Power Series Representations of Hypergeometric Type and Non-Holonomic Functions in Computer Algebra

By

Bertrand Teguia Tabuguia

A thesis submitted in partial fulfillment for the degree of Doktor der Naturwissenschaften (Dr. rer. nat.)

in the

Department of Mathematics and Natural Sciences
University of Kassel

Ph.D. thesis supervised by:

Prof. Dr. Wolfram Koepf

Mai 2020

Tag der Disputation:

27. Mai 2020

Erstgutachter:

Prof. Dr. Wolfram Koepf
Universität Kassel

Zweitgutachter:

Prof. Dr. Werner M. Seiler
Universität Kassel

To maman Mamguem Suzanne...

Acknowledgments

I am indebted to Prof. Dr. Wolfram Koepf for his supervision of this thesis which he initiated. I have enormously benefited from his thoroughness and dynamism. His timely and relevant comments on each part of this work made it much better. I could not have imagined having a better mentor to start research studies.

I am grateful to Prof. Dr. Mama Foupouagnigni, center president of the African Institute for Mathematical Sciences (AIMS) - Cameroon, for his continuous support since the beginning of this thesis. His will of helping youths in achieving their goals is really remarkable.

I thank Prof. Dr. Marco Garuti and Prof. Dr. Werner Varnhorn for worrying about the progress of this Ph.D. work.

I acknowledge the financial support of the DAAD STIBET fellowship, the University of Kassel, AIMS-Cameroon by the tutorship opportunity, the Erasmus+ programme between AIMS-Cameroon and the University of Kassel, and individually Prof. Dr. Wolfram Koepf, Prof. Dr. Mama Foupouagnigni, and Dr. Merlin Mouafo Wouodjié.

I would like to thank all the people that I met at the Institute of Mathematics of the University of Kassel for helping me feel comfortable during my stays in Germany. In particular, I thank Dr. Merlin Mouafo Wouodjié, Dr. Daniel Tcheutia, Sebastian Rauchhaus, Dr. Matthias Seiss, Matthias Orth, Sedrick Nkotto, Prof. Dr. Hans-Georg Rück, Martina Syborg, Yves Guemo, Prof. Dr. Dorothee Knees, M.Sc. Stephanie Thomas, Philipp Hahn, Saeed Hadjizadeh, Dr. Patrick Njionou Sadjang, Christian Ngono, Hendrikje Schmidtpott-Schulz, Prof. Dr. Werner M. Seiler, Prof. Dr. Marita Thomas, Dr. Michael Oeljeklaus, Dr. Eyebe Fouda, and Prof. Dr. Maria Specovius-Neugebauer.

Special thanks to my love Pavelle Djantou who constantly cares about me.

I acknowledge the encouragements from my best friend Jonas Ngnawe, the tutors team of AIMS-Cameroon 2017-2020, my AIMS-Cameroon 2017-2018 batch, the AIMS-Cameroon 2018-2019 batch, the 2016 Computer Engineering batch of Ecole Nationale Supérieure Polytechnique de Yaoundé, Mariette Petke, and my high school friends of the group AMISCI.

A big thank-you to my siblings Tabuguia, maman Suzanne Mamguem and Achile Foudjin who are always there for me. In particular, I thank my elder brother and his wife, Mr. and Mrs. Djoko, for the warm hospitality they gave me during my stays in Yaounde.

Abstract

A Laurent-Puiseux series

$$\sum_{n=n_0}^{\infty} a_n (z-z_0)^{n/k} \; (a_n \in \mathbb{K}, k \in \mathbb{N}, n_0 \in \mathbb{Z}), \tag{1}$$

where k denotes the corresponding Puiseux number and $\mathbb{K}$ an infinite computable field [1], is mainly characterized by the general coefficient a_n. We consider the case where a_n is a term of an m-fold hypergeometric sequence. That is $a_{n+m} = r(n)a_n$, for all sufficiently large integers n, $r(n)$ is a rational function over $\mathbb{K}$, and m is a positive integer. A Laurent-Puiseux series with an m-fold hypergeometric sequence as general coefficient is said to be of hypergeometric type, with type m. We call hypergeometric type function any expression (mostly meromorphic) that can be written as a hypergeometric type series.

To find the general coefficient in (1) of a given hypergeometric type function, three key steps are to be considered [Koepf, 1992]. Given an expression f,

1. find a holonomic differential equation (DE) satisfied by f;

2. deduce a holonomic recurrence equation (RE) satisfied by the Taylor coefficients of f;

3. find all m-fold hypergeometric term solutions of the obtained RE.

Last but not least, the series representation is handled by determining the linear combination of all the resulting hypergeometric type series provided some initial values using Taylor approximation of suitable order.

The understanding of these three steps is essential for our work. In [Koepf, 1992], Koepf described the first two steps for getting holonomic recurrence equations of any given hypergeometric type function. But the third step was not complete as he considered three sub-families of hypergeometric type functions: *exp-like* functions, *rational* functions, and the functions whose recurrence equation obtained in step 2 is a *two-term* recurrence relation. In this thesis, we clearly solve the third step and develop a complete algorithm to compute power series of linear combinations of hypergeometric type functions by using a new algorithm which finds all m-fold

[1] Mostly $\mathbb{K} := \mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ is the field of rational functions in several variables

hypergeometric term solutions of holonomic recurrence equations. Also, we investigate an algorithm to represent power series of non-holonomic and non-hypergeometric type functions like $\tan(z)$, $\frac{1-\tan(z)}{1+\tan(z)}$, $\frac{z}{\exp(z)-1}$, $\frac{\arctan(z)}{1+z}$, $\exp(z^2+z)$, *etc.*

In addition, we confirm the asymptotically fast behavior of an algorithm based on holonomic recurrence equations to compute Taylor expansions of holonomic functions (see [Koepf, 2006, Chapter 10]), and present some interesting results for the automatic proof of certain identities that are generally difficult to prove (see [Koepf, 2006, Chapter 9]) like

$$\frac{1+\tan(z)}{1-\tan(z)} = \exp\left(2 \text{ arctanh}\left(\frac{\sin(2z)}{1+\cos(2z)}\right)\right)$$

by characterizing non-holonomic functions with non-linear recurrence equations and some initial values.

Our implementations are done in the computer algebra system (CAS) Maxima 5.37.2 [Schelter, 2013], and regrouped in our package FPS. The CAS Maple is also used for comparison in order to show the improvement given by our algorithms and their implementations.

Contents

Chapter 1

Introduction

The applicability of complex analysis is essentially restricted to analytic functions, since it easily allows both differentiation and integration. These functions are represented by power series with positive radius of convergence. Power series are used to represent orthogonal polynomials [Koepf and Schmersau, 1998]; in combinatorics, generating functions are power series [Stanley, 2011]; in dynamical systems, algebraic properties of power series involve most of the constructions (see [Lubin, 1994]); we can also enumerate commutative algebra and algebraic geometry ([Brewer, 2014], [Zariski and Samuel, 1960, Chapter VII]). It is therefore important to know the exact general coefficient or formula of a power series. There is no algorithm which computes the power series of any given analytic function. We classify series with a certain common property, and build an algorithm which will always find the power series representation from an analytic expression, whenever possible. It is important to notice the word "expression", because we are not considering complex functions as abstract objects defined in a certain domain and its range, but instead as a differentiable object that we can manipulate symbolically to characterize its Taylor coefficients by a certain type of linear recurrence equation. Moreover, by the unique power series characterization, this approach does not only lead to the verification of known identities, but also to the discovery of new ones.

Let $\mathbb{K}$ be a field of characteristic zero and $(a_n)_{n\in\mathbb{Z}}, a_n \in \mathbb{K}$, be an m-fold hypergeometric sequence such that

$$a_{n+m} = r(n)a_n,\ \forall n > n_0,\ n_0 \in \mathbb{Z}, \tag{1.1}$$

where $r(n)$ denotes a rational function in $\mathbb{K}(n)$, $m \in \mathbb{N}$, and n_0 is the first non-zero term index. m-fold hypergeometric sequences are very useful in summation theory ([Koepf, 2014], [Koepf and Masjed-Jamei, 2018]). Our first interest is to describe an algorithm which computes power series (Puiseux series) of the form

$$\sum_{n=n_0}^{\infty} a_n(z-z_0)^{n/k}\ (a_n \in \mathbb{K}, k \in \mathbb{N}, n_0 \in \mathbb{Z}), \tag{1.2}$$

such that a_n is an m-fold hypergeometric term.

In 1992, Koepf published an algorithmic approach for computing power series [Koepf, 1992]. The algorithm was implemented in the computer algebra systems (CAS) Maple [Heck, 2003] and Mathematica [Wolfram, 2003]. In his original approach, Koepf considered three types of functions: *two-term recurrence relation* type which corresponds to expressions leading to a linear recurrence equation equivalent to (1.1). That is

$$Q_n a_{n+m} + P_n a_n = 0, \ n \in \mathbb{Z}, \tag{1.3}$$

where Q_n, P_n are polynomials in $\mathbb{K}[n]$. The second type called *exp-like*, corresponding to expressions leading to linear recurrence equations with constant coefficients in $\mathbb{K}$. And the third type with a completely different approach based on partial fraction decomposition corresponding to rational functions in $\mathbb{K}(z)$. All gathered in the Maple and Mathematica packages FPS could already recover the power series formulas of a wide family of analytic functions.

Note that in the rational function case, the algorithm can still find a linear recurrence equation satisfied by the general coefficient sought, but the issue was in solving that equation. Furthermore, it turns out that the general coefficient found for each type used in Koepf's approach is always a linear combination of m-fold hypergeometric terms. Therefore, if we could find all m-fold hypergeometric term solutions of a linear homogeneous recurrence equation, then we could considerably increase the family of power series computed automatically.

Marko Petkovšek later published an algorithm which finds all hypergeometric ($m = 1$) term solutions of linear recurrences [Petkovšek, 1992]. This algorithm was implemented in Maple by Koepf and in Mathematica by Petkovšek. Petkovšek brilliantly used tools involved in Gosper's algorithm (see Chapter 5 in [Koepf, 2014]) in his approach. However, the complexity of Petkovšek's algorithm can be very high depending on the degree of polynomial coefficients of the equation.

In 1999, Mark van Hoeij used a different approach and got a much more efficient algorithm for the same purpose. Indeed, he considered the local behavior of solution terms, which naturally decreases the complexity by reducing the number of candidates since hypergeometric term solutions are built from some factors of the leading and the trailing polynomial coefficients [Van Hoeij, 1999]. Van Hoeij implemented his algorithm in Maple as *LREtools[hypergeomsols]*.

Note, however, that the Petkovšek and van Hoeij algorithms might only find hypergeometric term solutions in an extension field of $\mathbb{Q}$, which in certain cases, for $m > 1$, can be equivalent to m-fold hypergeometric term solutions in $\mathbb{Q}$. Indeed, the algorithm is implemented to find all hypergeometric term solutions in $\mathbb{Q}(\alpha)$, where $\alpha \in \mathbb{C} \setminus \mathbb{Q}$; since α is not always explicitly known in advance, we will often replace extension fields of $\mathbb{Q}$ by $\mathbb{C}$. But this has some disadvantages of simplicity. If we consider the power series of the cosine function at $z_0 = 0$ given by

$$\cos(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} z^{2n}, \tag{1.4}$$

then we observe that its general coefficient satisfies the recurrence equation

$$(1+n)\,(2+n)\,a_{n+2} + a_n = 0. \tag{1.5}$$

Using Koepf's algorithm, the type $m = 2$ is detected and the formula (1.4) is obtained as a linear combination of the two 2-fold hypergeometric series involved, provided the initial values $a_0 = 1$ and $a_1 = 0$.

Using van Hoeij's algorithm implemented in Maple 2018, with the same initial values, we find the hypergeometric solution

$$\frac{\mathrm{i}^n}{2\Gamma(n+1)} + \frac{(-\mathrm{i})^n}{2\Gamma(n+1)},\ \mathrm{i} \in \mathbb{C}, \mathrm{i}^2 = -1 \tag{1.6}$$

which gives the expansion

$$\cos(z) = \sum_{n=0}^{\infty} \frac{(-\mathrm{i})^n + \mathrm{i}^n}{2\Gamma(n+1)} z^n. \tag{1.7}$$

Therefore thanks to Koepf's algorithm, Maple treats the cosine case well in $\mathbb{Q}$ since the recurrence equation obtained is a two-term recurrence relation. In general, an issue occurs with unnecessary algebraic extensions of $\mathbb{Q}$ when van Hoeij's algorithm is used, because it only looks for hypergeometric term solutions. For example, any linear combination of $\cos(z)$ or $\sin(z)$ with an expression having a hypergeometric general coefficient will have a formula involving (1.6).

```
>   convert(cos(z)+exp(z),FPS);
```

$$\sum_{k=0}^{\infty} \frac{\left(1 + \frac{\mathrm{i}^k}{2} + \frac{(-\mathrm{i})^k}{2}\right) z^k}{k!}$$

```
>   convert(log(1+z)+sin(z),FPS);
```

$$\sum_{k=0}^{\infty} \left(-\frac{(-1)^{k+1}}{k+1} - \frac{\mathrm{i} \cdot \mathrm{i}^{k+1}}{2(k+1)!} + \frac{\mathrm{i} \cdot (-\mathrm{i})^{k+1}}{2(k+1)!}\right) z^{k+1}$$

Note, however, that the aim of this thesis is not to find the power series formula with a simple hypergeometric general coefficient, but to find the formula with the simplest m-fold hypergeometric general coefficients. Simple here means that the coefficients are not taken in an extension field of $\mathbb{Q}$ whenever there exists an m-fold equivalent over $\mathbb{Q}$. We should highlight m-fold hypergeometric, because up to now there is no implemented algorithm able to find such solutions of a linear recurrence equation. And it is worth to have such an algorithm since in many cases, Maple's *convert*[1] command fails to find power series of this type.

```
>   convert(arcsin(z)+cos(z),FPS);
```

$$\arcsin(z) + \cos(z)$$

```
>   convert(exp(z^2)+log(1+z^3),FPS);
```

$$e^{z^2} + \ln(1+z^3)$$

[1]Maple's convert command uses Koepf's original approach followed by an invocation of van Hoeij's algorithm.

The above Maple failures rely on the incapacity of van Hoeij's algorithm to detect m-fold ($m > 1$) hypergeometric term solutions of so called holonomic recurrence equations, that is homogeneous linear recurrence equations with polynomial coefficients. Indeed, by using the Maple package *FormalPowerSeries* we get the following holonomic recurrence equations.

```
>   RE1:=SimpleRE(arcsin(z)+cos(z),z,a(n));
```

$$\text{RE1} := -n(n^3-10n^2+21n-22)a(n)+(n-4)^2a(n-4)+(n-2)(n^3-11n^2+39n-41)a(n-2) + 2(n+1)(n+2)(n^2+4n-1)a(n+2) - 2(n+1)(n+2)(n+3)(n+4)a(n+4) = 0$$

```
>   RE2:=SimpleRE(exp(z^2)+log(1+z^3),z,a(n));
```

$$\text{RE2} := -4(n-9)^2a(n-9) + 2(n-13)(n-7)^2a(n-7) - 4(n-6)(2n-15)a(n-6) + 2(n-7)(n-5)^2a(n-5) + 2(n-4)(2n^2-28n+107)a(n-4) - 4(n-3)(n-6)a(n-3) + (n-2)(n-4)(n-17)a(n-2) + 2(n-1)(n-4)^2a(n-1) - (n-1)(n-2)(n+1)a(n+1) = 0$$

Applying van Hoeij's algorithm to these two recurrence equations yields

```
>   LREtools[hypergeomsols](RE1,a(n),{},output=basis);
```

$$\left[\frac{(-\mathrm{i})^n}{\Gamma(n+1)}, \frac{\mathrm{i}^n}{\Gamma(n+1)}\right]$$

```
>   LREtools[hypergeomsols](RE2,a(n),{},output=basis);
```

$$\left[\frac{(-1)^n}{n}, \frac{\left(\frac{1}{2}-\frac{\mathrm{i}\sqrt{3}}{2}\right)^n}{n}, \frac{\left(\frac{1}{2}+\frac{\mathrm{i}\sqrt{3}}{2}\right)^n}{n}\right]$$

which show that the general coefficients of $\arcsin(z)$ in RE1 and the one of $\exp(z^2)$ in RE2 are missed.

Although some algorithms for computing m-fold hypergeometric term solutions of holonomic recurrence equations have been described, none of them is implemented. For example, in [Cluzeau and van Hoeij, 2006] and [Van Hoeij, 1999] an algorithm using linear operators is developed, but the described approach needs non-commutative factorization for its implementation. In our approach however, non-commutative algebra is not needed. We will use a different view of holonomic recurrence equations and develop a new algorithm to detect all their m-fold hypergeometric term solutions. Thus with the Maxima implementation of this thesis, the issue with m-fold hypergeometric term solutions of holonomic recurrence equations is completely solved as the use of our Maxima package demonstrates below.

```
(%i1) RE1:FindRE(asin(z)+cos(z),z,a[n]);
```

$$(\%o1) \quad -2\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot(4+n)\cdot a_{n+4}+2\cdot(1+n)\cdot(2+n)\cdot \left(-1+4\cdot n+n^2\right)\cdot a_{n+2}-n\cdot\left(-22+21\cdot n-10\cdot n^2+n^3\right)\cdot a_n + (n-2)\cdot\left(-41+39\cdot n-11\cdot n^2+n^3\right)\cdot a_{n-2}+(n-4)^2\cdot a_{n-4}=0$$

```
(%i2) mfoldHyper(RE1,a[n]);
```

$$(\%o2) \quad \left[\left[2,\left\{\frac{(-1)^n}{(2\cdot n)!},\frac{4^n\cdot n!^2}{n^2\cdot(2\cdot n)!}\right\}\right]\right]$$

```
(%i3) RE2:FindRE(exp(z^2)+log(1+z^3),z,a[n]);
```

$$(\%o3) \quad -(n-2)\cdot(n-1)\cdot(1+n)\cdot a_{n+1}+2\cdot(n-4)^2\cdot(n-1)\cdot a_{n-1} + (n-17)\cdot(n-4)\cdot(n-2)\cdot a_{n-2}-4\cdot(n-6)\cdot(n-3)\cdot a_{n-3} + 2\cdot(n-4)\cdot\left(107-28\cdot n+2\cdot n^2\right)\cdot a_{n-4}+2\cdot(n-7)\cdot(n-5)^2\cdot a_{n-5} - 4\cdot(n-6)\cdot(2\cdot n-15)\cdot a_{n-6}+2\cdot(n-13)\cdot(n-7)^2\cdot a_{n-7} - 4\cdot(n-9)^2\cdot a_{n-9}=0$$

```
(%i4) mfoldHyper(RE2,a[n]);
```

$$(\%o4) \quad \left[\left[1,\left\{\frac{(-1)^n}{n}\right\}\right],\left[2,\left\{\frac{1}{n!}\right\}\right],\left[3,\left\{\frac{(-1)^n}{n}\right\}\right]\right]$$

```
(%i5) FPS(asin(z)+cos(z),z,n);
```

$$(\%o5) \quad \left(\sum_{n=0}^{\infty}\frac{(2\cdot n)!\cdot z^{1+2\cdot n}}{(2\cdot n+1)\cdot 4^n\cdot n!^2}\right)+\sum_{n=0}^{\infty}\frac{(-1)^n\cdot z^{2\cdot n}}{(2\cdot n)!}$$

```
(%i6) FPS(exp(z^2)+log(1+z^3),z,n);
```

$$(\%o6) \quad \left(\sum_{n=0}^{\infty}\frac{(-1)^n\cdot z^{3\cdot(1+n)}}{n+1}\right)+\sum_{n=0}^{\infty}\frac{z^{2\cdot n}}{n!}$$

Another important issue that we solve is the step which consists in deducing, when it exists, the correct linear combination of m-fold hypergeometric term solutions of a holonomic recurrence equation. Let $P_0(z), P_1(z), \ldots, P_d(z)$ be $d+1$ polynomials in $\mathbb{K}(z)$, and $f_1(z), \ldots, f_d(z)$ some analytic expressions that have m-fold hypergeometric term coefficients in their power series expansions. More generally, our algorithm handles formal series formulas of expressions of the form

$$P_0(z)+\sum_{j=1}^{d}P_j(z)f_j(z). \tag{1.8}$$

The output of such an input is of course a linear combination of hypergeometric type series, plus a polynomial which might be zero. If the correct linear combination of m-fold hypergeometric term solutions of the corresponding linear recurrence equation is not found, then the output might be missed. This happens sometimes with Maple for the hypergeometric ($m = 1$) case. For example, Maple gives

```
> convert((z+z^2+1)*exp(z)+(z^3+3)*log(1+z),FPS);
```

$$(z + z^2 + 1)e^z + (z^3 + 3)\ln(z + 1)$$

```
> convert(1+z+z^2+z^3*arctan(z),FPS);
```

$$1 + z + z^2 + z^3 \cdot \arctan(z)$$

whereas our algorithm yields correctly

```
(%i7) FPS((z+z^2+1)*exp(z)+(z^3+3)*log(1+z),z,n);
```

$$(\%o7)\quad \frac{8 \cdot z^3}{3} + z^2 + 5 \cdot z + 1 + \left(\sum_{n=0}^{\infty} - \frac{\left(-68 - 117 \cdot n - 61 \cdot n^2 - 13 \cdot n^3 - n^4 - (-1)^n \cdot (4+n)! + 2 \cdot n \cdot (-1)^n \cdot (4+n)!\right) \cdot z^{4+n}}{(n+1) \cdot (n+4) \cdot (4+n)!}\right) \tag{1.9}$$

```
(%i8) FPS(1+z+z^2+z^3*atan(z),z,n);
```

$$(\%o8)\quad \left(\sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{2 \cdot n}}{2 \cdot n - 3}\right) + z + \frac{4}{3}$$

Note that for these two latter examples van Hoeij's algorithm finds the corresponding hypergeometric terms

```
> LREtools[hypergeomsols](SimpleRE((z^2+z+1)*exp(z)+(z^3+3)
> *log(1+z),z,a(n)),a(n),{},output=basis);
```

$$\left[\frac{(-1)^n(2n-9)}{(n-3)n}, \frac{(n^2+1)}{\Gamma(n+1)}\right]$$

```
> LREtools[hypergeomsols](SimpleRE(1+z+z^2+z^3*arctan(z),
> z,a(n)),a(n),{},output=basis);
```

$$\left[\frac{i^n}{n-3}, \frac{(-1)^n}{n-3}\right]$$

but the power series terms are missed by the Maple command *convert*. We mention that this issue is not related to an argument of *convert* which has to be specified, in particular the order of the differential equations involved in the computations. Indeed the default value used for the upper bound of the differential equations sought for power series computations is 4. However, using our Maxima procedure *HolonomicDE* which also implements the same Koepf's algorithm to compute holonomic differential equations, one finds the following differential equations of order less than 4.

```
(%i9) HolonomicDE((z+z^2+1)*exp(z)+(z^3+3)*log(1+z),F(z));
```

(%o9) $$(1+z)\cdot\left(63+99\cdot z-18\cdot z^2-84\cdot z^3-9\cdot z^4+33\cdot z^5+4\cdot z^6-2\cdot z^7+z^8+z^9\right)\cdot\left(\frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)\right)-\left(36+27\cdot z-171\cdot z^2-222\cdot z^3+54\cdot z^4+147\cdot z^5+39\cdot z^6+14\cdot z^8+9\cdot z^9+z^{10}\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+\left(-162-441\cdot z-441\cdot z^2-15\cdot z^3+186\cdot z^4+42\cdot z^5-8\cdot z^6+51\cdot z^7+35\cdot z^8+5\cdot z^9\right)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)-3\cdot\left(-1-z+z^2\right)\cdot\left(-42-42\cdot z+18\cdot z^2+55\cdot z^3+52\cdot z^4+22\cdot z^5+3\cdot z^6\right)\cdot \mathrm{F}(z)=0$$

```
(%i10) HolonomicDE(1+z+z^2+z^3*atan(z),F(z));
```

(%o10) $$z\cdot\left(1+z^2\right)\cdot\left(3+2\cdot z+4\cdot z^2+2\cdot z^3\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)-2\cdot\left(3+3\cdot z+8\cdot z^2+6\cdot z^3+4\cdot z^4+3\cdot z^5\right)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)+6\cdot\left(1+z+2\cdot z^2+z^4\right)\cdot \mathrm{F}(z)=0$$

Therefore we can say that the reason why Maple's command *convert* cannot find the power series formulas of $(1+z+z^2)\exp(z)+(z^3+3)\log(z)$ and $1+z+z^2+z^3\arctan(z)$ is that the linear combinations of hypergeometric term solutions of the corresponding holonomic recurrence equations are missed.

As observed with the previous computations, our implementation is written in the CAS Maxima whose internal command *powerseries* dedicated to power series computations is rather limited. Indeed, this command is based on a pattern matching instead of algorithmic model. The syntax is *powerseries(expr,*z*,*z_0*)* that calculates the power series formula of *expr* with respect to the variable z at the point of development z_0. Below are some examples showing certain arising issues with the command *powerseries* that are solved by our implementation.

- Power series written as a square of a power series.

```
(%i11) powerseries(asin(z)^2,z,0);
```

(%o11) $$\left(\sum_{i1=0}^{\infty}\frac{\mathrm{genfact}\left(2\cdot i1-1,i1,2\right)\cdot z^{1+2\cdot i1}}{\left(2\cdot i1+1\right)\cdot \mathrm{genfact}\left(2\cdot i1,i1,2\right)}\right)^2$$

```
(%i12) FPS(asin(z)^2,z,n);
```

(%o12) $$\sum_{n=0}^{\infty}\frac{4^n\cdot n!^2\cdot z^{2+2\cdot n}}{(n+1)\cdot(1+2\cdot n)!}$$

- Non-classical power series not detected.

```
(%i13) powerseries((1-sqrt(1-4*z))/2,z,0);
```

$$(\%o13)\qquad \mathrm{powerseries}\left(\frac{1-\sqrt{1-4\cdot z}}{2},z,0\right)$$

```
(%i14) FPS((1-sqrt(1-4*z))/2,z,n);
```

$$(\%o14)\qquad \sum_{n=0}^{\infty}\frac{(2\cdot n)!\cdot z^{1+n}}{(n+1)\cdot n!^{2}}$$

```
(%i15) powerseries(asech(z),z,0);
```

$$(\%o15)\qquad \sum_{i1=0}^{\infty}\frac{z^{i1}\cdot\left(\frac{d^{i1}}{d\,z^{i1}}\cdot\operatorname{asech}(z)\Big|_{z=0}\right)}{i1!}$$

```
(%i16) FPS(asech(z),z,n);
```

$$(\%o16)\qquad \left(\sum_{n=0}^{\infty}-\frac{4^{-1-n}\cdot(1+2\cdot n)!\cdot z^{2+2\cdot n}}{(1+n)^{2}\cdot n!^{2}}\right)-\log(z)+\log(2)$$

Observe that despite the general rule used for this latter example, the output given by *powerseries* is wrong since the logarithmic term $\log(z)$ does not allow the computations of derivatives at 0.

- Power series written as multiplication of two power series.

```
(%i17) powerseries(exp(z)*cos(z),z,0);
```

$$(\%o17)\qquad \left(\sum_{i4=0}^{\infty}\frac{z^{i4}}{i4!}\right)\cdot\sum_{i4=0}^{\infty}\frac{(-1)^{i4}\cdot z^{2\cdot i4}}{(2\cdot i4)!}$$

```
(%i18) FPS(exp(z)*cos(z),z,n);
```

$$(\%o18)\qquad \left(\sum_{n=0}^{\infty}-\frac{(-1)^{n}\cdot 4^{n}\cdot z^{3+4\cdot n}}{\left(\frac{1}{4}\right)_{n}\cdot\left(\frac{3}{4}\right)_{n}\cdot(2\cdot n+1)\cdot(4\cdot n+1)\cdot(4\cdot n+3)\cdot 64^{n}\cdot(2\cdot n)!}\right)+\left(\sum_{n=0}^{\infty}\frac{(-1)^{n}\cdot 4^{n}\cdot z^{1+4\cdot n}}{\left(\frac{1}{4}\right)_{n}\cdot\left(\frac{3}{4}\right)_{n}\cdot(4\cdot n+1)\cdot 64^{n}\cdot(2\cdot n)!}\right)+\sum_{n=0}^{\infty}\frac{(-1)^{n}\cdot 4^{n}\cdot z^{4\cdot n}}{\left(\frac{1}{4}\right)_{n}\cdot\left(\frac{3}{4}\right)_{n}\cdot 64^{n}\cdot(2\cdot n)!}$$

- A bug due to the involvement of complex numbers in the expansion.

```
(%i19) powerseries(log(1+z+z^2),z,0);
```

sign: argument cannot be imaginary; found %i

– an error. To debug this try: debugmode(true);

```
(%i20) FPS(log(1+z+z^2),z,n);
```

$$(\%o20) \quad \sum_{n=0}^{\infty} -\frac{2 \cdot \cos\left(\frac{2\cdot\pi\cdot(1+n)}{3}\right) \cdot z^{1+n}}{n+1}$$

In this example the general coefficient is deduced as the real part of a hypergeometric term solution in $\mathbb{C}$ (extension field of $\mathbb{Q}$ involving i and some irrational numbers) of the corresponding linear recurrence equation.

On the other hand, some expressions like $\tan(z), \sec(z), \csc(z), etc.$ do not lead to linear recurrence equations, although they are analytic in certain domains. Therefore, we should investigate their power series computation. For that purpose, in this dissertation we consider two approaches.

Our second approach is to follow the same procedure as Koepf, but this time, instead of looking for a linear differential equation, we look for quadratic ones. For example, for the tangent function, one can find the homogeneous differential equation

$$\frac{d^2}{d\,z^2} \cdot \mathrm{F}\,(z) - 2 \cdot \mathrm{F}\,(z) \cdot \left(\frac{d}{d\,z} \cdot \mathrm{F}\,(z)\right) = 0, \tag{1.10}$$

which after the use of the Cauchy product rule, will lead to the recurrence equation

$$(1+n) \cdot (2+n) \cdot a_{n+2} - 2 \cdot \sum_{k=0}^{n} (k+1) \cdot a_{k+1} \cdot a_{n-k} = 0 \tag{1.11}$$

for the corresponding Taylor coefficients. And finally, the power series can be given by a formula depending on two initial values.

Note, however, that this approach often gives rather complicated outputs. For example in the $\sec(z)$ case, we will find the recurrence equation

$$\begin{aligned} -\sum_{k=0}^{n} \Big(\left(\left(2 - 2\cdot k^2\right) \cdot a_{k+1} + (2\cdot k+2) \cdot a_{k+1} \cdot n\right) \cdot a_{n-k+1} \\ + \left(a_k + \left(-k^2 - 3\cdot k - 2\right) \cdot a_{k+2}\right) \cdot a_{n-k}\Big) = 0 \end{aligned} \tag{1.12}$$

The best thing to do would definitely be to "solve" the recurrence equation, but despite the fact that solutions can still be unpractical for computing power series, we intend to algorithmically find simple recursive formulas for the general coefficient. Observe that the formulas

$$\tan(z) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}\, 2^{2n}\, (2^{2n}-1)\, B_{2n}\, z^{2n-1}}{(2n)!}, \tag{1.13}$$

$$\sec(z) = \sum_{n=0}^{\infty} \frac{(-1)^n E_{2n} z^{2n}}{(2n)!}, \tag{1.14}$$

are not explicit because of the unknowns B_n and E_n which represent, respectively, Bernoulli and Euler numbers. Those numbers themselves satisfy rather complicated non-holonomic recurrence equations.

In our third approach, we extend our algorithm of m-fold hypergeometric (hypergeometric type) series. Here we consider reciprocals of formal power series and build an algorithm which can compute reciprocals of power series of some analytic expressions. Using Cauchy's product rule, some other power series are also deduced.

```
(%i21) FPS(tan(z),z,n);
```

$$(\%o21)\quad \left[\sum_{n=0}^{\infty}\left(\sum_{k=0}^{n}\frac{A_k\cdot(-1)^{n-k}}{(1-2\cdot k+2\cdot n)!}\right)\cdot z^{1+2\cdot n}, A_k=\sum_{j=1}^{k}-\frac{(-1)^j\cdot A_{k-j}}{(2\cdot j)!}, A_0=1\right]$$

```
(%i22) FPS(sec(z),z,n);
```

$$(\%o22)\quad \left[\sum_{n=0}^{\infty}A_n\cdot z^{2\cdot n}, A_n=\sum_{k=1}^{n}-\frac{(-1)^k\cdot A_{n-k}}{(2\cdot k)!}, A_0=1\right]$$

Furthermore, besides our main results, there are some other interesting ones involved in this work. Indeed, we have got some improvement toward the decision making on the equality of two analytic functions in a certain neighborhood, and the importance of such a study is well-known in computer algebra [Petkovšek et al., 1996]. Using our approach based on finding quadratic differential equations to represent the power series of non-holonomic functions, we are able to automatically prove identities like

$$\frac{1+\tan(z)}{1-\tan(z)}=\exp\left(2\cdot\operatorname{arctanh}\left(\frac{\sin(2z)}{1+\cos(2z)}\right)\right),\ |z|<1, \tag{1.15}$$

which cannot be recognized without using non-trivial transformations (see [Koepf, 2006, Chapter 9]). Indeed, computing quadratic differential equations for both sides yields two compatible[2] differential equations as shown below.

```
(%i23) DE1:QDE((1+tan(z))/(1-tan(z)),F(z),Inhomogeneous);
```

$$(\%o23)\quad \frac{d}{d\,z}\cdot \mathrm{F}(z)-\mathrm{F}(z)^2-1=0$$

```
(%i24) DE2:QDE(exp(2*atanh(sin(2*z)/(1+cos(2*z)))),F(z));
```

$$(\%o24)\quad \mathrm{F}(z)\cdot\left(\frac{d^3}{d\,z^3}\cdot\mathrm{F}(z)\right)-3\cdot\left(\frac{d}{d\,z}\cdot\mathrm{F}(z)\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot\mathrm{F}(z)\right)+4\cdot\mathrm{F}(z)\cdot\left(\frac{d}{d\,z}\cdot\mathrm{F}(z)\right)=0$$

```
(%i25) CompatibleDE(DE1,DE2,F(z));
```

The two differential equations are compatible

(%o25) **true**

Moreover, our FPS algorithm simplifies the difference to zero in a neighborhood of 0.

```
(%i26) FPS((1+tan(z))/(1-tan(z))
       -exp(2*atanh(sin(2*z)/(1+cos(2*z)))),z,n);
```

(%o26) 0

[2] Two differential equations are said to be compatible if every solution of the lower order DE is solution of the other.

We have also obtained an algorithm for asymptotically fast computation of Taylor expansions of large order for holonomic functions. This is a result already observed in [Koepf, 2006], Section 10.27. We have implemented a Maxima function named *Taylor* with the same syntax as *taylor(f,z,*z_0*,d)* which computes the Taylor expansion of order d of $f(z)$. And it turns out as expected that our *Taylor* command is clearly asymptotically faster than *taylor* for holonomic functions. As an example we have:

```
(%i27) taylor(sin(z)^2,z,0,10);
```

$$(\%o27)/\mathrm{T}/z^2 - \frac{z^4}{3} + \frac{2 \cdot z^6}{45} - \frac{z^8}{315} + \frac{2 \cdot z^{10}}{14175} + \cdots$$

```
(%i28) Taylor(sin(z)^2,z,0,10);
```

$$(\%o28) \quad \frac{2 \cdot z^{10}}{14175} - \frac{z^8}{315} + \frac{2 \cdot z^6}{45} - \frac{z^4}{3} + z^2$$

that illustrates the coincidence between both outputs. Testing the efficiency for large order gives:

```
(%i29) taylor(sin(z)^2,z,0,1000)$
```

Evaluation took 15.8500 seconds (19.6100 elapsed)

```
(%i30) Taylor(sin(z)^2,z,0,1000)$
```

Evaluation took 1.8300 seconds (1.8900 elapsed)

which shows that, asymptotically, our *Taylor* command takes just about a fraction of Maxima's internal *taylor* computation timing for $\sin(z)^2$.

