Maple Package on Formal Power Series

Dominik Gruntz
Institute for Scientific Computing
ETH Zürich
CH-8092 Zürich
gruntz@inf.ethz.ch

Wolfram Koepf
Konrad-Zuse-Zentrum für Informationstechnik
Heilbronner Str. 10
D-10711 Berlin
koepf@zib-berlin.de

March 1, 1995

1 Introduction

Formal Laurent-Puiseux series of the form

$$f(x) = \sum_{k=k_0}^{\infty} a_k x^{k/n} \tag{1}$$

with coefficients $a_k \in \mathbb{C}$ ($k \in \mathbb{Z}$) are important in many branches of mathematics. MAPLE supports the computation of *truncated* series with its `series` command, and through the `powseries` package infinite series are available. In the latter package the series is represented as a table of coefficients that have already been determined together with a function for computing additional coefficients. This representation is known as *lazy evaluation* [7]. But these tools fail, if one is interested in an explicit formula for the coefficients a_k.

In this article, which is a condensed version of [6], we describe the MAPLE implementation of an algorithm presented in [9]–[13] which computes an *exact* formal power series (FPS) of a given function. This procedure enables the user to reproduce most of the results of the extensive bibliography on

series [8]. We will give an overview of the algorithm and then present some parts of it in more detail.

This package is available through the MAPLE-share library with the name `FPS`. We will flavor this procedure with the following example. ¿ with(share): ¿ readshare(FPS, calculus): ¿ FormalPowerSeries(exp(x) * sin(x), x=0);

infinity —— k 1/2 k (2) sin(1/4 k Pi) x) ——————— / k! —— k = 0

The main method used works as follows: in a first step, a holonomic, i.e. a homogeneous linear differential equation with polynomial coefficients (DE) is generated for a given function $f(x)$. Then this DE is converted to an equivalent holonomic recurrence equation (RE) for the corresponding coefficients a_k. In a final step, this RE is solved if possible. Note, that the class of functions for which a holonomic DE is found includes all sums, products and compositions with algebraic functions of elementary functions for which holonomic DEs are known [16, 15, 12]. In particular, the following functions are supported: $\exp(x)$, $\sin(x)$, $\cos(x)$, $\arcsin(x)$, $\arctan(x)$, $\mathrm{erf}(x)$, Bessel functions, the Hankel functions Hankel1(n, x) and Hankel2(n, x) [1, (9.1)], the Fibonacci polynomials Fibonacci(n, x), the Whittaker functions WhittakerM(n, m, x) and WhittakerW(n, m, x) [1, (13.4)], and families of orthogonal polynomials JacobiP(n, a, b, x), GegenbauerC(n, a, x), ChebyshevT(n, x), ChebyshevU(n, x), LegendreP(n,x),LaguerreL(n,a,x), and HermiteH(n, x) [1, (22.7) and (22.8)]. The `FPS` package just knows these functions under these names.

2 Search for the Holonomic DE

In this section we present the algorithm that searches for a holonomic DE of degree k for a given function f. We set up the equation

$$f^{(k)}(x) + \sum_{j=0}^{k-1} A_j \, f^{(j)}(x) = 0$$

and expand it. Then we collect the coefficients of all the rationally dependent terms and equate them to zero. For testing whether two terms are rationally dependent, we divide one by the other and test whether the quotient is a rational function in x or not. This is an easy and fast approach. Of course, we could also use the Risch normalization procedure [14, 2] to generate the rationally independent terms, but firstly this normalization is

rather expensive and works only for elementary functions and secondly, our simplified approach will not lead to wrong results, it may at most happen that we miss a simpler solution, which, in practice, rarely happens, however. This procedure by its own is available under the name `SimpleDE`.

It may happen, that for a given function $f(x)$ a DE of degree k exists, which has neither constant coefficients (which we call the *explike case*), nor is the corresponding RE of hypergeometric type (3) and hence no closed form for the formal Laurent-Puiseux series (LPS) can be computed. In this situation, the `FPS` implementation looks for a DE of higher degree which will have free parameters as from the existence of a DE of degree k follows the existence of families of DEs of higher degree. These parameters can be set freely and we can try to choose them in such a way, that we can use our tools to compute a formal LPS, i.e. we either need a RE of hypergeometric type or a DE with constant coefficients.

