Fast evaluation of holonomic functions

Joris van der Hoeven

November 17, 1997

Abstract

A holonomic function is an analytic function, which satisfies a linear differential equation with polynomial coefficients. In particular, the elementary functions $\exp, \log, \sin$, etc. and many special functions like $\mathrm{erf}, \mathrm{Si}$, Bessel functions, etc. are holonomic functions.

Given a holonomic function f (determined by the linear differential equation it satisfies and initial conditions in a non singular point z), we show how to perform arbitrary precision evaluations of f at a non singular point z' on the Riemann surface of f, while estimating the error.

Moreover, if the coefficients of the polynomials in the equation for f are algebraic numbers, then our algorithm is asymptotically very fast: if $M(n)$ is the time needed to multiply two n digit numbers, then we need a time $O(M(n \log^2 n \log\log n))$ to compute n digits of $f(z')$.

1 Introduction

Let $\mathbb{K}$ be a subfield of $\mathbb{C}$. A *holonomic function* (over $\mathbb{K}$) is an analytic function f, which satisfies a linear differential equation

$$P_p(z) f^{(p)}(z) + \cdots + P_0(z) f(z) = 0, \tag{1}$$

where $P_0, \cdots, P_p$ are polynomials in $\mathbb{K}[z]$ with $P_p \neq 0$. The elementary functions $\exp, \log, \sin, \cdots$ and many special functions like $\mathrm{erf}, \mathrm{Si}, \cdots$, Bessel functions, hypergeometric functions, etc. are holonomic. The class of holonomic functions also admits several interesting algebraic properties which we recall in section 2.1, and has recently been the object of intensive study in computer algebra and mathematics (e.g. [12, 8, 16]).

The objective of this paper is to study holonomic functions from the *exact numerical* point of view: we require that all complex z numbers we compute with are *effective*, i.e. for any rational $\varepsilon > 0$ we can compute a "Gaussian rational" $\tilde{z} \in \mathbb{Q}[i]$ with $|\tilde{z} - z| \leqslant \varepsilon$. In this context, we are interested in algorithms to evaluate holonomic functions. Of course, some care is needed here, since f is actually defined on a Riemann surface $\mathcal{R}$. Given effective initial conditions

$$F(z) = \begin{pmatrix} f(z) \\ \vdots \\ f^{(p-1)}(z) \end{pmatrix}. \tag{2}$$

for f in a point z, and a suitably discretized path $z \rightsquigarrow z'$ on $\mathcal{R}$, we therefore want to compute f at z' by following the path. In cases where no confusion is possible, we will nevertheless implicitly identify points on $\mathcal{R}$ with their projections on $\mathbb{C}$.

The following three issues we be discussed in this paper:

Q1. How to guarantee the exactness of evaluation algorithms?

Q2. What is the asymptotic complexity of computing n digits of $f(z')$?

Q3. How does the choice of the path $z \rightsquigarrow z'$ influence the complexity of effective analytic continuation? In particular, what happens if the path approaches a singularity?

The remainder of the introduction is devoted to a brief discussion of these questions. We notice that much of the material presented here also appeared in [15], but we think that the presentation in the present paper is more elegant. The section 4.1 and algorithm **B** from section 2.2 are new.

1.1 Effective bounds

Since all our analytic continuation algorithms will be based on power series evaluations, question **Q1** reduces to the problem of computing bounds of the form

$$\forall k \quad |f_k| \leqslant A B^k \tag{3}$$

for the coefficients f_k of the Taylor series

$$f(z+u) = f_0 + f_1 u + f_2 u^2 + \cdots \tag{4}$$

in a non singular point z, where $f, \cdots, f^{(p-1)}$ are known. Now it is a well known fact that the zeros $\omega_1, \cdots, \omega_\nu$ of P_p are the only possible singularities of f. Therefore, denoting by $\delta(z)$ the distance between z and the set of these zeros, we may even require $B > \delta(z)^{-1}$ to be given and ask for an A such that (3) holds.

Another, equivalent problem would be to compute an upper bound C for $|f|$ on a sufficiently small compact disk $D(z, r)$ with center z, say of radius $r < \delta(z)$. Indeed, this yields the estimation

$$|f_k| = \left| \frac{1}{2\pi i} \int_{|u|=r} \frac{f(z+u)}{u^{k+1}} du \right| \leqslant \frac{C}{r^k} \tag{5}$$

for the coefficients f_k. On the other hand, given bounds (3), we have

$$|f| \leqslant \frac{A}{1 - rB}$$

on any compact disk with center z and radius $0 < r < \frac{1}{B}$. In section 2, we prove

Theorem 1. *There exists an algorithm which given z, $0 < r < \delta(z)$ and $F(z)$ computes an upper bound C for $|f|$ on $D(z, r)$.*

1.2 Fast multiple precision evaluations

For certain purposes it is interesting to evaluate holonomic functions up to many digits. First, this question is of theoretical interest, since many special functions are holonomic. Secondly, fast evaluation algorithms up to several hundreds of digits can be used in computer algebra systems in reliable heuristic zero tests for constant expressions involving special and/or holonomic functions. Finally, evaluations up to thousands or millions of digits can be used in order to obtain statistical information about real numbers, which finds its application in analytic number theory research.

In what follows, $M(n)$ denotes the time complexity to multiply two n digit numbers and we make the standard assumption that $M(n)/n$ is monotonic for $n \to \infty$. Asymptotically, $M(n) = O(n \log n \log \log n)$, when using FFT-multiplication [1, 7], but for intermediate precisions, Karatsuba's $O(n^{\log 3/\log 2})$ algorithm [7] is faster. When we measure the complexity of the evaluation of a function in a point, we will only count the time spent on the *real* evaluation. In our case of analytic continuation, this means that we do not count the time needed to compute $O(n)$ digits of $z, z', f(z), \cdots, f^{(p-1)}(z)$, if we need n digits of $f(z')$.