The next chapters are organized as follows.

In the second chapter, we give some basic notions about power series followed by the mathematics that governs the algorithmic development for the computation of power series in computer algebra.

The third chapter is devoted to some symbolic computations with Maxima, which will allow us to introduce those Maxima commands needed toward the implementation of our algorithms.

Chapter 4 describes the two first steps in Koepf's algorithm: computing holonomic differential equations and holonomic recurrence equations. In this chapter, we add some linear algebra tricks in order to gain more efficiency in the process of getting holonomic differential equations. This chapter ends with the description of our asymptotically fast algorithm for computing Taylor expansions of holonomic functions.

The fifth chapter focuses on the original Petkovšek algorithm, which is essential for the two following chapters.

In Chapter 6, we discuss van Hoeij's algorithm and present a version similar to his approach. As we will see, the efficiency obtained is the same, and moreover our algorithm gives outputs without Γ symbols, which is an advantage for the computation of power series.

Chapter 7 is devoted to our most important result, which is to present a complete algorithm to find all m-fold hypergeometric term solutions of linear recurrence equations with polynomial coefficients.

Finally, in Chapter 8, we complete Koepf's algorithm with our m-fold hypergeometric procedure. We will see in this chapter how our algorithm handles the Puiseux representation (1.2) for a given expression. We also present an extension to asymptotic expansions that is unfortunately reduced by the capabilities of Maxima in computing limits. Nevertheless, some known examples are well computed.
We will also present some algorithmic approaches that extend our algorithm to the computation of power series whose representations are close to those of hypergeometric expressions.
Furthermore, in Chapter 8 we will generalize Koepf's algorithm to quadratic differential equations in order to represent non-holonomic functions. This part is another main contribution of our work.

All the algorithms are implemented in our Maxima package FPS which is an essential part of this thesis.

Chapter 2

Power Series

Algebraically speaking, power series are a particular case of formal power series. This refers to the essence of series as a sequence of numbers in a certain field [Droste and Kuich, 2009, Semirings and formal power series, pages 3-28]. Hence the ignorance of any notion about convergence while regarding series as abstract objects that characterize a ring. This means that they can be manipulated algebraically without even existing analytically. And this is not our concern since we intend to use differentiability of expressions in a suitable field. That is the reason why the field $\mathbb{C}$ of complex numbers is chosen as the main field of our study, though most of the series expansions have their general coefficients as rational functions over extension fields of $\mathbb{Q}$. In this chapter, we present the analytic view of power series and their huge gathering in the generalized hypergeometric series for the need of automatic computing.

2.1 Power Series in Complex Analysis

This section is based on the books [Lelong-Ferrand and Arnaudiès, 1993] (see the Chapters VII-IX), [Stewart and Tall, 2018].

Definition 2.1. *A power series of the variable z (complex in general) is a series whose general term is of the form $a_n z^n$, $n \in \mathbb{N}$ where (a_n) denotes a given sequence of complex numbers. Precisely a_n is the $(n+1)^{th}$ coefficient, or coefficient of order n. The first term a_0 is usually called constant term.*

Having the definition of a power series, we can look at its convergence.

Lemma 2.1 (Abel's Lemma). *Let $z_0 \in \mathbb{C}$ such that the sequence $(a_n z_0^n)$ is bounded (which is the case when in particular the series $\sum a_n z_0^n$ is convergent). Then, for all $z \in \mathbb{C}$ such that $|z| < |z_0|$, the series $\sum a_n z^n$ is absolutely convergent; and this series is normally convergent[1] in the open disc $D(0, k|z_0|)$, $0 \leqslant k < 1$.*

[1] A series $\sum u_n(z)$ is said to be normally convergent if there exists a real series $\sum v_n$ such that $\forall n \in \mathbb{N}, \forall z \in \mathbb{C}$, $\|u_n(z)\| \leqslant v_n$

Thus we can talk about the set of values where we have the convergence for any power series which by Lemma 2.1 is a disc.

Definition 2.2 (Radius of Convergence)**.** *The radius of convergence of a power series $\sum a_n z^n$ is the supremum in $\overline{\mathbb{R}_+}$ of the set of positive real numbers r satisfying that $(a_n r^n)$ is bounded.*

Theorem 2.1. *Let R be the radius of convergence of the power series $\sum a_n z^n$, $(0 \leqslant R \leqslant \infty)$*

1. *If $R = 0$, this series converges only for $z = 0$.*

2. *If $R = \infty$, this series converges absolutely for any $z \in \mathbb{C}$. And this convergence is normal, so uniform in any bounded subset of $\mathbb{C}$.*

3. *If $0 < R < \infty$, the series is absolutely convergent for $|z| < R$, and divergent for $|z| > R$. Moreover this series converges normally (so uniformly) in any disc $\overline{D}(0, r)$, for any $r < R$.*

For $R \neq 0$, the open disc $D(0, R)$ is called disc of convergence of the series.

Proposition 2.1 (Hadamard Formula)**.** *The radius of convergence of the power series $\sum a_n z^n$ is the real number R defined by*

$$R = \frac{1}{\limsup\limits_{n \longrightarrow \infty} |a_n|^{1/n}}. \tag{2.1}$$

In practice, we often use the following D'Alembert approach.

Proposition 2.2. *Given a power series $\sum a_n z^n$ and assuming that the sequence $\left|\frac{a_{n+1}}{a_n}\right|$ is convergent, then we have the radius of convergence R verifying*

$$R = \frac{1}{\lim\limits_{n \to \infty} \left|\frac{a_{n+1}}{a_n}\right|}. \tag{2.2}$$

Example 2.1. *Whatever the polynomial $P \in \mathbb{C}[z] \setminus \{0\}$, the radius of convergence of the power series $\sum P(n) z^n$ is equal to 1 since $\frac{P(n+1)}{P(n)}$ tends to 1 when $n \to \infty$.*

It is easy to compute the derivative of a power series in its disc of convergence termwise. Moreover its derivative is also a power series.

Definition 2.3. *Let $\sum a_n z^n$ be a power series whose radius of convergence R is not 0. Then the sum $\sum_{n=0}^{\infty} a_n z^n$ is a holomorphic function (differentiable in $\mathbb{C}$) of z in its disc of convergence, and in that disc, we have*

$$f'(z) = \sum_{n=1}^{\infty} n a_n z^{n-1}. \tag{2.3}$$

Proof. A part of the proof is to show that (2.3) holds (see [Lelong-Ferrand and Arnaudiès, 1993, Chapter IV]). We assume it to be true and we show that the radii of convergence coincide. If we denote by R, R' the radius of convergence of the series $\sum a_n z^n$ and $\sum n a_n z^{n-1}$, then we have

$$R' = \frac{1}{\limsup\limits_{n \to \infty} |n a_n|^{1/n}} = \frac{1}{\limsup\limits_{n \to \infty} |a_n|^{1/n}} = R,$$

since $(n^{1/n}) \longrightarrow 1$ as $n \to \infty$. □

The sum and the product of two power series gives power series with radius of convergence at least equal to the smallest of their radii of convergence. Moreover, one can construct the ring of power series (hence the computation of the reciprocal of some power series). Other used operations for power series is the composition and the integration of power series.

The case of product of power series leads to some important formulas. Considering two power series $\sum a_n z^n$ and $\sum b_n z^n$, the resulting product has the general term

$$c_n z^n = \left(\sum_{k=0}^{n} a_k b_{n-k} \right) z^n. \tag{2.4}$$

This relation helps to compute many cases of power series. For instance, let $|z| < 1$, we know that

$$\frac{1}{1-z} = \sum_{n=0}^{\infty} z^n, \tag{2.5}$$

we can deduce

$$\begin{aligned} \frac{1}{(1-z)^2} &= \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} 1 \right) z^n = \sum_{n=0}^{\infty} (n+1) z^n = \sum_{n=0}^{\infty} \binom{n+1}{1} z^n, \\ \frac{1}{(1-z)^3} &= \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} (k+1) \right) z^n = \sum_{n=0}^{\infty} \frac{(n+1)(n+2)}{2} z^n = \sum_{n=0}^{\infty} \binom{n+2}{2} z^n. \end{aligned}$$

We can even generalize for any power $p \in \mathbb{N}$ by using Pascal's triangle. And then we obtain

$$\frac{1}{(1-z)^p} = \sum_{n=0}^{\infty} \binom{n+p-1}{p-1} z^n. \tag{2.6}$$

This formula is generalized for any real $\alpha \in \mathbb{R}$ as

$$(1+z)^\alpha = \sum_{n=0}^{\infty} \binom{\alpha}{n} z^n. \tag{2.7}$$

We will like to emphasize the above example to mention how the power series of rational functions are computed in general. A common tool used to find their expansions is the partial fraction decomposition. Indeed, since $\mathbb{C}$ is an algebraically closed field, we can always split any polynomial in the denominator and use (2.5) with initial values to compute a valid power series. Remark that this is already an algorithmic procedure for rational functions, which is also incorporated in the Maple command *convert*. Nevertheless, although $\mathbb{C}$ is an algebraically closed field, it is not generally possible to factorize polynomial expressions in Computer Algebra over $\mathbb{C}$. Moreover, even when the factorization is available, in some cases the obtained factors can really be inappropriate for computing power series. More details about this issue will be given in Chapter 7.

Definition 2.4 (Analytic Function)**.** *Let $U \subset \mathbb{C}$ [resp. $U \subset \mathbb{R}$] be an open set. A map $f\colon U \longrightarrow \mathbb{C}$ is said to be analytic in U if for any point $z_0 \in U$, the map $u \mapsto f(z_0 + u)$ can be expressed as a power series in a neighborhood of the origin in $\mathbb{C}$ [resp. $\mathbb{R}$].*

In other words, f is analytic in U if any point $z_0 \in U$ has a neighborhood in which $f(z)$ can be expressed as convergent power series of the variable $u = z - z_0$. This expansion coincides with the Taylor series of f at z_0. Moreover we can see that a power series can be identified by its coefficients as a unique analytic function defined in its disc of convergence.

Proposition 2.3. *Let U be an open set of $\mathbb{C}$ [resp. $\mathbb{R}$] and f an analytic function in U. Then f is indefinitely differentiable in U, and around any point $z_0 \in U$ the representation*

$$f(z) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(z_0)(z - z_0)^n \tag{2.8}$$

converges.

From this proposition about analytic functions, one can deduce that Taylor expansions are convergent power series. Some well known examples are the following.

Example 2.2. *(Some Power Series Expansions) Around $z_0 = 0$ we have the following convergent representations where R denotes the radius of convergence:*

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!}, \quad (R = \infty) \tag{2.9}$$

$$\cos z = \sum_{n=0}^{\infty} (-1)^n \frac{z^{2n}}{(2n)!}, \quad (R = \infty) \tag{2.10}$$

$$\sin z = \sum_{n=0}^{\infty} (-1)^n \frac{z^{2n+1}}{(2n+1)!}, \quad (R = \infty) \tag{2.11}$$

$$\cosh z = \sum_{n=0}^{\infty} \frac{z^{2n}}{(2n)!}, \quad (R = \infty) \tag{2.12}$$

$$\sinh z = \sum_{n=0}^{\infty} \frac{z^{2n+1}}{(2n+1)!}, \quad (R = \infty) \tag{2.13}$$

$$(1+z)^\alpha = \sum_{n=0}^{\infty} \frac{\alpha(\alpha-1)\cdots(\alpha-n+1)}{n!} z^n = \sum_{n=0}^{\infty} \binom{\alpha}{n} z^n, \quad (R = 1) \tag{2.14}$$

$$\ln(1+z) = \sum_{n=0}^{\infty} (-1)^n \frac{z^{n+1}}{n+1}, \quad (R = 1) \tag{2.15}$$

$$\arctan z = \sum_{n=0}^{\infty} (-1)^n \frac{z^{2n+1}}{2n+1}, \quad (R = 1) \tag{2.16}$$

$$\operatorname{arctanh} z = \sum_{n=0}^{\infty} \frac{z^{2n+1}}{2n+1}, \quad (R = 1) \tag{2.17}$$

$$\arcsin z = \sum_{n=0}^{\infty} \frac{(2n)!}{2^{2n}(n!)^2(2n+1)} z^{2n+1}, \quad (R = 1), \tag{2.18}$$

$$\operatorname{arcsinh} z = \sum_{n=0}^{\infty} (-1)^n \frac{(2n)!}{2^{2n}(n!)^2(2n+1)} z^{2n+1}, \quad (R = 1). \tag{2.19}$$

Next, let us move on to Laurent series.

We have seen that if f is differentiable in a domain D, and $z_0 \in D$, then we can write

$$f(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n, \tag{2.20}$$

for suitable coefficients a_n, and for z such that $|z - z_0| < R$, for some $R > 0$. The idea of Laurent series is to generalize (2.20) to allow negative powers of $(z - z_0)$. Observe that, given two convergent power series $f_1(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n$ and $f_2(z) = \sum_{n=n_0}^{\infty} b_n (z - z_0)^n$, $n_0 \in \mathbb{N}_{\geqslant 0}$ such that $a_{n_0} \neq 0$, the series

$$g(z) = \frac{f_1(z)}{f_2(z)} = \frac{1}{z^{n_0}} \frac{\sum_{n=0}^{\infty} a_n (z - z_0)^n}{\sum_{n=0}^{\infty} b_{n+n_0} (z - z_0)^{n+n0}}, \tag{2.21}$$

is well defined. Indeed, considering the ring of convergent power series, one can define the quotient field of it, which turns out to be the field of Laurent series.

Definition 2.5 (Laurent Series). *A Laurent series is a series of the form*

$$\sum_{n=-\infty}^{\infty} a_n (z - z_0)^n. \tag{2.22}$$

For the convergence, we split (2.22) in two parts and write

$$\sum_{n=1}^{\infty} a_{-n} (z - z_0)^{-n} + \sum_{n=0}^{\infty} a_n (z - z_0)^n = S_1 + S_2. \tag{2.23}$$

We have convergence only if S_1 and S_2 converge. Being familiar with S_2, we can say that S_2 converges for $|z - z_0| < R_2$ for some $R_2 \geqslant 0$, where R_2 is the radius of convergence of S_2. S_1 can be recognized as a power series in $\frac{1}{z - z_0}$, which has a radius of convergence $\frac{1}{R_1} \geqslant 0$. In other words S_2 converges when $|z - z_0| > R_1$.

Combining these, we see that if $0 \leqslant R_1 < R_2 \leqslant \infty$, then we have convergence in the annulus

$$\{z \in \mathbb{C} \mid R_1 < |z - z_0| < R_2\}. \tag{2.24}$$

Theorem 2.2 (Laurent's Theorem). *Suppose that f is holomorphic in the annulus $A = \{z \in \mathbb{C} \mid R_1 < |z - z_0| < R_2\}$, where $0 \leqslant R_1 < R_2 \leqslant \infty$. Then we can write f as a Laurent series*

$$f(z) = \sum_{n=1}^{\infty} a_{-n} (z - z_0)^{-n} + \sum_{n=0}^{\infty} a_n (z - z_0)^n, \ \forall z \in A. \tag{2.25}$$

Note that from this theorem we do not know that f is differentiable at z_0, because it might not be. S_1 in (2.23) is called the principal part of the Laurent series, and it is unique.

Definition 2.6.

- *A singularity of a function $f(z)$ is a point z_0 at which $f(z)$ is not differentiable.*

- *If there exists a punctured disc $0 < |z - z_0| < R$ such that f is differentiable on this punctured disc then we say that z_0 is an isolated singularity of f. This is an annulus with $R_1 = 0$.*

Here we are interested in isolated singularities where Laurent expansion is valid for $0 < |z - z_0| < R$. Depending on the form of the principal part, we define the notion of singularity.

Definition 2.7 (Removable Singularities)**.** *Suppose that f has an isolated singularity at z_0 and that the principal part of the Laurent series has no terms. In this case, for $0 < |z - z_0| < R$ we have*

$$f(z) = a_0 + a_1(z - z_0) + \cdots + a_n(z - z_0)^n + \cdots . \tag{2.26}$$

The radius of convergence of this power series is at least R, and so $f(z)$ extends to a function that is differentiable at z_0.

Example 2.3. *As example, for expansions at $z_0 = 0$ one could cite*

$$\frac{1}{\sin(z)} - \frac{1}{z} = \frac{z}{6} + \frac{7z^3}{360} + \frac{31z^5}{15120} + \ldots = 2\sum_{n=0}^{\infty} \frac{(-1)^n \left(2^{2n+1} - 1\right) B_{2n+2}}{(2(n+1))!} z^{2n+1}, \tag{2.27}$$

$$\frac{\arcsin\left(\sqrt{z}\right)}{\sqrt{z}} = 1 + \frac{1}{6}z + \frac{3}{40}z^2 + \ldots = \sum_{n=0}^{\infty} \frac{(2n)! \cdot}{(2n+1)\, 4^n n!^2} z^n, \tag{2.28}$$

$$\begin{aligned} \frac{z}{\exp(z) - 1} &= 1 - \frac{1}{2}z + \frac{1}{12}z^2 - \frac{1}{720}z^4 + \ldots \\ &= \sum_{n=0}^{\infty} A_n z^n, A_n = \sum_{k=1}^{n} -\frac{A_{n-k}}{(k+1)!}, A_0 = 1 \qquad (2.29) \\ &= \sum_{n=0}^{\infty} \frac{B_n}{n!} z^n, \qquad (2.30) \end{aligned}$$

where B_n denotes the n^{th} Bernoulli number.

Definition 2.8 (Poles)**.** *Suppose that f has an isolated singularity at z_0 and that the principal part of the Laurent series has finitely many terms. In this case, for $0 < |z - z_0| < R$, we can write*

$$f(z) = \frac{a_{-n_0}}{(z - z_0)^{n_0}} + \cdots + \frac{a_{-1}}{(z - z_0)} + \sum_{n=0}^{\infty} a_n (z - z_0)^n \tag{2.31}$$

where $a_{-n_0} \neq 0$. And we say that f has a pole of order n_0 at z_0.

Example 2.4.

$$\frac{\arctan(z)}{z^4} = \frac{1}{z^3} - \frac{1}{3z} + \frac{z}{5} - \frac{z^3}{7} + \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} z^{2n-3} \tag{2.32}$$

$$\cot(z) = \frac{1}{z} - \frac{z}{3} - \frac{z^3}{45} + \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n 2^{2n} B_{2n}}{(2n)!} z^{2n-1} \tag{2.33}$$

$$\begin{aligned} \csc(z) &= \frac{1}{z} + \frac{z}{6} + \frac{7 \cdot z^3}{360} + \ldots \\ &= \sum_{n=0}^{\infty} A_n z^{2n-1}, A_n = \sum_{k=1}^{n} -\frac{(-1)^k A_{n-k}}{(2k+1)!}, A_0 = 1 && (2.34) \\ &= 2\sum_{n=0}^{\infty} \frac{(-1)^{n-1}\left(2^{2n-1}-1\right) B_{2n}}{(2n)!} z^{2n-1} && (2.35) \end{aligned}$$

Functions with removable singularities or poles having a representation of type (2.31) are gathered in a class of functions called meromorphic functions, that corresponds to analytic functions having additionally finitely many poles.

Definition 2.9 (Essential Singularities)**.** *Suppose that f has an isolated singularity at z_0 and that the principal part of the Laurent series has infinitely many terms. In this case we say that f has an essential singularity.*

A classical example with an essential singularity at $z_0 = 0$ is $\exp(1/z)$. The Laurent series is defined on the annulus $\{z,\ 0 < |z| < \infty\}$, and we write

$$\exp\left(\frac{1}{z}\right) = 1 + \frac{1}{z} + \frac{1}{2z^2} + \frac{1}{6z^3} + \cdots = \sum_{n=0}^{\infty} \frac{1}{n! z^n}. \tag{2.36}$$

When there is an essential singularity, the manipulation of power series is quite critic. We will be dealing with some of them for asymptotic expansions. These particular power series converge only if the corresponding function is analytic at the point of expansion in a certain region (see [[Malham, 2005], [Copley, 2015]]). However, divergent asymptotic series have more interest as they provide meaningful information on the expanded functions [Boyd, 1999]. We will only consider asymptotic series on the real axis, essentially for expansions at ∞. The formal definition of asymptotic series can be stated as follows (see [Malham, 2005, Section 3.2]).

Definition 2.10 (Asymptotic Sequence)**.** *A sequence of scale (sometimes called gauge) functions $(\varepsilon_n(x))$, $n = 0, 1, \ldots$ is said to form an asymptotic sequence as $x \to \infty$, if for all n,*

$$\varepsilon_{n+1}(x) = o(\varepsilon_n(x)), \text{ that is } \lim_{n\to\infty} \frac{\varepsilon_{n+1}(x)}{\varepsilon_n(x)} = 0, \tag{2.37}$$

as $x \to \infty$.

Example 2.5. $x^{-n/k}, k \in \mathbb{N}$.

Definition 2.11 (Asymptotic Expansion). *If $(\varepsilon_n(x))$ is an asymptotic sequence of functions as $x \to \infty$, we say that*

$$\sum_{n=0}^{\infty} a_n \varepsilon_n(x) \tag{2.38}$$

where the a_n are constants, is an asymptotic expansion of $f(x)$ as $x \to \infty$ if for each N

$$f(x) = \sum_{n=0}^{N} a_n \varepsilon_n(x) + o(\varepsilon_N(x)), \tag{2.39}$$

as $x \to \infty$. And we write

$$f(x) \sim \sum_{n=0}^{\infty} a_n \varepsilon_n(x) \ (x \to \infty). \tag{2.40}$$

The coefficients of an asymptotic expansion are uniquely determined by the formulas

$$a_0 = \lim_{x\to\infty} \frac{f(x)}{\varepsilon_0(x)} \tag{2.41}$$

$$a_N = \lim_{x\to\infty} \frac{f(x) - \sum_{j=0}^{N-1} a_j \varepsilon_i(x)}{\varepsilon_N(x)}, N = 1, 2, \dots \tag{2.42}$$

Among the well known divergent asymptotic expansions, we can cite

$$x \exp(-x) E_i(x) \sim \sum_{n=0}^{\infty} \frac{n!}{x^n} \ (x \to \infty), \tag{2.43}$$

where $E_i(x) := \int_{-\infty}^{x} \frac{\exp(-t)}{t}\, dt$ denotes the exponential integral function;

$$\sqrt{\pi} \exp(x) \left(1 - \mathrm{erf}(\sqrt{x})\right) \sim \sum_{n=0}^{\infty} \frac{(-1)^n (2n)!}{4^n n! x^{1/2+n}} \ (x \to \infty), \tag{2.44}$$

where $\mathrm{erf}(x) := \frac{2}{\sqrt{(\pi)}} \int_0^x \exp(-t^2)\, dt$ denotes the error function.

Further examples for the convergent case at infinity are

$$\exp\left(\frac{1}{x}\right) = \sum_{n=0}^{\infty} \frac{1}{n! x^n}, \tag{2.45}$$

$$\arctan(x) = \frac{\pi}{2} - \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1) x^{2n+1}}. \tag{2.46}$$

A special thing with asymptotic expansions is that a given function can only have one asymptotic series. However, knowledge of an asymptotic series does not determine a corresponding function since different functions can generate the same asymptotic series. That is the case for $\exp\left(\frac{1}{z}\right)$ and $\exp\left(\frac{1}{z}\right) + \exp(-z)$ at ∞.

Finally let us say some few words about Puiseux series, more details can be found in ([Casas-Alvero, 2000, Newton-Puiseux algorithm, pages 15-38], [Nowak, 2000]). Referring to the so called Newton-Puiseux theorem, the Puiseux series come as roots of Laurent polynomials of two variables. In fact, the set of Puiseux series is the algebraic closure of the field of convergent Laurent series.

Definition 2.12 (Puiseux Series). *A Puiseux series is a series of the form*

$$\sum_{n=n_0}^{\infty} a_n (z-z_0)^{n/k} = \sum_{n=n_0}^{\infty} a_n \sqrt[k]{(z-z_0)^n}, \tag{2.47}$$

where k is a positive integer, and n_0 any integer.

In other words, Puiseux series differ from Laurent series in that they allow fractional exponents of the indeterminate, as long as these fractional exponents have bounded denominator (here k).

Observe that if f is analytic at $z_0 \in \mathbb{C}$, then we could have a Laurent series from f by shifting its power series expansion at z_0, say, z^{n_0}; and we could also have a Puiseux series by substituting the indeterminate by a fixed fractional power of it, say, $z^{1/k}$. The two transformations lead to a Laurent-Puiseux or Puiseux expansion which corresponds to the series expansion of $z^{n_0} f(z^{1/k})$. Therefore, one sees that our duty of computing a power series expansion, which mainly relies on the determination of a formula for the general coefficient, is first of all related to convergent power series. Secondly, the general shifted or fractional power has to be deduced implicitly.

Example 2.6. *At $z_0 = 0$ we have*

$$\frac{\sin(z^{1/2})}{z^3} = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} z^{\frac{2\cdot n-5}{2}} \tag{2.48}$$

$$= \frac{1}{z^{\frac{5}{2}}} - \frac{1}{6\cdot z^{\frac{3}{2}}} + \frac{1}{120\cdot\sqrt{z}} - \frac{\sqrt{z}}{5040} + \frac{z^{\frac{3}{2}}}{362880} + \dots \tag{2.49}$$

$$\sin(z^{1/2}) + \cos(z^{1/4}) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} z^{\frac{1+2\cdot n}{2}} + \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} z^{\frac{n}{2}} \tag{2.50}$$

$$= 1 + \frac{\sqrt{z}}{2} + \frac{z}{24} - \frac{121 z^{\frac{3}{2}}}{720} + \frac{z^2}{40320} + \dots \tag{2.51}$$

2.2 Power Series in Computer Algebra

As ended in the previous section, one sees that the Laurent-Puiseux series are mainly characterized by their general coefficients, which can be deduced using changes on the summation variable and the general power of the indeterminate. It is therefore important to first understand the case where $n_0 = 0$ and $k = 1$ in (2.47), as the goal of finding the general coefficient of a Laurent-Puiseux series can always be reduced to this case. Nevertheless, we will see that all the properties proved in this section hold for any Laurent-Puiseux series, and in Chapter 8, we will explain how by these properties the Puiseux number and the general coefficient are algorithmically found. For more details about this section, see [Koepf, 1992]. Without loss of generality we assume that $z_0 = 0$, and start with a power series of the form

$$f(z) = \sum_{n=0}^{\infty} a_n z^n,\ a_n \in \mathbb{K}, \tag{2.52}$$

where $\mathbb{K}$ is a field of characteristic zero. The idea of the power series computation is based on the knowledge about the generalized hypergeometric series [Koepf, 2014].

$$ {}_pF_q\left(\begin{matrix} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{matrix} \middle| \, z\right) := \sum_{n=0}^{\infty} \frac{(a_1)_n \cdot (a_2)_n \cdots (a_p)_n}{(b_1)_n \cdot (b_2)_n \cdots (b_q)_n n!} z^n = \sum_{n=0}^{\infty} A_n z^n. \tag{2.53} $$

Here $(a)_n$ denotes the *shifted factorial of* a, also called *Pochhammer symbol of* a, defined as

$$ (a)_n := \begin{cases} 1 & \text{if } n = 0 \\ a \cdot (a+1) \cdots (a+n-1) & \text{if } n \in \mathbb{N}^* \end{cases} . \tag{2.54} $$

The coefficients are

$$ A_n := \frac{(a_1)_n \cdot (a_2)_n \cdots (a_p)_n}{(b_1)_n \cdot (b_2)_n \cdots (b_q)_n n!}, \quad n = 0, 1, 2, \ldots, \tag{2.55} $$

where the denominator factors are chosen in such a way that they can never be zero. We have

$$ A_0 = \frac{(a_1)_0 \cdot (a_2)_0 \cdots (a_p)_0}{(b_1)_0 \cdot (b_2)_0 \cdots (b_q)_0 0!} = 1, \tag{2.56} $$

and for $n \geqslant 0$, using the fact that $(a)_{n+1} = (a)_n(a+n)$ and $(n+1)! = (n+1)n!$ we have

$$ \frac{A_{n+1}}{A_n} = \frac{(a_1)_n(n+a_1) \cdot (a_2)_n(n+a_2) \cdots (a_p)_n(n+a_p)}{(b_1)_n(n+b_1) \cdot (b_2)_n(n+b_2) \cdots (b_q)_n(n+b_q)(n+1)n!} \times \frac{(b_1)_n \cdot (b_2)_n \cdots (b_q)_n n!}{(a_1)_n \cdot (a_2)_n \cdots (a_p)_n}, $$

which gives

$$ \frac{A_{n+1}}{A_n} = \frac{(n+a_1)(n+a_2) \cdots (n+a_p)}{(n+b_1)(n+b_2) \cdots (n+b_q)(n+1)}. \tag{2.57} $$

When one of the numerator parameters a_i is a negative integer, there exists $n \in \mathbb{N}$, $a_i + n = 0$, so the generalized hypergeometric function will have a finite number of coefficients which is nothing but a polynomial in z (Section 1.4 in [Koekoek et al., 2010]). Otherwise, the radius of convergence R of the generalized hypergeometric series is given by

$$ R = \lim_{n \to \infty} \frac{A_n}{A_{n+1}} = \lim_{n \to \infty} \frac{n^{q+1}}{n^p} = \begin{cases} \infty & \text{if } \; p < q+1 \\ 1 & \text{if } \; p = q+1 \\ 0 & \text{if } \; p > q+1 \end{cases} . \tag{2.58} $$

Of course the most interesting cases are the cases where $R \neq 0$.

For the case where $R = 1$ we have the following examples:

$$ \begin{aligned} {}_1F_0\left(\begin{matrix} -a \\ - \end{matrix} \middle| -z\right) = {}_2F_1\left(\begin{matrix} -a & b \\ b & \end{matrix} \middle| -z\right) &= \sum_{n=0}^{\infty} \frac{(-a)_n (b)_n}{(b)_n n!} (-z)^n \\ &= \sum_{n=0}^{\infty} (-1)^{2n} \frac{a \cdot (a-1) \cdots (a-n+1)}{n!} z^n \\ &= \sum_{n=0}^{\infty} \frac{a \cdot (a-1) \cdots (a-n+1)}{n!} z^n \\ &= (1+z)^a. \end{aligned} \tag{2.59} $$

By remarking that $(1)_n = 1(1+1)\cdots(1+n-1) = n!$ and $(2)_n = 2\cdot 3\cdots(2+n-1) = (n+1)!$, we have

$$\begin{aligned} z\,{}_2F_1\left(\begin{matrix}1 & 1\\ 2 & \end{matrix}\middle|\, -z\right) &= z\cdot\sum_{n=0}^{\infty}\frac{(1)_n(1)_n}{(2)_n n!}(-z)^n \\ &= \sum_{n=0}^{\infty}\frac{n!n!}{(n+1)!n!}(-1)^n z^{n+1} \\ &= \sum_{n=0}^{\infty}(-1)^n\frac{z^{n+1}}{n+1} \\ &= \ln(1+z). \end{aligned} \tag{2.60}$$

We also have

$$\begin{aligned} \frac{\left(\frac{1}{2}\right)_n}{\left(\frac{3}{2}\right)_n} &= \frac{\frac{1}{2}\cdot\left(\frac{1}{2}+1\right)\cdots\left(\frac{1}{2}+n-1\right)}{\frac{3}{2}\cdot\left(\frac{3}{2}+1\right)\cdots\left(\frac{3}{2}+n-1\right)} \\ &= \frac{\frac{1}{2}\cdot\left(\frac{3}{2}\right)\left(\frac{3}{2}+1\right)\cdots\left(\frac{3}{2}+n-2\right)}{\frac{3}{2}\cdot\left(\frac{3}{2}+1\right)\cdots\left(\frac{3}{2}+n-2\right)\cdot\left(\frac{3}{2}+n-1\right)} = \frac{1}{2n+1} \end{aligned} \tag{2.61}$$

and

$$\begin{aligned} \left(\frac{1}{2}\right)_n &= \frac{1}{2}\cdot\left(\frac{1}{2}+1\right)\cdots\left(\frac{1}{2}+n-1\right) \\ &= \frac{\prod_{j=0}^{n-1}(2j+1)}{2^n} = \frac{\prod_{j=0}^{n-1}(2j+1)(2(j+1))}{2^n\prod_{j=0}^{n-1}2(j+1)} = \frac{(2n)!}{2^{2n}n!} \end{aligned} \tag{2.62}$$

which lead to

$$\begin{aligned} z\,{}_2F_1\left(\begin{matrix}1/2 & 1/2\\ 3/2 & \end{matrix}\middle|\, z^2\right) &= z\cdot\sum_{n=0}^{\infty}\frac{\left(\frac{1}{2}\right)_n\left(\frac{1}{2}\right)_n}{\left(\frac{3}{2}\right)_n n!}z^{2n} \\ &= \sum_{n=0}^{\infty}\frac{(2n)!}{2^{2n}(n!)^2(2n+1)}z^{2n+1} \\ &= \arcsin z. \end{aligned} \tag{2.63}$$

One can also show that

$$z\,{}_2F_1\left(\begin{matrix}1/2 & 1\\ 3/2 & \end{matrix}\middle|\, -z^2\right) = \arctan z. \tag{2.64}$$

For $R=\infty$, we have:

$${}_0F_0\left(\begin{matrix}-\\ -\end{matrix}\middle|\, z\right) = e^z, \tag{2.65}$$

$$z\cdot\ {}_0F_1\left(\begin{matrix}-\\ 3/2\end{matrix}\middle|\, -\frac{z^2}{4}\right) = z\cdot\sum_{n=0}^{\infty}\frac{1}{\left(\frac{3}{2}\right)_n n!}\left(-\frac{z^2}{4}\right)^n = \sum_{n=0}^{\infty}\frac{(-1)^n}{4^n\left(\frac{3}{2}\right)_n n!}z^{2n+1}$$

and using the same reasoning as we did for $\left(\frac{1}{2}\right)_n$, we find that

$$\left(\frac{3}{2}\right)_n = \frac{(2n+1)!}{4^n n!}, \tag{2.66}$$

which leads to

$$z \cdot {}_0F_1\left(\begin{matrix} - \\ 3/2 \end{matrix} \middle| -\frac{z^2}{4}\right) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} z^{2n+1} = \sin z. \tag{2.67}$$

In the same way

$$ {}_0F_1\left(\begin{matrix} - \\ 1/2 \end{matrix} \middle| -\frac{z^2}{4}\right) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} z^{2n} = \cos z, \tag{2.68}$$

and

$$ {}_0F_1\left(\begin{matrix} - \\ 1/2 \end{matrix} \middle| \frac{z^2}{4}\right) = \sum_{n=0}^{\infty} \frac{1}{(2n)!} z^{2n} = \cosh z. \tag{2.69}$$

It follows clearly that all the examples given in Example 2.2 can be expressed in generalized hypergeometric form. Thus one can see that many analytic functions can be written as generalized hypergeometric series.