In order to obtain a RE of hypergeometric type we convert the DE into the corresponding RE and try to set all the coefficients but two to zero. For example, let $f(x) = x^{-1}\,e^x\,\sin x$. We find DEs of degree 2 and of degree 3, but neither of the corresponding REs can be solved. The RE which corresponds to the DE of degree 4 has the form

$$\sum_{j=0}^{4} p_j(k)\,a_{k+j} = 0$$

where

$$\begin{aligned}
p_0(k) &= 2A_2 + 4A_3 + 4\\
p_1(k) &= -4A_2 - 10A_3 - 16 - 2A_3k - 2A_2k\\
p_2(k) &= 18A_3 + 5A_2k + 6A_2 + 48 + 8k + 6A_3k + A_2k^2\\
p_3(k) &= (4+k)(A_3k^2 + 2A_3k - 24 - 3A_3)\\
p_4(k) &= (k+5)(4+k)(k^2+k+6)
\end{aligned}$$

The solution of the equations $p_1(k) = 0, p_2(k) = 0, p_3(k) = 0$ (forcing that only two terms of the sum remain) with respect to A_2 and A_3 is

$$A_2 = -8\frac{k^2+5k+12}{k^3+4k^2+k-6}, \quad A_3 = \frac{24}{k^2+2k-3}.$$

This results (after multiplying by $\frac{(k+2)\,(k+3)}{k^2+k+6}$) in the following RE of hypergeometric type for $f(x)$,

$$(k+5)\,(k+4)\,(k+3)\,(k+2)\;a_{k+4}+4\,a_k=0$$

with symmetry number $m=4$ (3). Of course, we try to keep the symmetry number, that is the number of resulting sums, as small as possible. The final result for this example is ¿ FormalPowerSeries(exp(x)*sin(x)/x, x); /infinity /infinity — —— k (- k) k (4 k)— — —— k (- k) k (1 + 4 k)— — (-1) 64 256 x — — (-1) 64 256 x — —) ————————————-— + —) ———————————— — / (1 + 4 k)! — — / (1 + 4 k)! (2 k + 1) — — —— — — —— — k = 0 / k = 0 / /infinity — —— k (- k) k (4 k + 2)— — (-1) 64 256 x — + —) —— ———————————— — / (4 k + 3) (1 + 4 k)! (2 k + 1)— — —— — k = 0 / Note that one of the four sums (with the powers x^{4k+3}) vanishes as a result of a vanishing initial coefficient.

If this approach fails, we try to choose the parameters such that the DE gets constant coefficients. Let us look at a rather similar example, $f(x)=x\,e^x\,\sin(2x)$. We again find DEs of second and third order whose corresponding REs cannot be solved. The DE of degree 4 has two free parameters A_2 and A_3:

$$\begin{aligned}
&x^2\,\frac{d^4}{dx^4}f(x)+A_3x^2\,\frac{d^3}{dx^3}f(x)+A_2x^2\,\frac{d^2}{dx^2}f(x)+\\
&\quad\Big((A_3-2\,A_2+12)\,x^2+(-6\,A_3-2\,A_2+4)\,x\Big)\;\frac{d}{dx}f(x)+\\
&\quad\Big((10\,A_3+5\,A_2-5)x^2+(14\,A_3+2\,A_2+28)x+(6\,A_3+2\,A_2-4)\Big)\,f(x)=0\,.
\end{aligned}$$

