Guessing singular dependencies*

Joris van der Hoeven

CNRS, Département de Mathématiques
Bâtiment 425
Université Paris-Sud
91405 Orsay Cedex
France

Email: vdhoeven@lix.polytechnique.fr
Web: http://lix.polytechnique.fr/~vdhoeven

November 4, 2013

Given d complex numbers $z_1, \ldots, z_d$, it is classical that linear dependencies $\lambda_1 z_1 + \cdots + \lambda_d z_d = 0$ with $\lambda_1, \ldots, \lambda_d \in \mathbb{Z}$ can be guessed using the LLL-algorithm. Similarly, given d formal power series $f_1, \ldots, f_d \in \mathbb{C}[[z]]$, algorithms for computing Padé-Hermite forms provide a way to guess relations $P_1 f_1 + \cdots + P_d f_d = 0$ with $P_1, \ldots, P_d \in \mathbb{C}[z]$. Assuming that $f_1, \ldots, f_d$ have a radius of convergence $r > 0$ and given a real number $R > r$, we will describe a new algorithm for guessing linear dependencies of the form $g_1 f_1 + \cdots + g_d f_d = h$, where $g_1, \ldots, g_d, h \in \mathbb{C}[[z]]$ have a radius of convergence $\geqslant R$. We will also present two alternative algorithms for the special cases of algebraic and Fuchsian dependencies.

Keywords: guessing, asymptotic dependency, orthogonalization, analytic continuation, Fuchsian singularity, Padé-Hermite forms, algorithm

A.M.S. subject classification: 30E10, 30B40, 41A21, 65B05

1. Introduction

1.1. General background and results

Consider an infinite sequence $f_0, f_1, \ldots$ of complex numbers. If $f_0, f_1, \ldots$ are the coefficients of a formal power series $f \in \mathbb{C}[[z]]$, then it is well-known [Pól37, Wil94, FS96] that the asymptotic behaviour of the sequence f_n is closely related to the behaviour of the generating function $f(z)$ near its dominant singularity. Now, if f is the solution to some complicated equation, then it can be hard to compute the asymptotic behaviour using formal methods. On the other hand, the coefficients $f_0, f_1, \ldots$ of such a solution f can often be computed numerically up to a high order. With this numerical evidence at hand, it is natural to raise the following questions:

1. Can we *guess* the asymptotic behaviour of $f_0, f_1, \ldots$?
2. Can we guess the behaviour of $f(z)$ near its dominant singularity?

These questions can be regarded as part of the construction of a more general toolbox for the "experimental mathematician" [BBKW06, BD09]. More specifically, we advocate the systematic integration of "guessing tools" into symbolic computation packages. Indeed, current systems can be quite good at all kinds of formal manipulations. However, in the daily practice of scientific discovery, it would be helpful if these systems could also

*. This work was partially supported by the ANR Gecko and ANR-09-JCJC-0098-01 MaGiX projects, and by the Digiteo 2009-36HD grant and Région Ile-de-France.

detect hidden properties, which may not be directly apparent or expected. Furthermore, the guessing tool is allowed to be heuristic, so that it only *suggests* hidden properties; at a second stage, one may then search for full proofs using other techniques.

One well-known tool in this direction is the LLL-algorithm [LLL82]. Given d numbers $z_1, \ldots, z_d \in \mathbb{C}$, it can be used in order to guess relations of the form

$$\lambda_1 z_1 + \cdots + \lambda_d z_d = 0 \qquad (\lambda_1, \ldots, \lambda_d \in \mathbb{Z}). \tag{1}$$

Given d formal power series f_1, ..., $f_d \in \mathbb{C}[[z]]$, algorithms for the computation of Padé-Hermite forms [BL94, Der94] can be used in order to guess linear relations

$$P_1 f_1 + \cdots + P_d f_d = 0 \qquad (P_1, \ldots, P_d \in \mathbb{C}[z]). \tag{2}$$

A well-known implementation is provided by the Gfun package [SZ94]. Given a finite number of coefficients of a formal power series $f \in \mathbb{C}[[z]]$, the Gfun package is able to guess a closed form formula for f or a linear differential equation with coefficients in $\mathbb{C}[z]$ satisfied by f. Indeed, it suffices to take $f_1 = f, f_2 = f', \ldots, f_d = f^{(d-1)}$ in (2) in order search for small linear differential equations satisfied by f.

Unfortunately, many interesting formal power series f do not admit closed form formulas and do not satisfy linear differential equations with polynomial coefficients. In that case, we can still use asymptotic extrapolation [Hoe09] in order to guess the asymptotic behaviour of the coefficients. However, this only provides us some rough idea about the behaviour of f at its dominant singularity. In practice, it often happens that f locally satisfies an algebraic or differential equation with analytic coefficients, even though these coefficients fail to be polynomials. For instance, combinatorics [Pól37, FS96] is full with examples of generating functions which are not algebraic, but whose dominant singularities are algebraic.

In this paper, we will describe two approaches to detect analytic dependencies on a compact disk: the first one assumes that we have an algorithm for the analytic continuation of f and relies on the monodromy of f at its singularities. The second approach is purely numerical and makes no special assumptions on f. On our way, we will encounter various interesting related problems, such as the determination of the radius of convergence or the singularities of f. We will also propose heuristic solutions to these problems, thereby extending the basic toolbox for experimenting with analytic functions.

1.2. Overview of the paper

Since all algorithms in this paper are directly or indirectly based on heuristics, it is important to investigate how much confidence we can attach to the computed results. In section 2, we will present a survey of different kinds of heuristic algorithms which are used in symbolic computation. Some of these heuristics are relatively benign when compared to others, so it will be useful to have some general insights on the reliability of different types of heuristic algorithms.

The remainder of the paper is divided into two main parts. In the first part (sections 3, 4 and 5), we will develop some basic tools for later use. In the second part, we turn to the main topic of this paper: the disclosure of local analytic relations between analytic functions. Two main types of analytic input functions f will be considered:

AN1. Functions for which we can merely compute a large number of the Taylor coefficients $f_0, f_1, f_2, \ldots$ at the origin.

AN2. Functions for which we have a reasonably efficient algorithm for their analytic continuation, so that we can also compute the Taylor coefficients of f at other points besides the origin.

The second situation typically arises when f is the solution of a differential or more general functional equation; see section 4.1 for a further discussion on analytic continuation.

One of the most basic problems concerning an analytic function f which is only known by its Taylor coefficients, is to determine its radius of convergence. In section 3, we will present two methods. The first one is related Cauchy-Hadamard's formula and provides rough approximations under mild assumptions on f. The second method provides much better approximations, but only works if f admits a single dominant singularity of a simple type.

Building on the algorithm for approximating the radius of convergence of f, we next turn our attention to the problem of locating its singularities. In section 4.2, we first restrict ourselves to the dominant singularities of f (i.e. the singularities of minimal norm). Assuming **AN2**, we next present algorithms for the exploration of the Riemann surface of f beyond its dominant singularities.

In the special case when f is meromorphic on a compact disk $\bar{\mathcal{D}}_r := \{z \in \mathbb{C}: |z| \leqslant r\}$, section 5 contains a special purpose algorithm for the determination of a polynomial P such that $P f$ is analytic on $\bar{\mathcal{D}}_r$. This algorithm works under the hypothesis **AN1** and induces better algorithms for the problems in sections 3 and 4 in this special case.

In the second part of this paper, we turn to the detection of dependencies between analytic functions on a compact disk $\bar{\mathcal{D}}_r$. More precisely, assume that $f_1, \ldots, f_d \in \mathbb{C}[[z]]$ are fixed convergent power series. We are interested in the determination of linear dependencies of the form

$$g_1 f_1 + \cdots + g_d f_d = h \qquad (g_1, \ldots, g_d, h \in \mathbb{C}[[z]]), \tag{3}$$

where $g_1, \ldots, g_d, h$ are analytic on $\bar{\mathcal{D}}_r$. Modulo a scaling $f(z) \mapsto f(r z)$, we may assume without loss of generality that $r = 1$.

In section 6, we assume **AN2** and first consider the two special cases of algebraic and Fuchsian dependencies. In the case of algebraic dependencies, we take $f_i = \varphi^{i-1}$ for a fixed function φ, and also require that $h = 0$. In the case of Fuchsian dependencies, we take $f_i = \varphi^{(i-1)}$ for a fixed function φ, and again require that $h = 0$. The second algorithm only succeeds in the case when all singularities of f on the disk are of Fuchsian type (see sections 3.2 and 6.2 for detailed definitions). The main idea behind the method is to compute the set of of functions generated by φ and its analytic continuation around its singularities in $\bar{\mathcal{D}}_r$. If there exists an algebraic dependency, then this set is finite and we may use it to find the dependency. If there exists a Fuchsian dependency, then the set is contained in finite dimensional vector space, and we may construct the dependency from a basis of this vector space.

In section 7, we only assume **AN1** and consider the general problem of determining dependencies of the form (3). We will describe a purely numerical algorithm based on Gram-Schmidt orthogonalization for finding such relations. The idea is to simply truncate the power series expansions and then apply the numerical method. We will prove that an asymptotic convergence theorem which shows that this approach is at least correct "at the limit".

In the last section 8, we will present some numerical experiments with the algorithm from section 7. In section 8.1, we first consider some examples of relations which were recognized by the algorithm. In section 8.2, we will also examine the behaviour of the algorithm in the case when $f_1, \ldots, f_d$ are analytically independent. In fact, the algorithm from section 7 still requires some human cooperation for deciding whether we really found a relation. Based on the numerical experiments, we conclude with some perspectives for the design of a fully automatic algorithm.

Acknowledgments. We would like to thank the three referees for their detailed and valuable comments on a first version of this paper.

2. On the use of heuristic algorithms

As soon as we enter the area of heuristic and non mathematically proven algorithms, it is natural to ask how much confidence can be attached to the output. In fact, there is a large spectrum of situations which can occur, depending on the precise nature of the heuristics. There may also be a trade-off between computational complexity and correctness: is it better to rely on a slow deterministic algorithm for polynomial factorisation over a finite field, or on a fast probabilistic algorithm?

Without striving for exhaustion, section 2.1 a catalogue of different kinds of heuristic algorithms which occur in the area of symbolic computation. Each of the heuristic algorithms considered in this paper will fit into one of these categories. We will also describe some standard strategies in order to increase the level of confidence and examine typical examples for which heuristic algorithms may fail.

2.1. Various kinds of heuristic algorithms

Correct modulo conjectures. A high confidence can typically be attached to algorithms which are correct modulo conjectures. One important example of this situation in computer algebra is zero testing of transcendental functions or constants.

For example, consider the class of *exp-log constants*, built up from the rationals $\mathbb{Q}$ using the operations $+$, exp and log. The following conjecture [Lan71] is a major open problem in number theory:

Conjecture 1. (Schanuel) *Let $\alpha_1, \ldots, \alpha_k$ be complex numbers which are linearly independent over the rational numbers $\mathbb{Q}$. Then the transcendence degree of*

$$\mathbb{Q}(\alpha_1, \ldots, \alpha_k, \exp(\alpha_1), \ldots, \exp(\alpha_k)) : \mathbb{Q}$$

is at least k.

From the computer algebra point of view, the conjecture implies [CP78] that all numerical relations between exp-log constants can be deduced from the usual rules $\exp(x+y) = \exp x \exp y$ and $\log(x\, y) = \log x + \log y$. Based on this fact, Richardson has given a zero test for exp-log constants [Ric97, Ric07] with the following properties: (1) if the algorithm terminates on a given input, then the result is correct; (2) if the conjecture holds, then the algorithm always terminates; (3) if the algorithm does not terminate on a given input, then a counterexample to the conjecture can be constructed from this input.

Similar situations occur in algorithmic number theory, depending on the correctness of Riemann's hypothesis [Bel11]. In this paper, the algorithms in sections 4.3 and 6 rely on zero tests for analytic functions. If we know that our analytic functions lie in special classes (such as the class of exp-log functions, or the class of differentially algebraic functions), then these zero tests might be reduced to suitable conjectures [Hoe01].

Reduction to other heuristic algorithms. Another frequent situation is that an algorithm relies on one of more other heuristic subalgorithms, with the property that, for a given input, we obtain the right output whenever all calls of the subalgorithm(s) return the correct results. From a theoretical point of view, this corresponds to correctness modulo one or more oracles.

For instance, in section 3, we will give heuristic algorithms for the computation of the radius of convergence of an analytic function. This will be a basic building block for many of the other algorithms in this paper. Nevertheless, the correctness of several of these other algorithms reduces to the correctness of this basic building block.

If, for input functions in a more restricted class, we may determine their convergence radii with larger accuracy, then this will immediately make all the other algorithms more reliable. For instance, if we consider a class (e.g. rational or algebraic functions) for which convergence radii can be computed exactly, then the algorithms in section 4.2 are no longer heuristic.

Correctness with probability one. In computer algebra, there are plenty of algorithms which are correct unless some degeneracy occurs. For instance, solving a system of polynomial equations using numerical homotopy methods [Ver96, SW05] is correct unless the chosen paths accidentally hit a singularity. Moreover, if paths are chosen at random, then such accidents occur with probability zero. Similarly, in order to test whether a real analytic function f is zero, it suffices to pick a random number x and test whether $f(x)$ vanishes.