The recurrence relation (2.57) is the Recurrence Equation (RE) that characterizes the generalized hypergeometric series $f(z) = \sum_{n=0}^{\infty} A_n z^n$. Note that $\dfrac{A_{n+1}}{A_n}$ is a rational function in n. Generally, having a rational function $r(n) = \dfrac{A_{n+1}}{A_n}$ connects the corresponding function with a hypergeometric series. Indeed $r(n)$ can be factorized over the algebraic closure of $\mathbb{K}$ as

$$\begin{aligned} r(n) &= \frac{\alpha(n+a_1)(n+a_2)\cdots(n+a_p)}{\beta(n+b_1)(n+b_2)\cdots(n+b_q)(n+b_{q+1})} \\ &= c\frac{(n+a_1)(n+a_2)\cdots(n+a_p)}{(n+b_1)(n+b_2)\cdots(n+b_q)(n+b_{q+1})},\ c = \frac{\alpha}{\beta}, \end{aligned}$$

where the complex numbers α and β ($\alpha\beta \neq 0$) are, respectively, the leading coefficients of the numerator and the denominator, $-b_i, i \in [\![1, q+1]\!]$[2] are the poles of r and $-a_i, i \in [\![1, p]\!]$ are the zeros of r. If there is some $b_i = -1$ ($b_{q+1} = -1$), then the function corresponds to a hypergeometric series evaluated at some point cz (c being the quotient of the leading coefficients of the numerator and the denominator of r). Whereas if there is no such b_i, the extra factor $(n+1)$ can be compensated by one of the factors $(n+a_i)$ in the numerator by taking $a_{p+1} = -1$.

Theorem 2.3. *Let*

$$f(z) = {}_pF_q\left(\begin{matrix} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{matrix} \middle| z\right) = \sum_{n=0}^{\infty} A_n z^n,$$

and the differential operators $D = \dfrac{d}{dz}$ *and* $\theta = z\dfrac{d}{dz} = zD$. *Then* f *satisfies the differential equation*

$$\theta(\theta+b_1-1)\cdots(\theta+b_q-1)f = z(\theta+a_1)\cdots(\theta+a_p)f. \tag{2.70}$$

Proof.

[2]For $l, k \in \mathbb{N}, l < k$ we define $[\![l, k]\!] := \{l, l+1, \ldots, k\}$

To see how (2.70) is obtained we first remark that

$$\theta(f(z)) = zD\left(\sum_{n=0}^{\infty} A_n z^n\right) = z\sum_{n=1}^{\infty} nA_n z^{n-1} = \sum_{n=0}^{\infty} nA_n z^n$$

and for $j \geqslant 1$, assuming $\theta^j(f(z)) = \sum_{n=0}^{\infty} n^j A_n z^n$ implies

$$\theta^{j+1}(f(z)) = zD\left(\sum_{n\geqslant 0} n^j A_n z^n\right) = z\sum_{n=1}^{\infty} n^{j+1} A_n z^{n-1} = \sum_{n=0}^{\infty} n^{j+1} A_n z^n.$$

So by induction we have

$$\theta^j(f(z)) = \sum_{n=0}^{\infty} n^j A_n z^n, \; j \in \mathbb{N}. \tag{2.71}$$

Thus for any polynomial T we can state by linearity that

$$T(\theta)(f(z)) = \sum_{n=0}^{\infty} T(n) A_n z^n. \tag{2.72}$$

From the recurrence relation of the generalized hypergeometric series (2.57) we have

$$A_{n+1} Q(n) = A_n P(n), \; n \in \mathbb{N}_{\geqslant 0} \tag{2.73}$$

where

$$Q(n) = (b_1 + n)\cdots(b_q + n)(n+1), \text{ and } P(n) = (a_1 + n)\cdots(a_p + n), \tag{2.74}$$

and therefore,

$$\sum_{n=0}^{\infty} A_{n+1}(b_1 + n)\cdots(b_q + n)(n+1)z^{n+1} = \sum_{n=0}^{\infty}(a_1 + n)\cdots(a_p + n)A_n z^{n+1}. \tag{2.75}$$

We first work on the left hand side. Setting $j = n + 1$, this is equivalent to

$$\sum_{n=0}^{\infty} A_{n+1}(b_1 + n)\cdots(b_q + n)(n+1)z^{n+1} = \sum_{j=1}^{\infty} A_j(j + b_1 - 1)\cdots(j + b_q - 1)jz^j,$$

and according to (2.74), each coefficient gives

$$A_j(j + b_1 - 1)\cdots(j + b_q - 1)j = Q(j-1)A_j z^j.$$

Thus,

$$\begin{aligned}
\sum_{j=1}^{\infty} A_j(j + b_1 - 1)\cdots(j + b_q - 1)j &= \sum_{j=1}^{\infty} Q(j-1)A_j z^j \\
&= Q(\theta - 1)\left(f(z) - A_0\right) \text{ from } (2.72) \\
&= \theta(\theta + b_1 - 1)\cdots(\theta + b_q - 1)f(z),
\end{aligned}$$

where the last line comes from the substitution of n by $\theta - 1$ in $Q(n)$ in (2.74). Notice that $Q(\theta - 1)A_0 = 0$ since Q is a polynomial whose lowest monomial degree is 1 and $\theta(A_0) = 0$.

With a similar reasoning for the right hand side of (2.75) we get

$$\begin{aligned} z\left(\sum_{n=0}^{\infty}(n+a_1)\cdots(n+a_p)A_n z^n\right) &= z\left(\sum_{n=0}^{\infty}P(n)A_n z^n\right) \\ &= zP(\theta)(f(z)) \\ &= z(\theta+a_1)\cdots(\theta+a_p)f(z). \end{aligned}$$

Hence, we come up with the differential equation

$$\theta(\theta+b_1-1)\cdots(\theta+b_q-1)f = z(\theta+a_1)\cdots(\theta+a_p)f,$$

as expected. □

Furthermore, if we expand (2.70) in terms of the derivatives $D^j f = \frac{d^j f}{dz^j}$ of f, then we will obtain a differential equation of the form $(M := \max(p,q)+1,\ c_{j,l} \in \mathbb{C})$

$$\sum_{j=0}^{M} T_j(z)D^j f = \sum_{j=0}^{M}\sum_{l=0}^{M} c_{j,l} z^l D^j f = 0,. \tag{2.76}$$

Definition 2.13 (Holonomic Differential Equation [Koepf, 2014]). *A holonomic differential equation is a linear homogeneous ordinary differential equation with polynomial coefficients*

$$T_n(z)D^n f(z) + \cdots + T_1(z)Df(z) + T_0(z)f(z) = 0, \tag{2.77}$$

$T_n, \ldots, T_1, T_0 \in \mathbb{K}[z]$.

A function satisfying a holonomic differential equation is called holonomic function. In particular, the generalized hypergeometric function is holonomic.

Proposition 2.4. *Any Laurent-Puiseux series*

$$f(z) = \sum_{n=n_0}^{\infty} A_n z^{n/k},\ n_0 \in \mathbb{Z}, k \in \mathbb{N}, \tag{2.78}$$

with general coefficient A_n satisfying (2.57) for all integers $n \geqslant n_0$, is holonomic.

Proof.

Let $\theta_k = kz\frac{d}{dz}$.

By induction, one easily proves that

$$\theta_k^j(f(z)) = \sum_{n=0}^{\infty} n^j A_n z^{n/k},\ j \in \mathbb{N}. \tag{2.79}$$

and therefore for any polynomial $T \in \mathbb{K}[z]$

$$T(\theta_k)\,(f(z)) = \sum_{n=n_0}^{\infty} T(n)A_n z^{n/k}. \tag{2.80}$$

From (2.57), we have

$$A_{n+1} = r(n)A_n, \forall n \in \mathbb{Z}_{\geqslant n_0},\ r(n) = \frac{(n+a_1)(n+a_2)\cdots(n+a_p)}{(n+b_1)(n+b_2)\cdots(n+b_q)(n+1)},$$

therefore

$$A_{n+k} = \prod_{j=0}^{k-1} r(n+j)A_n = R(n)A_n, \tag{2.81}$$

which gives another representation of $f(z)$.

Without loss of generality, we assume that $R(n) = \dfrac{P(n)}{Q(n)}$, where

$$Q(n) = (n-n_0+1)\cdots(n-n_0+k)\prod_{j=0}^{k-1}(b_1+n+j)\cdots(b_q+n+j)(n+1+j) \tag{2.82}$$

and

$$P(n) = (n-n_0+1)\cdots(n-n_0+k)\prod_{j=0}^{k-1}(a_1+n+j)\cdots(a_p+n+j). \tag{2.83}$$

(2.81) is equivalent to

$$Q(n)A_{n+k} = P(n)A_n. \tag{2.84}$$

From (2.82), (2.80) and (2.84) we get

$$\begin{aligned}
Q(\theta_k-k)\,(f(z)) &= \sum_{n=n_0}^{\infty} Q(n-k)A_n z^{n/k} \quad \text{by (2.80) as } Q \text{ is a polynomial}\\
&= \sum_{n=n_0+k}^{\infty} Q(n-k)A_n z^{n/k} \quad \text{as } Q(n_0-1)=\cdots=Q(n_0-k)=0 \text{ by (2.82)}\\
&= \sum_{n=n_0}^{\infty} Q(n)A_{n+k} z^{(n+k)/k} \quad \text{by an index shift}\\
&= z\sum_{n=n_0}^{\infty} P(n)A_n z^{n/k} \quad \text{by (2.84)}\\
&= zP(\theta_k)\,(f(z)) \quad \text{by (2.80) again.}
\end{aligned}$$

Therefore $f(z)$ satisfies the differential equation

$$(\theta_k-n_0+1)\cdots(\theta_k-n_0+k)\prod_{j=0}^{k-1}Q_j(\theta_k)f = z(\theta_k-n_0+1)\cdots(\theta_k-n_0+k)\prod_{j=0}^{k-1}P_j(\theta_k)f, \tag{2.85}$$

where $P_j(n) = (b_1+n+j)\cdots(b_q+n+j)(n+1+j)$ and $Q_j(n) = (a_1+n+j)\cdots(a_p+n+j)$, $j = 0,\ldots,k-1$.

After expansion of (2.85) in terms of the derivatives $D^j f$ of f we obtain a holonomic differential equation. □

Hence Laurent-Puiseux series of representation (2.78) with hypergeometric general coefficient are holonomic. Nevertheless, for more specificity about the kind of function that we will be dealing with, we introduce the following more general definition.

Definition 2.14 (Series of Hypergeometric Type [Koepf, 1992]). *A Laurent-Puiseux series (LPS) $f := \sum_{n=n_0}^{\infty} a_n (z-z_0)^{n/k}$, $k \in \mathbb{N}$ is said to be of hypergeometric type if its coefficients a_n satisfy an RE of the form*

$$\begin{aligned} a_{n+m} &= r(n) a_n \quad \textit{for } n \geqslant n_0 \\ a_n &= I_n \quad \textit{for } n = n_0, n_0+1, \cdots, n_0+m-1 \end{aligned} \tag{2.86}$$

for some $m \in \mathbb{N}$, $I_n \in \mathbb{C}$ $(n = n_0, n_0+1, \cdots, n_0+m-1)$, $I_{n_0} \in \mathbb{C} \setminus \{0\}$, and some rational function r. The number m is then called symmetry number of (the given representation) of f. A RE of this type is also called to be of hypergeometric type.

Remark Each Laurent-Puiseux series with symmetry number m can be represented as the sum of m-fold symmetric functions as follows

$$f(z) = \sum_{j=0}^{m-1} \sum_{n=0}^{\infty} I_{n_0+j} r(j+n_0+m) r(j+n_0+2m) \cdots r(j+n_0+mn) (z-z_0)^{(n_0+mn+j)/k}. \tag{2.87}$$

For example the general coefficient of the power series of $f(z) = \sin z + \cos z$ satisfies the holonomic recurrence equation

$$(1+n) \cdot (2+n) \cdot a_{n+2} + a_n = 0, \tag{2.88}$$

which for $m = 1$, does not have generalized hypergeometric term solutions in $\mathbb{Q}(n)$. But for $m = 2$, we find

$$a_{2n} = \frac{(-1)^n z^{2n}}{(2n)!}, \quad \text{and} \quad a_{2n+1} = \frac{(-1)^n}{(2n+1)!}. \tag{2.89}$$

Therefore, using the initial values $a_0 = 1, a_1 = 1$ (remember that we assumed $z_0 = 0$), by writing the sum of the corresponding m-fold symmetric functions we obtain

$$\cos z + \sin z = \sum_{n=0}^{\infty} \frac{(-1)^n z^{2n}}{(2n)!} + \sum_{n=0}^{\infty} \frac{(-1)^n z^{2n+1}}{(2n+1)!}. \tag{2.90}$$

Thus $f(z)$ might not directly be deduced to be of generalized hypergeometric form. We say that $f(z)$ is a hypergeometric type function with type 2. A function is said to be of hypergeometric type with type $m \in \mathbb{N}$ if it can be expanded as a Laurent-Puiseux series with symmetry number m. This designation will often be used and the symmetry number or the type will be specified if needed.

The following lemma gives some transformations on power series that preserve the hypergeometric type.

Lemma 2.2. *[Lemma 2.1 in [Koepf, 1992]] Let f be a power series of hypergeometric type. Then*

$$\text{(a)}\ z^s f\ (s \in \mathbb{Z}), \qquad \text{(b)}\ f(z^t)\ (t \in \mathbb{Q}), \qquad \text{(c)}\ f(Cz)\ (C \in \mathbb{C}),$$
$$\text{(d)}\ \int f \qquad\qquad \text{(e)}\ \frac{f(z) \pm f(-z)}{2}, \qquad \text{(f)}\ f',$$

are of hypergeometric type, too. If f has symmetry number m, then $f(z^n)$ has symmetry number nm, and $\dfrac{f(z) \pm f(-z)}{2}$ has symmetry number $2m$ as odd or even part of f.

To deal with functions of hypergeometric type in our development, it is essential that they satisfy a holonomic DE.

Theorem 2.4 (Theorem 8.1 in [Koepf, 1992])**.** *Each LPS of hypergeometric type satisfies a holonomic DE.*

Proof. The proof is similar to the one of Proposition 2.4, see [Koepf, 1993, Page 200]. □

Chapter 3

Symbolic Computation with Maxima

Maxima is a computer algebra system (CAS) developed in Lisp [McCarthy and Levin, 1965]. A CAS is a software which has the capability to automatically manipulate abstract objects in the traditional manner of mathematicians, and also allows numerical calculations. On the other hand, CAS is also used for programming in such a way that mathematicians can elaborate algorithms as a sequence of steps to achieve a certain goal. And these algorithms might be seen as proofs in the sense that knowledge about inputs and outputs allow to establish formulas and identities or show their non-existence.

Maxima is among the most popular CAS in the world. The Maxima source code can be compiled on many systems, including Windows, Linux, and MacOS X. The source code for all systems and precompiled binaries for Windows and Linux are available at the SourceForge file manager.

Maxima is a descendant of Macsyma, the legendary computer algebra system developed in the late 1960s at the Massachusetts Institute of Technology. It is the only system based on that effort still publicly available and with an active user community, thanks to its open source nature. Macsyma was revolutionary in its days, and many later systems, such as Maple and Mathematica, were inspired by it [Maxima's developers, 2019].

All the usual arithmetic operations: addition (+), subtraction (-), multiplication (*), division (/), exponentiation (^), modulo (mod), etc. are similarly usable symbolically and numerically (one can use float(x) to get the real approximation of x) in Maxima. The commands *floor(x)* and *ceiling(x)* are, respectively, used to return the largest integer less than or equal to x and the least integer greater or equal to x.

wxMaxima is a user-friendly graphical front-end where a user can test Maxima commands, visualize the results and save them in worksheets. In our case, we write programs (or codes) in a file with the extension *.mac* or *.max*, and put it in the Maxima directory of our computer. For use, we load the file with the Maxima command *batchload("name_of_the_file")*. Let us move to some computations and Maxima programming on wxMaxima.

```
(%i1) number:1+2*3/4+5^2;
```

$$(\%o1) \quad \frac{55}{2}$$

```
(%i2) float(number);
```

$$(\%o2) \quad 27.5$$

```
(%i3) floor(number);
```

$$(\%o3) \quad 27$$

```
(%i4) ceiling(number);
```

$$(\%o4) \quad 28$$

```
(%i5) mod(number,2);
```

$$(\%o5) \quad \frac{3}{2}$$

Rational numbers are recognized in Maxima by the boolean function *numberp(x)* which returns *true* if *x* is rational and *false* otherwise. A more general function is *constantp* which recognizes all the Maxima constants. These commands are useful for restriction of computations in the field of rational numbers or its algebraic extensions.

The conditional evaluation, the for-loop and the while-loop work in Maxima as follows.

```
(%i6) number:1+2*3/4+5^2;
```

$$(\%o6) \quad \frac{55}{2}$$

```
(%i7) if %pi<%e or numberp(%pi)
      then ( print(%pi, "is rational"), number2:%pi*%e*%i*number)
      elseif constantp(number+%pi+%i+%e) and numberp(number)
      then number2:2*number
      else number2:number+%pi+%i+%e;
```

$$(\%o7) \quad 55$$

```
(%i8) for i:1 thru number2 step 2 do number: number-1;
```

$$(\%o8) \quad done$$

```
(%i9) while 2*number<number2 do number:number+1/2;
```

$$(\%o9) \quad done$$

```
(%i10) is(2*number=number2);
```

$$(\%o10) \quad \text{true}$$

where *%pi*, *%i* and *%e* denotes, respectively, the well known constants π, the imaginary number i and the Euler constant $\exp(1) = e$. The Maxima commands *print* is for displaying expressions, and *is* is used to evaluate boolean expressions. Note that when there are more than one sub-instruction in a conditional evaluation, a for-loop or a while loop, the instructions are separated by commas and gathered in parenthesis.

Maxima calculates integers and rational numbers with an arbitrary precision.

```
(%i11) 50!;
```

$$(\%o1) \quad 30414093201713378043612608166064768844377641568960512000000000000$$

The *factorial* command or the sign *!* will often be seen in the outputs of our algorithms. Let us see some manipulations.

```
(%i12) (n+1)!;
```

$$(\%o2) \quad (1+n)!$$

```
(%i13) n!;
```

$$(\%o3) \quad n!$$

```
(%i14) minfactorial((n+1)!/%);
```

$$(\%o4) \quad n+1$$

```
(%i15) makegamma((n+1)!);
```

$$(\%o5) \quad \Gamma(n+2)$$

```
(%i16) pochhammer(1,n);
```

$$(\%o6) \quad n!$$

```
(%i17) pochhammer(2,n);
```

$$(\%o7) \quad (2)_n$$

```
(%i18) makefact(makegamma(pochhammer(2,n)));
```

$$(\%o8) \quad (1+n)!$$

Note that the use of % in the input (%i14) above means use of the previous output, here (%o13).

One of the main mathematical objects used in CAS are polynomials. The Maxima command *expand* expands a polynomial expression, whereas the *factor* command succeeds in factorizing any multivariate polynomial over $\mathbb{Q}$ (see [Koepf, 1995c]). We can see the timings of computations by setting the Maxima global boolean variable *showtime* to *true*.

```
(%i19) showtime:true$
```

Evaluation took 0.0000 seconds (0.0000 elapsed)

```
(%i20) f:expand(product(product(j*z-y^k,k,1,3),j,1,2));
```

Evaluation took 0.0000 seconds (0.0000 elapsed)

$$(\%o20)\quad y^{12}-3\cdot z\cdot y^{11}+2\cdot z^2\cdot y^{10}-3\cdot z\cdot y^{10}+9\cdot z^2\cdot y^9-3\cdot z\cdot y^9 -6\cdot z^3\cdot y^8+11\cdot z^2\cdot y^8-12\cdot z^3\cdot y^7+9\cdot z^2\cdot y^7+4\cdot z^4\cdot y^6-27\cdot z^3\cdot y^6+2\cdot z^2\cdot y^6+18\cdot z^4\cdot y^5-12\cdot z^3\cdot y^5+ 22\cdot z^4\cdot y^4-6\cdot z^3\cdot y^4-12\cdot z^5\cdot y^3+18\cdot z^4\cdot y^3-12\cdot z^5\cdot y^2+4\cdot z^4\cdot y^2-12\cdot z^5\cdot y+8\cdot z^6$$

```
(%i21) factor(f);
```

Evaluation took 0.0100 seconds (0.0100 elapsed)

$$(\%o21)\quad (y-2\cdot z)\cdot(y-z)\cdot\left(y^2-2\cdot z\right)\cdot\left(y^2-z\right)\cdot\left(y^3-2\cdot z\right)\cdot\left(y^3-z\right)$$

The dollar sign at the end of a statement hides the output. This is often used at the end of our Maxima functions, so that there is no code printed out when we load our package.

We can also factorize rational expressions, and compute their partial fraction decompositions.

```
(%i22) g: factor((6+21*z+21*z^2+6*z^3)/(-2*z-4*z^2+6*z^3));
```

$$(\%o22)\quad \frac{3\cdot(1+z)\cdot(2+z)\cdot(1+2\cdot z)}{2\cdot(z-1)\cdot z\cdot(3\cdot z+1)}$$

```
(%i23) partfrac(g,z);
```

$$(\%o23)\quad \frac{5}{4\cdot(3\cdot z+1)}-\frac{3}{z}+\frac{27}{4\cdot(z-1)}+1$$

As there are algorithms to do so, one can solve linear systems of equations, as well as polynomial equations of order at most 4.

```
(%i24) e1: x + z = y$
(%i25) e2: 2*a*x - y = 2*a^2$
(%i26) e3: y - 2*z = 2$
(%i27) linsolve ([e1, e2, e3], [x, y, z]);
```

$$(\%o4)\quad [x=a+1, y=2\cdot a, z=a-1]$$

```
(%i28) solve(z^3+a*z^2-a*z=1,z);
```

$$(\%o28)\quad [z=-\frac{1+a+\sqrt{a^2+2\cdot a-3}}{2}, z=\frac{-1-a+\sqrt{a^2+2\cdot a-3}}{2}, z=1]$$

Besides the algebraic capabilities there are also analytic ones. We can calculate derivatives, limits and Taylor expansions of a certain order.

```
(%i29) f:atan(z)*sin(z);
```

$$(\%o29) \quad \operatorname{atan}(z)\cdot\sin(z)$$

```
(%i30) diff(f,z);
```

$$(\%o30) \quad \frac{\sin(z)}{z^2+1}+\operatorname{atan}(z)\cdot\cos(z)$$

```
(%i31) diff(f,z,2);
```

$$(\%o31) \quad -\operatorname{atan}(z)\cdot\sin(z)-\frac{2\cdot z\cdot\sin(z)}{(1+z^2)^2}+\frac{2\cdot\cos(z)}{z^2+1}$$

```
(%i32) limit(f,z,%pi/2);
```

$$(\%o32) \quad \operatorname{atan}\left(\frac{\pi}{2}\right)$$

```
(%i33) taylor(f,z,0,8);
```

$$(\%o33)/T/ z^2-\frac{z^4}{2}+\frac{19\cdot z^6}{72}-\frac{43\cdot z^8}{240}+\ldots$$

```
(%i34) DE:(1+z^2)^2*(2+2*z^2+z^4)*('diff(F(z),z,4))
       +4*z*(1+z^2)*(3+2*z^2+z^4)*('diff(F(z),z,3))
       +2*(6+16*z^2+9*z^4+4*z^6+z^8)*('diff(F(z),z,2))
       +4*z*(1+z^2)*(3+2*z^2+z^4)*('diff(F(z),z,1))
       +(10+26*z^2+11*z^4+4*z^6+z^8)*F(z);
```

$$(\%o34) \quad (1+z^2)^2\cdot(2+2\cdot z^2+z^4)\cdot\left(\frac{d^4}{d\,z^4}\cdot \mathrm{F}(z)\right)+4\cdot z\cdot(1+z^2)$$
$$\cdot(3+2\cdot z^2+z^4)\cdot\left(\frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)\right)+2\cdot(6+16\cdot z^2+9\cdot z^4+4\cdot z^6+z^8)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)$$
$$+4\cdot z\cdot(1+z^2)\cdot(3+2\cdot z^2+z^4)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)+(10+26\cdot z^2+11\cdot z^4+4\cdot z^6+z^8)\cdot \mathrm{F}(z)$$

```
(%i35) ratsimp(ev(DE,F(z)=f,diff));
```

$$(\%o35) \quad 0$$

We have used *ev* to evaluate the differential equation *DE* for *F(z)=f*, and *ratsimp* to simplify the computations. Some other commands used for simplifications are *rat, radcan, trigsimp, trigexpand, trigreduce*, etc; where of course those starting with *trig* correspond to trigonometric functions.

As we will be dealing with linear recurrence equations, let us see their writing in Maxima.

```
(%i36) RE:2*(1+n)*(2+n)*a[n+2]+(1+n)*(1+3*n)*a[n+1]
        +(-3-3*n+n^2)*a[n]-n*a[n-1]=0;
```

$$(\%o36)\quad 2\cdot(1+n)\cdot(2+n)\cdot a_{n+2}+(1+n)\cdot(1+3\cdot n)\cdot a_{n+1}+\left(-3-3\cdot n+n^2\right)\cdot a_n-n\cdot a_{n-1}=0$$

Remark the use of square bracket for sequence indices. We can take the left hand side (*lhs*) and collect the coefficients as follows

```
(%i37) L:args(lhs(RE));
```

$$(\%o37)\quad [2\cdot(n+1)\cdot(n+2)\cdot a_{n+2},(n+1)\cdot(3\cdot n+1)\cdot a_{n+1},\left(n^2-3\cdot n-3\right)\cdot a_n,-n\cdot a_{n-1}]$$

```
(%i38) term:L[1];
```

$$(\%o38)\quad 2\cdot(n+1)\cdot(n+2)\cdot a_{n+2}$$

```
(%i39) coeff(term,a[n+2]);
```

$$(\%o39)\quad 2\cdot(n+1)\cdot(n+2)$$

we use *args* to split an expression depending on the main operator (*op(expr)*) in it, here $+$. The output is a list, which is a very useful object. Connected with this Maxima object are the functions *map*, *sublist*, *makelist*, *append*, *lreduce* etc. Let us collect all the coefficients of the RE in $(\%o1)$ above.

```
(%i40) aterm:map(lambda([v], if op(v)="*" then args(v)
         else args(-v)),L);
```

$$(\%o40)\quad [[2,n+1,n+2,a_{n+2}],[n+1,3\cdot n+1,a_{n+1}],[n^2-3\cdot n-3,a_n],[n,a_{n-1}]]$$

```
(%i41) aterm:map(lambda([v], sublist(v,lambda([v1],
        not freeof(a,v1)))),aterm);
```

$$(\%o41)\quad [[a_{n+2}],[a_{n+1}],[a_n],[a_{n-1}]]$$

```
(%i42) aterm:lreduce(append,aterm);
```

$$(\%o42)\quad [a_{n+2},a_{n+1},a_n,a_{n-1}]$$

```
(%i43) makelist(coeff(L[i],aterm[i]),i,1,length(L));
```

$$(\%o43)\quad [2\cdot(n+1)\cdot(n+2),(n+1)\cdot(3\cdot n+1),n^2-3\cdot n-3,-n]$$

As one can see, the *lambda* command allows to define simple functions of one variable, usually for a single use. *freeof(a,expr)* returns a boolean value (true or false) depending on whether the variable *a* explicitly appears in *expr* or not.

Since 2014, the Maxima command *solve_rec* implements the Petkovšek algorithm to find hypergeometric term solutions of homogeneous linear recurrences. In case of hypergeometric term solutions, the output is a linear combination of them.

```
(%i44) load(solve_rec);
```

$$(\%o44) \quad /usr/share/maxima/5.37.2/share/solve\_rec/solve\_rec.mac$$

```
(%i45) showtime:true$
```

Evaluation took 0.0000 seconds (0.0000 elapsed)

```
(%i46) solve_rec(RE,a[n]);
```

WARNING: found some hypergeometrical solutions!
Evaluation took 0.4700 seconds (0.6300 elapsed)

$$(\%o11) \quad a_n = \frac{\%k_1}{n!} + \%k_2 \cdot (-1)^n$$

solve_rec implements four algorithms for recurrence equation with polynomial coefficients, namely, Abramov's algorithm for rational solutions, a similar approach than Koepf's to solve recurrence equations with constant coefficients [Gruntz and Koepf, 1995], and Petkovšek's algorithm Hyper [Petkovšek, 1992]. As we mentioned earlier, this is not necessary since they are all hypergeometric terms and this will be shown in Chapter 7. Though all its procedures are incorporated in the command *solve_rec(RE,a[n])*, this package contains commands like *solve_rec_poly*, *solve_rec_rat*, *solve_rec_hyper* that can be used in specific cases (see [Vodopivec, 2014]). Some of these commands will be used for comparison in the next chapter. For more details on the use of *solve_rec*, one can type *? solve_rec*, the *?* in front is used to ask for help. When two question marks are used instead, Maxima provides all its functions and variables that contain the specified word. Once all these commands are displayed, one can then select what is needed and ask for a specific help. This is very practical to get familiar with Maxima's commands. For example if one is looking for a particular solver, then one may type the following.

```
(%i47) ??solve;
```

0: Functions and Variables for solve_rec
1: Functions and Variables for to_poly_solve
2: Introduction to solve_rec
3: desolve (Functions and Variables for Differential Equations)
4: fast_linsolve (Functions and Variables for Affine)
5: funcsolve (Functions and Variables for Equations)
6: globalsolve (Functions and Variables for Equations)
7: linear_solver (Functions and Variables for zeilberger)
8: linsolve (Functions and Variables for Equations)
9: linsolvewarn (Functions and Variables for Equations)
10: linsolve_params (Functions and Variables for Equations)
11: minpack_solve (Functions and Variables for minpack)

12: modular_linear_solver (Functions and Variables for zeilberger)
13: solve (Functions and Variables for Equations)
14: solvedecomposes (Functions and Variables for Equations)
15: solveexplicit (Functions and Variables for Equations)
16: solvefactors (Functions and Variables for Equations)
17: solvenullwarn (Functions and Variables for Equations)
18: solveradcan (Functions and Variables for Equations)
19: solvetrigwarn (Functions and Variables for Equations)
20: solve_rec (Functions and Variables for solve_rec)
21: solve_rec_rat (Functions and Variables for solve_rec)
22: to_poly_solve (Functions and Variables for to_poly_solve)
Enter space-separated numbers, 'all' or 'none': none;

(%o12) true

Thus details on a particular command containing the word "solve" can be viewed by typing its corresponding number above (during run-time), or typing *none* and use one question mark *?* and the spelling of a chosen command to get its details. Another Maxima function in the same direction is *apropos* whose syntax is *apropos("word")*; it returns a list of Maxima's and user's (in the opened session) functions and variables containing the specified word.

The command that we use to print out a power series is *sum(expression, variable, first, last)* which sums *expression* for *variable=first* up to *variable=last*. Whenever *last - first* is a positive integer, Maxima's *sum* command always simplifies the output. For the other cases, one can set the global boolean variable *simpsum* to true in order to get some closed forms if available. But we are not interested in such computations. Note moreover that the Maxima *sum* command may behave differently from the user thought [Koepf, 1995c].

```
(%i46) f:n^2;
```

(%o48) n^2

```
(%i49) k:n;
```

(%o49) n

```
(%i50) sum(f,k,1,10);
```

(%o50) $10 \cdot n^2$

```
(%i51) sum(subst(j,k,f),j,1,10);
```

(%o51) 385

This is important because it shows which type of behavior we can have if an atomic variable is given as input (here *n*). The point is that the second argument is not evaluated, and in particular when dealing with infinite sums, the variable *j* has to be substituted by *k* again.

We have already seen that the Maxima command *powerseries* is limited in its computation of power series, since it uses a pattern matching model. Its procedure can be seen as follows: for *powerseries*(f,z,z_0),

- Maxima tries to expand f in the variable $z - z_0$ by using some additional knowledge on f,
- logarithms $\log f$ are handled by the rule $\int \frac{f'}{f}$,
- for rational functions a real partial fraction decomposition is used,
- the power series expansions of the standard elementary functions with point of development $z_0 = 0$ are incorporated.

Some disadvantages of this procedure are its failures

- in finding the result for all rational functions like $\frac{1}{z^2+z+1}$, which is worse than the Maple case since it only uses real partial fraction decompositions,

  ```
  (%i52) powerseries(1/(z^2+z+1),z,0);
  ```

  sign: argument cannot be imaginary; found %i

  – an error. To debug this try: debugmode(true);

- to get the power series of $\exp(z) \cdot \exp(z_0)$ for $z_0 \neq 0$ as the internal simplifier changes the input into $\exp(z + y)$ before processing,

  ```
  (%i53) powerseries(exp(z)*exp(1),z,0);
  ```

  $$(\%o53) \qquad \left(\sum_{i1=0}^{\infty} \frac{1}{i1!}\right) \cdot \sum_{i1=0}^{\infty} \frac{z^{i1}}{i1!}$$

  ```
  (%i54) powerseries(exp(z+1),z,0);
  ```

  $$(\%o54) \qquad \left(\sum_{i2=0}^{\infty} \frac{1}{i2!}\right) \cdot \sum_{i2=0}^{\infty} \frac{z^{i2}}{i2!}$$

  and the outputs are quite confusing,

- to get the power series of $\arctan(z + z_0)$ for $z_0 \neq 0$ by the lack of an addition formula of the inverse tangent function,

  ```
  (%i56) powerseries(atan(z+1),z,0);
  ```

  $$(\%o56) \qquad \mathrm{powerseries}\left(\mathrm{atan}\left(z+1\right), z, 0\right)$$

```
(%i57) powerseries(atan(z),z,1);
```

$$\text{(\%o57)}\quad \mathrm{powerseries}\left(\mathrm{atan}\left(z\right), z, 1\right)$$

- to solve the problem for products correctly. Usually a product of power series is returned rather than the power series of the product as requested. The Cauchy product rule is not applied.

```
(%i58) cauchysum:true$
       /* Maxima's boolean variable to allow Cauchy product */;
(%i59) powerseries(exp(z)*cos(z),z,0);
```

$$\text{(\%o59)}\quad \left(\sum_{i3=0}^{\infty} \frac{z^{i3}}{i3!}\right) \cdot \sum_{i3=0}^{\infty} \frac{\left(-1\right)^{i3} \cdot z^{2 \cdot i3}}{\left(2 \cdot i3\right)!}$$

```
(%i60) powerseries(exp(z)*log(1+z),z,0);
```

$$\text{(\%o60)}\quad -\left(\sum_{i4=1}^{\infty} \frac{\left(-1\right)^{i4} \cdot z^{i4}}{i4}\right) \cdot \sum_{i4=0}^{\infty} \frac{z^{i4}}{i4!}$$

Chapter 4

Computing Holonomic Differential Equations and Holonomic Recurrence Equations

Let $f(z) = \exp(z) + \cos(z)$. We intend to find a holonomic differential equation (DE) with coefficients in $\mathbb{Q}[z]$ satisfied by $f(z)$ and deduce a holonomic recurrence equation (RE) with coefficients in $\mathbb{Q}[n]$ satisfied by the Taylor coefficients a_n of f [Koepf, 1992].

