J. Symbolic Computation (1992) **11**, 581–603

Power Series in Computer Algebra

WOLFRAM KOEPF

Fachbereich Mathematik, Freie Universität Berlin, Deutschland

(Received 27 September 1991)

Formal power series (FPS) of the form $\sum_{k=0}^{\infty} a_k (x-x_0)^k$ are important in calculus and complex analysis. In some Computer Algebra Systems (CASs) it is possible to define an FPS by direct or recursive definition of its coefficients. Since some operations cannot be directly supported within the FPS domain, some systems generally convert FPS to finite truncated power series (TPS) for operations such as addition, multiplication, division, inversion and formal substitution. This results in a substantial loss of information. Since a goal of Computer Algebra is — in contrast to numerical programming — to work with formal objects and preserve such symbolic information, CAS should be able to use FPS when possible.

There is a one-to-one correspondence between FPS with positive radius of convergence and corresponding analytic functions. It should be possible to automate conversion between these forms. Among CASs only Macsyma provides a procedure `powerseries` to calculate FPS from analytic expressions in certain special cases, but this is rather limited.

Here we give an algorithmic approach for computing an FPS for a function from a very rich family of functions including all of the most prominent ones that can be found in mathematical dictionaries except those where the general coefficient depends on the Bernoulli, Euler, or Eulerian numbers. The algorithm has been implemented by the author and A. Rennoch in the CAS Mathematica, and by D. Gruntz in Maple.

Moreover, the same algorithm can sometimes be reversed to calculate a function that corresponds to a given FPS, in those cases when a certain type of ordinary differential equation can be solved.

1. Introduction

The methods we describe for combining *formal power series* (FPS) require that all series be expanded at the same point. Without loss of generality we take $x_0 := 0$, since by the substitution $x \mapsto (x - x_0)$ we get FPS with point of development x_0. Let us therefore consider FPS to be of the form

$$F := \sum_{k=0}^{\infty} a_k x^k \tag{1.1}$$

with coefficients $a_k \in \mathbb{C}$ $(k \in \mathbb{N}_0)$.

The *derivative* F' of an FPS F is defined by

$$F' := \sum_{k=0}^{\infty} k a_k x^{k-1} = \sum_{k=0}^{\infty} (k+1) a_{k+1} x^k \ , \tag{1.2}$$

0747–7171/90/000000 + 00 $03.00/0 © 2004 Academic Press Limited

and the *antiderivative* $\int F$ of an FPS F is given by

$$\int F := \sum_{k=0}^{\infty} \frac{a_k}{k+1} x^{k+1} = \sum_{k=1}^{\infty} \frac{a_{k-1}}{k} x^k \ . \tag{1.3}$$

All the algebraic operations for FPS like addition, multiplication, division, and substitution, can be done by a finite algorithm if one truncates the resulting FPS, i. e. only evaluates the first N coefficients of it (where N is an arbitrary fixed positive integer), which gives a truncated power series. These algorithms are implemented in certain Computer Algebra Systems (CAS), e. g. in Maple (see Maple Reference Manual).

If one is interested in the exact formal result, i. e. in an explicit formula for the coefficients a_k, one knows, however, that most algebraic operations are quite difficult and cannot be done by a finite algorithm.

We shall give an outline of how to resolve these issues for FPS of some special types, in order to implement manipulations of FPS algorithmically in a CAS. This FPS domain should be rich enough for doing calculus, and on the other hand restricted enough to be implementable in a CAS. Our class includes many special functions, but unfortunately does not necessarily deal with linear (or worse) combinations of such general functions. These problems are not trivial, as the examples $\arctan x + e^x$ and $\sec^2 x - \tan^2 x$ can illustrate. In the first case the individual terms can be treated by our approach, but the sum of the two cannot, whereas in the second case the entire sum is covered, but each term individually is not.

2. Series of Hypergeometric Type

We require an assumption that every FPS F has positive radius of convergence $r := 1/\limsup\limits_{k\to\infty} |a_k|^{1/k}$. In this situation the FPS represents an analytic function $f(x) = \sum\limits_{k=0}^{\infty} a_k x^k =: F$ in its disc of convergence $\mathbb{D}_r := \{x \in \mathbb{C} \mid |x| < r\}$, i. e. its sum converges locally uniformly in $\mathbb{D}_r$ to f. So there is a one-to-one correspondence between the functions f analytic at the origin and the FPS F with positive radius of convergence represented by their coefficient sequences $(a_k)_{k\in\mathbb{N}_0}$. We denote this correspondence by $f \leftrightarrow F$. As we are interested in the conversion $f \leftrightarrow F$ the restriction to FPS with positive radius of convergence makes sense even though algebraically this restriction is not necessary.

From the analytical point of view a reasonable family of power series should include the power series of the rational functions and the elementary transcendental functions and their inverses, i. e.

1 the *polynomials* $P := \sum\limits_{k=0}^{N} p_k x^k \quad (N \in \mathbb{N})$,

2 the *rational functions* P/Q, where P and $Q := \sum\limits_{k=0}^{M} q_k x^k$ $(M \in \mathbb{N})$ are polynomials with $q_0 \neq 0$,

3 and the *elementary functions*, i. e.

(a) the *binomial series* $\mathrm{B}_\alpha := \sum\limits_{k=0}^{\infty} \binom{\alpha}{k} x^k \ \leftrightarrow\ (1+x)^\alpha$, where $\binom{\alpha}{k}$ denotes the *binomial coefficient* defined by

$$\binom{\alpha}{k} := \begin{cases} 1 & \text{if } k = 0 \\ \frac{\alpha\cdot(\alpha-1)\cdots(\alpha-k+1)}{k!} & \text{if } k \in \mathbb{N} \end{cases} ,$$

and $k!$ denotes the *factorial*

$$k! := \begin{cases} 1 & \text{if } k = 0 \\ 1\cdot 2\cdots k & \text{if } k \in \mathbb{N} \end{cases} ,$$

(b) the *exponential series* $\mathrm{EXP} := \sum\limits_{k=0}^{\infty} \frac{1}{k!} x^k \;\leftrightarrow\; \exp(x)$,

(c) the *logarithmic series* $\mathrm{LN} := \sum\limits_{k=1}^{\infty} \frac{1}{k} x^k \;\leftrightarrow\; -\ln(1-x)$,

(d) the *sine* and *cosine series* $\mathrm{SIN} := \sum\limits_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} \;\leftrightarrow\; \sin(x)$, and
$\mathrm{COS} := \sum\limits_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k} \;\leftrightarrow\; \cos(x)$,

(e) the *hyperbolic sine* and *cosine series* $\mathrm{SINH} := \sum\limits_{k=0}^{\infty} \frac{1}{(2k+1)!} x^{2k+1} \;\leftrightarrow\; \sinh(x)$, and $\mathrm{COSH} := \sum\limits_{k=0}^{\infty} \frac{1}{(2k)!} x^{2k} \;\leftrightarrow\; \cosh(x)$,

(f) the *tangent* and *cotangent series* $\mathrm{TAN} := \sum\limits_{k=1}^{\infty} \frac{(-1)^{k-1} 4^k (4^k-1) B_{2k}}{(2k)!} x^{2k-1} \;\leftrightarrow\; \tan(x)$, and $\mathrm{x}\cdot\mathrm{COT} := \sum\limits_{k=0}^{\infty} \frac{(-1)^k 4^k B_{2k}}{(2k)!} x^{2k} \;\leftrightarrow\; x\cdot\cot(x)$, where B_n are the *Bernoulli numbers* that are recursively defined by

$$B_n := \begin{cases} 1 & \text{if } n = 0 \\ -\frac{1}{n+1} \sum\limits_{k=0}^{n-1} \binom{n+1}{k} B_k & \text{if } n \in \mathbb{N} \end{cases} \tag{2.1}$$

(see e. g. Walter (1985), p. 161),

(g) the *hyperbolic tangent* and *cotangent series* $\mathrm{TANH} := \sum\limits_{k=1}^{\infty} \frac{4^k (4^k-1) B_{2k}}{(2k)!} x^{2k} \;\leftrightarrow\; \tanh(x)$, and $\mathrm{x}\cdot\mathrm{COTH} := \sum\limits_{k=0}^{\infty} \frac{4^k B_{2k}}{(2k)!} x^{2k} \;\leftrightarrow\; x\cdot\coth(x)$,

(h) the *inverse sine* and *tangent series*
$\mathrm{ARCSIN} := x + \sum\limits_{k=1}^{\infty} \frac{1\cdot 3\cdot 5\cdots(2k-1)}{2^k k!\,(2k+1)} x^{2k+1} \;\leftrightarrow\; \arcsin(x)$, and
$\mathrm{ARCTAN} := \sum\limits_{k=0}^{\infty} \frac{(-1)^k}{2k+1} x^{2k+1} \;\leftrightarrow\; \arctan(x)$,

(i) and the *inverse hyperbolic sine* and *tangent series*
$\mathrm{ARSINH} := x + \sum\limits_{k=1}^{\infty} \frac{(-1)^k 1\cdot 3\cdot 5\cdots(2k-1)}{2^k k!\,(2k+1)} x^{2k+1} \;\leftrightarrow\; \mathrm{arsinh}\,(x)$, and
$\mathrm{ARTANH} := \sum\limits_{k=0}^{\infty} \frac{1}{2k+1} x^{2k+1} \;\leftrightarrow\; \mathrm{artanh}\,(x)$.