The DE will have constant coefficients, if both A_2 and A_3 are constants and if they meet the following equations

$$\begin{aligned}
6A_3+2A_2-4 &= 0\\
14A_3+2A_2+28 &= 0
\end{aligned}\quad.$$

The solution of this system of equations is $A_2=14$ and $A_3=-4$. If we insert these values in the differential equation and divide by x^2, then we get the DE

$$\frac{d^4}{dx^4}f(x)-4\frac{d^3}{dx^3}f(x)+14\frac{d^2}{dx^2}f(x)-20\frac{d}{dx}f(x)+25f(x)=0$$

which has constant coefficients leading to a constant coefficient RE for b_k given by $f(x) = \sum \frac{b_k}{k!} x^k$ that can be solved: ¿ FormalPowerSeries(x*exp(x)*sin(2*x),x);

infinity —— / k 1/2 k 1/2 — (5) cos(k arctan(2)) k (5) sin(k arctan(2)) k— k) —- 2/5 ———————— + 1/5 ———————— x / k! k! / —— k = 0 In the `gfun` package of the share library [15], the procedure '`diffeq` + `diffeq`'$(eq_1, eq_2, F(x))$ and '`diffeq` ∗ `diffeq`'$(eq_1, eq_2, F(x))$ calculate holonomic DEs for the sum and product of solutions f_1 and f_2 of $eq1$ and $eq2$, respectively. These procedures could also be used to derive holonomic DEs for expressions given as sums and products of elementary functions like the ones mentioned in the introduction. Note, however, that `SimpleDE` usually generates the holonomic DE of lowest order, e.g. ¿ SimpleDE(sqrt(1+x)+1/sqrt(1+x),x);

/ d 2 (2 + x) (1 + x) ——- F(x)— - x F(x) = 0 dx / whereas '`diffeq+diffeq`' calculates a holonomic DE valid for all linear combinations of f_1 and f_2, e.g. ¿ eq1 := SimpleDE(sqrt(1+x),x,F); / d eq1 := (2 + 2 x) ——- F(x)— - F(x) = 0 dx /

¿ eq2 := SimpleDE(1/sqrt(1+x),x,F); / d eq2 := (2 + 2 x) ——- F(x)— + F(x) = 0 dx / ¿ 'diffeq+diffeq'(eq1, eq2, F(x));

2 (2) F(x) + (- 4 - 4 x) D(F)(x) + (- 4 - 8 x - 4 x) D (F)(x) An upper bound for the order of the differential equation can be computed a priori [16], however this bound is not sharp for all cases. Therefore, and since the complexity of the algorithm increases rapidly with the order, we stop the search if no DE of order five has been found. This predefined upper bound can be overwritten by specifying an additional integer argument when calling the `FormalPowerSeries` function.

3 Conversion to the Holonomic Recurrence Equation

It is well-known, that the conversion of a holonomic DE into the corresponding holonomic RE is done by the substitution

$$x^l f^{(j)}(x) \mapsto (k+1-l)_j \cdot a_{k+j-l} \tag{2}$$

into the DE.

The search of a DE and its conversion to the RE is directly available through the command `SimpleRE`. We see that `AiryAi`(x^2)[1] is of hypergeo-

[1] `AiryAi` is the Airy wave function which is available in MAPLE V Release 4. In previous

metric type with symmetry number $m = 6$, whereas the second RE is not of hypergeometric type. ¿ SimpleRE(AiryAi(x^2), x);

(k - 1) (k + 1) a(k + 1) - 4 a(k - 5) = 0

¿ SimpleRE(x/(1-x^2), x);

(1 - k) a(k) + (k - 1) a(k - 1) + (k - 1) a(k - 2) = 0 The latter example is the generating function of the Fibonacci numbers, and we get the expected RE. Note, that the common factor $(k-1)$ ensures, that the RE holds $\forall\, k \in \mathbb{Z}$.

4 Solving a Recurrence Equation of Hypergeometric Type

If the recurrence equation of a function $f(x)$ is of the form

$$Q(k)\, a_{k+m} = P(k)\, a_k \tag{3}$$

(P, Q polynomials) then we call f to be of hypergeometric type, and m the symmetry number of the corresponding series representation. The explicit formula for the coefficients can be found by the hypergeometric coefficient formula and some initial conditions. Based on the analysis of the polynomials $P(k)$ and $Q(k)$, we will convert the RE into one corresponding to a *Taylor series* by applying a sequence of transformations stated in the following lemma which preserve the hypergeometric type.