Correctness with high probability. In practice, there is no such thing as a random real number, since we can only represent numbers with finite precision. The above examples of heuristic algorithms which are correct "with probability 1" are really correct with probability $1-\mathcal{O}(2^{-p})$, where p is the working precision. There are many other examples of heuristic algorithms which are correct with high probability. For instance, in order to show that two polynomials P, Q in $\mathbb{Z}[x]$ have no common divisors, it suffices to pick a "sufficiently random" prime p and show that the reductions of P and Q in $\mathbb{F}_p[z]$ have no common divisors.

We also recall that there are two classical types of probabilistic algorithms. The examples mentioned above are only correct with high probability and fall into the class of so-called Monte Carlo algorithms. The second kind of Las Vegas algorithms are always correct, but only fast with high probability. For instance, all known fast algorithms for factoring polynomials over a finite field $\mathbb{F}_p$ are probabilistic Las Vegas type algorithms [GG02].

Asymmetric correctness. We also would like to underline another aspect of heuristic tests, such as zero tests: it frequently occurs that only one direction of the test is difficult and based on heuristics, whereas the other direction is "easy". For instance, assume that we want to test whether an exp-log constant is zero. If the constant is non zero, then it is easy to prove this by evaluating the constant using interval arithmetic [Moo66] using a sufficiently high precision and verify that zero is not in the interval enclosure of our constant. The difficult case, for which Richardson's algorithm [Ric07] relies on Schanuel's conjecture, is when the exp-log constant is zero and we want to prove this.

Numerical algorithms. Essentially all numerical algorithms are heuristic in the sense that they only compute approximate answers. Numerical analysts therefore have a lot of experience in dealing with heuristics and intentionally vague arguments. The very aim of numerical algorithms is often not to compute a mathematically sound result, but rather to design a method which "works" for solving a practical problem.

This does not withstand that experts in this area have a great sense of how accurate and trustworthy various methods are. Ideally speaking, any numerical algorithm comes with a detailed error analysis. Most textbooks on numerical analysis start with a section on the various sources of error and how to take into account the machine accuracy [PTVF07, section 1.1]. For more statistical algorithms, there is also a tradition of trying to quantify the level of trust that can be attached to the computed results. Finally, for many algorithms, it is possible to automate the error analysis by using interval arithmetic [Moo66].

Ultimate correctness. Classical numerical algorithms operate at a fixed precision p. From a theoretical point of view [Wei00, Hoe11], it can be investigated whether a given numerical algorithm is at least "ultimately correct" in the sense that the computed output tends to the mathematically correct result if p tends to infinity. If the numerical algorithm comes with a rigourous error analysis, then we have a means for controlling the distance ε_p between our approximations at a given precision p and the ultimate limit. In particular, for any $\delta > 0$, we may then select a precision p with $\varepsilon_p < \delta$.

However, there are many situation in which we have no means for controlling the distance ε_p, even though our algorithm is ultimately correct. For instance, the computation of the radius of convergence ρ of a differentially algebraic function is undecidable [DL89], even though there exists an algorithm [Hoe05, Hoe07] for the computation of a sequence $\rho_1 \leqslant \rho_2 \leqslant \cdots$ with $\lim_{n \to \infty} \rho_n = \rho$. Similarly, for a large class of functions, the formula (4) below gives an approximation for the radius of convergence of a series (when computing with a precision of $p = n$ bits), but the formula is only ultimately correct (5).

2.2. Further considerations

Increasing the trustworthiness. Since the outcome of a heuristic algorithm cannot entirely be trusted, it is natural to search for ways to at least increase the trustworthiness of the result. Depending on the kind of heuristics that were used, this can either be done by rerunning the same algorithm for different parameters, or by combining the algorithm with other algorithms.

For instance, in numerical analysis, a frequent trick is to rerun the same algorithm for different working precisions and compare the results. Similarly, probabilistic algorithms of Monte Carlo type can be rerun several times in order to increase our confidence that we obtained the correct result.

We also mentioned the fact that equality testing of exp-log constants is quite asymmetric in the sense that proving equalities is usually much harder than proving inequalities. This suggests the use of a mixed strategy for this problem, which combines a heuristic equality test with a deterministic, interval arithmetic based, inequality prover. In case of equality, we may optionally run Richardson's more expensive algorithm and try to prove the equality.

In applications, we notice that heuristic algorithms are often used in a similar way: to speed up the main algorithm, which can be perfectly deterministic. A nice example is the computation of a Gröbner basis over the rationals using modular arithmetic. In Faugère's software, modular arithmetic is used in order to get a precise idea about those S-polynomials that reduce to zero, thereby avoiding many unnecessary reductions when performing the actual computations over the rationals [Fau94].

In the particular case of a heuristic algorithm whose purpose is to guess dependencies, we finally notice that we may always build in the trivial safeguard to check all proposed relations *a posteriori*; see also remark 10 in section 6.1.

Attacks. It is a good practice to attack heuristic algorithms and try to construct systematic examples on which the heuristics fail. Successful attacks usually indicate that the algorithm was either wrong, or that the heuristics only make sense under certain additional assumptions. Failed attacks generally increase the confidence in the algorithm and sometimes lead into theoretical insight on why precisely the heuristics work well.

One particular angle of attack concerns the computational complexity or the numerical stability. This means that we search for examples for which the heuristic algorithm is very slow or numerically unstable. This line of thought underlies much of the numerical experiments which are presented in section 8.

Instead of attacking the heuristic algorithms, it is also possible to look for interesting classes of examples on which the heuristic provably works (or works better). In this paper, some interesting special classes of analytic functions are meromorphic, algebraic and so-called Fuchsian functions on a compact disk (see section 6.2). Typical classes which have to be considered for counterexamples are lacunary series, functions with a natural boundary, or holonomic functions that are not Fuchsian.

3. Computing the radius of convergence

Let f be an analytic function which is given by its power series $f_0 + f_1 z + f_2 z^2 + \cdots$ at the origin. A natural problem is to compute the radius of convergence $\rho = \rho_f$ of f. For sufficiently large classes of analytic functions, such as solutions to algebraic differential equations over the rationals, this problem is generally undecidable [DL89, Hoe07], although efficient and high quality algorithms for the computation of lower bounds for ρ do exist in this case [Hoe07]. In this section, we will consider heuristic algorithms, and only assume that a large number of coefficients $f_0, f_1, \ldots, f_{n-1}$ of f are known numerically.

3.1. Rough approximation using the Newton polygon

The first idea which comes into our mind is to apply Cauchy-Hadamard's formula for the radius of convergence ρ:

$$\rho^{-1} = \limsup_{i \to \infty} \sqrt[i]{|f_i|}.$$

Given $n \geqslant 2$ coefficients $f_0, \ldots, f_{n-1}$ of f, we may for instance use the approximation

$$\rho^{-1} \approx \max_{n/2 \leqslant i < n} \sqrt[i]{|f_i|}. \tag{4}$$

For most convergent power series f, this formula is ultimately exact in the sense that

$$\rho^{-1} = \lim_{n \to \infty} \max_{n/2 \leqslant i < n} \sqrt[i]{|f_i|}. \tag{5}$$

This is in particular the case when $\log |f_i|$ is ultimately convex or ultimately concave (this follows from the fact that the sequence $|f_i| / |f_{i+1}|$ converges in an ultimately monotone way to ρ, in this case). The set of f for which (5) holds is also stable under the transformation $f(z) \mapsto f(z^k)$ for any $k \in \mathbb{N}^> = \{n \in \mathbb{N}: n > 0\}$. Of course, we may replace $n/2$ by αn in (5) for any $\alpha \in (0, 1)$. Notice however that the lacunary power series $\sum_n z^{3^n}$ does not satisfy (5).

The formula (4) has the disadvantage that it has not been scaled appropriately: when replacing f by $c f$, where $c \in \mathbb{C}^{\neq}$ is such that $|c| \neq 1$, we obtain different approximations for ρ_{cf} and ρ_f. This drawback can be removed by replacing f_i by f_i / f_k for some appropriate coefficient index $0 \leqslant k < n$ with $f_k \neq 0$. In fact, one can even do better, and compute ρ using the formula $\rho = |f_l / f_k|^{1/(k-l)}$ for appropriate indices $0 \leqslant k < l < n$ with $f_k \neq 0$ and $f_l \neq 0$. Let us now show how to read off such indices from the numerical Newton diagram of f.

Let $P_i = (i, \log |f_i|)$, where we understand that $\log 0 = -\infty$. Then the *Newton diagram* of f is the convex hull of the half lines

$$P_i + \{0\} \times \mathbb{R}^{\leqslant} \;=\; \{(i, \log |f_i| + y) : y \leqslant 0\}.$$

For a fixed $0 < \alpha < 1$, say $\alpha = 1/2$, let $a = \lfloor \alpha n \rfloor$, and consider the Newton diagram of $F = f_a z^a + \cdots + f_{n-1} z^{n-1}$. There exists a minimal subset $\{P_{i_1}, \ldots, P_{i_k}\} \subseteq \{P_a, \ldots, P_{n-1}\}$ with $a = i_1 < \cdots < i_k = n-1$, such that the Newton diagram is also the convex hull of the half lines $P_{i_j} + \{0\} \times \mathbb{R}^{\leqslant}$ for $1 \leqslant j \leqslant k$. Graphically speaking, the P_{i_j} are the vertices of the Newton diagram (see figure 1).

For a fixed $\beta \in (\alpha, 1)$, say $\beta = 3/4$, we may now determine the unique edge $P_{i_j} P_{i_{j+1}}$ of the Newton diagram such that $i_j \leqslant \beta n < i_{j+1}$, and replace the formula (4) by

$$\rho^{-1} \approx \left| \frac{f_{i_{j+1}}}{f_{i_j}} \right|^{\frac{1}{i_{j+1} - i_j}}. \tag{6}$$

For a large class of convergent power series f, the formula (6) is again ultimately exact. The indices $i_1 < \cdots < i_k$ can be computed from the coefficients $f_a, \ldots, f_{n-1}$ using a linear traversal in time $O(n)$. This yields an efficient algorithm for the approximation of ρ which turns out to be more accurate than (4) in practice. The formula (6) has been implemented in the Mathemagix system [HLM+02]. For $n \approx 64$, the computed radius is usually correct up to one decimal digit in the relative error.

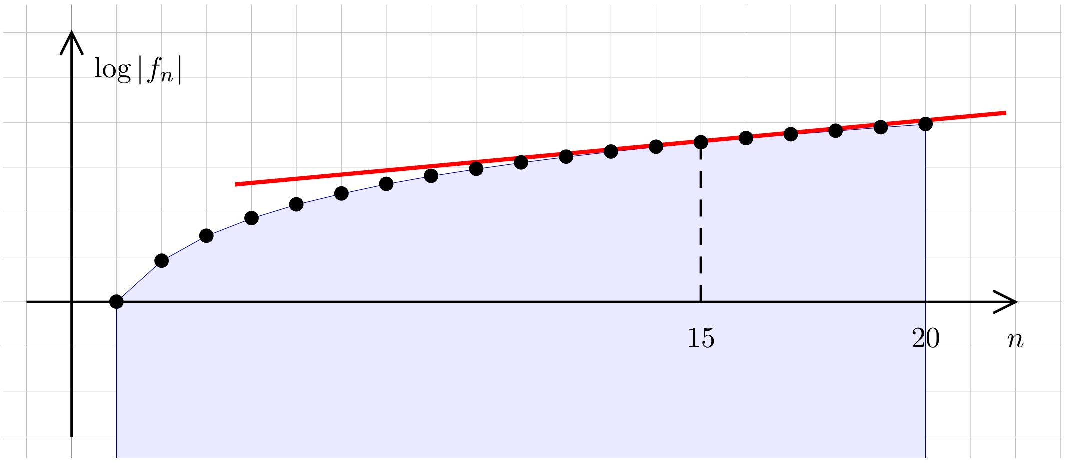

Figure 1. Illustration of the numerical Newton diagram for $f = (1 - z)^{-2} \log (1 - z)^{-1}$ and truncated at order $n = 20$. Taking $\beta = 3/4$, we get $i_j = 15$, $i_{j+1} = 16$ and $\rho^{-1} \approx |f_{16}/f_{15}| \approx 1.08875$. The red line is the unique line which extends the edge $P_{15} P_{16}$, of slope $\log |f_{16}/f_{15}| \approx 0.085$.

3.2. Asymptotic extrapolation

The formula (6) usually yields a reasonable estimate for ρ, even in very degenerate cases when there are several singularities at distance ρ or close to ρ. However, if f admits a single isolated singularity σ at distance ρ of the origin, with no other singularities at distance close to ρ, then it is often possible to read off much more information about this singularity from the coefficients $f_0, f_1, \ldots, f_{n-1}$.

For instance, it frequently (always?) occurs that the quotients f_{n-1}/f_{n-2} simply tend to σ^{-1} for $n \to \infty$. Moreover, as we will show below, if the singularity at σ has a known type, then the approximation $\sigma^{-1} \approx f_{n-1}/f_{n-2}$ can be further improved. If nothing is known about the singularity at σ, then we may still try and see what happens if we apply such specialized algorithms for various frequent types of singularities: with a bit of luck, we will both obtain useful information about the type of singularity and the numerical value of σ.