From the point of view of an algorithmic approach it seems to be impossible to include all cases where coefficients with dynamically growing recursions like (2.1) occur, so that we exclude those functions from our considerations.

On the other hand, to include many special functions, and to place the remaining above FPS in a more general setting, it is a good idea (and one of the essential ones of this paper) to include

4 the *(generalized) hypergeometric series*

$$ {}_pF_q \left(\begin{array}{cccc} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{array} \middle| \, x \right) := \sum_{k=0}^{\infty} \frac{(a_1)_k \cdot (a_2)_k \cdots (a_p)_k}{(b_1)_k \cdot (b_2)_k \cdots (b_q)_k \, k!} x^k \tag{2.2} $$

where $(a)_k$ denotes the *Pochhammer symbol* (or *shifted factorial*) defined by

$$ (a)_k := \begin{cases} 1 & \text{if } k = 0 \\ a \cdot (a+1) \cdots (a+k-1) & \text{if } k \in \mathbb{N} \end{cases} . $$

Note that $\frac{(a)_k}{k!} = \binom{a+k-1}{k}$.

The coefficients A_k of the hypergeometric series $\sum\limits_{k=0}^{\infty} A_k x^k$ are the unique solution of the special recurrence equation (RE)

$$ A_{k+1} := \frac{(k+a_1) \cdot (k+a_2) \cdots (k+a_p)}{(k+b_1) \cdot (k+b_2) \cdots (k+b_q)(k+1)} \cdot A_k \quad (k \in \mathbb{N}) \tag{2.3} $$

with the initial condition

$$ A_0 := 1 . $$

Note that $\frac{A_{k+1}}{A_k}$ is rational in k. Moreover if $\frac{A_{k+1}}{A_k}$ is a rational function $Q(k)$ in the variable k then the corresponding function f is connected with a hypergeometric series; i. e., if $k = -1$ is a pole of Q, then f corresponds to a hypergeometric series evaluated at some point ax (where a is the quotient of the highest coefficients of the numerator and the denominator of Q); whereas, if $k = -1$ is no pole of Q, then f may be furthermore shifted by some factor x^s ($s \in \mathbb{Z}$).

We further mention that the function f corresponding to the hypergeometric series

$$ f \; \leftrightarrow \; F := {}_pF_q \left(\begin{array}{cccc} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{array} \middle| \, x \right) $$

satisfies the differential equation (DE)

$$ \theta(\theta + b_1 - 1) \cdots (\theta + b_q - 1) f = x(\theta + a_1) \cdots (\theta + a_p) f \tag{2.4} $$

where θ is the differential operator $x \frac{d}{dx}$. An inspection of the hypergeometric DE (2.4) shows that it is of the form ($Q := \max\{p, q\} + 1$)

$$ \sum_{k=0}^{Q} \sum_{j=0}^{Q} c_{jk} x^j f^{(k)} = 0 \tag{2.5} $$

with certain constants $c_{jk} \in \mathbb{C}$ ($j, k = 0, \ldots, Q$). Because of their importance in our development, we call a DE of the form (2.5), i. e. a homogeneous linear DE with polynomial coefficients, *simple*. In Theorems 2.1 and 8.1 we show the existence of a simple DE for more general families of functions.

It is remarkable that the elementary functions that we did not exclude

$$\mathrm{B}_\alpha = {}_1\mathrm{F}_0(-\alpha \mid x)\,, \qquad \mathrm{EXP} = {}_0\mathrm{F}_0(x)\,, \qquad \mathrm{LN} = x \cdot {}_2\mathrm{F}_1\left(\left.\begin{array}{c} 1\ \ 1 \\ 2 \end{array}\right| x\right)\,,$$
$$\mathrm{SIN} = x \cdot {}_0\mathrm{F}_1\left(3/2\left|-\frac{x^2}{4}\right.\right)\,, \qquad \mathrm{COS} = {}_0\mathrm{F}_1\left(1/2\left|-\frac{x^2}{4}\right.\right)\,,$$
$$\mathrm{SINH} = x \cdot {}_0\mathrm{F}_1\left(3/2\left|\frac{x^2}{4}\right.\right)\,, \qquad \mathrm{COSH} = {}_0\mathrm{F}_1\left(1/2\left|\frac{x^2}{4}\right.\right)\,,$$
$$\mathrm{ARCSIN} = x \cdot {}_2\mathrm{F}_1\left(\left.\begin{array}{c} 1/2\ \ 1/2 \\ 3/2 \end{array}\right| x^2\right)\,, \quad \mathrm{ARCTAN} = x \cdot {}_2\mathrm{F}_1\left(\left.\begin{array}{c} 1/2\ \ 1 \\ 3/2 \end{array}\right| -x^2\right)\,,$$
$$\mathrm{ARSINH} = x \cdot {}_2\mathrm{F}_1\left(\left.\begin{array}{c} 1/2\ \ 1/2 \\ 3/2 \end{array}\right| -x^2\right)\,, \quad \mathrm{ARTANH} = x \cdot {}_2\mathrm{F}_1\left(\left.\begin{array}{c} 1/2\ \ 1 \\ 3/2 \end{array}\right| x^2\right)\,,$$

all can be represented by hypergeometric series.

Note that a function of the form $f(x^m)$ and so an FPS of the form $F(x^m)$ is called *m-fold symmetric*. Even functions are 2-fold symmetric and odd functions are shifted 2-fold symmetric functions.

By the above examples one is led to the following more general definition. First we extend the considerations to *formal Laurent series* (FLS) with a representation

$$F := \sum_{k=k_0}^{\infty} a_k x^k \qquad (a_{k_0} \neq 0) \tag{2.6}$$

for some $k_0 \in \mathbb{Z}$. FLS are shifted FPS, and they correspond to meromorphic f with a pole of order $-k_0$ at the origin.

Definition 2.1. (Functions of hypergeometric type) *An FLS F with representation* (2.6) *— as well as its corresponding function f — is called to be* of hypergeometric type *if it has a positive radius of convergence, and if its coefficients a_k satisfy a RE of the form*

$$\begin{array}{rcll} a_{k+m} &=& R(k)\, a_k & \text{for } k \geq k_0 \\ a_k &=& A_k & \text{for } k = k_0, k_0+1, \ldots, k_0+m-1 \end{array} \tag{2.7}$$

for some $m \in \mathbb{N}$, $A_k \in \mathbb{C}$ $(k = k_0+1, k_0+2, \ldots, k_0+m-1)$, $A_{k_0} \in \mathbb{C} \setminus \{0\}$, and some rational function R. The number m is then called the symmetry number *of (the given representation of) F. A RE of type* (2.7) *is also called to be of hypergeometric type.*

We want to emphasize that the above terminology of functions of hypergeometric type is slightly more general than the terminology of generalized hypergeometric functions. The function $\sin x$ e. g. is **not** a generalized hypergeometric function as obviously no RE of the type (2.7) holds for its series coefficients with $m = 1$. So $\sin x$ is not of hypergeometric type with symmetry number 1; it is however of hypergeometric type with symmetry number 2. A more difficult example of this kind is the function $e^{\arcsin x}$ which is neither even nor odd, and nevertheless turns out to be of hypergeometric type with symmetry number 2, too (see (9.3)).

Note that a function f may be of hypergeometric type (with respect to the origin) whereas it is not of hypergeometric type with respect to another point $x_0 \neq 0$, i. e. $f(x - x_0)$ may not be of hypergeometric type. An example for this situation is the function $\frac{\sin x}{x}$ which is of hypergeometric type; the function $\frac{\sin(x-x_0)}{x-x_0}$, however, is not of hypergeometric type for $x_0 \neq 0$.

