Effective Asymptotics of Linear Recurrences with Rational Coefficients

Xavier Gourdon[*] Xavier.Gourdon@inria.fr

Bruno Salvy[*] Bruno.Salvy@inria.fr

Abstract

We give algorithms to compute the asymptotic expansion of solutions of linear recurrences with rational coefficients and rational initial conditions in polynomial time in the order of the recurrence.

Introduction

We investigate sequences defined by a recurrence of the form

$$a_k u_{n+k} + a_{k-1} u_{n+k-1} + \cdots + a_0 u_n = 0, \qquad (1)$$

where the coefficients a_k and the initial conditions belong to $\mathbb{Q}$. This is probably the most simple type of recurrence one may encounter. Recurrences of this type are ubiquitous in many fields of applications (see [3] for numerous examples and references). Among the approximately 2300 sequences listed in Sloane's book [18], one can estimate that about 13% are of this type [13]. In the rest of this paper "linear recurrence" always means "linear recurrence with rational coefficients" and we shall refer to u_n as a "linear recurrent sequence".

Surprisingly, some problems related to linear recurrences remain open [3], and specially problems related to effectivity. Our aim in this paper is to describe an algorithm that computes an asymptotic expansion of a sequence obeying (1) in polynomial time in the order k of the recurrence. It is quite simple to find the asymptotic expansion of Fibonacci numbers with traditional tools, but these tools break down for ill-conditioned recurrences. The algorithm we describe works without any limitation on the value of k or those of the coefficients.

Given a recurrence such as (1), one usually computes its general term as a sum of *exponential polynomials* of the form $\sum_{k=0}^{N} p_k n^k \lambda^n$, where λ is an algebraic number. In Section 1 we shall describe an algorithm computing the coefficients p_k *without factoring any polynomial.* This general term does not solve the problem of asymptotic behaviour. To form a proper asymptotic expansion one has to order the moduli of the algebraic numbers λ occurring in the general terms. The problem which will occupy most of this paper is: How can one perform such an ordering *exactly*, i.e. we prove that the algorithms we propose work on the whole class of recurrences (1). We shall use techniques from computer algebra to free ourselves from problems of ill-conditioning related to the use of floating-point values. The result is an algorithm which, given a positive integer p and a linear recurrence (1) together with its initial conditions—or equivalently a rational function in $\mathbb{Q}(x)$ (see below)—outputs the p first exponential polynomials of the asymptotic expansion of the solution u_n of (1) as n tends to infinity.

As an example consider the recurrence

$$14448 u_n + 15136 u_{n+1} - 5761 u_{n+2} - 990 u_{n+3} + 315 u_{n+4} = 0. \qquad (2)$$

Its characteristic polynomial has four simple roots:

$$\alpha_1 = \frac{3}{2} + \sqrt{\frac{53}{12}}, \quad \alpha_2 = \frac{3}{2} - \sqrt{\frac{53}{12}}, \quad \alpha_3 = \sqrt{\frac{454}{35}}, \quad \alpha_4 = -\alpha_3.$$

[*] Algorithms Project, INRIA, 78153 Le Chesnay Cedex, France.

To compute the asymptotic behaviour of u_n, it is necessary to decide which roots have the largest modulus. In this example the roots are given explicitly and this make it possible (but not trivial) to sort them by modulus. The techniques we develop do not require an explicit expression of the roots.

We describe two essentially different decision procedures to compute the asymptotic expansion. The first approach, purely algebraic, completely avoids factorizations. It is made expensive by the increase of degrees due to resultant computations. Currently this is the most natural computer algebra approach to the problem, and the most easily implemented. However, as soon as $p \geqslant 2$, its cost becomes potentially exponential in the order of the recurrence.

The second approach is perhaps more natural: it consists in using numerical approximations. In our example, the approximations

$$\alpha_1 = 3.601586702\ldots, \alpha_2 = -0.601586702\ldots, \alpha_3 = 3.601586951\ldots$$

make it possible to conclude. This example also shows that these numerical approximations may require a lot of precision. To avoid ill-conditioning, we use *guaranteed* numerical approximations and we show that this can be done in polynomial time in the order of the recurrence. Numerical approximations have long been banned from computer algebra because of the reluctance inherited from fixed precision routines. However, with the arbitrary precision provided by most computer algebra systems, we feel that it is time for floating point numbers to be rehabilitated in computer algebra.

The first step of the algorithm is to compute a suitable partial fraction decomposition of the generating function of u_n. Since factorization of polynomials is known to be polynomial-time but depressingly expensive, we shall avoid factorization and rely instead on a recent decomposition algorithm [2]. This is described in Section 1. In Section 2 and 3, we address the problem of comparing the moduli of the singularities (corresponding to the roots of the characteristic polynomial). As opposed to what happens usually in most algorithms involving algebraic numbers, we have to distinguish between roots of a given polynomial. A first method is described in Section 2, based on an algorithm [6] for comparing real algebraic numbers. At this stage, we can produce the desired asymptotic expansion. Section 3 describes a numerical alternative to the algebraic algorithms of Section 2, where we show how to get exact information from numerical values. We prove that this can be done with a cost that is lower than that of the algebraic method. In Section 4, we study optimizations that can be applied to subparts of our algorithm in practical cases. In particular we show there how rough numerical estimates can be used fruitfully. We conclude in Section 5 with a few examples taken from classical combinatorics.

1 Outline of the algorithm

1.1 Generating function

One can translate (1) into the rational generating function $\sum u_n z^n$ with $O(k^2)$ rational operations: the generating function of the sequence (1) is

$$\frac{\sum_{i=0}^{k} a_i \sum_{j=0}^{i-1} u_j z^{k-i+j}}{\sum_{i=0}^{k} a_{k-i} z^i}.$$

The reciprocal conversion is also easy.

From the asymptotic point of view, the generating function approach enables us to use tools from complex analysis, like residue computation, which prove very effective. Because of the low cost of the conversion from a linear recurrence to the generating function, from now on we shall be concerned with rational functions only. Thus the input of our algorithm is a function $f \in \mathbb{Q}(z)$ regular at the origin, together with a positive integer p, and its output consists of the first p terms of the asymptotic expansion of $[z^n]f(z)$—the nth Taylor coefficient of f at the origin—as n tends to infinity.

1.2 Exact formula

In this section, we derive an exact formula for $[z^n]f(z)$, based on a partial fraction decomposition that does not require factorization. This allows for both an efficient implementation and possible future extensions to rational functions with parameters or non-rational coefficients.

Algorithm 1 (Exact formula) *Let $f(z) = P(z)/Q(z) \in \mathbb{Q}(z)$, with P and Q two relatively prime polynomials and $\deg(P) < \deg(Q)$. To compute $[z^n]f(z)$,*

1. *Compute $Q = D_1 D_2^2 \cdots D_n^n$ the square-free decomposition of Q. (Each D_i is a square-free polynomial.)*
2. *Using the decomposition algorithm [2], compute polynomials $P_{i,j} \in \mathbb{Q}[z]$ such that*

$$f(z) = \frac{P(z)}{Q(z)} = \sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{D_i(\alpha)=0} \frac{P_{i,j}(\alpha)}{(z-\alpha)^j}, \tag{3}$$

   with $\deg(P_{i,j}) < \deg(D_i)$. This requires only gcd computations.

