Truncation Bounds for Differentially Finite Series

Marc Mezzarobba

Sorbonne Université, CNRS, Laboratoire d'informatique de Paris 6, LIP6, F-75005 Paris, France

Email: `marc.mezzarobba@lip6.fr`, `marc@mezzarobba.net`

June 17, 2018

ABSTRACT. We describe a flexible symbolic-numeric algorithm for computing bounds on the tails of series solutions of linear differential equations with polynomial coefficients. Such bounds are useful in rigorous numerics, in particular in rigorous versions of the Taylor method of numerical integration of ODEs and related algorithms. The focus of this work is on obtaining tight bounds in practice at an acceptable computational cost, even for equations of high order with coefficients of large degree. Our algorithm fully covers the case of generalized series expansions at regular singular points. We provide a complete implementation in SageMath and use it to validate the method in practice.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Context. From the point of view of rigorous numerics, computing the sum of a power series

$$u(z) = \sum_{n=0}^{\infty} u_n z^n \tag{1}$$

at a point $\zeta \in \mathbb{C}$ of its disk of convergence means obtaining an arbitrarily tight enclosure of $u(\zeta)$ given ζ. This reduces to computing, on the one hand, enclosures of the coefficients u_n, and on the other hand, a rigorous bound on the remainder, that is, a quantity $B(\zeta, N)$ such that

$$|u_{N:}(\zeta)| \leqslant B(\zeta, N) \qquad \text{where} \qquad u_{N:}(z) = \sum_{n=N}^{\infty} u_n z^n. \tag{2}$$

In order for the method to yield arbitrarily tight enclosures, the bound $B(\zeta, N)$ should at least tend to zero as N grows.

In the present work, we are interested in the case where the coefficients u_n of the series (1) are generated by a *linear recurrence with polynomial coefficients*. In other words, there exist rational functions $b_1, \ldots, b_s$ over some subfield $\mathbb{K} \subseteq \mathbb{C}$ such that, for large enough n, the coefficient u_n is given by

$$u_n = b_1(n) u_{n-1} + \cdots + b_s(n) u_{n-s}. \tag{3}$$

Equivalently, the analytic function $u(z)$ satisfies a linear ordinary differential equation (ODE) with coefficients in $\mathbb{K}(z)$,

$$a_r(z) u^{(r)}(z) + \cdots + a_1(z) u'(z) + a_0(z) u(z) = 0. \tag{4}$$

Functions with this property are called *differentially finite* (D-finite) or holonomic.

Under this assumption, computing the coefficients and the partial sum given sufficiently many initial values is not a problem in principle (though numerical stability, error analysis, or interval blow-up issues may occur). The question we consider here is the following: assuming that the series (1) is given by means of the differential equation (4) and an appropriate set of initial values, how can we compute bounds of the form (2) on its remainders ?

Theoretical answers to this question have long been known. In fact, textbook proofs of existence theorems for solutions of linear ODEs implicitly contain error bounds whose computation often can be made algorithmic, even in the nonlinear case. More specifically, the basic idea of the very method we use here can be traced back to Cauchy's 1842 memoir [5] on the "*calcul des limites*", or *method of majorants* as it is now called. (See Henrici [20, §9.2] or Hille [21, Sec. 2.4–2.6] for modern presentations of the classical method of majorants.) Yet, the basic method is not enough, by itself, to produce tight bounds in practice at a reasonable computational cost.

1.2. Contribution. The goal of the present article is to describe a practical and versatile algorithm for these bound computations. Our method fully covers the case of expansions at *regular singular points* of the ODE (see Section 5.1 for reminders), including expansions in generalized series with non-integer powers and logarithms. (It does not apply in the *irregular* case, even to individual solutions that happen to be convergent.) It is designed to be easy to implement in a symbolic-numeric setting using interval arithmetic, and to be applicable to fast evaluation algorithms based on binary splitting and related techniques [7]. We provide a complete implementation in the SageMath computer algebra system, and use it to assess the tightness of the bounds with experiments on non-trivial examples.

To see more precisely what the algorithm does, consider first the following inequality that one might write to bound the remainder of the exponential series with pen and paper:

$$\left| \sum_{n \geqslant N} \frac{\zeta^n}{n!} \right| = \frac{|\zeta|^N}{N!} \sum_{n=0}^{N} \frac{N!}{(N+n)!} \, |\zeta|^N \leqslant \mathrm{e}^{|\zeta|} \frac{|\zeta|^N}{N!}. \tag{5}$$

We are aiming for bounds of a similar shape: like this one, our bounds decompose into a factor that mainly depends on the analytic behavior of the function of interest (actually, in our case, of the general solution of the ODE), multiplied by one that is related to the first few neglected terms of the series. Also like this one, they are parameterized by the truncation order and the evaluation point; however, we focus on the evaluation algorithm giving the bound as a function of N and ζ rather than expressing it as a formula.

Example 1. As a "real-world" example of medium difficulty, consider the differential equation

$$
\begin{aligned}
z^3 \, (z-1) \, (z+2) \, (z+3) \, (z+6) \, (z+8) \, (3 \, z+4)^2 \quad & P^{(4)}(z) \\
+ \, (126 \, z^9 + 2304 \, z^8 + 15322 \, z^7 + 46152 \, z^6 + 61416 \, z^5 + 15584 \, z^4 - 39168 \, z^3 - 27648 \, z^2) \quad & P^{(3)}(z) \\
+ \, (486 \, z^8 + 7716 \, z^7 + 44592 \, z^6 + 119388 \, z^5 + 151716 \, z^4 + 66480 \, z^3 - 31488 \, z^2 - 32256 \, z) \quad & P''(z) \\
+ \, (540 \, z^7 + 7248 \, z^6 + 35268 \, z^5 + 80808 \, z^4 + 91596 \, z^3 + 44592 \, z^2 + 2688 \, z - 4608) \quad & P'(z) \\
+ \, (108 \, z^6 + 1176 \, z^5 + 4584 \, z^4 + 8424 \, z^3 + 7584 \, z^2 + 3072 \, z) \quad & P(z) = 0
\end{aligned}
$$

for the lattice Green function of the four-dimensional face-centered cubic lattice [16, 26]. Make the change of unknown function $u(z) = P(1/2 + z)$ (as a Taylor method may do, cf. Section 2.2), and let $u(z)$ be a solution of the resulting "shifted" equation corresponding to small rational initial values $u(0)$, ..., $u^{(4)}(0)$ chosen at random. Figure 1 compares the truncation error after n terms of the Taylor expansion of $u(z)$ evaluated at $\zeta = 1/4$ (halfway from the singular points closest to the origin, which are at $\pm 1/2$ after the transformation) with the bound on the tail of this series that our method produces given the last few known coefficients before the truncation point. As we could hope in view of the above discussion of the "shape" of our estimates, the overestimation appears to be roughly constant for large n. (Under mild assumptions, it could actually be shown to be $\mathrm{O}(\log n)$ in the worst case.) On this example, it would cause us in this case to compute about 10% more terms than really necessary to reach an accuracy of 10^{-50}.
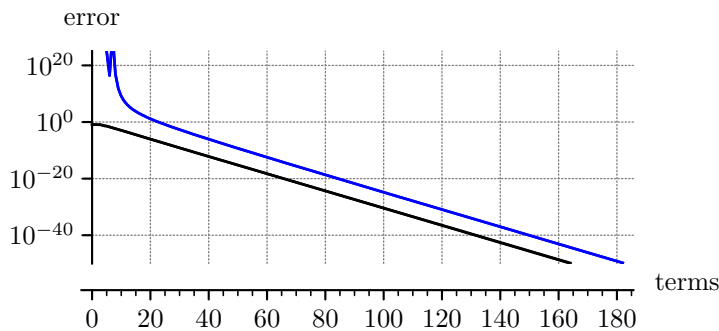
Figure 1. Bounds computed by our implementation for the problem described in Example 1. The bottom curve (in black in the color version) shows the actual error $|u_{n:}(\zeta) - u(\zeta)|$ committed as a function of the truncation order n, while the top curve (in blue) corresponds to the bound given by Algorithm 10 with $\ell = 5$. See Section 9 for details.

Internally, what the algorithm actually computes for fixed N is a *majorant series* (Definition 3 below) of the remainder $u_{N:}(z)$. Given such a series, one readily deduces bounds on $u_{N:}(\zeta)$ for a given ζ, but also on related quantities like remainders of the derivative $u'(z)$ or higher-order remainders $u_{N':}(z)$, $N' \geqslant N$. This last feature means that the method provides both "a priori" and "a posteriori" bounds: the knowledge of the coefficients $u_0, ..., u_{N-1}$ (for large enough N) can be used to bound $u_{N':}(\zeta)$ for any $N' \geqslant N$, and these bounds already tend to zero as N' increases, but taking N closer to N' leads to tighter bounds, as illustrated on Figure 3 on page 28.

For the method to be of any interest, the majorant series we are computing need to be significantly simpler than the series we are to bound. We seek *hyperexponential* majorant series, that is, majorant series of the form

$$\sum_{n=0}^{\infty} \hat{u}_n z^n = \exp \int_0^z a(w) \, \mathrm{d}w, \qquad a(z) \in \mathbb{R}(z),$$

or, equivalently, series that satisfy linear ODEs of the form (4) but of order $r = 1$.

The algorithm to compute these hyperexponential majorant series is based on a novel combination of two classical ideas:

1. estimating the error in the solution of linear equations using residuals,

2. bounding the solutions of ODEs with analytic coefficients by the method of majorants.

An analogy with an elementary situation might be helpful to see how this works. Consider the linear system $A x = b$, for some invertible matrix $A \in \mathbb{C}^{r \times r}$. Suppose that we have computed an approximation $\tilde{x}$ of the exact solution x and we want to bound the approximation error $\|\tilde{x} - x\|$. Suppose, additionally, that we are given an M such that $\|A^{-1}\| \leqslant M$. Writing

$$\|\tilde{x} - x\| = \|A^{-1}(\tilde{b} - b)\| \leqslant M \|\tilde{b} - b\|, \tag{6}$$

all we need to conclude is an approximation of the residual $\tilde{b} - b$, which is easy to obtain. In our setting, the residuals can be computed from a small number of terms of the coefficient sequence of the series, while the method of majorants provides the required "bound on the inverse" of the differential operator.

Continuing with the analogy, the more effort we spend computing M, the tighter we can make the final bound. Nevertheless, even if the inequality $\|A^{-1}\| \leqslant M$ is loose, as soon as the residual is computed accurately enough, the bound (6) overestimates the actual error by a constant factor only as $\tilde{x}$ tends to x for fixed A. In the differential case as well, tighter bounds come at a higher cost even for fixed N. Our algorithm contains additional parameters that can be adjusted to influence the trade-off between computational cost and accuracy. Thus, the same general algorithm can be used to obtain simple but sufficient bounds in "easy" cases or, with more effort, better bounds in "hard" cases where the simpler one are unusable.

1.3. Outline. The rest of this article is organized as follows. Section 2 compares the approach taken here with existing work and discusses some applications. Section 3 presents the notation used in the sequel while recalling a few classical results. Section 4 sketches our algorithm and the inequalities on which it relies, under some simplifying assumptions. The reader only interested in the general idea can stop there. The following sections are concerned with technical details and proofs related to the bound computation algorithm. In Section 5, we prove a majorization theorem that covers expansions at regular singular points, laying the foundation for the detailed description of our main algorithm in Section 6. We then consider the practical aspects of computing certain intermediate bounds on sequences defined by rational functions in Section 7, and the derivation of various kinds of concrete numerical bounds from the output of the main algorithm in Section 8. Finally, Section 9 describes experiments that illustrate the quality of these bounds and the effect of the tuning parameters.

2. Related work

While articles directly comparable to the present one are relatively few, similar questions appear naturally as sub-problems of various computational tasks. Somewhat arbitrarily, we group previous work by context in two categories: the evaluation of classical functions, and the numerical solution of ordinary differential equations.

2.1. Differentially finite functions as special functions. An important application of the summation of power series where tail bounds are needed is the rigorous multiple-precision evaluation of special functions [3]. For a fixed function, it is usually not too hard to derive good ad hoc bounds [32]. Bounds that cover wider classes of functions become more complicated as the number of parameters increases, yet adequate tail bounds are available in the literature and used in practice for common special functions depending on parameters. For example, Du and Yap [11, Sec. 3] or Johansson [22, Sec. 4.1] present bounds that cover general hypergeometric functions.

From this perspective, our goal is to describe an algorithm for bound computations that applies to the "general differentially finite function" of a complex variable, viewed as a special function parameterized by the complete coefficient list of the differential equation (4). Van der Hoeven [42, Sec. 2] [43, Sec. 2.4] [44, Sec. 3.5] already gave several such algorithms, some based, like ours, on the method of majorants. Like the theoretical algorithms read between the lines of existence proofs mentioned in the Introduction, van der Hoeven's algorithms suffer from overestimation issues that make them unsatisfactory in practice. Our method can be seen as a refinement of these ideas yielding tighter bounds.

The present author already considered related tightness issues in an article with B. Salvy [30] on *asymptotically* tight bounds on linear recurrence sequences, with some extensions (including a few of the ideas developed in the present paper) in the author's doctoral thesis [28]. While the focus of that article was not on tail of power series, the remainder bounds on tails of power series that came as a corollary seemed to perform well in experiments with simple special functions. Unfortunately, they later proved too pessimistic and costly to compute with more complicated equations. The setting of the present work is different in that we do not insist on asymptotic tightness, nor on producing human-readable formulae, and focus instead on making the bounds easy to evaluate numerically and as tight as possible for realistic values of the parameters.

2.2. Interval methods for ODEs. Directly summing the series (1) to compute $u(\zeta)$ only works when ζ lies within its disk of convergence $\mathcal{D}$. When that is not the case, an option is to evaluate the function by "numerical analytic continuation", that is, to first use (1) to compute the Taylor expansion of u at some point $\zeta_1 \in \mathcal{D}$ closer to the boundary, then use that series to compute the expansion of u at a point ζ_2, possibly outside of $\mathcal{D}$, and so on until we reach ζ. Because u satisfies the differential equation (4), it is enough at each step i to compute the first r derivatives of u at ζ_{i+1} using the Taylor expansion at ζ_i, and the rest of the expansion follows from the differential equation.

We can also take a slightly different perspective and view the evaluation algorithm as a numerical ODE solver. Seen from this angle, what we just outlined is the numerical solution of (4) by a *Taylor method*. Taylor methods are among the oldest numerical methods for ODEs. While often considered too costly for machine-precision computations, they remain the methods of choice for high precision and for interval computations [1, 33]. In addition, they can be made particularly efficient in the case of ODEs of the form (4) [7, Sec. 2–3], and the efficient variants generalize to deal with singular problems [45].

Rigorous Taylor methods (not limited to the linear case) have attracted a lot of interest from the field of Interval Analysis, starting with Moore's 1962 PhD thesis [31, Sec. 6]. Since a Taylor method reduces the solution of an ODE to the summation of series expansions of (derivatives of) solutions, the problem of bounding the remainders occurs naturally. Instructive overviews of the ideas used in this context can be found in the literature [40, 33, 34]. A popular approach is to start with a coarse enclosure of the solution and its first derivatives over the whole time step, obtained by a fixed-point argument based for instance on Picard's iteration. One then deduces an enclosure of the derivative of order N using the ODE, and a bound for the remainder of order N by Taylor's formula. If, instead of values on a grid, one computes a polynomial approximation of the solution over some domain, it is also possible to apply a fixed-point argument directly to a neighborhood of this approximation in function space, by evaluating an integral form of the differential equation in polynomial interval arithmetic (using Taylor models for instance). The way our algorithm uses residuals bears some conceptual similarity to this last approach, but our method does not require an explicit polynomial approximation of the solution as input.

Still in the context of Taylor methods, but closer in spirit to van der Hoeven's approach or ours, Warne *et al.* [46] develop explicit majorants for solutions of nonlinear differential equations. Neher [35, 36] considers the case of linear ODEs with analytic coefficients and bounds the coefficients of the local series expansions of the solution by geometric sequences, starting from bounds of the same type on the coefficients of the equation. In the special case of ordinary points, our method might be described as a blend of Neher's approach, simplified by the language of majorant series, with elements of the "continuous approximation" strategy.

2.3. The regular singular case. To the best of our knowledge, the present work is the first that directly applies to logarithmic solutions at regular singular points (with an early version of some of the ideas in the author's dissertation [28, Sec. 6.4]). The other methods mentioned above are limited to series solutions, often at ordinary points only, with the exception that van der Hoeven [43, Sec. 3.3.2] sketches an adaptation to the case of logarithmic solutions of the earliest of his algorithms for plain power series.

3. Notation and reminders

3.1. Formal power series. We denote by $R[[z]]$ the ring of formal power series over R in the variable z, and by $R((z))$ the ring of formal Laurent series. Given $f \in R((z))$, we denote by f_n or $[z^n] f(z)$ the coefficient of z^n in $f(z)$. As in equation (2), we also write $f_{n:}(z)$ for the remainder $\sum_{i \geqslant n} f_i z^i$. More generally, f_ν is the coefficient of z^ν in a generalized series $\sum_{\nu \in \Lambda} f_\nu z^\nu$ indexed by $\Lambda \subset \mathbb{C}$. In this context, Roman letters denote integers, while Greek coefficient indices can be arbitrary complex numbers. We sometimes identify expressions representing analytic functions with their series expansions at the origin. Thus, for instance, $[z^n] (1 - \alpha z)^{-1}$ is the coefficient of z^n in the Taylor expansion of $(1 - \alpha z)^{-1}$ at 0, that is, α^n.

We also consider polynomials in $\log(z)$ with formal power (or Laurent) series coefficients. The space $R((z))[\log z]$ of such expressions embeds in $R[\log z]((z))$, and for $f \in R((z))[\log z]$, the notations f_n and $f_{n:}$ refer to the coefficients of f viewed as an element of $R[\log z]((z))$.

3.2. Differential equations and recurrences. We assume some familiarity with the classical theory of linear differential equations with analytic coefficients, as described for instance in Henrici's or Hille's books [19, 21], and with the basic properties of D-finite formal power series, for which we refer the reader to Stanley [41, Sec. 6.4] or Kauers and Paule [25]. In particular, we will freely use the properties summarized below.

We usually write linear differential equations $a_r(z) y^{(r)}(z) + \cdots + a_0 y(z) = q(z)$ in operator form, i.e. as $\mathcal{L} \cdot y = q$ where $\mathcal{L}$ is a linear operator acting on some space of functions of interest. When the coefficients a_k are formal Laurent series, such an operator can be written as $\mathcal{L} = P(z, D_z)$, that is, as a polynomial $P(X, Y) \in \mathbb{C}((X))[Y]$ evaluated on the operators $z : y \mapsto (w \mapsto w\, y(w))$ that multiplies a function by the identity function and the standard differentiation operator $D_z : y \mapsto y'$. Note that the operators z and D_z do not commute; one can bring the coefficients $a_k(z)$ "to the right" of D_z^k using the relation $D_z z = z D_z + 1$ deduced from Leibniz' rule. By abuse of notation, we do not always make the difference between the operator $P(z, D_z)$ and the polynomial $P(X, Y)$.

Similarly, recurrences $b_s(n)\, u_{n+s} + \cdots + b_0(n)\, u_n = v_n$ with polynomial coefficients can be written $P(n, S_n) \cdot u = v$ where $n \colon (u_k) \mapsto (k\, u_k)$ is the operator that multiplies a sequence by its index, $S_n \colon (u_n) \mapsto (u_{n+1})$ is the shift operator, and $P(X, Y) \in \mathbb{C}[X][Y]$. The corresponding commutation rule reads $S_n\, n = (n+1)\, S_n$. We typically consider bi-infinite sequences (indexed by $\mathbb{Z}$ or $\lambda + \mathbb{Z}$, $\lambda \in \mathbb{C}$), so that we also have at our disposal the backward shift operator $S_n^{-1} \colon (u_n) \mapsto (u_{n-1})$.

In the differential case, when working in the neighborhood of the origin, it proves convenient to write differential operators in terms of the *Euler derivative* defined by $\theta = z\, D_z$ instead of the standard derivative D_z. Any operator $\mathcal{L} = a_r(z)\, D_z + \cdots + a_0(z) \in \mathbb{C}((z))[D_z]$ can be written $\mathcal{L} = L(z, \theta)$ with $L = \tilde{a}_r(X)\, Y^r + \cdots + \tilde{a}_0(X) \in \mathbb{C}((X))[Y]$. Conversely, an operator $\mathcal{L} \in \mathbb{C}[z][D_z]$ can be rewritten as an element of $\mathbb{C}[z][\theta]$ at the price of multiplying it on the left by a power of z. This change does not affect the space of solutions of the equation $\mathcal{L} \cdot y = 0$. One can check that the quotient $a_r(z)\, /\, \tilde{a}(z)$ of the leading coefficients with respect to D_z and to θ is a power of z. "Moving" the coefficients of either form "to the right" of the differentiation operators leaves the leading coefficient unchanged.

It is classical that the coefficients of a linear ODE with power series coefficients obey a recurrence relation obtained by "changing θ to n and z^{-1} to S_n" in the equation. The precise result is as follows, see Proposition 5 below for a sketch of a proof in a more general setting.