If f is m-fold symmetric, then there is a hypergeometric representation (2.7) with symmetry number m, whereas such a representation does not guarantee any symmetry. In fact, if f is of hypergeometric type with symmetry number j, then it is of hypergeometric type with each multiple mj ($m \in \mathbb{N}$) of m as symmetry number since by induction we get the RE

$$a_{k+jm} = R(k)\, R(k+j)\, R(k+2j) \cdots R(k+(m-1)j)\, a_k \ ,$$

and $R(k)\, R(k+j)\, R(k+2j) \cdots R(k+(m-1)j)$ is rational, too. In particular, each hypergeometric type function with symmetry number $j=1$ is of hypergeometric type for arbitrary symmetry number m.

On the other hand, it is clear that each FLS with symmetry number m can be represented as the sum of m shifted m-fold symmetric functions.

We extend the definition of the derivative to FLS by the rule

$$F' := \sum_{k=k_0}^{\infty} k a_k x^{k-1} = \sum_{k=k_0-1}^{\infty} (k+1) a_{k+1} x^k \ . \tag{2.8}$$

Now we give a list of transformations on FLS that preserve the hypergeometric type.

Lemma 2.1. *Let F be an FLS of hypergeometric type with representation* (2.6). *Then*

(a) xF , (b) F/x , (c) $F(Ax)$ $(A \in \mathbb{C})$,

(d) $F(x^n)$ $(n \in \mathbb{N})$, (e) $\dfrac{F(x) \pm F(-x)}{2}$, (f) F' ,

are of hypergeometric type, too. If F has symmetry number m, then $F(x^n)$ has symmetry number nm, and $\frac{F(x) \pm F(-x)}{2}$ has symmetry number $2m$. If F is an FPS, then so is

(g) $\int F$.

Proof. Suppose, for the coefficients of F the RE (2.7) holds.
(a) The operations (a) and (b) are shift operations. xF has the coefficients $b_k := a_{k-1}$, and so the RE $b_{k+m} = a_{(k-1)+m} = R(k-1)\, a_{k-1} = R(k-1)\, b_k$ holds. As R is rational in the variable k, so is $R(k-1)$, and xF is of hypergeometric type.
(b) Similarly we have here $b_{k+m} = R(k+1)\, b_k$.
(c) $F(Ax)$ has the coefficients $b_k := A^k\, a_k$, and so the RE

$$b_{k+m} = A^{k+m}\, a_{k+m} = R(k)\, A^{k+m}\ a_k = R(k)\, A^m b_k$$

holds.
(d) The coefficients b_k of $F(x^n)$ satisfy $b_{nk} := a_k$, and so we have

$$b_{nk+nm} = b_{n(k+m)} = a_{k+m} = R(k)\, a_k = R(k)\, b_{nk} \ ,$$

and finally $b_{k+nm} = R(k/n)\, b_k$.
(e) The FLS $G := \frac{F(x)+F(-x)}{2}$ represents the even part of F. Thus its odd coefficients vanish whereas its even coefficients b_k agree with those of F, i. e. $b_k = a_k$. Therefore for an even index k we have

$$b_{k+2m} = a_{k+2m} = R(k+m)\, a_{k+m} = R(k+m)\, R(k)\, a_k = R(k+m)\, R(k)\, b_k \ .$$

This formula obviously holds for the odd coefficients, too, and so G is of hypergeometric type with symmetry number $2m$. As the FLS $\frac{F(x)-F(-x)}{2}$ represents the odd part of F, a similar argument can be applied in that case.

(f) By its definition (2.8) F' has coefficients $b_k := (k+1)\, a_{k+1}$, and so the hypergeometric type RE

$$b_{k+m} = (k+1+m)\, a_{k+1+m} = (k+1+m) R(k+1)\, a_{k+1} = \frac{k+1+m}{k+1} R(k+1)\, b_k$$

holds.
(g) Here we have $b_k := \frac{a_{k-1}}{k}$, and so $b_{k+m} = \frac{k}{k+m} R(k-1)\, b_k$. $\square$

We note that as $\cos x = \frac{e^{ix}+e^{-ix}}{2}$ and $\sin x = \frac{e^{ix}-e^{-ix}}{2i}$ a combination of (c) and (e) shows that the hypergeometric type of $\cos x$ and $\sin x$ follows from that of the exponential function.

We remark further that in § 8 we will extend the definition of functions of hypergeometric type to include also the functions defined in (g) for arbitrary FLS, and to be able to reverse the process of (d). Note that these functions, in general, do not represent FLS.

It is essential for our development that functions of hypergeometric type satisfy a simple DE.

Theorem 2.1. *Each FLS of hypergeometric type satisfies a simple DE.*

Proof. Let F be given by

$$F := \sum_{k=k_0}^{\infty} a_k x^k \qquad (a_{k_0} \neq 0)\ .$$

Next, we mention a nice property of the differential operator θ. As

$$\theta F = \sum_{k=k_0}^{\infty} k a_k x^k\ ,$$

it follows by induction that for all $j \in \mathbb{N}$

$$\theta^j F = \sum_{k=k_0}^{\infty} k^j a_k x^k\ .$$

This shows that moreover, if P is any polynomial, we may formally write

$$P(\theta) F = \sum_{k=k_0}^{\infty} P(k)\, a_k x^k\ . \tag{2.9}$$

This commuting property is the reason why the differential operator θ is much more appropriate for the current discussion than the usual differential operator $\frac{d}{dx}$.

Now, from RE (2.7) we get for $k \in \mathbb{N}$

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

where P, and Q are polynomials. We write this as

$$Q(k)\, a_{k+m} = P(k)\, a_k\ , \tag{2.10}$$

and we assume without loss of generality that the polynomials P and Q are chosen such that $Q(k_0-1) = Q(k_0-2) = \cdots = Q(k_0-m) = 0$. This goal can be reached by multiplying both P and Q with the factors $(k-k_0+j)$ $(j=1,\ldots,m)$. From (2.9) and (2.10), it is easy to derive a DE for F; we get

$$\begin{aligned}
Q(\theta-m)\,F &= \sum_{k=k_0}^{\infty} Q(k-m)\,a_k x^k \qquad \text{by (2.9), as } Q \text{ is a polynomial} \\
&= \sum_{k=k_0+m}^{\infty} Q(k-m)\,a_k x^k \quad \text{as } Q(k_0-1)=Q(k_0-2)=\cdots=Q(k_0-m)=0 \\
&= \sum_{k=k_0}^{\infty} Q(k)\,a_{k+m} x^{k+m} \qquad \text{by an index shift} \\
&= x^m \sum_{k=k_0}^{\infty} P(k)\,a_k x^k \qquad \text{by (2.10)} \\
&= x^m P(\theta) F \qquad \text{by (2.9) again.}
\end{aligned}$$

This represents a DE for F which turns out to be of form (2.5). Note that for $m=1$ and $k_0=0$ we have exactly (2.4). □

Now assume, a function f representing an FLS is given. In order to find the coefficient formula, it is a reasonable approach to search for its DE, to transfer this DE into its equivalent RE, and you are done by an adaption of the coefficient formula for the hypergeometric function corresponding to the transformations given in the proof of Lemma 2.1. In § 3 we give the details of this algorithm, and in § 4–7 we present finite algebraic algorithms for its particular steps.

3. The First Conversion Procedure

There are two obvious transformation procedures: $f \mapsto F$ and $F \mapsto f$. At the moment we want to emphasize on the first situation. This transformation procedure $f \mapsto$ `powerseries(f,x,x0)` is implemented in Macsyma (see Macsyma Reference Manual). The implementation in Macsyma is not based on an algorithm but uses a chain of certain steps which give the result in some cases. Macsyma's procedure is at follows:

1 Macsyma tries to expand f in the variable $x-x_0$, e. g. using addition theorems,
2 logarithms $\ln f$ are handled by the rule $\ln f = \displaystyle\int \frac{f'}{f}$,
3 for rational functions a (real) partial fraction decomposition (PFD) is used,
4 the power series expansions of the standard elementary functions with point of development $x_0=0$ are implemented.

This procedure has the following disadvantages: it fails

1 to find the result for all rational functions, e. g. for $f := \frac{1}{x^2+2x+2}$, because of the use of a *real* PFD,
2 to find the result for $f := \frac{1}{x^2-2x-2}$ e. g., as the partial fraction implementation fails to find nonrational roots of the denominator,
3 to solve `powerseries(exp(x)*exp(y),x,0)` as the internal simplifier changes the input into e^{x+y} before proceeding,

4 to get `powerseries(atan(x+a),x,0)` or `powerseries(atan(x),x,b)` because of the lack of an addition formula for the inverse tangent function,
5 to solve the problem for products correctly. Usually (e. g. for $f := e^x \sin x$) a product of power series is returned instead of the power series of the product as requested.