3. *For each (i,j) such that $\gcd(P_{i,j}, D_i) \neq 1$, write $D_i = G_i H_i$, where $G_i = \gcd(P_{i,j}, D_i)$ and rewrite all terms in (3) involving the polynomial D_i as*

$$\sum_{D_i(\alpha)=0} \frac{P_{i,j}(\alpha)}{(z-\alpha)^j} = \sum_{G_i(\alpha)=0} \frac{A_{i,j}(\alpha)}{(z-\alpha)^j} + \sum_{H_i(\alpha)=0} \frac{B_{i,j}(\alpha)}{(z-\alpha)^j}$$

   where $A_{i,j}$ and $B_{i,j}$ are obtained by Euclidian division of $P_{i,j}$ by G_i and H_i. Repeat this process until all gcd's are units. This gives a factorization of each D_i in the form $D_i = D_{i,1} \cdots D_{i,n_i}$ and the partial fraction decomposition has the form

$$f(z) = \sum_{i=1}^{n}\sum_{j=1}^{n_i}\sum_{k=1}^{i}\sum_{D_{i,j}(\alpha)=0} \frac{P_{i,j,k}(\alpha)}{(z-\alpha)^k}, \tag{4}$$

   each $P_{i,j,k}$ being a polynomial with rational coefficients, $\deg(P_{i,j,k}) < \deg(D_{i,j})$.

4. *From this we get the value of $[z^n]f(z)$:*

$$[z^n]f(z) = \sum_{i=1}^{n}\sum_{j=1}^{n_i}\sum_{k=1}^{i}\sum_{D_{i,j}(\alpha)=0} \frac{(n+1)\cdots(n+k-1)}{(k-1)!} \cdot \frac{P_{i,j,k}(\alpha)}{\alpha^{n+k}}. \tag{5}$$

Step 1 is computed by repetitively differentiating and computing gcd's. Step 3 guarantees that each $P_{i,j,k}(\alpha)$ in (4) is non zero. Step 4 is a consequence of the usual series expansion of $(\alpha - z)^{-k}$. It is clear that Algorithm 1 runs in polynomial time. We do not worry about its complexity since the following algorithms are much more expensive.

Example Let $f(z)$ be the following input

$$\frac{z^2+2}{(1-z)^2(1+z-2z^2-z^3-2z^5)}.$$

In this case, the square-free decomposition of the denominator Q of f is given by $D_2 = 1-z$, $D_1 = 1+z-2z^2-z^3-2z^5$ (note that D_1 is not irreducible). Step (2) of the algorithm then produces the following decomposition

$$f(z) = -\frac{1}{(1-z)^2} - \frac{14/3}{1-z} + \sum_{D_2(\alpha)=0} \frac{h(\alpha)}{\alpha - z},$$

where $h(\alpha) = (11530\alpha^4 - 7778\alpha^3 + 11325\alpha^2 + 3889\alpha - 8080)/93$. Since h and D_2 are relatively prime, Step 3 does not do anything, and then Step 4 produces the result:

$$[z^n]f(z) = -(n+1) - \frac{14}{3} + \sum_{D_2(\alpha)=0} h(\alpha)\alpha^{-n-1}.$$

To get an asymptotic expansion from this, we have to sort the moduli of the roots of D_2 and compare them to 1. This is addressed in the following sections.

1.3 Specification of the algorithm

As already mentioned, the asymptotic expansion is governed by the successive "layers" of singularities of the rational function, sorted by increasing moduli. We shall need several ways to describe these moduli.

Notation *For $P(z) \in \mathbb{Q}[z]$, we note $\rho_1(P) < \rho_2(P) < \cdots < \rho_k(P)$ the* distinct *moduli of the roots of P in increasing order. When there is no ambiguity, we simply denote these numbers ρ_m, $1 \leqslant m \leqslant k$. We also note $Z_m(P)$ the set of zeroes of P whose modulus is $\rho_m(P)$, and $Z_m^+(P)$ the subset of $Z_m(P)$ whose elements have positive imaginary part.*

We first state the form of the output on the example of $f(z)$ from our previous section. The first four terms of the asymptotic expansion of $[z^n]f(z)$ as given by our algorithm are

$$\left[(-1)^{n+1}h(-\rho_1)\right]\frac{1}{\rho_1^{n+1}} + \left[h(\rho_2) + (-1)^{n+1}h(-\rho_2)\right]\frac{1}{\rho_2^{n+1}} + \left[\frac{-(n+1)}{\rho_3} - \frac{14}{3}\right]\frac{1}{\rho_3^{n+1}}$$
$$+\Big[\sum_{\beta\in Z_3^+(D_2)} \left(h(\beta) + h(\overline{\beta})\right)\cos[(n+1)\arg(\beta)]\Big]\frac{1}{\rho_4^{n+1}} + o\left(\frac{1}{\rho_4^n}\right),$$

together with the following information $\rho_1 = \rho_1(D_2)$, $\rho_2 = \rho_2(D_2)$, $\rho_3 = 1$, $\rho_4 = \rho_3(D_2)$ and $|Z_3^+(D_2)| = 1$. If requested, we can also give numerical approximations of the ρ_i and β.

All the features of the general case are present in this example. We now state precisely the specification of the algorithm, which we encourage the casual reader to skip. The input of our algorithm consists of $f(z) = P(z)/Q(z) \in \mathbb{Q}(z)$ and an integer $p \geqslant 1$. The output is the following asymptotic expansion of $[z^n]f(z)$:

$$[z^n]f(z) = \sum_{\ell=1}^{p} \frac{c_\ell(n)}{\rho_\ell^n} + o\left(\frac{1}{\rho_p^n}\right), \qquad (6)$$

where

$$c_\ell(n) = H_{\ell,0}(n) + (-1)^n H_{\ell,1}(n) + \sum_{j\geqslant 2}\sum_{\beta\in Z^+_{n_{\ell,j}}(D_{\ell,j})}\sum_k \frac{H_{\ell,j,k}(n)}{\rho_\ell^k}\cos[(n+k)\arg(\beta)]$$

and explicit values are given for the coefficients of the polynomials $H_{\ell,0}$ and $H_{\ell,1}$ (in $\mathbb{Q}(\rho_\ell)[z]$), $D_{\ell,j}$ (in $\mathbb{Q}[z]$) and $H_{\ell,j,k}$ (in $\mathbb{Q}(\beta,\overline{\beta})[z]$), as well as for the number of elements of the Z^+ involved and a definition of ρ_ℓ either explicit or as $\rho_\ell = \rho_{\ell'}(D_{i,j})$ for some divisor of Q.

In practice, the program will be able to give numerical approximations of the moduli and the roots implicitly defined. Note that, because of the trigonometric functions involved in the coefficients, the expansion (6) is not of Poincaré type. Instead, we resort to the extended definition of Schmidt [16] (see also [7]), according to which an asymptotic expansion is a sum of the form

$$f(x) = \sum_{k=1}^{p} a_k(n)\cdot f_k(n) + r(n) \qquad (7)$$

where as n tends to infinity $f_{k+1}(n) = o[f_k(n)]$, $(1 \leqslant k \leqslant p-1)$; $r(n) = o[f_p(n)]$; and $a_k(n)$ are bounded functions of n that do not tend to zero.

1.4 Main algorithm

Starting from the partial fraction decomposition (4), we need to order the moduli of the roots of the $D_{i,j}$ in (5) and find those roots that are purely real along with their signs. Our algorithm is based on the resolution of the two following computational problems.

Task 1 (Ordering the moduli) *Given $Q = \prod_{i,j} D_{i,j}^i$ a square-free decomposition of $Q \in \mathbb{Q}[z]$ and p a non-negative integer, compute for each (i,j) and for each k, $1 \leqslant k \leqslant p$, the number of roots of $D_{i,j}$ of modulus $\rho_k(Q)$.*

Task 2 (Real roots and their signs) *Given $P \in \mathbb{Q}[z]$ a square-free polynomial and k a non-negative integer, compute the number $p \in \{0,1\}$ (resp. $n \in \{0,1\}$) of positive (resp. negative) real roots of P of modulus $\rho_k(P)$.*

Most of the rest of this paper is devoted to algorithmic solutions to these tasks. Based on these, our main algorithm is as follows.