LEMMA 1 *Let f be a formal Laurent series (FLS) of hypergeometric type with representation $\sum_{k=k_0}^{\infty} a_k x^{k/n}$ and $a_{k_0} \neq 0$, whose coefficients a_k satisfy a RE of the form*

$$\begin{aligned} a_{k+m} &= R(k)\, a_k \quad && \text{for } k \geq k_0 \\ a_k &= A_k && \text{for } k = k_0, k_0+1, \ldots, k_0+m-1, \end{aligned}$$

then the following functions are of hypergeometric type, too. Their coefficients b_k satisfy a RE whose relation to the RE of f is also given.

(a)	$x^n f(x)$	$b_{k+m} = R(k-n)\, b_k$	$n \in \mathbb{Z}$
(b)	$f(Ax)$	$b_{k+m} = A^m\, R(k)\, b_k$	$A \in \mathbb{C}$
(c)	$f(x^n)$	$b_{k+n\,m} = R(k/n)\, b_k$	$n \in \mathbb{N}$
(d)	$f(x^{1/m})$	$b_{k+1} = R(k\,m)\, b_k$	
(e)	$f'(x)$	$b_{k+m} = \frac{k+m+1}{k+1}\, R(k+1)\, b_k$	

versions this function was called `Ai`, but the derivative was defined differently which led to a different DE and RE.

First of all, we inspect the roots of $P(k)$ and $Q(k)$ of the RE (3). If there are any non-integer rational roots, then we know that f corresponds (possibly) to a Puiseux series. In this case we transform f to a function of Laurent type by an application of transformation (c), where n is the least common multiple of the denominators of all rational roots of $P(k)$ and $Q(k)$. The FLS we get when we solve the transformed RE can be transformed back to the LPS of f by substituting x by $x^{1/n}$.

Let us now assume that f can be expanded in a FLS. We reduce this problem to solving the RE of a function which has a FPS expansion. For that purpose we remove the finite pole of f at the origin by multiplying f with a suitable power of x (transformation (a)). From the information of the RE we can determine which power we have to use. Let us assume that f may be expanded in a Laurent series, i.e. that $\exists\ k_0\ :\ \forall\ k \leq k_0\ :\ a_k = 0$. From these known coefficients we can derive further ones using the given RE in the form

$$a_{k+m} = \frac{P(k)}{Q(k)}\, a_k. \tag{4}$$

If we know that $a_k = 0$ then also $a_{k+m} = 0$ provided that $Q(k) \neq 0$. Let k_{min} be the smallest integer root of $Q(k)$. Consequently $a_k = 0\ \forall\ k < k_{min} + m$. From this it follows, that

$$g = x^{-(k_{min}+m)}\, f$$

may be expanded into a FPS, i.e. g has no pole at the origin, given the assumption that f may be expanded in a Laurent series. This latter assumption can be tested by computing the limit of g as x tends to 0. This limit must be finite. (Since we also allow logarithmic singularities, we test in fact whether the limit of $x\, g$ is 0.) If this is not the case, then the assumption that f may be expanded into a FLS is wrong and f must have an essential singularity, i.e. $\nexists\ k_0\ :\ \forall\ k \leq k_0\ :\ a_k = 0$. Otherwise the FPS of g exists and can again easily be transformed back to the FLS of f.

What we finally have to show is how to solve a RE which corresponds to a given function f which has a FPS expansion. The RE (4) is valid $\forall\ k\ :\ Q(k) \neq 0$, especially $\forall\ k > k_{max}$ where k_{max} is the largest root of $Q(k)$. Consequently we must determine a_k for $k = 0, 1, \ldots, k_{max} + m$ and have to solve the hypergeometric RE for $x^{-k}\, f(x^{1/m})$ for $k > k_{max}$. We investigate now how this condition can be weakened.