We say that f is *algebraic* at σ, if f satisfies a polynomial equation

$$P_d f^d + \cdots + P_0 = 0,$$

where $P_0, \ldots, P_d$ are analytic functions at σ with $P_d \neq 0$. In that case, we have

$$f_n \approx \sigma^{-n} n^{k/p} (a_0 + a_1 n^{-1/p} + a_2 n^{-2/p} + \cdots),$$

with $k \in \mathbb{Z}$ and ramification index $p \in \mathbb{N}^>$, whence

$$\log f_n \approx (-\log \sigma) n + \frac{k}{p} \log n + \log a_0 + \frac{a_1}{a_0} n^{-1/p} + \cdots. \tag{7}$$

Using the E-algorithm [Wen01, BZ91], we may now compute *simultaneous* approximations for the first coefficients $-\log \sigma$, k/p, $\log a_0$, a_1/a_0, etc. of the expansion (7). It turns out that this strategy greatly improves the accuracy of the approximation of σ (see also [Hoe09]).

Similarly, we say that f is *Fuchsian* at σ, if f satisfies a linear differential equation

$$L_d\, \delta^d f + \cdots + L_0\, f = 0,$$

where $\delta = (z - \sigma)\, \partial / \partial z$ and $L_0, \ldots, L_d$ are analytic functions at σ with $L_d(\sigma) \neq 0$. In that case, the Taylor coefficients f_n satisfy the asymptotic expansion

$$f_n \approx \sigma^{-n}\, n^{\lambda}\, (c_0(\log n) + c_1(\log n)\, n^{-1/p} + c_2(\log n)\, n^{-2/p} + \cdots),$$

where $\lambda \in \mathbb{C}$, $p \in \mathbb{N}^{>}$ and the c_i are polynomials in $\log n$ of degrees $< d$ [Fab85, Poi86, Bir13, Was67]. Again, the E-algorithm or more general algorithms for asymptotic extrapolation [Hoe09] can be used to compute σ with a high accuracy. Notice that these algorithms also provide estimates for the accuracies of the computed approximations.

In cases where nothing particular is known about the behaviour of f at its dominant singularity, the strategy of asymptotic extrapolation [Hoe09] still may work and both provide useful information about the nature of the singularity and the numerical value of σ.

4. LOCATING SINGULARITIES OF ANALYTIC FUNCTIONS

4.1. Analytic continuation

In order to explore the analytic function $f(z) = f_0 + f_1\, z + f_2\, z^2 + \cdots$ determined by the coefficients $f_0, f_1, f_2, \ldots$ more closely, and in particular determine its singularities in a given region, it is useful to have a means for performing the analytic continuation of f.

The way f is given sometimes provides us with such a means. For instance, if f is the solution to an initial value problem

$$\begin{aligned} f^{(r)}(z) &= P(z, f(z), \ldots, f^{(r-1)}(z)) \\ (f(0), \ldots, f^{(r-1)}(0)) &= I, \end{aligned}$$

then the continuation $f_{+u}(z) = f(u + z)$ of f at any sufficiently small u is again the solution of an initial value problem

$$\begin{aligned} f_{+u}^{(r)}(z) &= P\Big(z, f_{+u}(z), \ldots, f_{+u}^{(r-1)}(z)\Big) \\ (f_{+u}(0), \ldots, f_{+u}^{(r-1)}(0)) &= (f(u), \ldots, f^{(r-1)}(u)). \end{aligned}$$

In [Hoe05, Hoe07] it is shown in detail how to turn this into an algorithm for the analytic continuation of f. Notice also that the power series solutions to an initial value problem can be computed efficiently using the algorithms from [BK78, Hoe02].

In general, it is always possible to compute the continuation f_{+u} numerically. However, this kind of "poor man's" analytic continuation induces a big loss of accuracy. Let us illustrate this on the fundamental example

$$f = \frac{1}{1 - z}.$$

In order to compute $(f_{+u})_i$ with a precision of p bits using the formula

$$(f_{+u})_i = \sum_{j=0}^{\infty} \binom{i + j}{i} f_{i+j}\, u^j,$$

we need to truncate this expansion at an order j for which

$$\binom{i+j}{i}|u|^j \;\approx\; \left(\frac{1}{1-(z+u)}\right)_i 2^{-p} \;=\; (1-u)^{1+i}\,2^{-p}.$$

Putting $j=\alpha\,i$ and using Stirling's formula, this yields

$$(1+\alpha)\log(1+\alpha)-\alpha\log\alpha+\alpha\log u+\log(1-u) \;\approx\; -\frac{p}{i}\log 2. \tag{8}$$

For instance, in order to expand $f_{+1/2}$ at order n with precision $p=n$, we get

$$\begin{aligned}(1+\alpha)\log(1+\alpha)-\alpha\log\alpha-(1+\alpha)\log 2 \;&\approx\; -\log 2\\ \alpha \;&\approx\; 3.4034979,\end{aligned}$$

and we need an expansion of f at the origin at order $(\alpha+1)\,n$. In general, for every analytic continuation step, we need to multiply the expansion order by a constant factor which depends on the desired precision and the ratio between the step size and the radius of convergence.

An experimentally better strategy for poor man's analytic continuation is to postcompose f with a suitable analytic function $\phi(z)=\phi_1\,z+\phi_2\,z^2+\cdots$ instead of $\phi(z)=z+u$. If f admits a dominant singularity at σ, then ϕ should be chosen in such a way that $f\circ\phi$ has $\phi^{-1}(\sigma)$ as its only singularity in the unit disk, and such that $|\phi^{-1}(\sigma)|$ is as small as possible. One may typically take ϕ to be a composition of rational functions of the form $a\,z/(b+c\,z)$. Yet another strategy would be to consider a Padé-Hermite approximation $f\approx P/Q$ of f and then to expand P/Q (instead of f) at u.

4.2. Locating dominant singularities

Assuming that we have a means for analytic continuation (whether this means is efficient and accurate or only a "poor man's" solution), a natural question is to gather information about the singularities of f. First of all, we may want to locate the dominant singularity (or singularities) of f with a good precision.

Let us first assume that f admits a unique dominant singularity at σ and no other singularities of norm $<|\sigma|+\delta$. Section 3 provides us with heuristic algorithms for the computation of $|\sigma|$. More generally, for any $u\in\mathbb{C}$ with $|u|<\delta$, the point $\sigma-u$ is also the dominant singularity of f_{+u}, so we may compute $|\sigma-u|$ using the same heuristic algorithms. Computing $|\sigma-u|$ and $|\sigma-\mathrm{i}\,u|$ for two such points, the singularity σ is now uniquely determined by its distances $|\sigma|$, $|\sigma-u|$ and $|\sigma-\mathrm{i}\,u|$ to 0, u and $\mathrm{i}\,u$ (see figure 2).

Even if we do not know δ beforehand, then we apply the above method for u with $|u|=\varepsilon\,|\sigma|$, for some $\varepsilon<1$. In order to check the correctness of the computed value of σ *a posteriori*, we also compute the radius of convergence ρ of $f_{+\sigma/2}$. Then σ is a dominant singularity of f if and only if $\rho=\sigma/2$. If this check fails, then we simply divide ε by two and repeat the same procedure until the right value of σ is found.

In practice, the convergence radii can only be approximated, so all equality tests have to be replaced by approximate equality tests, for which the tolerance in relative precision is slightly larger than the estimated relative precision with which the convergence radii are computed. In particular, the above method computes σ with approximately the same accuracy as the radius of convergence $|\sigma|$.

Furthermore, if our procedure for analytic continuation is accurate enough, then we may "zoom in" on a singularity, and determine its location with higher precision. More precisely, assume that we have a rough approximation $\tilde{\sigma}$ of the dominant singularity σ. For some small $\varepsilon>0$, we then compute an approximation $\tilde{\tau}$ of the dominant singularity of $f_{+\lambda\sigma}$ with $\lambda=1-\varepsilon$. This yields the improved approximation $\lambda\,\tilde{\sigma}+\tilde{\tau}$ of σ.

Assume now that f admits more than one dominant singularity (or other singularities of norms close to $|\sigma|$), but still a finite number of them. On the one hand, by computing the radii of convergence of f_{+u} for a sufficiently large number of points u with $u<|\sigma|$, we may determine all dominant singularities. On the other hand, for any two adjacent dominant singularities σ and σ' with $|\sigma'-\sigma|<1$, we may check the absence of other singularities in between by looking at the radii of convergence of $f_{+\lambda(\sigma+2\sigma')/3}$ and $f_{+\lambda(2\sigma+\sigma')/3}$ for some $0<\lambda<1$. This yields a criterion for deciding whether we took a "sufficiently large number of points u".

From the numerical point of view, rough initial rough approximations can be further improved by zooming in. In figure 3 we have shown an example where we zoomed in on one among eight singularities. After zooming in, we see that all other singularities are far away, so they do not pollute the computation of the radius of convergence.

The generalized algorithm is particularly useful in the special case when there are two dominant singularities which are complex conjugates. This annoying situation prevents us from using the technique of asymptotic extrapolation in a direct way. Nevertheless, we may use it after zooming in on one of the two singularities.

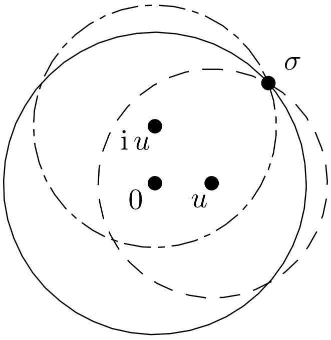

Figure 2. Determination of σ from its distances to 0, u and $\mathrm{i}\,u$.

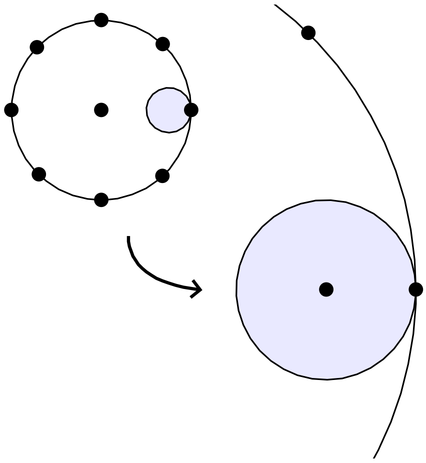

Figure 3. Zooming in on one among eight dominant singularities.

4.3. Exploring the Riemann surface

In the case when our method for analytic continuation allows us to turn around singularities, it is natural to ask whether we can explore the Riemann surface of f beyond the dominant singularities. In general, it is always possible to approximate the Riemann surface of f by an organically growing sequence of more and more precise Riemann surfaces of a special form; we refer to [Hoe07] for details. In this section, we will restrict ourselves to a more particular situation: given the closed disk $\bar{\mathcal{D}}_r$ of center 0 and radius r, we will assume that there exist a finite number of points $\sigma_1,\ldots,\sigma_s\in\mathcal{D}_r$ in its interior such that f is defined above $\mathcal{U}:=\bar{\mathcal{D}}_r\setminus\{\sigma_1,\ldots,\sigma_s\}$. Our task is to find the points $\sigma_1,\ldots,\sigma_s$.

One complication is that some of the singularities σ_i may not directly be visible from the origin by following a straight line, but only after turning around one or more other singularities. In principle, more and more singularities might therefore be disclosed by making successive turns around the singularities which were already found. In order to ensure our algorithm to terminate, we will make the additional assumption

H. There exists a finite dimensional vector space of analytic functions above $\mathcal{U}$, which contains f and all its analytic continuations.

This is in particular the case if

A. The Riemann surface of f admits a finite number of sheets above $\mathcal{U}$.

In sections 6.1 and 6.2, we will encounter two natural situations in which the assumptions **A** resp. **H** are satisfied.

Consider a finite number of points $z_1, ..., z_n \in \mathcal{U}$ together with a graph G which admits $V(G) = \{z_1, ..., z_n\}$ as its vertices. We will say that G is *admissible* if G is connected and for every edge $(z_i, z_j) \in E(G)$, the straightline segment between z_i and z_j lies in $\mathcal{U}$. Assume that f is known at one of the points z_i. For each $z_j \in V(G)$, we may then define V_j to be the vector space spanned by all analytic continuations of f by following edges of G. We may compute the collection $V_1, ..., V_n$ of these vector spaces using the following algorithm:

```
Algorithm continue($f_{+z_i}, z_i, G$)
Input: an analytic function $f$ at $z_i$ and an admissible graph $G$ with $V(G) = \{z_1, ..., z_n\}$
Output: for each $z_j \in V(G)$, a basis $B_j$ of $V_j$

Step 1. [Initialize]
    Set $B_i := \{f_{+z_i}\}$ and $B_j = \varnothing$ for all $j \neq i$

Step 2. [Saturate]
    For any $z_j \in V(G)$, $g \in B_j$ and $(z_j, z_k) \in E(G)$ do
        If $g_{+(z_k - z_j)} \notin \operatorname{span}(B_k)$, then
            Set $B_k := B_k \cup \{g_{+(z_k - z_j)}\}$ and repeat step 2

Step 3. [Terminate]
    Return $(B_1, ..., B_n)$
```

Remark 2. The test $g_{+(z_k - z_j)} \notin \operatorname{span}(B_k)$ requires an equality test for analytic functions. From the heuristic point of view, we may simply fix two numbers $k, p \in \mathbb{N}$ and test whether the first k Taylor coefficients coincide up to a relative error 2^{-p}.