Here we present an algorithm corresponding to a function call `PowerSeries[f,x,0]` — we use MATHEMATICA syntax as our implementation is written in MATHEMATICA language — i. e. producing a Laurent series expansion of the function f with respect to the variable x at the point of development $x_0 = 0$. For our algorithmic procedure the latter is no real restriction and will be eliminated later.

ALGORITHM 3.1. (for `PowerSeries[f,x,0]`) .

(1) **Rational functions** (see § 4)
If f is rational in x, then use the **rational algorithm** of § 4.
(2) **Find a simple DE** (for details, see § 5)
 (a) Fix a number $N_{\max} \in \mathbb{N}$, the maximal order of the DE searched for; a suitable value is $N_{\max} := 4$.
 (b) Set $N := 1$.
 (c) Calculate $f^{(N)}$; **either**, if the derivative $f^{(N)}$ is rational, apply the **rational algorithm** of § 4, and integrate;
 (d) **or** find a simple DE for f of order N

$$\sum_{k=0}^{N} p_k f^{(k)} = 0 \tag{3.1}$$

 where p_k $(k = 1, \ldots, N)$ are polynomials in the variable x.
 (e) If (c) was not successful, then increase N by one, and go back to (c), until $N = N_{\max}$.
(3) **Find the corresponding RE** (see § 6)
Suppose you found a simple DE in step (2), then transfer it into a RE for the coefficients a_n. The RE is then of the special type

$$a_{n+1} = \sum_{k=0}^{M} r_k a_{n-k} \ , \tag{3.2}$$

where r_k $(k = 0, \ldots, M)$ are rational functions in n, and $M \in \mathbb{N}$. This is done by the substitution

$$x^j f^{(k)} \mapsto (n{+}1{-}j)_k \cdot a_{n+k-j} \tag{3.3}$$

into the DE.
(4) **Type of RE** (see § 7)
Determine the type of the RE according to the following list
 (a) If the RE (3.2) contains only one summand $r_j a_{n-j}$ on its right hand side, then f is of hypergeometric type; it has symmetry number $m = j + 1$, and an explicit formula for the coefficients can be found by the hypergeometric coefficient formula (2.2), and some initial conditions.

(b) If the DE has constant coefficients ($c_k \in \mathbb{C}$ ($k = 0, \ldots, Q$))

$$\sum_{k=0}^{Q} c_k f^{(k)} = 0 \ ,$$

then f is of *exp-like type*. In this case the substitution $b_n := n! \cdot a_n$ leads to the RE

$$\sum_{k=0}^{Q} c_k b_{n+k} = 0 \ , \tag{3.4}$$

which has the same constant coefficients as the DE, and can be solved by a known algebraic scheme using the first Q initial coefficients.

(c) If the RE is none of the above types, try to solve it by other known RE solvers (a few are implemented in the MATHEMATICA package `DiscreteMath‘RSolve‘`, e. g.).

We shall present the details of the single parts of the algorithm in the next sections. Here we prefer to give some examples for the use of the algorithm.

1 Suppose $f(x) = e^x \sin x$, so $f'(x) = e^x \Big(\sin x + \cos x\Big)$ and $f''(x) = 2e^x \cos x$. The first step of the algorithm does not apply. In the second step for $N := 1$ the expression $A_0 = -\Big(1 + \cot x\Big)$ is not rational in x. For $N := 2$ we get the expression

$$f'' + A_1 f' + A_0 f = 2e^x \cos\ x + A_1 e^x \Big(\sin\ x + \cos\ x\Big) + A_0 e^x \sin\ x.$$

Under the summands

$$2e^x \cos x \ , \quad A_1 e^x \sin x \ , \quad A_1 e^x \cos x \ , \quad A_0 e^x \sin x$$

there are exactly the two rationally independent terms $e^x \cos x$ and $e^x \sin x$. We set the coefficient sums of these expressions to zero. The linear equations system

$$2 + A_1 = 0 \ , \qquad A_1 + A_0 = 0$$

has the solution $A_1 = -2$, $A_0 = 2$, and leads so to the DE for f

$$f'' - 2f' + 2f = 0 \ . \tag{3.5}$$

This DE has constant coefficients, so f is of exp-like type. For $b_n := n!\, a_n$ we have the RE

$$b_{n+2} - 2b_{n+1} + 2b_n = 0 \ .$$

The initial conditions are

$$b_0 = a_0 = f(0) = 0 \ , \qquad \text{and} \qquad b_1 = a_1 = \frac{f'(0)}{1!} = 1 \ .$$

For a RE with constant coefficients the setup $b_n := \lambda^n$ leads to a solution. Possible values for λ are solutions of the equation $\lambda^{n+2} - 2\lambda^{n+1} + 2\lambda^n = 0$, or solutions of the equivalent *characteristic equation*

$$\lambda^2 - 2\lambda + 2 = 0 \ ,$$

and so the values $\lambda_{1,2} := 1 \pm i$. Superposition (the RE is linear) leads to the general solution $b_n = A\lambda_1^n + B\lambda_2^n$ with constants $A, B \in \mathbb{C}$. The initial conditions lead then to the linear equations system for A and B

$$0 = b_0 = A + B \;, \qquad 1 = b_1 = A(1+i) + B(1-i) \;,$$

whose solution is $A = \frac{1}{2i}, B = -\frac{1}{2i}$. So we have finally

$$a_n = \frac{b_n}{n!} = \frac{1}{n!}\frac{(1+i)^n - (1-i)^n}{2i} = \frac{1}{n!}\mathrm{Im}\,(1+i)^n = \frac{1}{n!}\mathrm{Im}\,\left(\sqrt{2}e^{i\frac{\pi}{4}}\right)^n = \frac{1}{n!}2^{\frac{n}{2}}\sin\frac{n\pi}{4},$$

and

$$F = \sum_{n=0}^{\infty} \frac{1}{n!}2^{\frac{n}{2}}\sin\frac{n\pi}{4}\,x^n \;.$$

2 Suppose next $f(x) = \arcsin x$, so $f'(x) = \frac{1}{\sqrt{1-x^2}}$ and $f''(x) = \frac{x}{\sqrt{1-x^2}^3}$. The second step of the algorithm leads to the DE for f

$$(1-x^2)f'' - xf' = 0 \;, \tag{3.6}$$

and the transformation via the rule (3.3) generates then the RE

$$(n+2)(n+1)a_{n+2} - n^2 a_n = 0 \;,$$

which is of hypergeometric type as only two summands occur. Moreover one sees immediately that the symmetry number m equals 2. The fact that $f(0) = 0$ tells us that f is odd. So it will be convenient to work with the FPS

$$H(x) := \sum_{k=0}^{\infty} c_k x^k$$

for which $f(x) = xh(x^2)$, and so $c_k = a_{2k+1}$. The substitution $n \mapsto 2k+1$ then leads to the RE for c_k

$$c_{k+1} = \frac{(k+\frac{1}{2})^2}{(k+\frac{3}{2})(k+1)}c_k \;.$$

The initial condition is $c_0 = a_1 = f'(0) = 1$, so that finally by (2.2)

$$c_k = \frac{\left(\frac{1}{2}\right)_k \cdot \left(\frac{1}{2}\right)_k}{\left(\frac{3}{2}\right)_k k!} = \frac{(2k)!}{(2k+1)4^k(k!)^2} \;,$$

and

$$F = \sum_{k=0}^{\infty} \frac{(2k)!}{(2k+1)4^k(k!)^2}\,x^{2k+1} \;.$$

3 Now we look at the error function

$$f(x) = \mathrm{erf}\,(x) := \frac{2}{\sqrt{\pi}}\int\limits_0^x e^{-t^2}\,dt \;.$$

Here $f'(x) = \frac{2}{\sqrt{\pi}}e^{-x^2}$ and $f''(x) = -\frac{4}{\sqrt{\pi}}xe^{-x^2}$. The algorithm leads to the DE

$$f'' + 2xf' = 0 \;, \tag{3.7}$$

and so to the RE

$$(n+2)(n+1)a_{n+2} + 2na_n = 0 \;.$$

This is also of hypergeometric type, and the same argumentation as in Example 2 shows that f is odd. We use the same substitutions and get for the coefficients c_k of H

$$c_{k+1} = -\frac{k+\frac{1}{2}}{(k+\frac{3}{2})(k+1)} c_k$$

with initial condition $c_0 = a_1 = f'(0) = \frac{2}{\sqrt{\pi}}$, so that finally

$$c_k = \frac{2}{\sqrt{\pi}} \frac{(-1)^k \left(\frac{1}{2}\right)_k}{\left(\frac{3}{2}\right)_k k!} = \frac{2}{\sqrt{\pi}} \frac{(-1)^k}{(2k+1)k!} \;,$$

and

$$F = \sum_{k=0}^{\infty} \frac{2}{\sqrt{\pi}} \frac{(-1)^k}{(2k+1)k!} \, x^{2k+1} \;.$$