For the evaluation of elementary functions, several fast algorithms are known, such as binary splitting [3], which has time complexity $O(M(n \log^2 n))$ and the AGM method [4, 11, 2], of complexity $O(M(n \log n))$. Although the AGM algorithm is asymptotically faster, binary splitting is more efficient for precisions inferior to $\pm 1,000,000$ digits. Moreover, the binary splitting method has the advantage that it can be generalized to the evaluation of holonomic functions, if $\mathbb{K}$ is an algebraic number field (usually, $\mathbb{K} = \mathbb{Q}$ or $\mathbb{K} = \mathbb{Q}[i]$). In section 3 we first consider the case when z, z' are also in $\mathbb{K}$, and we prove

Theorem 2. *Assume that $\mathbb{K}$ is an algebraic number field and that $z \rightsquigarrow z' = z \to z'$ is the straight line path between $z, z' \in \mathbb{K}$ with $|z' - z| < \delta(z)$. Then n digits of $f(z')$ can be computed in time $O(M(n \log^2 n))$.*

We notice that Haible and Papanikolaou independently proved this theorem in the case of hypergeometric (and slightly more general) functions [6]. Moreover, they implemented the method and established a new world record in the calculation of Apéry's constant $\zeta(3)$ by computing 1,000,000 decimal digits. Hence, binary splitting indeed becomes efficient for large precisions. Moreover, the method can be easily parallelized, a fact which has also been exploited by Haible and Papanikolaou.

1.3 About the choice of the path $z \rightsquigarrow z'$

In order to treat the case when z and z' are arbitrary, it is important to study the dependency of the complexity of the algorithm from theorem 2 on $z, z' \in \mathbb{K}$. Let us first introduce some more notations. We denote by $\mathtt{size}(O)$ the size of an object O. For instance, the size of a natural number is its number of digits. For a fixed open domain U, we also denote by $\rho(z)$ the distance between a point $z \subseteq U$ and the boundary of U. In section 4.1 we prove

Theorem 3. *Assume that*

(a) U *is an open domain on which* $|f|$ *is bounded.*
(b) $\mathbb{K}$ *is an algebraic number field.*
(c) $z \rightsquigarrow z' = z \to z'$ *is the straight line path between two points* $z, z' \in \mathbb{K}$.
(d) We have $D(z, |z' - z|) \subseteq U$.

Denote $s = \texttt{size}(z) + \texttt{size}(z')$ *and* $\tau = \frac{\rho(z)}{|z'-z|}$. *Then* $f(z')$ *can be evaluated up to precision* 2^{-n} *in time*

$$O(M(n(s + \log n) \log n \log^{-1} \tau)),$$

uniformly in z *and* z', *provided that* $\log \log \tau = O(n)$.

In particular, we observe that analytic continuation from z to z' is the faster as the sizes of z and z' are smaller. Some other interesting corollaries of the above result concern optimal choices of paths to approach or turn around singularities; see section 4.1 for a further discussion. In section 4.3, we return to the case when the path $z \rightsquigarrow z'$ is arbitrary. Approximating $z \rightsquigarrow z'$ by a suitable broken line path (the endpoints are approached by a sequence of algebraic numbers, where we double the precision at each step), we prove

Theorem 4. *Assume that the coefficients of* $P_0, \cdots, P_p$ *are in* $\mathbb{K}[z]$. *Then* n *digits of* $f(z')$ *can be computed in time* $O(M(n \log^2 n \log \log n))$.

2 Bounds for holonomic functions

In this section we prove theorem 1. We first recall some classical facts about holonomic functions. We next give an algorithm to compute bounds for the coefficient of the Taylor expansion (4) of f, which yields theorem 1 by what has been said in section 1.1.

2.1 Preliminaries

In this section we recall some interesting properties of holonomic functions [8].

Proposition 1. *The set of holonomic functions over* $\mathbb{K}$ *form a* $\mathbb{K}$*-algebra stable under differentiation.*

The proof of this proposition is actually constructive and uses elimination techniques. We will only prove the stability of the set of holonomic functions over $\mathbb{K}$ under derivation, which is the only fact we need in what follows. So assume that f satisfies (1). Differentiating (1), we obtain

$$\begin{aligned} P_p(z) f^{(p+1)}(z) \; &+ \; (P_{p-1}(z) + P_p'(z)) f^{(p)}(z) \; + \; \cdots \; + \\ &+ \; (P_0(z) + P_1'(z)) f'(z) + P_0'(z) f(z) = 0 \end{aligned}$$

If $P_0' = 0$, then this yields the required equation for f'. Otherwise, we multiply the equation by $P_0(z)$ and subtract $P_0'(z)$ times the equation (1) in order to obtain the required equation for f'.

Proposition 2. *A function f is holonomic over $\mathbb{K}$ if and only if the coefficients of its Taylor series expansion*

$$f(z+u) = f_0 + f_1 u + f_2 u^2 + \cdots$$

in any non singular point $z \in \mathbb{K}$ satisfy a linear difference equation

$$Q_q(k) f_{k+q} + \cdots + Q_0(k) f_k = 0, \tag{6}$$

with $Q_0, \cdots, Q_{q-1} \in \mathbb{K}[k]$.

For what follows we will only need to show how to obtain a linear difference equation (6) from (1). Denoting the k-th coefficient of a power series g in u by $[u^k]$, we have the following rewriting rules

$$\begin{cases} [u^k] u g(u) = [u^{k-1}] g(u) & \text{for } k > 0; \\ [u^k] g'(u) = (k+1)[u^{k+1}] g(u) & \end{cases}$$

Clearly, substitution of z by $z+u$ in (1) yields a linear differential equation for the power series $f_0 + f_1 u + f_2 u^2 + \cdots$ with coefficients in $\mathbb{K}[u]$. Applying the above rewriting rules to this equation yields a linear difference relation with coefficients in $\mathbb{K}[k]$ between $f_\kappa, f_{\kappa+1}, f_{\kappa+2}, \cdots$, where $\kappa = \max(\deg P_0, \cdots, \deg P_p)$. Replacing k by $k+\kappa$ in this relation yields the desired recurrence relation (6). Notice that $Q_q(k) \neq 0$ for all k and that $Q_i(k)/Q_q(k) = O(1)$ for all i and $k \to \infty$.

2.2 Computation of bounds on compact disks

In this section, we shall use the usual Euclidean norm for vectors V with q entries:

$$\|V\| = \sqrt{V_1^2 + \cdots + V_q^2}$$

and the usual operator norm for matrices M:

$$\|M\| = \sup_{\|V\|=1} \|MV\|.$$

We recall that $\|MN\| \leqslant \|M\| \, \|N\|$ for all matrices M, N.