Assume now that t singularities in $\{\sigma_1, ..., \sigma_s\}$ are known, say $\sigma_1, ..., \sigma_t$. We need a test in order to check whether $t = s$, as well as a way to find at least one of the remaining singularities in $\{\sigma_{t+1}, ..., \sigma_s\}$ if $t < s$.

Let $\varepsilon > 0$ be such that $\varepsilon < |\sigma_i - \sigma_j|/8$ for all $i < j \leqslant t$ and $|\sigma_i| < r - 8\,\varepsilon$. We now consider a graph G in the form of a hexagonal honeycomb, which fills up the disk $\bar{\mathcal{D}}_r$ and such that each edge $(z_i, z_j) \in E(G)$ has length $|z_j - z_i| = \varepsilon$ (see also figure 4 below). Picking G sufficiently at random, the graph G is also admissible, so we may apply the algorithm `continue`. For each vertex $z_i \in V(G)$, we also compute the minimum ρ_i of the convergence radii of the elements in B_i. We claim that $t = s$ if and only if

$$\rho_i \;=\; \min_{j \leqslant t} |z_i - \sigma_j| \tag{9}$$

for all i. Indeed, if $t = s$, then the equality clearly holds. Assume for contradiction that the equality holds for all i, even though $t < s$. For each of the outermost vertices $z_i \in V(G)$ of the honeycomb with $|z_i| \geqslant r - \varepsilon$, we have $|z_i - \sigma_{t+1}| \geqslant \rho_i \geqslant 6\,\varepsilon$. This implies that $|\sigma_{t+1}| \leqslant r - 5\,\varepsilon$, whence σ_{t+1} lies in the interior of one of the cells of the honeycomb. Let σ_j with $j \leqslant t$ be such that $|\sigma_j - \sigma_{t+1}|$ is minimal. Now consider a vertex $z_i \in V(G)$ of the cell of the honeycomb which contains σ_{t+1} such that $\operatorname{Re}(z_i - \sigma_j)/(z_i - \sigma_{t+1}) > 1$ (see figure 4). Then $|z_i - \sigma_j| > |z_i - \sigma_{t+1}|$. Now $|z_i - \sigma_j| \leqslant |z_i - \sigma_{t+1}| + |\sigma_j - \sigma_{t+1}| \leqslant 4\,\varepsilon$ also implies that $|z_i - \sigma_j| = \rho_i$. Consequently, $\rho_i > |z_i - \sigma_{t+1}| \geqslant \rho_i$. This contradiction completes the proof of our claim.

Whenever (9) does *not* hold for some i, then this indicates the presence of a singularity in $\{\sigma_{t+1}, ..., \sigma_s\}$ near z_i. It thus suffices to determine the dominant singularities of the elements in B_i in order to find at least one of these missing singularities.

Remark 3. In practice, we only have heuristic algorithms for approximating the radii ρ_i. Therefore, and as in the previous subsection, the test (9) should really be replaced by an approximate equality test.

Remark 4. We picked a graph in the form of a honeycomb in order to simplify the proof of our claim. In practice, it is more efficient to construct graphs with small edges near the singularities and larger edges elsewhere, such as Voronoï diagrams. Following [CC90, Hoe99], it can also be shown that the optimal shape of a polygon for turning around a singularity contains 17 edges and not 6 (at least in the case when f is the solution to a differentially algebraic equation).

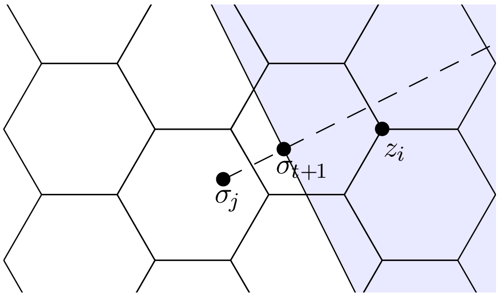

Figure 4. Illustration of the proof of our claim that $t=s$ if and only if (9) holds for all i. The colored region corresponds to the set of all points z with $\operatorname{Re}(z-\sigma_j)/(z-\sigma_{t+1})>1$.

5. Meromorphic functions

5.1. Computing the denominator on a compact disk

In the case when f is meromorphic on the compact disk $\bar{\mathcal{D}}_r$ of radius r, then there exists an alternative for the algorithms from section 4: we may directly search for a polynomial P such that the radius of convergence of $P\,f$ is strictly larger than r. If such a polynomial $\Delta_f=P$ exists, then we may select P to be monic and of minimal degree; this particular polynomial P will be called the *denominator* of f on $\bar{\mathcal{D}}_r$. If f is not meromorphic on $\bar{\mathcal{D}}_r$, then we define $\Delta_f=\bot$. Given $D\in\mathbb{N}$, we also define the *guarded denominator* $\Delta_{f,D}$ by $\Delta_{f,D}=\bot$ if $\Delta_f=\bot$ or $\deg\Delta_f>N$ and $\Delta_{f,N}=\Delta_f$ otherwise. Guarded denominators may be computed using simple linear algebra, as follows:

Algorithm `denom`(f,r,N,D)
INPUT: the first $N>2\,D$ coefficients of f, a radius r and a degree bound D
OUTPUT: an approximation of a monic polynomial P with $\rho_{Pf}>r$,
chosen of minimal degree $\deg P\leqslant D$, or $\bot$ (failed)

Step 1. [Initialize]
$d:=0$

Step 2. [Determine P]
Solve the linear system

$$\begin{pmatrix} f_{N-2d+1} & \cdots & f_{N-d} \\ \vdots & & \vdots \\ f_{N-d} & \cdots & f_{N-1} \end{pmatrix}\begin{pmatrix} P_{d-1} \\ \vdots \\ P_0 \end{pmatrix}+\begin{pmatrix} f_{N-2d} \\ \vdots \\ f_{N-d-1} \end{pmatrix}=0 \qquad (10)$$

Set $P:=z^d+P_{d-1}\,z^{d-1}+\cdots+P_0$

Step 3. [Terminate or loop]
Heuristically determine $\rho:=\rho_{Pf}$, based on the first N coefficients of $P f$

```
If $\rho > r$ then return $P$
If $d = D$ then return $\perp$
Set $d := d + 1$ and go to step 2
```

In order to study the reliability of this algorithm, let us first introduce some more notations. First of all, given integers $d \geqslant 0$ and $N \geqslant 2\,d - 1$, we will denote

$$H_{f,N,d} = \begin{pmatrix} f_{N-2d+1} & \cdots & f_{N-d} \\ \vdots & & \vdots \\ f_{N-d} & \cdots & f_{N-1} \end{pmatrix}.$$

The approximate correctness of the algorithm relies on the following technical lemma which will be proved in the next section.

LEMMA 5. *If f admits exactly d roots in the closed unit disk, when counted with multiplicities, then the matrix norm of $H_{f,N,d}^{-1}$ satisfies $\|H_{f,N,d}^{-1}\| = N^{O(1)}$ for $N \to \infty$.*

THEOREM 6. *Let $Q = \Delta_{f,D}$ and assume that the test $\rho_{Pf} > r$ in the algorithm always returns true if and only if $Q \neq \perp$ and $d \geqslant \deg Q$. Then $P = \perp$ if and only if $Q = \perp$. Moreover, if $P \neq \perp$, then*

$$P - Q = O(\rho^{-N})$$

for $N \to \infty$ and any $\rho < \rho_{Qf}$.

Proof. Modulo a scaling $z = r\,z'$, we may assume without loss of generality that $r = 1$. Furthermore, the proof is clear if $d = 0$, so we will assume that $d > 0$ and $\rho_f \leqslant 1$. The test $\rho_{Pf} > 1$ returns false as long as $d < \deg Q$ (or if $Q = \perp$). Assume now that d reaches $\deg Q$ in the algorithm. Then the computed P satisfies

$$\begin{pmatrix} f_{N-2d+1} & \cdots & f_{N-d} \\ \vdots & & \vdots \\ f_{N-d} & \cdots & f_{N-1} \end{pmatrix} \begin{pmatrix} P_{d-1} - Q_{d-1} \\ \vdots \\ P_0 - Q_0 \end{pmatrix} = -\begin{pmatrix} (Q\,f)_{N-d} \\ \vdots \\ (Q\,f)_{N-1} \end{pmatrix},$$

since

$$\begin{pmatrix} f_{N-2d+1} & \cdots & f_{N-d} \\ \vdots & & \vdots \\ f_{N-d} & \cdots & f_{N-1} \end{pmatrix} \begin{pmatrix} Q_{d-1} \\ \vdots \\ Q_0 \end{pmatrix} + \begin{pmatrix} f_{N-2d} \\ \vdots \\ f_{N-d-1} \end{pmatrix} = \begin{pmatrix} (Q\,f)_{N-d} \\ \vdots \\ (Q\,f)_{N-1} \end{pmatrix}.$$

Now let $\rho < \rho_{Qf}$. Given $\rho' \in (\rho, \rho_{Qf})$, we have $(Q\,f)_N = O((\rho')^{-N})$. Using lemma 5, it follows that $P - Q = O(\|H_{f,N,d}^{-1}\|\,(\rho')^{-N}) = O(N^{O(1)}\,(\rho')^{-N}) = O(\rho^{-N})$. □

Remark 7. The algorithm uses a very simple d-step search for finding a polynomial of minimal degree d with $\rho_{Pf} > r$. Using a binary search (doubling d at each step at a first stage, and using a dichotomic search at a second stage), the number of steps can be reduced to $O(\log d)$.

5.2. Proof of lemma 5

Proof. Let us first consider the case when $f = P/Q$ is a rational function with $\deg P < d := \deg Q$, $\gcd(P, Q) = 1$, and such that Q only admits roots in the closed unit disk. We denote $\Sigma = \{\sigma \in \mathbb{C} : Q(\sigma) = 0\}$ and $\nu_\sigma = \min_{i \in \mathbb{N}} Q^{(i)}(\sigma) \neq 0$ for each $\sigma \in \Sigma$. Using partial fraction decomposition, there exist numbers $c_{\sigma,i}$ with $i < \nu_\sigma$ such that

$$f_N = \sum_{\sigma \in \Sigma, i < \nu_\sigma} c_{\sigma,i}\,N^i\,\sigma^{-N}.$$

In particular, we may factor

$$H_{f,N,d} \;=\; M_N\,\Delta_N,$$

where M_N is a matrix with entries in $\mathbb{C}[N]$ and Δ_N is a diagonal matrix with entries σ^{-N} (and such that each $\sigma\in\Sigma$ occurs ν_σ times). The functions $N^i\,\sigma^{-N}$ with $i<\nu_\sigma$ being linearly independent, the matrix M_N is invertible in $\mathbb{C}(N)$, whence

$$\|H_{f,N,d}^{-1}\| \;=\; \|\Delta_N^{-1}\,M_N^{-1}\| = N^{O(1)}.$$

This completes the proof of the lemma in the special case.

In general, we may write $f=g+\varrho$, with $g=P/Q$ as above and where ϱ is an analytic function on the closed unit disk. Then

$$\begin{aligned}
\|H_{f,N,d}^{-1}\| &= \|(H_{g,N,d}+H_{\varrho,N,d})^{-1}\|\\
&= \|(H_{g,N,d}\,(1+H_{g,N,d}^{-1}\,H_{\varrho,N,d}))^{-1}\|\\
&\leqslant \|(1+H_{g,N,d}^{-1}\,H_{\varrho,N,d})^{-1}\|\;\|H_{g,N,d}^{-1}\|\\
&= (1+o(1))\,N^{O(1)}\\
&= N^{O(1)},
\end{aligned}$$

since

$$\|H_{g,N,d}^{-1}\,H_{\varrho,N,d}\| \;=\; N^{O(1)}\,r^{-N} \;=\; o(1)$$

for some $r>1$. This completes the proof in the general case. □

6. Detecting dependencies via analytic continuation

Keeping the same spirit as in section 5, we will now turn our attention to two larger classes of algebraic and Fuchsian analytic functions on the compact disk $\bar{\mathcal{D}}_r$ of radius r. Given a function f in one of these classes, we will show how to find the defining equation for f, under the assumption that we have a high accuracy algorithm for its analytic continuation. As in remark 2, we will also assume the existence of a heuristic equality test for analytic functions.

6.1. Algebraic dependencies on a compact disk

Let f be an analytic function which is given by its power series $f_0+f_1\,z+f_2\,z^2+\cdots$ at the origin. Assume that f can be continued analytically on a Riemann surface $\mathcal{R}$ above the closed disk $\bar{\mathcal{D}}_r$ of radius r minus a finite set of points $\sigma_1,\ldots,\sigma_s$. Let $\mathbb{A}$ be the set of analytic functions on $\bar{\mathcal{D}}_r$. We say that f is *algebraic* on $\bar{\mathcal{D}}_r$ if there exists a polynomial relation

$$P_d\,f^d+\cdots+P_0=0, \tag{11}$$

with $P\in\mathbb{A}[F]$. In that case, we may normalize the relation such that P has minimal degree and such that P_d is a monic polynomial of minimal degree. In particular, all roots of P_d are inside the disk $\bar{\mathcal{D}}_r$.