Proposition 2. *Let $L \in \mathbb{C}[[X]][Y]$. A formal power series $y \in \mathbb{C}[[z]]$ satisfies the differential equation $L(z, \theta) \cdot y = 0$ if and only if its coefficient sequence (y_n), extended with zeroes for negative n, satisfies $L(S_n^{-1}, n) \cdot (y_n)_{n \in \mathbb{Z}} = 0$.*

Note that in general, the recurrence operator $L(S_n^{-1}, n)$ has infinite order, but since y_n vanishes for $n < 0$, each coefficient of the sequence $L(S_n^{-1}, n) \cdot (y_n)$ only involves a finite number of y_n. When $L(z, \theta)$ has polynomial coefficients, the corresponding recurrence has finite order.

3.3. Majorant series. The language of majorant series offers a flexible framework to express bounds on series solutions of differential equations.

Definition 3. *A formal power series $\hat{f} \in \mathbb{R}_+[[z]]$ is said to be a* majorant series *(sometimes a* majorizing *series) of $f \in \mathbb{C}[[z]]$, if the coefficients of f and $\hat{f}$ satisfy $|f_n| \leqslant \hat{f}_n$ for all $n \in \mathbb{N}$. More generally, we say that $\hat{f} \in \mathbb{R}_+[[z]]$ is a majorant series of*

$$f(z) = f_0(z) + f_1(z) \log(z) + \cdots + f_{K-1}(z)\, \frac{\log(z)^{K-1}}{(K-1)!} \in \mathbb{C}[[z]][\log z]$$

if it is a majorant series of $f_0, \ldots, f_k$. In both cases, we write $f(z) \ll \hat{f}(z)$.

Series denoted with a hat in this article always have non-negative coefficients, and $\hat{f}$ is typically some kind of bound on f, but not necessarily a majorant series in the sense of the above definition.

The following properties are classical and easy to check (see, e.g., Hille [21, Sec. 2.4]). Note the assumption that f and g are plain power series: while convenient to state some results, the extension to $f \in \mathbb{C}[[z]][\log z]$ does not share all the nice properties of majorant series in the usual sense.

Proposition 4. *Let $f, g \in \mathbb{C}[[z]]$, $\hat{f}, \hat{g} \in \mathbb{R}_+[[z]]$ be such that $f \ll \hat{f}$ and $g \ll \hat{g}$. The following assertions hold:*

(a) $\quad f + g \ll \hat{f} + \hat{g}$, $\qquad$ *(b)* $\quad \gamma f \ll |\gamma|\, \hat{f}$ *for* $\gamma \in \mathbb{C}$, $\qquad$ *(c)* $\quad f_{n:}(z) \ll \hat{f}_{n:}(z)$ *for* $n \in \mathbb{N}$,

(d) $\quad f'(z) \ll \hat{f}'(z)$, $\qquad$ *(e)* $\quad \left(\displaystyle\int_0^z f \right) \ll \left(\displaystyle\int_0^z \hat{f} \right)$, $\qquad$ *(f)* $\quad f g \ll \hat{f} \hat{g}$.

Additionally, the disk of convergence $\hat{D}$ of $\hat{f}$ is contained in that of f, and when $\hat{g}_0 \in \hat{D}$, we have $f(g(z)) \ll \hat{f}(\hat{g}(z))$. In particular, $|f(\zeta)|$ is bounded by $\hat{f}(|\zeta|)$ for all $\zeta \in \hat{D}$.

4. A sketch of the method

In this section, we outline how our bounds are obtained, in a simplified setting and without giving detailed algorithms. Our simplified setting covers all solutions at *ordinary* (non-singular) points of the differential equation, where Cauchy's theorem applies and all solutions are analytic. The contents of this section may be enough for the reader only interested in the general idea of the algorithm. Sections 5 to 7 essentially repeat the same algorithm in the general case, while adding more detail and introducing refinements that help obtain tighter bounds.

4.1. Truncated solutions and residuals.

We start with a linear differential equation (4), with rational coefficients and right-hand side. As observed in Section 3.2, such an equation can always be brought into the form

$$P(z, \theta) \cdot u(z) = [\theta^r p_r(z) + \cdots + \theta p_1(z) + p_0(z)] \cdot u(z) = 0, \qquad p_k \in \mathbb{C}[z]$$

without changing its solution space. The unusual choice of writing the coefficients p_k to the right of θ^k is not essential, but will prove convenient later. We also assume without loss of generality that the polynomials $p_0, ..., p_r$ are globally coprime. Most importantly, we assume that the leading coefficient p_r does not vanish at 0. This is the case when the origin is an ordinary point of (4)[1], i.e., when none of the fractions $a_k(z)/a_r(z)$, $0 \leqslant k < r$, has a pole at 0.

Let $u(z)$ be a solution of $P(z, \theta) \cdot u(z) = 0$, and consider the truncation

$$\tilde{u}(z) = \sum_{n=0}^{N-1} u_n z^n$$

of the series $u(z)$ to some large enough order $N \geqslant 1$. Our goal is to obtain an explicit majorant series of the remainder $u(z) - \tilde{u}(z)$. This remainder satisfies

$$P(z, \theta) \cdot [\tilde{u}(z) - u(z)] = P(z, \theta) \cdot \tilde{u}(z) = q(z),$$

where $q(z)$, the residual associated to the approximate solution $\tilde{u}(z)$, is an explicit, computable polynomial. Because $\tilde{u}$ is a truncation of an exact series solution, the residual $q(z)$ has "small" support: more precisely, it can be checked to be of the form $q(z) = q_N z^N + \cdots + q_{N+s} z^{N+s}$ where $s = \deg_z P(z, \theta)$.

4.2. The majorant equation.

Setting

$$y(z) = p_r(z) (\tilde{u}(z) - u(z)), \tag{7}$$

we get an equation with rational function coefficients

$$L(z, \theta) \cdot y(z) = \left[\theta^r + \cdots + \theta \frac{p_1(z)}{p_r(z)} + \frac{p_0(z)}{p_r(z)}\right] \cdot y(z) = q(z), \tag{8}$$

in which we immediately rewrite $L(z, \theta)$ as

$$L(z, \theta) = \sum_{j=0}^{\infty} Q_j(\theta) z^j$$

by expanding $p_r(z)^{-1}$ in power series and rearranging. Crucially, since $p_r(0) \neq 0$, the polynomial Q_0 has degree r, while the degree of Q_j for $j \geqslant 1$ is at most $r - 1$.

By the correspondence of Proposition 2, the coefficient sequence $(y_n)_{n \in \mathbb{Z}}$ of $y(z)$ then satisfies

$$L(S_n^{-1}, n) \cdot (y_n)_{n \in \mathbb{Z}} = [Q_0(n) + Q_1(n) S_n^{-1} + \cdots] \cdot (y_n)_{n \in \mathbb{Z}} = (q_n)_{n \in \mathbb{Z}},$$

that is,

$$Q_0(n) y_n = q_n - \sum_{j=1}^{\infty} Q_j(n) y_{n-j}, \qquad n \in \mathbb{Z}.$$

---

1. Assuming $p_r(0) \neq 0$ also allows for a *regular singular point* at the origin, cf. Section 5.1. However, we restrict ourselves in this section to *power series solutions* at regular singular points, leaving out a number of technicalities related to non-integer exponents and logarithms.

When n is large enough, $Q_0(n)$ does not vanish and y_n is given recursively by

$$y_n = \frac{1}{n}\left(\frac{n\,q_n}{Q_0(n)} - \sum_{j=1}^{\infty} \frac{n\,Q_j(n)}{Q_0(n)}\,y_{n-j}\right). \tag{9}$$

Because of the degree constraints noted above, the quantities $n\,q_n/Q_0(n)$ and $n\,Q_j(n)/Q_0(n)$ (for fixed j) remain bounded as n grows. Assume for a moment that we have at our disposal bounds $\hat{q}_n$ and $\hat{a}_j$ such that

$$\left|\frac{n\,q_n}{Q_0(n)}\right| \leqslant \hat{q}_n < \infty, \qquad n \geqslant n_0, \tag{10}$$

$$\left|\frac{n\,Q_j(n)}{Q_0(n)}\right| \leqslant \hat{a}_j < \infty, \qquad n \geqslant n_0, \quad j \geqslant 1, \tag{11}$$

with $\hat{a}_j = \mathrm{O}(\alpha^j)$ for some α as $j \to \infty$.

Equation (9) then leads to

$$|y_n| \leqslant \frac{1}{n}\left(\hat{q}_j + \sum_{j=1}^{\infty} \hat{a}_j\,|y_{n-j}|\right), \qquad n \geqslant n_0.$$

Thus, if $(\hat{y}_n)_{n\in\mathbb{Z}}$ satisfies

$$\hat{y}_n = \frac{1}{n}\left(\hat{q}_n + \sum_{j=1}^{\infty} \hat{a}_j\,\hat{y}_{n-j}\right), \qquad n \geqslant n_0, \tag{12}$$

and

$$|y_n| \leqslant \hat{y}_n \qquad \text{for all} \qquad n < n_0, \tag{13}$$

then $|y_n|$ is bounded by $\hat{y}_n$ for all n.

Translating (12) back to a differential equation, we obtain

$$[\theta - \hat{a}(z)]\,\hat{y}(z) = \hat{q}(z) \qquad \text{where} \qquad \hat{a}(z) = \sum_{j=1}^{\infty} \hat{a}_j\,z^j. \tag{14}$$

Because $\hat{a}_j = \mathrm{O}(\alpha^j)$, the power series $\hat{a}(z)$ is convergent. We call the equation (14) a *majorant equation* associated with (8).

4.3. Majorant series for the remainders. The general solution of (14) reads

$$\hat{y}(z) = h(z)\left(c + \int_0^z \frac{w^{-1}\,\hat{q}(w)}{h(w)}\,\mathrm{d}w\right), \qquad \hat{h}(z) = \exp\left(\int_0^z w^{-1}\,\hat{a}(w)\,\mathrm{d}w\right) \tag{15}$$

for an arbitrary constant $c \in \mathbb{C}$. We have just seen that any $\hat{y}(z)$ of this form such that (13) holds is a majorant series for $y(z)$.

It remains to choose the parameters $\hat{q}$, $\hat{a}$, and c in such a way that (10), (11), and (13) hold. Recall that, in our context, $y(z) = p_r(z)\,(\tilde{u}(z) - u(z))$ is a power series of valuation at least $N \geqslant 1$. Provided that N is large enough, we can assume $n_0 = N$ and take $c = 0$. Equation (13) is then automatically satisfied. Since at most s of the constraints (10) are nontrivial, it is not hard to compute explicit values $\hat{q}_n$ fulfilling these constraints. Knowing $h(z)$, we can even choose $\hat{q}(z)$ as a polynomial multiple[2] of $h(z)$, so that the integral in the expression of $\hat{y}(z)$ reduces to an explicit polynomial $\hat{g}(z)$.

The case of (11) is slightly more involved, as, in appearance, it entails an infinite number of inequalities. However, the polynomials Q_j stem from the power series expansions of the finitely many rational functions p_k/p_r of Eq. (8). By bounding the numerator and denominator of $L(z,\theta)$ separately, we can find a single rational function $\hat{a}(z)$ that satisfies all these inequalities; moreover, we can arrange that $\hat{a}(z)$ have radius of convergence arbitrarily close to that of $p_r(z)^{-1}$.

---

2. Since the power series $h(z)$ has nonnegative coefficients and starts with $h(0) = 1$, replacing any polynomial $\hat{q}(z)$ computed from (10) alone by $\hat{q}(z)\,h(z)$ will do.

A similar (but simpler) computation yields a polynomial $\check{p}(z)$ such that $\check{p}(z)^{-1}$ is a majorant series of $p_r(z)^{-1}$ with the same condition on the radius of convergence. Putting everything together, we obtain an explicit majorant series $\hat{u}(z)$ for the error $u(z) - \tilde{u}(z)$, in the form

$$\hat{u}(z) = \frac{\hat{y}(z)}{\check{p}(z)} = \frac{\hat{g}(z)}{\check{p}(z)}\,\hat{h}(z) = \frac{\hat{g}(z)}{\check{p}(z)}\exp\left(\int_0^z w^{-1}\,\hat{a}(w)\,\mathrm{d}w\right) = \mathrm{rat}_1(z)\exp\left(\int_0^z \mathrm{rat}_2\right). \tag{16}$$

As noted in Proposition 4, we have in particular $|u(\zeta) - \tilde{u}(\zeta)| \leqslant \hat{u}(|\zeta|)$ for all ζ in the disk of convergence of $\hat{u}(z)$.

Observe that, in (16), $\hat{g}(z)$ can be chosen of valuation N, so that $\hat{u}(z)$ itself has valuation N. Moreover, its coefficients can be taken of roughly the same order of magnitude as those of the first neglected terms of the series $u(z)$. The radius of convergence of $\hat{a}(z)$ and $\hat{p}(z)^{-1}$ can be made arbitrarily close to the distance from the origin to the smallest nonzero singularity of the differential equation (4) (or even equal to it, but manipulating this quantity exactly may be costly). In the absence of apparent singularities, this distance is the local radius of convergence of a generic solution of the differential equation. Thus, there is reason to hope that $\hat{u}(|\zeta|)$ does not stray too far from $|u(\zeta) - \tilde{u}(\zeta)|$ as N and ζ vary.

5. Majorant equations: the general case

The key step of the reasoning outlined above is the construction in Section 4.2 of a "majorant equation" of the inhomogeneous equation (8). Our goal is now to extend this construction to the general case of solutions at regular singular points. To make them applicable to variants of the algorithms adapted to other situations, the results are stated in slightly more general form than actually needed in the sequel.

5.1. Regular singular points. Let us start with some reminders on regular singular points. Consider an operator $\mathcal{D} = a_r(z)\,D_z^r + \cdots + a_1(z)\,D_z + a_0(z)$, where $a_0, ..., a_r$ are analytic functions with no common zero. Recall that $\zeta \in \mathbb{C}$ is called a *singular point* of the operator $\mathcal{D}$ when $a_r(\zeta) = 0$, and an *ordinary point* otherwise.

At an ordinary point, by Cauchy's theorem, the equation $\mathcal{D}\cdot y = 0$ admits r linearly independent analytic solutions. If ζ is a singular point, analytic solutions defined on domains with ζ on their boundary are in general not analytic at ζ. *Regular singular points* are a class of singular points that share many similarities with ordinary points, starting with the existence of convergent (generalized) series solutions.

A singular point ζ of $\mathcal{D}$ is called regular singular when, for all k, its multiplicity as a pole of a_k/a_r is at most $r - k$. We say that an operator $\mathcal{D}$ is *regular* at a point ζ (or that ζ is a *regular point* of $\mathcal{D}$) if ζ is either an ordinary point or a regular singular point of $\mathcal{D}$. An operator written in the form $\tilde{a}_r(z)\,\theta^r + \cdots + \tilde{a}_0(z)$ where at least one of $\tilde{a}_0, ..., \tilde{a}_r$ has a nonzero constant coefficient is regular at the origin if and only if $\tilde{a}_r(0) \neq 0$. This is equivalent to saying that the univariate polynomial $Q_0(\theta) = \tilde{a}_r(0)\,\theta^r + \cdots + \tilde{a}_0(0)$, called the *indicial polynomial* of the operator, has degree r.

When $\zeta \in \mathbb{C}$ is a regular point of $\mathcal{D}$, the equation $\mathcal{D}\cdot y = 0$ admits r linearly independent solutions of the form

$$(z-\zeta)^\lambda\big(f_0(z) + f_1(z)\log(z-\zeta) + \cdots + f_{t-1}(z)\log(z-\zeta)^{t-1}\big) \tag{17}$$

(possibly with different λ and f_i), where the f_i are *analytic at ζ*. The possible λ are exactly the roots of Q_0.

Formally at least, it is often convenient to rewrite (17) as a single series

$$\sum_{\nu \in \lambda + \mathbb{N}} c_\nu(w)\,w^\nu \in w^\lambda\,\mathbb{C}[\log w][[w]], \qquad w = z - \zeta.$$

We call expressions of this form *logarithmic series*, and refer to Henrici [20, §9.5] for a rigorous presentation of their algebraic manipulation.

5.2. Recurrences on the coefficients of solutions. The coefficients of the logarithmic series expansions of solutions of linear ODEs at regular points are related by systems of recurrence relations whose existence and explicit description go back to Frobenius [13] and Heffter [18, Kap. VIII]. Here we recall these results in a compact formalism [27, 28] close to that of Poole [39] which makes them appear as direct generalizations of Proposition 2.

Proposition 5. *[39, 27] Let $y(z), q(z) \in z^\lambda \mathbb{C}[[z]][\log z]$ (with $\lambda \in \mathbb{C}$) be logarithmic series. Write*

$$y(z) = \sum_{\nu \in \lambda + \mathbb{Z}} \sum_{k=0}^{\infty} y_{\nu,k} z^\nu \frac{\log(z)^k}{k!} = z^\lambda \sum_{n=0}^{\infty} \sum_{k=0}^{K-1} y_{\lambda+n,k} z^n \frac{\log(z)^k}{k!},$$

where it is understood that $y_{\lambda+n,k} = 0$ for $n \leqslant 0$ or $k \geqslant K$, and similarly for $q(z)$. Let $L(X, Y)$ be an element of $\mathbb{C}[[X]][Y]$. The differential equation

$$L(z, \theta) \cdot y(z) = q(z)$$

holds if and only if the double sequences $(y_{\nu,k}), (q_{\nu,k}) \in \mathbb{C}^{(\lambda+\mathbb{Z}) \times \mathbb{N}}$ satisfy

$$L(S_\nu^{-1}, \nu + S_k) \cdot (y_{\nu,k})_{\nu,k} = (q_{\nu,k})_{\nu,k}, \tag{18}$$

where S_ν and S_k are the shift operators on $\mathbb{C}^{(\lambda+\mathbb{Z}) \times \mathbb{N}}$ mapping a double sequence $(u_{\nu,k})$ respectively to $(u_{\nu+1,k})$ and $(u_{\nu,k+1})$.

Proof sketch. By calculating the action of the operators z and θ on logarithmic monomials $z^\nu \log(z)^k / k!$, one checks that the coefficient sequence of $L(z, \theta) \cdot y(z)$ is

$$L(S_\nu^{-1}, \nu + S_k) \cdot (y_{\nu,k})_{\nu,k}.$$

The result follows. $\qquad \square$

Observe that, restricted to $k \geqslant k_0$ where k_0 is the largest index for which the $u_{\nu,k}$ and $q_{\nu,k}$ are not all zero, (18) reduces to the univariate recurrence $L(S_\nu^{-1}, \nu) \cdot (y_{\nu,k_0})_\nu = (q_{\nu,k_0})_\nu$. In particular, if $y(z)$ and $q(z)$ are plain power series ($k_0 = 0$, $\lambda = 0$), we get the formula of Proposition 2.

In the general case, we can put the "implicit" equation (18) into "solved" form as follows. Consider again an operator $L(z, \theta) \in \mathbb{C}[[z]][\theta]$, and, similar to what we did in Section 4.2, write

$$L(z, \theta) = \sum_{j=0}^{\infty} Q_j(\theta) z^j.$$

Note that Q_0 coincides with the indicial polynomial introduced above. Let $\mu(\nu)$ denote the multiplicity of a complex number ν as a root of Q_0. To simplify the notation somewhat, we limit ourselves here to right-hand sides of the form $Q_0(\theta) \cdot q(z)$.

Corollary 6. *Assume that $y(z), q(z) \in z^\lambda \mathbb{C}[[z]][\log z]$ satisfy $L(z, \theta) \cdot y = Q_0(\theta) \cdot q$, and define $y_{\nu,k}$ and $q_{\nu,k}$ as in Proposition 5. Let $\kappa(n) \geqslant 0$ be such that $q_{\lambda+n,k} = 0$ for all $k \geqslant \kappa(n)$. Let $\tau(n) \geqslant 0$ be such that $y_{\lambda,k} = 0$ for $k \geqslant \tau(0)$, and*

$$\tau(n) \geqslant \max\left(\kappa(n), \tau(n-1) + \mu(\lambda+n)\right), \qquad n \geqslant 1. \tag{19}$$

Then, the coefficients of $y(z)$ satisfy $y_{\lambda+n,k} = 0$ for all $k \geqslant \tau(n)$, and are given by

$$y_{\lambda+n,\atop \mu(\lambda+n)+k} = q_{\lambda+n,\atop \mu(\lambda+n)+k} - \sum_{j=1}^{n} \sum_{t=0}^{\tau(n-j)-1} [X^t] \frac{Q_j(\lambda+n+X)}{X^{-\mu(\lambda+n)} Q_0(\lambda+n+X)} y_{\lambda+n-j,\atop k+t}, \tag{20}$$

for all $n \in \mathbb{Z}$ and $k \in \mathbb{N}$. (Both sums can be extended to range from the indicated lower index to infinity.)

Conversely, any solution $(y_{\lambda+n,k})$ of (20) with $y_{\lambda+n,k} = 0$ for $k \geqslant \tau(n)$ is the coefficient sequence of a solution of $L(z, \theta) \cdot y = Q_0(\theta) \cdot q$.