The coefficient a_k is given by the limit $\lim\limits_{x\to 0} f^{(k)}(x)/k!$. Let us assume, that this limit is finite. Then the hypergeometric RE can be solved in the

case that $\forall\ j \geq 0\ :\ Q(k + j\,m) \neq 0$, otherwise simply $a_k\,x^k$ is added to the result. Moreover note, that

$$(P(k) = 0 \vee a_k = 0) \wedge \forall\ j \geq 0\ :\ Q(k+j\,m) \neq 0 \wedge a_k \text{ is finite} \Longrightarrow \forall\ j > 0\ :\ a(k+j\,m) = 0$$

and in this case the RE does not have to be solved and it is enough to add $a_k\,x^k$ to the result. The indices $k + j\,m, j > 0$ no longer need to be considered.

If a_k is infinite, then we found a logarithmic singularity which we can remove by working with

$$g = (x^{-k}\,f(x))'$$

and by integrating and shifting the resulting power series S. The RE of g can be obtained from the RE of f by applying transformations (a) and (e). The constant term which we lose by the differentiation can be determined by computing the limit

$$\lim_{x \to 0} f(x)/x^k - \int S(x)\,dx.$$

Note that g in general is a function with a FLS expansion and we first must remove the pole to get a function with a FPS expansion. This is the reason why we have chosen a recursive implementation of the RE solver. Every application of a transformation of Lemma 1 is nothing but a recursive call of the RE solver. One may inspect which steps the algorithm performs by assigning the variable `infolevel[FormalPowerSeries]`. As an example we trace the computation for $f(x) = x^{-1} \sin\sqrt{x}$. A DE of degree 2 is found whose corresponding RE is of hypergeometric type. From the root of $2k + 3$ it follows that the Puiseux number is 2 and so transformation (c) with $n = 2$ is applied. The resulting function is of Laurent type. The smallest integer root of $Q(k)$ of the transformed RE is -4 and the symmetry number is $m = 2$, hence we multiply the function with x^2 and adjust the RE accordingly. We end up with the function $\sin x$ whose FPS can be computed directly. This result is then transformed back to the LPS of $f(x)$ according to the two transformations we applied. ¿ infolevel[FormalPowerSeries] := 4: ¿ FormalPowerSeries(sin(sqrt(x))/x,x); FPS/FPS: looking for DE of degree 1 FPS/FPS: looking for DE of degree 2 FPS/FPS: DE of degree 2 found. FPS/FPS: DE = 2 4 x F"(x) + 10 x F'(x) + (2 + x) F(x) = 0

FPS/hypergeomRE: RE is of hypergeometric type. FPS/hypergeomRE: Symmetry number m = 1 FPS/hypergeomRE: RE: 2 (k + 2) (2 k + 3) a(k +

1) = - a(k) FPS/hypergeomRE: RE modified by k = 1/2*k FPS/hypergeomRE: =¿ f := sin(x)/x^2

FPS/hypergeomRE: RE is of hypergeometric type. FPS/hypergeomRE: Symmetry number m = 2 FPS/hypergeomRE: RE: (k + 4) (k + 3) a(k + 2) = - a(k) FPS/hypergeomRE: working with x^2 $*$ $fFPS/hypergeomRE$:=> $f := sin(x)$

FPS/hypergeomRE: RE is of hypergeometric type. FPS/hypergeomRE: Symmetry number m = 2 FPS/hypergeomRE: RE: (k + 2) (k + 1) a(k + 2) = - a(k) FPS/hypergeomRE: RE valid for all k ¿= 0 FPS/hypergeomRE: a(0) = 0 FPS/hypergeomRE: a(2*j) = 0 for all j¿0. FPS/hypergeomRE: a(1) = 1

infinity —— k (k - 1/2) (-1) x) ————————- / (2 k + 1)! —— k = 0

¿ infolevel[FormalPowerSeries] := 1:

5 Rational Algorithm

If the given function is rational in x, first the complex partial fraction decomposition of $f(x)$ is calculated. Each term of the form $\frac{c}{(x-\alpha)^j}$ can be expanded by the binomial series whose coefficients are

$$a_k = \frac{(-1)^j\, c}{\alpha^{j+k}} \left(\begin{array}{c} j+k-1 \\ k \end{array} \right).$$

¿ FormalPowerSeries(1/((x-1)^2 $*$ $(x-2)), x$);

infinity —— k k k (k! - 2 2 k! + 2 (1 + k)! 2) x) - 1/2 ———————————— / k —— 2 k! k = 0