Given a function f which is algebraic on $\bar{\mathcal{D}}_r$, this function clearly satisfies the assumption **A** in section 4.3, whence we may compute the singularities $\sigma_1,\ldots,\sigma_s$ using the algorithm described there. Conversely, the example

$$f=\mathrm{e}^{\frac{1}{z-\sigma_1}}+\cdots+\mathrm{e}^{\frac{1}{z-\sigma_s}}$$

shows that there are functions which satisfy the assumption **A**, but which are not algebraic on the disk $\bar{\mathcal{D}}_r$.

Assume that f is algebraic on $\bar{\mathcal{D}}_r$. For each singularity σ_i, consider a path φ_i from the origin to a point near σ_i which avoids the other singularities, and let C_i be the operator which performs an analytic continuation along φ_i, one turn around σ_i, followed by an analytic continuation along φ_i^{-1}. With these operators, we may use the following algorithm for the detection of algebraic dependencies:

```
Algorithm alg_dep($f, r, (\sigma_1, ..., \sigma_s), N, D, B$)
Input: an analytic function $f$ above $\bar{\mathcal{D}}_r \setminus \{\sigma_1, ..., \sigma_s\}$ and bounds $N$, $D$ and $B$
Output: a normalized algebraic dependency (11) with $d \leqslant B$ and $\deg P_d \leqslant D$, or $\bot$ (failed)
Step 1. [Initialize]
    Set $\Phi := \{f\}$
Step 2. [Saturate]
    If $\operatorname{card}(\Phi) > B$ then return $\bot$
    If $C_i \Phi \subseteq \Phi$ for all $i$, then go to step 3
    $\Phi := \Phi \cup C_1 \Phi \cup \cdots \cup C_s \Phi$
    Repeat step 2
Step 3. [Terminate]
    Denote $\Phi := \{\varphi_1, ..., \varphi_k\}$
    Compute $Q := (F - \varphi_1) \cdots (F - \varphi_k) = F^k + Q_{k-1} F^{k-1} + \cdots + Q_0$
    For each $i \in \{0, ..., k\}$, compute $D_i :=$ denom$(Q_i, r, N, D)$
    If $D_i = \bot$ for some $i$, then return $\bot$
    $P := \operatorname{lcm}(D_0, ..., D_k)\, Q$
    If $\deg P_d > D$ then return $\bot$
    Return $P$
```

Theorem 8. *Modulo oracles for equality testing of analytic functions and the computation of guarded denominators and least common multiples, the algorithm* `alg_dep` *is correct.*

Proof. Assume that f satisfies a normalized relation (11), with $d = \deg P \leqslant B$ and $\deg P_d \leqslant D$. Since Φ only contains distinct roots of P, we have $\prod_{\varphi \in \Phi} (F - \varphi) \mid P$ in $\mathbb{C}((z))[F]$ throughout the algorithm. In particular $\operatorname{card}(\Phi) \leqslant d$, and we ultimately obtain stabilization $C_1 \Phi \cup \cdots \cup C_s \Phi \subseteq \Phi$.

At this point, analytic continuation around any of the points σ_i leaves the polynomial Q invariant, so the coefficients of Q are analytic and single-valued on $\bar{\mathcal{D}}_r \setminus \{\sigma_1, ..., \sigma_s\}$. On the other hand, given a singularity σ_i, each solution $\varphi \in \Phi$ is also given by a convergent Puiseux series near σ_i, whence so are the coefficients of Q. Since the only Puiseux series without monodromy around σ_i are Laurent series, it follows that the coefficients of Q are meromorphic on $\bar{\mathcal{D}}_r$.

By the minimality assumption on $\deg P$, it follows that $\deg Q = \deg P$ and $P = P_d\, Q$. Since each coefficient $Q_i = P_i / P_d$ is meromorphic on $\bar{\mathcal{D}}_r$, we may use the algorithm `denom` from the previous section in order to compute a polynomial D_i of minimal degree $\deg D_i \leqslant \deg P_d \leqslant D$ such that $Q_i\, D_i \in \mathbb{A}$. The monic least common multiple $\operatorname{lcm}(D_0, ..., D_k)$ is nothing but the monic polynomial P_d of minimal degree such that $P_d\, Q \in \mathbb{A}[F]$. □

Remark 9. In a world without oracles, the algorithm remains correct in the following sense: if the tests $C_i\, \Phi \subseteq \Phi$ and $D_i = \bot$ always return the right answer, then we will reach the line $P := \operatorname{lcm}(D_0, ..., D_k)\, Q$ whenever the desired relation exists. For the final normalization step, we need a numerical algorithm for the computation of least common multiples which allows for small errors in the input coefficients. This is similar to the computation of approximate g.c.d.s for which there exists an extensive literature; see for example [CGTW95, KL96], or [Zen04] and references therein.

Remark 10. As an additional safeguard, we may heuristically compute the radii of convergence of $P_0, \ldots, P_d$ and check whether they are indeed superior to r.

6.2. Fuchsian dependencies on a compact disk

We say that f is *Fuchsian* on $\bar{\mathcal{D}}_r$, if f satisfies a linear differential equation

$$L_d f^{(d)} + \cdots + L_0 f = 0, \tag{12}$$

where $L \in \mathbb{A}[\partial]$, and L is Fuchsian at each point in $\bar{\mathcal{D}}_r$. Again, we may normalize (12) such that L has minimal order d and such that L_d is a monic polynomial of minimal degree.

Given a function f which is Fuchsian on $\bar{\mathcal{D}}_r$, the function now satisfies the assumption **H** in section 4.3, so we may again compute the singularities $\sigma_1, \ldots, \sigma_s$ using the algorithm described there. With $C_1, \ldots, C_s$ as in the previous section, we may detect Fuchsian dependencies as follows:

```
Algorithm fuch_dep($f, r, (\sigma_1, \ldots, \sigma_s), N, D, B$)
INPUT: an analytic function $f$ above $\bar{\mathcal{D}}_r \setminus \{\sigma_1, \ldots, \sigma_s\}$ and bounds $N$, $D$ and $B$
OUTPUT: a normalized Fuchsian dependency (12) with $d \leqslant B$ and $\deg L_d \leqslant D$, or $\perp$ (failed)

Step 1. [Initialize]
    Set $\Phi := \{f\}$

Step 2. [Saturate]
    If card$(\Phi) > B$ then return $\perp$
    Let Vect $X$ denote the vector space generated by a finite set of vectors $X$
    If Vect$(C_i \Phi) \subseteq$ Vect$(\Phi)$ for all $i$, then go to step 3
    $\Phi := \Phi \cup \{C_i \varphi\}$ for $i$ and $\varphi \in \Phi$ with $C_i \varphi \notin$ Vect$(\Phi)$
    Repeat step 2

Step 3. [Terminate]
    Denote $\Phi := \{\varphi_1, \ldots, \varphi_k\}$
    Compute $K := \mathrm{lcm}(\partial - \varphi_1^{\dagger}, \ldots, \partial - \varphi_k^{\dagger}) = K_k \partial^k + \cdots + K_0$
        in the skew polynomial ring $\mathbb{C}((z))[\partial]$, where $\varphi^{\dagger}$ denotes $\varphi' / \varphi$
    For each $i \in \{0, \ldots, k\}$, compute $D_i :=$ denom$(K_i, r, N, D)$
    If $D_i = \perp$ for some $i$, then return $\perp$
    $L := \mathrm{lcm}(D_0, \ldots, D_k) K$
    If $\deg L_d > D$ then return $\perp$
    Return $L$
```

Theorem 11. *Modulo oracles for equality testing of analytic functions and the computation of guarded denominators and least common multiples, the algorithm* `fuch_dep` *is correct.*

Proof. Assume that f satisfies a normalized Fuchsian relation (12), with $d \leqslant B$ and $\deg\, L_d \leqslant D$. Throughout the algorithm, the set Φ only contains linearly independent solutions to $L\varphi = 0$. Therefore, the smallest operator $\mathrm{lcm}_{\varphi \in \Phi}(\partial - \varphi^{\dagger})$ which vanishes on Φ divides L in $\mathbb{C}((z))[\partial]$ on the right. In particular card$(\Phi) \leqslant d$, and we ultimately obtain stabilization $C_1 \Phi + \cdots + C_s \Phi \subseteq \Phi$.

Consider one of the singularities σ_i. Since f is Fuchsian at σ_i, the equation $L f = 0$ admits a fundamental system of solutions of the form

$$\varphi(\sigma_i + u) = \left(\varphi_{d-1}(u^{1/p}) \, (\log u)^{d-1} + \cdots + \varphi_0(u^{1/p}) \right) u^{\lambda},$$

where $\lambda \in \mathbb{C}$ and $\varphi_0, \ldots, \varphi_{d-1}$ are convergent power series. The coefficients of K lie in the field $\mathbb{K}$ generated by all generalized power series of this form. Again, the only elements of $\mathbb{K}$ with a trivial monodromy around σ_i are convergent Laurent series at σ_i. Since analytic continuation around σ_i leaves the operator K unchanged, it follows that the coefficients of K are meromorphic on $\bar{\mathcal{D}}_r$. We conclude in a similar way as in the proof of theorem 8. □

Remark 12. The hypothesis that f admits at worse a Fuchsian singularity at σ_i is essential for the algorithm to work. For instance, in the case of the function

$$f = \mathrm{e}^{\frac{1}{z-\sigma}} + \mathrm{e}^{-\frac{1}{z-\sigma}},$$

the monodromy around σ is trivial, whence applying the algorithm for $r > |\sigma|$ would simply result in having $\Phi = \{f\}$ at step 3. Even though $f^\dagger$ has no monodromy around σ, this function is no longer a Laurent series. In fact, the desired vanishing operator has second order in this case, but it cannot be read off directly from Φ.

7. Detecting dependencies via orthogonalization

In the case when we have no algorithm or no efficient algorithm for the analytic continuation of f, the algorithms of section 6 can no longer be used. In this section, we will describe a purely numerical approach for the detection of analytic relations on a closed disk of radius R. Modulo the change of variables $z \mapsto R\,z$, it suffices to consider the case when $R = 1$.

Given an order $n \in \mathbb{N}$, we will denote by $\mathbb{C}[[z]]_{;n}$ the set of power series $f \in \mathbb{C}[[z]]$ with $f_i = 0$ for all $i \geqslant n$. When truncating the usual multiplication on $\mathbb{C}[[z]]$ at order n, we give $\mathbb{C}[[z]]_{;n}$ a ring structure, which is isomorphic to $\mathbb{C}[z]/(z^n)$. We will denote by $\mathbb{C}[[z]]_{;\infty}$ the Hilbert space of all power series $f \in \mathbb{C}[[z]]$ with finite $\mathcal{L}_2$ norm

$$\|f\| = \sqrt{|f_0|^2 + |f_1|^2 + \cdots}.$$

Notice that any power series that converges on the closed unit disk has finite norm, but a power series with finite norm is only guaranteed to converge on the open unit disk. More generally, the norm of a vector $f = (f_1, \ldots, f_d) \in \mathbb{C}[[z]]_{;\infty}^d$ is given by

$$\|f\| = \sqrt{\|f_1\|^2 + \cdots + \|f_d\|^2}.$$

The spaces $\mathbb{C}[[z]]_{;0} \subseteq \mathbb{C}[[z]]_{;1} \subseteq \cdots$ can be considered as an increasing sequence of Hilbert spaces, for the restrictions of the norm on $\mathbb{C}[[z]]_{;\infty}$ to the $\mathbb{C}[[z]]_{;n}$.

Let $0 < r < 1$ and assume that $f_1, \ldots, f_d \in \mathbb{C}[[z]]$ are power series with radii of convergence at least r (notice that we do *not* assume the f_i to be of finite norm). We want to solve the equation

$$g_1\,f_1 + \cdots + g_d\,f_d = h \qquad (g_1, \ldots, g_d, h \in \mathbb{C}[[z]]_{;\infty}). \tag{13}$$

We will denote the vector space of all such relations $(\varphi_1, \ldots, \varphi_{d+1}) = (g_1, \ldots, g_d, h) \in \mathbb{C}[[z]]_{;\infty}^{d+1}$ by $\mathbf{\Phi}_{;\infty}$. Since the equation (13) involves an infinite number of coefficients, we need to consider its truncated version at a finite order $n \in \mathbb{N}$. Replacing $f_1, \ldots, f_d$ by their truncations in $\mathbb{C}[[z]]_{;n}$, we will search for non-trivial solutions of the equation

$$g_1\,f_1 + \cdots + g_d\,f_d = h \qquad (g_1, \ldots, g_d, h \in \mathbb{C}[[z]]_{;n}). \tag{14}$$

In a way which will be made more precise below, we will require the norms of $g = (g_1, \ldots, g_d)$ and h to remain of the same orders of magnitude as the vector $(g_{1,0}, \ldots, g_{d,0})$. We will denote by $\mathbf{\Phi}_{;n}$ the vector space of all $(\varphi_1, \ldots, \varphi_{d+1}) = (g_1, \ldots, g_d, h) \in \mathbb{C}[[z]]_{;n}^d$ which satisfy (14). If the norms of the solutions to the truncated problems tend to a limit for $n \to \infty$, then we will prove that these solutions tend to a solution of the original problem (13).