In general, to satisfy (19), one can take

$$\tau(n) = \max_{0 \leqslant i \leqslant n} \left(\kappa'(i) + \sum_{j=i+1}^{n} \mu(\lambda+j) \right) \quad \text{where} \quad \kappa'(i) = \begin{cases} \kappa'(0) = \tau(0), \\ \kappa'(i) = \kappa(i), & i \geqslant 1. \end{cases} \tag{21}$$

Proof. Consider the equation

$$L(S_\nu^{-1}, \nu + S_k) \cdot (y_{\nu,k}) = Q_0(\nu + S_k) \cdot (q_{\nu,k})$$

which results from Proposition 5. By extracting the sequence term of index ν on both sides, we get a relation between sequences $y_{\nu-j} = (y_{\nu-j,k})_{k \in \mathbb{N}}$ and $q_\nu = (q_{\nu,k})_{k \in \mathbb{N}}$ of a single index k:

$$Q_0(\nu + S_k) \cdot y_\nu = Q_0(\nu + S_k) \cdot q_\nu - \sum_{j=1}^\infty Q_j(\nu + S_k) \cdot y_{\nu-j}. \tag{22}$$

Let us first prove by induction that for all n, the sequence y_ν where $\nu = \lambda + n$ vanishes under the action of $S_k^{\tau(n)}$. This is the case by assumption for $n \leqslant 0$. Now assume that this holds true for all $n' < n$. Write $Q_0(\nu + X) = X^{\mu(\nu)} \tilde{Q}_0(X)$ with $Q_0(0) \neq 0$. Note that $\tau(n) \geqslant \mu(\nu)$ by definition of τ. Applying $S_k^{\tau(n)-\mu(\nu)}$ to (22) and using the induction hypothesis gives $\tilde{Q}_0(S_k) S_k^{\tau(n)} \cdot (y_\nu - q_\nu) = 0$. Since $\tau(n) \geqslant \kappa(n)$, we have $S_k^{\tau(n)} \cdot q_\nu = 0$, and hence $\tilde{Q}_0(S_k) S_k^{\tau(n)} \cdot y_\nu = 0$. But, by assumption, $y(z)$ belongs to $z^\lambda \mathbb{C}[[z]][\log z]$, so that $S_k^{\tau(n)} \cdot y_\nu$ is zero almost everywhere. As $Q_0(0) \neq 0$, the term of index k is a linear combination of those of index $k' > k$, therefore $S_k^{\tau(n)} \cdot y_\nu = 0$.

Now, the polynomial $X^{-\mu(\nu)} Q_0(\nu + X) \in \mathbb{C}[X]$ is invertible in $\mathbb{C}[X] / \langle X^{\tau(n)-\mu(\nu)} \rangle$. Let $A_\nu(X) \in \mathbb{C}[X]$ denote the canonical representative of its inverse, and write

$$A_\nu(X) Q_0(\nu + X) = X^{\mu(\nu)} + U_\nu(X) X^{\tau(n)}.$$

Considering again (22) and applying $A_\nu(S_k)$, we get

$$\left(S_k^{\mu(\nu)} + U_\nu(S_k) S_k^{\tau(n)}\right) \cdot y_\nu = \left(S_k^{\mu(\nu)} + U_\nu(S_k) S_k^{\tau(n)}\right) \cdot q_\nu - \sum_{j=1}^\infty A_\nu(S_k) Q_j(\nu + S_k) \cdot y_{\nu-j}. \tag{23}$$

We have seen that $S_k^{\tau(n)} \cdot y_\nu = S_k^{\tau(n)} \cdot q_\nu = 0$, so that both terms involving $U(S_k)$ vanish, and (23) reduces to

$$S_k^{\mu(\nu)} \cdot y_\nu = S_k^{\mu(\nu)} \cdot q_\nu - \sum_{j=1}^\infty A_\nu(S_k) Q_j(\nu + S_k) \cdot y_{\nu-j}.$$

Finally, for $j \geqslant 1$, the sequence $S_k^{\tau(n-j)} \cdot y_{\nu-j}$ is zero, and $A_\nu(X) Q_j(\nu + X)$ coincides at least to the order $\tau(n) - \mu(\nu) \geqslant \tau(n-j)$ with the series $X^{\mu(\nu)} Q_j(\nu + X) / Q_0(\nu + X)$. Equation (20) then follows by extracting the coefficient of index k.

If, conversely, $(y_{\lambda+n,k})$ is a solution of (20) such that $S_k^{\tau(n)} \cdot y_{\lambda+n} = 0$ for all n, then it satisfies (23), and hence (22) because the sequences y_ν and q_ν have finitely many nonzero coefficients and $A_\nu(0) \neq 0$. Equation (22) is equivalent to $L(S_\nu^{-1}, \nu + S_k) \cdot (y_{\nu,k}) = (q_{\nu,k})$, and to $L(z, \theta) \cdot y = Q_0(\theta) \cdot q$ by Proposition 5. $\qquad\square$

Remark 7. While the recurrence (20) giving the whole sequence y_ν "at once" is useful in the context of bound computations, specializing (22) without "inverting" Q_0 yields an alternative expression of $y_{\nu,\mu(\nu)+k}$, as a function of the $y_{\nu-j}$, $j \geqslant 1$, and the $y_{\nu,k'}$, $k+1 \leqslant k' < \mu(\nu)$, which can be better adapted to the iterative computation of the $y_{\nu,k}$.

Suppose now that $L(z, \theta)$ is regular at 0. Let

$$E = \{(\nu, k) : 0 \leqslant k < \mu(\nu)\} \subset \mathbb{C} \times \mathbb{N}.$$

Thus E is a set of cardinality r. As already mentioned, it is well known that the homogeneous equation $L(z, \theta) \cdot y = 0$ has an r-dimensional vector space of convergent solutions spanned by elements of $z^\lambda \mathbb{C}[[z]][\log z]$ for $Q_0(\lambda) = 0$. Corollary 6 suggests a specific choice of basis, giving rise to a dual notion of "initial values".

Corollary 8. *A solution $y(z)$ of $L(z, \theta) \cdot y = 0$ is uniquely determined by the vector of* generalized initial values $(y_{\nu,k})_{(\nu,k) \in E}$.

Proof sketch. According to Corollary 6, the coefficient $y_{\nu,k}$ of a solution $y \in z^\lambda \, \mathbb{C}[[z]][\log z]$ is determined by the $y_{\nu-j,k}$, $j \geqslant 1$, as soon as $k \geqslant \mu(n)$. In contrast, (20) imposes no constraint on the $y_{\nu,k}$ with $(\nu, k) \in E$, and it is not too hard to see that taking $y_{\nu_0,k_0} = 1$ for a certain $(\nu_0, k_0) \in E$ and $y_{\nu,k} = 0$ for $(\nu, k) \in E \backslash \{(\nu_0, k_0)\}$ determines a solution $(y_{\nu,k})_{\nu \in \nu_0+\mathbb{Z}, k \in \mathbb{N}}$ such that $y_{\nu,k} = 0$ for $k \geqslant \tau(\nu)$. The collection of the logarithmic series associated to these sequences for all choices of (ν_0, k_0) is a basis of the solution space of $L(z, \theta)$. $\qquad\square$

For each $(\nu_0, k_0) \in E$, Corollary 6 applied with $\lambda = \nu_0$, $\kappa \equiv 0$ and $\tau(0) = k_0$ provides a bound of the form (21) on the degree with respect to $\log(z)$ of the coefficients of the corresponding element of this basis. One gets a uniform bound valid for all coefficients of all solutions $y \in z^{\lambda_0} \, \mathbb{C}[[z]][\log z]$ by taking $\lambda = \lambda_0 - 1$, $\kappa \equiv 0$, $\tau(0) = 0$, and letting n tend to infinity.

5.3. The majorant equation. Equipped with these preliminaries, it is now easy to extend the method of Section 4.2 to logarithmic series solutions. The next proposition, in some sense, reduces the problem of bounding the solutions of arbitrary regular ODEs with polynomial coefficients to the case of first-order equations. Aiming for first-order majorant equations makes it possible to write down the solutions explicitly while leaving us with enough freedom to match the analytic behavior of solutions of the original equation. Other choices are possible though, and the method could in principle be adapted to other forms of majorant equations.

We still consider an operator $L(z, \theta) = \sum_{j \geqslant 0} Q_j(\theta) \, z^j$ and a solution $y(z)$ of

$$L(z, \theta) \cdot y(z) = Q_0(\theta) \cdot q(z)$$

with $y, q \in z^\lambda \, \mathbb{C}[[z]]$ for a fixed $\lambda \in \mathbb{C}$, and define $\mu(\nu)$, E, $y_{\nu,k}$ and $q_{\nu,k}$ as above.

Proposition 9. *Fix $\hat{a}(z), \hat{q}(z) \in z \, \mathbb{R}_+[[z]]$, and let $\hat{y}(z) \in \mathbb{R}[[z]]$ be a solution of*

$$z \, \hat{y}'(z) = \hat{a}(z) \, \hat{y}(z) + \hat{q}(z). \tag{24}$$

Assume, additionally, that the following assertions hold for a certain $n_0 \geqslant 1$:

(i) for all $j \geqslant 1$ and all $n \geqslant n_0$,

$$n \sum_{t=0}^{\tau(n)-1} \left| [X^t] \frac{Q_j(\lambda+n+X)}{X^{-\mu(\lambda+n)} \, Q_0(\lambda+n+X)} \right| \leqslant \hat{a}_j, \tag{25}$$

where $\tau(n)$ is a non-decreasing sequence such that $y_{\lambda+n,k} = 0$ for $k \geqslant \tau(n)$,

(ii) for all $n \geqslant n_0$ and all $k \geqslant 0$,

$$n \, |q_{\lambda+n,\mu(\lambda+n)+k}| \leqslant \hat{q}_n,$$

(iii) $|y_{\lambda+n,k}| \leqslant \hat{y}_n$ for $0 \leqslant n < n_0$ and for all $k \geqslant 0$,

(iv) $|y_{\lambda+n,k}| \leqslant \hat{y}_n$ for $n \geqslant n_0$ and $0 \leqslant k < \mu(\lambda+n)$.

We then have the bound

$$|y_{\lambda+n,k}| \leqslant \hat{y}_n$$

for all $n, k \geqslant 0$. In other words, $\hat{y}(z)$ is a majorant series (in the extended sense) of $z^{-\lambda} y(z)$.

Note in particular that, when $\mu(n)$ is zero, (25) can also be written $\sum_{t=0}^{\tau(n)-1} \left| \frac{n}{t!} \, f^{(t)}(n) \right| \leqslant \hat{a}_j$ where $f(n) = (Q_j/Q_0)(\lambda+n)$. When additionally $\tau(n) = 1$, it reduces to $|n \, f(n)| \leqslant \hat{a}_j$.

Proof. We prove the result by induction on n. The inequality holds for $n < n_0$ by assumption (iii). Fix $n \geqslant n_0$ (in particular, $n > 0$), and assume that $|y_{\lambda+n',k}| \leqslant \hat{y}_{n'}$ for all $n' < n$ and $k \geqslant 0$. Denote $m = \mu(\lambda + n)$. For $k < m$, the required inequality is given by assumption (iv). Now consider $y_{\lambda+n,m+k}$ for some $k \geqslant 0$. By Corollary 6 applied to the equation $L(z, \theta) \cdot y = Q_0(\theta) \cdot q$, we have

$$y_{\lambda+n,\atop m+k} = q_{\lambda+n,\atop k+t} - \sum_{j=1}^{\infty} \sum_{t=0}^{\tau(n)-1} [X^t] \frac{Q_j(\lambda+n+X)}{X^{-m} \, Q_0(\lambda+n+X)} \cdot y_{\lambda+n-j,\atop k+t}.$$

It follows that

$$\left|y_{\lambda+n,}\atop{m+k}\right| \leqslant \left|q_{\lambda+n,}\atop{m+k}\right| + \sum_{j=1}^{\infty}\sum_{t=0}^{\tau(n)-1}\left|[X^t]\frac{Q_j(\lambda+n+X)}{X^{-m}Q_0(\lambda+n+X)}\right|\cdot\left|y_{\lambda+n-j,}\atop{k+t}\right|,$$

and hence, using assumptions (i)–(ii) and the induction hypothesis,

$$|y_{\lambda+n,}\atop{m+k}| \leqslant \frac{\hat{q}_n}{n} + \sum_{j=1}^{\infty}\frac{\hat{a}_j}{n}\,\hat{y}_{n-j}. \tag{26}$$

Extracting the coefficient of z^n on both sides of the differential equation (24) yields

$$n\,\hat{y}_n = \hat{q}_n + \sum_{j=1}^{\infty}\hat{a}_n\,\hat{y}_{n-j}, \tag{27}$$

so that (26) becomes $|y_{\lambda+n,m+k}| \leqslant \hat{y}_n$. This completes the induction step and the proof. $\qquad\square$

The general solution of (24) reads

$$\hat{y}(z) = \hat{h}(z)\left(c + \int_0^z\frac{w^{-1}\,\hat{q}(w)}{\hat{h}(w)}\,\mathrm{d}w\right), \qquad \hat{h}(z) = \exp\int_0^z w^{-1}\,\hat{a}(w)\,\mathrm{d}w$$

for an arbitrary constant c. Observe that $\hat{h}(z)$ has valuation zero, and hence $\hat{y}(z)$ either has valuation zero as well (if $c \neq 0$) or has the same valuation as $\hat{q}(z)$ (if $c = 0$).

Conditions (i) and (ii) in Proposition 9 ensure that the solutions of (24) can be used to control those of the equation $L(z,\theta)\cdot y = q$. For the proposition to be applicable, $\hat{a}(z)$ and $\hat{q}(z)$ should be chosen so that these conditions hold.

Conditions (iii) and (iv) depend on the specific solution y we are interested in. They express that the bound holds "initially", and most importantly for the generalized initial values described in Corollary 8 (which are not determined by the "previous terms" of the series). In particular, condition (iii) becomes trivial if $n_0 \leqslant \mathrm{val}(z^{-\lambda}y(z))$, where $\mathrm{val}(f)$ is the valuation of f viewed as an element of $\mathbb{C}[\log z][[z]]$. As for condition (iv), it vanishes as soon as n_0 is larger than all the zeros n of $Q_0(\lambda + n)$, or even than the zeros of $Q_0(\lambda + n)$ corresponding to nonzero generalized initial values in $y(z)$.

Of special interest is the situation where both (iii) and (iv) are automatically satisfied. This happens in particular when the valuation of $z^{-\lambda}y(z)$ is larger than that of any $h(z) \in \mathbb{C}((z))$ such that $z^\lambda h(z)$ satisfies the homogeneous part $L(z,\theta)\cdot(z^\lambda h(z)) = 0$ of the differential equation on y (i.e., larger than $\max(\{\nu - \lambda : (\nu, k) \in E\} \cap \mathbb{N})$), and n_0 is chosen between these two values.

Even in the general case, assuming $\hat{a}_1 > 0$, it is always possible to adjust c or increase selected coefficients of $\hat{q}(z)$ so that (iii) and (iv) hold. Therefore, for any choice of n_0, the other parameters can be selected in such a way that Proposition 9 yields a nontrivial bound.

6. The main algorithm

Let us now go back to homogeneous equations with polynomial coefficients, and specialize the previous results to develop an algorithm that computes bounds on remainders of logarithmic series solutions in this case. The description of the algorithm is split into two parts. The first part, Algorithm 10, does not depend on the particular solution or truncation order. Roughly speaking, it aims at satisfying condition (i) of Proposition 9, which yields what one might call a "pseudo-inverse bound" on the differential operator. The second part, Algorithm 19, uses the result of the first one to bound a specific remainder of a specific solution.

6.1. Setting. For the whole of this section, $P(z, \theta) \in \mathbb{K}[z][\theta]$ denotes a differential operator, assumed to be regular at the origin, over a fixed number field $\mathbb{K} \subset \mathbb{C}$. We consider a solution

$$u(z) = z^\lambda\sum_{n=0}^{\infty}\sum_{k=0}^{K-1}u_{\lambda+n,k}\,z^n\,\frac{\log(z)^k}{k!} \tag{28}$$

of the homogeneous equation $P(z, \theta) \cdot u(z) = 0$, and a truncation

$$\tilde{u}(z) = z^\lambda \sum_{n=0}^{N-1} \sum_{k=0}^{K-1} u_{\lambda+n,k} \, z^n \frac{\log(z)^k}{k!} \tag{29}$$

of the logarithmic series expansion of u.

Our goal is to give an algorithm to bound the error $u - \tilde{u}$, based on Proposition 9. Note that a solution of $P(z, \theta)$ can in general be a linear combination of solutions of the form (28) with λ belonging to different cosets in $\mathbb{C}/\mathbb{Z}$. We limit ourselves here to a single λ, and will if necessary bound the remainders corresponding to exponents in other cosets $\lambda' + \mathbb{Z}$ separately.

As in Section 4.2, we also set

$$L(z, \theta) = P(z, \theta) \, p_r(z)^{-1} = \sum_{j=0}^{\infty} Q_j(\theta) \, z^j \quad \text{and} \quad y(z) = p_r(z) \big(\tilde{u}(z) - u(z) \big)$$

where $p_r(z)$ is the leading coefficient of $P(z, \theta)$. Note that $Q_0(\theta)$, the indicial polynomial of $L(z, \theta)$, is monic, and that the indicial polynomial of $P(z, \theta)$ is equal to $p_r(0) \, Q_0(\theta)$. Let $\mu(\nu)$ denote the multiplicity of ν as a root of Q_0, and let $E = \{(\nu, k) : 0 \leqslant k < \mu(\nu)\}$ as usual.

6.2. "Pseudo-inverse bounds" on differential operators. Algorithm 10 below corresponds to the first part of the main bound computation algorithm.

Compared with the outline in Section 4, and beside supporting regular singular points, this version of the algorithm provides more freedom in the choice of the rational function $\hat{a}(z)$, via the input parameter ℓ. The method of Section 4 corresponds to $\ell = 1$. Increasing ℓ leads to tighter bounds, at the price of increased computation time. Taking $\ell \approx s/2$ often gives good results in practice for evaluations far from the border of the disk of convergence of $p_r(z)^{-1}$ (but see Section 9.4 for more on that).

Except for step 3, the algorithm can be run in interval arithmetic and will return finite bounds if all operations are performed with sufficient precision. Besides, the "exact" data needed at step 3, that is the roots of Q_0 in $\lambda + \mathbb{Z}$ and their multiplicities, will typically be known beforehand by the choice of λ.

Algorithm 10. [Bound differential operator]

    Input. A differential operator $P(z, \theta) \in \mathbb{K}[z][\theta]$. An algebraic number λ. Integers $\ell, n_0 > 0$.

    Output. A pair $(\check{p}, \hat{a}) \in \mathbb{R}[z] \times \mathbb{R}(z)$ such that $p_r(z)^{-1} \ll \check{p}(z)^{-1}$ and $\hat{a}(z)$ satisfies condition (i) of Proposition 9.

  1. Compute operators

$$Q(z, \theta) = Q_0(\theta) + \cdots + Q_{\ell-1}(\theta) \, z^{\ell-1}, \qquad U(z, \theta) = U_0(\theta) + \cdots + U_{s-1}(\theta) \, z^{s-1} \quad \in \mathbb{K}[\theta][z]$$

    such that

$$P(z, \theta) = Q(z, \theta) \, p_r(z) + U(z, \theta) \, z^\ell.$$

    (To do so, expand the fractions p_i / p_r in power series up to the order ℓ and set $[\theta^i] \, Q_j$ to $[z^j] \, (p_i / p_r)$. Then compute U_j as the coefficient of $z^{\ell+j}$ in $P(z, \theta) - Q(z, \theta) \, p_r(z)$. If working in approximate arithmetic, force the coefficient of θ^r in Q_0 to the exact value 1 and the coefficients of θ^r in $Q_2, ..., Q_{\ell-1}, U_0, ..., U_{s-1}$ to 0.)

  2. Compute lower bounds $0 < \rho_i \leqslant |\zeta_i|$ on the roots $\zeta_1, \zeta_2, ...$ of $p_r(z)$, and a lower bound c on the absolute value of its leading coefficient. Set

$$\check{p}(z) = c \prod_i (\rho_i - z)^{m_i}$$

    where m_i is the multiplicity of the root ζ_i.

  3. Set $\tau(n) \equiv 1$ if 0 is an ordinary point of $P(z, \theta)$, and $\tau(n) = \sum_{k=0}^{n} \mu(\lambda + n)$, where $\mu(\nu)$ is the multiplicity of ν as a root of Q_0, otherwise.

4. Denoting

$$F(f,n) = n \cdot \sum_{t=0}^{\tau(n)-1} \left| [X^t] \frac{f(\lambda+n+X)}{X^{-\mu(\lambda+n)} Q_0(\lambda+n+X)} \right|, \tag{30}$$

use Algorithm 25 and Remark 26 to compute bounds $\hat{Q}_1, ..., \hat{Q}_{\ell-1}, \hat{U}_0, ..., \hat{U}_{s-1}$ such that

$$F(Q_j, n) \leqslant \hat{Q}_j, \quad 1 \leqslant j < \ell, \qquad \text{and} \qquad F(U_j, n) \leqslant \hat{U}_j, \quad 0 \leqslant j < s,$$

for all $n \geqslant n_0$. Let $\hat{Q}(z) = \hat{Q}_1 z + \cdots + \hat{Q}_{\ell-1} z^{\ell-1}$ and $\hat{U}(z) = \hat{U}_0 + \cdots + \hat{U}_{s-1} z^{s-1}$.