Algorithm 2 (Main algorithm) *Let $f(z) = P(z)/Q(z) \in \mathbb{Q}(z)$ be a rational function, with $\deg(P) < \deg(Q)$. Let p be a non-negative integer. To compute the p first terms of the asymptotic expansion of the coefficients of $f(z)$,*

1. *Compute the partial fraction decomposition (4) by Algorithm 1.*
2. *Perform Task 1 to compute, for each (i,j) and for each $\ell, 1 \leqslant \ell \leqslant p$, the number $m_{i,j,\ell}$ of roots of $D_{i,j}$ of modulus $\rho_\ell = \rho_\ell(Q)$.*
3. *Select those terms in the expansion (4) for which $|\alpha| \in \{\rho_1, \dots, \rho_p\}$, and rewrite (4) in the form*
$$[z^n]f(z) = \sum_{\ell=1}^{p} \sum_{\substack{(i,j)\\ m_{i,j,\ell} \neq 0}} \sum_{\substack{D_{i,j}(\alpha)=0\\ |\alpha|=\rho_\ell}} \sum_{k=1}^{i} \binom{n+k-1}{n} \frac{P_{i,j,k}(\alpha)}{\alpha^{n+k}} + o\left(\frac{1}{\rho_p^n}\right). \tag{8}$$
4. *Perform Task 2 to compute, for each (i,j) and for each ℓ, $1 \leqslant \ell \leqslant p$, the number $p_{i,j,\ell} \in \{0,1\}$ (resp. $n_{i,j,\ell}$) of positive (resp. negative) real roots of $D_{i,j}$ of modulus ρ_ℓ.*
5. *Rewrite relation (8) in the following form which is exactly the sought expansion (6):*
$$\sum_{\ell=1}^{p} \Bigg\{ \sum_{\substack{(i,j)\\ m_{i,j,\ell} \neq 0}} \Bigg[p_{i,j,\ell} \sum_{k=1}^{i} \binom{n+k-1}{n} \frac{P_{i,j,k}(\rho_\ell)}{\rho_\ell^k} + (-1)^n n_{i,j,\ell} \sum_{k=1}^{i} \binom{n+k-1}{n} \frac{P_{i,j,k}(-\rho_\ell)}{(-\rho_\ell)^k}$$
$$+ \sum_{\substack{D_{i,j}(\rho_\ell e^{i\theta})=0\\ \sin\theta > 0}} \left(\sum_{k=1}^{i} \binom{n+k-1}{n} \left(P_{i,j,k}(\rho_\ell e^{i\theta}) + P_{i,j,k}(\rho_\ell e^{-i\theta})\right) \frac{\cos[(n+k)\theta]}{\rho_\ell^k} \right) \Bigg] \Bigg\} \frac{1}{\rho_\ell^n} + o\left(\frac{1}{\rho_p^n}\right).$$

2 The algebraic method

To complete our main algorithm, there still remains to exhibit algorithms that perform Tasks 1 and 2. We describe in this section how this can be done purely algebraically. We rely principally on three tools:

(1) a method to order real algebraic numbers due to M. Coste and M.-F. Roy [6], based on Sturm sequences;

(2) a resultant computation that, given two polynomials P and Q produces a polynomial $P \otimes Q$ whose roots are the pairwise products of the roots of P and Q. In particular the smallest non-negative real root of $P \otimes P$ is the square of $\rho_1(P)$ the smallest modulus of the roots of P;

(3) the Graeffe process: $\mathcal{G}_k(P)$ has for roots the kth power of the roots of P;

(4) the construction of a polynomial $\mathcal{P}_k(P)$ whose roots are the products $\alpha_{i_1}\cdots\alpha_{i_k}$ for $i_1 < \cdots < i_k$, the α_i's being the roots of P.

Using (1) and (2) we can compare the smallest moduli $\rho_1(P)$ and $\rho_1(Q)$ of the roots of two polynomials P and Q. This will be done in Section 2.1.1. Using (2) and (3), we can produce the polynomials $\mathcal{P}_k(P)$ the modulus of the smallest root of which is $|\alpha_1|\cdots|\alpha_k|$. This in turn enables us to compare any pair of moduli $\rho_i(P)$ and $\rho_j(Q)$ as will be shown in Section 2.1.2.

Note that other methods than Coste-Roy's algorithm are known to compare real algebraic numbers (see, e.g. [14]). One of the reasons for our choice is that the complexity of Coste-Roy's algorithm is known [15].

The polynomials mentioned above are computed by the formulas:

$$P \otimes Q(y) = \mathrm{Resultant}_z\left(P(z), z^{\deg(Q)}Q(y/z)\right), \quad \mathcal{G}_k(P)(z^k) = \prod_{j=0}^{k-1} P(e^{2ij\pi/k}z),$$

$$\forall k,\ 1 \leqslant k \leqslant n, \qquad [\mathcal{P}_k(P)]^k = \frac{\displaystyle\prod_{i=0}^{\lfloor (k-1)/2\rfloor} \mathcal{P}_{k-(2i+1)}(P) \otimes \mathcal{G}_{2i+1}(P)}{\displaystyle\prod_{i=1}^{\lfloor k/2\rfloor} \mathcal{P}_{k-2i}(P) \otimes \mathcal{G}_{2i}(P)}, \tag{9}$$

where by convention we set $\mathcal{P}_0(P)(z) = z - 1$. Apart from the last one, these polynomials are well known. That the last polynomial has the roots we expect is not difficult to check. All these polynomials have coefficients in the same field as P and Q.

2.1 Sorting the moduli

Given the polynomial Q, its factors $D_{i,j}$ and an integer p, we need to determine the number of roots of each factor belonging to $Z_k(Q)$, $1 \leqslant k \leqslant p$. To simplify our description, we first concentrate on the case $p = 1$, corresponding to the first order estimate of the asymptotic expansion.

2.1.1 First order estimate

In this case ($p = 1$), our task can be performed in polynomial time in the degree of Q by Algorithm 4 below (which is an extension of an algorithm communicated to us by M.-F. Roy, taking into account multiplicities). We first describe an algorithm to compute the number of roots of smallest modulus of a polynomial.

Algorithm 3 (Number of roots of smallest modulus) *Let $P \in \mathbb{Q}[z]$.*

1. *Compute $P_1P_2^2\cdots P_n^n$ the square-free decomposition of $P \otimes P$.*

2. *Using Coste-Roy's algorithm, find i_0 such that P_{i_0} has the smallest non-negative real root.*

3. *Then $|Z_1(P)| = i_0$.*

Proof. By construction, the smallest non-negative real root of $P \otimes P(z)$ is $\rho_1^2(P)$. Moreover, its order of multiplicity is the number of roots of P of smallest modulus. Computing square-free decompositions in Step 1 ensures that only one of the polynomials P_i has the smallest non negative real root. □

Algorithm 4 (Smallest moduli comparison) *Let P and $Q \in \mathbb{Q}[X]$.*

1. *Compute P_{i_0} and Q_{j_0} as in Algorithm 3. Their smallest non-negative real roots are $\rho_1^2(P)$ and $\rho_1^2(Q)$.*

2. *Applying Coste-Roy's algorithm to P_{i_0} and Q_{j_0}, compare $\rho_1^2(P)$ and $\rho_1^2(Q)$.*
3. *The number of roots of P (resp. Q) of modulus $\rho_1(PQ) = \min(\rho_1(P), \rho_1(Q))$ is given by i_0 (resp. j_0) if $\rho_1(PQ)$ is equal to $\rho_1(P)$ (resp. $\rho_1(Q)$), and 0 otherwise.*

Proof. This algorithm works for the same reason as Algorithm 3. □

Applying this algorithm to the polynomials $D_{i,j}$ and Q gives the result we are after. Task 1 is therefore solved for $p = 1$. From the complexity estimates in [15], it follows that the complexity of Algorithm 4 is $O(n^{20}(n + \log|P| + \log|Q|)^2)$, where $n = \max(\deg(P), \deg(Q))$ and $|P|$ denotes the sum of the absolute values of the coefficients of the polynomial P.