Let us reformulate the truncated problem in terms of linear algebra. The series $f_1, \ldots, f_d$ give rise to an $n\,d \times (n + n\,d)$ matrix

$$M = \begin{pmatrix} f_{1,0} & & & & & 1 & & & & & & & & & & \\ \vdots & & & & & & \ddots & & & & & & & & & \\ f_{d,0} & & & & & & & 1 & & & & & & & & \\ f_{1,1} & f_{1,0} & & & & & & & 1 & & & & & & & \\ \vdots & \vdots & & & & & & & & \ddots & & & & & & \\ f_{d,1} & f_{d,0} & & & & & & & & & 1 & & & & & \\ \vdots & \vdots & \ddots & & & & & & & & & \ddots & & & & \\ \vdots & \vdots & & \ddots & & & & & & & & & \ddots & & & \\ f_{1,n-1} & f_{1,n-2} & \cdots & \cdots & f_{1,0} & & & & & & & & & 1 & & \\ \vdots & \vdots & & & \vdots & & & & & & & & & & \ddots & \\ f_{d,n-1} & f_{d,n-2} & \cdots & \cdots & f_{d,0} & & & & & & & & & & & 1 \end{pmatrix}.$$

The unknown series $g_1, \ldots, g_d$ give rise to a row vector

$$G = \begin{pmatrix} g_{1,n-1} & \cdots & g_{d,n-1} & \cdots & \cdots & g_{1,0} & \cdots & g_{d,0} \end{pmatrix}.$$

Setting $h = g_1\,f_1 + \cdots + g_d\,f_d$, we then have

$$G\,M = \begin{pmatrix} h_{n-1} & \cdots & h_0 & G \end{pmatrix}.$$

Putting the first n entries on the right hand side and grouping by packets of n entries, it follows that $\Phi = G\,M$ encodes the relation $(\varphi_1, \ldots, \varphi_{d+1}) = (g_1, \ldots, g_d, h)$. This reduces the problem to finding those vectors G for which $\|G\,M\|$ is of the same order of magnitude as the vector $(g_{1,0}, \ldots, g_{d,0})$.

We start with the computation of a thin LQ decomposition of M. This can for instance be done using the Gram-Schmidt process: starting with the first row, we orthogonally project each row on the vector space spanned by the previous rows. This results in a decomposition

$$M = L\,Q,$$

where L is a lower triangular $n\ d \times n\ d$ matrix with ones on the diagonal and Q is an $n\,d \times (n + n\,d)$ matrix, whose rows are mutually orthogonal (i.e. $Q\,Q^* = \mathrm{Id}$). Now consider the matrix Λ formed by the d last rows of L^{-1}. Then each row $G = \Lambda_i$ gives rise to a relation (14), encoded by $\Phi = \Lambda_i\ M$. Moreover, this relation is *normal* or *i-normal*, in the sense that $g_{i,0} = 1$ and $g_{j,0} = 0$ for all $j > i$. Since $\Lambda_i\ M$ is the shortest vector in $(\Lambda_i + \mathrm{Vect}(\Lambda_0, \ldots, \Lambda_{i-1}))\ M$, the relation is also minimal in norm, among all i-normal relations. Choosing the row Λ_i for which $\|\Lambda_i\,M\|$ is minimal, our algorithm simply returns the corresponding relation. Then our algorithm has the following fundamental property:

PROPOSITION 13. *The algorithm returns a normal relation g for which $\|G\,M\|$ is minimal.*

Let us now return to the case when $f_1, \ldots, f_d \in \mathbb{C}[[z]]$ are no longer truncated, but all have a radius of convergence $\geqslant r$. A relation (13) is again said to be *normal* or *i-normal* if $g_{i,0} = 1$ and $g_{j,0} = 0$ for all $j > i$. Under the limit $n \to \infty$, we claim that our algorithm finds a minimal normal relation, if there exists a relation of the form (13):

THEOREM 14.

a) Assume that $\mathbf{\Phi}_{;\infty} \neq 0$. Then $\mathbf{\Phi}_{;\infty}$ contains a minimal normal relation. For each i, $\mathbf{\Phi}_{;\infty}$ contains at most one minimal i-normal relation.

b) *Assume that $\varphi \in \mathbf{\Phi}_{;\infty}$ is a minimal i-normal relation. For each $n \in \mathbb{N}$, let $f_{;n}$ be the truncation of f at order n and consider the corresponding minimal i-normal relation $\varphi_{;n} \in \mathbf{\Phi}_{;n}$. Then the relations $\varphi_{;n}$ converge to φ in $\mathbb{C}[[z]]_{;\infty}^{d+1}$.*

c) *If $\mathbf{\Phi}_{;\infty} = \{0\}$, then the norms $\|\varphi_{;n}\|$, with $\varphi_{;n}$ as in (b), are unbounded for $n \to \infty$.*

Proof. A non trivial relation $\varphi = (g, h)$ is easily normalized: we first divide g by z^v, where $v = \min \{\operatorname{val} g_1, ..., \operatorname{val} g_d\}$. We next divide by $g_{i,0}$, where i is largest with $g_{i,0} \neq 0$. Now the set of all i-normal relations is a closed affine subspace of $\mathbb{C}[[z]]_{;\infty}^d$. The orthogonal projection of 0 on this subspace yields a unique i-normal relation of minimal norm. This proves (a).

Assume that there exists an i-normal relation (13), and let $\varphi \in \mathbf{\Phi}_{;\infty}$ be the minimal such relation. Given an order $n \in \mathbb{N}$, consider the minimal i-normal relation $\varphi_{;n}$ at this order. Truncation of this relation at a smaller order $m < n$ yields an i-normal relation $\varphi_{;n;m}$ at order m with $\varphi_{;n;m} \perp (\varphi_{;n} - \varphi_{;n;m})$, whence

$$\|\varphi_{;n}\|^2 = \|\varphi_{;n;m}\|^2 + \|\varphi_{;n} - \varphi_{;n;m}\|^2. \tag{15}$$

Moreover, since $\varphi_{;m}$ is the projection of 0 on the affine space of i-normal relations at order m, we have $\varphi_{;m} \perp (\varphi_{;n;m} - \varphi_{;m})$ and

$$\|\varphi_{;n;m}\|^2 = \|\varphi_{;n;m} - \varphi_{;m}\|^2 + \|\varphi_{;m}\|^2. \tag{16}$$

Similarly, truncation of the relation φ at finite order n yields an i-normal relation $\varphi_{;\infty;n}$ which satisfies

$$\|\varphi\|^2 = \|\varphi_{;\infty;n}\|^2 + \|\varphi - \varphi_{;n}\|^2.$$

In particular, $\|\varphi_{;m}\| \leqslant \|\varphi_{;n;m}\| \leqslant \|\varphi_{;n}\| \leqslant \|\varphi\|$, so that

$$\|\varphi_{;0}\| \leqslant \|\varphi_{;1}\| \leqslant \cdots \leqslant \|\varphi\|.$$

For $n \geqslant m \to \infty$, it follows that $\|\varphi_{;n}\| - \|\varphi_{;m}\| \to 0$, whence

$$\|\varphi_{;n} - \varphi_{;n;m}\|^2 + \|\varphi_{;n;m} - \varphi_{;m}\|^2 = \|\varphi_{;n}\|^2 - \|\varphi_{;m}\|^2 \to 0,$$

and

$$\|\varphi_{;n} - \varphi_{;m}\| \leqslant \|\varphi_{;n} - \varphi_{;n;m}\| + \|\varphi_{;n;m} - \varphi_{;m}\| \to 0.$$

In other words, $\varphi_{;n}$ is a Cauchy sequence, which converges to a limit $\tilde{\varphi} \in \mathbb{C}[[z]]_{;\infty}^d$ with $\|\tilde{\varphi}\| \leqslant \|\varphi\|$. Now for each n, we may write $\varphi_{;n} = (g_{;n,1}, ..., g_{;n,d}, h_{;n})$ and denote by $\psi_{;n}$ the truncation of the vector $\psi = (f_1, ..., f_d, -1)$ at order n. By assumption, the inner products $\varphi \cdot \psi$ and $\varphi_{;n} \cdot \psi_{;n}$ vanish. Since $\|\psi_i - (\psi_{;n})_i\| \to 0$ for all i and $n \to \infty$, we also have $\|\varphi_{;n} \cdot \psi\| \to 0$ for $n \to \infty$, whence $\tilde{\varphi} \cdot \psi = 0$. Consequently, $\tilde{\varphi}$ is an i-normal relation in $\mathbf{\Phi}_{;\infty}$. By the minimality hypothesis on φ, we conclude that $\tilde{\varphi} = \varphi$, which proves (b).

In general, the existence of a bound B with

$$\|\varphi_{;0}\| \leqslant \|\varphi_{;1}\| \leqslant \cdots \leqslant B$$

still ensures $\varphi_{;n}$ to be a Cauchy sequence, and its limit yields an i-normal relation (13). This proves the last assertion (c). □

Remark 15. For some very specific types of singularities, we would like to mention the existence of an alternative strategy for the determination of linear dependencies. This strategy relies on asymptotic extrapolation [Hoe09] and does not involve analytic continuations around singularities, as in section 6. Let us illustrate the method on the example of a function f with an isolated smallest singularity at 1 of the form

$$f(z) = a(z) + b(z) \log \frac{1}{1 - z},$$

where a and b are analytic at 1. Then the asymptotic extrapolation of $f = f_0 + f_1 z + \cdots$ yields an asymptotic expansion of the form

$$f_n \approx \frac{c_1}{n} + \frac{c_2}{n^2} + \cdots.$$

Using singularity analysis [FS96], we may then recover the function b from the coefficients $c_1, c_2, \ldots$. However, this technique only works in special cases, since the first ω terms of the asymptotic expansion may hide other terms, which need to be taken into account when searching for exact dependencies.

8. NUMERICAL EXPERIMENTS

A first implementation of the guessing algorithm from section 7 has been made in the MATHEMAGIX system [HLM+02]. It is instructive to test the algorithm on examples for which the existence or absence of linear dependencies is known. In particular, in order to turn theorem 14 into an algorithm which requires no manual tweaking, we need more precise information about the corresponding rates of convergence and divergence of the norms $\|\varphi_{;n}\|$.

8.1. Examples of recognized relations

We have tested the orthogonalization algorithm on three examples of increasing difficulty, for which linear dependencies are known to exist. In order to avoid problems due to numerical instability, we have used a large working precision, of 1024 or 2048 bits.

Single poles. The easiest example on which we can run the algorithm is

$$f_1 = \frac{1}{1 - \lambda z} \qquad (\lambda > 1).$$

Here follow the values for $g_{;n}$ at different orders and $\lambda = 2$:

$$\begin{aligned}
g_{;4,1} &= 1.0000 - 1.6000\, z - 0.60000\, z^2 - 0.20000\, z^3\\
g_{;16,1} &= 1.0000 - 1.6180\, z - 0.61803\, z^2 - \cdots - 5.8340\, 10^{-6}\, z^{14} - 1.9447\, 10^{-6}\, z^{15}\\
g_{;64,1} &= 1.0000 - 1.6180\, z - 0.61803\, z^2 - \cdots - 5.0484\, 10^{-26}\, z^{62} - 1.6828\, 10^{-26}\, z^{63}\\
g_{;256,1} &= 1.0000 - 1.6180\, z - 0.61803\, z^2 - \cdots - 2.8307\, 10^{-106}\, z^{254} - 9.4357\, 10^{-107}\, z^{255}
\end{aligned}$$

The $g_{;n,1}$ clearly converge to a limit g_1 with radius of convergence >1. It should be noticed that we do not have $g_1 = 1 - \lambda z$. In other words, the “best” numeric relation (from the point of view of $\mathcal{L}_2$-norms) does not coincide with the “best” algebraic relation (as found by using Padé-Hermite approximation). A closer examination of the result shows that

$$\begin{aligned}
g_1 &= \frac{1 - \lambda z}{1 - \alpha z}\\
\alpha + \frac{1}{\alpha} &= \lambda + \frac{2}{\lambda} \qquad (\alpha < 1)
\end{aligned}$$

In particular, α decreases if λ increases.