Searching for a holonomic DE: $f'(z) = \exp(z) - \sin(z)$, and therefore there is no $A_0(z) \in \mathbb{Q}(z)$ such that $f'(z) + A_0(z)f(z) = 0$ because $A_0(z)$ should be $-\dfrac{\exp(z) - \sin(z)}{\exp(z) + \cos(z)}$ which is not rational. Therefore we move to the second order. We search for $A_0(z), A_1(z) \in \mathbb{Q}(z)$ such that

$$f''(z) + A_1(z)f'(z) + A_0(z)f(z) = 0.$$

We write the sum in terms of linearly independent parts and we obtain

$$(1 + A_0(z) + A_1(z))\exp(z) + (A_0(z) - 1)\cos(z) - A_1(z)\sin(z) = 0,$$

and we get the linear system

$$\begin{cases} A_0(z) - 1 = 0 \\ A_1(z) = 0 \\ A_0(z) + A_1(z) + 1 = 0 \end{cases},$$

which has no solution. However, for the third order, the relation

$$f^{(3)}(z) + A_2(z)f^{(2)}(z) + A_1(z)f^{(1)}(z) + A_0(z)f(z) = 0,$$

with $f^{(3)}(z) = \exp(z) + \sin(z)$, leads to a solvable system. By writing the sum in terms of linearly independent parts, we get

$$(1 + A_0(z) + A_1(z) + A_2(z))\exp(z) + (A_0(z) - A_2(z))\cos(z) + (1 - A_1(z))\sin(z) = 0, \tag{4.1}$$

so

$$\begin{cases} 1 - A_1(z) = 0 \\ A_0(z) - A_2(z) = 0 \\ 1 + A_0(z) + A_1(z) + A_2(z) = 0 \end{cases} \iff \begin{cases} A_0(z) = A_2(z) = -1 \\ A_1(z) = 1 \end{cases}. \tag{4.2}$$

Hence the corresponding holonomic DE

$$\frac{d^3}{d\,z^3} \cdot \mathrm{F}\,(z) - \frac{d^2}{d\,z^2} \cdot \mathrm{F}\,(z) + \frac{d}{d\,z} \cdot \mathrm{F}\,(z) - \mathrm{F}\,(z) = 0. \tag{4.3}$$

for $f(z)$ is valid.

Notice that if some of the coefficients found in (4.2) did have polynomials different from 1 as their denominators, then a further step would be the multiplication of the resulting holonomic DE with the least common multiple of the denominators.

Transformation of (4.3) into its corresponding RE: We set

$$f(z) = \sum_{n=0}^{\infty} a_n z^n,$$

so

$$f'(z) = \sum_{n=1}^{\infty} n a_n z^{n-1} = \sum_{n=0}^{\infty} (n+1) a_{n+1} z^n, \tag{4.4}$$

$$f''(z) = \sum_{n=2}^{\infty} n(n-1) a_n z^{n-2} = \sum_{n=0}^{\infty} (n+2)(n+1) a_{n+2} z^n, \tag{4.5}$$

$$f^{(3)}(z) = \sum_{n=3}^{\infty} n(n-1)(n-2) a_n z^{n-3} = \sum_{n=0}^{\infty} (n+3)(n+2)(n+1) a_{n+3} z^n. \tag{4.6}$$

By substitution of these identities in (4.3) for f, we get

$$\begin{aligned} 0 &= f^{(3)}(z) - f''(z) + f'(z) + f(z) \\ &= \sum_{n=0}^{\infty} (n+3)(n+2)(n+1) a_{n+3} z^n - \sum_{n=0}^{\infty} (n+2)(n+1) a_{n+2} z^n \\ &\qquad + \sum_{n=0}^{\infty} (n+1) a_{n+1} z^n - \sum_{n=0}^{\infty} a_n z^n \\ &= \sum_{n=0}^{\infty} \left[(n+3)(n+2)(n+1) a_{n+3} - (n+2)(n+1) a_{n+2} + (n+1) a_{n+1} - a_n \right] z^n, \end{aligned}$$

hence by equating the coefficients we find the holonomic RE

$$(n+3)(n+2)(n+1) a_{n+3} - (n+2)(n+1) a_{n+2} + (n+1) a_{n+1} - a_n = 0,\ n = 0, 1, 2, \dots. \tag{4.7}$$

for a_n.

We have just described how the two first steps of Koepf's algorithm in [Koepf, 1992] apply to $\exp(z) + \cos(z)$. Throughout this chapter, we describe these two steps in the general case. The second step looks identical to its initial description in [Koepf, 1992], however, for the first step, we will use a slightly more efficient algorithm compared to its original version.

4.1 Computing Holonomic Differential Equations

We have a holonomic function f given, and we search for a DE for f of the form (2.76). There are several equations of this type. Indeed, the fact of having a DE allows to get other ones by differentiation. And adding two DE gives another one. For example, if we consider the case of the cosine function $f(z) = \cos(z)$ we have the holonomic DE

$$f''(z) + f(z) = 0, \tag{4.8}$$

which corresponds to the RE $(n+2)(n+1)a_{n+2} + a_n = 0$. And by differentiation of (4.8) we also have $f^{(3)}(z) + f'(z) = 0$ which leads to the same series coefficients but adding the two equations we get $f^{(3)}(z) + f''(z) + f'(z) + f(z) = 0$ which is a DE essentially different as it is equivalent to the RE

$$(n+1)(n+2)(n+3)a_{n+3} + (n+1)(n+2)a_{n+2} + (n+1)a_{n+1} + a_n = 0. \tag{4.9}$$

The algorithm we present here often finds the holonomic DE of lowest order.

4.1.1 Koepf's original algorithm to find holonomic DE

Here we give and prove the initial algorithm as done in ([Koepf, 1992], [Gruntz and Koepf, 1995]). Let $\mathbb{K}$ be a field of characteristic zero.

Algorithm 1 Searching for a holonomic DE of a holonomic function f

Input: A holonomic expression $f(z)$.
Output: Find a holonomic DE with coefficients in $\mathbb{K}(z)$ of least order satisfied by $f(z)$.

1. If $f = 0$ then the DE is found and we stop.
2. $f \neq 0$, compute $A_0(z) = \dfrac{Df(z)}{f(z)}$,

   (1-a) if $A_0(z) \in \mathbb{K}(z)$ i.e $A_0(z) = \dfrac{P(z)}{Q(z)}$ where P and Q are polynomials, then we have found a holonomic DE satisfied by f:

   $$Q(z)Df(z) - P(z)f(z) = 0.$$

   (1-b) If $A_0(z) \notin \mathbb{K}(z)$, then go to 3.
3. Fix a number $N_{max} \in \mathbb{N}$, the maximal order of the DE searched for; a suitable value is $N_{max} := 5$.

   (3-a) set $N := 2$;

   (3-b) compute $D^N f$;

Algorithm 1 Searching for a holonomic DE of a holonomic function f

(3-c) expand the expression

$$D^N f(z) + A_{N-1} D^{N-1} f(z) + \cdots + A_0 f(z) = \sum_{i=0}^{E} S_i,$$

in elementary summands with $A_N, A_{N-1}, \ldots, A_0$ as unknowns. $E \geqslant N$ is the total number of summands S_i obtained after expansion.

(3-d) For each pair of summands S_i and S_j $(0 \leqslant i \neq j \leqslant E)$, group them additively together if there exists $r(z) = \frac{S_i(z)}{S_j(z)} \in \mathbb{K}(z)$. If the number of groups is N then we have N linearly independent expressions. In that case, there exists a solution which can be found by equating each group to zero. The resulting system is linear for the unknowns $A_0, A_1, \ldots, A_{N-1}$. Solving this system gives rational functions in z, and the solution is unique since we normalized $A_N = 1$. After multiplication by the the least common multiple of the denominators of $A_0(z), A_1(z), \ldots A_{N-1}(z)$ we get the holonomic DE searched for. If otherwise the number of groups is larger than N, then there is no solution and the step is not successful.

(3-e) If (3-d) is not successful, then increment N, and go back to (3-b), until $N = N_{max}$.

Assuming that the first step of searching the holonomic DE has failed, we have to show for any integer $N \geqslant 2$ of the algorithm searching for a holonomic DE

$$D^N f(z) + A_{N-1} D^{N-1} f(z) + \cdots + A_0 f(z) = 0 \tag{4.10}$$

for f, that either

1. the number of linearly independent summands of (4.10) equals N and the linear system that we get by setting the coefficients of the linearly independent terms to zero, has a unique solution $(A_0(z), A_1(z), \ldots A_{N-1}(z)) \in \mathbb{K}(z)^N$.

2. or the number of linearly independent summands of (4.10) is larger than N, and there is no solution.

Proof. Since the first step has failed, the number of linearly independent terms is at least 2, and we must proceed with $2 \leqslant N \leqslant N_{\max}$.

Now, for $2 \leqslant N \leqslant N_{\max}$, if we assume that the algorithm searching for a holonomic DE (4.10) has failed until N, then the number of linearly independent terms is at least $N+1$, and we must proceed with $N+1$. In that case, suppose now the number of linearly independent terms is less than or equal to $N+1$. Then we are able to find a solution vector $(A_0, A_1, \cdots, A_N) \in \mathbb{K}(z)^N$, and it remains to show that the solution is unique. Indeed, if we have another solution $(B_0, B_1, \ldots, B_N)$ then f verifies

$$D^{N+1} f + A_N D^N f + \cdots + A_0 f(z) = D^{N+1} f + B_N D^N f + \cdots + B_0 f = 0$$

which implies that the DE

$$(A_N - B_N)D^N f + (A_{N-1} - B_{N-1})D^{N-1} f + \cdots (A_0 - B_0)f = 0$$

is also valid for f. This new DE is of order N but we know from our hypothesis that this is not possible. Hence we must have $A_N = B_N, A_{N-1} = B_{N-1}, \dots A_0 = B_0$.

Therefore by induction we have shown our statement. □

Nevertheless, the above algorithm can be seen in a different way. Indeed, if we consider $f(z) = \cos(z) + \sin(z)$, it is clear that the differential equation that the algorithm will find is a null linear combination of the derivatives of $f(z)$ expanded in the basis $(\cos(z), \sin(z))$. Thus, we can save the time spent by computing all the derivatives up to N in step 3 of **Algorithm 1**, by trying to write each derivative in the same basis.

4.1.2 Second method for computing holonomic DE

Let $(A_0, A_1, \dots, A_{N-1}) \in \mathbb{K}(z)^N$, $N \in \mathbb{N}$ such that an analytic expression f satisfies

$$\mathcal{F}\left(f, f', \dots, f^{(N-1)}, f^{(N)}\right) = f^{(N)} + A_{N-1} \cdot f^{(N-1)} + \cdots + A_1 \cdot f + A_0 f = 0. \qquad (4.11)$$

We consider a basis $(e_1, e_2, \dots, e_l)$ of the linear span of all linearly independent summands over $\mathbb{K}(z)$ that appear in the complete expansions of the derivatives $f, f', \dots, f^N$. For example, assume for $0 \leqslant i \neq j \leqslant N$, that

$$\begin{aligned} f^{(i)} &= e_{i,1} + \cdots + e_{i,k_i}, \\ f^{(j)} &= e_{j,1} + \cdots + e_{j,k_j}, \end{aligned}$$

for some positive integers k_i and k_j, such that $\frac{e_{i,u}}{e_{i,v}} \notin \mathbb{K}(z)$ for all $u, v \in [\![1, k_i]\!]$ and $\frac{e_{j,u}}{e_{j,v}} \notin \mathbb{K}(z)$ for all $u, v \in [\![1, k_j]\!]$. Then for $f^{(i)}$ and $f^{(j)}$ we consider a basis of the linear span of $\{e_{i,1}, \dots, e_{i,k_i}, e_{j,1}, \dots, e_{j,k_j}\}$ which may have less elements since some $e_{i,u}$, $u \in [\![1, k_i]\!]$ and $e_{j,v}$, $v \in [\![1, k_j]\!]$ can be linearly dependent.

Thus each derivative $f^{(j)}, j \in \mathbb{N}_{\geqslant 0}$ $\left(f^{(0)} = f\right)$ can be seen as a vector in the linear space $\langle e_1, e_2, \dots, e_l\rangle$.
Since

$$\mathcal{F}\left(f, f', \dots, f^{(N-1)}, f^{(N)}\right) = 0 \iff -f^{(N)} = A_0 \cdot f + A_1 \cdot f' + \cdots + A_{N-1} \cdot f^{(N-1)}, \quad (4.12)$$

we can write in a matrix representation

$$-f^{(N)} = \left[f, f', \dots, f^{(N-1)}\right]_{(e_1, e_2, \dots, e_l)} (A_0, A_1, \dots, A_{N-1})^T. \qquad (4.13)$$

Therefore, one sees that seeking for a holonomic DE of order N satisfied by a given expression $f(z)$ is equivalent to find a basis in a $\mathbb{K}(z)$-linear space where the system

$$\left(f^{(N)}(z), f^{(N-1)}(z), \dots, f'(z), f(z)\right)$$

is linearly dependent. The idea of the method described in this section is to construct such a basis while computing each derivative of $f(z)$ and their components. Thus, in each iteration N, if all the $N+1$ derivatives are expanded in the same basis, then we try to solve the resulting linear system.

Now let us present what is done in general. Consider an expression $f(z)$ which is not identically zero with l_0 linearly independent sub-terms over $\mathbb{K}(z)$. Then we can write

$$f(z) = f_1(z) + f_2(z) + \cdots + f_{l_0}(z) \tag{4.14}$$

with $\frac{f_i(z)}{f_j(z)} \notin \mathbb{K}(z),\ 1 \leqslant i \neq j \leqslant l_0$. $f(z)$ is seen as a vector in the basis $E_0 = (e_1, e_2, \ldots, e_{l_0})$ where $e_i = f_i$. Then we compute the first derivative of $f(z)$, and we get the following two possibilities:

- either $f'(z)$ is expanded in E_0, which means that there exist $\alpha_{1,i} = \alpha_{1,i}(z) \in \mathbb{K}(z), i = 1, \ldots, l_0$ such that
$$f'(z) = \alpha_{1,1}e_1 + \alpha_{1,2}e_2 + \ldots + \alpha_{1,l_0}e_{l_0}. \tag{4.15}$$
Here in the worst case, $f'(z)$ and $f(z)$ are linearly independent, but from there we know that all the derivatives can be expanded in E_0.

- Or $f'(z)$ is not expanded in E_0, which means that E_0 has to be augmented and there exist $\alpha_{1,i} \in \mathbb{K}(z), i = 1, \ldots, l_0$ and an integer $l_1 > l_0$ such that
$$f'(z) = \alpha_{1,1}e_1 + \alpha_{1,2}e_2 + \ldots + \alpha_{1,l_0}e_{l_0} + e_{l_0+1} + \ldots + e_{l_1}. \tag{4.16}$$
Observe here that the new basis is $E_1 = (e_1, \ldots, e_{l_1})$ with $e_{l_0+1}, \ldots, e_{l_1}$ corresponding to independent terms brought by $f'(z)$. And also $\alpha_{1,i}$, $i \leqslant l_0$ could be zero.

Actually in the first case we may find the DE sought, but in order to present a general overview of the algorithm, let us assume that $f(z)$ satisfies a DE of order $N \geqslant 1$. It is clear that the process will lead to the following representation

$$f(z) = e_1 + \ldots + e_{l_0} \tag{4.17}$$
$$f'(z) = \alpha_{1,1}e_1 + \cdots + \alpha_{1,l_0}e_{l_0} + e_{l_0+1} + \cdots + e_{l_1} \tag{4.18}$$
$$f''(z) = \alpha_{2,1}e_1 + \cdots + \alpha_{2,l_0}e_{l_0} + \alpha_{2,l_0+1}e_{l_0+1} + \cdots + \alpha_{2,l_1}e_{l_1} + e_{l_1+1} + \ldots + e_{l_2} \tag{4.19}$$
$$\cdots \tag{4.20}$$
$$f^{(N-1)}(z) = \alpha_{N-1,1}e_1 + \cdots + \alpha_{N-1,l_{N-2}}e_{l_{N-2}} + e_{l_{N-2}+1} + \cdots + e_{l_{N-1}} \tag{4.21}$$
$$f^{(N)} = \alpha_{N,1}e_1 + \cdots + \alpha_{N,l_{N-1}}e_{l_{N-1}}, \tag{4.22}$$

with positive integers $l_0 \leqslant l_1 \leqslant \ldots \leqslant l_{N-1}$, and $\alpha_{i,j} \in \mathbb{K}(z),\ i = 1, \ldots, N,\ j = 1, \ldots, l_{i-1}$.

Note, however, that only $f^{(N)}(z)$ is computed by differentiating $f^{(N-1)}(z)$. In each step, the algorithm keeps the coefficients $\alpha_{N,i}$, the augmented basis and the current derivative.

It is straightforward to see that the final basis considered is $E_{N-1} = \left(e_1, \ldots, e_{l_{N-1}}\right)$. The algorithm keeps information in a matrix form, say H, and at this step we have

$$H = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 & \cdots & 0 \\ \alpha_{1,1} & \cdots & \alpha_{1,l_0} & 1 & \cdots & 1 & 0 & \cdots & \cdots & 0 & \cdots & 0 \\ \alpha_{2,1} & \cdots & \alpha_{2,l_0} & \cdots & \cdots & \alpha_{2,l_1} & 1 & \cdots & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{N-1,1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \alpha_{N-1,l_{N-2}} & 1 & \cdots & 1 \\ \alpha_{N,1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \alpha_{N,l_{N-2}} & \alpha_{N,l_{N-2}+1} & \cdots & \alpha_{N,l_{N-1}} \end{bmatrix}. \tag{4.23}$$

H is a $(N+1) \times l_{N-1}$ matrix in $\mathbb{K}(z)$, and it contains all information that we need to find the holonomic DE sought. Indeed, one can observe that the coefficients $A_i(z) \in \mathbb{K}(z), i = 0, \ldots, N-1$ in Algorithm 1 constitute the rational components of the unique vector solution of the matrix system

$$Av = b, \tag{4.24}$$

with

$$A = \begin{bmatrix} 1 & \alpha_{1,1} & \alpha_{2,1} & \ldots & \alpha_{N-1,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_{1,l_0} & \alpha_{2,l_0} & \ldots & \alpha_{N-1,l_0} \\ 0 & 1 & \alpha_{2,l_0+1} & \ldots & \alpha_{N-1,l_0+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \tag{4.25}$$

and

$$b = -\begin{bmatrix} \alpha_{N,1} \\ \alpha_{N,2} \\ \vdots \\ \alpha_{N,l_{N-1}} \end{bmatrix}. \tag{4.26}$$

Observe that b is the negative (note the minus in front) of the transpose of the last row of H, and A is the transpose of H deprived of its last row. The above linear system has l_{N-1} linear equations and N unknowns.

Example 4.1.

- $f(z) = \sin(z) + z\cos(z)$. *We have two linearly independent terms over* $\mathbb{Q}(z)$*, and we can write*

  $$f(z) = e_1 + e_2,$$

  with $e_1 = \sin(z)$ *and* $e_2 = z\cos(z)$*. Computing the first derivative, we get*

  $$f'(z) = -z\sin(z) + 2\cos(z) = -z \cdot e_1 + \frac{2}{z} \cdot e_2.$$

At this step we have

$$H = \begin{bmatrix} 1 & 1 \\ -z & \frac{2}{z} \end{bmatrix},$$

and we get the system $\begin{bmatrix} 1 \\ 1 \end{bmatrix} v = \begin{bmatrix} z \\ -\frac{2}{z} \end{bmatrix}$*, which has no solution* $v \in \mathbb{Q}(z)$ *(seen as a one dimensional vector space). Now we compute the second derivative, and we get*

$$f''(z) = -3\sin(z) - z\cos(z) = -3 \cdot e_1 - e_2.$$

H *becomes*

$$H = \begin{bmatrix} 1 & 1 \\ -z & \frac{2}{z} \\ -3 & -1 \end{bmatrix},$$

which gives the system

$$\begin{bmatrix} 1 & -z \\ 1 & \frac{2}{z} \end{bmatrix} v = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \ v \in \mathbb{Q}(z)^2,$$

and we get the solution

$$\left\{ \left(\frac{z^2+6}{z^2+2}, \frac{-2z}{z^2+2} \right) \right\}. \tag{4.27}$$

The differential equation sought is therefore

$$(2+z^2)f''(z) - 2zf'(z) + (6+z^2)f = 0. \tag{4.28}$$

- $f(z) = \arctan(z)$. *We have only one term so* $e_1 = \arctan(z)$. *For the first derivative*

$$f'(z) = \frac{1}{1+z^2} = 0 \cdot e_1 + e_2,$$

where $e_2 = \frac{1}{1+z^2}$*. Since the basis has been augmented there is no system to be solved, and at this step we have*

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The second derivative gives

$$f''(z) = -\frac{2z}{(1+z^2)^2} = 0 \cdot e_1 - \frac{2z}{1+z^2} \cdot e_2,$$

and we get

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -\frac{2z}{1+z^2} \end{bmatrix}$$

which produces the system

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} v = \begin{bmatrix} 0 \\ \frac{2z}{1+z^2} \end{bmatrix}, \ v \in \mathbb{Q}(z)^2.$$

We get $v = \left(0, \frac{2z}{1+z^2}\right)$*, hence the holonomic DE*

$$(z^2+1)f''(z) + 2zf'(z) = 0. \tag{4.29}$$

- $f(z) = \exp(z) + \log(1+z) = e_1 + e_2$, *with* $e_1 = \exp(z)$ *and* $e_2 = \log(1+z)$. *The first derivative yields*
$$f'(z) = \exp(z) + \frac{1}{1+z} = e_1 + 0 \cdot e_2 + e_3,$$
with $e_3 = \dfrac{1}{1+z}$. *Since a new term is added to the basis, the next step is to compute the second derivative*
$$f''(z) = \exp(z) - \frac{1}{(1+z)^2} = e_1 + 0 \cdot e_2 - \frac{1}{(1+z)} \cdot e_3.$$
No term is added to the basis. We try to solve the resulting system. At this stage
$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & -\frac{1}{1+z} \end{bmatrix},$$
and we get the system
$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} v = \begin{bmatrix} -1 \\ 0 \\ \frac{1}{1+z} \end{bmatrix}, \; v \in \mathbb{Q}(z)^2,$$
which has no solution. We move on and compute the third derivative
$$f^{(3)}(z) = \exp(z) + \frac{2}{(1+z)^3} = e_1 + 0 \cdot e_2 + \frac{2}{(1+z)^2} \cdot e_3.$$
Thus
$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & -\frac{1}{1+z} \\ 1 & 0 & \frac{2}{(1+z)^2} \end{bmatrix},$$
and we obtain the system
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & -\frac{1}{1+z} \end{bmatrix} v = \begin{bmatrix} -1 \\ 0 \\ -\frac{2}{(1+z)^2} \end{bmatrix},$$
whose solution in $\mathbb{Q}(z)^3$ *is*
$$\left\{ \left(0, -\frac{z+3}{(z+1)(z+2)}, -\frac{z^2+2z-1}{(z+1)(z+2)} \right) \right\}. \tag{4.30}$$
Therefore we get the holonomic DE
$$(z+1)(z+2)f^{(3)}(z) - (z^2+2z-1)f''(z) - (z+3)f'(z) = 0. \tag{4.31}$$

In our package FPS, we implemented the second method in Maxima as *HolonomicDE(f,F(z))* to compute a holonomic DE with the indeterminate *F(z)* for an expression f of the variable z.

Here are some examples. The package contains a global variable *Nmax* which can be changed in order to look for higher order differential equations.

```
(%i1) HolonomicDE(asin(z),F(z));
```

Evaluation took 0.0000 seconds (0.0100 elapsed)

$$(\%o1)\qquad (z-1)\cdot(1+z)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+z\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)=0$$

```
(%i2) HolonomicDE(cos(z)+exp(z),F(z));
```

Evaluation took 0.0100 seconds (0.0100 elapsed)

$$(\%o2)\qquad \frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)-\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)+\frac{d}{d\,z}\cdot \mathrm{F}(z)-\mathrm{F}(z)=0$$

```
(%i3) HolonomicDE(atan(z),F(z));
```

Evaluation took 0.0000 seconds (0.0100 elapsed)

$$(\%o3)\qquad \left(1+z^2\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+2\cdot z\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)=0$$

```
(%i4) HolonomicDE(exp(asin(z)),F(z));
```

Evaluation took 0.0100 seconds (0.0100 elapsed)

$$(\%o4)\qquad (z-1)\cdot(1+z)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+z\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)+\mathrm{F}(z)=0$$

```
(%i5) HolonomicDE(asin(z)+cos(z),F(z));
```

Evaluation took 0.0400 seconds (0.0500 elapsed)

$$(\%o5)\qquad (z-1)\cdot(1+z)\cdot\left(2+z^4\right)\cdot\left(\frac{d^4}{d\,z^4}\cdot \mathrm{F}(z)\right)+z\cdot\left(10+4\cdot z^2+z^4\right)\cdot\left(\frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)\right)$$
$$+(z-1)\cdot(1+z)\cdot\left(2+z^4\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+z\cdot\left(10+4\cdot z^2+z^4\right)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)=0$$

```
(%i6) HolonomicDE(cos(z)*log(1+z),F(z));
```

Evaluation took 0.0500 seconds (0.0700 elapsed)

$$(\%o6)\qquad (1+z)^2\cdot(1+2\cdot z)\cdot(3+2\cdot z)\cdot\left(\frac{d^4}{d\,z^4}\cdot \mathrm{F}(z)\right)$$
$$+4\cdot(1+z)\cdot\left(1+4\cdot z+2\cdot z^2\right)\cdot\left(\frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)\right)+2\cdot z\cdot(2+z)\cdot\left(5+8\cdot z+4\cdot z^2\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)$$
$$+4\cdot(1+z)\cdot\left(1+4\cdot z+2\cdot z^2\right)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)+\left(-3+6\cdot z+19\cdot z^2+16\cdot z^3+4\cdot z^4\right)\cdot \mathrm{F}(z)=0$$

```
(%i7) HolonomicDE(sin(z)^4*asin(z),F(z));
```

Evaluation took 0.5700 seconds (0.6600 elapsed)

$$(\%o7) \qquad \text{false}$$

```
(%i8) Nmax:10$
```

Evaluation took 0.0000 seconds (0.0000 elapsed)

```
(%i9) HolonomicDE(sin(z)^4*asin(z),F(z))$
```

Evaluation took 4.2000 seconds (5.0500 elapsed)

The latter is a big differential equation of order $10 >$*Nmax*, that is why the value of *Nmax* was changed to 10. Our first Maxima implementation which was using the first method takes about 7 seconds in this example. The timings of both algorithms get closer as Nmax becomes large. This is due to the use of memory for the second algorithm, and it also depends on the given expression. However the second approach is more efficient for our goal of computing power series as we have fixed the maximum order of differential equations sought Nmax to 5. One could increase *Nmax* as wanted, but generally this is not needed.

4.2 Computing Holonomic Recurrence Equations

We have seen in Theorem 2.4 that any hypergeometric type series

$$g(z) = \sum_{n=n_0}^{\infty} a_n z^{n/k}, \tag{4.32}$$

satisfies a homogeneous differential equation with polynomial coefficients. So after substituting the power series representation of $g(z)$, the general power of the indeterminate z is shifted by an integer power. Therefore the operations used to compute the recurrence equation of the general coefficient a_n from a given holonomic differential equation are identical to those used for the power series

$$f(z) = \sum_{n=0}^{\infty} a_n z^n. \tag{4.33}$$

In this section, we establish the rewrite rule to compute a holonomic recurrence equation from a holonomic differential equation by assuming $f(z)$ as in (4.33). We will see in Chapter 8 that applying this rule for $g(z)$ in (4.32) implicitly gives candidates for the Puiseux number k, which is an important step to compute Puiseux series.

Let $f(z)$ be as in (4.33), we can find a map which allows us to go from each term of the expansion of a DE to the term of the RE (see [Koepf, 1992, Section 6]). Firstly it is easy to see that

$$(a_n z^n)^{(j)} = n(n-1)\cdots(n-j+1)a_n z^{n-j} \tag{4.34}$$
$$= (n+1-j)_j \cdot a_n z^{n-j}, \tag{4.35}$$

hence if we multiply by z^l, we obtain

$$z^l\,(a_n z^n)^{(j)} = (n+1-j)_j \cdot a_n z^{n-j+l}. \tag{4.36}$$

Shifting the index by setting $i = n - j + l \Rightarrow n = k + j - l$, we obtain the correspondence

$$z^l\,(a_n z^n)^{(j)} \longrightarrow (i+1-l)_j \cdot a_{i+j-l} z^i, \tag{4.37}$$

which allows us to consider the following rewrite rule between the summands of a DE and those of its corresponding RE

$$z^l f^{(j)} \longrightarrow (n+1-l)_j \cdot a_{n+j-l}. \tag{4.38}$$

For example in the case of $f(z) = \arccos(z)$, our Maxima procedure *HolonomicDE* gives

```
(%i1) HolonomicDE(acos(z),F(z));
```

$$(\%o1) \qquad (z-1)\cdot(1+z)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right) + z\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right) = 0,$$

which after expansion can also be written as

$$f''(z) - z^2 f''(z) - z f'(z) = 0.$$

Thus,

$$f'' = z^0 f^{(2)} \longrightarrow (n+1-0)_2 \cdot a_{n+2-0} = (n+1)(n+2)a_{n+2} \tag{4.39}$$
$$z^2 f^{(2)} \longrightarrow (n+1-2)_2 \cdot a_{n+2-2} = (n-1)na_n \tag{4.40}$$
$$z f^{(1)} \longrightarrow (n+1-1)_1 \cdot a_{n+1-1} = na_n. \tag{4.41}$$

Finally taking the linear combination we get the holonomic recurrence equation

$$n^2 \cdot a_n - (1+n)\cdot(2+n)\cdot a_{n+2} = 0. \tag{4.42}$$

The following algorithm combines these steps in the general case.

Algorithm 2 From a holonomic DE to its Recurrence Equation.

Input: A holonomic differential equation DE.
Output: Conversion of DE into a holonomic recurrence equation.

(a) Expand the DE to write it in the form

$$\sum_{j=0}^{s}\sum_{l=0}^{s} c_{j,l} z^l D^j f = 0 \tag{4.43}$$

where $s \in \mathbb{N}$ and $c_{j,l}$ are some constants.

(b) Use the rewrite rule

$$z^l D^j f \longrightarrow (n+1-l)_j \cdot a_{n+j-l}. \tag{4.44}$$

to substitute each term. Therefore we get

$$\sum_{j=0}^{s}\sum_{j=0}^{s} c_{j,l}(n+1-l)_j \cdot a_{n+j-l} = 0, \tag{4.45}$$

and finally the holonomic RE can be brought in a particularly nice form after factorizing the coefficients.

Our package contains the function *DEtoRE(DE,F(z),a[n])* which converts the holonomic differential equation DE depending on the variable *z* into its corresponding recurrence equation for the coefficients *a[n]*.

```
rectermfun(term,F,a):=block([z,n,j,mterm,zpow,coef],
    z: first(F),
    n: first(a),
    j: derivdegree(term,F,z),
    mterm: xthru(term/'diff(F,z,j)),
    zpow: hipow(mterm,z),
    coef: coeff(mterm,z,zpow),
    coef*pochhammer(n+1-zpow, j)*subst(n+j-zpow,n,a)
)$

DEtoRE(DE,F,a):= block([de,terms,re,RE,aterm,i],
    de: lhs(expand(DE)),
    if(freeof("+",de)) then terms: [de]
    else terms: args(de),
    rec: map(lambda([v],rectermfun(v,F,a)),terms),
    RE: apply("+",rec),
    aterm: sublist(listofvars(RE), lambda([v], not atom(v))),
    aterm: sort(sublist(aterm, lambda([v], is(equal(op(a),op(v)))))),
    if(length(rec)<2) then RE=0
    else (
```

```
        for i: 1 thru length(aterm) do RE: map(factor, \
                                            collectterms(RE,aterm[i])),
        RE=0
    )
)$
```

Example 4.2.

```
(%i2) DE:HolonomicDE(asin(z),F(z));
```

$$(\%o2)\qquad (z-1)\cdot(1+z)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+z\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)=0$$

```
(%i3) DEtoRE(DE,F(z),a[n]);
```

$$(\%o3)\qquad n^2\cdot a_n-(1+n)\cdot(2+n)\cdot a_{n+2}=0$$

```
(%i4) DE:HolonomicDE(exp(z)+log(1+z),F(z));
```

$$(\%o4)\qquad (1+z)\cdot(2+z)\cdot\left(\frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)\right) - \left(-1+2\cdot z+z^2\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)-(3+z)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)=0$$

```
(%i5) DEtoRE(DE,F(z),a[n]);
```

$$(\%o5)\qquad 2\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}+(1+n)\cdot(2+n)\cdot(1+3\cdot n)\cdot a_{n+2} + (1+n)\cdot\left(-3-3\cdot n+n^2\right)\cdot a_{n+1}-n^2\cdot a_n=0$$

Our package contains another Maxima function *FindRE(f,z,a[n])* which calls our functions *HolonomicDE(f,F(z))* and *DEtoRE(DE,z,n)* to produce a recurrence equation for the Taylor coefficients a_n of a given holonomic expression f.

```
(%i6) FindRE(cos(z)+sin(z),z,a[n]);
```

$$(\%o6)\qquad (1+n)\cdot(2+n)\cdot a_{n+2}+a_n=0$$

```
(%i7) FindRE(exp(z)+atan(z),z,a[n]);
```

$$(\%o7)\qquad (1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}+(n-3)\cdot(1+n)\cdot(2+n)\cdot a_{n+2} + (1+n)\cdot\left(2+2\cdot n+n^2\right)\cdot a_{n+1}+(n-3)\cdot n\cdot(1+n)\cdot a_n-(n-1)\cdot n\cdot a_{n-1}=0$$

```
(%i8) FindRE(cosh(z)^2+log(1+z),z,a[n]);
```

$$(\%o8)\quad (1+n)\cdot(2+n)\cdot(3+n)\cdot(4+n)\cdot a_{n+4}+(1+n)\cdot(2+n)\cdot(3+n)\cdot(5\cdot n-1)\cdot a_{n+3}$$
$$+2\cdot(n-1)\cdot(1+n)\cdot(2+n)\cdot(2+3\cdot n)\cdot a_{n+2}+2\cdot(1+n)\cdot\left(2-9\cdot n-2\cdot n^2+n^3\right)\cdot a_{n+1}$$
$$-8\cdot n\cdot(3\cdot n-1)\cdot a_n-8\cdot(n-1)^2\cdot a_{n-1}=0$$

```
(%i9) FindRE(exp(z^(1/3)),z,a[n]);
```

$$(\%o9)\quad 3\cdot(1+n)\cdot(1+3\cdot n)\cdot(2+3\cdot n)\cdot a_{n+1}-a_n=0$$

4.3 Fast Computation of Taylor Expansions of Holonomic Functions

Hypergeometric type functions are strictly contained in the family of holonomic functions. Indeed, it is proved that linear combinations and products of holonomic functions are also holonomic ([Koepf, 1997], [Stanley, 1980]). Although power series expansions of linear combinations of hypergeometric type functions remain accessible through the use of an algorithm that finds all m-fold hypergeometric term solutions of a holonomic RE, it is not generally the case with their products. Thus, it is clear that our m-fold hypergeometric algorithm cannot find explicit formulas for the coefficients of power series expansions of certain holonomic functions. Nevertheless, as we are able to find recurrence equations for the general coefficients, the use of enough initial values coupled with their corresponding holonomic RE's uniquely characterizes their Taylor coefficients in a certain neighborhood. It is thanks to this observation that Koepf proceeded in computing Taylor polynomials of holonomic expressions by using the output of Algorithm 2 (see [Koepf, 2006, Chapter 10]). In this section, using *FindRE*, we develop an algorithm to compute Taylor polynomials of holonomic functions and compare the result with Maxima's internal command *taylor*. First, we give some particular normal forms for holonomic functions.