For each k, let us denote

$$\Phi_k = \begin{pmatrix} f_k \\ \vdots \\ f_{k+q-1} \end{pmatrix}.$$

Then the linear difference equation (6) yields a matrix identity

$$\Phi_{k+1} = N_k \Phi_k$$

with

$$N_k = \begin{pmatrix} 0 & 1 & & \mathrm{O} \\ \vdots & & \ddots & \\ 0 & \mathrm{O} & & 1 \\ -\frac{Q_0(k)}{Q_q(k)} & -\frac{Q_1(k)}{Q_q(k)} & \cdots & -\frac{Q_{q-1}(k)}{Q_q(k)} \end{pmatrix}. \tag{7}$$

In order to estimate $|f_k|$ it therefore suffices to estimate the entries of the product

$$\Phi_k = N_{k-1} \cdots N_0 \Phi_0.$$

Let N_∞ be the matrix which is obtained formally by replacing k by infinity in (7). Then we have

$$N_k = N_\infty + o(1),$$

i.e. the difference $N_k - N_\infty$ is an error matrix whose coefficients tend to zero for $k \to \infty$. Let $\varepsilon > 0$. We claim that there exists a computable diagonal matrix D and an invertible matrix U, such that

$$\|D - U N_\infty U^{-1}\| \leqslant \tfrac{\varepsilon}{2}.$$

Indeed, we first triangulate N_∞ and obtain an expression of the form $T = \hat{U} N_\infty \hat{U}^{-1}$ for some invertible matrix U. Now let μ be the maximum of the absolute values of the non diagonal coefficients of T and set $\lambda = \min(\frac{\varepsilon}{2\mu\sqrt{q}}, 1)$.

Then we take $U = \mathrm{Diag}\ (1, \lambda, \cdots, \lambda^{q-1})\hat{U}$, where $\mathrm{Diag}\ (1, \lambda, \cdots, \lambda^{q-1})$ denotes the diagonal matrix with entries $1, \lambda, \cdots, \lambda^{q-1}$, and we let D be the diagonal part of $UN_\infty U^{-1}$.

Now for each k, we let

$$\begin{cases} D_k = UN_kU^{-1}; \\ E_k = D_k - D, \end{cases}$$

Since D_k tends to $D_\infty = UN_\infty U^{-1}$ for $k \to \infty$, there exists a constant k_0 (which is easily computed, since the entries of E_k are rational fractions in k), such that

$$\|E_k\| \leqslant \varepsilon, \tag{8}$$

for all $k \geqslant k_0$. Putting $P = D_{k_0-1} \cdots D_0$, we finally obtain the bound

$$\begin{aligned} \|N_{k-1} \cdots N_0 \Phi_0\| &= \|U^{-1} D_{k-1} \cdots D_0 U \Phi_0\| \\ &\leqslant \|U^{-1}\|\ (\|D\| + \varepsilon)^{k-k_0} \|P\|\ \|U\|\ \|\Phi_0\|, \end{aligned}$$

for all $k \geqslant k_0$. In particular,

$$|f_k| \leqslant \sqrt{q} \|U^{-1}\|\ \|P\|\ \|U\|\ \|\Phi_0\|\ (\|D\| + \varepsilon)^{k-k_0}$$

for all $k \geqslant k_0$.

Let $P_p(z+u) = c_0 + c_1 u + \cdots + c_p u^p$ as a polynomial in u. Examining the construction of the polynomials $Q_0, \cdots, Q_q$, we observe that the coefficients of the bottom row of N_∞ are $-c_1 c_0^{-1}, \cdots, -c_p c_0^{-1}$. In particular the largest eigenvalue of N_∞ is $\delta(z)^{-1}$ and so is the largest eigenvalue of D. Hence the above method yields an algorithm to compute a bound of the form (3). In view of what has been said in section 1.1, this proves theorem 1. We finally notice that it actually suffices to perform a numerical triangulation of N_∞ instead of an exact one in our algorithm, that is $T \approx \hat{U} N_\infty \hat{U}^{-1}$. Indeed, (8) is the key bound we need. Modulo this modification, our algorithm can now be resumed as follows:

Algorithm B. The algorithm takes a number $B < \delta(z)^{-1}$ on input and produces a number A, such that $|f_k| \leqslant AB^k$ for all k.

B1. [Triangulate] Let $\varepsilon \leqslant B - \delta(z)$ and compute numerically a triangular matrix T and an invertible matrix $\hat{U}$, such that

$$\|T - \hat{U} N_\infty \hat{U}^{-1}\| \leqslant \tfrac{\varepsilon}{4}.$$

B2. ["Diagonalize"] Let $\mu \geqslant \max_{i<j} |T_{i,j}|$ and $\lambda \leqslant \min(\frac{\varepsilon}{4\mu\sqrt{q}}, 1)$.
Let $U := \mathrm{Diag}\,(1, \lambda, \cdots, \lambda^{q-1})\hat{U}$.
Let $D := \mathrm{Diag}\,((UN_\infty U^{-1})_{1,1}, \cdots, (UN_\infty U^{-1})_{q,q})$.
We have

$$\|D - UN_\infty U^{-1}\| \leqslant \tfrac{\varepsilon}{2}.$$

B3. [Compute k_0] Compute the symbolic matrix $U(N_\infty - N_k)U^{-1}$ whose coefficients are rational fractions in k. Let K be such that the norm each of these rational fractions is bounded by $\frac{K}{k+1}$ for all k. Let $k_0 \geqslant \frac{4K}{\varepsilon}$ be an integer. We have for all $k \geqslant k_0$

$$\|D - UN_k U^{-1}\| \leqslant \tfrac{3\varepsilon}{4}.$$

B4. [Return bound] Let $\Pi \geqslant (\|UN_\infty U^{-1}\| + qK)^{k_0} \geqslant \|P\|$.
Return $A \geqslant \sqrt{q}\Pi B^{-k_0}\|U^{-1}\|\ \|U\|\ \|\Phi_0\|$.

3 Evaluation in algebraic points

In this section we assume that $\mathbb{K}$ is an algebraic number field and $z \rightsquigarrow z'$ is a straight line path $z \to z'$ between two points $z, z' \in \mathbb{K}$ with $|z' - z| < \delta(z)$. We will show how the binary splitting technique can be used to compute $f(z')$ from $F(z)$ and (1) in an asymptotically efficient way. To guarantee exactness of our algorithm, we assume that

$$\sup_{u \in D(z,r)} |f(u)| \ \leqslant\ C \tag{9}$$

on a disk $D(z, r)$ with $|z' - z| < r < \delta(z)$. The previous section yields an algorithm to compute such a bound C for, say, $r = \frac{|z'-z|+\delta(z)}{2}$.