4 Let's now look at $f(x) = \arctan x$. Here $f'(x) = \frac{1}{1+x^2}$ is rational. So we apply the rational algorithm (see § 4), and integrate. We find the complex PFD

$$f'(x) = \frac{1}{1+x^2} = \frac{1}{2}\left(\frac{1}{1+ix} + \frac{1}{1-ix}\right) \;,$$

from which we deduce for the coefficients b_n of the derivative $F'(x) = \sum\limits_{n=0}^{\infty} b_n x^n$ by using the binomial series that

$$b_n = \frac{1}{2}\left(i^n + (-i)^n\right) = \frac{i^n}{2}\left(1 + (-1)^n\right) \;.$$

By the calculation

$$b_{2k+1} = \frac{i^{2k+1}}{2}\left(1 + (-1)^{2k+1}\right) = 0$$

it follows that F' turns out to be even. Moreover

$$b_{2k} = \frac{i^{2k}}{2}\left(1 + (-1)^{2k}\right) = (-1)^k \;,$$

so that by integration

$$F = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} \, x^{2k+1} \;.$$

We remark that the arctan function can also be handled by the hypergeometric procedure similarly as Example 2.

5 Finally we look at a rational example. Let $f(x) = \frac{1}{x^2+2x+2}$. We apply the rational algorithm, and find the complex PFD

$$f(x) = \frac{1}{x^2+2x+2} = \frac{i}{2}\left(\frac{1}{x+1-i} + \frac{1}{x+1+i}\right) = -\frac{i}{2}\cdot\frac{1}{1-i}\cdot\frac{1}{1+\frac{x}{1-i}} + \frac{i}{2}\cdot\frac{1}{1+i}\cdot\frac{1}{1+\frac{x}{1+i}} \;,$$

so that for the coefficients a_n of the corresponding FPS F it follows

$$a_n \;=\; \frac{i}{2}\left(\frac{-1}{1-i}\right)^{n+1} - \frac{i}{2}\left(\frac{-1}{1+i}\right)^{n+1}$$

$$= \frac{(-1)^n}{2i} \left(\left(\frac{e^{i\frac{\pi}{4}}}{\sqrt{2}} \right)^{n+1} - \left(\frac{e^{-i\frac{\pi}{4}}}{\sqrt{2}} \right)^{n+1} \right) = \frac{(-1)^n}{\sqrt{2}^{n+1}} \sin \left((n+1)\frac{\pi}{4} \right),$$

and

$$F = \sum_{k=0}^{\infty} \frac{(-1)^k}{\sqrt{2}^{k+1}} \sin \left((k+1)\frac{\pi}{4} \right) x^k .$$

Finally, we remark that if one is interested in the Laurent series at another point of development than zero, the procedure (in MATHEMATICA code)

```
PowerSeries[f_,x_,x0_] := Module[
   {y},
   PowerSeries[f/.x->y+x0,y,0] /. y->x-x0
]
```

which establishes two simple substitutions, gives then the Laurent series expansion at the point x_0.

So it turns out that the success of our algorithm does not depend on the simplification facilities which are available. However we note again that a function may be of hypergeometric type at one point of development, and fails to have this property at another point.

4. The Rational Algorithm

The rational algorithm is as follows.

ALGORITHM 4.1. (THE RATIONAL ALGORITHM) .

1 Calculate a **complex** PFD of f

$$f(x) = \sum_{k=1}^{n} \left(\frac{c_{k1}}{x - z_k} + \frac{c_{k2}}{(x - z_k)^2} + \cdots + \frac{c_{kp_k}}{(x - z_k)^{p_k}} \right)$$

(see e. g. Walter (1985), p. 171) which can be algorithmically done at least if the denominator has a rational factorization.

2 By this procedure f is the sum of expressions of the type

$$\frac{c_{kj}}{(x - z_k)^j} = \frac{(-1)^j c_{kj}}{z_k^j \left(1 - \frac{x}{z_k}\right)^j}$$

which can be expanded by the binomial series, and have the coefficients

$$(\alpha_n)_{kj} = \frac{(-1)^j c_{kj}}{z_k^{j+n}} \binom{j+n-1}{n} .$$

Note that exactly the rational functions correspond to a RE of type (3.2) with *constant coefficients* r_k $(k = 0, \ldots, M)$ (see e. g. Walter (1985), p. 163 – 164).

5. Simple Differential Equations

Suppose, a function f is been given, and we search for a DE for f of the form (2.5), i. e. a simple DE. First of all we note that, if there exists one simple DE, then there exists a second simple DE for f that is essentially different from the first one in the sense that the corresponding RE's are different (see § 6). E. g. for the exponential function $f(x) := e^x$ we have the simple DE

$$f' = f \; , \tag{5.1}$$

corresponding to the RE $a_{n+1} = \frac{1}{n+1} a_n$. Differentiation shows that moreover $f'' = f'$ (which is not essentially different as it leads to the same RE for the series coefficients), and so by combination we get $f'' + 2f' = f' + 2f$, or equivalently

$$f'' + f' = 2f \; , \tag{5.2}$$

which is a DE essentially different from (5.1) as it is equivalent to the RE

$$(n+2)(n+1)a_{n+2} = -(n+1)a_{n+1} + 2a_n$$

which is not of hypergeometric type. We present here an algorithm that finds one simple DE for f whenever one exists. The order of the DE found is minimal with respect to all simple DE's satisfied by f.

Algorithm 5.1. (Search for a simple DE) .

(a) Find out whether there exists a simple DE for f of order $N := 1$. Therefore differentiate f, and solve the linear equation

$$f'(x) + A_0 f(x) = 0$$

for A_0; i. e. set $A_0 := -\frac{f'(x)}{f(x)}$. Is A_0 rational in x, then you are done after multiplication with its denominator.

(b) Increase the order N of the DE searched for by one. Decompose the expression

$$f^{(N)}(x) + A_{N-1} f^{(N-1)}(x) + \cdots + A_0 f(x)$$

in elementary summands (with respect to the constants $A_0, A_1, \ldots, A_{N-1}$). Test, if the summands contain exactly N rationally independent expressions considering the numbers $A_0, A_1, \ldots, A_{N-1}$ as constants. Just in that case there exists a solution as follows: sort with respect to the rationally independent terms and create a system of linear equations by setting their coefficients to zero. Solve this system for the numbers $A_0, A_1, \ldots, A_{N-1}$. Those are rational functions in x, and there exists a unique solution. After multiplication by the common denominator of $A_0, A_1, \ldots, A_{N-1}$ you get the DE searched for. Finally cancel common factors of the polynomial coefficients.

(c) If part (b) was not successful, repeat step (b).

Proof. We will have to prove, that if we are at step N of the algorithm searching for a simple DE

$$f^{(N)} + A_{N-1} f^{(N-1)} + \cdots + A_1 f' + A_0 f = 0 \tag{5.3}$$

for f (i. e. for all $n < N$ the corresponding DE did *not* have exactly n rationally independent summands), then either

(a) the number of rationally independent summands of (5.3) equals N,

and

(b) the linear equations system that we get by setting the coefficients of the rationally independent terms to zero, has a unique solution $A_0, A_1, \ldots, A_{N-1}$,

or

(c) the number of rationally independent summands of (5.3) is larger than N, and there is no solution.

Let f be given. We start with $N := 1$. Then the DE (5.3) contains at least one rationally independent term, namely f, as we excluded the rational functions from this procedure. If the number of rationally independent terms equals 1 (and so f' is a rational multiple of f), then we get a unique solution $A_0 := -f'/f$. If not, then the number of rationally independent terms is at least 2, and we must proceed with $N := 2$.