4.3.1 On Normal Forms of Holonomic Functions

By an application of the well known Cauchy-Lipschitz (also called Picard-Lindelöf) theorem (see [Teschl, 2012, Theorem 2.2]) for uniqueness, the holonomic differential equation of lowest order and enough initial values corresponding to a holonomic function can be used for identification purposes. Therefore, such a representation constitutes a normal form (see [Geddes et al., 1992, Chapter 3]).

Thus for example, one can use the differential equation

```
(%i1) HolonomicDE(log(1+z),F(z));
```

$$(\%o1) \qquad (1+z)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+\frac{d}{d\,z}\cdot \mathrm{F}(z)=0$$

for $|z-1|<1$, and the initial values $\mathrm{F}(0)=0$, $\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)(0)=1$ to define the logarithm function. Note that the use of Algorithm 1 reduces this normal form definition of functions to expressions, because it might happen that equivalent expressions have two different representations. This is for example the case with the Chebyshev polynomials. For $|x|<1$, the following two differential equations define the same function.

```
(%i2) HolonomicDE(cos(4*acos(x)),F(x));
```

$$(\%o2) \qquad (x-1)\cdot(1+x)\cdot\left(\frac{d^2}{d\,x^2}\cdot \mathrm{F}(x)\right)+x\cdot\left(\frac{d}{d\,x}\cdot \mathrm{F}(x)\right)-16\cdot \mathrm{F}(x)=0$$

```
(%i3) HolonomicDE(8*x^4-8*x^2+1,F(x));
```

$$(\%o3) \qquad \left(1-8\cdot x^2+8\cdot x^4\right)\cdot\left(\frac{d}{d\,x}\cdot \mathrm{F}(x)\right)-16\cdot x\cdot\left(2\cdot x^2-1\right)\cdot \mathrm{F}(x)=0$$

Note that this happens because *HolonomicDE* does not use simplifications on its input expressions. However, one can easily prove that these two differential equations are compatible by substituting the lower order differential equation into the larger one. The following sequence of Maxima instructions demonstrates their compatibility[1].

```
-->   /* Computatations of the holonomic DEs */

(%i4) DE1: HolonomicDE(cos(4*acos(x)),F(x))$
(%i5) DE2: HolonomicDE(8*x^4-8*x^2+1,F(x))$
-->   /* Writing the first derivative diff(F(x),x)
      in  terms of F(x) */;

(%i6) subst_rule1: solve(DE2,diff(F(x),x))[1];
```

$$(\%o6) \qquad \frac{d}{d\,x}\cdot \mathrm{F}(x)=\frac{\left(32\cdot x^3-16\cdot x\right)\cdot \mathrm{F}(x)}{8\cdot x^4-8\cdot x^2+1}$$

```
-->   /* Differentiating the obtained relation
      to find a relation for the second derivative*/;

(%i7) subst_rule2: diff(subst_rule1,x);
```

$$(\%o7) \qquad \frac{d^2}{d\,x^2}\cdot \mathrm{F}(x)=\frac{\left(32\cdot x^3-16\cdot x\right)\cdot\left(\frac{d}{d\,x}\cdot \mathrm{F}(x)\right)}{8\cdot x^4-8\cdot x^2+1} +\frac{\left(96\cdot x^2-16\right)\cdot \mathrm{F}(x)}{8\cdot x^4-8\cdot x^2+1}-\frac{\left(32\cdot x^3-16\cdot x\right)^2\cdot \mathrm{F}(x)}{\left(1-8\cdot x^2+8\cdot x^4\right)^2}$$

[1] Two differential equations are said to be compatible if every solution of the lower order DE is solution of the other.

```
-->     /* Use the second relation as a rule for
        substitution in DE1  */;

(%i8) DE: subst(subst_rule2,lhs(DE1));
```

$$(\%o8) \quad (x-1)\cdot(1+x)\cdot\left(-\frac{\left(32\cdot x^3-16\cdot x\right)^2\cdot \mathrm{F}(x)}{\left(1-8\cdot x^2+8\cdot x^4\right)^2}+\frac{\left(96\cdot x^2-16\right)\cdot \mathrm{F}(x)}{8\cdot x^4-8\cdot x^2+1} + \frac{\left(32\cdot x^3-16\cdot x\right)\cdot\left(\frac{d}{d\,x}\cdot \mathrm{F}(x)\right)}{8\cdot x^4-8\cdot x^2+1}\right)+x\cdot\left(\frac{d}{d\,x}\cdot \mathrm{F}(x)\right)-16\cdot \mathrm{F}(x)$$

```
-->     /* Use the first relation as a rule for
        substitution in DE */;

(%i9) DE: subst(subst_rule1,DE);
```

$$(\%o9) \quad \frac{x\cdot\left(32\cdot x^3-16\cdot x\right)\cdot \mathrm{F}(x)}{8\cdot x^4-8\cdot x^2+1}+\frac{(x-1)\cdot(1+x)\cdot\left(96\cdot x^2-16\right)\cdot \mathrm{F}(x)}{8\cdot x^4-8\cdot x^2+1}-16\cdot \mathrm{F}(x)$$

```
-->     /* Finally, normalizing DE yields 0. We only have
        to factorize the coefficient of F(x) in DE*/;

(%i10) factor(coeff(DE,F(x)));
```

$$(\%o10) \quad 0$$

This process of deciding whether two holonomic differential equations are compatible can be generalized. Our package contains the function *CompatibleDE(DE1,DE2,F(z))* that can be use for this purpose.

It is remarkable that, as we will see how our algorithm works in the general case, such identities were already recovered by Koepf's original FPS implementation in Maple!

On the other hand, by Proposition 2.3 we can also use power series representations of the form $\sum_{n=n_0}^{\infty} a_n(z-z_0)^n$ with a recursive definition of the general coefficient a_n to identify analytic holonomic functions. Thus, given an analytic expression $f(z)$ at z_0 whose Taylor coefficients satisfy a holonomic recurrence equation of the form

$$T_d(n)\cdot a_{n+d}+T_{d-1}(n)\cdot a_{n+d-1}+\cdots+T_0(n)\cdot a_n=0,\ n\in\mathbb{Z},\ d\in\mathbb{N} \tag{4.46}$$

with $T_0(n)\cdot T_d(n)\neq 0, \forall n\geqslant n_0$, $f(z)$ is identified to

$$\sum_{n=0}^{\infty} a_{n+n_0}(z-z_0)^{n+n_0}, \ \text{ with } \begin{cases} a_{n+d}=\dfrac{T_{d-1}(n)\cdot a_{n+d-1}+\ldots+T_0(n)\cdot a_n}{T_d(n)}, \ n\geqslant n_0 \\ a_j=\lim_{z\to z_0}\dfrac{\left(\frac{d^j}{d\,z^j}\cdot f\right)(z)}{j!}, \ j=n_0,n_0+1,\ldots,n_0+d-1 \end{cases} . \tag{4.47}$$

The value of n_0 is deduced by the property $T_0(n) \cdot T_d(n) \neq 0, \forall n \geqslant n_0$, that is

$$n_0 = 1 + \max\{n \in \mathbb{Z},\ T_0(n) \cdot T_d(n) = 0\}. \tag{4.48}$$

Notice that n_0 is computed before any cancellation of common factors in (4.46), which guaranties that n_0 does exist in general for any holonomic RE output of *FindRE* of order $d \geqslant 1$. Indeed, the rewrite rule (4.44) allows to remark that the differential equation terms with derivative order greater than 1 lead to recurrence equation terms with non constant polynomial coefficients. The determination of n_0 is crucial to extract parts in the series expansion that are not involved in the summation formula. Much details about the computation of power series extra parts are given in Section **7.1**.

Let us consider the logarithm function $\log(1+z)$. It can be seen in the neighborhood of zero as $\sum_{n=0}^{\infty} a_n z^n$, where a_n satisfies

```
(%i11) FindRE(log(1+z),z,a[n]);
```

$$(\%o11) \qquad (1+n) \cdot (2+n) \cdot a_{n+2} + (1+n)^2 \cdot a_{n+1} = 0,$$

with $a_0 = 0,\ a_1 = 1$. By shifting the order so that the equation writes in the form (4.46), we obtain $n_0 = 1$ and therefore,

$$\log(1+z) = \sum_{n=0}^{\infty} a_{n+1} z^{n+1}, \quad \text{with } \begin{cases} a_{n+1} = \dfrac{n}{(n+1)} \cdot a_n,\ n \geqslant 1 \\ a_1 = 1 \end{cases}, \ |z| < 1. \tag{4.49}$$

This representation allows to generate any number of Taylor coefficients as desired.

4.3.2 Taylor Expansions of Holonomic Functions

Let $f(z)$ be a holonomic function. The Taylor expansion of $f(z)$ at z_0 is computed as the one of $g(z) = f(z+z_0)$ at 0 if z_0 is a constant, of $g(z) = f\left(-\frac{1}{z}\right)$ if $z_0 = -\infty$, and of $g(z) = f\left(\frac{1}{z}\right)$ if $z_0 = \infty$. This algorithm is an immediate use of (4.47) whose the steps are as follows.

Algorithm 3 Computing Taylor Polynomials of Holonomic Functions at $z_0 \in \mathbb{C} \cup \{-\infty, \infty\}$

Input: A holonomic expression $f(z)$, a point z_0, and an integer N.
Output: Taylor polynomial of order N of $f(z)$.

1. If $z_0 \in \mathbb{C}$, set $g(z) := f(z+z_0)$, else if $z_0 = -\infty$, set $g(z) := f\left(-\frac{1}{z}\right)$, else set $g(z) := f\left(\frac{1}{z}\right)$.
2. Use *FindRE* to compute a holonomic recurrence equation satisfied by the Taylor coefficients of $g(z)$ and write it in the form

$$T_d(n) \cdot a_{n+d} + T_{d-1}(n) \cdot a_{n+d-1} + \cdots + T_0(n) \cdot a_n = 0,\ n \in \mathbb{Z},\ d \in \mathbb{N}. \tag{4.50}$$

Algorithm 3 Computing Taylor Polynomials of Holonomic Functions at $z_0 \in \mathbb{C} \cup \{-\infty, \infty\}$

3. If $d = 0$ then return the Taylor expansion of order N with the internal command of Taylor expansions, say $taylor(f(z), z_0, N)$.
4. Compute
$$n_0 = 1 + \max\{n \in \mathbb{Z},\ T_0(n) \cdot T_d(n) = 0\}. \tag{4.51}$$
5. If $N \leqslant n_0 + d - 1$ then stop and return $taylor(f(z), z_0, N)$.
6. If $N > n_0 + d - 1$ then
$$\mathcal{T} := taylor(f(z), z_0, n_0 + d - 1). \tag{4.52}$$

1-1-1 If $z_0 \in \mathbb{C}$ then compute
$$a_j := \text{coeff}\,(T, z - z_0, j)\,,\ j = n_0, n_0 + 1, \ldots, n_0 + d - 1, \tag{4.53}$$
where coeff$(T, z - z_0, j)$ collects the coefficient of $(z - z_0)^j$ in T. The Maxima syntax is adopted.

1-1-2 For $j = n_0, n_0 + 1, \ldots, N - d$, compute
$$a_{j+d} = \frac{T_{d-1}(j) \cdot a_{j+d-1} + \ldots + T_0(j) \cdot a_j}{T_d(j)} \tag{4.54}$$
$$\mathcal{T} = \mathcal{T} + a_{j+d} \cdot (z - z_0)^{j+d} \tag{4.55}$$

1-2-1 If $|z_0| = \infty$ then compute
$$a_j := \text{coeff}\,(T, 1/z, j)\,,\ j = n_0, n_0 + 1, \ldots, n_0 + d - 1, \tag{4.56}$$

1-2-2 For $j = n_0, n_0 + 1, \ldots, N - d$, compute
$$a_{j+d} = \frac{T_{d-1}(j) \cdot a_{j+d-1} + \ldots + T_0(j) \cdot a_j}{T_d(j)} \tag{4.57}$$
$$\mathcal{T} = \mathcal{T} + a_{j+d} \cdot \left(\frac{1}{z}\right)^{j+d} \tag{4.58}$$

- Return $\mathcal{T}$.

Remark

- The relation (4.54) shows that the Taylor coefficients are computed in the same finite number of operations. Therefore the complexity is linear.
- As we are interested by the asymptotic complexity of this algorithm, there is no issue of comparison when the internal command is called in step 1 and 1. And moreover, the Maxima's $taylor(f(z), z_0, N)$ is generally either 0 or a term not analytic at the point of expansion involved in the power series expansion of $f(z)$. An example is $\text{arcsech}(z)$ whose Maxima's command *taylor* gives the following expansion of order 4 at 0.

(%i12) taylor(asech(z),z,0,4);

$$(\%o12)/T/\ -\log(z)+\log(2)+\cdots-\frac{z^2}{4}-\frac{3\cdot z^4}{32}+\cdots$$

- In order to treat certain interesting non analytic cases like $\operatorname{arcsech}(z)$ in Maxima, instead of the limit command which can generate error due to singularities, the internal Maxima command *taylor* is used. The initial values are then the coefficients of $(z-z_0)^j$, $j = n_0, \ldots, n_0+d-1$.

Our Maxima package FPS contains an implementation of Algorithm 3 named *Taylor* with the same syntax as the internal command *taylor*. The following Maxima program can be used to collect polynomial coefficients as in (4.50) of a recurrence equation found by *FindRE*.

```
allcoeffsBound(P,n,t,M):= block([exP,c,j],
     c: [],
     exP: expand(P),
     for j:t thru M do (c : cons(coeff(P,n,j),c)),
     c
)$
```

Example 4.3.

(%i13) RE:FindRE(cos(z),z,a[n]);

$$(\%o13)\qquad (1+n)\cdot(2+n)\cdot a_{n+2}+a_n=0$$

(%i14) REcoeff(RE,a[n]);

$$(\%o14)\qquad [1,0,(n+1)\cdot(n+2)]$$

(%i15) RE:FindRE(sin(z)+exp(z),z,a[n]);

$$(\%o15)\qquad (1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}-(1+n)\cdot(2+n)\cdot a_{n+2}+(1+n)\cdot a_{n+1}-a_n=0$$

(%i16) REcoeff(RE,a[n]);

$$(\%o16)\qquad [-1,n+1,-(1+n)\cdot(2+n),(n+1)\cdot(n+2)\cdot(n+3)]$$

The code *REcoeff* will often be used to collect polynomial coefficients.

Finally, our *Taylor* code is given by

```
Taylor(f,z,z0,N):=catch(block([g,limitz0,RE,d,C,Sn0,n0,T,I,rec,a,j],
    if(z0=inf or z0=-inf) then g: subst(Sign(z0)*1/z,z,f)
    else g: subst(z+z0,z,f),
    assume(z>0),
    errormsg: false,
    limitz0: errcatch(taylor(g,z,0,lopow(f,z))),
    if(length(limitz0)<1) then (
        print("Not analytic at", z0),
        throw(f)
    ),
    errormsg: true,
    RE: FindRE(g,z,a[n]),
    if(RE=false) then throw(false),
    C: REcoeff(RE,a[n]),
    d: length(C)-1,
    if(d>0) then Sn0: map(rhs, solve(first(C)*last(C),n))
    else throw(ratdisrep(taylor(f,z,z0,N))),
    Sn0: sublist(Sn0, integerp),
    n0: lmax(Sn0)+1,
    if(N<=n0+d-1) then throw(ratdisrep(taylor(f,z,z0,N))),
    if(z0=inf or z0=-inf) then (
        T: ratdisrep(taylor(f,z,z0,n0+d-1)),
        I: makelist(a[j]=coeff(T,1/z,j),j,n0,n0+d-1),
        rec: a[n+d]=-sum(C[j]*a[n+j-1],j,1,d)/C[d+1],
        for j:n0 thru N-d do
            I: endcons(radcan(subst(I,subst(j,n,rec))),I),
        T + sum(rhs(I[j])*(1/z)^(n0+j-1),j,d+1,N+1-n0)
    )
    else(
        T: ratdisrep(taylor(f,z,z0,n0+d-1)),
        I: makelist(a[j]=coeff(T,z-z0,j),j,n0,n0+d-1),
        rec: a[n+d]=-sum(C[j]*a[n+j-1],j,1,d)/C[d+1],
        for j:n0 thru N-d do
            I: endcons(radcan(subst(I,subst(j,n,rec))),I),
        T + sum(rhs(I[j])*(z-z0)^(n0+j-1),j,d+1,N+1-n0)
    )
))$
```

Example 4.4.

```
(%i17) Taylor(asech(z),z,0,7);
```

$$(\%o17) \qquad -\log(z) - \frac{5 \cdot z^6}{96} - \frac{3 \cdot z^4}{32} - \frac{z^2}{4} + \log(2)$$

```
(%i18) Taylor(atan(z),z,0,7);
```

$$(\%o18) \qquad -\frac{z^7}{7}+\frac{z^5}{5}-\frac{z^3}{3}+z$$

```
(%i19) Taylor(cos(z),z,0,8);
```

$$(\%o19) \qquad \frac{z^8}{40320}-\frac{z^6}{720}+\frac{z^4}{24}-\frac{z^2}{2}+1$$

```
(%i20) Taylor(log(1+z)+sin(z),z,1,7);
```

$$(\%o20) \qquad \frac{(1+2\cdot\cos(1))\cdot(z-1)}{2}-\frac{(8\cdot\cos(1)-45)\cdot(z-1)^7}{40320} -\frac{(15+8\cdot\sin(1))\cdot(z-1)^6}{5760}+\frac{(3+4\cdot\cos(1))\cdot(z-1)^5}{480}+\frac{(8\cdot\sin(1)-3)\cdot(z-1)^4}{192} -\frac{(4\cdot\cos(1)-1)\cdot(z-1)^3}{24}-\frac{(1+4\cdot\sin(1))\cdot(z-1)^2}{8}+\log(2)+\sin(1)$$

```
(%i21) Taylor(atan(z),z,inf,7);
```

$$(\%o21) \qquad -\frac{1}{z}+\frac{1}{3\cdot z^3}-\frac{1}{5\cdot z^5}+\frac{1}{7\cdot z^7}+\frac{\pi}{2}$$

Let us now evaluate the timings for larger order. We mention that when the given expression is a classical one like $\sin(z), \arctan(z), \exp(z)$, *etc*, Maxima seems to use the power series formula and has good asymptotic timings. Therefore, for tests we rather use expressions for which the internal Maxima command *powerseries* cannot find the power series formulas.

```
(%i22) Taylor(exp(z)*cos(z),z,0,300)$
```

Evaluation took 0.2400 seconds (0.2700 elapsed)

```
(%i23) taylor(exp(z)*cos(z),z,0,300)$
```

Evaluation took 0.4500 seconds (0.4900 elapsed)

```
(%i24) Taylor(atan(z)*exp(z),z,0,300)$
```

Evaluation took 0.5200 seconds (0.6200 elapsed)

```
(%i25) taylor(atan(z)*exp(z),z,0,300)$
```

Evaluation took 2.9000 seconds (3.0900 elapsed)

We have used $N=300$ *but note that the time gap between the two computations increases as* $N\to\infty$.

```
(%i26) Taylor(atan(z)*exp(z),z,0,1000)$
```

Evaluation took 7.7100 seconds (11.3600 elapsed)

```
(%i27) taylor(atan(z)*exp(z),z,0,1000)$
```

Evaluation took 186.0100 seconds (191.2100 elapsed)

Hence the fast computation of Taylor expansions of holonomic expressions using the output of FindRE.

Chapter 5

Petkovšek's Algorithm

When the recurrence equation obtained using *FindRE* is a two-term recurrence relation of order $d \in \mathbb{N}$, then one finds a hypergeometric term representation for the Taylor coefficients of the holonomic function considered using d initial values. However, in many cases the recurrence equation found is not of lowest order possible. And this does not happen only when the given holonomic function is a linear combination of hypergeometric type functions, but also when it has a single type in its power series expansion. For example

```
(%i1) FindRE(sqrt(1+z)+1/sqrt(1+z),z,a[n]);
```

$$(\%o1) \qquad 4\cdot(1+n)\cdot a_{n+1} + 6\cdot n\cdot a_n + (2\cdot n-3)\cdot a_{n-1} = 0$$

is the recurrence equation found for the hypergeometric type series

$$\sum_{n=0}^{\infty} \frac{2\cdot(n-1)\cdot(-1)^n\cdot(2\cdot n)!\cdot z^n}{(2\cdot n-1)\cdot 4^n\cdot n!^2} \tag{5.1}$$

whose general coefficient clearly satisfies a two-term recurrence relation of order 1. This figures out the connection between power series computation and hypergeometric and further m-fold hypergeometric term solutions of holonomic recurrence equations. Marko Petkovšek developed an algorithm (see [Petkovšek, 1992]) which finds all hypergeometric term solutions of such equations. The understanding of this algorithm is essential for the remaining part of this thesis. We consider the holonomic recurrence equation

$$P_d(n)a_{n+d} + P_{d-1}(n)a_{n+d-1} + \cdots P_1(n)a_{n+1} + P_0(n)a_n = 0, \tag{5.2}$$

where $P_j(n) \in \mathbb{K}[n], 0 \leqslant j \leqslant d$ and $d \in \mathbb{N}$ is the order. We denote by the maximum degree of the polynomial coefficients

$$M = \max\{\deg(P_j),\ 0 \leqslant j \leqslant d\}, \tag{5.3}$$

the degree of the holonomic recurrence equation (5.2). Note that substituting polynomials by rational functions in (5.2) does not affect the procedure, because for the recurrence

$$r_d(n)a_{n+d} + r_{d-1}(n)a_{n+d-1} + \cdots r_1(n)a_{n+1} + r_0(n)a_n = 0, \tag{5.4}$$

where $r_j(n) = \frac{P_j(n)}{Q_j(n)}$, $0 \leqslant j \leqslant d$, for polynomials $P_j(n)$ and $Q_j(n) \neq 0$, multiplying both sides of (5.4) by $\operatorname{lcm}(Q_0(n), \ldots, Q_d(n))$ leads to an equivalent recurrence equation which is holonomic.

Petkovšek's algorithm comes as a decision procedure to find hypergeometric term solutions of any holonomic recurrence equation. The algorithm has two parts:

- In the first step, Petkovšek gives an algorithm to find all polynomial solutions of a given holonomic recurrence equation.

- In the second part, this sub-algorithm is used to determine the hypergeometric term solutions of a given holonomic recurrence equation.

5.1 Polynomial Solutions of Holonomic Recurrence Equations

Let $\mathbb{K}$ be a field of characteristic zero. In this section, we describe an algorithm which finds polynomial solutions $a_n \in \mathbb{K}[n]$ of (5.2). Abramov has independently proposed a similar algorithm in [Abramov, 1989] which uses the difference operators of linear recurrences to compute their polynomial solutions. Petkovšek's approach is slightly more different.

As we want to compute polynomial solutions of (5.2) in $\mathbb{K}[n]$, the purpose can be reduced by computing the degrees of those polynomials so that for each computed degree N, we can substitute a generic polynomial of degree N for a_n in the recurrence equation and solve the resulting system of linear equations by equating the coefficients. Moreover, consider two integers N_1 and N_2 as the degrees of two polynomial solutions of (5.2), say P_1 and P_2, respectively. If we assume $N_2 > N_1$, then the linear system obtained by substituting a generic polynomial of degree N_2 in (5.2) also gives solutions for which P_1 is a particular case. That are polynomials whose monomial terms of degrees $N_1 + 1, \ldots, N_2$ are zero. Therefore, it is natural to think of an upper bound N for the possible degrees of polynomial solutions of (5.2).

Note that information on the degree bound of polynomial solutions of holonomic recurrence equations may appear hidden from their orders and degrees. Consider for example the holonomic recurrence equation of the general coefficient of the power series of $\frac{1}{(1-z)^k}$, $k \in \mathbb{N}$. Using *FindRE* we get the holonomic RE

```
(%i1) FindRE(1/(1-z)^k,z,a[n]);
```

$$(\%o1) \qquad (k+n)\cdot a_n - (1+n)\cdot a_{n+1} = 0$$

of degree and order equal 1 whose polynomial solutions are multiples of $\binom{n+k-1}{k-1}$ which is a polynomial of degree k as seen in (2.6).

Petkovšek's main observation is that information on the bound of the degree of polynomial solutions of (5.2) can be found by using the binomial theorem in the expansion of

$$\begin{aligned} P_d(n)\left(\alpha_N (n+d)^N + \cdots + \alpha_0\right) + P_{d-1}(n)\left(\alpha_N (n+d-1)^N + \cdots + \alpha_0\right) + \cdots \\ + P_1(n)\left(\alpha_N (n+1)^N + \cdots + \alpha_0\right) + P_0(n)\left(\alpha_N n^N + \cdots + \alpha_0\right), \end{aligned} \tag{5.5}$$

which is the result of the substitution of a generic polynomial of arbitrary order N in the left-hand side of (5.2).

For the sake of simplicity, we follow the convention $0^0 = 1$.

(5.5) is a polynomial of degree at most $N+M$ whose leading coefficient can be written in terms of N since $(n+j)^N = \sum_{k=0}^{N} \binom{N}{j} n^k j^{n-k}$, $0 \leqslant k \leqslant d$, is expanded in terms of N. The idea is then to compute starting from degree $N+M$, the first non-zero coefficient of that polynomial which is necessarily a polynomial of the variable N and find its maximal positive integer root.

Once we have found a bound for the degree of polynomial solutions, if there are non-zero solutions of the obtained linear system, then they must depend on certain multiplicative constants since any multiple of a polynomial solution of a holonomic RE is another one. Thus, the output of Petkovšek's algorithm Poly is a linear space that we will represent as a general linear combination of a basis of all polynomial solutions of the given input RE.

The algorithm works as follows.

Algorithm 4 Algorithm Poly for holonomic recurrence equation of order $d \in \mathbb{N}$

Input: Polynomials

$$P_i(n) := \sum_{j=0}^{M} c_{i,j} n^{M-j}, \; 0 \leqslant i \leqslant d, \tag{5.6}$$

where $c_{i,j} \in \mathbb{K}$ such that at least one of $c_{i,0}, 0 \leqslant i \leqslant d$, is non-zero.

Output: A linear combination $\mathcal{L}$ for the space of polynomial solutions over $\mathbb{K}$ of (5.2).

Set $t := \min(M-d, 0)$ and $c_{i,M-j} = 0$ for $t \leqslant j < 0$.

Initialize $s := -1$.

repeat $s = s + 1$; for $0 \leqslant j \leqslant s$, compute

$$b_j^{(s)} := \sum_{i=0}^{d} i^j c_{i,s-j} \tag{5.7}$$

until $\exists j \in \{0, \ldots, s\}$ such that $b_j^{(s)} \neq 0$

Let $\mathcal{D}$ be the set of non-negative integer roots N of the polynomial

$$D(N) := \sum_{j=0}^{s} \binom{N}{j} b_j^s. \tag{5.8}$$

if $\mathcal{D} = \emptyset$ **then** $\mathcal{L} = 0$

7: **else**

$N_0 := \max \mathcal{D}$;

find a linear combination $\mathcal{L}$ for the space of polynomial solutions of (5.2) over $\mathbb{K}$ of degree at most N_0, by solving the linear system obtained by plugging in (5.2) an arbitrary polynomial of degree N_0.

8: **end if**

9: **return** $\mathcal{L}$.

The correctness of the algorithm Poly relies on the proof of the following items.

- The loop in step 3 terminates.
- The set $\mathcal{D}$ is step 5 is a a finite set.
- The set $\mathcal{D}$ is the set of degrees of any polynomial solution of (5.2).

These facts are established by the next three lemmas.

Lemma 5.1. *In algorithm Poly, $s \leqslant d$ at all iterations.*

Proof. Assume that at some point $s = s_0$. Then $b_j^{(s)} = 0$ for $0 \leqslant j \leqslant s < s_0$. In particular, $b_s^{(s)} = 0$ for $0 \leqslant s < s_0$. If $s_0 > d$ this implies that for $0 \leqslant s \leqslant d$,

$$b_s^{(s)} = \sum_{i=0}^{d} i^s c_{i,0} = 0 \iff (i^s)_{i,s=0}^d \cdot \left((c_{i,0})_{i=0}^d\right)^T = 0$$
$$\iff (c_{i,0})_{i=0}^d = (0)_{i=0}^d,$$

since $|(i^s)_{i,s=0}^d|$ is the non-zero Vandermonde determinant $V(0,1,\ldots,d)$. However, the $c_{i,0}$, $0 \leqslant i \leqslant d$ are the coefficients of n^M in the input polynomials of the algorithm, and by assumption we know that at least one of them is non-zero, therefore we have a contradiction. □

Thus, from this lemma we know that the number of iterations in step 3 is at most $d+1$.

Observe that if $t \neq 0$ then $b_j^{(s)} = 0$ for $s - j > M$. The advantage of adding the variable t in the algorithm is that it permits to fix the maximum number of coefficients $c_{i,j}$ that the algorithm will need to find the non-zero $b_j^{(s)}$. This is helpful for the implementation, because it gives the exact number of zero coefficients $c_{i,j}$, $j > M$ that have to be taken into consideration. Indeed if $M < d$, then after M iterations, the non-zero $b_j^{(s)}$ may not yet be found. And since $s \leqslant d$, the remaining number of iterations is at most $d - M = t$. For example, consider the recurrence equation

$$a_{n+3} - 3 \cdot a_{n+2} + 3 \cdot a_{n+1} - a_n = 0. \tag{5.9}$$

Then we have $M = 0$, $d = 3$, and at the first iteration in step 3 we get $b_0^{(0)} = 0$. So s increments to 1, and the algorithm has to compute the obvious zero $b_0^{(1)}$ and ends at $b_1^{(1)} = -1 + 3 - 2 \cdot 3 + 3 \cdot 1 = -1$.

Lemma 5.2. *In algorithm Poly, $\mathcal{D}$ is a finite set.*

This is straightforward from Lemma 5.1 since it implies that $D(N)$ in (5.8) is a non-zero polynomial.

Lemma 5.3. *Let s_0 be the value of s obtained at the last iteration in step 3 of Algorithm 4. Let*

$$a_n = \sum_{k=0}^{N_0} \alpha_k n^k,\ \alpha_{N_0} \neq 0 \tag{5.10}$$

be a polynomial solution of (5.2). Then

$$\sum_{j=0}^{s_0} \binom{N_0}{j} b_j^{s_0} = 0 \tag{5.11}$$

where $b_j^{(s)}$ is as in (5.7).

For the proof of this lemma, one can use the binomial theorem for a general expansion of (5.5) and deduce the relation (5.11) by equating the general monomial coefficient to zero. This will show that if a polynomial of degree N_0 is a solution of (5.2) then N_0 is a solution of the polynomial equation (5.8).

For a Maxima implementation of this algorithm, we first need a function to collect the polynomial coefficients, this is done by our *REcoeff* procedure presented in Section 4.3.2. After that, once t is fixed we collect the coefficients $c_{i,j}$ for $t \leqslant j \leqslant m$, $0 \leqslant i \leqslant d$. Thus, from step 3, the implementation of the algorithm is immediate.

The following Maxima program can be used to collect the coefficients of a polynomial.

```
allcoeffsBound(P,n,t,M):= block([exP,c,j],
     c: [],
     exP: expand(P),
     for j:t thru M do (c : cons(coeff(P,n,j),c)),
     c
)$
```

The syntax is *allcoefBound(P,n,t,M)* which returns a list of coefficients of the given polynomial *P* for the integer powers $t \leqslant j \leqslant M$ of the indeterminate *n*. Below we give an example.

```
(%i1) P:sum(i*n^i,i,-2,5);
```

$$(\%o2) \qquad 5 \cdot n^5 + 4 \cdot n^4 + 3 \cdot n^3 + 2 \cdot n^2 + n - \frac{1}{n} - \frac{2}{n^2}$$

```
(%i2) allcoeffsBound(P,n,-3,6);
```

$$(\%o3) \qquad [0, 5, 4, 3, 2, 1, 0, -1, -2, 0]$$

Let us take some examples and compare the results of our implementation with the those given commands in Maxima's package *solve_rec*. Our implementation has the syntax *PolyPetkov(RE,a[n])*, for a given holonomic recurrence equation *RE* of the indeterminate sequence *a[n]*.