5. Return $(\check{p}, \hat{a})$ where

$$\hat{a}(z) = \hat{Q}(z) + z^\ell \frac{\hat{U}(z)}{\check{p}(z)}.$$

Remark 11. A simple strategy for step 2 (cf. Grégoire [15]) is to compute a single tight lower bound ρ on the smallest root of p_r by the method based on Graeffe iterations of Davenport and Mignotte [9], and take $\check{p}(z) = c\,(\rho - z)^{\deg p}$. This method is fast and typically yields good bounds in simple cases like standard special functions. Unsurprisingly, though, it can also lead to large overestimations for operators with many singularities. At the other extreme, one can call a (rigorous) numerical root finder to obtain arbitrarily tight bounds. While, on paper, this option complexifies the algorithm a lot, very good implementations of complex root-finding exist, so that it works well in practice. It would also likely be feasible to extend the method of Davenport and Mignotte to get better estimates of larger roots.

Proposition 12. *Given an operator $P(z,\theta) = p_r(z)\,\theta^r + \cdots$, regular at 0, Algorithm 10 returns a pair consisting of a polynomial $\check{p} \in \mathbb{R}[z]$ such that $p_r(z)^{-1} \ll \check{p}(z)^{-1}$, and a rational function $\hat{a}(z)$ with $\hat{a}(0) = 0$ that satisfies condition (i) of Proposition 9 for any solution $y \in z^\lambda \, \mathbb{C}[[z]][\log z]$ of $P(z,\theta) \cdot y = 0$.*

Proof. At step 1, the fractions p_k / p_r can be expanded in power series since $p_r(0) \neq 0$. The operator $Q(z,\theta)$ defined by taking $[\theta^k]\,Q_j = [z^k]\,(p_k / p_r)$ for $0 \leqslant j < \ell$ then agrees with $P(z,\theta)\,p_r(z)^{-1}$ up to the order $O(z^\ell)$. In particular, it is monic with respect to θ, so that $[\theta^r]\,Q_0 = 1$ and $\deg Q_j \leqslant r-1$ for $j \geqslant 1$. For the same reason, the difference $P(z,\theta) - Q(z,\theta)\,p_r$ is of the form $U(z,\theta)\,z^\ell$, with $\deg U_j \leqslant r-1$ for all j since $\ell \geqslant 1$. The operator $Q(z,\theta)\,p_r$ has degree (with respect to z) at most $\ell - 1 + s$, and $P(z,\theta)$ has degree at most s, hence the degree of their difference is strictly less than $\ell + s$, and the coefficients we compute correctly represent $U(z,\theta)$.

The polynomials $Q_0, ..., Q_{\ell-1}$ computed here coincide with those defined before by the series expansion $L(z,\theta) = \sum_j Q_j(\theta)\,z^j$. Additionally, the remaining coefficients of the expansion are related to $U(z,\theta)$ by

$$\sum_{j=0}^{\infty} Q_{\ell+j}(Y)\,X^j = U(X,Y)\,p_r(X)^{-1} = \frac{1}{p_r(X)} \sum_{j=0}^{s-1} U_j(Y)\,X^j. \tag{31}$$

After step 2, we have for all i

$$\frac{1}{z - \zeta_i} = \frac{-\zeta_i^{-1}}{1 - \zeta_i^{-1} z} \ll \frac{\rho_i^{-1}}{1 - \rho_i^{-1} z}$$

and hence $p_r(z)^{-1} \ll \check{p}(z)^{-1}$.

By Corollary 6, setting $\tau(n) = \sum_{k=0}^{n} \mu(\lambda + n)$ at step 3 guarantees that $u_{\lambda+n,k} = y_{\lambda+n,k} = 0$ for all $k \geqslant \tau(n)$. When the origin is an ordinary point of $P(z,\theta)$, though, this generic choice only gives $\tau(n) = r$ for $n \geqslant r-1$, but we can take $\tau(n) = 1$ since all solutions of $P(z,\theta) \cdot u = 0$ lie in $\mathbb{C}[[z]]$.

Let us now show that, when the bounds from steps 2 and 4 hold, the coefficients $\hat{a}_j$ of the rational series

$$\hat{a}(z) = \hat{Q}(z) + z^\ell \frac{\hat{U}(z)}{\check{p}(z)}$$

satisfy condition (i) of Proposition 9, that is, $F(Q_j, n) \leqslant \hat{a}_j$ for all $j \geqslant 1$ and $n \geqslant n_0$. The first $\ell - 1$ inequalities of step 4 say that this holds true for the initial coefficients $\hat{a}_1 = \hat{Q}_1, ..., \hat{a}_\ell = \hat{Q}_\ell$. The last s inequalities translate into

$$\sum_{j=0}^{s-1} F(U_j, n) \, z^j \ll \hat{U}(z), \qquad n \geqslant n_0. \tag{32}$$

According to (31), we have for fixed n

$$\sum_{j=0}^{\infty} \frac{Q_{\ell+j}(\lambda + n + X)}{X^{-\mu(\lambda+n)} Q_0(\lambda + n + X)} z^j = \frac{1}{p_r(z)} \sum_{j=0}^{s-1} \frac{U_j(\lambda + n + X)}{X^{-\mu(\lambda+n)} Q_0(\lambda + n + X)} z^j \in \mathbb{C}(X)[z].$$

Using the bound $p_r(z)^{-1} \ll \check{p}(z)^{-1}$ from step 2 (and Proposition 4), it follows that

$$\sum_{j=0}^{\infty} \left| [X^t] \frac{Q_{\ell+j}(\lambda + n + X)}{X^{-\mu(\lambda+n)} Q_0(\lambda + n + X)} \right| z^j \ll \frac{1}{\check{p}(z)} \sum_{j=0}^{s-1} \left| [X^t] \frac{U_j(\lambda + n + X)}{X^{-\mu(\lambda+n)} Q_0(\lambda + n + X)} \right| z^j$$

for all $t \geqslant 0$. By summing over $0 \leqslant t < \tau(n)$ and multiplying by n, then applying (32), we get

$$\sum_{j=0}^{\infty} F(Q_{\ell+j}, n) \, z^j \ll \frac{1}{\check{p}(z)} \sum_{j=0}^{s-1} F(U_j, n) \, z^j \ll \frac{\hat{U}(z)}{\check{p}(z)}, \qquad n \geqslant n_0,$$

and hence finally

$$\sum_{j=0}^{\infty} F(Q_{\ell+j}, n) \, z^j \ll \hat{a}(z), \qquad n \geqslant n_0. \qquad \square$$

Remark 13. In our implementation, the code corresponding to Algorithm 10 actually does not take n_0 as input. Instead, the bounds $\hat{Q}_j$ and $\hat{U}_j$ of step 4 are functions of n_0, and the algorithm returns an object (called a DiffOpBound) that can then be evaluated at a given n_0 to get the pair $(\check{p}, \hat{a})$. The DiffOpBound objects serve to share part of the computation when one needs to bound several tails of the same series—for example, to keep adding terms to the sum until the bound becomes small enough—or tails of several solutions of the same equation.

Similarly, when Algorithm 10 is run on the same operator with increasing values of ℓ, almost all of the previous computations can be reused from one run to the next. We use that in the implementation to provide *refinable* bounds: DiffOpBound objects start off as relatively coarse and inexpensive bounds, and offer a method to increase ℓ if the bounds turn out to be too pessimistic. The switch from the "fast" strategy for step 2 discussed in Remark 11 to the more accurate one works in the same way.

Remark 14. We also experimented with a variant of Algorithm 10 that performs a partial fraction decomposition with respect to z of the operator instead of splitting it into a truncated series and a rational remainder term, but did not manage to obtain better bounds using this approach.

It is not hard to compute a majorant series $\hat{u}(z)$ of $u(z)$ from the output of this algorithm, as illustrated by the following corollary (not used in the sequel).

Corollary 15. *Let $(\check{p}, \hat{a})$ be the output of Algorithm 10 called on $(\mathcal{L}, \lambda, \ell, n_0)$. For any solution $u(z)$ of $P(z, \theta) \cdot u = 0$ lying in $z^\lambda \mathbb{C}[[z]][\log z]$, there is a computable constant $c \in \mathbb{R}$ such that*

$$u(z) \ll \hat{u}(z) = \frac{c}{\check{p}(z)} \exp \int_0^z \left(1 + w^{-1} \hat{a}(w)\right) dw$$

is a majorant series (in the extended sense of Definition 3) of $u(z)$.

Proof. Apply Proposition 9 to the homogeneous equation $L(z, \theta) \cdot p_r(z) \, u(z) = 0$ where $L(z, \theta) = P(z, \theta) \, p_r(z)^{-1}$. By the previous proposition, Algorithm 10 provides a series $\hat{a}(z)$ satisfying the first condition. The same is true a fortiori of $\hat{a}(z) + z$. In addition, the corresponding $\hat{y}(z) = \exp \int_0^z (1 + w^{-1} \, \hat{a}(w)) \, \mathrm{d}w$ has strictly positive coefficients. The second condition is trivial for homogeneous equations. We choose c large enough that $|y_{\lambda+n,k}| \leqslant c \, \hat{y}_n$ for $n < n_0$ or $k < \mu(\lambda+n)$, fulfilling the last two constraints. Proposition 9 then implies $y(z) \ll c \, \hat{y}(z)$, and hence $u(z) = y(z) / p_r(z) \ll \hat{u}(z)$. $\square$

The remainders of $\hat{u}(z)$ are majorants of the corresponding remainders of $u(z)$ and, when $\hat{u}$ converges at $|\zeta|$, provide us with a sequence of bounds on the error $|u(\zeta) - \tilde{u}(\zeta)|$ that tends to zero as the truncation order increases. While the remainders of $\hat{u}(z)$ may not admit nice closed-form expressions, it is not hard to extract reasonably good bounds on their values algorithmically, as discussed in Mezzarobba and Salvy [30] or in Section 8.2 below, thus solving our problem in principle. We can do better, though, by taking into account the residual associated to $u(z)$.

6.3. Normalized residuals. Like with Corollary 6 and Proposition 9, instead of working with the residual $P(z, \theta) \cdot \tilde{u}(z)$, it is convenient to introduce the *normalized residual* $q(z)$ defined by

$$P(z, \theta) \cdot \tilde{u}(z) = Q_0(\theta) \cdot q(z)$$

with the additional condition that $q_{\nu,k} = 0$ for $(\nu, k) \in E$. That the normalized residual is well defined follows from the following lemma. When 0 is an ordinary point and $N \geqslant r$, or more generally when $\tilde{u} \in \mathbb{C}[[z]]$ and $Q_0(n) \neq 0$ for all $n \geqslant N$, the normalized residual is simply the residual with the coefficient of z^n divided by $Q_0(n)$, cf. (10).

Lemma 16. *Given $f \in z^\lambda \, \mathbb{C}[[z]][\log z]$ of degree at most K with respect to $\log(z)$, the differential equation $Q_0(\theta) \cdot q(z) = f(z)$ has a unique logarithmic series solution*

$$q(z) = \sum_\nu \sum_k q_{\nu,k} \, z^\nu \, \frac{\log(z)^k}{k!}$$

such that $q_{\nu,k} = 0$ for all $(\nu, k) \in E$. This solution belongs to $z^\lambda \, \mathbb{C}[[z]][\log z]$, and the coefficient $q_{\nu,k}$ is zero for $k < \mu(n)$ or $k > \mu(n) + K$.

Proof. This is a consequence of Proposition 5, applied to the operator $Q_0(\theta)$ (which has the same associated set E as $P(z, \theta)$) and the right-hand side $f(z)$. More precisely, the recurrence (18) for $\nu \in \lambda + \mathbb{Z}$ reduces in this case to $Q_0(\nu + S_k) \cdot (q_{\nu,k}) = (f_{\nu,k})$, that is,

$$c_{\mu(\nu)} \, q_{\nu,\mu(\nu)+k} = f_{\nu,k} - \sum_{\ell > \mu(\nu)} c_\ell \, q_{\nu,k+\ell} \qquad \text{where} \qquad Q_0(\nu + X) = c_0 + c_1 X + \cdots + c_r X^r. \qquad (33)$$

For all ν, this equation has a (unique) solution $(q_{\nu,k})_k$ with $q_{\nu,k} = 0$ outside of the interval

$$\mu(\nu) \leqslant k < \mu(\nu) + \max\{\ell : f_{\nu,\ell} \neq 0\} = \mu(\nu) + K. \qquad (34)$$

The corresponding logarithmic series $q(z) \in z^\lambda \, \mathbb{C}[[z]][\log z]$ satisfies $Q_0(\theta) \cdot q = f$. By Corollary 8 applied to the homogeneous equation $Q_0(\theta) \cdot q = 0$, it is the only solution of $Q_0(\theta) \cdot q = f$ with $q_{\nu,k} = 0$ for all $(\nu, k) \in E$. $\square$

Now consider the special case of $f(z) = P(z, \theta) \cdot \tilde{u}(z)$ used to define the normalized residual.

Lemma 17. *The logarithmic series $q(z)$ of Lemma 16 corresponding to $f(z) = P(z, \theta) \cdot \tilde{u}(z)$ is of the form*

$$q(z) = \sum_{n=N}^{N+s-1} \sum_{k=0}^{K} q_{\lambda+n,\mu(\lambda+n)+k} \, z^{\lambda+n} \, \frac{\log(z)^k}{k!}, \qquad (35)$$

where $s = \deg_z P(z, \theta)$ and K is the largest power of $\log(z)$ appearing in $\tilde{u}(z)$.

Proof. With the convention that the exponents are ordered by real part, call z-degree of a logarithmic series the maximum exponent of z that appears, z-valuation the minimum one, and log-degree the largest power of $\log(z)$ involved. The application of θ to a term $c\,z^\nu \log(z)^k$ yields a (possibly empty) sum of terms of the same z-degree and at most the same log-degree. Since $Q_0(\theta) \cdot q = L(z, \theta) \cdot \tilde{u}$, where $\tilde{u}$ has z-degree less than $\lambda + N$, the z-degree of $Q_0(\theta) \cdot q$ is less than $\lambda + N + s$. Since we also have $Q_0(\theta) \cdot q = -L(z,\theta) \cdot (u - \tilde{u})$, where $u - \tilde{u}$ is of z-valuation $\lambda + N$ or more, $Q_0(\theta) \cdot q$ has z-valuation at least $\lambda + N$. The same bounds apply to $q(z)$ due to (33) and the requirement that $q_{\nu,k} = 0$ for $(\nu, k) \in E$. By a similar reasoning, the log-degree of $Q_0(\theta) \cdot q$ is at most K. By Lemma 16, this implies that the $q_{\nu,k}$ can only be nonzero for $\mu(\nu) \leqslant k \leqslant \mu(\nu) + K$. $\quad\square$

When computing the series expansion of the solution $u(z)$ using the recurrence from Proposition 2 or its generalization (20), the state one needs to maintain from one iteration to the next is a vector $(u_{\lambda+n-1}, ..., u_{\lambda+n-s})$ where, in general, each entry may be a polynomial in $\log(z)$. The following algorithm details how to compute the normalized residual at a given iteration from these coefficients only. For simplicity, we limit ourselves to the generic case where neither the truncation index nor its next few shifts by integers are roots of the indicial polynomial. Note that the most expensive part is typically the evaluation at $\nu + j$ of the polynomial coefficients of the recurrence and their derivatives (step 2), but these coefficients can be recycled from the iterative computation of the $u_{\lambda,k}$. Additionally, fast algorithms dedicated to the evaluation of polynomials at regularly spaced points [14, 2] are applicable.

Algorithm 18. [Normalized residual]

    Input. An operator $P(z,\theta) \in \mathbb{K}[z][\theta]$ of z-degree s. An algebraic number λ. An integer N such that the indicial polynomial Q_0 of $P(z,\theta)$ does not vanish at any of the points $\lambda + N, ...,$ $\lambda + N + s - 1$. The coefficients $u_{j,k} = u_{\lambda+N-j,k}$ as in (28) of a solution $u(z)$, for $1 \leqslant j \leqslant s$ and $k \geqslant 0$.

    Output. The coefficients $q_{j,k} = q_{\lambda+N+j,k}$, $0 \leqslant j < s$, $k \geqslant 0$, of the normalized residual (35) associated to the truncation $\tilde{u}(z)$ of $u(z)$ defined by (29).

1. Write $P(S_n^{-1}, n) = b_0(n) + b_1(n)\, S_n^{-1} + \cdots + b_s(n)\, S_n^{-s}$. Let $c = p_r(0)$ where $p_r(z)$ is the leading coefficient of $P(z, \theta)$. Let $K = \max_j \max\{k : u_{j,k} \neq 0\}$ (and $K = 1$ if $s = 1$).

2. Compute $b_{i,j,k} = b_i^{(k)}(\lambda + N + j)/k!$ for $0 \leqslant i \leqslant j < s$, $0 \leqslant k \leqslant K$.

3. For $j = 0, ..., s - 1$

    a. For $k = K, K - 1, ..., 0$, compute

$$v_{j,k} = \sum_{j'=1}^{s-j} \sum_{k'=0}^{K-k} b_{j+j',j,k'}\, u_{j',k+k'},$$

$$q_{j,k} = \frac{1}{b_{0,j,0}}\left(c\, v_{j,k} - \sum_{k'=1}^{K-k} b_{0,j,k'}\, q_{j,k+k'} \right).$$

4. Return q.

Proof. The algorithm amounts to the computation of the coefficients $(v_{\nu,k})$ of $v(z) = P(z,\theta) \cdot u(z)$ by application of $P(S_n^{-1}, \nu + S_k)$ to $(u_{\nu,k})$, interleaved with the solution of a triangular linear system that expresses the relation $Q(\nu + S_k) \cdot (q_{\nu,k}) = (v_{\nu,k})$. The table v is filled with $v_{j,k} = v_{\lambda+N+j,k}$, using the relation

$$v_{\nu,k} = \sum_{i=0}^{s} \sum_{k'=0}^{K-k'} \frac{b^{(k')}(\nu)}{k!} u_{\nu-i,k+k'}$$

with $\nu = \lambda + N - j$, and noting that $u_{\nu-i,\cdot} = 0$ when $j' = i - j \leqslant 0$. The factor c accounts for the discrepancy between the monic indicial polynomial Q_0 used in the definition of the normalized residual and the leading coefficient of the recurrence: they are related by $b_0(n) = c\, Q_0(n)$, so that the equation $Q(\nu + S_k) \cdot (q_{\nu,k}) = (v_{\nu,k})$ translates into $\sum_{k'} c^{-1} b_{0,j,k'}\, q_{j,k+k'} = v_{j,k}$. $\quad\square$

6.4. Bounds on tails of differentially finite series. In terms of $L(z, \theta)$ and $y(z)$, the normalized residual we just defined satisfies $L(z, \theta) \cdot y = Q_0(\theta) \cdot q$. Given suitable bounds on the coefficients of L, Proposition 9 applies and provides a bound on $y(z)$. If $(\hat{p}, \hat{a})$ is the pair computed by Algorithm 10, it follows that the truncation error satisfies

$$u(z) - \tilde{u}(z) \ll \frac{\hat{h}(z)}{\check{p}(z)} \int_0^z \frac{w^{-1} \hat{q}(w)}{\hat{h}(w)} \, dw, \qquad \hat{h}(z) = \exp \int_0^z w^{-1} \hat{a}(w) \, dw$$

where $\hat{q}(z)$ is a power series satisfying condition (ii) of Proposition 9. The obvious choice is to take $\hat{q}(z)$ as a polynomial with the same support as $z^{-\lambda} q(z)$. However, choosing $\hat{q}(z)$ as a polynomial multiple of $\hat{h}(z)$ instead makes the integral explicitly computable. Combined with a parameter choice that render the last two conditions of Proposition 9 trivial (cf. the discussion at the end of Section 5.3), this leads to the following algorithm.

Algorithm 19. [Tail majorant]

Input. An operator $\mathcal{L} \in \mathbb{K}[z][D_z]$. An algebraic number λ. An integer $\ell \geqslant 1$. A truncation order $N \geqslant 1$. The coefficients $u_{\lambda+N-1}, u_{\lambda+N-2}, \ldots \in \mathbb{K}[\log z]$ of a solution $u(z)$ of $\mathcal{L} \cdot u = 0$ (the last s coefficients, where s is defined in step 1, are enough).

Output. A majorant series $\hat{u}(z)$ of $u(z) - \tilde{u}(z)$, where $\tilde{u}(z)$ is the order-N truncation of $u(z)$.

1. Rewrite $\mathcal{L}$ in the form $P(z, \theta)$ with θ on the left: compute coefficients $p_0, \ldots, p_r \in \mathbb{K}[z]$ such that $z^\rho \mathcal{L} = \theta^r p_r(z) + \cdots + \theta p_1(z) + p_0$ where $\rho \in \mathbb{Z}$ is such that $p_r(0) \neq 0$. Let $s = \max_k \deg p_k$.

2. Using Algorithm 18, compute the normalized residual

$$q(z) = z^{\lambda+N} \sum_{k=0}^{K-1} f^{[k]}(z) \frac{\log(z)^k}{k!}, \qquad f^{[k]} \in \mathbb{C}[z],$$

associated to $\tilde{u}(z)$.

3. Call Algorithm 10 on $(P(z, \theta), \lambda, \ell, n_0)$ with $n_0 = N$ to compute a pair $(\check{p}, \hat{a})$. Let

$$\hat{h}(z) = \exp \int_0^z w^{-1} \hat{a}(w) \, dw.$$

4. Compute a polynomial $\hat{f}(z) = \hat{f}_0 + \hat{f}_1 z + \cdots + \hat{f}_s z^s$ with $\hat{f}_i \geqslant (N+i) \max_k |f_i^{[k]}|$.