2.1.2 Ordering the p smallest moduli

We now want to compute for each k, $1 \leqslant k \leqslant p$ and each (i, j), the number of roots of $D_{i,j}$ of modulus $\rho_k(Q)$. Although all the $|\alpha_i|^2$ are roots of $P \otimes P$, Algorithm 4 does not generalize well because in general $P \otimes P$ has other non-negative real roots. We first give a generalization of Algorithm 3.

Algorithm 5 (Number of roots of a given modulus) *Let $P \in \mathbb{Q}[z]$. Given an integer $q \geqslant 1$, and $m_i = |Z_i(P)|$, for $1 \leqslant i \leqslant q$, such that $m_1 + \cdots + m_q < \deg(P)$, to compute $m_{q+1} = |Z_{q+1}(P)|$, apply Algorithm 3 to the polynomial $\widehat{P} = \mathcal{P}_{m_1+\cdots+m_q+1}(P)$.*

Proof. If $P(z) = \prod_i (z - \alpha_i)$, then by (9) we have $\widehat{P} = \prod_{i_1 < \cdots < i_{k+1}} (z - \alpha_{i_1} \cdots \alpha_{i_{k+1}})$, where $k = m_1 + m_2 + \cdots + m_q$. Since $|\alpha_1| = \cdots = |\alpha_{m_1}| < |\alpha_{m_1+1}| = \cdots = |\alpha_{m_1+m_2}| < \cdots < |\alpha_{k+1}| = \cdots = |\alpha_{k+m_{q+1}}| < \cdots$, the roots of smallest modulus of $\widehat{P}(z)$ are $\alpha_{k+j} \prod_{i=1}^{k} \alpha_i$, $1 \leqslant j \leqslant m_{q+1}$. □

We can now give the generalization of Algorithm 4:

Algorithm 6 ($(q + 1)$st smallest moduli comparison) *Let P and $Q \in \mathbb{Q}[z]$. Given an integer $q \geqslant 1$ and m_i (resp. n_i) the number of roots of P (resp. Q) of modulus $\rho_i(PQ)$ for $1 \leqslant i \leqslant q$, such that $(m_1 + \cdots + m_q) + (n_1 + \cdots + n_q) < \deg(P) + \deg(Q)$, to find the number m_{q+1} (resp. n_{q+1}) of roots of P (resp. Q) of modulus $\rho_{q+1}(PQ)$,*

- *If $m_1 + \cdots + m_q = \deg(P)$ (resp. $n_1 + \cdots + n_q = \deg(Q)$), then m_{q+1} (resp. n_{q+1}) is 0 and n_{q+1} (resp. m_{q+1}) is given by Algorithm 5.*
- *Otherwise, these values are obtained by applying Algorithm 4 to $\widetilde{P} = \mathcal{P}_{m_1+\cdots+m_q+1}(P) \otimes \mathcal{P}_{n_1+\cdots+n_q}(Q)$ and $\widetilde{Q} = \mathcal{P}_{n_1+\cdots+n_q+1}(Q) \otimes \mathcal{P}_{m_1+\cdots+m_q}(P)$.*

Proof. The first part is obvious. Denote by α_i the roots of P and by β_j the roots of Q. Let M_p be the number of roots of P whose modulus is the smallest $\rho_i(P)$ strictly greater than $\rho_q(PQ)$ and define similarly M_q. The second part follows from noticing that the polynomial $\widetilde{Q}$ has been built so that it has M_q roots of smallest modulus, namely $\beta_k \prod_{i=1}^{m_1+\cdots+m_q} \alpha_i \prod_{j=1}^{n_1+\cdots+n_q} \beta_j$, $n_1 + \cdots + n_q + 1 \leqslant k \leqslant n_1 + \cdots + n_q + M_q$. Writing similarly the M_p roots of smallest modulus of $\widetilde{P}$, one deduces the result. □

By induction on $p \geqslant 1$, using Algorithm 6, it is now easy to find for each (i, j) the number of roots of $D_{i,j}$ of modulus $\rho_1(Q), \rho_2(Q), \ldots \rho_p(Q)$ with $Q = \prod_{i,j} D_{i,j}^i$. Task 1 is thus solved.

Because $k = m_1 + \cdots + m_q$ can take any value between 1 and n, and since the degree of $\mathcal{P}_{m_1+\cdots+m_q}$ is $\binom{n}{k}$, Algorithm 6 runs in exponential time as soon as $p \geqslant 2$.

2.2 Finding the real roots

We now attack Task 2: given an integer k, $1 \leqslant k \leqslant p$, we want to find for each $D_{i,j}$ the number and the sign of the real roots of $D_{i,j}$ of modulus $\rho_k(Q)$. The following algorithm solves this problem.

Algorithm 7 (Real roots and their sign) *Let $P \in \mathbb{Q}[z]$ be a square-free polynomial. Given an integer q, the number $m_i = |Z_i(P)|$ and the number $n_i \in \{0,1\}$ of real negative roots of P of modulus $\rho_i(P)$ for $1 \leqslant i \leqslant q$, with $m_1 + \cdots + m_q < \deg(P)$; to compute the number $p_{q+1} \in \{0,1\}$ (resp. n_{q+1}) of real positive (resp. negative) roots of P of modulus $\rho_{q+1}(P)$,*

1. *Compute the polynomial $\widehat{P} = \mathcal{P}_{m_1+\cdots+m_q+1}(P)$ and m_{q+1} by Algorithm 5.*
2. *If m_{q+1} is odd, then P has exactly one real root of modulus ρ_{q+1}. To find its sign, compare the smallest positive real roots r of $\widehat{P}(z)$ and ρ of $\widehat{P}(-z)$ by Coste-Roy's algorithm. If $r > \rho$ or if $\widehat{P}(z)$ has no positive real roots, then $p_{q+1} \equiv n_1 + \cdots + n_q \pmod 2$ and $n_{q+1} \equiv 1 + n_1 + \cdots + n_q \pmod 2$. Otherwise, either $\rho > r$ or $\widehat{P}(-z)$ has no positive real roots, and then $p_{q+1} \equiv 1 + n_1 + \cdots + n_q \pmod 2$ and $n_{q+1} \equiv n_1 + \cdots + n_q \pmod 2$.*
3. *If m_{q+1} is even, compute $R(z^2) = \gcd(\widehat{P}(z), \widehat{P}(-z))$. If its degree is 0 then $p_{q+1} = n_{q+1} = 0$, otherwise use Coste-Roy's algorithm to compare the smallest positive real roots r of R and ρ of $\widehat{P} \otimes \widehat{P}$. If R has no positive real roots, then $p_{q+1} = n_{q+1} = 0$. If $r = \rho$ then $p_{q+1} = n_{q+1} = 1$. Otherwise we must have $r > \rho$ and so $p_{q+1} = n_{q+1} = 0$.*

Proof. First, note that P being square-free, $p_{q+1} \in \{0,1\}$ and $n_{q+1} \in \{0,1\}$. Let $P = \prod_i (z - \alpha_i)$. The roots of P being either real or coming by pairs of conjugates, the number $M = \prod_{i=1}^{m_1+\cdots+m_q} \alpha_i$ is real and its sign is the sign of $(-1)^{n_1+\cdots+n_q}$. The polynomial $\widehat{P}$ has m_{q+1} roots of smallest modulus, namely

$$M \cdot \alpha_i, \qquad m_1 + \cdots + m_q + 1 \leqslant i \leqslant m_1 + \cdots + m_q + m_{q+1}, \tag{10}$$

so that if m_{q+1} is odd, then P has exactly one real root of modulus $\rho_{q+1}(P)$ and $\widehat{P}(z)$ has only one real root of smallest modulus. Step 2 is now obvious.