Example 5.1.

```
(%i3) RE: -a[n]+3*a[n+1]-3*a[n+2]+a[n+3]=0;
```

$$(\%o3) \qquad a_{n+3} - 3 \cdot a_{n+2} + 3 \cdot a_{n+1} - a_n = 0$$

```
(%i5) PolyPetkov(RE,a[n]);
```

Evaluation took 0.0000 seconds (0.0000 elapsed)

$$(\%o5) \qquad \%r1 \cdot n^2 + \%r2 \cdot n + \%r3$$

```
(%i6) solve_rec(RE,a[n]);
```

Evaluation took 0.0200 seconds (0.0500 elapsed)

$$(\%o6) \qquad a_n = \%k_3 \cdot n^2 + \%k_2 \cdot n + \%k_1$$

```
(%i7) RE:n*(n+1)*a[n+2]-2*n*(n+10)*a[n+1]+(n+9)*(n+10)*a[n]=0;
```

$$(\%o7) \qquad n \cdot (1+n) \cdot a_{n+2} - 2 \cdot n \cdot (10+n) \cdot a_{n+1} + (9+n) \cdot (10+n) \cdot a_n = 0$$

```
(%i8) PolyPetkov(RE,a[n]);
```

Evaluation took 0.0200 seconds (0.0100 elapsed)

$$(\%o8) \qquad \%r_4 \cdot (n-36) \cdot n \cdot (1+n) \cdot (2+n) \cdot (3+n) \cdot (4+n) \cdot (5+n) \cdot (6+n) \cdot (7+n) \cdot (8+n) + \%r_5 \cdot n \cdot (1+n) \cdot (2+n) \cdot (3+n) \cdot (4+n) \cdot (5+n) \cdot (6+n) \cdot (7+n) \cdot (8+n)$$

```
(%i9) solve_rec(RE,a[n]);
```

Evaluation took 0.1800 seconds (0.2000 elapsed)

$$(\%o9) \qquad a_n = \%k_1 \cdot (n-36) \cdot n \cdot (1+n) \cdot (2+n) \cdot (3+n) \cdot (4+n) \cdot (5+n) \cdot (6+n) \cdot (7+n) \cdot (8+n) + \%k_2 \cdot n \cdot (1+n) \cdot (2+n) \cdot (3+n) \cdot (4+n) \cdot (5+n) \cdot (6+n) \cdot (7+n) \cdot (8+n)$$

As one can see the outputs are given as the sum of factorized linearly independent polynomials.

```
(%i10) RE:n*(n+1)*a[n+2]-2*n*(n+100)*a[n+1]+(n+99)*(n+100)*a[n]=0;
```

$$(\%o10) \qquad n \cdot (1+n) \cdot a_{n+2} - 2 \cdot n \cdot (100+n) \cdot a_{n+1} + (99+n) \cdot (100+n) \cdot a_n = 0$$

```
(%i11) P:PolyPetkov(RE,a[n])$
```

Evaluation took 40.0300 seconds (40.2800 elapsed)

```
(%i12) factor(P)$
```

Evaluation took 0.0100 seconds (0.0100 elapsed)

$$(\%o12) \quad n\cdot(n+1)\cdot(n+2)\cdot(n+3)\cdot(n+4)\cdot(n+5)\cdot(n+6)\cdot(n+7)\cdot(n+8)\cdot(n+9)$$
$$\cdot(n+10)\cdot(n+11)\cdot(n+12)\cdot(n+13)\cdot(n+14)\cdot(n+15)\cdot(n+16)\cdot(n+17)\cdot(n+18)\cdot(n+19)$$
$$\cdot(n+20)\cdot(n+21)\cdot(n+22)\cdot(n+23)\cdot(n+24)\cdot(n+25)\cdot(n+26)\cdot(n+27)\cdot(n+28)\cdot(n+29)$$
$$\cdot(n+30)\cdot(n+31)\cdot(n+32)\cdot(n+33)\cdot(n+34)\cdot(n+35)\cdot(n+36)\cdot(n+37)\cdot(n+38)\cdot(n+39)$$
$$\cdot(n+40)\cdot(n+41)\cdot(n+42)\cdot(n+43)\cdot(n+44)\cdot(n+45)\cdot(n+46)\cdot(n+47)\cdot(n+48)\cdot(n+49)$$
$$\cdot(n+50)\cdot(n+51)\cdot(n+52)\cdot(n+53)\cdot(n+54)\cdot(n+55)\cdot(n+56)\cdot(n+57)\cdot(n+58)\cdot(n+59)$$
$$\cdot(n+60)\cdot(n+61)\cdot(n+62)\cdot(n+63)\cdot(n+64)\cdot(n+65)\cdot(n+66)\cdot(n+67)\cdot(n+68)\cdot(n+69)$$
$$\cdot(n+70)\cdot(n+71)\cdot(n+72)\cdot(n+73)\cdot(n+74)\cdot(n+75)\cdot(n+76)\cdot(n+77)\cdot(n+78)\cdot(n+79)$$
$$\cdot(n+80)\cdot(n+81)\cdot(n+82)\cdot(n+83)\cdot(n+84)\cdot(n+85)\cdot(n+86)\cdot(n+87)\cdot(n+88)\cdot(n+89)$$
$$\cdot(n+90)\cdot(n+91)\cdot(n+92)\cdot(n+93)\cdot(n+94)\cdot(n+95)\cdot(n+96)\cdot(n+97)\cdot(n+98)$$
$$\cdot(\%r6\cdot n+\%r7-4851\cdot\%r6)$$

```
(%i13) solve_rec(RE,a[n])$
```

Evaluation took 41.7900 seconds (42.2400 elapsed)

We hide the output in this example to save space because the corresponding polynomial solutions are linear combinations of $(n)_{99}$ and $(n)_{100}$ that are polynomials of degree 99 and 100, respectively.

```
(%i14) RE:(11323+22134*n+15924*n^2+4992*n^3+576*n^4)*a[n+2]
        -4*(10718+18741*n+11706*n^2+3072*n^3+288*n^4)*a[n+1]+
        (54949+71262*n+34356*n^2+7296*n^3+576*n^4)*a[n]=0;
```

$$(\%o14) \quad \left(11323+22134\cdot n+15924\cdot n^2+4992\cdot n^3+576\cdot n^4\right)\cdot a_{n+2}$$
$$-4\cdot\left(10718+18741\cdot n+11706\cdot n^2+3072\cdot n^3+288\cdot n^4\right)\cdot a_{n+1}$$
$$+\left(54949+71262\cdot n+34356\cdot n^2+7296\cdot n^3+576\cdot n^4\right)\cdot a_n=0$$

```
(%i15) PolyPetkov(RE,a[n]);
```

Evaluation took 0.0100 seconds (0.0100 elapsed)

$$(\%o15) \quad \%r_8\cdot\left(-1771-1596\cdot n+192\cdot n^3\right)+\%r_9\cdot(5+3\cdot n)^2$$

```
(%i16) solve_rec(RE,a[n]);
```

Evaluation took 0.1900 seconds (0.2400 elapsed)

$$(\%o16) \quad a_n=\%k_1\cdot(5+3\cdot n)^2+\%k_2\cdot\left(-1771-1596\cdot n+192\cdot n^3\right)$$

From (2.6) *we know that the general coefficient of* $\frac{1}{(1-z)^k}$, $k \in \mathbb{N}$, *is a polynomial. Therefore, one can generate interesting examples using FindRE as follows.*

```
(%i17) RE:FindRE(exp(z)+1/(1-z)^2,z,a[n]);
```

$$(\%o18) \quad -(1+n)\cdot(2+n)\cdot a_{n+2}+5\cdot(1+n)\cdot a_{n+1}+\left(-4+n+n^2\right)\cdot a_n-(1+n)\cdot a_{n-1}=0$$

```
(%i19) PolyPetkov(RE,a[n]);
```

Evaluation took 0.0000 seconds (0.0100 elapsed)

$$(\%o19) \quad \%r1\cdot(n+1)$$

```
(%i20) solve_rec(RE,a[n]);
```

WARNING: found some hypergeometrical solutions!
Evaluation took 0.1700 seconds (0.1900 elapsed)

$$(\%o20) \quad a_n=\frac{\%k_1}{n!}+\%k_2\cdot(1+n)$$

The warning message appears because the given recurrence equation has less hypergeometric term solutions than its order. One sees that *solve_rec* prioritizes hypergeometric term solutions. This is always the case when the given holonomic recurrence equation is of order greater than 1. For the next example we use *solve_rec_poly* instead, which is also an implementation of Petkovšek's algorithm Poly though it does not seem to be made accessible to users.

```
(%i21) RE:FindRE(exp(z)+log(1+z^2)+1/(1-z)^20,z,a[n]);
```

(%o21) $$201\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}-2\cdot(1+n)\cdot(2+n)\cdot(2305+107\cdot n)\cdot a_{n+2} + (1+n)\cdot\left(4409-879\cdot n+866\cdot n^2\right)\cdot a_{n+1}-n\cdot\left(-9016+6193\cdot n+2596\cdot n^2\right)\cdot a_n + (n-1)\cdot\left(77139-54494\cdot n+11241\cdot n^2\right)\cdot a_{n-1}-(n-2)\cdot\left(443220-278709\cdot n+43558 \cdot n^2\right)\cdot a_{n-2}+2\cdot(n-3)\cdot\left(1264406-590388\cdot n+69039\cdot n^2\right)\cdot a_{n-3}-2\cdot(n-4)\cdot\left(4846109-1873600\cdot n + 181243\cdot n^2\right)\cdot a_{n-4}+(n-5)\cdot\left(29279246-9636913\cdot n+795675\cdot n^2\right)\cdot a_{n-5}-12\cdot(n-6) \cdot\left(5954765-1712600\cdot n+123717\cdot n^2\right)\cdot a_{n-6}+253\cdot(n-7)\cdot\left(573365-146607\cdot n+9424\cdot n^2\right) \cdot a_{n-7}-437\cdot(n-8)\cdot\left(572314-131727\cdot n+7626\cdot n^2\right)\cdot a_{n-8}+437\cdot(n-9) \cdot\left(852511-178150\cdot n+9367\cdot n^2\right)\cdot a_{n-9}-7429\cdot(n-10)\cdot\left(65466-12493\cdot n+600\cdot n^2\right)\cdot a_{n-10} + 29716\cdot(n-11)\cdot\left(18945-3315\cdot n+146\cdot n^2\right)\cdot a_{n-11}-29716\cdot(n-12)\cdot\left(19620-3159\cdot n +128\cdot n^2\right)\cdot a_{n-12}+7429\cdot(n-13)\cdot\left(73138-10879\cdot n+407\cdot n^2\right)\cdot a_{n-13}-1748\cdot(n-14)\cdot\left(261176 -36075\cdot n+1252\cdot n^2\right)\cdot a_{n-14}+437\cdot(n-15)\cdot\left(789293-101879\cdot n+3300\cdot n^2\right)\cdot a_{n-15} -253\cdot(n-16)\cdot\left(918910-111615\cdot n+3398\cdot n^2\right)\cdot a_{n-16}+253\cdot(n-17)\cdot\left(545395 -62772\cdot n+1809\cdot n^2\right)\cdot a_{n-17}-253\cdot(n-18)\cdot\left(280006-30733\cdot n+844\cdot n^2\right)\cdot a_{n-18} +46\cdot(n-19)\cdot\left(668757-70415\cdot n+1854\cdot n^2\right)\cdot a_{n-19}-2\cdot(n-20)\cdot\left(5494632 -558381\cdot n+14180\cdot n^2\right)\cdot a_{n-20}+(n-21)\cdot\left(3114660-307785\cdot n+7591\cdot n^2\right)\cdot a_{n-21} -2\cdot(n-22)\cdot\left(332476-32323\cdot n+782\cdot n^2\right)\cdot a_{n-22}+(n-23)\cdot\left(97967-9597\cdot n+232\cdot n^2\right)\cdot a_{n-23} -(n-24)\cdot\left(8182-869\cdot n+22\cdot n^2\right)\cdot a_{n-24}+(n-25)^2\cdot(n-5)\cdot a_{n-25}-(n-26)^2\cdot a_{n-26}=0 \tag{5.12}$$

```
(%i22) PolyPetkov(RE,a[n]);
```

Evaluation took 2.4400 seconds (2.5600 elapsed)

(%o22) $$\%r10\cdot(n+1)\cdot(n+2)\cdot(n+3)\cdot(n+4)\cdot(n+5)\cdot(n+6)\cdot(n+7)\cdot(n+8)\cdot(n+9) \cdot(n+10)\cdot(n+11)\cdot(n+12)\cdot(n+13)\cdot(n+14)\cdot(n+15)\cdot(n+16)\cdot(n+17)\cdot(n+18)\cdot(n+19)$$

```
(%i23) solve_rec_poly(RE,a[n]);
```

Evaluation took 2.7300 seconds (2.8400 elapsed)

(%o23) $$a_n=\%k_{20}\cdot n^{19}+190\cdot\%k_{20}\cdot n^{18}+16815\cdot\%k_{20}\cdot n^{17}+920550\cdot\%k_{20}\cdot n^{16} +34916946\cdot\%k_{20}\cdot n^{15}+973941900\cdot\%k_{20}\cdot n^{14}+20692933630\cdot\%k_{20}\cdot n^{13}+342252511900\cdot\%k_{20}\cdot n^{12} +4465226757381\cdot\%k_{20}\cdot n^{11}+46280647751910\cdot\%k_{20}\cdot n^{10}+381922055502195\cdot\%k_{20}\cdot n^{9} +2503858755467550\cdot\%k_{20}\cdot n^{8}+12953636989943896\cdot\%k_{20}\cdot n^{7}+52260903362512720\cdot\%k_{20}\cdot n^{6} +161429736530118960\cdot\%k_{20}\cdot n^{5}+371384787345228000\cdot\%k_{20}\cdot n^{4}+610116075740491776\cdot\%k_{20}\cdot n^{3} +668609730341153280\cdot\%k_{20}\cdot n^{2}+431565146817638400\cdot\%k_{20}\cdot n+121645100408832000\cdot\%k_{20}$$

```
(%i24) factor(%o23);
```

Evaluation took 0.0100 seconds (0.2000 elapsed)

$$(\%o24) \quad a_n = \%k_{20}\cdot(n+1)\cdot(n+2)\cdot(n+3)\cdot(n+4)\cdot(n+5)\cdot(n+6)\cdot(n+7)\cdot(n+8)\cdot(n+9)$$
$$\cdot(n+10)\cdot(n+11)\cdot(n+12)\cdot(n+13)\cdot(n+14)\cdot(n+15)\cdot(n+16)\cdot(n+17)\cdot(n+18)\cdot(n+19)$$

Note that this output can also be obtained using *solve_rec_rat* which implements Abramov's algorithm [Abramov, 1989].

```
(%i25) solve_rec_rat(RE,a[n]);
```

Evaluation took 3.3200 seconds (3.7700 elapsed)

$$(\%o25) \quad a_n = \%k_{21}\cdot(n+1)\cdot(n+2)\cdot(n+3)\cdot(n+4)\cdot(n+5)\cdot(n+6)\cdot(n+7)\cdot(n+8)\cdot(n+9)$$
$$\cdot(n+10)\cdot(n+11)\cdot(n+12)\cdot(n+13)\cdot(n+14)\cdot(n+15)\cdot(n+16)\cdot(n+17)\cdot(n+18)\cdot(n+19)$$

For this latter example the command *solve_rec* crashes after about 2 minutes.

```
(%i26) solve_rec(RE,a[n]);
```

SERVER: Lost socket connection ...
Trying to restart Maxima.

We will give more details about what happened in the next section where we describe the Petkovšek algorithm Hyper.

5.2 Hypergeometric Term Solutions of Holonomic Recurrence Equations

We gave a general definition of hypergeometric terms in Chapter 2 (see p. 28). In this section, we consider the case where the symmetry number is 1. Petkovšek's algorithm Hyper [Petkovšek, 1992] is a direct use of the normal form of rational functions described in Lemma 5.4 below.

Lemma 5.4 (A normal form of rational functions)**.** *Let $\mathbb{K}$ be a field of characteristic zero and $r(n)$ a non-zero rational function over $\mathbb{K}$. Then there exists a non-zero constant $Z \in \mathbb{K}$ and monic polynomials (leading coefficient is 1) $A(n)$, $B(n)$, and $C(n)$ over $\mathbb{K}$ such that*

$$r(n) = Z\frac{A(n)}{B(n)}\frac{C(n+1)}{C(n)}, \tag{5.13}$$

where

(L1) $\gcd(A(n), B(n+k)) = 1$ *for all* $k \in \mathbb{N}_{\geqslant 0}$*,*

(L2) $\gcd(A(n), C(n)) = 1$,

(L3) $\gcd(B(n), C(n+1)) = 1$.

Proof. Proving this lemma needs existence and uniqueness. Showing the existence is equivalent to give an algorithm which finds the form (5.13) for a given rational function. For that purpose one can use the Gosper rewriting procedure in a specific way (see [Koepf, 2014, Lemma 9.7]). Below we prove the uniqueness of this form.

Let $r(n)$ be a rational function such that

$$r(n) = Z\frac{A(n)}{B(n)}\frac{C(n+1)}{C(n)} = z\frac{a(n)}{b(n)}\frac{c(n+1)}{c(n)}, \tag{5.14}$$

where A, B, C, a, b, c are monic polynomials satisfying $(L1)$, $(L2)$ and $(L3)$ of Lemma 5.4. Since all these polynomials are monic, we have $Z = z$. Now from (5.14) we can write

$$A(n)b(n)c(n)C(n+1) = a(n)B(n)C(n)c(n+1). \tag{5.15}$$

We are going to show that $c(n)$ must be equal to $C(n)$.

If we denote by

$$\begin{aligned} \delta(n) &:= \gcd(c(n), C(n)), & (5.16)\\ c'(n) &:= c(n)/\delta(n), & (5.17)\\ C'(n) &:= C(n)/\delta(n), & (5.18)\end{aligned}$$

then

$$\gcd(c'(n), C'(n)) = 1 \text{ since } \delta(n) = \delta(n)\gcd(c'(n), C'(n));$$

$$\gcd(a(n), c'(n)) = \gcd\left(a(n), \frac{c(n)}{\delta(n)}\right) = 1 \text{ by } (L2);$$

and in the same way

$$\gcd(b(n), c'(n+1)) = 1 \text{ by } (L3).$$

$$(5.15) \Rightarrow A(n)b(n)c'(n)C'(n+1)\delta(n)\delta(n+1) = a(n)B(n)c'(n+1)C'(n)\delta(n+1)\delta(n),$$

so $A(n)b(n)c'(n)C'(n+1) = a(n)B(n)c'(n+1)C'(n)$ and therefore, for $k \in \mathbb{N}$

$$\begin{aligned} c'(n)|B(n)c'(n+1) &\iff c'(n)|B(n+k-1)c'(n+k),\\ c'(n+1)|A(n)c'(n) &\iff c'(n)|A(n-k)c'(n-k).\end{aligned}$$

Since $\mathbb{K}$ has characteristic zero $\gcd(c'(n), c'(n+k)) = \gcd(c'(n), c'(n-k)) = 1$, for all large enough k. Thus, $c'(n)|B(n+k-1)$ and $c'(n)|A(n-k)$ but from $(L1)$ we know that $1 = \gcd(A(n), B(n+k)) = \gcd(A(n), B(n+k+k-1)) = \gcd(A(n-k), B(n+k-1))$ so $c'(n) = c'$ which is nothing but 1 since we are dealing with monic polynomials.

Hence by (5.17) and (5.18) we obtain

$$\delta(n) = c(n) \Rightarrow c(n)|C(n).$$

By changing the roles of $c(n)$ and $C(n)$ we also get $C(n)|c(n)$, therefore since they are monic polynomials we finally get $c(n) = C(n)$.

Thus from (5.15) it remains $A(n)b(n) = a(n)B(n)$. By $(L1)$, $a(n)|A(n)$ and vice versa, so $a(n) = A(n)$ and similar for $b(n)$ and $B(n)$.

We have just shown that two equal representations of the normal form (5.13) have identical data and this proves the uniqueness of that normal form. □

We move on to our main problem of finding the hypergeometric term solutions of a given holonomic recurrence equation. Proving the algorithm is equivalent to show how each of its steps works. So let us consider a hypergeometric sequence a_n. Then there exists a rational function $r(n) \in \mathbb{K}(n)$ such that

$$r(n) = \frac{a_{n+1}}{a_n}. \tag{5.19}$$

Substituting $a_{n+1} = r(n)a_n$ in (5.2) and dividing by a_n gives

$$P_d(n)\prod_{j=0}^{d-1} r(n+j) + P_{d-1}(n)\prod_{j=0}^{d-2} r(n+j) + \cdots + P_1(n)r(n) + P_0(n) = 0. \tag{5.20}$$

Using the normal form (5.13) for $r(n)$ yields for $1 \leqslant i \leqslant d$

$$\prod_{j=0}^{i-1} r(n+j) = Z^i \left(\prod_{j=0}^{i-1} \frac{A(n+j)}{B(n+j)}\right) \frac{C(n+i)}{C(n)}. \tag{5.21}$$

We substitute this in (5.20) and we multiply the result by $C(n)\prod_{j=0}^{d-1} B(n+j)$. The resulting equation has the form

$$Z^d Q_d(n)C(n+d) + Z^{d-1}Q_{d-1}(n)C(n+d-1) + \cdots + ZQ_1(n)C(n+1) + Q_0(n)C(n) = 0, \tag{5.22}$$

where $Q_i(n) = P_i \prod_{j=0}^{i-1} A(n+j) \prod_{j=i}^{d-1} B(n+j)$.

We remark that $A(n)$ appears as a factor of all the first d terms. Using properties $(L1)$ and $(L2)$ of Lemma 5.4, we deduce that $A(n)$ must be a monic factor of $P_0(n)$. Similarly $B(n+d-1)$ is a monic factor of $P_d(n)$. Thus, we have a finite set of candidates for $A(n)$ and $B(n)$ from the RE (5.2). Candidates for Z are roots of the equation

$$\sum_{i=0}^{d} l_i Z^i = 0, \tag{5.23}$$

where $l_i, 0 \leqslant i \leqslant d$ is the coefficient of n^M in Q_i in (5.22), where

$$M = \max\{\deg(Q_i), 0 \leqslant i \leqslant d\}. \tag{5.24}$$

Therefore for fixed choice of $A(n)$, $B(n)$ and Z, (5.22) is a holonomic RE for $C(n)$. And finally it remains to find non-zero polynomial solutions of that recurrence equation. For this purpose one can use the Algorithm 4. If there are polynomial solutions, then a_n exists and we have the normal form of its characteristic rational function. On the other hand, if for all monic factors $A(n)$ of $P_0(n)$ and $B(n)$ of $P_d(n-d+1)$ none rational function is found, then there is no hypergeometric term solution.

Note that the general expression of each hypergeometric term solution has to be deduced from the relation $a_{n+1} = r(n)a_n$, by using the hypergeometric formula as defined in (2.53). But we leave that step for the next chapter. In our work, Petkovšek's algorithm is somewhat like a lantern for our approach of van Hoeij's algorithm whose hypergeometric terms in the outputs are given in a "simple" formula form.

Remarks

- Observe that shifting $A(n)$ and $B(n)$ by an integer k in the normal form (5.13) does not change the hypergeometric term solution candidate, but however, in this case, $A(n)$ and $B(n)$ have to be monic factors of, respectively, $P_0(n-k)$ and $P_d(n-d+1-k)$. $k=1$ is the case that we use for our implementation.

- Sometimes extension fields have to be considered. Indeed the choice of the monic factors $A(n)$ and $B(n)$ depends on the field where the factorization of $P_0(n-1) \cdot P_d(n-d)$ is done. It should be mentioned that the algorithm needs complete factorizations to try all the possible combinations. However, algorithmically we can only have factorizations over $\mathbb{Q}$ or some of its extension fields when some roots of the involved polynomials are computable.

- The candidates for Z also depend on the considered field since they represent the roots of the polynomial (5.23).

- Depending on the degree of the leading and trailing polynomial coefficients of a given holonomic recurrence equation, the complexity of Petkovšek's algorithm can be quite high, because in this case many factors have to be checked over the considered field. And moreover if the degrees of the other polynomial coefficients are also high, then the search for polynomial solutions of (5.22) can also increase the timing.

  Observe, however, that for two given monic factors $A(n)$ and $B(n)$ in Petkovšek's algorithm, the coefficients of the equation (5.23) which determines Z depend only on the difference $D(A,B) := \deg(A(n)) - \deg(B(n))$ and not on $A(n)$ and $B(n)$ themselves. Therefore it is advantageous to test pairs of factors $A(n)$, $B(n)$ according to the value of $D(A,B)$ [Petkovšek, 1992, Remark 4.1]. Moreover, it is important to take into account that the recurrence equation (5.2) cannot have more than d linearly independent hypergeometric term solutions.

Let us practice the algorithm on the recurrence equation

$$4 \cdot (2+n) \cdot a_{n+2} + 6 \cdot (1+n) \cdot a_{n+1} + (2 \cdot n - 1) \cdot a_n = 0 \tag{5.25}$$

of the Taylor coefficients of $\sqrt{1+z} + \dfrac{1}{\sqrt{1+z}}$.

1. The leading and trailing polynomial coefficients are, respectively, $P_2(n) = 4(n+2)$ and $P_0(n) = (2n-1)$. To build the normal form (5.13) of a hypergeometric term solution of (5.25), we need monic factors $A(n)$ of $P_0(n-1)$ and $B(n)$ of $P_2(n-2)$. Thus we have $A(n) \in \{1, (n-\frac{3}{2})\}$ and $B(n) \in \{1, n\}$.

2. For each $A(n)$ and $B(n)$, we set

$$\begin{aligned} Q_2(n) &= \frac{P_2(n)\cdot A(n+1)}{B(n+1)}, && (5.26)\\ Q_1(n) &= P_1(n), && (5.27)\\ Q_0(n) &= \frac{P_0(n)\cdot B(n)}{A(n)}, && (5.28)\\ M &= \max\{\deg(Q_0), \deg(Q_1), \deg(Q_2)\}, && (5.29)\\ l_i &= \text{coeff}(Q_i, n, M). && (5.30) \end{aligned}$$

   By Petkovšek algorithm (see (5.22)), a hypergeometric term solution of (5.25) with the normal form of its characteristic ratio (5.13) (where $A(n)$ and $B(n)$ are substituted by $A(n+1)$ and $B(n+1)$), exists if and only if there exists a number Z such that

$$l_2 Z^2 + l_1 Z + l_0 = 0, \tag{5.31}$$

   and the holonomic RE

$$Z^2 Q_2(n) C(n+2) + Z Q_1(n) C(n+1) + Q_0(n) C(n) = 0 \tag{5.32}$$

   has non-zero polynomial solutions.

3. Taking $A(n) = B(n) = 1$ leads to $Z = -\frac{1}{2}$ or $Z = -1$; however there is no non-zero polynomial solutions of the auxiliary holonomic recurrence equations.

4. If exactly one of $A(n), B(n)$ is equal to 1, the equation for Z is either $4Z^2 = 0$ or contradictory.

5. Finally the choice $A(n) = n - \frac{3}{2}$ and $B(n) = n$ leads to $Z = -\frac{1}{2}$ or $Z = -1$ again. For $Z = -\frac{1}{2}$ there is no non-zero polynomial solution of the corresponding auxiliary RE; however for $Z = -1$ we get the recurrence equation

$$(2+n)\cdot(2\cdot n-1)\cdot(1+2\cdot n)\cdot \mathrm{C}\,(n+2) - 3\cdot(1+n)\cdot(2+n)\cdot(2\cdot n-1)\cdot \mathrm{C}\,(n+1) + (1+n)\cdot(2+n)\cdot(2\cdot n-1)\cdot \mathrm{C}\,(n) = 0 \tag{5.33}$$

   whose polynomial solutions are multiples of $(n-1)$ as our implementation of Algorithm 4 shows below.

```
(%i1) RE:(n+2)*(2*n-1)*(2*n+1)*a[n+2]-3*(1+n)*(2+n)
      *(2*n-1)*a[n+1] + (n+1)*(n+2)*(2*n-1)*a[n]=0$
```

```
(%i2) PolyPetkov(RE,a[n]);
```

$$(\%o2) \qquad \%r1 \cdot (n-1)$$

6. Hence the unique hypergeometric term solution of (5.25) has the characteristic ratio

$$-\frac{n\cdot(2\cdot n-1)}{2\cdot(n-1)\cdot(n+1)}. \tag{5.34}$$

From this previous example one sees how Petkovšek's algorithm applies to the particular case of holonomic recurrence equations of order $d=2$. More generally the Petkovšek algorithm Hyper works as follows.

Algorithm 5 Algorithm Hyper for holonomic recurrence equation of order $d\in\mathbb{N}$

Input: Polynomials $P_i(n), i=0,\dots,d$ over a field of characteristic zero $\mathbb{K}$. $\mathbb{K}$ might be specified by the user.

Output: Set of term ratios of all hypergeometric term solutions over $\mathbb{K}$ of (5.2).

1: Set $\mathcal{H}=\{\}$.

2: For all monic factors $A(n)$ of $P_0(n-1)$ and $B(n)$ of $P_d(n-d)$ over $\mathbb{K}$ compute:

2 − 1 $Q_i(n)=P_i\prod_{j=0}^{i-1}A(n+j)\prod_{j=i}^{d-1}B(n+j)$, for $i=0,\dots,d$;

2 − 2 $M=\max\{\deg(Q_i), 0\leqslant i\leqslant d\}$;

2 − 3 $l_i=\text{coeff}(Q_i,n,M)$;

2 − 4 For all non-zero Z satisfying $\sum_{i=0}^{d}l_iZ^i=0$ do:

2 − 4 − 1 Apply Algorithm 4 to the recurrence equation

$$\sum_{i=0}^{d}Z^iQ_i(n)C(n+i)=0 \tag{5.35}$$

to find all its non-zero polynomial solutions $C(n)$. If there is a polynomial solution $C(n)$, then add the term ratio

$$Z\frac{A(n+1)}{B(n+1)}\frac{C(n+1)}{C(n)}$$

to $\mathcal{H}$.

3: Return $\mathcal{H}$.

We implemented Petkovšek's algorithm as *HyperPetkov(RE,a[n],[K])*, where *RE* is the input holonomic recurrence with the unknown *a[n]*. The output is a set of ratios of all hypergeometric term solutions. The optional variable *K* with default value *Q* for solutions over the field of rational numbers can be changed to *C* in order to consider algebraic extension fields of $\mathbb{Q}$.

Now, we present some examples. We start with the last example of the previous section.

```
(%i3) RE:FindRE(exp(z)+log(1+z^2)+1/(1-z)^20,z,a[n])$
```

```
(%i4) HyperPetkov(RE,a[n]);
```

Evaluation took 339.0600 seconds (342.2200 elapsed)

$$(\%o4) \qquad \left\{\frac{20+n}{n+1}, \frac{1}{n+1}\right\}$$

After about 5 minutes, we get the ratios of the general coefficient of the power series representations of $\frac{1}{(1-z)^{20}}$ and $\exp(z)$ at 0. The following Maxima code can be used to compute ratios of expressions involving factorials, Pochhammer or Γ symbols.

```
ratio(term,n):=block([r],
    r: subst(n+1,n,term)/term,
    factor(ratsimp(minfactorial(makefact(makegamma(r)))))
)$
```

```
(%i5) term1:1/n!;
```

$$(\%o5) \qquad \frac{1}{n!}$$

```
(%i6) term2:binomial(n+19,9);
```

$$(\%o6) \qquad \frac{(11+n)\cdot(12+n)\cdot(13+n)\cdot(14+n)\cdot(15+n)\cdot(16+n)\cdot(17+n)}{362880} \cdot(18+n)\cdot(19+n)$$

```
(%i7) ratio(term1,n);
```

$$(\%o7) \qquad \frac{1}{n+1}$$

```
(%i8) ratio(term2,n);
```

$$(\%o8) \qquad \frac{20+n}{n+11}$$

In order to get the ratio of the general coefficient for the power series representation of $\log(1+z^2)$, we need to allow extension fields of $\mathbb{Q}$.

```
(%i9) HyperPetkov(RE,a[n],C);
```

Evaluation took 342.2400 seconds (347.8800 elapsed)

$$(\%o9) \qquad \left\{\frac{20+n}{n+1}, \frac{1}{n+1}, -\frac{i\cdot n}{n+1}, \frac{i\cdot n}{n+1}\right\}$$

Observe that

$$\log(1+z^2) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n+1} z^{2n+2} = -\sum_{n=0}^{\infty} \frac{\mathrm{i}^{n+1}+(-\mathrm{i})^{n+1}}{2(n+1)} z^{n+1}, \tag{5.36}$$

hence the result found by our implementation. Nevertheless, with our m-fold algorithm presented in Chapter 7, the 2-fold and the 1-fold hypergeometric term solutions will correctly be found over $\mathbb{Q}$.

As we now have described Petkovšek's algorithm to find hypergeometric term solutions of holonomic recurrence equations, we can analyze the source code of *solve_rec* and explain why it crashes for some examples, in particular (5.12). The source code is available at [Vodopivec, 2014]. Examining each step of the implementation of Hyper in *solve_rec*, we found that the origin of the problem is related to the fact that a polynomial equation of degree 25 for Z corresponding to the normal form (5.13) was not solved by the Maxima command *solve* which has returned the equation as output. That equation is used in the remaining part of the algorithm and the crash raises up when the polynomial coefficients of the corresponding holonomic recurrence equation for $C(n)$ in (5.22) are simplified and expanded. We added the Maxima commands *throw/catch* inside a copy of the package *solve_rec* to leave the running program when a polynomial is given as a possible value for Z and we obtained the following.

```
(%i10) RE:FindRE(exp(z)+log(1+z^2)+1/(1-z)^20,z,a[n])$
(%i11) solve_rec(RE,a[n]);
```

$$\begin{aligned}&\big[-i,i,1,0,201\cdot\%z^{25}-13\cdot\%z^{24}+652\cdot\%z^{23}-1730\cdot\%z^{22}+8846\cdot\%z^{21}\\&-32330\cdot\%z^{20}+95172\cdot\%z^{19}-226138\cdot\%z^{18}+442035\cdot\%z^{17}-721259\cdot\%z^{16}\\&+994840\cdot\%z^{15}-1174428\cdot\%z^{14}+1202852\cdot\%z^{13}-1085280\cdot\%z^{12}+875976\cdot\%z^{11}\\&-639540\cdot\%z^{10}+422807\cdot\%z^{9}-250173\cdot\%z^{8}+129580\cdot\%z^{7}-57134\cdot\%z^{6}\\&+20790\cdot\%z^{5}-6028\cdot\%z^{4}+1332\cdot\%z^{3}-210\cdot\%z^{2}+21\cdot\%z-1\big]\end{aligned}$$

We get a list of four complex numbers and the corresponding polynomial. Note that due to the lack of an algorithm for complete factorization over the complex field $\mathbb{C}$, Maxima's command *solve* cannot find all complex roots of many polynomials and this is a general situation in all computer algebra systems. When *solve* does not find any root of a given polynomial, this latter is returned. Despite the fact that this can be seen as an algebraic number in a suitable extension field, such a situation should be avoided according to the expected output. The polynomial whose roots are element of the above list is obtained twice with the recurrence equation (5.12) but the problem occurs in the second case where higher degrees are involved. One can solve this issue by adding a step to filter the values found for Z in order to only take values that belong to the considered field. After having done that in our copy of *solve_rec*, we finally get the same hypergeometric terms as *HyperPetkov* did.

```
(%i12) solve_rec(RE,a[n]);
```

WARNING: found some hypergeometrical solutions!