In that case, suppose now the number or linearly independent terms is less than or equal to 2. Then we are able to find a solution vector (A_0, A_1), and it remains to show that the solution is unique. Assume both

$$f'' + A_1 f' + A_0 f = 0 \qquad \text{and} \qquad f'' + a_1 f' + a_0 f = 0$$

are valid for f, then their difference

$$(A_1 - a_1)f' + (A_0 - a_0)f = 0 \tag{5.4}$$

is also a valid DE for f. Observe that (5.4) is also simple and has order 1. But by the above step we know that there is no simple DE for f of order 1, so that it must follow that all coefficients of (5.4) equal zero, i. e. $a_k = A_k$ $(k = 1, 2)$, and so the solution was unique. In the other case, the number of rationally independent terms is larger than N, and we proceed with $N := 3$. The proof for general N is then by applying our argument inductively. □

We mention that one cannot guarantee that for functions f of hypergeometric type the algorithm produces a DE corresponding to a RE of hypergeometric type. Indeed, for the function $f(x) := e^x \sin x$ the algorithm produces the DE $f'' - 2f' + 2f = 0$; transferring this DE into its equivalent RE (see § 6) leads to

$$(n+2)(n+1)a_{n+2} = 2(n+1)a_{n+1} - 2a_n$$

which is *not* of hypergeometric type. However f also fulfills the fourth order DE $f^{(iv)} + 4f = 0$ corresponding to the hypergeometric RE

$$(n+4)(n+3)(n+2)(n+1)a_{n+4} = -4a_n \ .$$

On the other hand, this seems to be a rare situation. In all but very few cases hypergeometric type FLS are recognized by the given algorithm.

6. Getting the Recurrence Equation

The underlying method to find the RE from a simple DE was presented in § 3. We have to prove the statements given there. So assume, an FLS F with a representation

$$F = \sum_{n=k_0}^{\infty} a_n x^n$$

satisfies the DE

$$\sum_{j=0}^{Q} \sum_{k=0}^{Q} c_{jk} x^j F^{(k)}(x) = 0 \, . \tag{6.1}$$

As by induction

$$F^{(k)} = \sum_{n=k_0}^{\infty} (n+1-k)_k \, a_n x^{n-k} \, , \tag{6.2}$$

we may substitute (6.2) into (6.1) to get

$$\begin{aligned}
0 &= \sum_{j=0}^{Q} \sum_{k=0}^{Q} c_{jk} x^j \sum_{n=k_0}^{\infty} (n+1-k)_k \, a_n x^{n-k} \\
&= \sum_{j=0}^{Q} \sum_{k=0}^{Q} c_{jk} \sum_{n=k_0}^{\infty} (n+1-k)_k \, a_n x^{n-k+j} \\
&= \sum_{j=0}^{Q} \sum_{k=0}^{Q} c_{jk} \sum_{n=k_0-k+j}^{\infty} (n+1-j)_k \, a_{n+k-j} x^n \\
&= \sum_{j=0}^{Q} \sum_{k=0}^{Q} c_{jk} \left(\sum_{n=k_0-k+j}^{k_0+Q-1} (n+1-j)_k \, a_{n+k-j} + \sum_{n=k_0+Q}^{\infty} (n+1-j)_k \, a_{n+k-j} \right) x^n \\
&= \sum_{n=k_0+Q}^{\infty} \sum_{j=0}^{Q} \sum_{k=0}^{Q} c_{jk} (n+1-j)_k \, a_{n+k-j} x^n + \sum_{j=0}^{Q} \sum_{k=0}^{Q} c_{jk} \sum_{n=k_0-k+j}^{k_0+Q-1} (n+1-j)_k \, a_{n+k-j} x^n .
\end{aligned}$$

Equating coefficients leads to (3.3) for $n \geq k_0 + Q$, whereas the finite number of initial coefficients a_n $(n < k_0 + Q)$ have to be checked separately.

It remains to observe that as $(n+1-j)_k$ is a polynomial in n for each fixed j and k, the calculated RE is then of the special type (3.2) where r_k $(k = 0, \ldots, M)$ are rational functions in n, and $M \in \mathbb{N}$.

We want to present here a rule oriented MATHEMATICA procedure `subst[eq,x,a,n]` that calculates the left-hand side of the RE in the form $\sum q_k(n) a_{n-k} = 0$ from the left-hand side `eq` of the DE that is assumed to be of the formal form $\sum p_k(x) \mathrm{diff}\,[f,x,k] = 0$, using the rules above. Here diff $[f,x,k]$ is a formal representation for the expression $f^{(k)}(x)$, and it is assumed that the function involved has the name f.

```
subst[g_+h_,x_,a_,n_]:=subst[g,x,a,n]+subst[h,x,a,n]
subst[c_*g_,x_,a_,n_]:=c*subst[g,x,a,n] /; FreeQ[c,x] && FreeQ[c,f]
subst[diff[f,x_,k_],x_,a_,n_]:=Pochhammer[n+1,k]*a[n+k]
subst[diff[f,x_],x_,a_,n_]:=(n+1)*a[n+1]
subst[f,x_,a_,n_]:=a[n]
```

```
subst[x_^j_.*diff[f,x_,k_],x_,a_,n_]:=Pochhammer[n+1-j,k]*a[n+k-j]
subst[x_^j_.*diff[f,x_],x_,a_,n_]:=(n+1-j)*a[n+1-j]
subst[x_^j_.*f,x_,a_,n_]:=a[n-j]
subst[c_,x_,a_,n_]:=0 /; FreeQ[c,x]
subst[x_^j_.,x_,a_,n_]:=0 /; IntegerQ[j]
subst[x_,x_,a_,n_]:=0
subst[g_,x_,a_,n_]:=subst[Expand[g],x,a,n] /; (Head[g]==Times)
```

The function call `subst[diff[f,x]-f,x,a,n]` for the exponential function, e. g., produces the output

```
-a[n] + (1 + n) a[1 + n]
```

For the inverse sine function, we call `subst[(1-x^2)diff[f,x,2]-x*diff[f,x],x,a,n]`, and get

```
-(n a[n]) - (-1 + n) n a[n] + (1 + n) (2 + n) a[2 + n]
```

7. Solution of the Recurrence Equation

We do not want to emphasize on the solution of a RE of hypergeometric type as these details can easily be calculated with the use of the transformations presented in Lemma 2.1.

We only mention that for symmetry number $m \neq 1$ the FLS F has to be decomposed by m shifted m-fold symmetric series.

However, we consider the exp-like case. We have to prove that from the DE

$$\sum_{k=0}^{Q} c_k F^{(k)}(x) = 0 \;, \tag{7.1}$$

with constants c_k $(k = 0, \ldots, Q)$ relation (3.4) follows for the numbers $b_n := n!\, a_n$. Indeed, substituting (6.2) into (7.1), we have

$$0 = \sum_{k=0}^{Q} c_k \sum_{n=k_0}^{\infty} (n+1)_k a_{n+k} x^n = \sum_{n=k_0}^{\infty} \sum_{k=0}^{Q} c_k (n+1)_k \frac{b_{n+k}}{(n+k)!} x^n = \sum_{n=k_0}^{\infty} \frac{1}{n!} \sum_{k=0}^{Q} c_k b_{n+k} x^n \;.$$

Equating coefficients, we get (3.4).
Now we give the algebraic scheme to solve this kind of RE.

Algorithm 7.1. (Solution of a constant coefficient RE) .

(1) Substituting the setting $b_n := \lambda^n$ into the RE (3.4) produces the *characteristic equation*

$$\sum_{k=0}^{Q} c_k \lambda^k = 0 \;.$$

Solve that equation for λ, and produce the R pairwise different solutions λ_k $(k = 1, \ldots, R)$ which have multiplicities α_k $(k = 1, \ldots, R)$, respectively.

(2) The general solution has the form

$$b_n := \sum_{k=1}^{R} \sum_{j=1}^{\alpha_k} A_{kj} n^{j-1} {\lambda_k}^n \tag{7.2}$$

with the Q constants A_{kj} $(k = 1, \ldots, R\ (j = 1, \ldots, \alpha_k))$.

(3) Substituting Q initial values into (7.2) produces a linear system of Q equations for the Q unknowns A_{kj} $(k = 1, \ldots, R\ (j = 1, \ldots, \alpha_k))$. This linear system is regular. Solve this system to get a representation for b_n.

Proof. In the case that the Q solutions λ_k $(k = 1, \ldots, Q)$ all have multiplicity $\alpha_k = 1$, and so are pairwise different, the solution can be found e. g. in Walter (1985), p. 35, Exercise 5. The general case is similarly proved by an induction argument. □

We remark that it may happen that for an exp-like function f Algorithm 5.1 finds a DE which does not have constant coefficients, and so fails to recognize the exp-like type. Clearly the found DE then has lower order than the constant coefficient DE which also holds for f. This situation occurs e. g. for $f(x) := (1+x)\, e^x$ for which Algorithm 5.1 finds the simple DE $(1 + x)f' = (2 + x)f$ which has non-constant coefficients, and order one. On the other hand, f is exp-like as it is solution of the second order constant coefficient DE $f'' - 2f' + f = 0$.