5. Compute the first $s+1$ coefficients of the Taylor expansion of $\hat{h}(z)^{-1}$. Deduce a polynomial $g(z) = g_N z^N + \cdots + g_{N+s} z^{N+s} \in \mathbb{R}[z]$ such that

$$\int_0^z w^{N-1} \frac{\hat{f}(w)}{\hat{h}(w)} \, dw = g(z) + O(z^{N+s+1}).$$

Define $\hat{g}(z) = \hat{g}_N z^N + \cdots + \hat{g}_{N+s} z^{N+s}$ by $\hat{g}_n = \max(0, g_n)$.

6. Return the symbolic expression

$$\hat{u}(z) = \frac{\hat{g}(z)}{\check{p}(z)} \hat{h}(z).$$

In our implementation, the series $\hat{h}(z)$ (actually $\hat{h}(z)/\check{p}(z)$) and $\hat{u}(z)$ are represented by objects of type "hyperexponential majorant" that encode formal power series of the form

$$z^\delta \frac{F(z)}{G(z)} \exp\left(\int_0^z H(w) \, dw \right), \qquad \delta \in \mathbb{N}, \quad F, G \in \mathbb{R}[z], \quad H \in \mathbb{R}(z),$$

where the polynomial $G(z)$ is represented in factored form and $H(z)$ is an unevaluated sum of rational functions, also with factored denominators. With the algorithm of this paper, the sum reduces to two terms (a polynomial part related to $\hat{Q}(z)$ in Algorithm 10, and a rational part of denominator $p_r(z)$), but variants like that of Remark 14 can introduce additional terms. This representation makes it easy to extract numerical bounds from the majorant series, as discussed in Section 8 below.

Proposition 20. *Consider the generalized initial values $(u_{\nu,k})_{(\nu,k)\in E}$ associated to u, and assume that N is at least equal to the largest index of a nonzero initial value, that is,*

$$N \geqslant \max\{n \in \mathbb{N} : \exists k, (\lambda + n, k) \in E \wedge u_{\lambda+n,k} \neq 0\}. \tag{36}$$

Then Algorithm 19 returns an expression representing a series $\hat{u}(z) \in \mathbb{R}_+[[z]]$ such that

$$u(z) - \tilde{u}(z) \ll \hat{u}(z).$$

Proof. Step 1 of the algorithm consists in rewriting the equation $\mathcal{L} \cdot u = 0$ into an equivalent one of a special form. We have seen in Section 3.2 that this can always be done, and in Section 5.1 that, due to the assumption that 0 is a regular point, the resulting p_r does not vanish at 0. By Lemmas 16 and 17, the normalized residual $q(z)$ is well defined and the polynomials $f^{[k]}$ have degree bounded by s. Additionally, the rational function returned by Algorithm 10 satisfies $\hat{a}(0) = 0$ (by Proposition 12), so that $\hat{h}(z)$ is analytic at the origin. Thus, at step 5, the computation of $g(z)$ amounts to that of a Taylor expansion of $\hat{a}(z)$ (a rational function with no pole at the origin) followed by routine operations on truncated formal power series. Therefore, the algorithm makes sense and runs without error.

To see that the result satisfies the claim, let us check that Proposition 9 is applicable to $\hat{a}(z)$ and $\hat{q}(z) = z\,\hat{g}'(z)\,\hat{h}(z)$, where $\hat{g}(z)$ is the polynomial computed at step 5. By Proposition 12, the series $\hat{a}(z)$ satisfies condition (i). By definition of $g(z)$, we have

$$g'(z)\,\hat{h}(z) = z^{N-1}\,\hat{f}(z) + \mathrm{O}(z^{N+s}).$$

Since $\hat{f}(z)$ has degree s, the first $N + s$ coefficients of $g'(z)\,\hat{h}(z)$ are exactly the coefficients of the polynomial $z^{N-1}\,\hat{f}(z)$. While these coefficients are non-negative, this may not be true of the remaining coefficients of $g'(z)\,\hat{h}(z)$. However, replacing $g(z)$ by $\hat{g}(z)$ yields a product $\hat{g}'(z)\,\hat{h}(z)$ with

$$\max\left(0, [z^n]\,(g'(z)\,\hat{h}(z))\right) \leqslant [z^n]\,(\hat{g}'(z)\,\hat{h}(z)), \qquad n \geqslant 0.$$

We thus have

$$z^{N-1}\,\hat{f}(z) \ll \hat{g}'(z)\,\hat{h}(z),$$

and hence $z^N\,\hat{f}(z) \ll \hat{q}(z)$. In combination with the inequalities resulting from steps 2 and 4, this implies

$$(N+i)\,|q_{\lambda+N+i,k}| \leqslant \hat{f}_i \leqslant \hat{q}_{N+i}, \qquad i, k \geqslant 0,$$

and condition (ii) is satisfied. Because $N \geqslant n_0$ and $y(z) = p_r(z)\,y(z)$, we have $y_{\lambda+n,k} = 0$ for all $n < n_0$, hence condition (iii) holds. Similarly, condition (iv) holds due to the assumption (36).

Therefore, the relation $y(z) \ll \hat{y}(z)$ holds for *any* solution of $z\,\hat{y}'(z) = \hat{a}(z)\,\hat{y}(z) + \hat{q}(z)$, and in particular for

$$\hat{y}(z) = \hat{h}(z) \int_0^z \frac{w^{-1}\,\hat{q}(w)}{\hat{h}(w)}\,\mathrm{d}w = \hat{h}(z)\,\hat{g}(z).$$

It follows that $u(z) - \tilde{u}(z) = p_r(z)^{-1}\,y(z) \ll \check{p}(z)\,\hat{y}(z).$ $\qquad\qquad\square$

Remark 21. Algorithm 19 is but one way to compute remainder bounds based on Proposition 9. It admits many variants that use the additional flexibility of the framework of the previous section.

1. If N is replaced by n_0 in (36), the proof of Proposition 20 applies verbatim to any $N \geqslant n_0$ instead of only $N = n_0$. One can thus compute majorants corresponding to multiple truncation orders without running Algorithm 10 again, or even specializing again the majorant sequence of Remark 13.

2. It may also happen that (36) fails to hold: typically, if the indicial polynomial Q_0 has a root at $\lambda + n_1$ for some very large n_1, one may want to truncate the series expansion of $u(z)$ at an order $N \ll n_1$ even if one of the generalized initial values at $\lambda + n_1$ is nonzero. One can modify Algorithm 19 with a more general choice of $\hat{y}(z)$, as discussed at the end of Section 5.3, so as to cover this case.

3. There is no need to run the algorithm several times to bound the tails of several solutions of the same equation (corresponding to the same λ) truncated at the same order: it is enough to compute each of the corresponding normalized residuals, and modify step 4 to take them all into account. In particular, using the fact that derivatives of majorants are majorants of derivatives, bounding the remainder of a *fundamental matrix* of the equation at an ordinary point only requires a single call to Algorithm 19.

4. It would be possible to choose the quantity $\tau(n)$ needed at step 4 of Algorithm 10 in a slightly tighter way, based on (21) and the observed degree of the normalized residual after step 2 of Algorithm 19. One would then recover the special case of ordinary points hard-coded in Algorithm 10. However, proceeding this way complicates the reuse of computations when n_0 varies.

5. Step 5 of Algorithm 19 is optional: taking $\hat{g}(z) = \int_0^z w^{N-1} \hat{f}(w)\, \mathrm{d}w$ also yields a valid, if coarser, bound. Indeed, we have $\hat{f}(z) \ll \hat{f}(z)\, \hat{h}(z)$ since $\hat{h}_0 = 1$, meaning that we can replace $\hat{f}(z)$ by $\hat{f}(z)\, \hat{h}(z)$ without contradicting the inequality from step 4, but then the integral at step 5 reduces to $\int_0^z w^{N-1} \hat{f}(w)\, \mathrm{d}w$.

7. Bounds on rational sequences

The main algorithm presented in the previous section crucially relies on bounds on quantities of the form $\sup_{n \geqslant n_0} |f(n)|$, where $f(n)$ is a rational function with complex coefficients. While it is not hard to come up with such bounds (isolating the poles and local extrema of f yields optimal bounds), their computation can easily become costly in practice. This section describes an approach that we found to be a good trade-off between speed and quality for an implementation based on interval arithmetic.

7.1. The generic case. When n is large (larger than any of the poles of the rational function f of interest, at least), a simple and effective way to bound $|f(n)|$ for $n \geqslant n_0$ is to perform an interval evaluation.

More precisely, if $f(n) = p(n)/q(n)$, write $f(x^{-1}) = x^\delta\, \bar{p}(x)/\bar{q}(x)$, where $\bar{p}, \bar{q}$ are the reciprocal polynomials of p, q, and $\delta \geqslant 0$. The evaluation of this expression on $\boldsymbol{x} = [0, 1/n_0]$ in interval arithmetic yields a bound on $f(n)$ valid for all $n \geqslant n_0$. Moreover, this bound converges to the same limit as $|f(n)|$ as n tends to infinity. Note that the rewriting step is essential, as illustrated by the example of $f(n) = (n-1)/n$, whose naive interval evaluation on $[10, \infty]$ gives $[0, \infty]$, to be compared with $[0.9, 1]$ after rewriting the expression.

A big advantage of this approach is that it generalizes naturally to the simultaneous computation of bounds on several derivatives of $f(n)$, as required by step 4 of Algorithm 10. The generalization is formalized as Algorithm 22 below. Given $f = p/q \in \mathbb{C}(n)$ and $T \geqslant 1$, we denote

$$F^{[T]}(f, n) = \sum_{t=0}^{T-1} \left| n\, [\varepsilon^t] \left(\frac{p(n+\varepsilon)}{q(n+\varepsilon)} \right) \right| \tag{37}$$

(this is not exactly the same notation as in Algorithm 10). This definition already covers the bounds on rational sequences needed in the main algorithm when n_0 is larger than the roots of the indicial polynomial. Algorithm 22 also accepts a parameter Z that can make it "ignore" some of the poles of f, and will be useful when we turn to the remaining (non-generic) cases.

In this algorithm and the next one, boldface letters stand for intervals or polynomials with real or complex interval coefficients. All interval operations are extended in a natural way to handle intervals containing ∞. An expression like "compute $\boldsymbol{g}(\varepsilon) = \varphi(\varepsilon) + \mathrm{O}(\varepsilon^T)$" means "compute a polynomial $\boldsymbol{g}$ of degree at most $T-1$ whose coefficients are interval enclosures of the first T Taylor coefficients of $\varphi(\varepsilon)$". The computation reduces to routine operations on truncated power series.

Algorithm 22. [Bound rational sequence, generic case]

   Input. A monic polynomial $q \in \mathbb{C}[n]$ of degree d, a polynomial $p \in \mathbb{C}[n]$ of degree strictly less than d, an integer $T \geqslant 1$, a finite set of "exceptions" $Z \subseteq \mathbb{N}$, a starting index $n_0 \in \mathbb{N} \backslash Z$.

   Output. A bound $M \in \mathbb{R}_+ \cup \{\infty\}$ such that $F^{[T]}(p/q, n) \leqslant M$ for all $n \geqslant n_0$ with $n \notin Z$.

   1. Set $\boldsymbol{x}$ to an interval containing $[0, n_0^{-1}]$ (if $n_0 = 0$, set $\boldsymbol{x} = [0, \infty]$).

2. Compute $\boldsymbol{i}(\varepsilon) = (1 + \boldsymbol{x}\,\varepsilon)^{-1} + \mathrm{O}(\varepsilon^T)$ and $\boldsymbol{j}(\varepsilon) = \boldsymbol{x}\,\boldsymbol{i}(\varepsilon)$.

3. Compute $\bar{\boldsymbol{p}}(\varepsilon) = \bar{p}(\boldsymbol{j}(\varepsilon)) + \mathrm{O}(\varepsilon^T)$ where $\bar{p}(n) = n^{d-1}\,p(n^{-1})$.

4. Compute $\bar{\boldsymbol{q}}(\varepsilon) = \bar{q}(\boldsymbol{j}(\varepsilon)) + \mathrm{O}(\varepsilon^T)$ where $\bar{q}(n) = n^d\,q(n^{-1})$.

5. [Optional; alternatively, set $\boldsymbol{\gamma} = 1$.] Use Lemma 23 below to compute a lower bound $\rho \geqslant 0$ on $|n^{-d}\,q(n)|$ valid for all $n \in \mathbb{N}_{\geqslant n_0} \backslash Z$. Let $\boldsymbol{\gamma}$ be a complex interval of radius $\geqslant \rho^{-1}$ centered at 0. Multiply $\bar{\boldsymbol{q}}(\varepsilon)$ by $\boldsymbol{\gamma}$, and replace the constant coefficient of the result by 1.

6. Compute $\boldsymbol{s}(\varepsilon) = \boldsymbol{s}_0 + \boldsymbol{s}_1\,\varepsilon + \cdots + \boldsymbol{s}_{T-1}\,\varepsilon^{T-1} = \big(\boldsymbol{i}(\varepsilon)\,/\,\bar{\boldsymbol{q}}(\varepsilon)\big)\,\bar{\boldsymbol{p}}(\varepsilon) + \mathrm{O}(\varepsilon^T)$, with the convention that $\infty \in \boldsymbol{s}_0$ if the constant coefficient of $\bar{\boldsymbol{q}}$ contains zero.

7. Return the right endpoint of the interval $\boldsymbol{\gamma}\,(|\boldsymbol{s}_0| + |\boldsymbol{s}_1| + \cdots + |\boldsymbol{s}_{T-1}|)$.

Before proving that this algorithm works as stated, let us discuss the step marked as optional. Without this step, the algorithm is a direct generalization of the method for $T = 1$ sketched above. It returns an infinite bound as soon as q has a real root in $[n_0, \infty)$, and only gives satisfying results when n_0 is sufficiently larger than the largest real root. Note that this may be enough in the context of solutions of differential equations at ordinary points, since the only denominator that arises is then $n\,(n-1)\cdots(n-r+1)$, where r is the order of the equation. In the general case, the next lemma offers a simple way to mitigate the issue.

Lemma 23. *Let $q \in \mathbb{C}[n]$ be a monic polynomial of degree d, and let $n_0 \geqslant 1$. Given $\alpha \in \mathbb{C}$, denote*

$$\pi(\alpha) = \lceil\,|\alpha|^2 / \mathrm{Re}(\alpha)\,\rceil, \qquad b_\alpha(n) = \begin{cases} \min\left(\left|1 - \dfrac{\alpha}{\pi(\alpha)}\right|, \left|1 - \dfrac{\alpha}{\pi(\alpha) - 1}\right|\right), & n < \pi(\alpha), \\[2mm] \left|1 - \dfrac{\alpha}{n}\right|, & n \geqslant \pi(\alpha). \end{cases}$$

Then, for all $n \geqslant n_0$, we have

$$|q(n)| \geqslant n^d \prod_\alpha |b_\alpha(n_0)|$$

where the product is over the multi-set of roots α of q with $\mathrm{Re}\,\alpha > 0$.

Proof. Write $q(n) = n^d \prod_\alpha |1 - \alpha/n|$. When $\mathrm{Re}\,\alpha \leqslant 0$, the sequence $|1 - \alpha/n|$ decreases to 1. Otherwise, it first decreases to a minimum (which may be 0 if α is an integer) and then increases to 1. In the latter case, the minimum is reached for either $n = \pi(\alpha)$ or $n = \pi(\alpha) - 1$. $\qquad\square$

Note that this lower bound can be somewhat expensive to compute compared to the rest of the algorithm. For this reason, our implementation actually decides whether to run the optional step based on the accuracy of the interval $\bar{\boldsymbol{q}}(0)$ at the beginning of step 5.

In practice, there is no need for the check that $\mathrm{Re}\,\alpha > 0$ to be exact, since false positives can only decrease the result. Also, rough enclosures of the roots of q are sufficient: one can add more terms to the minimum in the definition of $b_\alpha(n)$ if the enclosure of α does not uniquely determine $\pi(\alpha)$. Similarly, given $Z \subset \mathbb{N}$, one obtains a lower bound valid for $n \geqslant n_0$ with $n \notin Z$ by replacing $\pi(\alpha)$ by the adjacent integers when $\alpha \in Z$.

Proposition 24. *Algorithm 25 returns a quantity $M \in \mathbb{R}_+ \cup \{\infty\}$ such that $F^{[T]}(p\,/\,q, n) \leqslant M$ for all $n \geqslant n_0$ with $n \notin Z$. The version that includes the optional step returns a finite bound as soon as Z contains all the roots of q in $\mathbb{N}_{\geqslant n_0}$ (provided that the working precision for interval operations is large enough).*

Proof. When $n_0 = 0$, the algorithm returns ∞. Assume $n_0 \geqslant 1$. Fix $n \geqslant n_0$ such that $q(n) \neq 0$, and let $x = n^{-1}$. The quantities $\bar{p}$ and $\bar{q}$ defined in the algorithm are polynomials in ε (since $d = \deg q > \deg p$) with complex coefficients. Letting

$$j(\varepsilon) = \frac{1}{n + \varepsilon} = \frac{x}{1 + x\,\varepsilon} \in \mathbb{R}[[\varepsilon]],$$

they satisfy

$$n\, f(n+\varepsilon) = n\, \frac{p(n+\varepsilon)}{q(n+\varepsilon)} = n\, j(\varepsilon)\, \frac{\bar{p}(j(\varepsilon))}{\bar{q}(j(\varepsilon))} = \frac{1}{1+x\,\varepsilon}\, \frac{\bar{p}(j(\varepsilon))}{\bar{q}(j(\varepsilon))}.$$

Since $q(n) \neq 0$, one can compute the first T Taylor coefficients of $n\,f(n+\varepsilon)$ by evaluating the right-hand side of this expression in $\mathbb{C}[[\varepsilon]]$ and truncating the intermediate results to the order T after each operation. Without step 5, this is exactly what the algorithm does (in interval arithmetic) to compute $\boldsymbol{s}(\varepsilon)$. Since the interval $\boldsymbol{x}$ contains n^{-1}, we have $n\,[\varepsilon^t]\,f(n+\varepsilon) \in \boldsymbol{s}_t$ for $0 \leqslant t < T$.

Now consider the case where we include step 5. Suppose additionally that $n \notin Z$, and let $c = n^d\,\bar{q}(n^{-1}) = n^{-d}\,q(n)$. Note that $c \neq 0$. After step 5, the modified polynomial $\bar{\boldsymbol{q}}(\varepsilon)$ satisfies

$$c^{-1}\,[\varepsilon^t]\,\frac{\bar{q}(j(\varepsilon))}{j(\varepsilon)^d} \in [\varepsilon^t]\,\bar{\boldsymbol{q}}(\varepsilon), \qquad 0 \leqslant t < T.$$

Indeed, the relation for $t=0$ reduces to $1=1$, while for $t>0$, it follows from the fact that $|c| \geqslant \rho$ and hence $c^{-1} \in \boldsymbol{\gamma}$. Therefore, we have $c\,n\,[\varepsilon^t]\,f(n+\varepsilon) \in \boldsymbol{s}_t$ after step 6.

In both cases, we conclude that

$$F^{[T]}(p/q,n) = \sum_{t=0}^{T-1} n\,[\varepsilon^t]\,f(n+\varepsilon) \in \boldsymbol{\gamma}\,(|\boldsymbol{s}_0| + |\boldsymbol{s}_1| + \cdots + |\boldsymbol{s}_{T-1}|)$$

for all $n \in \mathbb{N}_{\geqslant n_0}\backslash Z$ such that $q(n) \neq 0$. If $q(n)$ vanishes for some $n \in \mathbb{N}_{\geqslant n_0}\backslash Z$, then the algorithm returns infinity (either because the constant coefficient of $\bar{\boldsymbol{q}}$ contains zero or because $\boldsymbol{\gamma}$ is unbounded), hence the bound holds for all $n \in \mathbb{N}_{\geqslant n_0}\backslash Z$. Otherwise, we can take $\rho > 0$, and the result is finite. $\qquad\square$

7.2. The general case. In its general form, Algorithm 10 requires bounds for $n \geqslant n_0$ on sequences similar to $F^{[T]}(f,n)$, but with a number of summands T that varies with n and a special handling of poles of the denominator. Both modifications to the formula can be summed up by introducing as an additional parameter the sequence of multiplicities of the poles that we wish to treat in a special way. Thus, given a rational function $f = p/q$ as before and a sequence $m(n)$ with finitely many nonzero terms, let us define

$$F(n) = \sum_{t=0}^{\tau(n)-1} \left| n\,[\varepsilon^t]\left(\frac{p(n+\varepsilon)}{\varepsilon^{-m(n)}\,q(n+\varepsilon)} \right) \right| \qquad \text{where} \qquad \tau(n) = \sum_{k=0}^{n} m(k). \tag{38}$$

The quantities that we have to bound at step 4 of Algorithm 10 are of this form[3]. The following algorithm computes the associated bounds.

Algorithm 25. [Bound rational sequence, general case]

Input. Polynomials p and q, a set Z, and an integer $n_0 \in \mathbb{N}\backslash Z$ as in Algorithm 22. A "multiplicity" function $m\colon \mathbb{N} \to \mathbb{N}$ with $m(n) = 0$ for $n \notin Z$.

Output. A bound $M \in \mathbb{R}_+ \cup \{\infty\}$ such that the function $F\colon \mathbb{N} \to \mathbb{R}_+ \cup \{\infty\}$ defined by (38) satisfies $F(n) \leqslant M$ for all $n \geqslant n_0$.