3.1 Naive evaluation of $f(z')$

From the Cauchy formula (5), we get the bound

$$\begin{aligned} & |f_m(z'-z)^m + f_{m+1}(z'-z)^{m+1} + \cdots| \\ \leqslant\ & \frac{C|z'-z|^m}{r^m} + \frac{C|z'-z|^{m+1}}{r^{m+1}} + \cdots = \frac{Cr}{r - |z'-z|}\left(\frac{|z'-z|}{r}\right)^m \end{aligned}$$

for the tail of the Taylor series of f in z, when evaluating in z' up to order m. Let $\tau = \frac{r}{|z'-z|}$ and

$$\begin{array}{rcl} C_1 & = & \frac{\log 2}{\log \tau}; \\ C_2 & = & \frac{\log 2C - \log(1-\tau^{-1})}{\log \tau}. \end{array} \tag{10}$$

Then it follows that for

$$m \geqslant C_1 n + C_2,$$

we have

$$|f(z') - (f_0 + \cdots + f_{m-1}(z'-z)^{m-1})| \leqslant \tfrac{\varepsilon}{2}.$$

It suffices therefore to evaluate $f_0 + \cdots + f_{m-1}(z'-z)^{m-1}$ with error $\leqslant \frac{\varepsilon}{2}$ in order to obtain an approximation for $f(z')$ with error $\leqslant \varepsilon$.

A "naive" way to do this would be to compute the coefficients $f_0, \cdots, f_{m-1}$ one by one and to use Horner's method for the evaluation of a polynomial (for instance). We first notice that $f_0 = f(z), \cdots, f_{p-1} = f^{(p-1)}(z)/(p-1)!$ are directly determined by $F(z)$. In order to obtain $f_p, \cdots, f_{\kappa-1}$, we use the relation (1) and its derivatives up to order $q-p-1$. The remaining coefficients are deduced from the recurrence relation (6). Clearly the time complexity of the evaluation of $f_0+\cdots+f_{m-1}(z'-z)^{m-1}$ by this naive method is $O(M(n)n)$, but the method has the advantage that it can also be used when z, z' and the coefficients of $P_0, \cdots, P_p$ are not algebraic.

3.2 The binary splitting algorithm

Let us now give a more sophisticated way to compute $f_0 + \cdots + f_{m-1}(z' - z)^{m-1}$. We first notice that the sequence $f_0, f_1(z'-z), f_2(z'-z)^2, \cdots$ also satisfies a linear difference equation of order q with coefficients in $\mathbb{K}[k]$, since (6) implies

$$Q_q(k) f_{k+q}(z'-z)^{k+q} + \cdots + [(z'-z)^q Q_0(k)] f_k (z'-z)^k = 0,$$

for all k. Therefore we may assume without loss of generality that $z' - z = 1$, and we just have to show how to evaluate $f_0 + \cdots + f_{m-1}$.

Let Φ_k be defined as in section 2.2 and define

$$\Sigma_{k;l} = \begin{pmatrix} f_0 + \cdots + f_{l-1} \\ \vdots \\ f_{q-1} + \cdots + f_{q+l-2} \end{pmatrix}$$

for each k. We claim that for all $k \geqslant 0$ and $l \geqslant 1$, there exist matrices $M_{k;l}, N_{k;l}$ with coefficients in $\mathbb{K}$, such that

$$\begin{array}{rcl} \Sigma_{k;l} & = & M_{k;l}\Phi_k; \\ \Phi_{k+l} & = & N_{k;l}\Phi_k. \end{array} \tag{11}$$

Indeed, if $l = 1$, we take $M_{k;1} = N_{k;1} = N_k$ with the notations from section 2.2. For $l \geqslant 2$, we compute $M_{k;l}$ and $N_{k;l}$ using binary splitting: let $l = l_1 + l_2$, with $l_1 = \lfloor \frac{l}{2} \rfloor$. Then we take

$$\begin{array}{rcl} M_{k;l} & = & M_{k;l_1} + M_{k+l_1;l_2}\ N_{k;l_1}; \\ N_{k;l} & = & N_{k+l_1;l_2}\ N_{k;l_1}. \end{array} \tag{12}$$

Actually, in order to avoid computations with rational numbers, it is more efficient to write $M_{k;l} = q_{k;l}^{-1} M'_{k;l}$ and $N_{k;l} = q_{k;l}^{-1} N'_{k;l}$, where the numbers $q_{k;l}$ are positive integers and the matrices $M'_{k;l}, N'_{k;l}$ have coefficients in the subring of algebraic integers of $\mathbb{K}$. In this representation, (12) becomes

$$\begin{array}{rcl} q_{k;l} & = & q_{k+l_1;l_2}\ q_{k;l_1}; \\ M'_{k;l} & = & q_{k+l_1;l_2}\ M'_{k;l_1} + M'_{k+l_1;l_2}\ N'_{k;l_1}; \\ N'_{k;l} & = & N'_{k+l_1;l_2}\ N'_{k;l_1}. \end{array} \tag{13}$$

These recurrence relations yield the following algorithm to compute $f(z')$:

Algorithm E. Given $\varepsilon = 2^{-n}$, the algorithm computes an approximation $\tilde{f}(z')$ for $f(z)$ with $|\tilde{f}(z') - f(z')| \leqslant \varepsilon$. We assume that $\mathbb{K}$ is an algebraic number field and $z \rightsquigarrow z' = z \to z'$ with $z, z' \in \mathbb{K}$ and $|z' - z| < \delta(z)$, and $P_0, \cdots, P_p \in \mathbb{K}[z]$.

E1. [Precomputation] Compute constants C, r, C_1, C_2 with (9) and (10), using algorithm **B**. Compute the difference equation (6) from (1) and reduce the general case to the case when $z' - z = 1$.

E2. [Binary splitting] Let $m = \max(\lceil C_1 n + C_2 \rceil, 1)$.
Compute $\Sigma_{0;m}$ using binary splitting (13).
Let L denote the first line of the matrix $\Sigma_{0;m}$.

E3. [Return approximation] Compute an approximation $\tilde{\Phi}_0$ of Φ_0 with entries in $\mathbb{K}$ and $\|\tilde{\Phi}_0 - \Phi_0\| \leqslant \frac{\varepsilon}{2\|L\|}$ from $F(z)$. Return $L\tilde{\Phi}_0$.