Evaluation took 323.8400 seconds (329.3000 elapsed)

(%o12) $$a_n = \frac{\%k_1}{n!} - \frac{i \cdot \%k_2 \cdot (-1)^{\frac{3 \cdot n}{2}}}{n} - \frac{i \cdot \%k_3 \cdot (-1)^{\frac{n}{2}}}{n} + \%k_4 \cdot (1+n) \cdot (2+n) \cdot (3+n) \cdot (4+n) \cdot (5+n) \cdot (6+n) \cdot (7+n) \cdot (8+n) \cdot (9+n) \cdot (10+n) \cdot (11+n) \cdot (12+n) \cdot (13+n) \cdot (14+n) \cdot (15+n) \cdot (16+n) \cdot (17+n) \cdot (18+n) \cdot (19+n)$$

We will now use this particular updated version of the package *solve_rec* for our next examples.

```
(%i13) RE:FindRE(sin(3*z)+z*cos(z),z,a[n]);
```

(%o13) $$-3 \cdot (1+n) \cdot (2+n) \cdot (3+n) \cdot (4+n) \cdot a_{n+4} + 2 \cdot (1+n) \cdot (2+n) \cdot \left(-3 - 24 \cdot n + 8 \cdot n^2\right) \cdot a_{n+2} + \left(189 - 448 \cdot n + 160 \cdot n^2\right) \cdot a_n + 144 \cdot a_{n-2} = 0$$

```
(%i14) HyperPetkov(RE,a[n],C);
```

(%o14) $$\left\{-\frac{i}{n}, \frac{i}{n}, -\frac{3 \cdot i}{n+1}, \frac{3 \cdot i}{n+1}\right\}$$

```
(%i15) solve_rec(RE,a[n]);
```

WARNING: found some hypergeometrical solutions!
Evaluation took 0.4200 seconds (0.4600 elapsed)

(%o15) $$a_n = -\frac{\%k_1 \cdot 3^n \cdot (-1)^{\frac{3 \cdot n}{2}}}{n!} + \frac{\%k_2 \cdot (-1)^{\frac{n}{2}} \cdot 3^n}{n!} - \frac{i \cdot \%k_3 \cdot (-1)^{\frac{3 \cdot n}{2}}}{(n-1)!} - \frac{i \cdot \%k_4 \cdot (-1)^{\frac{n}{2}}}{(n-1)!}$$

```
(%i16) RE: FindRE(z/(1-z)^4+5*z^3*exp(z)
       +(1+7*z^2)*log(1+z)+(2+z)/sqrt(1+z),z,a[n])
```

(%o16)
$$
\begin{aligned}
&-5280\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot(4+n)\cdot a_{n+4}\\
&-2640\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot(5\cdot n-8)\cdot a_{n+3}\\
&+24\cdot(1+n)\cdot(2+n)\cdot\left(-93-658\cdot n+1631\cdot n^2\right)\cdot a_{n+2}\\
&+6\cdot(1+n)\cdot\left(-10138+83737\cdot n-92168\cdot n^2+19487\cdot n^3\right)\cdot a_{n+1}\\
&-\left(-31248-34967726\cdot n+36618923\cdot n^2-12131065\cdot n^3+1282474\cdot n^4\right)\cdot a_n\\
&-2\cdot\left(221079135-380139783\cdot n+195296872\cdot n^2-38772534\cdot n^3+2574866\cdot n^4\right)\cdot a_{n-1}\\
&-\left(3712130260-3767615216\cdot n+1256567953\cdot n^2-164084872\cdot n^3+6852934\cdot n^4\right)\cdot a_{n-2}\\
&-\left(14107137708-10084423479\cdot n+2378149452\cdot n^2-209132861\cdot n^3+4800640\cdot n^4\right)\cdot a_{n-3}\\
&-\left(35157222828-19558483362\cdot n+3594516098\cdot n^2-239400297\cdot n^3+3470080\cdot n^4\right)\cdot a_{n-4}\\
&+\left(-45621140674+19519960278\cdot n-2558764947\cdot n^2+84771027\cdot n^3+2203084\cdot n^4\right)\cdot a_{n-5}\\
&+\left(-5734417752-4013264964\cdot n+2087281998\cdot n^2-278988169\cdot n^3+11499176\cdot n^4\right)\cdot a_{n-6}\\
&+\left(25381011810-15595818800\cdot n+3270465181\cdot n^2-283827079\cdot n^3+8685320\cdot n^4\right)\cdot a_{n-7}\\
&-\left(22497576024-7539646356\cdot n+939204722\cdot n^2-53297847\cdot n^3+1227622\cdot n^4\right)\cdot a_{n-8}\\
&-6\cdot\left(5082246416-1762890103\cdot n+227898928\cdot n^2-13053535\cdot n^3+280578\cdot n^4\right)\cdot a_{n-9}\\
&+\left(31901663148-10946543180\cdot n+1404565623\cdot n^2-79902790\cdot n^3+1700970\cdot n^4\right)\cdot a_{n-10}\\
&+\left(26218208116-7915623153\cdot n+886891990\cdot n^2-43658973\cdot n^3+795448\cdot n^4\right)\cdot a_{n-11}\\
&-\left(14887498140-4406091486\cdot n+488721582\cdot n^2-24089215\cdot n^3+445364\cdot n^4\right)\cdot a_{n-12}\\
&-\left(10525830534-2722517874\cdot n+261550075\cdot n^2-11048019\cdot n^3+172844\cdot n^4\right)\cdot a_{n-13}\\
&+7\cdot\left(296448016-80163176\cdot n+8106388\cdot n^2-363487\cdot n^3+6100\cdot n^4\right)\cdot a_{n-14}\\
&+7\cdot\left(268434654-61319032\cdot n+5216245\cdot n^2-195709\cdot n^3+2730\cdot n^4\right)\cdot a_{n-15}\\
&+147\cdot(n-18)\cdot\left(-116184+17111\cdot n-808\cdot n^2+12\cdot n^3\right)\cdot a_{n-16}\\
&-882\cdot(n-19)^2\cdot(2\cdot n-35)\cdot a_{n-17}=0
\end{aligned}
\tag{5.37}
$$

```
(%i17) HyperPetkov(RE,a[n]);
```

Evaluation took 53.9300 seconds (55.2500 elapsed)

(%o17)
$$
\left\{\frac{1}{n-2},-\frac{n-2}{n-1},\frac{3+n}{n},-\frac{2\cdot n-1}{2\cdot n},-\frac{n\cdot(2\cdot n-1)}{2\cdot(n-1)\cdot(n+1)},-\frac{(n-2)\cdot n\cdot(3+4\cdot n)}{(n-1)\cdot(n+1)\cdot(4\cdot n-1)}\right\}
$$

Observe that the algorithm Hyper does not consider linear dependence of hypergeometric term solutions. In this example, the given output is the set of ratios corresponding to each summand in the expression

$$5z^3\exp(z) + \log(z+1) + 7z^2\log(z+1) + \frac{2}{\sqrt{z+1}} + \frac{z}{\sqrt{z+1}} + \frac{z}{(1-z)^4}. \tag{5.38}$$

Note that using van Hoeij's approach in the next chapter, we will get outputs which form a basis of all hypergeometric term solutions.

With *solve_rec*, however, linear dependence is tested by means of the Casoratian determinant as explained in [Petkovšek and Salvy, 1993, Section 5.].

```
(%i18) solve_rec(RE,a[n]);
```

WARNING: found some hypergeometrical solutions!
Evaluation took 53.9400 seconds (55.3400 elapsed)

$$(\%o18)\qquad a_n = -\frac{\%k_1\cdot(n-1)\cdot(-1)^n\cdot\Gamma\left(\frac{2\cdot n-1}{2}\right)}{\sqrt{\pi}\cdot n!} + \frac{\%k_2}{(n-3)!} + \frac{\%k_3\cdot(4\cdot n-1)\cdot(-1)^n}{(n-2)\cdot n} + \%k_4\cdot n\cdot(1+n)\cdot(2+n)$$

```
(%i19) RE:(8*n-56)*a[n-3]+(-8*n+24)*a[n-1]+(n+1)*a[n+1] = 0;
```

$$(\%o19)\qquad (1+n)\cdot a_{n+1} + (24-8\cdot n)\cdot a_{n-1} + (8\cdot n-56)\cdot a_{n-3} = 0$$

```
(%i20) HyperPetkov(RE,a[n],C);
```

Evaluation took 0.1000 seconds (0.4100 elapsed)

$$(\%o20)\qquad \left\{-\frac{\sqrt{2}\cdot\sqrt{\sqrt{2}+2}\cdot(n-4)\cdot n\cdot\left(2+2^{\frac{5}{2}}-3\cdot n-2^{\frac{3}{2}}\cdot n+n^2\right)}{(n+1)\cdot\left(n^3-2^{\frac{3}{2}}\cdot n^2-6\cdot n^2+2^{\frac{7}{2}}\cdot n+11\cdot n-3\cdot 2^{\frac{3}{2}}-6\right)},\right.$$
$$\frac{\sqrt{2}\cdot\sqrt{\sqrt{2}+2}\cdot(n-4)\cdot n\cdot\left(2+2^{\frac{5}{2}}-3\cdot n-2^{\frac{3}{2}}\cdot n+n^2\right)}{(n+1)\cdot\left(n^3-2^{\frac{3}{2}}\cdot n^2-6\cdot n^2+2^{\frac{7}{2}}\cdot n+11\cdot n-3\cdot 2^{\frac{3}{2}}-6\right)},$$
$$-\frac{\sqrt{2-\sqrt{2}}\cdot\sqrt{2}\cdot(n-4)\cdot n\cdot\left(2-2^{\frac{5}{2}}-3\cdot n+2^{\frac{3}{2}}\cdot n+n^2\right)}{(n+1)\cdot\left(n^3+2^{\frac{3}{2}}\cdot n^2-6\cdot n^2-2^{\frac{7}{2}}\cdot n+11\cdot n+3\cdot 2^{\frac{3}{2}}-6\right)},$$
$$\left.\frac{\sqrt{2-\sqrt{2}}\cdot\sqrt{2}\cdot(n-4)\cdot n\cdot\left(2-2^{\frac{5}{2}}-3\cdot n+2^{\frac{3}{2}}\cdot n+n^2\right)}{(n+1)\cdot\left(n^3+2^{\frac{3}{2}}\cdot n^2-6\cdot n^2-2^{\frac{7}{2}}\cdot n+11\cdot n+3\cdot 2^{\frac{3}{2}}-6\right)}\right\}$$

For this example, *solve_rec* crashes by trying to find "simple" formulas of the corresponding hypergeometric term solutions. Indeed, some considerations should be taken into account while looking for a "simple" formula. This will be explained in Section 6.1.

```
(%i21) RE:FindRE((1+z)^k+asin(sqrt(z))/sqrt(z),z,a[n]);
```

$$(\%o22)\quad 2\cdot(6\cdot k-1)\cdot(1+n)\cdot(2+n)\cdot(5+2\cdot n)\cdot a_{n+2} + (1+n)\cdot(3+2\cdot n)\cdot\left(3+20\cdot k-20\cdot k^2-4\cdot n-12\cdot k\cdot n+8\cdot k^2\cdot n\right)\cdot a_{n+1} - (1+2\cdot n)\cdot\left(19\cdot k-10\cdot k^2-9\cdot n+10\cdot k\cdot n-52\cdot k^2\cdot n+8\cdot k^3\cdot n+20\cdot k\cdot n^2 +8\cdot k^2\cdot n^2\right)\cdot a_n-(2\cdot n-1)\cdot\left(1+4\cdot k+6\cdot k^2+12\cdot k^3+3\cdot n+12\cdot k\cdot n+4\cdot k^2\cdot n-16\cdot k^3\cdot n-4\cdot n^2 - 12\cdot k\cdot n^2+8\cdot k^2\cdot n^2\right)\cdot a_{n-1}+(1+2\cdot k)^2\cdot(-2-k+n)\cdot(2\cdot n-3)^2\cdot a_{n-2}=0$$

```
(%i23) HyperPetkov(RE,a[n]);
```

Evaluation took 0.9400 seconds (0.9600 elapsed)

$$(\%o23)\qquad \left\{-\frac{n-k}{n+1},\frac{(1+2\cdot n)^2}{2\cdot(n+1)\cdot(2\cdot n+3)}\right\}$$

```
(%i24) solve_rec(RE,a[n]);
```

WARNING: found some hypergeometrical solutions!
Evaluation took 4.7700 seconds (4.8900 elapsed)

$$(\%o24)\qquad a_n=\frac{\%k_1\cdot\Gamma\left(\frac{1+2\cdot n}{2}\right)}{\sqrt{\pi}\cdot(2\cdot n+1)\cdot n!}-\frac{\%k_2\cdot(-1)^n\cdot(-1-k+n)!}{(-1-k)!\cdot n!}$$

5.3 Holonomic Recurrence Equations of Linearly Independent Hypergeometric Terms

This section is motivated by the aim of generating more examples of holonomic recurrence equations having non-empty sets of hypergeometric term solutions. In particular, this can be used to show the lack of efficiency of Petkovšek's algorithm when the hypergeometric terms considered lead to holonomic recurrence equations with higher degrees for their leading and trailing coefficients, because such equations cannot always be found by playing with analytic holonomic functions. A general algorithm which deals with the algebra of holonomic recurrence equations satisfied by linear combinations of hypergeometric terms appeared in ([Koepf, 2006, Section 10.9, Section 10.16], [Koepf, 1997], [Stanley, 1980]). However, we describe an algorithm for the particular case of generating holonomic recurrence equations satisfied by a given linear combination of linearly independent hypergeometric terms, which is enough for our purpose.

The algorithm that we present is just the generalization of the case of two given linearly independent hypergeometric terms. Thus, we treat this particular case and by simple analogy we give the general approach for a given set of linearly independent hypergeometric terms, and we end with a Maxima implementation.

Let $\mathbb{K}$ be a field of characteristic zero, and a_n and b_n be two linearly independent hypergeometric terms over $\mathbb{K}$ such that

$$a_{n+1} = r_1(n)a_n \text{ and } b_{n+1} = r_2(n)b_n, \tag{5.39}$$

where r_1 and r_2 are rational functions in $\mathbb{K}(n)$. As we consider two terms, the order of the RE sought is 2, so we are looking for a recurrence equation of the form

$$P_2(n)s_{n+2} + P_1(n)s_{n+1} + P_0(n)s_n = 0, \tag{5.40}$$

where P_0, P_1, P_2 are polynomials over $\mathbb{K}$, satisfied by a_n and b_n. We must assume that $P_0 \cdot P_2 \neq 0$, otherwise the recurrence equation can be reduced to a first order recurrence relation. Thus finding (5.40) is equivalent to searching for rational functions R_2 and R_1 such that

$$R_2(n)s_{n+2} + R_1(n)s_{n+1} + s_n = 0. \tag{5.41}$$

Using (5.39), we have

$$a_{n+2} = r_1(n+1)a_{n+1} \text{ and } b_{n+2} = r_2(n+1)b_{n+1}. \tag{5.42}$$

By substitution, a_n and b_n satisfy (5.41) if and only if

$$\begin{cases} r_1(n+1)R_2 + R_1 = -\frac{1}{r_1(n)} \\ r_2(n+1)R_2 + R_1 = -\frac{1}{r_2(n)} \end{cases}, \tag{5.43}$$

which is a linear system of two equations with two unknowns in $\mathbb{K}(n)$. Furthermore, a solution exists and is unique since the determinant of the system

$$r_a(n+1) - r_b(n+1) \neq 0 \tag{5.44}$$

by assumption. As a linear system of two equations, the exact solution is easy to compute, that is

$$R_1(n) = \frac{r_2(n+1)r_2(n) - r_1(n+1)r_1(n)}{r_1(n)r_2(n)(r_1(n+1) - r_2(n+1))}, \tag{5.45}$$

$$R_2(n) = \frac{r_1(n) - r_2(n)}{r_1(n)r_2(n)(r_1(n+1) - r_2(n+1))}. \tag{5.46}$$

Finally, the holonomic recurrence equation sought is found by multiplying the equation (5.41) by the least common multiple of the denominators of $R_1(n)$ and $R_2(n)$ and canceling the common factors.

In the general case, let $a_n^{[i]}, i = 1, \ldots, d$ $(d \geqslant 1)$ be d given linearly independent hypergeometric terms over $\mathbb{K}$ such that

$$a_{n+1}^{[i]} = r_i(n)a_n^{[i]},\ i = 1, \ldots, d, \tag{5.47}$$

for some rational functions r_i. The vectors $(R_1(n), R_2(n), \ldots, R_d(n))^T \in \mathbb{K}(n)^d$ of rational coefficients of the recurrence equation

$$R_d(n)s_{n+d} + R_{d-1}(n)s_{n+d-1} + \ldots + R_1(n)s_{n+1} + s_n = 0 \tag{5.48}$$

satisfied by each hypergeometric term $a_n^{[i]}$, is the unique vector solution $v \in \mathbb{K}(n)^d$ of the matrix system

$$\left[\prod_{k=1}^{j-1} r_i(n+k)\right]_{i,j=1,\ldots,d} \cdot v = -\left(\frac{1}{r_i(n)}\right)^T_{i=1,\ldots,d}. \tag{5.49}$$

The following Maxima program computes a holonomic recurrence equation of order d for a given set of d hypergeometric terms.

```
sumhyperRE(H,a):=catch(block([n,term,R,d,k,i,j,M,b,Vcomp,V,aterm,RE],
    if( (not listp(H)) or atom(a)) then (
        print("wrong input(s)"),
        throw(false)
    ),
    n: first(a),
    R: map(lambda([term], ratio(term,n)),H),
    d: length(H),
    R: sublist(R, lambda([term], ratfunp(term,[n]))),
    if(length(R)<d) then (
        print("There are some non hypergeometric terms in Q"),
        print(false)
    ),
    M: apply('matrix, makelist(makelist(prod(subst(n+k,n,R[i])
                                    ,k,1,j-1),j,1,d),i,1,d)),
    b: apply('matrix, makelist([-1/R[i]],i,1,d)),
    Vcomp: makelist(concat('%v,i),i,1,d),
    V: transpose(matrix(Vcomp)),
    linsolvewarn:false,
    V: factor(linsolve(xreduce('append,args(M.V-b)),Vcomp)),
    linsolvewarn:true,
    if(length(%rnum_list)>0) then V:subst(map(lambda([v],v=0),
    %rnum_list),V),
    V: map(rhs, V),
    aterm: makelist(subst(n+i,n,a),i,0,d),
    RE: num(factor(aterm[1] + sum(V[i]*aterm[i+1],i,1,d))),
    for i: 1 thru d+1 do RE: map(factor, collectterms(RE, aterm[i])),
    RE=0
))$
```

Of course, to use this Maxima function, our function *ratio* has to be available. Nevertheless all these Maxima functions are available in our package FPS. We mention that this algorithm can also be used to find holonomic recurrence equations for polynomials since they are also hypergeometric.

Let us apply this algorithm to some examples. We start with the example given in [Koepf, 2014, Section 9.14] for generating a time consuming holonomic RE for Petkovšek's algorithm.

```
(%i1) term1:(pochhammer(1/2,n)^5*pochhammer(1,n))
      /(pochhammer(3/4,n)^3*pochhammer(1/3,n));
```

$$(\%o1) \quad \frac{\left(\frac{1}{2}\right)_n^{\,5} \cdot n!}{\left(\frac{1}{3}\right)_n \cdot \left(\frac{3}{4}\right)_n^{\,3}}$$

```
(%i2) term2:pochhammer(1/4,n)
      /(pochhammer(1,n)^3*pochhammer(1/3,n)^4);
```

$$(\%o2) \quad \frac{\left(\frac{1}{4}\right)_n}{\left(\frac{1}{3}\right)_n^{\,4} \cdot n!^3}$$

```
(%i3) RE:sumhyperRE([term1,term2],a[n]);
```

Evaluation took 0.0700 seconds (0.0800 elapsed)

$$\begin{aligned}(\%o3) \quad & 4\cdot(2+n)^3\cdot(1+3\cdot n)\cdot(4+3\cdot n)^4\cdot(7+4\cdot n)^3\cdot\Big(-721-5648\cdot n-13640\cdot n^2 \\ & -5416\cdot n^3+42024\cdot n^4+141288\cdot n^5+292648\cdot n^6+437960\cdot n^7+469808\cdot n^8 \\ & +352192\cdot n^9+175104\cdot n^{10}+51840\cdot n^{11}+6912\cdot n^{12}\Big)\cdot a_{n+2}-3\cdot(1+3\cdot n)\cdot \\ & (-17831097+132123312\cdot n+5524092864\cdot n^2+59712834816\cdot n^3+393755684352\cdot n^4 \\ & +1877330292224\cdot n^5+6904026511616\cdot n^6+20215755160896\cdot n^7+48035486487104\cdot n^8 \\ & +93817776256832\cdot n^9+151961459800128\cdot n^{10}+205363007155392\cdot n^{11}+232395627484608\cdot n^{12} \\ & +220546391942592\cdot n^{13}+175425566746048\cdot n^{14}+116634920972032\cdot n^{15}+64488148739328\cdot n^{16} \\ & +29413823444992\cdot n^{17}+10937158309888\cdot n^{18}+3259238326272\cdot n^{19}+759170949120\cdot n^{20} \\ & +133078892544\cdot n^{21}+16502538240\cdot n^{22}+1289945088\cdot n^{23}+47775744\cdot n^{24})\cdot a_{n+1} \\ & +486\cdot(1+n)\cdot(1+2\cdot n)^5\cdot(1+4\cdot n)\cdot(1944351+14979384\cdot n+52242624\cdot n^2 \\ & +109737216\cdot n^3+155030016\cdot n^4+155330368\cdot n^5+113205728\cdot n^6+60469320\cdot n^7 \\ & +23494256\cdot n^8+6475072\cdot n^9+1201536\cdot n^{10}+134784\cdot n^{11}+6912\cdot n^{12})\cdot a_n=0\end{aligned} \tag{5.50}$$

Of course, we obtain the same equation found in [Koepf, 2014, Section 9.14], as expected. Despite the fact that we are not using Maple and that the computers used do not have the same capacities (RAM memory and processor speed), let us run our implementation of Petkovšek's algorithm for this example as well.

```
(%i4) HyperPetkov(RE,a[n]);
```

Evaluation took 17.7400 seconds (18.4800 elapsed)

$$(\%o4) \quad \left\{\frac{81\cdot(1+4\cdot n)}{4\cdot(1+n)^3\cdot(1+3\cdot n)^4}, \frac{6\cdot(1+n)\cdot(1+2\cdot n)^5}{(3\cdot n+1)\cdot(3+4\cdot n)^3}\right\}$$

The solutions can be checked by computing the ratios of the two given hypergeometric terms as follows.

```
(%i5) ratio(term1,n);
```

$$(\%o5)\quad \frac{6\cdot(1+n)\cdot(1+2\cdot n)^5}{(3\cdot n+1)\cdot(3+4\cdot n)^3}$$

```
(%i6) ratio(term2,n);
```

$$(\%o6)\quad \frac{81\cdot(1+4\cdot n)}{4\cdot(1+n)^3\cdot(1+3\cdot n)^4}$$

The obtained timing is lower than the one in [Koepf, 2014, Section 9.14]. This is because the implementation used in that book did not consider the impact of the difference of the degrees of monic factors of the leading and trailing polynomial coefficients. The Maxima command *solve_rec* uses this remark as well, and for this example we get the following output which this time is given without a warning message since the order of the holonomic RE equals the number of linearly independent hypergeometric term solutions.

```
(%i7) solve_rec(RE,a[n]);
```

Evaluation took 97.1700 seconds (98.1100 elapsed)

$$(\%o7)\quad a_n=\frac{\Gamma\left(\frac{1}{3}\right)\cdot\Gamma\left(\frac{3}{4}\right)^3\cdot\%k_1\cdot 6^n\cdot 2^{5\cdot n}\cdot n!\cdot\Gamma\left(\frac{1+2\cdot n}{2}\right)^5}{\pi^{\frac{5}{2}}\cdot 4^{3\cdot n}\cdot 3^n\cdot\Gamma\left(\frac{1+3\cdot n}{3}\right)\cdot\Gamma\left(\frac{3+4\cdot n}{4}\right)^3}+\frac{\Gamma\left(\frac{1}{3}\right)^4\cdot\%k_2\cdot 81^n\cdot\Gamma\left(\frac{1+4\cdot n}{4}\right)}{\Gamma\left(\frac{1}{4}\right)\cdot 3^{4\cdot n}\cdot n!^3\cdot\Gamma\left(\frac{1+3\cdot n}{3}\right)^4}$$

```
(%i8) ratio((gamma(1/3)*gamma(3/4)^3*%k[1]*6^n*2^(5*n)*n!
      *gamma((1+2*n)/2)^5)/(%pi^(5/2)*4^(3*n)*3^n
      *gamma((1+3*n)/3)*gamma((3+4*n)/4)^3),n);
```

$$(\%o8)\quad \frac{6\cdot(1+n)\cdot(1+2\cdot n)^5}{(3\cdot n+1)\cdot(3+4\cdot n)^3}$$

```
(%i9) ratio((gamma(1/3)^4*%k[2]*81^n*gamma((1+4*n)/4))
      /(gamma(1/4)*3^(4*n)*n!^3*gamma((1+3*n)/3)^4),n);
```

$$(\%o9)\quad \frac{81\cdot(1+4\cdot n)}{4\cdot(1+n)^3\cdot(1+3\cdot n)^4}$$

Let us consider one last example. We add another hypergeometric term to look for a more complicated holonomic RE of third order.

```
(%i10) term3:pochhammer(1/5,n)^2/(pochhammer(1/7,n)
       *pochhammer(2,n));
```

$$(\%o10)\quad \frac{\left(\frac{1}{5}\right)_n^{\,2}}{\left(\frac{1}{7}\right)_n\cdot(2)_n}$$

```
(%i11) RE:sumhyperRE([term1,term2,term3],a[n])$
```

Evaluation took 0.7200 seconds (1.0400 elapsed)

This time we do not display the corresponding recurrence equation since it is very big. *HyperPetkov* finds the corresponding solution ratios after about 20 minutes as presented below.

```
(%i12) HyperPetkov(RE,a[n]);
```

Evaluation took 1220.5300 seconds (1223.7700 elapsed)

$$(\%o11) \quad \left\{ \frac{81\cdot(1+4\cdot n)}{4\cdot(1+n)^3\cdot(1+3\cdot n)^4}, \frac{6\cdot(1+n)\cdot(1+2\cdot n)^5}{(3\cdot n+1)\cdot(3+4\cdot n)^3}, \frac{7\cdot(1+5\cdot n)^2}{25\cdot(n+2)\cdot(7\cdot n+1)} \right\}$$

After about 6 hours, the command *solve_rec* keeps running.

At the end of this chapter a clear understanding of Petkovšek's algorithm is obtained. We can deduce from the presented examples, in particular (5.50), that our implementation of Algorithm 5 presents some gain of efficiency through the use of a heuristic filter on the difference of the degrees of monic factors of the leading and trailing coefficients for a given holonomic recurrence equation [Petkovšek, 1992, Remark 4.1]. Compared to the Maxima package *solve_rec*, we have seen that the latter package has some unstable behaviors that could be corrected easily even though in terms of efficiency our implementation seems to win. Nevertheless, although robust, it is helpful to move on to an algorithm which reduces the computation timing in the worst-case time complexity. In the following chapter we will discuss a much more efficient algorithm for finding hypergeometric term solutions of holonomic REs. We will also show how to get "simple" formulas of hypergeometric terms by means of Pochhammer and factorial symbols.

Chapter 6

A Variant of van Hoeij's Algorithm

"The first role of imagination for a mathematician is to create mental images."
Alain Connes.

Compared to Petkovšek's approach which uses familiar algebra tools in summation theory to compute hypergeometric term solutions of holonomic recurrence equations, Mark van Hoeij introduced some other notions like finite singularities and valuation growths in his algorithm. Nevertheless, one could see van Hoeij's approach as a fast improved version of the one of Petkovšek. Indeed, for the question "which are the hypergeometric term solutions of a holonomic recurrence equation?", Petkovšek's algorithm tells us that the normal forms (5.4) of their ratios can be built from the monic polynomial factors of the recurrence equation's leading and trailing terms; whereas van Hoeij's algorithm reduces that set modulo certain properties. One needs to focus on notions about apparent and semi-apparent singularities, linear operators and the Newton polygon algorithm to get what the algorithm really does ([Van Hoeij, 1999], [Cluzeau and van Hoeij, 2006]). However, this is not what we will do, rather, also by considering local behaviors of hypergeometric term solutions of holonomic REs, without any explicit use of the Newton polygon algorithm, we similarly construct these solution terms as van Hoeij's algorithm does.

This chapter is divided into two sections. In the first part, we give an algorithm to compute "simple" formulas of hypergeometric terms by mean of factorials and Pochhammer symbols, and the second section is devoted to our variant of van Hoeij's algorithm which incorporates the algorithm of the first section as one of its steps.

6.1 Computing A "Simple" Formula of A Hypergeometric Term

In this section, we explain how the general coefficient of a hypergeometric type power series is computed. The corresponding algorithm works together with our description of van Hoeij's approach in computing hypergeometric term solutions of holonomic REs in the next section, which indeed reduces the rational functions to be considered here to a particular case. A similar algorithm is described in [Koepf, 2014, Algorithm 2.8] but with the goal of finding the hypergeometric representations. Here, however, we give more emphasizes on the computations of formulas as in [Koepf, 1995b] and we assume that the given ratios are related to hypergeometric or hypergeometric type terms. Our goal is to compute "simple" formulas as much as we can by using factorial and Pochhammer symbols.

We mention that the definition of a "simple" formula cannot be properly stated. The desire is to reduce unfamiliar objects and avoid complicated representations. We could say that a formula is considered to be "simple" when it presents more familiar objects from mathematical dictionaries in a reduced form. In the sense of computing formulas of hypergeometric terms, this consists in simplifying as much as possible, Pochhammer symbols to rational multiples of factorials with integer-linear arguments.

We consider a rational function

$$r(k) := \frac{P(k)}{Q(k)},\ P(k), Q(k) \in \mathbb{K}[k],\ Q(k) \neq 0 \text{ for integers } k \geqslant 0 \tag{6.1}$$

such that P and Q do not have non-negative integer roots. Observe that, if there are non-negative integer zeros of P and Q, then r may be shifted to $r(k + k_M)$, where $k_M = \max\{k \in \mathbb{N}_{\geqslant 0} : Q(k) \cdot P(k) = 0\}$. However, our approach ensures that all rational functions used in this section for the computations of "simple" formulas of hypergeometric terms do not have non-negative integer zeros and poles.

The hypergeometric terms are taken with the representation

$$a_{k+1} = r(k) a_k, \quad \text{for integer } k \geqslant 0. \tag{6.2}$$

Computing a "simple" formula of such terms is to find their general expressions a_n for a positive integer n provided that their corresponding initial values a_0 are given. That is the result of the product

$$\prod_{k=0}^{n-1} r(k). \tag{6.3}$$

For that purpose, the first step is to factorize r in the form of linear factors as follows

$$r(k) = C \frac{(k + a_1)(k + a_2) \cdots (k + a_p)}{(k + b_1)(k + b_2) \cdots (k + b_q)}, \tag{6.4}$$

where p and q are, respectively, the degrees of P and Q, C is a constant, and the $-a_i$ and $-b_i$ are the zeros and poles of r, respectively. From the Pochhammer symbol definition, using (6.4) one can see that

$$\prod_{k=0}^{n-1} r(k) = C^n \frac{(a_1)_n (a_2)_n \cdots (a_p)_n}{(b_1)_n (b_2)_n \cdots (b_q)_n}. \tag{6.5}$$

Now, we are going to use a property of the Gamma function (see Chapter 1 in [Koepf, 2014]). The Pochhammer symbol of a constant x can also be given by

$$(x)_n = \frac{\Gamma(x + n)}{\Gamma(x)}, \tag{6.6}$$

and as $\Gamma(1) = 1$ we have $\Gamma(n) = (n-1)!$ for any positive integer. Thus in our algorithm, the Pochhammer symbols in (6.5) can be converted into factorial symbols by using the Gamma transformation. However, this conversion can bring inconvenient results. In Maxima when $(x)_n$ is replaced by $\frac{\Gamma(x+n)}{\Gamma(x)}$, $\Gamma(x)$ is immediately computed since the knowledge that $n \in \mathbb{Z}$ is disregarded. For example

```
(%i1) makefact(makegamma(pochhammer(3/2,n)));
```

$$(\%o1) \qquad \frac{2\cdot\left(\frac{1}{2}+n\right)!}{\sqrt{\pi}} \tag{6.7}$$

cannot be used to replace $(\frac{3}{2})_n$ due to the computation of $\Gamma(\frac{3}{2})$. The function *makefact()* only reacted on $\Gamma(n+\frac{3}{2})$ and the obtained result is not as simple as we want. Therefore we must describe an algorithmic approach to simplify some individual Pochhammer symbols.

Let α be a constant representing any of the a_i and b_i in (6.4). We compute $(\alpha)_n$ according to the following cases.

1. If $\alpha \notin \mathbb{Q}$, then generally $(\alpha)_n$ cannot be simplified, in this case no simplification is done.

2. Else if $\alpha \in \mathbb{Q}$ then

   - if $\alpha > 0$ then

     – if $\alpha \in \mathbb{N}$, then

$$\begin{aligned}(\alpha)_n &= \alpha\cdot(\alpha+1)\cdots(\alpha+n-1)\\ &= \frac{(\alpha+n-1)!}{(\alpha-1)!}\end{aligned} \tag{6.8}$$

     – Else if α has a denominator equal to 2, then let $s \in \mathbb{N}$ such that $\alpha = \frac{s}{2}$ (see similar computations in [Koepf, 1995b, Page 19]. s is necessarily an odd integer since $\alpha \notin \mathbb{N}$. We set $s = 2t+1$, $t \in \mathbb{N}$, then it follows that

$$\begin{aligned}(\alpha)_n = \left(\frac{s}{2}\right)_n &= \frac{s}{2}\cdot\left(\frac{s}{2}+1\right)\cdots\left(\frac{s}{2}+n-1\right)\\ &= \frac{s\cdot(s+2)\cdot(s+2\cdot 2)\cdots(s+2\cdot(n-1))}{2^n}\\ &= \frac{(2t+1)\cdot(2(t+1)+1)\cdot(2(t+2)+1)\cdots(2(t+n-1)+1)}{2^n}\\ &= \frac{(2\,(t+n))!}{(2t)!\cdot(2t+2)\cdot(2(t+1)+2)\cdots(2(t+n-1)+2)\cdot 2^n}\\ &= \frac{(2\,(t+n))!}{(2t)!\cdot(t+1)\cdot(t+2)\cdots(t+n)\cdot 4^n}\\ &= \frac{(2(t+n))!}{(2t)!4^n\binom{t+n}{n}n!}.\end{aligned} \tag{6.9}$$

     – Else α is a rational number with denominator not equal to 2. In this case $(\alpha)_n$ is kept again since simplifications might carry out more complicated formulas.

- Else α is a negative rational number which is not an integer since r is assumed to not have non-negative integer zeros and poles. Then let u be the least positive integer such that $\alpha + u > 0$. In fact $u = -\lfloor \alpha \rfloor$, that is the negative of the floor of α. Indeed, since α is not an integer,

  $$\Big(\lfloor \alpha \rfloor < \alpha \leqslant 0 \text{ and } \alpha < \lfloor \alpha \rfloor + 1\Big) \Rightarrow \Big(0 < \alpha - \lfloor \alpha \rfloor < 1, \text{ with } -\lfloor \alpha \rfloor \in \mathbb{N}\Big).$$

  Whence we can write

  $$\begin{aligned} (\alpha)_n &= \alpha \cdots (\alpha + u - 1) \cdot (\alpha + u) \cdots (\alpha + n - 1) \\ &= \frac{(\alpha)_u \cdot (\alpha + u)_n}{(\alpha + n) \cdots (\alpha + n + u - 1)} \\ &= \frac{(\alpha)_u}{(\alpha + n)_u} \cdot (\alpha + u)_n. \end{aligned} \tag{6.10}$$

  Therefore the process is repeated for $\alpha + u$ which is a positive rational number, and the final result is multiplied by $\frac{(\alpha)_u}{(\alpha+n)_u}$ which is already simplified since u is well known.

In a nutshell we have the following algorithm.

Algorithm 6 Pochhammersimp(α, n): Simplification of $(\alpha)_n$

Input: A constant α and a variable n.
Output: A simplification of $(\alpha)_n$ (that could be itself).