8. Generalization

It turns out that for functions that do not represent an FLS, it may happen that the algorithm formally leads to a RE of hypergeometric type.

We make this a formal definition to extend the class of functions of hypergeometric type, and give three example classes of functions that are covered by this approach.

Definition 8.1. (Functions of hypergeometric type) *A function f is called to be of hypergeometric type, if the procedure of* § 5 *leads to a simple DE which by the formal application described in* § 6 *is converted into a RE of hypergeometric type* (2.7).

First we give some examples.

1 Let $f(x) := \ln x$. Then the algorithm finds the DE $f' + xf'' = 0$ for f. This DE formally is transferred into the RE $a_n = 0$ (for $n \in \mathbb{N}$) which is of hypergeometric type with rational function zero. Obviously this behaviour occurs as f' is of hypergeometric type (namely, a monomial with negative exponent).
Similarly for $f(x) := \int \ln x\, dx = -x + x \ln x$, we get the DE $f - xf' + x^2 f'' = 0$, and the formal RE $a_n = 0$ (for $n \geq 2$) corresponding to the fact that f'' equals a monomial with negative exponent.
So our extended definition above covers antiderivatives of FLS's. The antiderivative of an FLS is defined by

$$\int F := \sum_{\substack{k=k_0 \\ k\neq -1}}^{\infty} \frac{a_k}{k+1} x^{k+1} + a_{-1} \ln x = \sum_{\substack{k=k_0+1 \\ k\neq 0}}^{\infty} \frac{a_{k-1}}{k} x^k + a_{-1} \ln x\ .$$

2 Let $f(x) := e^{\sqrt{x}}$. We get the DE $-f + 2f' + 4xf'' = 0$, and the formal RE

$$2(1+2n)(1+n)a_{n+1} = a_n \ . \tag{8.1}$$

On the other hand, we know that

$$f(x) = e^{\sqrt{x}} = \sum_{k=0}^{\infty} \frac{1}{k!} x^{\frac{k}{2}} = \sum_{n=0}^{\infty} \frac{1}{(2n)!} x^n \ ,$$

where here the summation index n runs over all numbers $n = 0, \frac{1}{2}, 1, \frac{3}{2}, \ldots$. Indeed, for $a_n = \frac{1}{(2n)!}$ the RE (8.1) holds.

Similarly, for $f(x) := \dfrac{1}{1-\sqrt[3]{x}}$, we get the DE

$$2f + (-2+38x)f' + (-18x+45x^2)f'' + (-9x^2+9x^3)f''' = 0 \ ,$$

and the formal RE $a_{n+1} = a_n$, corresponding to the fact that $f(x) = g(x^{1/3})$, and g is the hypergeometric function $g(x) = \frac{1}{1-x}$.

So our new definition covers functions of the form $f(x^{1/n})$ ($n \in \mathbb{N}$) where f represents an FLS F of hypergeometric type.

3 Let $f(x) := e^{1/x}$. We get the DE $f + x^2 f' = 0$, and the corresponding RE $a_{n+1} = -n a_n$. Here the RE contains the information that $a_1 = 0$, and so $a_k = 0$ for all $k \in \mathbb{N}$. This guarantees that only coefficients with nonpositive indices occur, and so the function $g(x) := f(1/x)$ is an FPS. Indeed, from the usual exponential series we see that

$$f(x) = \sum_{k=0}^{\infty} \frac{1}{k!} x^{-k} \ .$$

We shall show now that these three cases are covered by our new approach.

THEOREM 8.1. *Let f correspond to an FLS of hypergeometric type with representation* (2.6). *Then*

(a) $\displaystyle\int f$, (b) $f(x^{1/n})$ $(n \in \mathbb{N})$, (c) $f\left(\dfrac{1}{x}\right)$,

are functions of hypergeometric type.

PROOF. Suppose, for the coefficients of F the RE (2.7) holds.
(a) First we observe that $\int f$ fulfills a simple DE as its derivative does. Further for the formal coefficients b_n of $\int f$ (i. e. ignoring the logarithmic term) we have the same situation as in Lemma 2.1.
(b) Assume F is an FLS of hypergeometric type with representation (2.6), and $g(x) := f\left(x^{1/n}\right)$. Then G has the representation

$$G := \sum_{k=k_0}^{\infty} a_k x^{k/n} = \sum_{\substack{k=(k_0+j)/n \\ j \in \mathbb{N}_0}} b_k x^k \ . \tag{8.2}$$

If the representation of F has symmetry number m, we know that there is also a representation with symmetry number nm

$$Q(k)\, a_{k+nm} = P(k)\, a_k$$

with two polynomials P, and Q. Using these polynomials and the differential operator $\theta_n := nx\frac{d}{dx}$ rather than θ the same argumentation as in Theorem 2.1 shows that G satisfies the DE

$$Q(\theta_n - nm)g = x^m\, P(\theta_n)g$$

which is simple. So G is covered by the given approach. For the coefficients b_k ($k = \frac{k_0+j}{n}$ ($j \in \mathbb{N}_0$)) of the formal series G of g by (8.2) the relation $b_k = a_{nk}$ holds so that

$$b_{k+m} = a_{nk+m} = R(nk)\, a_{nk} = R(nk)\, b_k\ ,$$

and g is of hypergeometric type. We mention that g can be reconstructed as the sum of n shifted FLS.
(c) As in (a) we observe that the existence of a simple DE for f implies the existence of a simple DE for $g(x) := f\left(\frac{1}{x}\right)$. Substituting k by $-k-m$ in the RE (2.7) gives

$$a_{-k-m} = \frac{1}{R(-k-m)} a_{-k}\ .$$

For the coefficients b_k of g we have $b_k = a_{-k}$, so that

$$b_{k+m} = a_{-k-m} = \frac{1}{R(-k-m)}\, a_{-k} = \frac{1}{R(-k-m)}\, b_k\ ,$$

and g is of hypergeometric type. □

We note that the rules of the lemma can be applied recursively without changing the hypergeometric type. In our Mathematica implementation, we covered these three cases, too.

9. Some Results

Our implementation of the given algorithm in Mathematica was tested on many arguments. We want to mention here four unexpected examples of functions of hypergeometric type

$$\left(\frac{\arcsin x}{x}\right)^2 = \sum_{k=0}^{\infty} \frac{4^k\ (k!)^2}{(1+k)(1+2k)!} x^{2k} \tag{9.1}$$

$$e^{\operatorname{arcsinh} x} = x - \sum_{k=0}^{\infty} \frac{(2k)!}{(-4)^k\ (2k-1)\ (k!)^2} x^{2k}\ , \tag{9.2}$$

$$e^{\arcsin x} = \sum_{k=0}^{\infty} \frac{4^k \prod\limits_{j=1}^{k} \left(\frac{1}{4} + (j-1)^2\right)}{(2k)!} x^{2k} + \sum_{k=0}^{\infty} \frac{4^k \prod\limits_{j=1}^{k} \left(\frac{1}{2} - j + j^2\right)}{(2k+1)!} x^{2k+1}\ , \tag{9.3}$$

and

$$\sqrt{\frac{1-\sqrt{1-x}}{x}} = \sum_{k=0}^{\infty} \frac{(4k)!}{16^k \sqrt{2}\,(2k)!\,(1+2k)!} x^k\ . \tag{9.4}$$

Similar formulae can be calculated for odd powers of $\sqrt{\frac{1-\sqrt{1-x}}{x}}$.

However, all of these examples essentially are known. Example (9.1) can be found e. g.

in Gradshteyn and Ryzhik (1980), p. 52, formula (1.645.1), or Hansen (1975), formula (5.18.4), and it is a particular case of the so-called *Clausen formula* that tells in which cases the square of a ${}_2\mathrm{F}_1$ hypergeometric function is a ${}_3\mathrm{F}_2$ hypergeometric function. This is exactly the case if the parameters are as in

$${}_3F_2\left(\begin{array}{ccc} 2a & 2b & a+b \\ a+b+1/2 & 2a+2b \end{array}\middle|\, x\right) = \left({}_2F_1\left(\begin{array}{cc} a & b \\ a+b+1/2 \end{array}\middle|\, x\right)\right)^2 .$$

In our case, we have $a = b = 1/2$, and Clausen's formula reads

$$\left(\frac{\arcsin x}{x}\right)^2 = {}_3F_2\left(\begin{array}{ccc} 1 & 1 & 1 \\ 3/2 & 2 \end{array}\middle|\, x^2\right) = \left({}_2F_1\left(\begin{array}{cc} 1/2 & 1/2 \\ 3/2 \end{array}\middle|\, x^2\right)\right)^2 .$$

The second example (9.2) follows easily from the representation

$$e^{\operatorname{arcsinh} x} = x + \sqrt{1+x^2} \tag{9.5}$$

for the inverse hyperbolic sine function. This is a representation as a binomial expression which obviously has the power series representation (9.2) (see also Hansen (1975), formula (10.49.32)). On the other hand, our algorithm found this representation independently of an explicit knowledge of formula (9.5), a fact that we will consider in more detail in § 10.