When m_{q+1} is even, either $p_{q+1} = n_{q+1} = 0$ or $p_{q+1} = n_{q+1} = 1$ for conjugacy reasons. The roots of $R(z^2)$ are the roots α of $\widehat{P}$ such that $-\alpha$ is also a root of $\widehat{P}$. Thus if $\deg(R) = 0$ we cannot have $p_{q+1} = n_{q+1} = 1$ because of (10) . Otherwise, the smallest positive real root of R is the square of the smallest real root β of $\widehat{P}$ such that $-\beta$ is also a root of $\widehat{P}$. The smallest positive real root of $\widehat{P} \otimes \widehat{P}$ being the square of the moduli of the roots (10), Step 3 is now clear. □

For the same reasons as Algorithm 6, Algorithm 7 runs in exponential time as soon as $p \geqslant 2$.

Tasks 1 and 2 have been solved, and we are now able to give the asymptotic expansion of the coefficients of a rational function by Algorithm 2.

3 The numerical method

In the last section, we solved Tasks 1 and 2 using only algebraic tools, which is currently the most natural solution to our problem from the computer algebra point of view. We shall now present an alternative method showing that numerical tools can be used reliably to perform our task in polynomial time. We only need to order the moduli of the roots of a polynomial and find which of them are real. Although there exists algorithms which achieve these tasks (for instance, we could use Graeffe's method to approximate the moduli of the roots of Q), it is cheaper to find directly all the roots of Q with a sufficiently sharp bound on their errors. Our numerical method will depend on a complex root finding algorithm, that we first describe briefly.

3.1 A root finding algorithm

We want to find the complex roots of a polynomial with rational coefficients with arbitrary precision. Numerous algorithms exist to achieve this task, but only few of them are reliable. Newton's method does not always converge; Traub and Jenkins' method [9], usually used for root finding in computer algebra systems, converges theoretically but it turns out that precision control is badly handled in practical implementations. Besides, its complexity is not known to be polynomial. We present here a root finding algorithm for which the precision control has been carefully studied. In the following, $|P| = \sum_i |a_i|$ denotes the norm of the polynomial $P = a_0 + \cdots + a_n z^n$. An immediate consequence of a theorem from [12] is the following result.

Proposition 1 (Pan) *Let $P \in \mathbb{C}[z]$ be a monic polynomial, $n = \deg(P) > 0$. All the zeros of P can be computed with absolute error $\epsilon > 0$ using $O\left[n^2 \log n(n\log(n) + \log(|P|/\epsilon))\right]$ arithmetic operations.*

Unfortunately, the constant term in front of the time bound is very high and therefore the result seems to be only of theoretical importance. For instance, this algorithm relies on FFT techniques, which makes it efficient only for very large degrees.

3.2 Necessary precision

The reason why we can rely on numerical methods to solve our task is that two different roots of a polynomial with integer coefficients cannot be too close. The following result [11] makes this precise.

Proposition 2 (Mahler) *Let $P(z) = a_0 + a_1 z + \cdots + a_n z^n = a_n \prod_{i=1}^{n}(z - \alpha_i)$ be a polynomial of degree $n > 0$ with integer coefficients. Then*

$$\alpha_i \neq \alpha_j \quad \Longrightarrow \quad |\alpha_i - \alpha_j| \geqslant \sqrt{3} n^{-(n+2)/2} M(P)^{1-n},$$

where $M(P) = |a_n| \prod_{i=1}^{n} \max(1, |\alpha_i|)$.

From this we deduce the following theorem.

Theorem 1 *Let $P(z)$ be a polynomial with integer coefficients, $n = \deg(P) > 0$ and $\alpha_1, \ldots, \alpha_n$ its roots. Define $\kappa(P)$ to be the following quantity*

$$\kappa(P) = \frac{\sqrt{3}}{2} [n(n+1)/2]^{-[n(n+1)/4+1]} \cdot M(P)^{-n(n^2+2n-1)/2}, \tag{11}$$

then $|\alpha_i| \neq |\alpha_j| \Longrightarrow \big||\alpha_i| - |\alpha_j|\big| \geqslant \kappa(P)$ and $|\Im(\alpha_i)|$ is either 0 or larger than $\kappa(P)$.

Proof. Let C be the leading coefficient of P. We first prove that the polynomial $Q(z) = C^{n+1} \prod_{i \leqslant j}(z - \alpha_i \alpha_j)$ has integer coefficients. We can suppose that the polynomial P is primitive, i.e. the gcd of its coefficients is 1. The polynomial $P \otimes P$ has for roots $\alpha_i \alpha_j$, $1 \leqslant i, j \leqslant n$, its coefficients are integers, and from classical results on the resultant algorithm, its leading coefficient is C^{2n}. Since the polynomial $\mathcal{G}_2(P)(z) = C^2 \prod_i (z - \alpha_i^2)$ has integer coefficients, is primitive and divides $P \otimes P$, we deduce that the quotient $C^{2(n-1)} \prod_{i \neq j}(z - \alpha_i \alpha_j) = [C^{n-1} \prod_{i<j}(z - \alpha_i \alpha_j)]^2$ has integer coefficients and therefore, so has its square root. Finally $Q(z)$ is the product of this polynomial by $\mathcal{G}_2(P)(z)$ which implies it has integer coefficients.

Next, let x and y be two distinct roots of Q. Since $M(Q) \leqslant M(P)^{n+1}$, Mahler's result applied to Q yields

$$|x - y| \geqslant \gamma = \sqrt{3}\, [n(n+1)/2]^{-[n(n+1)/4+1]}\, M(P)^{(n+1)(1-n(n+1)/2)}. \tag{12}$$

If α_i and α_j are roots of P with distinct moduli, then $|\alpha_i|^2$ and $|\alpha_j|^2$ are two distinct roots of Q and we have $\big||\alpha_i|^2 - |\alpha_j|^2\big| \geqslant \gamma$, hence $\big||\alpha_i| - |\alpha_j|\big| \geqslant \gamma/(|\alpha_i| + |\alpha_j|)$. As $|\alpha_i|$ and $|\alpha_j|$ are smaller than $M(P)$, we finally deduce

$$\big||\alpha_i| - |\alpha_j|\big| \geqslant \frac{\gamma}{2M(P)} = \kappa(P).$$

The last part of the theorem can be derived analogously from the inequality $|\alpha_i \overline{\alpha_i} - \alpha_i^2| \geqslant \gamma$. □

A sharper lower bound on $|\Im(\alpha_i)|$ can be derived by considering only the polynomial P. We do not need this sharper bound since we need to compute the roots with an absolute error $\kappa(P)$ to sort their moduli.

3.3 Numerical algorithm

Using these results, we now give an algorithm which performs reliably Tasks 1 and 2 by purely numerical methods.