¿ FormalPowerSeries((1+x+x^2 + x^3)/((x − 1) $*$ (x − 2)), x);

/infinity — —— k k— — (8 2 - 15) x — x + 4 + —) 1/2 ———————— / k — — —— 2 — k = 0 /

¿ FormalPowerSeries(C/(B*A - A*x - B*x+x^2), x);

infinity —— k k k C (A A - B B) x) ———————— / k k —— (A - B) B B A A k = 0 To get the complex partial fraction decomposition we must factor the denominator which may be rather complicated, hence the rational algorithm to compute the full partial fraction expansion presented in [3] may be used. The following example uses this code. This method can be forced to be used, if the environment variable `_EnvExplicit` is set to false. ¿ FormalPowerSeries(1/(x^4 + x + 1), x);

infinity ----- / ----- 2 3 — 27 + 64 alpha - 48 alpha + 36 alpha — k) —) 1/229 ———————————————-— x / — / (1 + k) — ----- — ----- alpha — k = 0 α =

4 A closed form of the Fibonacci numbers can be derived by computing the FPS of their generating function $x/(1{-}x{-}x^2)$. ¿ expand(FormalPowerSeries(x/(1-x-x^2), x));

/infinity — ----- / k k $\|1/2|\ |xx||1/55|)| - |||/|1/2k1/2k||| - - - - - -1/2+1/25)(-1/2-1/25)/|\ k=0/$ The `expand` command converts the coefficient in the usual notation. If the factorization of the numerator is avoided, then the following result is obtained: ¿ $_EnvExplicit := false :> FormalPowerSeries(x/(1-x-x^2), x);$

infinity ----- / ----- — alpha - 2 — k) —) - 1/5 ——————— x / — / (1 + k)— ----- — ----- alpha — k = 0 — 2 — $\alpha = RootOf(-1 +_Z +_Z)/$

¿ $_EnvExplicit :=_E' nvExplicit' :$

6 Additional Examples

In this section we present some more results generated with the procedure `FormalPowerSeries`.

First, we consider the FPS of the Airy wave function `AiryAi`. ¿ FormalPowerSeries(AiryAi(x),x);

/infinity — ----- (- k) (3 k) — 1/3 — 9 x — 3 —) ——————————— — / pochhammer(2/3, k) k!— — ----- — k = 0 / 1/3 ———————————————- GAMMA(2/3)

/infinity — ----- (- k) (3 k + 1) — 1/6 — 9 x — 3 GAMMA(2/3) —) ————————— — / pochhammer(4/3, k) k!— — ----- — k = 0 / - 1/2 ——————————————————— Pi For the next example the internal series command is not able to compute a truncated power series as long a s the parameter a is left as an unknown. ¿ FormalPowerSeries(x^a$*sin(x^2), x);$

infinity ----- k (4 k + 2 + a) (-1) x) ————————— / (1 + 2 k)! ----- k = 0

The following example has a logarithmic singularity at $x = 0$ and hence the FPS of f' is computed. ¿ FormalPowerSeries(arcsech(x), x, right);

/infinity — ----- (- k) (2 k + 2)— — (1 + 2 k)! 4 x — ln(2) - ln(x) - 1/2 —) ——————————-— — / 2 — — ----- (k!) (k + 1) (2 k + 2) — k = 0 /

The next two examples are FPS expansions of two definite integrals,

namely the polylogarithm function and $\int_0^x \mathrm{erf}(t)/t\,dt$. Note that for the second integral we have used the inert function of `Int` which is only a placeholder and does not try to compute a closed form. ¿ FormalPowerSeries(dilog(1-x), x);

infinity — (k + 1) x) ——— / 2 — (k + 1) k = 0

¿ FormalPowerSeries(Int(erf(t)/t,t=0..x), x);

infinity — k (2 k + 1) (-1) x) ————————- / 2 — k! (2 k + 1) k = 0 2 ——————————- 1/2 Pi