Let us now estimate the complexity of the algorithm. Step **E1** is a precomputation of cost $O(1)$. In step **E2** we have $m = O(n)$ and $\mathtt{size}(q_{k;1}) + \mathtt{size}(M'_{k;1}) + \mathtt{size}(N'_{k;1}) = O(\log k)$, since the Q_i are rational fractions in $\mathbb{K}(k)$. By induction, it follows that $\mathtt{size}(q_{k;l}) + \mathtt{size}(M'_{k;l}) + \mathtt{size}(N'_{k;l}) =$

$O(l \log k)$, uniformly in k and l. Hence, the computation of $M_{0;m}$ by binary splitting takes a time

$$\begin{aligned} & O(M(m \log m) + 2M(\lceil \tfrac{m}{2} \rceil \log m) + \cdots + 2^{\lfloor \log m / \log 2 \rfloor} M(2 \log m)) \\ = \quad & O(M(n \log n) \log n) = O(M(n \log^2 n)). \end{aligned}$$

By convention, we do not count the time to compute $\tilde{\Phi}_0$ in step **E3**. Since $\mathtt{size}(L) = O(n \log n)$ and $\log \frac{\varepsilon}{2\|L\|} = O(n + \mathtt{size}(L))$, the final multiplication $L\tilde{\Phi}_0$ takes a time $O(M(n \log n))$. Altogether, this proves theorem 2. We note that the complexity bound can be improved to $O(M(n \log n))$ if $\delta(z) = \infty$.

4 Analytic continuation

In this section, we show how to perform the analytic continuation of holonomic along arbitrary paths. Throughout this section, we assume that $\mathbb{K}$ is an algebraic number field.

4.1 Analytic continuation by algorithm E

Both the naive algorithm and algorithm **E** can actually be used — even in two different ways — to perform the numerical analytic continuation along $z \to z'$, i.e. to compute an approximation of the vector $F(z')$ instead of $f(z')$ only. Indeed, a first way would be to use the fact that $f', \cdots, f^{(p-1)}$ are holonomic by proposition 1. A second idea is to replace z' by the formal element $z' + \eta$ in the ring $\mathbb{K}[\eta]/(\eta^p)$ and to use either the naive algorithm or algorithm **E** with coefficients in the ring $\mathbb{K}[\eta]/(\eta^p)$ instead of $\mathbb{K}$. Doing so, the result $a_0 + a_1\eta + \cdots + a_{p-1}\eta^{p-1}$ of the evaluation approximates the Taylor series expansion $f(z') + f'(z')\eta + \cdots + \frac{1}{(p-1)!} f^{(p-1)} \eta^{p-1}$ of f in z' at the order $p - 1$.

In order to perform analytic continuation along an arbitrary path, the path first needs to be "discretized" by a broken line segment. In order to do this in an optimal way, we must study the complexity of algorithm **E**, if z and z' are allowed to vary in some open domain U on which $|f|$ is bounded by C, while ensuring that $D(z, |z' - z|) \subseteq U$. Then we may choose $r = \rho(z)$ as large as possible in (9). Hence, (10) implies

$$m = O(n \log^{-1} \tau + \log^{-1} \tau \,|\log \log \tau|),$$

uniformly in z and z'. Under the assumption that $\log \log \tau = O(n)$, this simplifies to

$$m = O(n \log^{-1} \tau).$$

Let $s = \mathtt{size}(z) + \mathtt{size}(z')$. It is not hard to see that after the normalization $z' - z = 1$ the total size of the recurrence relation (6) is bounded by $O(s)$. Hence,

$$\mathtt{size}(q_{k;1}) + \mathtt{size}(M_{k;1}) + \mathtt{size}(N'_{k;1}) = O(s + \log k),$$

uniformly in z, z', k. Consequently,

$$\mathtt{size}(q_{k;l}) + \mathtt{size}(M_{k;l}) + \mathtt{size}(N'_{k;l}) = O(l(s + \log k)),$$

uniformly in z, z', k and l. We infer that the binary splitting in step **E2** takes a time

$$O(M(m \log m(s + \log m))) = O(M(n \log n(s + \log n) \log^{-1} \tau)).$$

The final multiplication $L\tilde{f}\Phi_0$ in step **E3** takes a time $O(M(n(s + \log n)$ $\log^{-1} \tau))$. This establishes theorem 3.

If $|f|, \cdots, |f^{(p-1)}|$ are all bounded by C on U, then the analytic continuation by algorithm **E** clearly has time complexity $O(M(n \log n(s + \log n)$ $\log^{-1} \tau))$ as well. Now reconsider the problem of discretizing

$$z \rightsquigarrow z' \quad \approx \quad z = v_0 \to \cdots \to v_l = z'.$$

an arbitrary path $z \rightsquigarrow z'$ with $z, z' \in \mathbb{K}$. The uniform complexity bound for algorithm **E** enables us to give some heuristics of how to choose the points $v_1, \cdots, v_l$. Clearly, it will always be beneficial to choose them such that $\mathtt{size}(v_1) + \cdots + \mathtt{size}(v_l)$ is as small as possible. Let us now consider two other special cases of interest.

Turning around a singularity. Assume that $z \rightsquigarrow z' =\circlearrowleft z$ is a small circle around one of the singularities ω_i. Taking $v_0, \cdots, v_l$ on the circle, we need to find the optimal angle by which we progress. If we progress by an angle $\alpha = \frac{2\pi}{a}$, then we have

$$\tau = \frac{1}{2 \sin \frac{\alpha}{2}},$$

so the time needed to perform the analytic continuation is proportional to

$$\frac{-1}{\alpha \log\left(2 \sin \frac{\alpha}{2}\right)}.$$

Hence, the optimal value for a is 17.

Approaching singularities. An other situation which is often encountered is when U is a small open sector with a singularity ω_i at its corner. Given points $z, z' \in \mathbb{K}$, such that z' lies on the line segment between z and v_i, we want to perform analytic continuation along the straight line segment between z and z'. In this case, the optimal strategy is to choose $v_0, \cdots, v_l$ on the line segment between z and z' in such a way that $\delta(v_1) = \lambda\delta(v_0), \cdots, \delta(v_l) = \lambda\delta(v_{l-1})$. Then the time needed to perform the analytic continuation is proportional to

$$\frac{1}{\log(\lambda)\log(1-\lambda)}$$

and the optimal factor λ is seen to be close to $\frac{1}{2}$. Of course, in order to take $\lambda \approx \frac{1}{2}$, the sector U should have an opening of at least $\frac{\pi}{3}$; otherwise λ is taken to be minimal such that the disks $D(v_0, |v_1 - v_0|), \cdots, D(v_{l-1}, |v_l - v_{l-1}|)$ fit into U.