Algorithm 8 (Numerical) *Let $Q = \prod_{i,j} D_{i,j}^i$ be a square-free decomposition of the polynomial Q. Our aim is to compute, for each (i,j) and for each q the number of roots of $D_{i,j}$ of modulus $\rho_q(Q)$ and the number of these that are real along with their signs.*

1. *For each (i,j), compute the number $\gamma_{i,j} = \inf_{(k,\ell)\neq(i,j)} \gamma(D_{i,j}D_{k,\ell})$ where $\gamma(D_{i,j}D_{k,\ell})$ is defined by*
$$\gamma(D_{i,j}D_{k,\ell}) = \frac{\sqrt{3}}{2}[d(d+1)/2]^{-[d(d+1)/4+1]} \cdot |Q|^{-d(d^2+2d-1)/2},$$
with $d = \deg(D_{i,j}) + \deg(D_{k,\ell})$. (Take $D_{k,\ell} = 1$ if $D_{i,j}$ is the only polynomial).
2. *Using Pan's Algorithm [12], compute for each (i,j) the roots of the polynomial $D_{i,j}$ with an absolute error $\epsilon_{i,j} = \gamma_{i,j}/4$.*
3. *Let α be a root of $D_{i,j}$, β a root of $D_{k,\ell}$, $\hat{\alpha}$ and $\hat{\beta}$ their approximations found at Step 2. If $\left||\hat{\alpha}| - |\hat{\beta}|\right| < \gamma_{i,j}/2$, then $|\alpha| = |\beta|$, else the inequality between $|\alpha|$ and $|\beta|$ is given by the inequality between $|\hat{\alpha}|$ and $|\hat{\beta}|$. This way, all the moduli of the roots of Q are sorted.*
4. *Let α be a root of some $D_{i,j}$. If its approximation $\hat{\alpha}$ satisfies $|\Im(\hat{\alpha})| > \gamma_{i,j}/2$, then α is not real. Otherwise α is real, and its sign is given by the sign of $\Re(\hat{\alpha})$.*

Proof. The validity of this algorithm results from Theorem 1 applied to each of the polynomials $D_{i,j}D_{k,\ell}$ and $D_{i,j}$, and from the inequalities:

$$(i,j) \neq (k,\ell) \quad \Longrightarrow \quad M(D_{i,j}D_{k,\ell}) \leqslant M(Q) \leqslant |Q|,$$
$$\forall(i,j), \qquad M(D_{i,j}) \leqslant M(Q) \leqslant |Q|.$$

The inequality $M(Q) \leqslant |Q|$ is due to Mahler [10]. In Step 4, the fact that the sign of α (when α is real) is the sign of $\Re(\hat{\alpha})$ results from the inequality $|\alpha| \geqslant \gamma_{i,j}$. (This latter inequality can be derived, for example, from the inequality $|\alpha| \geqslant 1/M(D_{i,j})$ which is easily proved). □

Proposition 1 *Algorithm 8 runs in time $O\left[n^5 \log n \log |Q|\right]$ where n is the degree of Q.*

Proof. Apply Theorem 1 to each of the polynomials $D_{i,j}$ with $\epsilon = \epsilon_{i,j}$. □

4 Optimizations

In the last two sections, we presented two methods that achieve Tasks 1 and 2. In practice, these two methods are awfully expensive. We present here another algorithm, which works on most of the rational functions, and which is much quicker. Another advantage of this new algorithm is that we can know whether it works or not. When it does not, then we can revert to one of the previous methods. This method is essentially numerical. We compute approximations of the roots of $Q(z)$ using a root finding algorithm, with a relatively crude absolute error (compared to what it was in the previous section). In most cases though, everything can be deduced from these estimates.

In [17], A. Schönhage gave a root finding algorithm and demonstrated the following result.

Theorem 2 (Schönhage) *Let $P \in \mathbb{C}[z]$ be a monic polynomial, $n = \deg(P)$, and $\epsilon > 0$. We can compute n complex numbers $v_1, \cdots, v_n$ such that*

$$|P - (z - v_1)\cdots(z - v_n)| < \epsilon \tag{13}$$

within the time bound of $O\left[(n^3\log(n) + \log(|P|/\epsilon)n^2)\log(n\log(|P|/\epsilon))\log\log(n\log(|P|/\epsilon))\right]$.

Although this bound seems slightly weaker than the previous one in Proposition 1, this one is in terms of bit complexity. Note that this algorithm does not approach directly the roots with an absolute precision ϵ. But from inequality (13) one can derive absolute error bounds on the roots of P. This algorithm was optimized by Gourdon [8] who implemented it in MAPLE; the program gives the right result in a reasonable time. We shall rely on this method to approximate roots of polynomials.

Let $Q(z) = \prod_{i,j} D_{i,j}^i$ be a square-free decomposition of the polynomial Q. Using Schönhage's algorithm, we compute for each (i,j) approximations $\hat{\alpha}_1, \ldots, \hat{\alpha}_p$ of the roots of $D_{i,j}$ such that $|D_{i,j}(z) - \prod_k (z - v_k)| < \epsilon$ (we can assume $D_{i,j}$ monic), with $\epsilon = 10^{-n}$, where $n = \deg(Q)$. We have already seen that from this we can compute for each root α_k of $D_{i,j}$ an absolute error bound $\tau_k > 0$ such that $|\hat{\alpha}_k - \alpha_k| < \tau_k$. Suppose that the absolute bounds τ_k determine which roots are conjugates, which roots are real and what their sign is. To achieve Tasks 1 and 2, it then remains to compare the moduli of the non-conjugate roots. If again, the absolute error bounds τ_k make it possible to decide these comparisons, then we have finished. Otherwise, we have a certain number of couples of non-conjugates and distinct roots (α, β) of Q such that, if $\hat{\alpha}$ and $\hat{\beta}$ are the approximations of α and β found and τ and τ' the absolute error bounds found for these approximations, $\big||\hat{\alpha}| - |\hat{\beta}|\big| < \tau + \tau'$. We call these couples candidates. In this case, we use Algorithm 9 (see below) to test the equality of the moduli of the candidates. If all the candidates have the same modulus (this is often the case), then we have solved Tasks 1 and 2. Else, this algorithm failed and we use one of the previous methods discussed in Sections 2 and 3. The underlying idea is that it is very unlikely that two non-real roots of distinct moduli have the same argument.

Algorithm 9 (Equality of candidates) *Let $P = \prod_{i=1}^n (z - \alpha_i)$ be a square-free polynomial $\in \mathbb{Q}[z]$. We are given approximations $\hat{\alpha}_i$, absolute error bounds τ_i such that $|\hat{\alpha}_i - \alpha_i| < \tau_i$, and for each j, $1 \leqslant j \leqslant n$, the number s_j of elements of the set $\Gamma_j = \{i, \big||\hat{\alpha}_i| - |\hat{\alpha}_j|\big| < \tau_i + \tau_j\}$.*

1. *Compute the square-free decomposition $P \otimes P = P_1 P_2^2 \cdots P_r^r$.*
2. *By Sturm sequences [19], compute for each k the number m_k of non-negative real roots of P_k.*
3. *If (a) $m_1 + 2m_2 + \cdots + rm_r = n$, (b) for all (i,j), either $\Gamma_i \cap \Gamma_j = \varnothing$ or $\Gamma_i = \Gamma_j$, (c) for all ℓ, $\ell m_\ell = |\cup_{s_i = \ell} \Gamma_i|$, then for all j, all the elements of Γ_j have the same modulus.*

Proof. Since $P \otimes P = \prod_{i,j}(z - \alpha_i \alpha_j)$, the $|\alpha_i|^2$ are roots of it. If $m_1 + 2m_2 + \cdots + rm_r = n$, then these are its only positive roots. The result is now obvious. □