The next two examples are FPS of functions which contain orthogonal polynomials. ¿ FormalPowerSeries(exp(-x)*LaguerreL(n,a,x),x); FPS/hypergeomRE: provided that -1 ¡= min(-1,-a-1)

/infinity ——— k k—— — (-1) pochhammer(n + 1 + a, k) x — binomial(n + a, n) —) ———————————— — / pochhammer(1 + a, k) k! ——— —— — k = 0 /

¿ FormalPowerSeries(exp(-x^2) $*$ $HermiteH(n, x), x)$;

/infinity — ——— k (2 k)—— — (1/2 n + 1/2 + k)! (-4) x — cos(1/2 n Pi) (n + 1)! —) ————————————————-—— — / (n + 1 + 2 k) (2 k)! — —— —— — k = 0 / ———————————————————————————— (1/2 n)! (1/2 n + 1/2)!

/infinity — ——— k (2 k + 1)—— — (-4) (1/2 n + k)! x — sin(1/2 n Pi) (n + 1)! —) ———————————— — / (2 k + 1)! — —— —— — k = 0 / + ———————————————————————————— (1/2 n + 1/2)! (1/2 n)!

As we have seen above, we have three tools available to compute a formal power series representation of a given function, and it may happen for some examples, that several of these tools may be applied. Normally the solutions of a RE of hypergeometric type leads to the simplest results, but this is not true in general. Hence we added the possibility for the user to choose a method by adding an additional argument which is either `hypergeometric`, `explike` or `rational`.

¿ f := x*arctan(x)-1/2*ln(1+x^2) :> $FormalPowerSeries(f, x, hypergeometric)$;

/infinity — ——— k (2 k + 2)—— — (-1) x — 1/2 —) ———————— — / (k + 1) (2 k + 1)—— — —— — k = 0 /

¿ FormalPowerSeries(f, x, rational);

infinity — k k (k + 2) ((- I) + I) x) 1/2 ——————————- / k k — I (- I) (k + 1) (k + 2) k = 0

¿ f := sin(x)*exp(x): ¿ FormalPowerSeries(f, x, explike);

infinity — k 1/2 k (2) sin(1/4 k Pi) x) ——————————— / k! — k = 0

¿ FormalPowerSeries(f, x, hypergeometric);
/infinity — —— k (- k) k (4 k + 1)— — (-1) 64 256 x — —) ——— ——————— — / (4 k + 1)! — — —— — k = 0 /
/infinity — —— k (- k) k (4 k + 2)— — (-1) 64 256 x — + —) ——————————— — / (2 k + 1) (4 k + 1)! — — —— — k = 0 /
/infinity — —— k (- k) k (4 k + 3)— — (-1) 64 256 x — + —) —— ———————————— — / (2 k + 1) (4 k + 1)! (4 k + 3)— — —— — k = 0 /

7 Asymptotic Series

The same algorithm can also be used to compute asymptotic expansions. Note, that we only look for asymptotic series of the form of a Laurent-Puiseux series.

One special thing of Laurent-Puiseux asymptotic expansions is, that they are only valid as long as the indeterminate approaches the expansion point from one side. If the expansion point is not ∞, then one may specify with an option `right` or `left` from which side one approaches the expansion point.
¿ FormalPowerSeries(erf(x), x=infinity);
1
¿ FormalPowerSeries(arctan(1/x), x=0, right);
/infinity — —— k (2 k + 1)— — (-1) x — 1/2 Pi - —) ————————-—— / 2 k + 1 — — —— — k = 0 /
¿ FormalPowerSeries(exp(x), x=-infinity);
0
¿ FormalPowerSeries(exp(x), x=infinity): FPS/FPS: ERROR: essential singularity
¿ FormalPowerSeries(exp(x)*Ei(-x) + exp(-x)*Ei(x), x=infinity);
/infinity — —— — — (2 k + 2)— 2 —) (1 + 2 k)! (1/x) — — / —— —— —— — k = 0 /
¿ FormalPowerSeries(exp(x)*(1-erf(sqrt(x))), x=infinity);
infinity —— (- k) (- k) (1/2 + k) (-1) (2 k)! 4 (1/x)) ——————————— ——————— / k! —— k = 0 ———————————————————————— 1/2 Pi By a plot of `arctan(1/x)`, e.g, one may realize that, indeed, the resulting series representation is one sided, only.