Remark. We note that if truncated power series are evaluated by the naive algorithm, then the time complexity is bounded by $O(n \log^{-1} \tau\, M(n))$ and the optimal values for a and λ are the same.

4.2 Transition matrices

In this section we introduce the concept of transition matrices and prove some bounds which will be useful in the next section. Let $z \rightsquigarrow z'$ be an arbitrary non singular path. Since the value $F(z')$ of F in z' depends linearly on the initial conditions $F(z)$ of F in z, there exists a matrix $\Delta_{z \rightsquigarrow z'}$ such that

$$F(z') = \Delta_{z \rightsquigarrow z'} F(z)$$

for all possible initial conditions $F(z)$. This matrix is called the *transition matrix* along $z \rightsquigarrow z'$. Obviously,

$$\Delta_{z \rightsquigarrow z' \rightsquigarrow z''} = \Delta_{z' \rightsquigarrow z''} \Delta_{z \rightsquigarrow z'},$$

for all compositions $z \rightsquigarrow z' \rightsquigarrow z''$ of $z \rightsquigarrow z'$ with a path $z' \rightsquigarrow z''$.

We claim that the problem of computing transition matrices is equivalent to the problem of analytic continuation. Indeed, for $0 \leqslant i \leqslant p-1$, let $f[i]$ be the function f which satisfies (1) with initial conditions $F[i] = E_i$, where

E_i denotes the column vector with entries $0, \overset{i \text{ times}}{\cdots}, 0, 1, 0, \overset{p-i-1 \text{ times}}{\cdots}, 0$. Then $\Delta_{z \rightsquigarrow z'}$ is just the matrix with columns $F[0](z'), \cdots, F[p-1](z')$.

Let us now study $\Delta_{z \to z'}$, for $|z' - z| \to 0$. Let $r < \delta(z)$ be given and assume that we have an upper bound C for the $|f[i]^{(j)}|$ with $0 \leqslant i, j \leqslant p-1$ on $D(z, r)$; such a bound can be computed by algorithm **B**. Then Cauchy's formula yields the bounds

$$|f[i]_k^{(j)}| \leqslant \frac{C}{r^k}$$

for the coefficients of the Taylor series expansions

$$f[i]^{(j)}(z+u) = f[i]_0^{(j)} + f[i]_1^{(j)} u + f[i]_2^{(j)} u^2 + \cdots$$

of the $f[i]^{(j)}$ in z. Consequently, if $|z' - z| < r$, then

$$|f[i]^{(j)}(z') - f[i]_0^{(j)}| \leqslant \frac{C}{r - |z' - z|} |z' - z|.$$

Setting $C_3 = \frac{pC}{2(r - |z' - z|)}$, it follows that

$$\|\Delta_{z \to z'} - Id\| \leqslant \tfrac{1}{2} C_3 |z' - z|. \tag{14}$$

For $|z' - z| < \frac{1}{C_3}$ we also obtain

$$\|\Delta_{z' \to z} - Id\| \leqslant C_3 |z' - z|, \tag{15}$$

since $(Id + E)^{-1} = Id - E + E^2 - E^3 + \cdots$ for matrices E with $\|E\| < 1$.

4.3 Analytic continuation: the general case

Let us now turn to the general case of performing the numerical analytic continuation along an arbitrary non singular path $z \rightsquigarrow z'$. We split the path into three parts

$$z \rightsquigarrow z' \quad \approx \quad z \rightsquigarrow w \rightsquigarrow w' \rightsquigarrow z',$$

where w and w' are approximations of z resp. z' in $\mathbb{K}$. In section 4.1 we have already shown how to perform the analytic continuation along $w \rightsquigarrow w'$. In order to perform the analytic continuation between z resp. z' and their approximations w resp. w', we approximate the paths $z \rightsquigarrow w$ in $w' \rightsquigarrow z'$ by broken line paths

$$\begin{aligned} z \rightsquigarrow w &\quad\approx\quad v = v_l \to \cdots \to v_1 = w; \\ w' \rightsquigarrow z' &\quad\approx\quad w' = v_1' \to \cdots \to v_{l'}' = v', \end{aligned}$$

which depend on the desired precision of the approximation of $f(z')$. Here $v_1, \cdots, v_l, v'_1, \cdots, v'_{l'}$ are in $\mathbb{K}$ and for efficiency reasons we choose them such that the precision of the approximation is doubled at each step (i.e. $\texttt{size}(v_i) = O(2^i)$ and $\log|z - v_i| = O(2^i)$ for $i \to \infty$). For this purpose (see also algorithm **C** below), we implement a function `truncate`, which given u and rational $2^{-n-1} < \varepsilon \leqslant 2^{-n}$ returns the element $(\lfloor\frac{\Re u}{\varepsilon}\rfloor + \lfloor\frac{\Im u}{\varepsilon}\rfloor i)\varepsilon$ of $\mathbb{K}$.

Assume now that we want to compute $F(z')$ modulo an error $\leqslant \varepsilon = 2^{-n}$. In the previous section we have shown how to compute constants C_3, r and C_4, r', such that

$$\begin{array}{rcll} \|\Delta_{z\to t} - Id\| & \leqslant & C_3|t-z| & (\text{for } |t-z| < r); \\ \|\Delta_{t\to z} - Id\| & \leqslant & C_3|t-z| & (\text{for } |t-z| < r); \\ \|\Delta_{t\to z'} - Id\| & \leqslant & C_4|z'-t| & (\text{for } |z'-t| < r'); \\ \|\Delta_{z'\to t} - Id\| & \leqslant & C_4|z'-t| & (\text{for } |z'-t| < r'). \end{array} \tag{16}$$

We will require $|v_i - z| < r$ and $|z' - v'_{i'}| < r'$ for all i, i'. Denote $\Delta = \Delta_{v\rightsquigarrow v'}$ and assume that $v, v', \tilde{F}(z), \tilde{\Delta}$ are such that

$$\begin{array}{rcl} |v - z| & \leqslant & c_1\varepsilon; \\ |z' - v'| & \leqslant & c_2\varepsilon; \\ \|\tilde{F}(z) - F(z)\| & \leqslant & c_3\varepsilon; \\ \|\tilde{\Delta} - \Delta\| & \leqslant & c_4\varepsilon. \end{array}$$