Logarithmic singularity. Our second example concerns a logarithmic singularity:

$$\begin{aligned}
f_1 &= \cos(z) \log(1 - 2z)\\
f_2 &= f_1'
\end{aligned}$$

We obtain:

$$\begin{aligned}
g_{;32,1} &= 0.070136+0.66832\,z-1.1491\,z^2-\cdots-0.010473\,z^{29}+0.0033638\,z^{30}\\
g_{;64,1} &= 0.013789+0.88839\,z-1.3375\,z^2-\ \cdots\ -0.0083494\,z^{61}+0.0012131\,z^{62}\\
g_{;128,1} &= 0.000138+0.99781\,z-1.6176\,z^2-\quad\cdots\quad -2.3908\cdot 10^{-4}\,z^{125}+1.6114\cdot 10^{-5}\,z^{126}\\
g_{;256,1} &= 4.7\cdot 10^{-9}+1.00000\,z-1.6204\,z^2-0.24832\,z^3-0.75949\,z^4-0.31293\,z^5-\cdots\\
&\quad -9.8386\cdot 10^{-5}\,z^{128}-9.5406\cdot 10^{-5}\,z^{129}-9.3630\cdot 10^{-5}\,z^{130}-9.3108\cdot 10^{-5}\,z^{131}-\cdots\\
&\quad -2.3728\cdot 10^{-6}\,z^{251}+2.7475\cdot 10^{-7}\,z^{252}-1.9342\cdot 10^{-8}\,z^{253}+6.2727\cdot 10^{-10}\,z^{254}\\
g_{;32,2} &= 1.0000-1.6886\,z-0.35145\,z^2-\cdots-0.0071095\,z^{30}+0.0033638\,z^{31}\\
g_{;64,2} &= 1.0000-1.6782\,z-0.42492\,z^2-\ \cdots\ -0.0071362\,z^{62}+0.0012131\,z^{63}\\
g_{;128,2} &= 1.0000-1.6326\,z-0.54349\,z^2-\quad\cdots\quad -2.2297\cdot 10^{-4}\,z^{126}+1.6114\cdot 10^{-5}\,z^{127}\\
g_{;256,2} &= 1.0000-1.6204\,z-0.58164\,z^2-0.21949\,z^3-0.25182\,z^4-0.016358\,z^5-\cdots\\
&\quad -1.0928\cdot 10^{-4}\,z^{128}-9.8597\cdot 10^{-5}\,z^{129}-8.6828\cdot 10^{-5}\,z^{130}-7.3899\cdot 10^{-5}\,z^{131}-\cdots\\
&\quad 2.1091\cdot 10^{-6}\,z^{252}+2.5583\cdot 10^{-7}\,z^{253}-1.8714\cdot 10^{-8}\,z^{254}+6.2727\cdot 10^{-10}\,z^{255}
\end{aligned}$$

The convergence only becomes apparent at higher orders. Contrary to the previous example, it seems that the series in the computed relation all have a radius of convergence 1. The norms of the computed results are given by $\|\varphi_{;32}\| = 3.5577$, $\|\varphi_{;64}\| = 3.5955$, $\|\varphi_{;128}\| = 3.6739$ and $\|\varphi_{;256}\| = 3.6862$.

Number of alcohols. A more interesting example from combinatorics is the enumeration of the number of alcohols of the form $C_nH_{2n+1}OH$ [Pól37]. Its generating series satisfies the functional equation

$$s(z)=1+z\,\frac{s(z)^3+2\,s(z^3)}{3}.$$

The asymptotic behaviour of the coefficients s_n is determined by the asymptotic behaviour of $s(z)$ at its singularity of smallest norm. Since $s_n \geqslant 0$ for all n, this so-called dominant singularity necessarily lies on the positive real axis, and coincides with the radius of convergence r of s. Using asymptotic extrapolation [Hoe09], we find that $r \approx 0.304218409$. In order to investigate $s(z)$ for z close to r, we apply our algorithm to

$$\begin{aligned}
f_1 &= s(0.15\,(z+0.25))\\
f_2 &= f_1'.
\end{aligned}$$

The translation $z \mapsto z+0.25$ is done using power series evaluation until stabilization at the working precision. Although this was most straightforward to implement for this particular example, there exist better strategies for zooming in, as mentioned at the end of section 4.1. At different orders, we obtain:

$$\begin{aligned}
g_{;32,1} &= 1.6836-0.2908\,z+0.0013\,z^2-\cdots+0.022645\,z^{30}-0.0029284\,z^{31}\\
g_{;64,1} &= 1.7104-0.3645\,z+0.2378\,z^2-\ \cdots\ +1.6968\cdot 10^{-4}\,z^{62}-1.0104\cdot 10^{-5}\,z^{63}\\
g_{;96,1} &= 1.7105-0.3623\,z+0.2372\,z^2-\quad\cdots\quad +9.4113\cdot 10^{-7}\,z^{94}-3.6428\cdot 10^{-8}\,z^{95}\\
g_{;128,1} &= 1.7105-0.3622\,z+0.2370\,z^2-0.052535\,z^3+0.033518\,z^4-\cdots\\
&\quad +6.3727\cdot 10^{-5}\,z^{96}-3.4222\cdot 10^{-5}\,z^{97}-1.2120\cdot 10^{-4}\,z^{98}-1.5462\cdot 10^{-4}\,z^{99}-\cdots\\
&\quad +7.2542\cdot 10^{-7}\,z^{124}-7.4945\cdot 10^{-8}\,z^{125}+4.6860\cdot 10^{-9}\,z^{126}-1.3437\cdot 10^{-10}\,z^{127}\\
g_{;32,2} &= 1.0000-2.4006\,z-0.8867\,z^2-\cdots+0.0061880\,z^{30}-9.9191\cdot 10^{-4}\,z^{31}\\
g_{;64,2} &= 1.0000-2.3257\,z-1.2027\,z^2-\ \cdots\ +5.2359\cdot 10^{-5}\,z^{62}-3.4225\cdot 10^{-6}\,z^{63}\\
g_{;96,2} &= 1.0000-2.3236\,z-1.2135\,z^2-\quad\cdots\quad +3.0034\cdot 10^{-7}\,z^{94}-1.2339\cdot 10^{-8}\,z^{95}\\
g_{;128,2} &= 1.0000-2.3235\,z-1.2138\,z^2-0.0067226\,z^3-0.080434\,z^4+\cdots\\
&\quad -2.6028\cdot 10^{-5}\,z^{96}-3.4490\cdot 10^{-5}\,z^{97}-2.9260\cdot 10^{-5}\,z^{98}-1.1090\cdot 10^{-5}\,z^{99}+\cdots\\
&\quad +2.1258\cdot 10^{-7}\,z^{124}-2.3161\cdot 10^{-8}\,z^{125}+1.5192\cdot 10^{-9}\,z^{126}-4.5516\cdot 10^{-11}\,z^{127}
\end{aligned}$$

The corresponding norms are given by $\|\varphi_{;32}\| = 4.3276$, $\|\varphi_{;64}\| = 4.3845$, $\|\varphi_{;96}\| = 4.3863$ and $\|\varphi_{;128}\| = 4.3863$. Again, the convergence is rather slow, and the computed series all seem to have radius of convergence 1. Since $1 > (r - 0.25)/0.15$, this means that we indeed guessed an analytic relation between f_1 and f_2 on the unit disk.

Remark 16. The last two examples show that our algorithm usually returns a relation whose series all have radius of convergence 1, if such a relation indeed exists. Larger radii of convergence R can be obtained simply by running the algorithm on $f(R\,z)$ and scaling back the result.

8.2. Rate of divergence in case of independence

It is also instructive to run the algorithm on examples where the f_i are known to be analytically independent. In that case, it is important to examine the behaviour of the norms $\|\varphi_{;n}\|$ for $n \to \infty$ and find more precise criteria which will enable us to discard the existence of a nice dependency with a reasonable degree of certainty.

Logarithmic singularity. Let us return to the case of a simple logarithmic singularity

$$f_1 = \log(1 - \lambda\, z),$$

where $\lambda > 1$. Running our algorithm directly on this function yields the following results for $\lambda = 2$:

$$\begin{aligned}
g_{;32,1} &= 1.0000 - 4.2642\,z + 4.3396\,z^2 - \cdots + 1.3621\,z^{29} - 0.68197\,z^{30}\\
g_{;64,1} &= 1.0000 - 8.1566\,z + 23.882\,z^2 - \cdots + 4.1515\,z^{61} - 0.67813\,z^{62}\\
g_{;128,1} &= 1.0000 - 16.183\,z + 112.02\,z^2 - \cdots + 9.2483\,z^{125} - 0.65589\,z^{126}\\
g_{;256,1} &= 1.0000 - 32.213\,z + 480.89\,z^2 - \cdots + 19.957\,z^{253} - 0.66367\,z^{254}
\end{aligned}$$

The results indeed do not converge and the corresponding sequence of norms $\|\varphi_{;32}\| = 12.324$, $\|\varphi_{;64}\| = 77.306$, $\|\varphi_{;128}\| = 3383.9$ and $\|\varphi_{;256}\| = 6.4461 \cdot 10^6$ diverges at moderate exponential speed.

Various singularities. In the above example, it is interesting to study the rate of divergence of the sequence of norms for other values of λ. More generally, we can consider various types of singularities, such as

$$\begin{aligned}
\psi_{1,\lambda} &= \log(1 - \lambda\, z)\\
\psi_{2,\lambda} &= \sqrt{1 - \lambda\, z}\\
\psi_{3,\lambda} &= \mathrm{e}^{\frac{\lambda z}{1 - \lambda z}}\\
\psi_{4,\lambda} &= \mathrm{random}(\lambda\, z).
\end{aligned}$$

The series $\mathrm{random}(z) = \rho_0 + \rho_1\, z + \rho_2\, z^2 + \cdots$ is a series whose coefficients are chosen according to a random uniform distribution on $[0, 1]$. The results are shown in table 1 below. For $\psi_{1,\lambda}$, $\psi_{2,\lambda}$ and $\psi_{3,\lambda}$, it seems that the norm does not much depend on the precise type of singularity, but only on λ and the truncation order n. For the last series $\psi_{4,\lambda}$, the dependencies on λ and n approximately seem to follow the law

$$\log \|\varphi_{;n}\| \approx \frac{n}{2} \log \lambda. \tag{17}$$

This idealized law needs to be adjusted for functions $\psi_{1,\lambda}$, $\psi_{2,\lambda}$, $\psi_{3,\lambda}$ with more common types of singularities. Although (17) remains asymptotically valid for large values of $\log \lambda$, the factor $\log\,\lambda$ needs to be replaced by a smaller number for moderate values. For $\sqrt{2} \leqslant \lambda \leqslant 4$, this factor rather seems to be of the form $\alpha\ (\log\ \lambda)^2$, where the constant α depends on the nature of the singularity. Also, the linear dependency of $\log \|\varphi_{;n}\|$ on n is only reached for large values of n.

	32	64	128	256
$\psi_{1,\sqrt{2}}$	2.5897	4.2958	$1.1107\cdot 10^{1}$	$7.5308\cdot 10^{1}$
$\psi_{1,2}$	$1.2324\cdot 10^{1}$	$7.7306\cdot 10^{1}$	$3.3839\cdot 10^{3}$	$6.4461\cdot 10^{6}$
$\psi_{1,4}$	$2.7074\cdot 10^{3}$	$3.1503\cdot 10^{6}$	$5.0806\cdot 10^{12}$	$1.3215\cdot 10^{25}$
$\psi_{1,8}$	$5.7101\cdot 10^{6}$	$1.6078\cdot 10^{13}$	$1.2829\cdot 10^{26}$	$8.6064\cdot 10^{51}$
$\psi_{1,16}$	$6.0964\cdot 10^{10}$	$1.7814\cdot 10^{21}$	$1.5612\cdot 10^{42}$	$1.2279\cdot 10^{84}$
$\psi_{1,32}$	$1.1152\cdot 10^{15}$	$9.1674\cdot 10^{29}$	$4.9020\cdot 10^{59}$	$1.6662\cdot 10^{119}$
$\psi_{2,\sqrt{2}}$	2.2960	3.7208	9.7394	$6.5890\cdot 10^{1}$
$\psi_{2,2}$	9.3539	$6.1551\cdot 10^{1}$	$2.6678\cdot 10^{3}$	$5.0111\cdot 10^{6}$
$\psi_{2,4}$	$1.7232\cdot 10^{3}$	$2.1500\cdot 10^{6}$	$3.4149\cdot 10^{12}$	$8.7128\cdot 10^{24}$
$\psi_{2,8}$	$3.5980\cdot 10^{6}$	$1.0031\cdot 10^{13}$	$7.9396\cdot 10^{25}$	$5.2284\cdot 10^{51}$
$\psi_{3,\sqrt{2}}$	6.2155	$1.3295\cdot 10^{1}$	$4.1722\cdot 10^{1}$	$4.2679\cdot 10^{2}$
$\psi_{3,2}$	$4.0164\cdot 10^{1}$	$3.9310\cdot 10^{2}$	$1.4047\cdot 10^{4}$	$1.1091\cdot 10^{7}$
$\psi_{3,4}$	$6.9272\cdot 10^{3}$	$6.8304\cdot 10^{6}$	$4.7564\cdot 10^{11}$	$5.1308\cdot 10^{20}$
$\psi_{3,8}$	$3.6660\cdot 10^{6}$	$9.4896\cdot 10^{11}$	$4.2745\cdot 10^{21}$	$1.1809\cdot 10^{39}$
$\psi_{4,2}$	$8.0487\cdot 10^{4}$	$3.5565\cdot 10^{9}$	$2.1354\cdot 10^{19}$	$4.8792\cdot 10^{38}$
$\psi_{4,4}$	$5.0774\cdot 10^{9}$	$1.1548\cdot 10^{19}$	$2.9916\cdot 10^{38}$	$1.2335\cdot 10^{77}$
$\psi_{4,8}$	$3.2564\cdot 10^{14}$	$4.3151\cdot 10^{28}$	$5.1348\cdot 10^{57}$	$2.9654\cdot 10^{115}$

Table 1. Computed values of $\|\varphi_{;n}\|$ for various types of singular behaviour $\psi_{i,\lambda}$ and orders n.

Various singularities and $d > 1$. We have also applied our algorithm jointly to several $\psi_{i,\lambda}$ as above and studied the dependency of $\|\varphi_{;n}\|$ on d. The results are shown in table 2 below. In the bottom rows, $\psi_{4,\lambda}^{*}$, $\psi_{4,\lambda}^{**}$, ... stand for distinct uncorrelated random series random$(\lambda\, z)$. In that case, the relation (17) generalizes to

$$\log \|\varphi_{;n}\| \approx \frac{n}{d+1} \log \lambda. \tag{18}$$

It also seems that the law can be adapted to functions with more common types of singularities, along similar lines as before. It would be interesting to have a theoretical explanation for the empirical laws (17) and (18). The division by $d+1$ in (18) is probably due to the fact that there are $d+1$ unknowns in the equation (13).