8 Conclusion

We have presented an algorithm, and its implementation in MAPLE, to compute FPS. Since it is a goal of Computer Algebra to work with *formal* objects, we think this is a very powerful and valuable addition.

The algorithm is beside of rational operations based on two basic tools: solving systems of equations and computing symbolic limits. No truncated series is ever computed. For the case of asymptotic series, one sided limits are computed and for all other cases complex ones. The power of the procedure for computing LPS stands and falls with the capabilities of the tool for computing limits. The algorithms which are used in MAPLE to compute limits are described in [4, 5].

Further, in cases when the resulting RE cannot be solved explicitly, it can, in principle be used to calculate the coefficients iteratively in a lazy evaluation scheme. This is particularly efficient as the resulting RE always is homogeneous and linear, so that each coefficient can be calculated by finitely many of its predecessors. We note that the algorithm to find a holonomic DE works, and so such a RE always exists, if the input function is constructed by integration, differentiation, addition, multiplication, and the composition with rational functions or rational powers, from functions with the same property [16, 15, 12]. This gives a huge class of functions to which the method can be applied.

References

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions,* Dover Publ., New York, 1964.

[2] M. Bronstein, *Simplification of Real Elementary Functions,* ISSAC'89, ed. G.H. Gonnet, 207 – 211, 1989.

[3] M. Bronstein and B. Salvy, *Full Partial Fraction Decomposition of Rational Functions,* ISSAC'93, ed. M. Bronstein, 157 – 160, 1993.

[4] K.O. Geddes and G.H. Gonnet, *A New Algorithm for Computing Symbolic Limits using Hierarchical Series*, Proceedings of the International Symposium ISSAC 1988, Lecture Notes In Computer Science **358**, 490 – 495, 1988.

[5] G.H. Gonnet and D. Gruntz, *Limit Computation in Computer Algebra*, Technical Report **187**, Department for Computer Science, ETH Zürich, 1992, (submitted to the JSC).

[6] D. Gruntz and W. Koepf, *Formal Power Series,* Preprint SC 93-31, ZIB, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1993.

[7] D. Gruntz, *Infinite Structures in Maple,* MapleTech **1** (2), pp. 19–30, 1994.

[8] E.R. Hansen, *A table of series and products,* Prentice-Hall, Englewood Cliffs, NJ, 1975.

[9] W. Koepf, *Power series in Computer Algebra,* J. Symb. Comp. **13**, 581 – 603, 1992.

[10] W. Koepf, *Examples for the algorithmic calculation of formal Puiseux, Laurent and power series,* SIGSAM Bulletin **27**, 20 – 32, 1993.

[11] W. Koepf, *Algorithmic development of power series*, In: Artificial intelligence and symbolic mathematical computing, Proceedings of the International Conference AISMC-1, Karlsruhe, Germany, August 1992, (J. Calmet and J.A. Campbell, ed.), Lecture Notes in Computer Science **737**, Springer-Verlag Berlin–Heidelberg, 195 – 213, 1993.

[12] W. Koepf and D. Schmersau, *Spaces of functions satisfying simple differential equations,* Konrad-Zuse-Zentrum Berlin (ZIB), Technical Report TR 94-2, 1994.

[13] W. Koepf, *Algorithmic work with orthogonal polynomials and special functions,* Konrad-Zuse-Zentrum Berlin (ZIB), Technical Report TR 94-5, 1994.

[14] R. Risch, *Algebraic Properties of the Elementary Functions of Analysis,* Amer. J. of Math. **101**, 743 – 759, 1979.

[15] B. Salvy and P. Zimmermann, *GFUN: A package for the manipulation of generating and holonomic functions in one variable*, AMS Transactions on Mathematical Software **20**, pp. 163–177, 1994.

[16] R.P. Stanley, *Differentiably finite power series,* Europ. J. Combinatorics **1**, pp. 175–188, 1980.