1. [Precomputation, independent of n_0.]
   Set $S(\infty) = 0$. Then, for $n \in Z$, in decreasing order:

   a. Compute $\boldsymbol{f}(\varepsilon) = p(n+\varepsilon)/(\varepsilon^{-m(n)}\,q(n+\varepsilon)) + \mathrm{O}(\varepsilon^{\tau(n)})$. Deduce a bound $b_1 \geqslant F(n)$.

   b. If $n+1 \notin Z$, compute $b_2 \geqslant \sup\{F^{[\tau(n)]}(p/q,k) : k \geqslant n+1, k \notin Z\}$ (cf. (37)) using Algorithm 22, and set $b = \max(b_1, b_2)$. Otherwise, set $b = b_1$.

   c. If b is larger than the maximum of the $S(k)$ defined so far, set $S(n) = b$.

2. Find the smallest $n \geqslant n_0$ on which S is defined. Set $M_{\mathrm{exn}} = S(n)$.

---

[3]. There is no harm in also replacing $\mu(\cdot)$ by zero in (30) when we take $\tau(n) = 1$ because the origin is an ordinary point: doing so only makes the first few terms of the bound infinite.
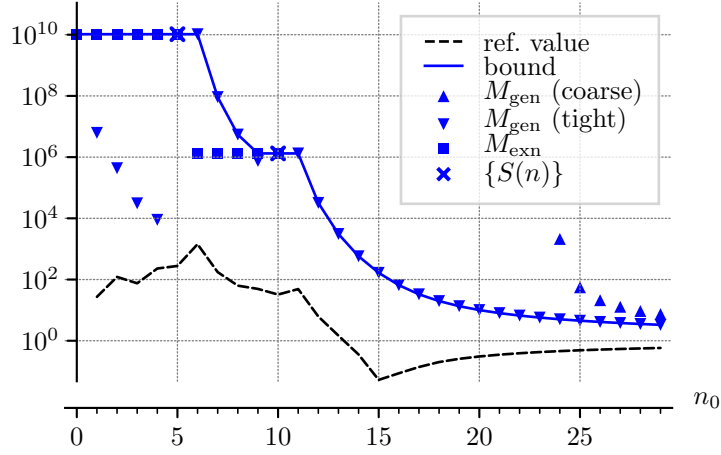
Figure 2. Bounds and intermediate values computed by Algorithm 25 called with $p = (n^2 + n + 3)(n - 15)^2$, $q = n(n - 3/2)^2(n - 5)^2(n - 10)$, $Z = \{0, 5, 10\}$, and $m(0) = 1$, $m(5) = 2$, $m(10) = 1$. The main bound (solid curve) and the intermediate values marked with ▼ include the optional step of Algorithm 22. The values of M_{gen} without the optional step, shown for comparison when finite, are marked with ▲.

3. If $n_0 \in Z$, then return M_{exn}.

4. Compute $M_{\mathrm{gen}} \geqslant \sup \{F^{[\tau(n_0)]}(p/q, n) : n \geqslant n_0, n \notin Z\}$ using Algorithm 22.

5. Return $\max(M_{\mathrm{exn}}, M_{\mathrm{gen}})$.

Remark 26. Algorithms 22 and 25 immediately extend to vectors $(p_1/q, ..., p_\ell/q)$ of rational functions with the same denominator. A large part of the computation can be shared between the entries.

Proposition 27. *Given parameters that define a sequence $F(n)$ of the form (38) and an integer n_0, Algorithm 25 returns a bound M such that $\sup_{n \geqslant n_0} F(n) \leqslant M$. If, for every integer $n \geqslant n_0$, the multiplicity of n as a zero of q is at most $m(n)$ (and the working precision is large enough), then M is finite.*

Proof. First observe that, due to step 1c, the values $S(n)$ defined at step 1 are decreasing (in the sense that $S(m) < S(n)$ when both are defined and $m > n$). We extend S to a staircase function defined on $\mathbb{N} \cup \{\infty\}$ by setting the undefined $S(n)$ to the smallest values that make S non-increasing (see Figure 2).

Let us prove by induction that

$$n \in Z \quad \Rightarrow \quad (\forall k \geqslant n, \quad F(k) \leqslant S(n)). \tag{39}$$

This is true for $n > \max Z$. Then fix $n \in Z$ and assume (39) holds for larger indices. Step 1a of the algorithm ensures that $S(n) \geqslant F(n)$. For $n < k < \min(\mathbb{N}_{>n} \cap Z)$, we have $F(k) = F^{[\tau(n)]}(p/q, k)$, hence $S(n) \geqslant F(k)$ by the correction of Algorithm 22 (Proposition 24). Finally, by the induction hypothesis, we also have $S(n) \geqslant F(k)$ for larger k.

The effect of steps 2 to 5 is to construct a quantity

$$M \geqslant \begin{cases} \max\left(S(n_0), \displaystyle\sup_{n \geqslant n_0, n \notin Z} F^{[\tau(n_0)]}(p/q, n)\right), & n_0 \notin Z, \\ S(n_0), & n_0 \in Z. \end{cases}$$

When $n_0 \in Z$, the inequality $\sup_{n \geqslant n_0} F(n) \leqslant M$ holds by (39). When $n_0 \notin Z$, as with steps 1b–1c, we have $M \geqslant S(n_0) \geqslant S(n_0') \geqslant S(n)$ for $n \geqslant n_0'$, where $n_0' = \min \mathbb{N}_{\geqslant n_0} \cap Z$, and $M \geqslant F^{[\tau(n_0)]}(p/q, n)$ for $n_0 \leqslant n < n_0'$. It follows that $M \geqslant F(n)$ for all $n \geqslant n_0$. $\qquad\square$

8. Numerical bounds

The algorithms developed at this stage bound the tails of differentially finite series by hyperexponential majorant series. We have seen that a series $\hat{u}(z)$ with $u(z) \ll \hat{u}(z)$ encodes bounds $|u^{(j)}(\zeta)| \leqslant \hat{u}^{(j)}(|\zeta|)$ on the values of u and all its derivatives. Yet, in order to use the majorants in a concrete setting, we still have to explain how to effectively evaluate the $\hat{u}^{(j)}(|\zeta|)$. Let us first consider the details of this operation, and then outline two other ways—appropriate for different settings—of deriving numerical tail bounds from the majorant series.

8.1. Values. When j is large, it is essential for performance to compute the values of the derivatives by working with truncated power series (or another similar compact representation) rather than by naive symbolic differentiation. The process is laid out in Algorithm 28 below, which takes as input a hyperexponential majorant represented in the form discussed after Algorithm 19. Despite the heavy notation, the algorithm is mostly straightforward, the only subtlety being that we bound sub-expressions of the form $\int (f / g)$ by $(\int f) / g$ to avoid computing integrals of rational functions. As in the previous section, boldface letters stand for intervals or polynomials with interval coefficients, and "compute $\varphi(\varepsilon) + \mathrm{O}(\varepsilon^m)$" means "compute a Taylor expansion with interval coefficients of $\varphi(\varepsilon)$, truncated to the order m, using power series operations".

Algorithm 28. [Numerical remainder bounds]

Input. Parameters $\delta \in \mathbb{N}$, $\hat{p}, \check{q}, \hat{f}_0, \check{g}_0, ..., \hat{f}_{n-1}, \check{g}_{n-1} \in \mathbb{R}[z]$ defining a series

$$\hat{u}(z) = z^\delta \frac{\hat{p}(z)}{\check{q}(z)} \exp \int_0^z \left(\sum_{i=0}^{n-1} \frac{\hat{f}_i(w)}{\check{g}_i(w)} \mathrm{d}w \right) \in \mathbb{R}_+[[z]].$$

An evaluation point $x \geqslant 0$. A differentiation order m.

Output. Non-negative reals $M_0, ..., M_{m-1}$.

1. Let $\boldsymbol{z}(\varepsilon) = x + \varepsilon$, where ε is an indeterminate. Initialize $\boldsymbol{J}(\varepsilon)$ to 0.

2. For $i = 0, ..., n - 1$:

    a. Compute the antiderivative $\boldsymbol{F}_i(z) = \int_0^z \hat{f}_i(w) \, \mathrm{d}w$.

    b. Compute $\boldsymbol{F}_i(\boldsymbol{z}(\varepsilon)) / \check{g}_i(\boldsymbol{z}(\varepsilon)) + \mathrm{O}(\varepsilon^m)$ and add it to $\boldsymbol{J}(\varepsilon)$.

3. Compute $\boldsymbol{S}(\varepsilon) = \boldsymbol{z}(\varepsilon)^\delta \hat{p}(\boldsymbol{z}(\varepsilon)) \exp(\boldsymbol{J}(\varepsilon)) / \check{q}(\boldsymbol{z}(\varepsilon)) + \mathrm{O}(\varepsilon^m)$.

4. Return the right bounds of the intervals $j! \, [\varepsilon^j] \, \boldsymbol{S}(\varepsilon)$, $0 \leqslant j < m$.

Proposition 29. *In Algorithm 28, suppose that for all i, the series $\hat{u}_i(z)$ and $\check{v}_i(z)^{-1}$ have non-negative coefficients. Then the values M_j returned by the algorithms are bounds for the derivatives of $\hat{u}(z)$ at x, with $0 \leqslant \hat{u}^{(j)}(x) \leqslant M_j$, $0 \leqslant j < m$.*

Proof. Steps 1 to 3 compute the Taylor expansion to the order m at $z = x$ of

$$z^\delta \frac{\hat{p}(z)}{\check{q}(z)} \exp \left(\sum_{i=0}^{n-1} \frac{1}{\check{g}_i(z)} \int_0^z \hat{f}_i(w) \, \mathrm{d}w \right).$$

An integration by parts shows that

$$\int_0^z \frac{\hat{f}_i(w)}{\check{g}_i(w)} \, \mathrm{d}w = \frac{1}{\check{g}(z)} \int_0^z \hat{f}_i(w) \, \mathrm{d}w - \int_0^z \left(\frac{1}{\check{g}_i} \right)'(w_1) \int_0^{w_1} \hat{f}_i(w_2) \, \mathrm{d}w_2 \, \mathrm{d}w_1,$$

where the left-hand side and both terms on the right-hand side are series with non-negative coefficients, so that we have

$$\int_0^z \frac{\hat{f}_i(w)}{\check{g}_i(w)} \, \mathrm{d}w \ll \frac{1}{\check{g}(z)} \int_0^z \hat{f}_i(w) \, \mathrm{d}w.$$

It follows that the coefficients of $\boldsymbol{S}(\varepsilon)$ are enclosures of upper bounds on the corresponding Taylor coefficients of $\hat{u}(|\zeta| + \varepsilon)$. $\qquad\qquad\square$

When $u(z)$ is a plain power series and $u_{n:}(z) \ll \hat{u}(z)$, the above algorithm yields bounds on $|(u_{n:})^{(j)}(\zeta)|$, that is, on values of the successive derivatives of the tail. This is what one typically needs in applications, yet, since $(\hat{u}_{n:})'(z) = N\,\hat{u}_n\,z^{n-1} + (\hat{u}')_{n:}(z)$, the same quantities also bound the tails of the derivatives.

If $u(z) = z^{\lambda} \sum_k f_k(z)\,\log(z)^k / k!$ is a logarithmic series, though, we only get bounds on derivatives of tails of the components $f_k(z)$. To deduce bounds that apply to tails of the generalized series expansion of $u(z)$ itself, it is usually best to form the expansion of $f_k(|\zeta| + \varepsilon)$ with respect to ε and compute $(|\zeta| + \varepsilon)^{\lambda} \sum_k f_k(|\zeta| + \varepsilon)\,\log(|\zeta| + \varepsilon)^k / k!$ in power series arithmetic, similar to what Algorithm 28 does.

8.2. "A priori" bounds. A small drawback of the technique discussed above is that it requires knowing the last few coefficients $u_{n-1}, ..., u_{n-s}$ just before the truncation point in order to bound the tail $u_{n:}(\zeta)$ of a series $u(z)$. Thus, the results are in a sense *a posteriori* bounds, that cannot be used to decide with certainty where to truncate the series before even starting the computation of the coefficients. (It is of course possible to "guess" a plausible truncation order based on asymptotic considerations, or to compute the required coefficients at low precision before running the full-precision computation.)

If however, given a majorant $\hat{u}(z)$ of $u_{n_0:}$ obtained by the previous methods for some n_0, we can get sufficiently tight bounds on the values of the remainders $\hat{u}_{n_1:}(|\zeta|)$ for $n_1 > n_0$, then we have a way of bounding the higher-order remainders $u_{n_1:}(\zeta)$ without computing all the coefficients up to n_1. We now adapt to the setting of this paper a technique of Mezzarobba and Salvy [30] for doing so, based on the classical saddle point method in asymptotics. The idea is that for suitably chosen ρ, the bound (40) below is relatively tight.

Lemma 30. *Let $\hat{v}(z)$ be a power series with non-negative coefficients. Fix real numbers $\rho \geqslant x > 0$ within the disk of convergence of $\hat{v}$. For all $n \in \mathbb{N}$, the series expansion at x of the remainder of order n of $\hat{v}(z)$ is bounded as*

$$\hat{v}_{n:}(x + \varepsilon) \ll \left(\frac{x}{\rho}\right)^n \hat{v}\big(\rho\,(1 + x^{-1}\varepsilon)\big). \qquad\qquad (40)$$

In particular, we have $\hat{v}_{n:}(x) \leqslant (x/\rho)^n\,\hat{v}(\rho)$.

Proof. Write

$$\hat{v}_{n:}(x\,z) = \sum_{k=0}^{\infty} \hat{v}_{n+k}\,x^{n+k}\,z^{n+k} = \left(\frac{x}{\rho}\right)^n \sum_{k=0}^{\infty} \hat{v}_{n+k}\,x^k\,\rho^n\,z^{n+k}.$$

Since $\rho \geqslant x$, we have

$$\hat{v}_{n:}(x\,z) \ll \left(\frac{x}{\rho}\right)^n \hat{v}_{n:}(\rho\,z) \ll \left(\frac{x}{\rho}\right)^n \hat{v}(\rho\,z).$$

The result follows by substituting $1 + x^{-1}\varepsilon$ for z. $\qquad\qquad\square$

For simplicity, let us focus on series $\hat{u}(z)$ of the shape returned by Algorithm 19, viz.,

$$\hat{u}(z) = z^{n_0}\,\hat{v}(z) = z^{n_0}\,\hat{b}(z)\,\exp\int_0^z w^{-1}\,\hat{a}(w)\,\mathrm{d}w, \qquad \hat{a}, \hat{b} \in \mathbb{C}(z), \qquad\qquad (41)$$

and assume that $\hat{a}(z)$ and $\hat{b}(z)$ are not both constant[4]. Additionally, we limit ourselves in the analysis to the bound $(x/\rho)^n\,\hat{v}(\rho)$ on the value of the remainder, leaving to the reader the case of derivatives.

---

4. There is no loss in generality in doing so. Besides, the only case that would lead to majorant series breaking the assumption is that of differential equations of the form $P(\theta) \cdot u = 0$, whose solutions are linear combinations of a finite number of generalized monomials.

Let ρ^* be the pole of $\hat{a}(z)\,\hat{b}(z)$ closest to the origin, with $\rho^* = \infty$ if both $\hat{a}(z)$ and $\hat{b}(z)$ are polynomials. As $\hat{a}(z)$ and $\hat{b}(z)$ have non-negative coefficients, ρ^* is real (or infinite) and positive. Besides, in our setting, the denominatorsSe of $\hat{a}(z)$ and $\hat{b}(z)$ divide the polynomial $\check{p}(z)$ computed by Algorithm 10, hence ρ^* can be taken arbitrarily close to the modulus of the singularity of the differential equation closest to the origin. The next lemma shows that, in (40), it is possible to choose $\rho = \rho_n$ close to ρ^*, so that $(x / \rho_n)^n$ decreases fast, without letting $\hat{v}(\rho_n)$ grow too large. In particular, the resulting sequence of bounds tends to zero at least exponentially fast, with the same exponential rate x / ρ^* as the coefficients of $\hat{v}_n$ in the case $\rho^* < \infty$.

Lemma 31. *Fix $x < \rho^*$ and $c > 0$.*

- *If ρ^* is finite, let $m_0 \in \mathbb{N}$ be its multiplicity as a pole of $\hat{a}(z)$ and $\hat{b}(z)$. Let $m = \max(1, m_0)$. For $n > c^m$, define $\rho_n = \rho^*(1 - c\,n^{-1/m})$.*

- *If $\rho^* = \infty$, let $d = \deg \hat{a}(z)$ and $\rho_n = c\,n^{1/d}$.*

Then, as $n \to \infty$, the following asymptotic bounds hold:

- *$(x / \rho_n)^n\,\hat{v}(\rho_n) = (x / \rho^*)^n\,n^{O(1)}$ if $\rho^* < \infty$ and $m = 1$,*

- *$(x / \rho_n)^n\,\hat{v}(\rho_n) = (x / \rho^*)^n \exp\big(O(n^{1-1/m})\big)$ if $\rho^* < \infty$ and $m \geqslant 2$,*

- *$(x / \rho_n)^n\,\hat{v}(\rho_n) = n^{-n/d}\,e^{O(n)}$ if $\rho^* = \infty$.*

Proof. Assume first that ρ^* is finite. When ρ tends to ρ^*, we have

$$\int_0^\rho w^{-1}\,\hat{a}(w)\,\mathrm{d}w = \begin{cases} O\big((\rho^* - \rho)^{-m_0+1}\big), & m_0 \geqslant 2, \\ O(-\log(\rho^* - \rho)), & m_0 = 1, \\ O(1), & m_0 = 0, \end{cases}$$

and thus, since $\rho^* - \rho_n = c\,\rho^*\,n^{-1/m}$,

$$\exp\int_0^{\rho_n} w^{-1}\,\hat{a}(w)\,\mathrm{d}w = \begin{cases} \exp\big(O(n^{1-1/m_0})\big), & m_0 \geqslant 2, \\ n^{O(1)}, & m_0 = 1, \\ O(1), & m_0 = 0, \end{cases}$$

as $n \to \infty$. The analogous estimates when $\rho^* = \infty$ read

$$\int_0^\rho w^{-1}\,\hat{a}(w)\,\mathrm{d}w = O(\rho^d) \quad (\rho \to \infty), \qquad \exp\int_0^{\rho_n} w^{-1}\,\hat{a}(w)\,\mathrm{d}w = \exp O(n) \quad (n \to \infty).$$

In both cases, the other factor of $\hat{v}(\rho_n)$ satisfies $\hat{b}(\rho_n) = n^{O(1)}$. Finally, the prefactor $(x / \rho_n)^n$ is bounded as

$$(x / \rho_n)^n = \begin{cases} (x / \rho^*)^n\,(1 - c\,n^{-1/m})^{-n} = \exp\big(O(n^{1-1/m})\big), & \rho^* < \infty, \\ O(x^n\,n^{-n/d}), & \rho^* = \infty. \end{cases}$$

The result follows by combining these estimates. $\qquad\square$

While, asymptotically, these formulas yield tight bounds for any fixed c, the actual values of $(x / \rho_n)^n\,\hat{v}(\rho_n)$ for finite n are quite sensitive to that of ρ_n, and the bounds implied by the previous lemma are typically very poor for moderate n. We could, of course, derive more precise asymptotic estimates of the optimal choice of ρ_n as a function of n, but in practice it is best to minimize $F_n(\rho) = (x / \rho)^n\,\hat{v}(\rho)$ numerically—not necessarily in a rigorous way. Note that, for large n, the function F_n a unique local minimum[5] on $[0, \rho^*)$, so that simple numerical methods work well for this purpose.

---

5. Indeed, its logarithmic derivative reads $F_n'(\rho) / F_n(\rho) = \rho^{-1}(G(\rho) - n)$ where $G(\rho) = \rho\,\hat{b}'(\rho) / \hat{b}(\rho) + \hat{a}(\rho)$. Since the numerator of $\hat{b}(z)$ has non-negative coefficients, $G(\rho)$ is a rational function without poles on $[0, \rho^*)$, and hence a continuous function with finitely many local extrema. Additionally, it satisfies $G(0) = 0$ and $G(\rho) \to +\infty$ as $\rho \to \rho^*$. Therefore, for large n, the equation $G(\rho) = n$ has exactly one solution.
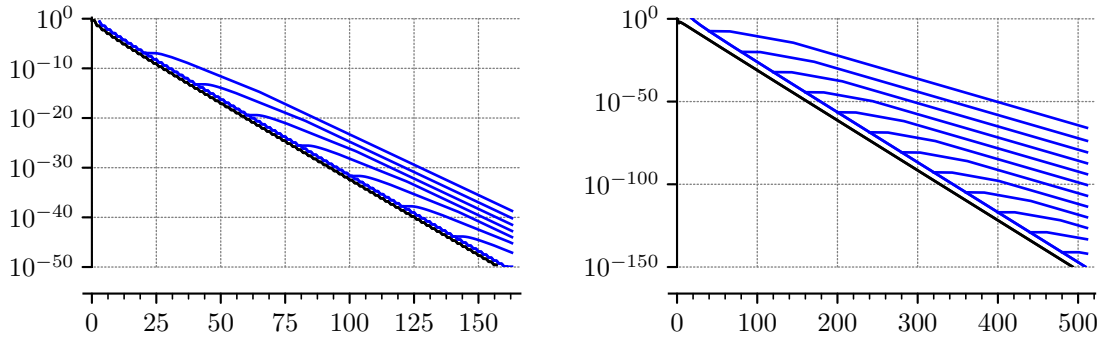
Figure 3. "A priori" vs. "a posteriori" bounds on the remainders $u_{n:}(\zeta)$ of a series $u(z)$, as a function of the truncation order n. Left plot: $u(z) = \arctan(z)$ at $\zeta = 1/2$. Right plot: $u(z)$ and ζ as in Example 1 (p. 2).
  On each plot, the bottommost curve (in black) is the actual truncation error. The next curve from bottom to top is the "a posteriori" bound on $u_{n:}(\zeta)$ given by Algorithm 10 with $\ell = 10$ and Algorithm 19, based on the coefficients $u_0, ..., u_{n-1}$ of $u(z)$. It connects the starting points of the remaining curves, which represent, for several values of n_0, the bounds on $u_{n:}(\zeta)$ for $n \geqslant n_0$ computed from the same majorant series using only the coefficients $u_0, ..., u_{n_0-1}$, as described in Section 8.2.
  See Section 9 for more information on the implementation.