5 Examples

Denumerants [4, p. 108]: the number of ways to make n francs with coins of 1, 2, 5, and 10 francs has for generating function

$$f(z) = \frac{1}{(1-z)(1-z^2)(1-z^5)(1-z^{10})}.$$

The ten singularities have the same modulus, but 1 being a singularity of order 4 is isolated in the decomposition (4) produced by Algorithm 1:

$$\frac{1/100}{(1-z)^4} + \frac{7/100}{(1-z)^3} + \frac{91/400}{(1-z)^2} + \frac{21/50}{1-z} + \sum_{\beta^4-\beta^3+\beta^2-\beta+1=0} \frac{4\beta^3 - 3\beta^2 + 2\beta - 6}{100(\beta - z)}$$
$$+ \sum_{\alpha^5+2\alpha^4+2\alpha^3+2\alpha^2+2\alpha+1=0} \frac{17\alpha^4 + 9\alpha^3 + 17\alpha^2 + \alpha + 1}{2000(z-\alpha)^2} + \frac{\alpha^4 - 27\alpha^3 - 14\alpha^2 - 33\alpha + 3}{500(z-\alpha)}.$$

From this we deduce easily the first terms of the asymptotic expansion of $[z^n]f(z)$:

$$\frac{n^3}{600} + \frac{9n^2}{200} + \left(\frac{421}{1200} + \frac{(-1)^n}{80} - \sum_{\alpha \in Z_1^+(z^4+z^3+z^2+z+1)} \frac{\alpha^3 + \overline{\alpha}^3 + 2(\alpha + \overline{\alpha}) + 4}{250} \cos[(n+2)\arg(\alpha)] \right) n + O(1).$$

Sum of powers of Fibonacci numbers Since rational functions (when they are regular at infinity) are closed under Hadamard product, and the sum of a sequence is obtained by multiplying its generating function by $1/(1-z)$, many operations that can be applied to a linear recurrent sequence yield another linear recurrent sequence. We consider here $\sum_{k=1}^{n} F_k^p$. It is not difficult (tedious, rather) to show that the generating function of F_n^p has the following expression for fixed p:

$$5^{\frac{1-p}{2}} \sum_{k=0}^{\frac{p-1}{2}} \frac{\binom{p}{k} F_{p-2k} z}{1-(-1)^k L_{p-2k} z - z^2}, \quad \text{if } p \text{ is odd;}$$
$$5^{-p/2} \left[\sum_{k=0}^{\frac{p}{2}-1} \binom{p}{k} (-1)^k \frac{2-(-1)^k L_{p-2k} z}{1-(-1)^k L_{p-2k} z + z^2} + \frac{(-1)^{p/2}\binom{p}{p/2}}{1-(-1)^{p/2} z} \right], \quad \text{if } p \text{ is even;}$$

where L_n denote the Lucas numbers. From this we construct the generating function of the sum of the tenth powers of the Fibonacci numbers, which we give in compact form to our algorithm:

$$\frac{z^9 - 87\,z^8 - 4047\,z^7 + 42186\,z^6 + 205690\,z^5 + 42186\,z^4 - 4047\,z^3 - 87\,z^2 + z}{z^{11} - 89z^{10} - 4895z^9 + 83215z^8 + 582505z^7 - 1514513z^6 - 1514513z^5 + 582505z^4 + 83215z^3 - 4895z^2 - 89z + 1}.$$

The first stage of the algorithm produces the decomposition $f(z) = \sum_{Q(\alpha)=0} P(\alpha)/(\alpha - z)$, where Q is the denominator of f and P is the following polynomial:

$$\begin{aligned} P(z) = \quad & -\frac{1088331670771}{46065550781250000} + \frac{147618897967}{128854687500000}\,z + \frac{25161090223051}{98535937500000}\,z^2 + \frac{473498073791}{4692187500000}\,z^3 \\ & -\frac{48654728411}{541406250000}\,z^4 + \frac{178616503}{8789062500}\,z^5 - \frac{2060862361}{541406250000}\,z^6 - \frac{4685959559}{4692187500000}\,z^7 \\ & +\frac{383607377}{7579687500000}\,z^8 + \frac{1645213621}{1675110937500000}\,z^9 - \frac{496515521}{46065550781250000}\,z^{10}. \end{aligned}$$

This decomposition implies that all the singularities are simple poles. The next stage of the algorithm is to determine the number of real and complex roots of each modulus for the first moduli of the roots. This is done by a numerical evaluation of the roots with error bound 10^{-4} which shows that all the roots are real, and yields their signs. For instance, the three first terms of the expansion are

$$[z^n]f(z) = \frac{P(\rho_1)}{\rho_1^{n+1}} + (-1)^{n+1}\frac{P(-\rho_2)}{\rho_2^{n+1}} + \frac{P(\rho_3)}{\rho_3^{n+1}} + o\left(\frac{1}{\rho_3^n}\right),$$

with $\rho_1 \simeq 0.00812$, $\rho_2 \simeq 0.0212$ and $\rho_3 \simeq 0.0753$.

A large problem This combinatorial problem was considered in [5]. Starting with 1, we write down a sequence of words by counting the number of contiguous identical digits in the previous word. Thus the second word is 11 because there is one 1 in "1". Then we have two 1s, hence the third word is 21, and so on. The first few words are: 1, 11, 21, 1211, 111221, 312211, 13112221,... We then consider the sequence of lengths of these words: 1, 2, 2, 4, 6, 6, 6, 8,.... What happens is that this sequence is rational of degree 72! From the table in [5, pp. 177–178], it is possible to compute this fraction by solving a linear system. The numerator is found to be

$$\begin{aligned} P(z) = {} & 1 + z - z^2 - z^3 - z^4 + z^5 - 5z^7 + 6z^9 + 8z^{10} - 10z^{12} - 5z^{13} + z^{14} + 4z^{15} + z^{16} + 4z^{17} + 7z^{18} \\ & +9z^{19} - 4z^{20} - 22z^{21} - 39z^{22} + 4z^{23} + 52z^{24} + 38z^{25} + 17z^{26} - 68z^{27} - 28z^{28} + 22z^{29} - 12z^{30} \\ & +13z^{31} - 37z^{32} + 45z^{33} + 54z^{34} - 12z^{35} - 34z^{36} - 82z^{37} + 17z^{38} + 89z^{39} + 13z^{40} - 34z^{42} - 89z^{43} \\ & +73z^{44} + z^{45} + 26z^{46} + 31z^{47} - 128z^{48} - 14z^{49} + 49z^{50} + 56z^{51} + 74z^{52} - 99z^{53} - 20z^{54} - 43z^{55} \\ & +33z^{56} + 47z^{57} - 41z^{58} + 18z^{59} + 50z^{60} - 10z^{61} - 13z^{62} - 9z^{63} - 17z^{64} + 38z^{65} - 42z^{66} + 37z^{67} \\ & +8z^{68} - 4z^{69} - 29z^{70} - 19z^{71} + 28z^{72} + 30z^{73} - 22z^{74} - 18z^{76} + 12z^{77}, \end{aligned}$$

and the denominator is

$$\begin{aligned}Q(z) = 1 - z - z^2 - z^3 + z^4 + 3z^5 - z^7 - 2z^8 + 3z^{13} + 3z^{14} - 2z^{15} - 5z^{16} - 8z^{17} + 7z^{18} + z^{19} + 8z^{20} \\ - 5z^{22} + 8z^{23} - 12z^{24} - 4z^{25} - z^{26} + 18z^{28} - 4z^{29} + 2z^{30} - 13z^{31} - 7z^{32} + 19z^{33} - 14z^{34} + 14z^{35} \\ - 6z^{36} - 4z^{37} + 13z^{38} - 9z^{39} - 7z^{40} + 4z^{41} - 8z^{42} + 7z^{43} + 5z^{44} + 7z^{45} - 12z^{46} + 17z^{47} - 22z^{48} \\ + 8z^{49} - 7z^{50} + 16z^{51} - 6z^{52} - 7z^{53} - 6z^{54} + 3z^{55} + 19z^{56} - 5z^{57} - 5z^{58} - 14z^{59} + 8z^{60} + 2z^{61} \\ + 7z^{62} - 5z^{63} + z^{64} - 8z^{65} + 14z^{66} - 11z^{67} + 16z^{68} - 18z^{69} + 9z^{70} - 9z^{71} + 6z^{72}.\end{aligned}$$

One of the nice theorems in [5] states that this denominator is actually independent of the starting string, provided it different from "22". Thus in the leading term of the asymptotic expansion, only the constant factor depends on the initial string.