Here c_1, c_2, c_3 and c_4 are indeterminates for the moment. We have the following error estimations:

$$\begin{array}{rcl} \|F(v) - \tilde{F}(z)\| & \leqslant & (c_1C_3 + c_3)\varepsilon \\ \|F(v') - \tilde{\Delta}\tilde{F}(z)\| & \leqslant & ((1 + c_1C_3\varepsilon)c_4 + (c_1C_3 + c_3)(1 + c_4\varepsilon)\|\Delta\|)\varepsilon \\ \|F(z') - \tilde{\Delta}\tilde{F}(z)\| & \leqslant & \|F(v') - \tilde{\Delta}\tilde{F}(z)\| + (c_2C_4(1 + c_1C_3\varepsilon)\|\Delta\|)\varepsilon \end{array}$$

Imposing the conditions $c_1 \leqslant \frac{1}{C_3\varepsilon}$ and $c_4 \leqslant \frac{1}{\varepsilon}$, this yields

$$\|F(z') - \tilde{\Delta}\tilde{F}(z)\| \leqslant 2(c_4 + (c_1C_3 + c_3 + c_2C_4)\|\Delta\|)\varepsilon.$$

At this point, it suffices to compute an upper bound for $\|\Delta\|$. This is done by computing an approximation $\tilde{\Delta}_{w\rightsquigarrow w'}$ for $\Delta_{w\rightsquigarrow w'}$ with $\|\tilde{\Delta}_{w\rightsquigarrow w'} - \Delta_{w\rightsquigarrow w'}\| \leqslant 1$ by algorithm **E**. Then we have

$$\|\Delta\| \leqslant |w - z|C_3\ (\|\tilde{\Delta}_{w\rightsquigarrow w'}\| + 1)\ |z' - w'|C_4 = C_5. \tag{17}$$

Taking $c_1 \leqslant \frac{1}{8C_3C_5}, c_2 \leqslant \frac{1}{8C_4C_5}, c_3 \leqslant \frac{1}{8C_5}, c_4 \leqslant \frac{1}{8}$ now suffices to ensure that

$$\|F(z') - \tilde{\Delta}\tilde{F}(z)\| \leqslant \varepsilon.$$

The analytic continuation algorithm can be resumed as follows:

Algorithm C. The algorithm takes a non singular broken line path $z \rightsquigarrow z'$ and a rational number $\varepsilon > 0$ on input, and computes an approximation $\tilde{F}(z')$ for $F(z')$ with $\|\tilde{F}(z') - F(z')\| \leqslant \varepsilon$. We assume that $\mathbb{K}$ is an algebraic number field.

C1. [Precomputation] Compute constants r, r', C_3, C_4 such that (16) holds using algorithm **B**. Choose $w, w' \in \mathbb{K}$ with $|w - z| < r$ and $|w' - z'| < r'$ and replace the path $w \to z \rightsquigarrow z' \to w'$ with a homotopic broken line path, along which the transition matrix $\Delta_{w \rightsquigarrow w'}$ can be evaluated by algorithm **E**. Use algorithm **E** and (17) to compute an upper bound C_5 for $\|\Delta_{w \rightsquigarrow w'}\|$.

C2. [Compute constants] Let $c_1 \leqslant \frac{1}{8C_3C_5}, c_2 \leqslant \frac{1}{8C_4C_5}, c_3 \leqslant \frac{1}{8C_5}$ and $c_4 \leqslant \frac{1}{8}$. Decrease c_1 and c_4 if necessary, such that $c_1 \leqslant \frac{1}{C_3 \varepsilon}$ and $c_4 \leqslant \frac{1}{\varepsilon}$.

C3. [Approximate endpoints]
Let $v_2 := \texttt{truncate}\,(z, \frac{|w-z|}{2}), \cdots, v_l := \texttt{truncate}\,(z, \frac{|w-z|}{2^l})$,
where l is minimal such that $|z - v_l| \leqslant \frac{\varepsilon}{c_1}$.
Let $v'_2 := \texttt{truncate}\,(z, \frac{|z'-w'|}{2}), \cdots, v'_{l'} := \texttt{truncate}\,(z, \frac{|z'-w'|}{2^{l'}})$,
where l' is minimal such that $|z' - v'_{l'}| \leqslant \frac{\varepsilon}{c_2}$.

C4. [Return approximation] Compute $\tilde{F}(z)$, with $\|\tilde{F}(z) - F(z)\| \leqslant \frac{\varepsilon}{c_3}$.
Compute $\tilde{\Delta}_{v_l \rightsquigarrow v'_{l'}}$ with $\|\tilde{\Delta}_{v_l \rightsquigarrow v'_{l'}} - \Delta_{v_l \rightsquigarrow v'_{l'}}\| \leqslant \frac{\varepsilon}{c_4}$ by algorithm **E**.
Return $\tilde{\Delta}_{v_l \rightsquigarrow v'_{l'}} \tilde{F}(z)$.

Let us finally estimate the time complexity of algorithm **C**, which is determined by the computation time of $\tilde{\Delta}_{v_l \rightsquigarrow v'_{l'}}$ in step **C4**. By theorem 3, this computation time is bounded by

$$O\left(M(n \log^2 n) + \sum_{i=0}^{O(\log n)} M\left(\frac{n}{2^i}(2^i + \log n)\right)\right),$$

since $l = O(\log n)$ and $l' = O(\log n)$. Now

$$\begin{aligned}\sum_{i=0}^{O(\log n)} M\left(\frac{n}{2^i}(2^i+\log n)\right) &= \\ \left(\sum_{i=0}^{\lfloor \log\log n \rfloor} + \sum_{i=\lfloor \log\log n \rfloor+1}^{O(\log n)}\right) M\left(\frac{n}{2^i}(2^i+\log n)\right) &= O(M(n\log^2 n\log\log n)),\end{aligned}$$

which completes the proof of theorem 4.

5 Conclusion

We have described several algorithms for the multiple precision evaluation and analytic continuation of holonomic functions, such that the user has explicit control over the computation errors. For holonomic functions over the algebraic numbers, the asymptotic time complexities of our algorithms as a function of the number of required digits are the best actually known, except in the case of elementary functions, where the AGM method applies. In particular, many mathematical constants involving special functions can be approximated extremely fast both theoretically and in practice [6]. We conclude this section with some remarks.

The naive method versus binary splitting. Although the binary splitting method for summing power series has a better asymptotic complexity than the naive method, it would be interesting to know for which precisions it becomes more efficient in practice. The answer to this question is hard to give at the moment and depends on several issues.