The third expression (9.3), can be found e. g. in Hansen (1975), formula (5.27.2). In Gradshteyn and Ryzhik (1980), p. 22, formula (1.216.1), however, the five initial coefficients of $e^{\arcsin x}$ are given which shows that these authors did not know a general formula for the power series coefficients of this function. Both formulas (9.2) and (9.3) are the special cases $a = 1$ and $a = i$ ($x \mapsto ix$) of the identity

$$\left(x + \sqrt{1+x^2}\right)^a = \sum_{k=0}^{\infty} \frac{2^k \cdot \left(\frac{a}{2} - \frac{k}{2} + 1\right)_k}{(1+k/a)\, k!}\, x^k$$

which holds in a neighbourhood of the origin, see e. g. Hansen (1975), formula (10.49.32), and is also found by our implementation.

Formula (9.4) can be found e. g. by the identity

$$\sqrt{\frac{1-\sqrt{1-x}}{x}} = \frac{\sqrt{1+\sqrt{x}} - \sqrt{1-\sqrt{x}}}{\sqrt{2x}}$$

which holds in a neighbourhood of the origin.

Other successful examples of our implementation are (see Hansen (1975), formula (5.10.28))

$$e^x - 2e^{-\frac{x}{2}} \cos\left(\frac{\sqrt{3}x}{2} + \frac{\pi}{3}\right) = \sum_{k=0}^{\infty} \frac{3}{(1+3k)!} x^{1+3k} ,$$

and (see Hansen (1975), formula (5.4.22))

$$\frac{1}{2} \ln\left(\frac{1+x}{1-x}\right) - \arctan x = \sum_{k=0}^{\infty} \frac{2}{3+4k} x^{3+4k}$$

We expect that by a further use of the algorithm further, perhaps new, Laurent series expansions may be found.

10. The Algorithm as a Simplifier

In some sense, the algorithm can be used as a simplifier. If, namely, the resulting FPS represents a finite sum, we have a polynomial. Often this polynomial representation may be different from the original input, in which case the corresponding equality represents an identity, and the resulting polynomial simplifies the input. In many cases, the resulting polynomial may even be the zero polynomial. As an example we consider $f(x) := \cos(4 \arccos x)$. An expert in analysis may know that this is a Chebyshev polynomial. By algebraic means, however, there is no way to discover from the given representation that f is a polynomial so that the rational algorithm does not apply. An application of our implementation yields the output

```
In[1]:= PowerSeries[Cos[4 ArcCos[x]],x,0]
ps-info: 2 step(s) for DE:
                                          2
         -16 f + x diff[f, x, 1] + (-1 + x ) diff[f, x, 2] = 0
ps-info: RE for all n >= 0:
                    (-4 + n) (4 + n) a[n]
         a[2 + n] = ---------------------
                       (1 + n) (2 + n)
ps-info: function of hypergeometric type
ps-info: a[0] = 1
ps-info: a[1] = 0
ps-info: 2-fold symmetric function
ps-info: Cos[4 ArcCos[x]] =
         Infinity       k
         ----     2 (-4)  (1 + k)!  2 k
         >        ---------------- x
         ----     (2 - k)! (2 k)!
         k = 0
              2      4
Out[1]= 1 - 8 x  + 8 x
```

Thus our algorithm is able to find a closed form for the power series coefficients of f, as f is of hypergeometric type with symmetry number 2, and to decide that f is a polynomial.

The following is a list of examples of simplifications that are done by the algorithm.

```
PowerSeries[Sin[x]^2+Cos[x]^2,x,0] -> 1
PowerSeries[Cos[ArcSin[x]]-Sqrt[1-x^2],x,0] -> 0
PowerSeries[Exp[ArcSinh[x]]-Sqrt[1+x^2],x,0] -> x
PowerSeries[ArcSin[x]+I ArcSinh[I x],x,0] -> 0
PowerSeries[ArcTan[(x+y)/(1-x y)]-ArcTan[x],x,0] -> ArcTan[y]
PowerSeries[ArcSin[x]+ArcCos[x],x,0] -> Pi/2
PowerSeries[Exp[x+y]/Exp[x],x,0] -> Exp[y]
PowerSeries[(Exp[y/x])^(x),x,0] -> Exp[y]
PowerSeries[Sin[3 ArcCos[x]]/Sqrt[1-x^2],x,0] -> -1 + 4 x^2
PowerSeries[Beta[x,1/2,2]Sqrt[x],x,0] -> 2 x - (2 x^2)/3
```

Note that these results are independent from prior knowledge, e. g. no addition theorems or trigonometrical identities are used, but our procedure leads to this knowledge. Only the knowledge about the derivatives, and implicitly the assumption (and the user's knowledge) of the existence of a Laurent series development, is used. Note further that some of the above identities do not hold for all real arguments. Indeed, it is the responsibility of the user to check the domain of applicability.

We expect that a further use of the algorithm will lead to further, perhaps new, identities.

11. The Second Conversion Procedure

The algorithm given in § 3 has a natural inverse $F \mapsto f$, that calculates the meromorphic function f from its FLS F. We omit the details of that procedure `Convert[F,x]` that converts an FLS F into its meromorphic equivalent f with respect to the variable x. Given a formula for the general coefficient a_n, the first step consists of finding a RE of the type (3.2). This step is algebraically equivalent to the search for the DE, presented in § 5.

Rather we want to emphasize on the back-substitution that produces the left-hand side of the DE from the left-hand side of the RE. This can be done, e. g., by the following MATHEMATICA procedure `backsubst[re,n,f,x]` defined by the rules

```
theta[f_,x_]:=x D[f,x]
backsubst[eq1_+eq2_,n_,f_,x_]:=backsubst[eq1,n,f,x]+backsubst[eq2,n,f,x]
backsubst[c_*eq_,n_,f_,x_]:=c*backsubst[eq,n,f,x] /;
        (FreeQ[c,n] && FreeQ[c,f] && FreeQ[c,x])
backsubst[a[n_+m_.],n_,f_,x_]:=f[x]/x^m
backsubst[n^j_.*a[n_+m_.],n_,f_,x_]:=theta[backsubst[n^(j-1)*a[n+m],n,f,x],x]
backsubst[p_*a[n_+m_.],n_,f_,x_]:=backsubst[Expand[p*a[n+m]],n,f,x] /;
        PolynomialQ[p]
```

The back-substitution of the RE for the exponential function

```
Simplify[backsubst[(n+1)*a[n+1]-a[n],n,f,x]]
```

e. g. produces the output `-f[x] + f'[x]`.

The main part of the procedure `Convert` is to solve the finally generated simple DE. At the moment, with MATHEMATICA Version 2.0, this is, at least in the case of DE's of order greater than 2, generally beyond its capabilities. On the other hand, all but very few examples that we tested were solved by MACSYMA's `ode` procedure (version 417).

References

GR Gradshteyn, I. S., Ryzhik, I. M. (1980). *Table of integrals, series, and products – corrected and enlarged version.* New York-London: Academic Press.

Han Hansen, E. R. (1975). *A table of series and products.* Englewood Cliffs, NJ: Prentice-Hall.

Hen Henrici, P. (1974). *Applied and computational complex analysis I.* Pure and applied Mathematics, New York-London-Sydney-Toronto: John Wiley and Sons.

Mac MACSYMA: Reference Manual, Version 13. Symbolics, USA.

Map MAPLE: Reference Manual, fifth edition (1988). Waterloo, Canada: Watcom publications.

Wal Walter, W. (1985). *Analysis I.* Springer Grundwissen Mathematik 3, Berlin-Heidelberg-New York-Tokyo: Springer-Verlag.

Wol Wolfram, St. (1991) MATHEMATICA. *A system for doing mathematics by Computer.* Redwood City, Calif.: Addison-Wesley Publ. Comp..