication. J. Symb. Comput. 18 (3), 183–186.

Pan, V.Y., 2000. New techniques for the computation of linear recurrence coefficients. Finite Fields Appl. 6 (1), 93–118.

Pochekutov, D.Y., 2009. Diagonals of the Laurent series of rational functions. Sib. Mat. Zh. 50 (6), 1370–1383.

Pólya, G., 1921. Sur les séries entières, dont la somme est une fonction algébrique. Enseign. Math. 22, 38–47.

Rothstein, M., 1976. Aspects of Symbolic Integration and Simplification of Exponential and Primitive Functions. Ph.D. thesis.

Safonov, K.V., 1987. On conditions for the sum of a power series to be algebraic and rational. Mat. Zametki 41 (3), 325–332. English translation: Math. Notes 41 (3) (1987) 185–189.

Schönhage, A., 1982. The fundamental theorem of algebra in terms of computational complexity. Tech. rep., Tübingen.

Stanley, R.P., 1999. Enumerative Combinatorics, vol. 2. Cambridge University Press.

Trager, B.M., 1976. Algebraic factoring and rational function integration, SYMSAC'76. ACM, pp. 219–226.

van der Waerden, B.L., 1949. Modern Algebra, vol. I. Frederick Ungar Publ. Co.

Walker, R.J., 1978. Algebraic Curves. Springer-Verlag, New York. Reprint of the 1950 edition.

Yap, C.K., 2000. Fundamental Problems of Algorithmic Algebra. Oxford University Press Inc., New York.