Of course, in the case that we are considering functions $f_1, \ldots, f_d$ with several singularities $\sigma_1, \ldots, \sigma_l$ on our disk of interest, only the singularities σ_i for which $|\sigma_i|$ is smallest are "numerically visible" (except when there exist relations which not involve these dominant singularities σ_i).

	32	64	128
$\psi_{1,8}$	$5.7101\cdot 10^{6}$	$1.6078\cdot 10^{13}$	$1.2829\cdot 10^{26}$
$\psi_{1,8}$, $\psi_{2,8}$	$4.8124\cdot 10^{3}$	$4.1720\cdot 10^{7}$	$2.0911\cdot 10^{15}$
$\psi_{1,8}$, $\psi_{2,8}$, $\psi_{1,8}(-z)$	$1.4867\cdot 10^{3}$	$3.8890\cdot 10^{6}$	$1.5664\cdot 10^{13}$
$\psi_{1,8}$, $\psi_{2,8}$, $\psi_{1,8}(-z)$, $\psi_{2,8}(-z)$	$7.9704\cdot 10^{2}$	$1.4235\cdot 10^{6}$	$4.9236\cdot 10^{12}$
$\psi_{4,8}$	$3.2564\cdot 10^{14}$	$4.3151\cdot 10^{28}$	$5.1348\cdot 10^{57}$
$\psi_{4,8}$, $\psi_{4,8}^{*}$	$2.3631\cdot 10^{9}$	$2.8202\cdot 10^{18}$	$1.9011\cdot 10^{38}$
$\psi_{4,8}$, $\psi_{4,8}^{*}$, $\psi_{4,8}^{**}$	$5.4571\cdot 10^{6}$	$5.2439\cdot 10^{13}$	$4.0431\cdot 10^{28}$
$\psi_{4,8}$, $\psi_{4,8}^{*}$, $\psi_{4,8}^{**}$, $\psi_{4,8}^{***}$	$2.4329\cdot 10^{5}$	$2.4748\cdot 10^{11}$	$2.4206\cdot 10^{22}$

Table 2. Computed values of $\|\varphi_{;n}\|$ for different input vectors of increasing dimensions d.

8.3. Towards criteria for an automatic final decision

In order to design a blackbox algorithm for the detection of dependencies (13), we need to make a final decision on whether we declare the input functions to be dependent or not. In view of the empirical law (18), and whatever decision criterion we implement, the outcome can only be trusted if the following two conditions are satisfied:

1. The number d should not be too large.

2. There exists a large constant $\lambda > 1$ for which all singularities of f are concentrated inside the disk of radius λ^{-1} or outside the disk of radius λ.

As usual, if we are interested in dependencies near an isolated singularity, then the second condition can often be achieved by zooming in on the singularity.

Our empirical observations in the previous section suggest that we should base the final decision on the empirical law (18). First of all, in the case of independence, we should observe a more or less geometric increase of the norm $\|\varphi_{;n}\|$ when running the algorithm for different orders n. If f only admits an isolated singularity in $\bar{\mathcal{D}}_1$, then we may also compute its norm $r = \lambda^{-1}$ and compare the computed values of $\|\varphi_{;n}\|$ with the expected values, as given by the law (18), or a suitable adaptation of this law when $\log \lambda$ becomes small. This method is most effective when λ is large, so it is usually recommended to zoom in on the singularity before doing this comparison. In fact, if the law (18) is satisfied for *different* zooming factors, then this may further increase our confidence that the input functions are independent.

For any numerical checks based on the law (18) or a refinement of it, we also recommend to precondition the input series $f_1, \ldots, f_d$. In particular, we recommend to multiply each f_i by a suitable constant, ensuring that

$$\max_{j<n} |f_{i,j}|\, r^j = 1,$$

whenever we apply our algorithm at order n. Here r is computed using one of the algorithms from section 3.

In fact, there is a trade-off between zooming in on the dominant singularity (thereby ensuring a large value of λ) and using the original input coefficients (which are given with a higher accuracy and order). In order to get more insight in how far we should zoom in, we need to analyze the cost of the algorithm from section 7.

The current working precision p should in general be taken larger than $n \log_2 \lambda$ in order to keep the method numerically stable. Denoting by $\mathsf{M}(p)$ the cost for multiplying two p-bit numbers, a naive implementation of the Gram-Schmidt orthogonalization procedure yields a total cost of $O(d^3\, n^3\, \mathsf{M}(p))$. Denoting by $\mathsf{MM}(k, p)$ the cost of multiplying two $k \times k$ matrices with p bit entries, and using a blockwise Gram-Schmidt procedure (similar to the procedure for matrix inversion in [Str69]), we obtain the better bound $O(\mathsf{MM}(d\, n, p))$. However, the matrix M from section 7 has a very special form. With more work, it might therefore be possible to save an additional factor $O(n)$, but we have not actively tried to do so yet.

Taking into account the effect of a possible zoom, we may next evaluate the computational cost in terms of the desired “output quality”. As a definition of the output quality, we may take the expected value $q \approx \frac{n}{d+1} \log \lambda$ of $\log \|\varphi_{;n}\|$ in the case when $f_1, \ldots, f_d$ are independent. In order to obtain accurate results, we need to compute with a bit precision larger than $q\, d$. In terms of q, the time complexity thus becomes

$$O\left(\mathsf{MM}\left(\frac{q\, d^2}{\log \lambda}, q\, d\right)\right) = \tilde{O}\left(\frac{q^{\omega+1}\, d^{2\omega+1}}{(\log \lambda)^{\omega}}\right) = \tilde{O}\left(\frac{q^4\, d^7}{(\log \lambda)^3}\right),$$

where $\omega < 2.373 \leqslant 3$ is the exponent of matrix multiplication [Str69, Pan84, CW87, Vas12]. The complexity bound makes it clear that we should put a lot of effort into keeping $\log \lambda$ large. If f satisfies a differential equation, then zooming in can be done with low computational cost (see section 4.1). Otherwise, the required expansion order grows quickly with the inverse of the distance to the singularity. Indeed, for $u \rightarrow 1$, the formula (8) becomes

$$\begin{aligned} \log \alpha &\approx -\frac{p}{i} \log 2 - \log (1 - u) \\ &\approx -\frac{p}{i} \log 2 - \log \lambda, \end{aligned}$$

where we recall that, in order to apply the orthogonalization process after zooming in, we need to expand f at an order which is α times larger than before.

Bibliography

[BBKW06] D. Bailey, J.M. Borwein, V. Kapoor, and E.W. Weisstein. Ten problems in experimental mathematics. *The American Mathematical Monthly*, 113(6):481–509, 2006.

[BD09] J.M. Borwein and K.J. Devlin. *The Computer As Crucible: An Introduction to Experimental Mathematics*. A.K. Peters Series. A.K. Peters, 2009.

[Bel11] K. Belabas. *Journées Nationales de Calcul Formel (2011)*, volume 2 of *Les cours du CIRM*, chapter Théorie algébrique des nombres et calcul formel. CEDRAM, 2011. Exp. No. 1, 40 pages, http://ccirm.cedram.org/ccirm-bin/item?id=CCIRM_2011__2_1_A1_0.

[Bir13] G.D. Birkhoff. Equivalent singular points of ordinary differential equations. *Math. Ann.*, 74:134–139, 1913.

[BK78] R.P. Brent and H.T. Kung. Fast algorithms for manipulating formal power series. *Journal of the ACM*, 25:581–595, 1978.

[BL94] B. Beckermann and G. Labahn. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM J. Matrix Analysis and Applications*, 15:804–823, 1994.

[BZ91] C. Brezinski and R. Zaglia. *Extrapolation Methods. Theory and Practice*. North-Holland, Amsterdam, 1991.

[CC90] D.V. Chudnovsky and G.V. Chudnovsky. Computer algebra in the service of mathematical physics and number theory (computers in mathematics, stanford, ca, 1986). In *Lect. Notes in Pure and Applied Math.*, volume 125, pages 109–232, New-York, 1990. Dekker.

[CGTW95] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The singular value decomposition for polynomial systems. In *Proc. ISSAC '95*, pages 195–207. ACM Press, 1995.

[CP78] B. F. Caviness and M. J. Prelle. A note on algebraic independence of logarithmic and exponential constants. *SIGSAM Bull.*, 12/2:18–20, 1978.

[CW87] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. In *Proc. of the* 19^{th} *Annual Symposium on Theory of Computing*, pages 1–6, New York City, may 25–27 1987.

[Der94] H. Derksen. An algorithm to compute generalized Padé-Hermite forms. Technical Report Rep. 9403, Catholic University Nijmegen, January 1994.

[DL89] J. Denef and L. Lipshitz. Decision problems for differential equations. *The Journ. of Symb. Logic*, 54(3):941–950, 1989.

[Fab85] E. Fabry. *Sur les intégrales des équations différentielles linéaires à coefficients rationnels*. PhD thesis, Paris, 1885.

[Fau94] J.-C. Faugère. Parallelization of Gröbner basis. In Hoon Hong, editor, *Parallel and Symbolic Computation*, volume 5 of *Lect. Notes Series in Computing*. Pasco'94, World Scientific, 1994.

[FS96] Ph. Flajolet and R. Sedgewick. *An introduction to the analysis of algorithms*. Addison Wesley, Reading, Massachusetts, 1996.

[GG02] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2-nd edition, 2002.

[HLM+02] J. van der Hoeven, G. Lecerf, B. Mourrain, et al. Mathemagix, 2002. http://www.mathemagix.org.

[Hoe99] J. van der Hoeven. Fast evaluation of holonomic functions. *TCS*, 210:199–215, 1999.

[Hoe01] J. van der Hoeven. Zero-testing, witness conjectures and differential diophantine approximation. Technical Report 2001-62, Prépublications d'Orsay, 2001.

[Hoe02] J. van der Hoeven. Relax, but don't be too lazy. *JSC*, 34:479–542, 2002.

[Hoe05] J. van der Hoeven. Effective complex analysis. *JSC*, 39:433–449, 2005.

[Hoe07] J. van der Hoeven. On effective analytic continuation. *MCS*, 1(1):111–175, 2007.

[Hoe09] J. van der Hoeven. On asymptotic extrapolation. *JSC*, 44(8):1000–1016, 2009.

[Hoe11] J. van der Hoeven. *Journées Nationales de Calcul Formel (2011)*, volume 2 of *Les cours du CIRM*, chapter Calcul analytique. CEDRAM, 2011. Exp. No. 4, 85 pages, http://ccirm.cedram.org/ccirm-bin/fitem?id=CCIRM_2011__2_1_A4_0.

[KL96] N. K. Karmarkar and Y. N. Lakshman. Approximate polynomial greatest common divisors and nearest singular polynomials. In Y.N. Lakhsman, editor, *Proc. ISSAC '96*, pages 35–42, Zürich, Switzerland, July 1996. ACM Press.

[Lan71] S. Lang. Transcendental numbers and diophantine approximation. *Bull. Amer. Math. Soc.*, 77/5:635–677, 1971.

[LLL82] A.K. Lenstra, H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

[Moo66] R.E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, N.J., 1966.

[Pan84] V. Pan. *How to multiply matrices faster*, volume 179 of *Lect. Notes in Math.* Springer, 1984.

[Poi86] H. Poincaré. Sur les intégrales irrégulières des équations linéaires. *Acta Math.*, 8:295–344, 1886.

[Pól37] G. Pólya. Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen. *Acta Mathematica*, 68:145–254, 1937.

[PTVF07] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes, the art of scientific computing*. Cambridge University Press, 3rd edition, 2007.

[Ric97] D. Richardson. How to recognise zero. *JSC*, 24:627–645, 1997.

[Ric07] D. Richardson. Zero tests for constants in simple scientific computation. *MCS*, 1(1):21–37, 2007.

[Str69] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:352–356, 1969.

[SW05] A.J. Sommese and C.W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, 2005.

[SZ94] B. Salvy and P. Zimmermann. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Trans. on Math. Software*, 20(2):163–177, 1994.

[Vas12] V. Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In 44^{th} *ACM Symp. on Theory of Computing (STOC'12)*, 2012. To appear. Preliminary version available at http://cs.berkeley.edu/~virgi/matrixmult.pdf.

[Ver96] J. Verschelde. *Homotopy continuation methods for solving polynomial systems*. PhD thesis, Katholieke Universiteit Leuven, 1996.

[Was67] W. Wasow. *Asymptotic expansions for ordinary differential equations*. Dover, New York, 1967.

[Wei00] K. Weihrauch. *Computable analysis*. Springer-Verlag, Berlin/Heidelberg, 2000.

[Wen01] E. J. Weniger. Nonlinear sequence transformations: Computational tools for the acceleration of convergence and the summation of divergent series. Technical Report math.CA/0107080, Arxiv, 2001.

[Wil94] H.S. Wilf. *Generating functionology*. Academic Press, 2nd edition, 1994. http://www.math.upenn.edu/~wilf/DownldGF.html.

[Zen04] Z. Zeng. The approximate gcd of inexact polynomials Part I: a univariate algorithm. http://www.neiu.edu/~zzeng/uvgcd.pdf, 2004.