Despite the large degree of this denominator, it turns out that the asymptotic expansion is not too difficult to find. For the sequence we consider, the decomposition of P/Q is

$$\frac{P(z)}{Q(z)} = R(z) + \sum_{Q(\alpha)=0} \frac{F(\alpha)}{z-\alpha},$$

where R is a polynomial induced by the first terms, and F is a polynomial of degree 71 with 250-digit rational coefficients. This means that all the singularities are simple poles. If one is only interested in the first order estimate, it then remains to determine the number of roots of smallest modulus. As expected since the coefficients of the generating function are positive, one of these roots is a positive real number. Using the program of X. Gourdon based on A. Schönhage's algorithm [8], we get that the two smallest moduli are approximately 0.767 and 0.861, with error bounds of the order 10^{-40}, which shows that the root of smallest modulus is alone (and therefore real). Thus, $[z^n]f(z) \sim F(\rho_1)\rho_1^{-n-1}$, $\rho_1 \simeq 0.767119$ and $F(\rho_1) \simeq 1.566$. All the 72 moduli belong to the interval (0.767,1.151), showing the need for caution with numerical estimates.

Conclusion

Algorithm 8 should not be implemented blindly. Although its complexity is polynomial, the constant implied in the $O()$ of Proposition 1 is very large. Thus in our last example above, the precision needed to compute the roots would be approximately 522000 digits. Instead, one should use this algorithm as an upper bound in an adaptative program based on a good numerical program such as [8] and Algorithm 9, increasing the precision if necessary.

Note also that we have never used the fact that in combinatorial contexts, the generating functions have only positive coefficients and thus by Pringsheim's theorem (see [20]), one of their singularities of smallest modulus is real positive, the other ones having arguments commensurable with π. The computation of the first-order estimate could take advantage of this extra information.

This very simple problem of linear recurrences with rational coefficients is not yet completely solved. It would be useful in practice to have some control over the periodicities that may occur in the asymptotic expansions. This problem is exemplified with the following generating function:

$$\frac{z^2+2z-2}{(1-2z^2)(1-z)^2},$$

or equivalently $u_n = 2u_{n-1} + u_{n-2} - 4u_{n-3} + 2u_{n-4}$, $u_0 = u_1 = 2, u_2 = 5, u_3 = 4$. The first few terms are 2, 2, 5, 4, 9, 6, 15, 8, 25, 10, ... The first-order asymptotic approximation obtained from this generating function is $(1 + \cos n\pi)2^{n-1} + o(2^n)$. What happens is that although valid for all positive n, this expression reduces to $o(2^n)$ when n is odd. Better precision necessitates to look for further terms in the expansion. The ideal algorithm outputs a list of asymptotic expansions depending on arithmetic properties of n. Cancellation in this context is not a trivial problem. For instance, no algorithm is known to determine whether a linear recurrent sequence takes the value 0 for some index. It is known that when such a sequence cancels infinitely often, the indices where it cancels asymptotically form a finite union of arithmetic progressions that can be computed [1], but our problem is different since we are only concerned with indefinite cancellation of the dominant part, which does not satisfy a linear recurrence in general.

Acknowledgement

This work was supported in part by the ESPRIT III Basic Research Action Programme of the E.C. under contract ALCOM II (#7141).

References

[1] J. Berstel and M. Mignotte, Deux propriétés décidables des suites récurrentes linéaires, Bulletin de la Société Mathématique de France 104 (1976) 175–184.

[2] M. Bronstein and B. Salvy, Full partial fraction decomposition of rational functions, in: M. Bronstein, ed., ISSAC'93 (ACM Press, 1993) 157–160. Proceedings ISSAC'93, Kiev, Ukraine, July 1993.

[3] L. Cerlienco, M. Mignotte and F. Piras, Suites récurrentes linéaires. Propriétés algébriques et arithmétiques, L'Enseignement Mathématique XXXIII Fascicule 1-2 (1987) 67–108.

[4] L. Comtet, Advanced Combinatorics, (Reidel, Dordrecht, 1974).

[5] J.H. Conway, The weird and wonderful chemistry of audioactive decay, in: T.M. Cover and B. Gopinath, eds., Open Problems in Communication and Computation (Springer-Verlag, 1987) 173–188.

[6] M. Coste and M.-F. Roy, Thom's lemma, the coding of real algebraic numbers and the topology of semi-algebraic sets, Journal of Symbolic Computation 5 (1988) 121–129.

[7] J. Dieudonné, Calcul Infinitésimal, (Hermann, Paris, 1968).

[8] X. Gourdon, Algorithmique du théorème fondamental de l'algèbre, Tech. rep. 1852, Institut National de Recherche en Informatique et en Automatique, (February 1993).

[9] M.A. Jenkins and J.F. Traub, A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration, Numerische Mathematik 14 (1970) 252–263.

[10] K. Mahler, An application of Jensen's formula to polynomials, Mathematika 7 (1960) 98–100.

[11] K. Mahler, An inequality for the discriminant of a polynomial, Michigan Mathematical Journal 11 (1964) 257–262.

[12] V. Pan, Algebraic complexity of computing polynomial zeros, Computers and Mathematics with Applications 14 (1987) 285–304.

[13] S. Plouffe, Approximations de séries génératrices et quelques conjectures, Master's thesis, Université du Québec à Montréal, Sept. 1992. Also available as Research Report 92-61, Laboratoire Bordelais de Recherche en Informatique, Bordeaux, France.

[14] R. Rioboo, Real algebraic closure of an ordered field. Implementation in Axiom, in: P.S. Wang, ed., Symbolic and Algebraic Computation (ACM Press, 1992) 130–137. Proceedings of ISSAC'92, Berkeley, July 1992.

[15] M.F. Roy and A. Szpirglas, Complexity of the computation with real algebraic numbers, Journal of Symbolic Computation 10 (1990) 39–51..

[16] H. Schmidt, Beiträge zu einer Theorie der allgemeinen asymptotischen Darstellungen, Mathematische Annalen 113 (1936) 629–656.

[17] A. Schönhage, The fundamental theorem of algebra in terms of computational complexity, Tech. rep., Mathematisches Institut der Universität Tübingen, 1982. Preliminary report.

[18] N.J.A. Sloane, A Handbook of Integer Sequences, (Academic Press, 1973).

[19] C. Sturm, Mémoire sur la résolution des équations numériques, Institut de France de Sciences Mathématiques et Physiques 6 (1835) 271–318.

[20] E.C. Titchmarsh, The Theory of Functions, second ed., (Oxford University Press, 1939).