To sum up, we can use the following algorithm to compute bounds on $\hat{u}_{n:}(x)$ and its first m derivatives given a series $\hat{u}(z)$ of the form (41): first call Algorithm 28 on $\hat{u}(z)$ and x. If $n \leqslant n_0$, return its result. Otherwise, define ρ^* as above, and search for a value $\rho \in [x, \rho^*)$ that minimizes $\log((x/\rho)^n f(\rho))$, where $f(\rho)$ is the value computed by Algorithm 28 called with $m = 1$. Then compute $j! [\varepsilon^j] S(\varepsilon)$, $0 \leqslant j < m$, where $S(\varepsilon) = (x + \varepsilon)^{n_0} (x/\rho)^n \hat{v}(\rho(1 + x^{-1}\varepsilon)) + \mathrm{O}(\varepsilon^m)$. Compare with the bounds computed at the first step, and return the better one. (This last step is not strictly necessary, since (40) reduces to the trivial bound $\hat{v}_{n:}(x + \varepsilon) \ll \hat{v}(x + \varepsilon)$ when $\rho = x$, but it can help if the numerical computation of ρ is inaccurate.)

Proposition 32. *If $u(z) \ll \hat{u}(z)$ and $n \geqslant n_0$, the algorithm outlined above yields valid bounds on $u_{n:}(\zeta), ..., (u_{n:})^{(m-1)}(\zeta)$ for all ζ such that $|\zeta| \leqslant x$. Provided that the numerical method used to choose ρ is accurate enough, the bound on $u_{n:}(\zeta)$ tends to zero at least exponentially fast as n grows.*

Figure 3 shows the results of a simple implementation of this strategy. While the sub-exponential overhead promised by Lemma 31 can be observed in practice, at least for the simpler of the equations, we also see that these "a priori" bounds are much more pessimistic than the "a posteriori" ones in slightly more complicated cases.

8.3. Remainders of Laplace transforms. Another downside of our framework is that hyperexponential majorant series are not expressive enough to capture the remainder asymptotics of solutions of arbitrary differential equations with polynomial coefficients. For example, the Airy function satisfies $\mathrm{Ai}(z) = \exp(\sigma_\varphi z^{3/2} + \mathrm{O}(\log z))$ as $z \to \infty$ in a generic direction φ, and $[z^n]\,\mathrm{Ai}(z) = \mathrm{O}(n!^{-2/3})$, but the best majorants our method can express are of the form $\hat{u}(z) = \hat{g}(z) \exp(\sigma z^3)$, $\hat{g}(z) \in \mathbb{Q}[z]$, with u_n of the order of $n!^{-1/3}$. More generally, for any rational $\kappa \geqslant 0$, it is not hard to find an equation whose solutions are entire functions $u(z)$ of order κ or higher, and hence have tails $u_{n:}(\zeta)$ bounded by $n!^{-1/\kappa} e^{\mathrm{O}(n)}$ (ζ fixed). In contrast, hyperexponential entire functions are necessarily of integer order, with tails that decrease like $n!^{1/d} e^{\mathrm{O}(n)}$ where $d \in \mathbb{N}_{>0}$. Besides, even when the fastest-growing solution of the differential equation we are considering is of integer order, there is no guarantee that the hyperexponential majorant will respect that order.

This overestimation is not too much of an issue for the "a posteriori" error bounds discussed in the main part of this paper because, even if the choice of $\hat{h}(z)$ in Section 6.4 is not as tight as it could, the factor derived from the residual has the "correct" asymptotics with respect to n. For the "a priori" bounds of the previous subsection this argument does not apply, and they can be far from matching the actual speed of convergence of the solutions, even in non-degenerate cases.

A possible remedy (essentially a streamlined version of a similar idea proposed by Mezzarobba and Salvy [30], to which we refer for more information) is to re-scale the solutions of the original equation by a generalized Laplace transform

$$\sum_{n=0}^{\infty} u_n z^n \quad \mapsto \quad \sum_{n=0}^{\infty} \phi(n) u_n z^n, \qquad \phi(n) = q^{\frac{p}{q} n} \Gamma\left(1 + \frac{n}{q}\right)^p$$

This transformation can be performed algorithmically at the level of differential operators, and it is possible to choose p, q in such a way that the resulting operator has at least one nonzero finite singular point, while remaining regular at the origin. One can then compute majorant series of the solutions of the transformed equation, and obtain bounds on the remainders of the original solutions by the following variant of Lemma 30. (We have not experimented with this technique.)

Lemma 33. *Let* $\hat{u}(z) = \sum_{n\geqslant 0} \hat{u}_n z^n \in \mathbb{R}_+[[z]]$, *and let* $\hat{v}(z) = \sum_{n\geqslant 0} \phi(n) \hat{u}_n z^n$ *for integers* $p \in \mathbb{Z}$, $q \in \mathbb{N}_{>0}$ *chosen so that* $\hat{v}(z)$ *has a nonzero radius of convergence. Let* $\rho > 0$ *lie within the disk of convergence of* $\hat{v}(z)$. *Then, for all* x *and* n *such that* $0 \leqslant x \leqslant \rho(n+1)^{p/q}$, *we have the bound*

$$\hat{u}_{n:}(x) \leqslant \frac{1}{\phi(n)} \left(\frac{x}{\rho}\right)^n \hat{v}(\rho).$$

Proof. Write

$$\hat{u}_{n:}(x) = \sum_{k=0}^{\infty} \frac{\hat{v}_{n+k}}{\phi(n+k)} x^{n+k} = \frac{1}{\phi(n)} \left(\frac{x}{\rho}\right)^n \sum_{k=0}^{\infty} \frac{\phi(n)}{\phi(n+k)} \rho^n x^k.$$

Since the function $t \mapsto \Gamma(t + a) / \Gamma(t)$ is non-decreasing for fixed a, and using Gautschi's inequality [10, Sec. 5.6.4], we have

$$\frac{\Gamma(1 + n/q)}{\Gamma(1 + (n+k)/q)} \leqslant \left(\frac{\Gamma(1 + n/q)}{\Gamma(1 + (n+1)/q)}\right)^k \leqslant \left(\frac{q}{n+1}\right)^{k/q}$$

and hence

$$\frac{\phi(n)}{\phi(n+k)} = q^{-\frac{p}{q}k} \left(\frac{\Gamma(1 + n/q)}{\Gamma(1 + (n+k)/q)}\right)^p \leqslant (n+1)^{-(p/q)k} \leqslant \left(\frac{\rho}{x}\right)^k.$$

The result follows. □

In practice, when applying this lemma, x is given, n must be large enough that $x \leqslant \rho^* (n+1)^{p/q}$ hold, and ρ is to be chosen as a function of n as in the previous subsection. For large x, Lemma 33 starts being applicable when $n \approx x^{p/q}$, which typically matches the position where the terms of the series $u(z)$ "start converging" (cf. Section 9.2).

9. Implementation and examples

9.1. Implementation. We have implemented the algorithms of this paper in `ore_algebra` [24, 29], a library for working with Ore polynomials in the SageMath (Sage) computer algebra system. Among the features of Sage and the many external libraries on which it is based, our code relies in an essential way on Arb [23], which provides all necessary basic operations on intervals and truncated power series with interval coefficients. The implementation also makes use of PARI for complex root finding, and of `ore_algebra` itself for basic arithmetic with differential and recurrence operators. The operations on rational numbers and exact polynomials mainly come from GMP/MPIR, Flint and Singular.

The `ore_algebra` package is available from https://github.com/mkauers/ore_algebra/ under the GNU General Public License (version 2 or later). The code for computing error bounds can be found in the file `src/ore_algebra/analytic/bounds.py` of the source tree. We performed the experiments described below using the git revision `dd88e8d5` under Sage 8.2.

Except when otherwise noted, we run the full version of the algorithm described in this paper, that is, the combination of Algorithms 10, 19, 25, and 28, including their optional steps, and using PARI's root finder at step 2 of Algorithm 10. Numerical coefficients are represented by real or complex intervals almost everywhere, with a fixed precision of 53 bits.

In examples involving generalized series expansions at singularities, the displayed truncation error and error bound correspond to the complete logarithmic series, including the non-analytic factors.
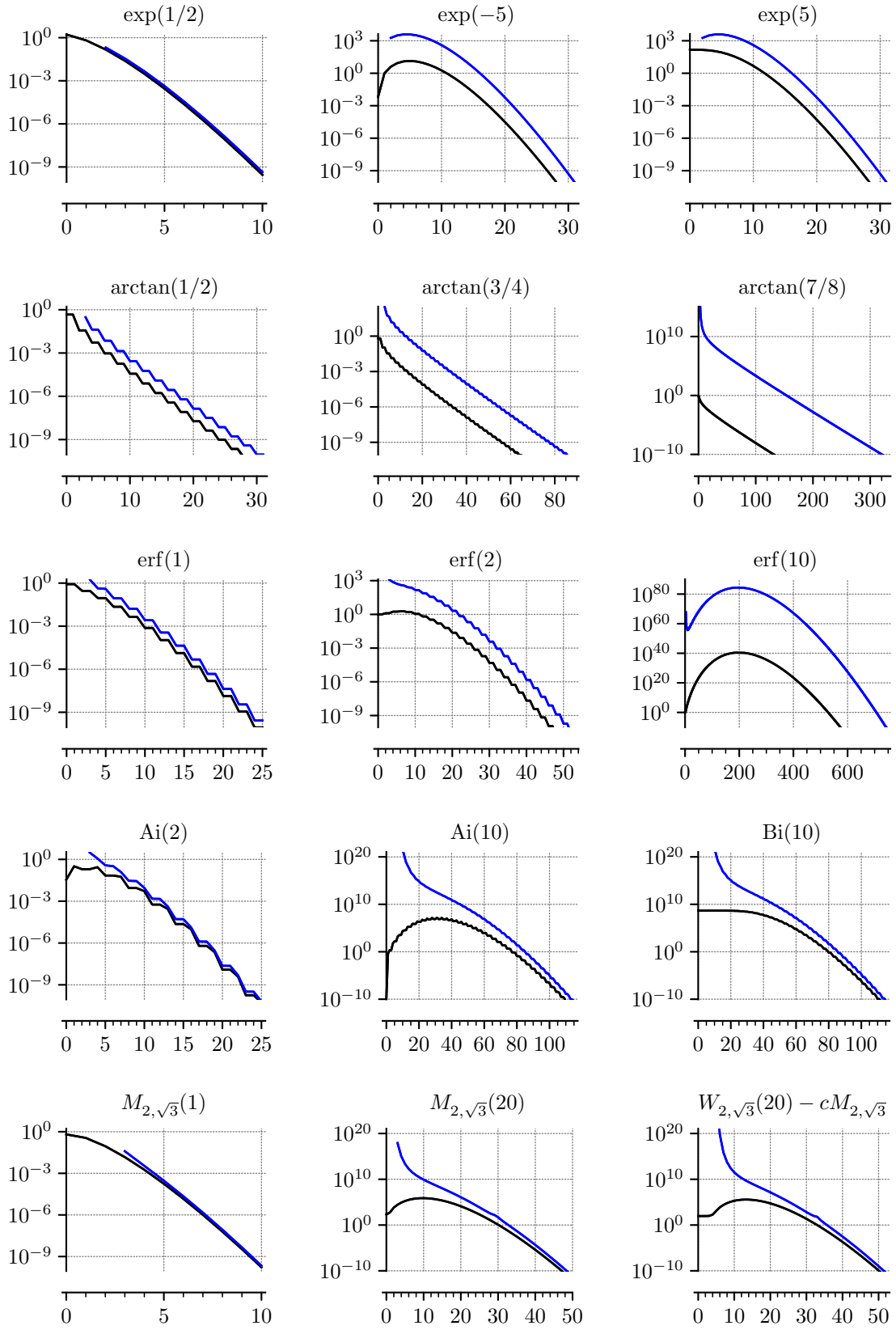
Figure 4. Truncation errors of series expansions at the origin of classical functions (see Section 9.2). The two curves on each plot are the actual truncation error and the bound, as a function of the truncation order. All bounds were computed using Algorithm 10 with $\ell = 3$. In the last row, $M_{\kappa,\mu}$ and $W_{\kappa,\mu}$ denote the Whittaker functions, and $c = \pi / \left(\sin(\alpha\pi)\,\Gamma(\alpha)\,\Gamma(-\sqrt{3} - 3/2) \right)$ where $\alpha = 1 + 2\sqrt{3}$.

Remark 34. Testing the implementation for correctness is a significant issue, as the final truncation bounds are likely to be valid even if the code is wrong, due to overestimation. Without a complete formal proof of the implementation, it is very hard to be certain that it is fully correct. Nevertheless, the fact that our bounds are close to optimal in simple cases is of great help in catching bugs. We further limit the risks of missing issues hidden by overestimations by testing not only the final bounds, but also various intermediate results. In our experience, plausible but incorrect changes to the bound computation algorithm tend to be caught by the test suite.

9.2. Elementary and special functions. Figure 4 illustrates the behavior of the algorithm in a variety of "easy" model cases. We use very simple differential equations satisfied by classical elementary and special functions and focus on the evaluation of their solutions at low precision.

Each row shows three examples of solutions of the same equation, from top to bottom

$$u'(z) - u(z) = 0, \qquad (z^2 + 1)\, u''(z) + 2\, z\, u'(z) = 0, \qquad u''(z) + 2\, z\, u'(z) = 0,$$

$$u''(z) - z\, u(z) = 0, \qquad 4\, z^2\, u''(z) - (z^2 - 8\, z + 11)\, u(z) = 0.$$

For each of these equations, we consider the series expansion at the origin of one or several solutions (respectively the exponential, the arctangent, the error function, the Airy functions Ai and Bi, and certain linear combination of Whittaker functions of parameters $\kappa = 2$, $\mu = \sqrt{3}$), evaluated at different points ζ of their disk of convergence. The first four equations are ordinary at the origin. The last one is regular singular, with irrational exponents $1/2 \pm \sqrt{3}$. The local expansion of $M_{\kappa,\mu}$ lies in $z^{1/2-\sqrt{3}}\,\mathbb{C}[[z]]$, and the constant c is chosen so that $W_{\kappa,\mu} - c\, M_{\kappa,\mu} \in z^{1/2+\sqrt{3}} \in \mathbb{C}[[z]]$.

We plot the truncation error $|u_{n:}(\zeta)|$ and the corresponding bound on a logarithmic scale, as a function of the number of terms n, until they become smaller than 10^{-10}. The left column corresponds to evaluations at points ζ where the series $u(\zeta)$ converges nicely—halfway from the circle of convergence in the case of $\arctan(z)$, and at points where the magnitude of the terms of $u(\zeta)$ start decreasing early in the case of entire functions. For such simple equations, the experiments confirm that the bounds are very tight, in spite of the very low accuracy target and the moderate value of ℓ.

Though a bit larger, the overestimation remains moderate in the two other columns. We will come back to the case of $\arctan(7/8)$ later. We can observe that when the terms exhibit a "hump" before the series really starts converging, our bounds tend to overestimate the true error by a factor comparable to the height of the hump. A heuristic explanation is that, in the majorant series $\hat{u}(z) = \check{p}(z)^{-1}\, \hat{g}(z)\, \hat{h}(z)$ that we compute for $u_{n:}(z)$, the coefficients of $\hat{g}(z)$ are roughly of the order of $|u_n|$, while $\hat{h}(z)$ is not too far from $\sum_{j \geqslant 0} |u_j|\, z^j$. Thus, the value of the majorant series at $x = |\zeta| > 0$ is about $\sum_{j \geqslant 0} |u_j|\, x^j \approx \max_{j \geqslant 0} (|u_j|\, x^j)$ times larger than $|u_{n:}(\zeta)| \approx |u_n|\, x^n$. This overestimation has a limited impact in applications, because one typically tries to avoid humps in the coefficients in the first place—for example, in a Taylor method, by adjusting the step size. Nevertheless, it would be interesting to find a way of avoiding it.

9.3. "Real-world" examples. Next, we consider a family of larger examples borrowed from an application. Other families of "interesting" operators, as well as tools to produce plots similar to the ones in this paper, are available in the `ore_algebra` source tree.

Figure 5 shows remainder bounds for solutions at the origin of the differential equations for lattice Green functions of face-centered hypercubic lattices discovered (heuristically or rigorously) by several authors over the last decade [16, 4, 26, 48, 17]. In particular, the case $d = 4$ corresponds to the operator of Example 1, but without the shift from the neighborhood of 0 to that of $1/2$. The other operators can be found on Ch. Koutschan's web page[6]; here we limit ourselves to collecting some statistics about their size:

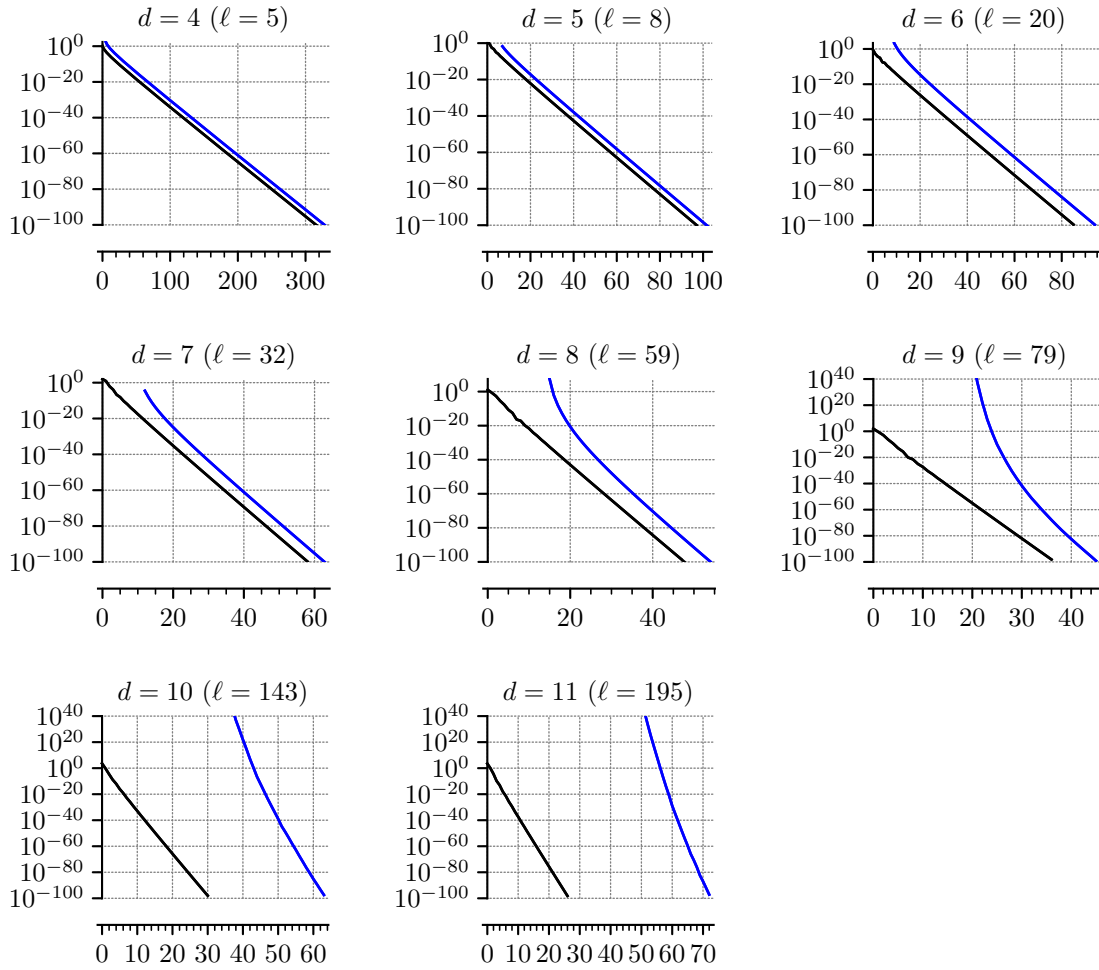| dimension d	4	5	6	7	8	9	10	11
order r	4	5	8	11	14	18	22	27
max. coefficient (w.r.t. D_z) degree	10	17	43	68	126	169	300	409
max. polynomial coefficient size (bits)	18	50	143	273	654	959	1907	2888

Figure 5. Truncation errors and remainder bounds computed by Algorithm 10, run with the indicated ℓ, for random solutions of annihilating operators of lattice Green functions of d-dimensional face-centered cubic lattices (see Section 9.3).