First, the binary splitting method clearly suffers from the fact that it uses q by q matrix multiplications, whence it has a bad dependence on q. Here we notice that the size of the matrices can sometimes be reduced. For instance, if we want to evaluate $\operatorname{ch} 1 = 1 + \frac{1}{2} + \frac{1}{24} + \cdots$, then we may take one by one matrices $(\frac{1}{(2k-1)(2k)})$ for the N_k and sum only the first $\lceil \frac{m}{2} \rceil$ terms of the expansion $\operatorname{ch} 1 = 1 + N_1 + N_2 N_1 + \cdots$. We also notice that for large values of q, FFT-multiplication becomes profitable for smaller precisions, since we can FFT-transform the entire matrices.

The binary splitting method also suffers from a large amount of overhead for small precisions. The first reason is that the final division $M_{0;l} = \frac{M'_{0;l}}{q_{0;l}}$ is quite expensive and the second reason is that binary splitting is quite expensive when the ratio $\tau = \frac{\delta(z)}{|z'-z|}$ is too small. For frequently used special functions, with say $z = 0$ and $0 < z' < 1$, a solution might be to tabulate the values of $F(\frac{1}{2^8}), \cdots, F(\frac{2^8-1}{2^8})$.

We finally notice that the binary splitting method may very well be combined with the naive method, by computing the matrices $M_{k;l}, N_{k;l}$ up to some order $m' < m$ only. Horner's method is used to complete the computation in order. Consequently, we avoid that the coefficients of the $M_{k;l}$ and $N_{k;l}$ grow to large.

Initial conditions in "fake singularities". Sometimes, the zeros of P_p are not actual singularities of f and for certain classical special functions, the initial conditions are even specified in such "fake singularities". For example, the sine-integral function

$$\operatorname{Si} x = \int_0^x t^{-1} \sin t dt,$$

satisfies the equation

$$z \operatorname{Si}''' z + 2 \operatorname{Si}'' z + z \operatorname{Si}' z = 0,$$

with initial conditions $\operatorname{Si}(0) = 0, \operatorname{Si}'(0) = 1, \operatorname{Si}''(0) = 0$. Using the recurrence relation

$$\operatorname{Si}_{k+2} + \frac{2}{k} \operatorname{Si}_{k+1} + \frac{1}{(k+1)(k+2)} \operatorname{Si}_k = 0,$$

algorithms **B**, **E** and **C** still apply in this case. Actually, this is a general situation: it suffices that the power series expansion be convergent and that $\mathbb{K}$ contains z.

Multivariate holonomic functions. A multivariate function $f(z_1, \cdots, z_k)$ is said to be holonomic, if f is holonomic in each of its variables. It is classical that the restriction of a multivariate holonomic function to a straight line segment is a holonomic function in one variable only. Moreover, the differential equation satisfied by this restriction can be computed in a generic way, i.e. for a generic straight line segment. Consequently, our algorithms generalize in a straightforward way to the multivariate case.

Small perturbations. The trick to compute $f(z'), \cdots, f^{(p-1)}(z')$ simultaneously, by introducing the infinitesimal variable η and working in the ring $\mathbb{K}[\eta]/(\eta^p)$ instead of $\mathbb{K}$ can be generalized: if we allow the coefficients of $P_0, \cdots, P_p$ to depend on η (i.e. by taking $P_0, \cdots, P_p \in \mathbb{K}[\eta]/(\eta^r)[z]$), then we may compute the effect of small perturbations of (1) in η up to a finite number of terms.

Singularities. When the point z' in which we want to evaluate f is near to a singularity, the bounds for the transition matrices may become very bad. No straightforward numerical methods can be applied to solve this problem, and numerical resummation techniques are essentially needed to handle this situation [13, 9]. Here we notice that the Borel and Laplace transforms preserve holonomy, therefore our algorithm can theoretically be used in the resummation process. We intend to study this issue more closely in a forthcoming paper.

We also notice that the binary splitting algorithm can be used to efficiently evaluate holonomic functions in the neighbourhood of points where the series expansion diverges, by summing "up to the smallest term". Of course, we only get limited approximations of the exact value of the holonomic function in this way, but it is well known that these approximations have exponential accuracy when we approach the singularity. Furthermore, such approximations may again be useful for heuristic zero tests in computer algebra.

6 References

References

[1] V. Strassen A. Schönhage. Schnelle Multiplikation grosser Zahlen. *Computing 7*, 7:281–292, 1971.

[2] J.M. Borwein and P.B. Borwein. *Pi and the AGM.* Wiley, 1987.

[3] R.P. Brent. The complexity of multiprecision arithmetic. In *Complexity of Computational problem solving*, 1976.

[4] R.P. Brent. Fast multiple-precision evaluation of elementary functions. *Journal of the ACM*, 23(2):242–251, 1976.

[5] R.P. Brent and E. MacMillan. Some new algorithms for high-precision computation of Euler's constant. *Journal of the ACM*, 23:242–251, 1980.

[6] B. Haible and T. Papanikolaou. Fast multiple-precision evaluation of elementary functions. Technical Report TI-7/97, Universität Darmstadt, 1997. Available on the web at `http://www.informatik.th-darmstadt.de/TI/LIDIA`.

[7] D.E. Knuth. *The art of computer programming*, volume 2: Seminumerical algorithms. Addison-Wesley, 2-nd edition, 1981.

[8] L. Lipshitz. D-finite power series. *Journal of Algebra*, 122:353–373, 1989.

[9] F. Naegele. *Autour de quelques équations fonctionnelles analytiques.* PhD thesis, l'I.N.P.G., Grenoble, France, 1995.

[10] D. Richardson. A simplified method for recognizing zero among elementary constants. In *Proc. ISSAC '95*, pages 104–109, 1995. See also "How to recognize zero", accepted for publication in JSC.

[11] E. Salamin. Computation of π using the arithmetic-geometric mean. *Mathematics of Computation*, 30:565–570, 1976.

[12] R.P. Stanley. Differentially finite power series. *European J. Combin.*, 1:175–188, 1980. MR # 81m:05012.

[13] J. Thomann. Procédés formels et numériques de sommation de séries d'équations différentielles. *Expositiones Math.*, 13:223–246, 1995.

[14] J. van der Hoeven. Fast evaluation of holonomic functions. Technical Report LIX/RR/96/05, LIX, École polytechnique, France, 1996. Available on the web at `http://lix.polytechnique.fr:~vdhoeven`.

[15] J. van der Hoeven. *Automatic asymptotics*. PhD thesis, École polytechnique, France, 1997.

[16] D. Zeilberger. A holonomic systems approach to special functions identities. *Journal of Comp. and Appl. Math.*, 32:321–368, 1990.