(Thus, the textual representation of the operator for $d = 11$ takes about 8 megabytes.) The value at $z = 1$ of the lattice Green function gives access to the return probability of the random walk on the lattice, and, to the best of our knowledge, going through the differential equation is the only known way of computing that probability to high precision in good complexity.

All these operators have regular singularities with integer exponents at the origin. For example, the monic indicial polynomial for $d = 9$ is $n^9 (n - 1)^5 (n - 2)^3 (n - 3)$. In each case, we select small initial values at random and evaluate the logarithmic series solution characterized by these initial values about halfway from the boundary of its circle of convergence. We run Algorithm 10 with $\ell = \lfloor s/2 \rfloor + 2$, in accordance with the heuristic suggested on page 14.

It is especially noticeable here that the left endpoint of the upper curve is located significantly to the right of that of the other curve, even when no part of the curve is clipped out of the plot. This is because our implementation can return infinite bounds, and does so at least until it has passed all initial values.

The fast convergence of the power series as d increases can be explained by the presence of *apparent singularities* close to the origin, which leads us to select evaluation points much smaller than the actual radius of convergence of the *solutions*. Thus, roughly speaking, a rigorous order-adaptive Taylor method using our algorithm would perform many more steps to evaluate the lattice Green function at $z = 1$ for $d = 9$ than for $d = 4$, but sum fewer terms at each step. Although our bounds tend to infinity when the magnitude of the evaluation point approaches that of the singular point of the *equation* closest to the origin—apparent or not—, we can see that in these examples,

they adapt nicely to the unexpectedly fast convergence, thanks to the use of residuals. Having to perform many small steps is nevertheless sub-optimal, which raises the question of adapting the algorithm to handle apparent singularities in a way that allows for larger step sizes.

We can also observe that overestimation becomes significant for the larger of these operators. Further increasing the tuning parameter ℓ does not seem to help. Other methods may hence be needed for the really efficient numerical solution of equations with coefficients of very large degree at comparatively low precision.

It would be hard to draw any conclusion from a detailed analysis of the running time of our Python implementation. Still, anecdotal data confirms that the bounds of this paper are usable in practice as part of symbolic-numeric algorithms. For example, the simple variable-order interval Taylor method implemented in `ore_algebra` can compute an enclosure with radius less than 10^{-60} of the value at $z=1$ of the lattice Green function for $d=4$ in less than a second on a modest laptop. The computation of the remainder bounds accounts for an estimated 5 to 10% of the total running time. Of course, the solver does not compute a complete tail bound at every iteration, but starts with a heuristic convergence check, and reuses various intermediate results when the rigorous check does not succeed on the first try, as discussed in particular in Remark 13.

9.4. Effort-tightness trade-off. Figure 6 shows the influence of the tuning parameter ℓ of Algorithm 10 and its importance for obtaining good bounds in relatively hard cases. We consider the following examples:

(a) The evaluation of $\arctan(z)$ close to its circle of convergence for which $\ell = 3$ did not give satisfactory results on Figure 4.

(b) The operator of the previous subsection for $d=6$, but taken in the neighborhood of $z=1/2$ as in Example 1, with initial values $u(0) = 1$, $u'(0) = \cdots = u^{(7)}(0)=0$. The closest singular point is at distance about 0.256, and we evaluate the expansion at $z=9/70 \approx 0.129$.

(c) The operator below, handpicked among examples generated at random (by first picking elements of $\mathbb{Z}[z]\langle D_z \rangle$ of balanced orders and degrees with small integer coefficients, and then
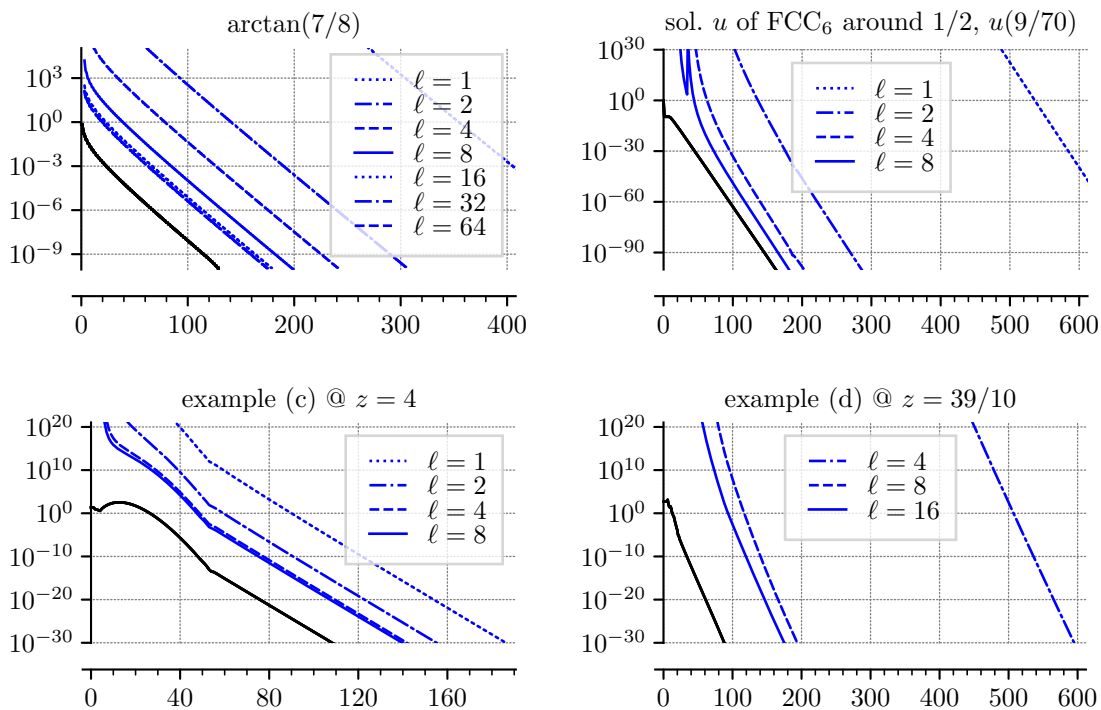

Figure 6. Remainder bounds computed by Algorithm 10 for various values of the tuning parameter ℓ (see Section 9.4). The bottommost curve (in black) is the actual truncation error, the remaining curves from top to bottom correspond to increasing values of ℓ.

point ζ	0.95		4.75		9.5	
truncation order n	50	100	50	100	50	100
NumGfun	$6.2 \cdot 10^{-30}$	$2.5 \cdot 10^{-78}$	$5.5 \cdot 10^{5}$	$1.9 \cdot 10^{-8}$	$1.1 \cdot 10^{22}$	$2.5 \cdot 10^{22}$
vdH2001	$2.7 \cdot 10^{-35}$	$7.5 \cdot 10^{-82}$	$6.1 \cdot 10^{0}$	$1.3 \cdot 10^{-11}$	$5.5 \cdot 10^{23}$	$4.9 \cdot 10^{23}$
vdH2003	$4.7 \cdot 10^{-37}$	$3.4 \cdot 10^{-83}$	$1.1 \cdot 10^{-1}$	$5.5 \cdot 10^{-13}$	$4.1 \cdot 10^{23}$	$4.1 \cdot 10^{23}$
ACETAF	$3.2 \cdot 10^{-48}$	$6.2 \cdot 10^{-77}$	$2.0 \cdot 10^{-5}$	$1.3 \cdot 10^{-28}$	$4.2 \cdot 10^{4}$	$1.2 \cdot 10^{3}$
ore_algebra	$8.6 \cdot 10^{-50}$	$5.2 \cdot 10^{-101}$	$2.9 \cdot 10^{-14}$	$1.4 \cdot 10^{-30}$	$7.2 \cdot 10^{3}$	$2.7 \cdot 10^{2}$
reference	$6.9 \cdot 10^{-50}$	$4.1 \cdot 10^{-101}$	$5.0 \cdot 10^{-15}$	$2.7 \cdot 10^{-31}$	$3.6 \cdot 10^{0}$	$2.2 \cdot 10^{-1}$

Table 1. Bounds on the tails $|u_{n:}(\zeta)|$ of the series defined by (42) computed by various methods (see Section 9.5). All values have been rounded upwards to two significant digits.

changing z to $5\,i + 7 + z$, whence the special distribution of coefficient sizes):

$$\begin{aligned}
&(-9\,z^4 + (-179\,i - 254)z^3 + (-3790\,i - 1356)z^2 + (-22352\,i + 6164)z - 31888\,i + 38654))\,D_z^3 \\
&+ (29\,i\,z^4 + (815\,i - 582)\,z^3 + (4208\,i - 12268)\,z^2 + (-21341\,i - 71530)z - 127224\,i - 98798)\,D_z^2 \\
&+ ((i+1)\,z^4 + (41\,i + 7)\,z^3 + (470\,i - 189)\,z^2 + (1981\,i - 2407)\,z + 1555\,i - 7918)\,D_z \\
&+ ((-4\,i + 1)z^4 + (-96\,i + 107)z^3 + (-256\,i + 1865)z^2 + (4867\,i + 9840)\,z + 20950\,i + 11833) \qquad .
\end{aligned}$$

The evaluation point is a simple rational halfway from the circle of convergence, and we select the initial values at random.

(d) An operator of order 7 and degree 8 of the form

$$((20\,i - 1)\,z^8 + \cdots + 178371756\,i + 577700775)\,D_z^7 + \cdots + ((1 - i)\,z^8 + \cdots - 42079467\,i - 23600391)$$

obtained in a similar way, also evaluated at a point halfway from the circle of convergence and with random initial values.

We can observe that going from $\ell = 1$ to $\ell = 2$ already brings very significant improvements, and that larger values of ℓ are useful for nontrivial examples. Clearly, the heuristic $\ell \approx s/2$ suggested earlier is but a crude rule of thumb. An implementation that adaptively increases ℓ starting from $\ell = 2$ based on a heuristic measure of tightness is likely to perform better. These conclusions are consistent with experiments on various other examples. We also see again that operators with a few tens of monomials and large (but not huge) coefficients can be challenging in terms of overestimation, at least at low precision.

A second parameter involved in the trade-off between speed and accuracy is the optional step of Algorithm 22, whose effect already was illustrated on Figure 2. This optional step can bring huge benefits on some inputs but makes no difference in the most common cases, while being relatively expensive. Hence, it makes sense to only run it when the previous step does not suffice.

As regards the other algorithmic variants mentioned in this article, only the more accurate of the options is typically relevant for large operators. In particular, trying to save time by doing with a single shared lower bound on the singularities, as suggested in Remark 11, only makes sense for very simple operators.

9.5. Comparison with existing software. A systematic comparison with the related methods discussed in Section 2 would be meaningless, as they were designed for different goals, widely differ in scope, often (even more than ours) involve tuning parameters with a crucial influence on the quality of the results, and usually are not implemented. Nevertheless, it makes sense to check that, in simple cases where several methods apply, our bounds are no worse than what could be computed by other approaches.

We limit ourselves to a very simple example borrowed from Neher [37, Ex. 3], namely the problem of bounding the remainders of the Taylor expansion of $u(z) = \cos(z)/(z^2 + 101)$, solution of

$$\left((z^2 + 101)\,D_z^2 + 4\,z\,D_z + (z^2 + 103)\right) \cdot u = 0, \qquad u(0) = 1/101, \quad u'(0) = 0. \tag{42}$$

Table 1 lists bounds on $u_{n:}(\zeta)$ for several values of ζ and n computed by the following methods:

NumGfun. The author's own Maple package NumGfun [27], whose limitations prompted the present work. We use the function `bound_diffeq_tail()` with no particular options or configuration, and substitute the appropriate ζ and n in the result.

vdH2001. A method described by van der Hoeven [43, Sec. 2], here applied manually with the help of non-rigorous floating-point computations. While the description leaves out a number of algorithmic details related to the computation of bounds on rational functions, the optimal choice of parameters ($\lambda = 101^{-1/2}$, $M_0 = 103/101$, $M_1 = 0$) is unambiguous here thanks to the special shape of our example. Like the "a priori" bounds of Section 8.2 and for similar reasons, this method is highly sensitive to the choice of the remaining parameter μ, which we select by numerical optimization to minimize the bound. The six test cases of Table 1 respectively use $\mu = 0.139$, $\mu = 0.121$, $\mu = 0.139$, $\mu = 0.121$, $\mu = 0.105$, and $\mu = 0.105$.

vdH2003. A related method also suggested by van der Hoeven [44, Sec. 3.5], no longer involving μ. Again, it is possible for our particular example to choose optimal parameter values $\alpha = 101^{-1/2}$, $M = 103/101$, $C = 1/101$. We assume that the remainder of the majorant series obtained as output can be bounded as tightly as needed.

ACETAF. We also list bounds obtained by Neher [37] using version 2.71 of ACETAF [12]. ACETAF is mainly designed for computing bounds on the coefficients and remainders of series *coefficients* of linear analytic ODEs before applying the method of majorants. Therefore, unlike the other techniques we are comparing, it does not start from the equation (42) but from the closed-form expression of $u(z)$. Both the quality of the bounds it returns and the running time vary widely depending on the values of several tuning parameters. We only list the best of the three bounds reported in Neher's article for each instance.

ore_algebra. Finally, we run the implementation in ore_algebra of our algorithm, with $\ell = 2$, using the simplified method of Remark 11 for bounding the denominator.

Only ACETAF approaches the accuracy of the bounds of the present paper in some cases. In addition, the alternative methods considered here tend to degrade rapidly, in quality, running time, or both, as the size of the operator increases, so that the gap would likely be wider for realistic problems.

Bibliography

[1] R. Barrio, M. Rodríguez, A. Abad, and F. Blesa. Breaking the limits: the Taylor series method. *Applied Mathematics and Computation*, 217, 2011.

[2] Alin Bostan and Éric Schost. Polynomial evaluation and interpolation on special sets of points. *Journal of Complexity*, 21(4):420–446, 2005.

[3] Richard P. Brent and Paul Zimmermann. *Modern Computer Arithmetic*. Cambridge University Press, 2010.

[4] David Broadhurst. Bessel moments, random walks and Calabi-Yau equations. 2009.

[5] Augustin Cauchy. Mémoire sur l'emploi du nouveau calcul, appelé calcul des limites, dans l'intégration d'un système d'équations différentielles. *Comptes-rendus de l'Académie des Sciences*, 15:14, jul 1842. Reproduced in [6], §169, p. 5-17.

[6] Augustin Louis Cauchy. *Oeuvres complètes d'Augustin Cauchy, Ière série, tome VII*. Gauthier-Villars, 1892.

[7] David V. Chudnovsky and Gregory V. Chudnovsky. Computer algebra in the service of mathematical physics and number theory. In [8], pages 109–232.

[8] David V. Chudnovsky and Richard D. Jenks, editors. *Computers in Mathematics*, volume 125 of *Lecture Notes in Pure and Applied Mathematics*. Dekker, 1990. Talks from the International Conference on Computers and Mathematics, Stanford University, 1986.

[9] J. H. Davenport and M. Mignotte. On finding the largest root of a polynomial. *Modélisation Mathématique et Analyse Numérique*, 24:693–696, 1990.

[10] Digital library of mathematical functions. 2010. Companion to the NIST Handbook of Mathematical Functions [38].

[11] Zilin Du and Chee Yap. Uniform complexity of approximating hypergeometric functions with absolute error. In Sung-Il Pae and Hyungju Park, editors, *Proceedings of the 7th Asian Symposium on Computer Mathematics (ASCM 2005)*, pages 246–249. Korea Institute for Advanced Study, 2005.

[12] Ingo Eble and Markus Neher. ACETAF: A software package for computing validated bounds for Taylor coefficients of analytic functions. *ACM Transactions on Mathematical Software*, 29(3):263–286, 2003.

[13] Ferdinand Georg Frobenius. Über die Integration der linearen Differentialgleichungen durch Reihen. *Journal für die reine und angewandte Mathematik*, 76:214–235, 1873.

[14] Jürgen Gerhard. *Modular Algorithms in Symbolic Summation and Symbolic Integration*, volume 3218 of *Lecture Notes in Computer Science*. Springer, 2004.

[15] Thomas Grégoire. Certified polynomial approximations for D-finite functions. Rapport de stage, École normale supérieure de Lyon, aug 2012.

[16] Anthony J. Guttmann. Lattice Green functions and Calabi-Yau differential equations. *Journal of Physics A: Mathematical and Theoretical*, 42(23):232001, 2009.

[17] S. Hassani, Ch. Koutschan, J.-M. Maillard, and N. Zenine. Lattice Green functions: the d-dimensional face-centered cubic lattice, $d = 8, 9, 10, 11, 12$. *Journal of Physics A: Mathematical and Theoretical*, 49(16), 2016.

[18] Lothar Heffter. *Einleitung in die Theorie der linearen Differentialgleichungen*. Teubner, Leipzig, 1894.

[19] Peter Henrici. *Applied and Computational Complex Analysis*, volume II. Wiley-Interscience, 1977.

[20] Peter Henrici. *Applied and Computational Complex Analysis*, volume III. Wiley-Interscience, 1986.

[21] Einar Hille. *Ordinary differential equations in the complex domain*. Wiley, 1976. Dover reprint, 1997.

[22] Fredrik Johansson. Computing hypergeometric functions rigorously. Technical Report 1606.06977, arXiv, jun 2016.

[23] Fredrik Johansson. Arb: efficient arbitrary-precision midpoint-radius interval arithmetic. *IEEE Transactions on Computers*, 66(8):1281–1292, 2017.

[24] Manuel Kauers, Maximilian Jaroschek, and Fredrik Johansson. Ore polynomials in Sage. In Jaime Gutierrez, Josef Schicho, and Martin Weimann, editors, *Computer Algebra and Polynomials*, pages 105–125. Springer, 2015.

[25] Manuel Kauers and Peter Paule. *The Concrete Tetrahedron: Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates*. Springer, 2011.

[26] Christoph Koutschan. Lattice Green's functions of the higher-dimensional face-centered cubic lattices. *Journal of Physics A: Mathematical and Theoretical*, 46(12):125005, 2013.

[27] Marc Mezzarobba. NumGfun: a package for numerical and analytic computation with D-finite functions. In [47], pages 139–146.

[28] Marc Mezzarobba. *Autour de l'évaluation numérique des fonctions D-finies*. Thèse de doctorat, École polytechnique, nov 2011.

[29] Marc Mezzarobba. Rigorous multiple-precision evaluation of D-Finite functions in SageMath. Technical Report 1607.01967, arXiv, 2016. Extended abstract of a talk at the 5th International Congress on Mathematical Software.

[30] Marc Mezzarobba and Bruno Salvy. Effective bounds for P-recursive sequences. *Journal of Symbolic Computation*, 45(10):1075–1096, 2010.

[31] Ramon E. Moore. *Interval arithmetic and automatic error analysis in digital computing*. PhD thesis, Stanford University, 1962. Published as Applied Mathematics and Statistics Laboratories Technical Report No. 25.

[32] MPFR Team. The MPFR library: algorithms and proofs. 2001-. Available in the MPFR source tree.

[33] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105(1):21–68, 1999.

[34] M. Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45(1):236–262, 2007.

[35] Markus Neher. An enclosure method for the solution of linear ODEs with polynomial coefficients. *Numerical Functional Analysis and Optimization*, 20:779–803, 1999.

[36] Markus Neher. Geometric series bounds for the local errors of Taylor methods for linear n-th-order ODEs. In Götz Alefeld, Jiri Rohn, Siegfried M. Rump, and Tetsuro Yamamoto, editors, *Symbolic Algebraic Methods and Verification Methods*, pages 183–193. Springer, 2001.

[37] Markus Neher. Improved validated bounds for Taylor coefficients and for Taylor remainder series. *Journal of Computational and Applied Mathematics*, 152:393–404, 2003.

[38] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors. *NIST Handbook of Mathematical Functions*. Cambridge University Press, 2010.

[39] Edgar Girard Croker Poole. *Introduction to the theory of linear differential equations*. Clarendon Press, New York, 1936.

[40] Robert Rihm. Interval methods for initial value problems in ODEs. In Jürgen Herzberger, editor, *Topics in Validated Computations: Proceedings of the IMACS-GAMM International Workshop on Validated Computations, University of Oldenburg*, Elsevier Studies in Computational Mathematics, pages 173–207. Elsevier, 1994.

[41] Richard P. Stanley. *Enumerative combinatorics*, volume 2. Cambridge University Press, 1999.

[42] Joris van der Hoeven. Fast evaluation of holonomic functions. *Theoretical Computer Science*, 210(1):199–216, 1999.

[43] Joris van der Hoeven. Fast evaluation of holonomic functions near and in regular singularities. *Journal of Symbolic Computation*, 31(6):717–743, 2001.

[44] Joris van der Hoeven. Majorants for formal power series. Technical Report 2003-15, Université Paris-Sud, Orsay, France, 2003.

[45] Joris van der Hoeven. Efficient accelero-summation of holonomic functions. *Journal of Symbolic Computation*, 42(4):389–428, 2007.

[46] P. G. Warne, D. A. P. Warne, J. S. Sochacki, G. E. Parker, and D. C. Carothers. Explicit a-priori error bounds and adaptive error control for approximation of nonlinear initial value differential systems. *Computers and Mathematics with Applications*, 52(12):1695–1710, 2006.

[47] Stephen M. Watt, editor. *ISSAC '10: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. ACM, 2010.

[48] N. Zenine, S. Hassani, and J.-M. Maillard. Lattice Green functions: the seven-dimensional face-centred cubic lattice. *Journal of Physics A: Mathematical and Theoretical*, 48(3), 2015.