1. If $\alpha \notin \mathbb{Q}$ then return $(\alpha)_n$.
2. If $\alpha \in \mathbb{Q}$ and $\alpha > 0$ then
   (a) if $\alpha \in \mathbb{N}$ then return
   $$\frac{(\alpha + n - 1)!}{(\alpha - 1)!}, \tag{6.11}$$
   (b) else if $\alpha \in \mathbb{Q}$ with denominator equal to 2, then set $t = \lfloor \alpha \rfloor$ and return
   $$\frac{(2(t + n))!}{(2t)! 4^n \binom{t+n}{t} n!}, \tag{6.12}$$
   (c) else return $(\alpha)_n$.
3. Else set $u = -\lfloor \alpha \rfloor$ and return
   $$\frac{(\alpha)_u}{(\alpha + n)_u} \cdot \text{Pochhammersimp}(\alpha + u, n). \tag{6.13}$$

The following Maxima code simplifies a Pochhammer symbol according to the above description.

```
pochhammersimp(alpha,n):= block([q,u,t],
    if(not (constantp(alpha) and atom(n)
        and not(constantp(n)))) then false,
    if(not ratnump(alpha)) then pochhammer(alpha,n)
    elseif(alpha>0) then (
        if(integerp(alpha)) then (alpha+n-1)!/(\alpha-1)!
        else (
            q: denom(alpha),
            if(q#2) then pochhammer(alpha,n)
            else(
                t: floor(alpha),
                minfactorial(makefact((2*(t+n))!/((2*t)!
                *binomial(t+n,n)*n!*4^n)))
            )
        )
    )
    else(
           u: ceiling(abs(alpha)),
           pochhammersimp(alpha+u,n)
           *pochhammer(alpha,u)/pochhammer(alpha+n,u)
    )
)$
```

Example 6.1.

```
(%i2) pochhammersimp(3/2,n);
```

$$(\%o2) \qquad \frac{(2\cdot(1+n))!}{2\cdot(n+1)\cdot 4^n\cdot n!}$$

Observe that this is another representation of (6.7).

```
(%i3) pochhammersimp(1/2,n);
```

$$(\%o3) \qquad \frac{(2\cdot n)!}{4^n\cdot n!}$$

```
(%i4) pochhammersimp(-9/2,n);
```

$$(\%o4) \qquad -\frac{945\cdot 4^{-2-n}\cdot(2\cdot n)!}{2\cdot\left(n-\frac{9}{2}\right)\cdot\left(n-\frac{7}{2}\right)\cdot\left(n-\frac{5}{2}\right)\cdot\left(n-\frac{3}{2}\right)\cdot\left(n-\frac{1}{2}\right)\cdot n!}$$

```
(%i5) pochhammersimp(-7/4,n);
```

$$(\%o5) \qquad \frac{21\cdot\left(\frac{1}{4}\right)_n}{16\cdot\left(n-\frac{7}{4}\right)\cdot\left(n-\frac{3}{4}\right)}$$

```
(%i6) pochhammersimp(4,n);
```

$$(\%o6) \qquad \frac{(3+n)!}{6}$$

Thus we get quite nice representations of certain Pochhammer symbols through Algorithm 6.

On the other hand, if we consider two numbers x and y such that $x - y = j \in \mathbb{N}$, then we have

$$\frac{(y)_n}{(x)_n} = \frac{\frac{\Gamma(y+n)}{\Gamma(x+n)}}{\frac{\Gamma(y)}{\Gamma(x)}} = \frac{\frac{(y-1+n)!}{(x-1+n)!}}{\frac{(y-1)!}{(x-1)!}}$$

$$= \frac{\frac{(y-1+n)!}{(y+j-1+n)!}}{\frac{(y-1)!}{(y+j-1)!}}$$

$$= \frac{y(y+1)\cdots(y+j-1)}{(y+n)(y+n+1)\cdots(y+n+j-1)}$$

$$= \frac{(y)_j}{(y+n)_j}. \tag{6.14}$$

Example 6.2. *Since* $\frac{3}{2} - \frac{1}{2} = 1$, *then*

$$\frac{\left(\frac{1}{2}\right)_n}{\left(\frac{3}{2}\right)_n} = \frac{\left(\frac{1}{2}\right)_1}{\left(\frac{1}{2}+n\right)_1} = \frac{1}{2n+1},$$

Generally for any numbers x and y such that $x - y = j \in \mathbb{Z}$, we have

$$\frac{(y)_n}{(x)_n} = \begin{cases} \dfrac{(y)_j}{(y+n)_j} & \text{if } j > 0 \\ \dfrac{(x+n)_{-j}}{(x)_{-j}} & \text{if } j < 0 \end{cases}. \tag{6.15}$$

This shows that differences between the zeros and the poles of r in (6.4) should be checked before applications of Algorithm 6 in order to apply (6.15) which can simplify two Pochhammer symbols at the same time. Such computations are given in detail in [Koepf, 2014, Algorithm 2.2] which decides the rationality of the ratio of consecutive terms of a given expression involving Pochhammer, factorial or Gamma symbols. Fortunately, these nice computations can be done by Maxima when we combine *makegamma()*, *makefact()*, *minfactorial()* and *factor()* as below.

```
(%i8) r1:pochhammer(7/3,n)/pochhammer(1/3,n);
      r2:pochhammer(3/5,n)/pochhammer(13/5,n);
```

(%o7) $$\frac{\left(\frac{7}{3}\right)_n}{\left(\frac{1}{3}\right)_n}$$

(%o8) $$\frac{\left(\frac{3}{5}\right)_n}{\left(\frac{13}{5}\right)_n}$$

```
(%i9) factor(minfactorial(makefact(makegamma(r1))));
```

(%o9) $$\frac{(1+3\cdot n)\cdot(4+3\cdot n)}{4}$$

```
(%i10) factor(minfactorial(makefact(makegamma(r2))));
```

$$(\%o10) \quad \frac{24}{(5\cdot n+3)\cdot(5\cdot n+8)}$$

More generally, one can find a "simple" formula of a hypergeometric term having a characteristic ratio with non-negative integer roots and poles by applying the following algorithm.

Algorithm 7 Pochfactosimp(r,n): computing $\prod_{k=1}^{n-1} r(k)$

Input: A rational function $r := r(n)$ and a variable n.
Output: A formula of $\prod_{k=1}^{n-1} r(k)$ in terms of factorial and Pochhammer symbols.

1. Factorize r and write it in terms of linear factors and set

$$h := r = C\frac{(n+a_1)(n+a_2)\cdots(n+a_p)}{(n+b_1)(n+b_2)\cdots(n+b_q)}. \tag{6.16}$$

2. Substitute C by C^n in h.
3. For each a_i, $i=1,\ldots,p$ do
   (a) if there is $b_j,\ j\in[\![1,q]\!]$ in h such that $a_i-b_j\in\mathbb{Z}$ then substitute $\frac{n+a_i}{n+b_i}$ by

$$\begin{cases} \dfrac{(a_i)_{b_j-a_i}}{(a_i+n)_{b_j-a_i}} & \text{if } a_i-b_j<0 \\ \dfrac{(b_j+n)_{a_i-b_j}}{(b_j)_{a_i-b_j}} & \text{if } a_i-b_j>0 \end{cases}, \tag{6.17}$$

   (b) else substitute $(n+a_i)$ by Pochhammersimp(a_i,n) (see Algorithm (6)) in h.
4. Substitute the remaining $b_j,\ j\in[\![1,q]\!]$ (if there are some) in h by Pochhammersimp(b_j,n).
5. Return h.

This algorithm is implemented in our package as *pochfactorsimp(r,n)*. Below we give some examples of computations.

Example 6.3.

```
(%i11) pochfactorsimp(-1/(2*(n+1)*(2*n+1)),n);
```

$$(\%o11) \quad \frac{(-1)^n}{(2\cdot n)!}$$

```
(%i12) pochfactorsimp((2*n+3)^2/((n+1)*(2*n+1)),n);
```

$$(\%o12) \quad \frac{(1+2\cdot n)\cdot 2^{n-1}\cdot(2\cdot(1+n))!}{(n+1)\cdot 4^n\cdot n!^2}$$

```
(%i13) pochfactorsimp(((2*n-%i)*(%i+2*n))/(2*(n+1)*(2*n+1)),n);
```

$$(\%o13) \quad \frac{\left(-\frac{i}{2}\right)_n\cdot\left(\frac{i}{2}\right)_n\cdot 4^n}{(2\cdot n)!}$$

The algorithm can also be used to compute "simple" formulas of certain ratios obtained by using our implementation of Petkovšek's algorithm.

```
(%i14) H:HyperPetkov(FindRE(exp(z)+log(1+z)/z,z,a[n]),a[n]);
```

$$(\%o14)\quad \left\{\frac{1}{n+1}, -\frac{1+n}{n+2}\right\}$$

```
(%i15) map(lambda([r],pochfactorsimp(r,n)),H);
```

$$(\%o15)\quad \left\{\frac{(-1)^n}{n+1}, \frac{1}{n!}\right\}$$

```
(%i16) H:HyperPetkov(FindRE(sin(z)+sqrt(1+z),z,a[n]),a[n],C);
```

$$(\%o16)\quad \left\{-\frac{2\cdot n-1}{2\cdot(n+1)}, -\frac{i}{n+1}, \frac{i}{n+1}\right\}$$

```
(%i17) map(lambda([r],pochfactorsimp(r,n)),H);
```

$$(\%o17)\quad \left\{\frac{(-i)^n}{n!}, \frac{(-1)^{\frac{n}{2}}}{n!}, -\frac{(-1)^n\cdot(2\cdot n)!}{(2\cdot n-1)\cdot 4^n\cdot n!^2}\right\}$$

Now that we have described an algorithm for computing "simple" formulas of hypergeometric terms given their ratios with no non-negative integer zeros and poles, let us move to the main algorithm of this chapter where such rational functions are computed.

6.2 Computing Hypergeometric Term Solutions of Holonomic Recurrence Equations

Let us consider the recurrence equation

$$P_d(n)a_{n+d} + P_{d-1}(n)a_{n+d-1} + \cdots P_1(n)a_{n+1} + P_0(n)a_n = 0, \tag{6.18}$$

with polynomials $P_i(n) \in \mathbb{K}[n], i = 0, \ldots, d$ such that $P_0(n)\cdot P_d(n) \neq 0$. $\mathbb{K}$ is a field of characteristic zero.

Remember in ((5.49), on p. 87) we have seen how to compute a holonomic recurrence equation of lowest order for a given number of linearly independent hypergeometric terms. Any computed hypergeometric term solution of such a holonomic RE is a linear combination of these linearly independent hypergeometric terms considered. The algorithm of this section is a kind of reverse process which for a given holonomic recurrence equation (6.18) computes a basis of at most d hypergeometric terms of the set of all hypergeometric term solutions of (6.18).

In the sequel, we would like to present a variant of van Hoeij's algorithm that finds a basis of all hypergeometric term solutions of (6.18). In particular, this will give every solution of (6.18) as a linear combination of hypergeometric terms.

In the first place, we look for a particular representation of a hypergeometric term. Let $a_n, n \in \mathbb{N}_{\geqslant 0}$, be a hypergeometric sequence such that

$$r(n) = \frac{a_{n+1}}{a_n} \in \mathbb{K}(n). \tag{6.19}$$

Then we have

$$\frac{a_1}{a_0} = r(0),\ \frac{a_2}{a_1} = r(1),\ \ldots,\ \frac{a_n}{a_{n-1}} = r(n-1),\ \ n \geqslant 1,$$

and therefore

$$\frac{a_n}{a_0} = \prod_{k=0}^{n-1} \frac{a_{k+1}}{a_k} = \prod_{k=0}^{n-1} r(k) \Rightarrow a_n = a_0 \prod_{k=0}^{n-1} r(k). \tag{6.20}$$

Factorizing $r(n)$ over $\mathbb{K}$ gives

$$r(n) = C \frac{\prod_{i=1}^{I}(n-\alpha_i)}{\prod_{j=1}^{J}(n-\beta_j)}, \tag{6.21}$$

where C is a constant representing the ratio of the leading coefficients of the numerator and the denominator of r. Note that some α_i and β_j may coincide in (6.21). Thus combining (6.20) and (6.21) we obtain

$$a_n = a_0 \cdot C^n \cdot \prod_{k=0}^{n-1} \frac{\prod_{i=1}^{I}(k-\alpha_i)}{\prod_{j=1}^{J}(k-\beta_j)}. \tag{6.22}$$

Since $-\alpha_i(-\alpha_i+1)\cdots(-\alpha_i+n-1) = (-\alpha_i)_n$ (see (2.54)) occurs, (6.22) can be rewritten as

$$a_n = \frac{(-\alpha_1)_n(-\alpha_2)_n\cdots(-\alpha_I)_n}{(-\beta_1)_n(-\beta_2)_n\cdots(-\beta_J)_n} \cdot a_0 \cdot C^n. \tag{6.23}$$

Observe moreover that each Pochhammer symbol can be seen modulo $\mathbb{Z}$ in a certain real interval. That is to say that the real parts of the arguments of Pochhammer terms can be chosen belonging to an interval of amplitude 1. This is one of the interesting observations made by van Hoeij. In our case, we choose to rewrite the Pochhammer symbols modulo $\mathbb{Z}$ in $[-1,0)$. Each Pochhammer symbol is then substituted by a polynomial times another Pochhammer term whose argument differs by an integer u. Precisely, let y be a real number (for the case of complex numbers, the computations are applied on their real parts), then its corresponding value in $[-1,0)$ is $u = y - \lfloor y \rfloor - 1$ and we have

$$\begin{aligned}
(y)_n &= \frac{(u)_n \cdot (u+n)\cdots(y+n-1)}{u\cdot(u+1)\cdots(y-1)} \\
&= (u)_n \cdot \frac{(u+n)_{y-u}}{(u)_{y-u}} && (6.24)\\
&= (y-\lfloor y\rfloor -1)_n \cdot \frac{(n+y-\lfloor y\rfloor-1)_{\lfloor y\rfloor+1}}{(y-\lfloor y\rfloor-1)_{\lfloor y\rfloor+1}}. && (6.25)
\end{aligned}$$

After applying (6.25) to each Pochhammer symbol in (6.23), the remaining expression will have Pochhammer terms having arguments with real parts in $[-1,0)$. These terms may have more coincidence than the $(-\alpha_i)_n$ and $(-\beta_j)_n$ in (6.23) since all Pochhammer terms in (6.23) whose arguments differ by an integer give the same Pochhammer term modulo $\mathbb{Z}$ after substitution. Therefore there exists a rational function $R(n) \in \mathbb{K}(n)$ and some constant numbers $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_I$, $\tilde{\beta}_1, \ldots, \tilde{\beta}_J$, with real parts in $[-1, 0)$, such that

$$a_n = R(n) \cdot C^n \cdot \frac{(-\tilde{\alpha}_1)_n(-\tilde{\alpha}_2)_n \cdots (-\tilde{\alpha}_I)_n}{(-\tilde{\beta}_1)_n(-\tilde{\beta}_2)_n \cdots (-\tilde{\beta}_J)_n}. \tag{6.26}$$

The constant a_0 is neglected by linearity since we will look for a basis of hypergeometric term solutions of (6.18).

Considering multiplicities e_k over $\mathbb{Z} \setminus \{0\}$ and replacing $-\tilde{\alpha}_i$ and $-\tilde{\beta}_j$, $i \in [\![1, I]\!]$, $j \in [\![1, J]\!]$, by θ_k, we get the general form

$$a_n = R(n) \cdot C^n \cdot \prod_{k=1}^{K} (\theta_k)_n^{e_k} \; (-\theta_k \in \mathbb{K}, \text{ with real part in } [-1, 0)), \tag{6.27}$$

$K \leqslant I + J$. This time all the involved data are uniquely determined. The ratio $r(n)$ can be rewritten as

$$r(n) = \frac{a_{n+1}}{a_n} = \frac{R(n+1)}{R(n)} \cdot C \cdot \prod_{k=1}^{K} (n + \theta_k)^{e_k} \; \in \mathbb{K}(n). \tag{6.28}$$

Mark van Hoeij uses Gamma representations in (6.27) and denotes it singularity structure of a_n (see Chapter 9 in [Koepf, 2014]). This representation can be seen as the end point of the algorithm when it computes an element of the basis of hypergeometric terms looked for. In fact, the goal of computing a basis of all hypergeometric term solutions of (6.18) is equivalent to finding all possible structures (6.27) of solutions of (6.18).

As Petkovšek's algorithm proved, hypergeometric term solutions of (6.18) are built from the monic factors of its leading and its trailing polynomial coefficients $P_d(n-d)$ and $P_0(n-1)$. Thus van Hoeij's approach must also consider these factors modulo $\mathbb{Z}$. Remark that the fact of taking monic factors modulo $\mathbb{Z}$ makes that the shifts $P_d(n) \to P_d(n-d)$ and $P_0(n) \to P_0(n-1)$ are not necessary. Let us take an example. We reuse the holonomic recurrence equation (5.37).

```
(%i1) RE:FindRE(z/(1-z)^4+5*z^3*exp(z)
      +(1+7*z^2)*log(1+z)+(2+z)/sqrt(1+z),z,a[n])$
```

The leading term is

```
(%i2) first(lhs(RE));
```

$$(\%o2) \qquad -5280 \cdot (1+n) \cdot (2+n) \cdot (3+n) \cdot (4+n) \cdot a_{n+4}$$

and the trailing term is

```
(%i3) last(lhs(RE));
```

$$(\%o3) \qquad -882 \cdot (n-19)^2 \cdot (2 \cdot n - 35) \cdot a_{n-17}.$$

Taking the factorization modulo $\mathbb{Z}$ with roots in $[-1, 0)$ gives the monic factors

$$(n+1)^e, \; 0 \leqslant e \leqslant 4,$$

for the leading polynomial coefficient, and

$$\left(n+\frac{1}{2}\right)^{e_1}(n+1)^{e_2} \text{ for } 0 \leqslant e_1 \leqslant 1,\ 0 \leqslant e_2 \leqslant 2,$$

for the trailing one. Therefore the possible ratios of Pochhammer parts in the form (6.27) of hypergeometric term solutions are

$$\begin{aligned} &1, n+\frac{1}{2}, n+1, \left(n+\frac{1}{2}\right)\cdot(n+1), \frac{1}{(1+n)^4}, \frac{\frac{1}{2}+n}{(1+n)^4}, \frac{1}{(1+n)^3}, \\ &\frac{\frac{1}{2}+n}{(1+n)^3}, \frac{1}{(1+n)^2}, \frac{\frac{1}{2}+n}{(1+n)^2}, \frac{1}{n+1}, \frac{\frac{1}{2}+n}{n+1}, (1+n)^2, \left(n+\frac{1}{2}\right)\cdot(1+n)^2. \end{aligned} \tag{6.29}$$

Observe that none of these ratios has a non-negative integer zero or pole, hence the type of rational function that we treat with Algorithm 7. This is made possible by the fact that we consider factorization modulo $\mathbb{Z}$ in $[-1,0)$.

At this stage one can already see an advantage of van Hoeij's approach. Indeed, in the worst-case complexity we only have 14 choices for the possible hypergeometric term solutions as listed in (6.29) while with Petkovšek's algorithm one has $2\cdot2\cdot2\cdot2\cdot3\cdot2=96$ choices that correspond to the number of combinations of monic factors of the trailing and the leading terms.

Moreover, not all the ratios in (6.29) should be considered because the exponents of each linear factor appearing in the possible ratios of hypergeometric term solutions can be bounded from the given holonomic recurrence equation. For this purpose van Hoeij's algorithm uses the notion of valuation growth or local types of difference operators at finite singularities [Van Hoeij, 1999, Definition 9]. A point $\alpha+\mathbb{Z}, \alpha\in\mathbb{K}$ is called finite singularity of (6.18) if there exists $\tau\in\mathbb{Z}$ such that $\alpha+\tau$ is a root of the polynomial $P_d(n-d)\cdot P_0(n)$ [Cluzeau and van Hoeij, 2006, Definition 8]. Such a point is simply a root modulo $\mathbb{Z}$ of the trailing or the leading polynomial coefficient of (6.18) as we considered.

Since we are already computing ratios of Pochhammer parts of hypergeometric term solutions, we proceed in a slightly different way than the one described in ([Van Hoeij, 1999], [Cluzeau and van Hoeij, 2006]) for the computation of the exponent bounds at a finite singularity. To determine the valuation growths of finite singularities, we have to use the minimum exponent (or valuation) taken by the corresponding factors modulo $\mathbb{Z}$ in the trailing and the leading polynomial coefficients as lower bounds.

Thus if we come back to our previous example, for the leading term, the monic factors modulo $\mathbb{Z}$ in $[-1,0)$ to be considered reduce to

$$(n+1)^e,\ e=1,2,3,4, \tag{6.30}$$

and for the trailing term we get

$$\left(n+\frac{1}{2}\right)^{e_1}\cdot(n+1)^{e_2},\ e_1=1,\ e_2=2. \tag{6.31}$$

This second consideration on monic factors of the leading and the trailing polynomial coefficients does not decrease the number of candidates in this example though, because of the fact that the leading and trailing polynomial coefficients share the same singularity from the factor $(n+1)$, which makes this example relatively less simple. Nevertheless, with the remaining part of the algorithm we will see that the gain of efficiency compare to Algorithm 5 is even better.

Once the ratios of Pochhammer parts in the form (6.27) of hypergeometric term solutions of (6.18) are found, Algorithm 7 is used to compute these Pochhammer parts (C^n included). And thus the representations of the form (6.27) are obtained after the computations of the corresponding rational functions R whose numerators and denominators are deduced from the holonomic RE and the ratios of the corresponding Pochhammer parts.

Furthermore, van Hoeij's algorithm goes even further. Below we give an example whose three iterations with Petkovšek's algorithm represent only a single iteration in the van Hoeij approach. Let us consider the three following hypergeometric terms.

```
(%i4) term1:(1+n)/n!;
```

$$(\%o4) \qquad \frac{1+n}{n!}$$

```
(%i5) term2:(1+n+n^2)/n!;
```

$$(\%o5) \qquad \frac{1+n+n^2}{n!}$$

```
(%i6) term3:(1+n+n^2+n^3)/n!;
```

$$(\%o6) \qquad \frac{1+n+n^2+n^3}{n!}$$

We use *sumhyperRE* to compute a holonomic recurrence equation valid for all linear combinations of these hypergeometric terms.

```
(%i7) RE:sumhyperRE([term1,term2,term3],a[n]);
```

$$(\%o7) \qquad -(1+n)\cdot(2+n)\cdot(3+n)\cdot\left(2+8\cdot n+6\cdot n^2+n^3\right)\cdot a_{n+3}$$
$$+3\cdot(1+n)\cdot(2+n)\cdot\left(3+11\cdot n+7\cdot n^2+n^3\right)\cdot a_{n+2}-3\cdot(1+n)\cdot\left(6+16\cdot n+8\cdot n^2+n^3\right)\cdot a_{n+1}$$
$$+\left(17+23\cdot n+9\cdot n^2+n^3\right)\cdot a_n=0 \quad (6.32)$$

Looking for hypergeometric term solutions over $\mathbb{Q}$, considering the possible exponents, the possible ratios for their Pochhammer parts are only

$$1, \frac{1}{n+1}, \frac{1}{(1+n)^2}, \frac{1}{(1+n)^3}. \tag{6.33}$$

As one can observe, a basis for these three hypergeometric terms is

$$\left\{\frac{1+n}{n!}, \frac{n^2}{n!}, \frac{n^3}{n!}\right\}. \tag{6.34}$$

Each element of this basis is a multiple of $\frac{1}{n!}$ over $\mathbb{Q}(n)$. That is why in van Hoeij's approach the only ratio used for the computation of hypergeometric term solutions of (6.32) is $\frac{1}{n+1}$. Indeed, whereas Petkovšek's algorithm tries to compute the ratio of hypergeometric term solutions with all monic factors of the leading and the trailing terms of a given holonomic recurrence equation, that is 32 cases with (6.32) if we only consider the field of rational numbers, van Hoeij's approach does not only reduce the number of cases modulo $\mathbb{Z}$ with their corresponding possible exponents, but it also filters the set of ratio terms of Pochhammer parts of hypergeometric term solutions of that equation by using a characteristic property of its hypergeometric term solutions. Next, we explain what this characteristic property of hypergeometric term solutions of holonomic recurrence equations is about.

It is a study of the behavior of the ratio $r(n)$ at infinity. Indeed at ∞ we can write

$$r(n) = c \cdot n^{\nu} \cdot \left(1 + \frac{b}{n} + O\left(\frac{1}{n^2}\right)\right), \tag{6.35}$$

with the unique triple (ν, c, b) called the local type of a_n at ∞. Without ambiguity, we will more often use the word local type instead of local type at infinity.

Theorem 6.1 (Fuchs Relations). *Let $R(n) = \frac{N(n)}{U(n)}$ with $N(n), U(n) \in \mathbb{K}[n]$. The following relations between the local type of a hypergeometric term a_n given by (6.27) hold:*

i. $\nu = \sum_{k=1}^{K} e_k$,

ii. $b = \sum_{k=1}^{K} \theta_k\, e_k \; + \deg(N(n)) - \deg(U(n))$,

iii. $c = C$,

where (ν, c, b) denotes the local type of a_n at ∞.

Proof. From (6.28) we know that

$$r(n) = \frac{a_{n+1}}{a_n} = C \cdot \left(\frac{R(n+1)}{R(n)} \cdot \prod_{k=1}^{K} (n + \theta_k)^{e_k}\right). \tag{6.36}$$

We would like to compute a truncated asymptotic expansion of (6.36). This can be seen as the result of the product of asymptotic expansions of the form (6.35) of $\frac{R(n+1)}{R(n)}$ and $\prod_{k=1}^{K}(n+\theta_k)^{e_k}$ times C. Since $R(n) = \frac{N(n)}{U(n)}$, the highest degree of n in its asymptotic expansion is $\delta = \deg(N(n)) - \deg(U(n))$. Hence we read as

$$R(n) = c_R \cdot n^{\delta} \cdot \left(1 + \frac{b_R}{n} + O\left(\frac{1}{n^2}\right)\right), \tag{6.37}$$

for some constants c_R, b_R. Let us now deduce a truncated asymptotic expansion of $R(n+1)$.

$$\begin{aligned} R(n+1) &= c_R \cdot (n+1)^\delta \cdot \left(1 + \frac{b_R}{n+1} + O\left(\frac{1}{n^2}\right)\right) \\ &= c_R \cdot n^\delta \left(1+\frac{1}{n}\right)^\delta \cdot \left(1 + \frac{b_R}{n\left(1+\frac{1}{n}\right)} + O\left(\frac{1}{n^2}\right)\right) \\ &= c_R \cdot n^\delta \left(1 + \frac{\delta}{n} + \sum_{j=2}^{\delta} \binom{\delta}{j} \left(\frac{1}{n}\right)^j\right) \cdot \left(1 + \frac{b_R}{n} + O\left(\frac{1}{n^2}\right)\right) \\ &= c_R \cdot n^\delta \left(1 + \frac{b_R+\delta}{n} + O\left(\frac{1}{n^2}\right)\right). \end{aligned} \tag{6.38}$$

Thus from (6.37) and (6.38) the first order asymptotic expansion of $\frac{R(n+1)}{R(n)}$ yields

$$\begin{aligned} \frac{R(n+1)}{R(n)} &= \frac{1 + \frac{b_R+\delta}{n} + O\left(\frac{1}{n^2}\right)}{1 + \frac{b_R}{n} + O\left(\frac{1}{n^2}\right)} \\ &= \left(1 + \frac{b_R+\delta}{n} + O\left(\frac{1}{n^2}\right)\right) \cdot \left(1 - \frac{b_R}{n} + O\left(\frac{1}{n^2}\right)\right) \\ &= 1 + \frac{\delta}{n} + O\left(\frac{1}{n^2}\right) = 1 + \frac{\deg(N(n)) - \deg(U(n))}{n} + O\left(\frac{1}{n^2}\right). \end{aligned} \tag{6.39}$$

On the other hand

$$\begin{aligned} (n+\theta_k)^{e_k} &= n^{e_k} \cdot \left(1 + \frac{\theta_k}{n}\right)^{e_k} \\ &= n^{e_k} \cdot \left(1 + \frac{\theta_k e_k}{n} + \sum_{j=2}^{e_k} \binom{e_k}{j} \left(\frac{\theta_k}{n}\right)^j\right) \\ &= n^{e_k} \cdot \left(1 + \frac{\theta_k e_k}{n} + O\left(\frac{1}{n^2}\right)\right), \end{aligned} \tag{6.40}$$

therefore

$$\prod_{k=1}^{K} (n+\theta_k)^{e_k} = n^{\sum_{k=1}^K e_k} \cdot \left(1 + \frac{\sum_{k=1}^K \theta_k e_k}{n} + O\left(\frac{1}{n^2}\right)\right). \tag{6.41}$$

Finally according to (6.36), the expansion searched for is obtained by the product of (6.39) and (6.41) times C. That is

$$\begin{aligned} r(n) &= C \cdot n^{\sum_{k=1}^K e_k} \cdot \left(1 + \frac{\sum_{k=1}^K \theta_k e_k}{n} + O\left(\frac{1}{n^2}\right)\right) \\ &\quad \cdot \left(1 + \frac{\deg(N(n)) - \deg(U(n))}{n} + O\left(\frac{1}{n^2}\right)\right) \\ &= C \cdot n^{\sum_{k=1}^K e_k} \left(1 + \frac{\sum_{k=1}^K \theta_k e_k + \deg(N(n)) - \deg(U(n))}{n} + O\left(\frac{1}{n^2}\right)\right), \end{aligned} \tag{6.42}$$

from which one easily read off the data of the theorem. □

The first two relations in this theorem tell us that for the local type (ν, c, b) of a hypergeometric term a_n, ν and b can be found directly from the ratios that represent its Pochammer part. Indeed, observe that modulo $\mathbb{Z}$, the second relation of the theorem reads as

$$b = \sum_{k=1}^{K} \theta_k \, e_k. \tag{6.43}$$

The third relation cannot be considered for the moment since the rational function candidates for the Pochhammer parts of hypergeometric term solutions of a holonomic RE are all monic.

In the case of factorization in $\mathbb{Q}$, the following Maxima code gives b modulo $\mathbb{Z}$ and ν for a given ratio candidate of the Pochhammer part of a hypergeometric term [Koepf, 2014, Section 9.7].

```
check_localtype(term,n):=block([Rn,nuRn,t,tayRn,cRn,bRn],
    Rn: ratsimp(term),
    nuRn: hipow(num(Rn),n) - hipow(denom(Rn),n),
    Rn: ratsimp(subst(1/t,n,ratsimp(Rn/n^nuRn))),
    tayRn: ratdisrep(taylor(Rn,t,0,2)),
    bRn: coeff(tayRn,t,1),
    [nuRn,bRn-floor(bRn)-1]
)$
```

As mentioned earlier, the map $y \mapsto y - \lfloor y \rfloor - 1$ is used to find the correspondence of y modulo $\mathbb{Z}$ in $[-1, 0)$.

Let us compute the local type for some ratios from (**??**).

```
(%i8)  term:(n+1/2)/(n+1)$

(%i9)  check_localtype(term,n);
```

$$(\%o9) \qquad [0, -\frac{1}{2}]$$

```
(%i10) term:(n+1)^2$

(%i11) check_localtype(term,n);
```

$$(\%o11) \qquad [2, -1]$$

And for the holonomic RE (6.32) we have

```
(%i12) map(lambda([r], check_localtype(r,n)),
       makelist(1/(n+1)^e,e,0,3));
```

$$(\%o12) \qquad [[0, -1], [-1, -1], [-2, -1], [-3, -1]]$$

As indicated by its name, this Maxima function is only used for verification purposes. Indeed it is used to check whether the local type computed from a ratio candidate of the Pochammer part of a hypergeometric term solution of a given holonomic RE, has its ν and b modulo $\mathbb{Z}$ of Theorem 6.1 equal to those of a hypergeometric term solution of that RE. Thus in the next paragraph, we explain how the local types of hypergeometric term solutions of the holonomic recurrence equation (6.18) are computed.

Note that this step must be considered with the highest priority in the algorithm, because if the set of local types of hypergeometric term solutions of a given holonomic recurrence equation is empty, then there is no hypergeometric term solution. This again shows a main advantage of van Hoeij's approach compared to the one of Petkovšek which does not have a quick test to know when a holonomic RE does not have hypergeometric term solutions.

For this step, van Hoeij's algorithm uses the Newton polygon of the difference operator of the given holonomic RE [Van Hoeij, 1999, Section 3]. However, we proceed differently. Our idea is to substitute (6.35) in (5.20) and compute the asymptotic expansion of the left hand side of the given holonomic recurrence equation and find equations for the local types by equating the result to 0. Let us write the recurrence equation as

$$\sum_{i=0}^{d} P_i a_{n+i} = 0, \tag{6.44}$$

for polynomial coefficients $P_i \in \mathbb{K}[n]$, $P_0 \cdot P_d \neq 0$. Let a_n be a hypergeometric term solution of this equation such that $a_{n+1} = r(n) a_n$ for a rational function r. (6.44) can then be written for $r(n)$ as

$$\sum_{i=0}^{d} P_i \prod_{j=0}^{i-1} r(n+i) = 0. \tag{6.45}$$

We consider

$$r(n) = c \cdot n^{\nu} \cdot \left(1 + O\left(\frac{1}{n}\right)\right) \tag{6.46}$$

and we substitute this in (6.45). Similarly as we did in the proof of Theorem 6.1, we make computations that yield the possible values of ν and c. If such values are found, say $(\nu_{\text{cand}}, c_{\text{cand}})$, then we rewrite $r(n)$ as

$$c_{\text{cand}} \cdot n^{\nu_{\text{cand}}} \cdot \left(1 + \frac{b}{n} + O\left(\frac{1}{n^2}\right)\right) \tag{6.47}$$

and we make new computations to find b.

Summarized, our procedure to find the local types (ν, c, b) of hypergeometric term solutions of a given holonomic RE consists in the following items:

1. we compute the possible values for ν;

2. for each value of ν,

   2-a we compute possible values for c,

   2-b for each value found for c, we use ν and c to compute the possible values for b;

   2-c for each value found for b, (ν, c, b) constitutes a local type of a hypergeometric term solution of (6.44).

Let us now explain how each value is computed.

- Computing ν:

  Substitute (6.46) in (6.45) gives the following terms on the left-hand side

$$c^i \cdot n^{i \cdot \nu} \cdot P_i \cdot \left(1 + O\left(\frac{1}{n}\right)\right), \quad (0 \leqslant i \leqslant d) \tag{6.48}$$

  which is equivalent to

$$l_i \cdot c^i \cdot n^{i \cdot \nu + \deg(P_i)} \cdot \left(1 + O\left(\frac{1}{n}\right)\right), \quad (0 \leqslant i \leqslant d) \tag{6.49}$$

  where l_i denotes the leading coefficient of P_i. Since we are dealing with a solution of an equation of the right-hand side 0, the terms having the highest power of n in the asymptotic expansion of the left-hand side of the equation must be zero. However this is only possible if a term of the form (6.49) has the same power of n with some other terms such that they add to 0. Therefore we deduce that the possible candidates for ν are integer solutions of linear equations coming from equalities of powers of n for two different terms of the form (6.49). That is for $0 \leqslant i \neq j \leqslant d$, we have the equation

$$i \cdot \nu + \deg(P_i) = j \cdot \nu + \deg(P_j) \tag{6.50}$$

  and therefore a possible value for ν is

$$\nu_{i,j} = \frac{\deg(P_j) - \deg(P_i)}{i - j}, \tag{6.51}$$

  if the computed value is an integer.

  We then compute $\binom{d}{2}$ such values for the equation (6.44) and keep those that are integers. Note that two different couples of terms may give the same value for ν, meaning that the addition to zero involves all the corresponding terms, which is the point of the next item.

- Computing c:

  Assume that we have found a value $\nu_{i,j} \in \mathbb{Z}$ corresponding to k terms in the equation (6.45) with indices $0 \leqslant u_1 \neq u_2 \neq \ldots \neq u_k \leqslant d$. Then from representation (6.49) it is straightforward to see that a valid candidate for c is a solution of the polynomial equation

$$l_{u_1} \cdot c^{u_1} + l_{u_2} \cdot c^{u_2} + \cdots + l_{u_k} \cdot c^{u_k} = 0. \tag{6.52}$$

  In fact, since the corresponding terms must add to zero in the asymptotic expansion, their leading coefficients must equal zero.

  Thus, each value $c_{i,j} \in \mathbb{K}$ which is a zero of (6.52) for a given $\nu_{i,j}$, $(\nu_{i,j}, c_{i,j})$ is already a possible couple to be completed for the local type of a hypergeometric term solution of (6.44).

- Computing b:

  For a computed couple $(\nu_{i,j}, c_{i,j})$ as explained above, we rewrite $r(n)$ as

  $$c_{i,j} \cdot n^{\nu_{i,j}} \cdot \left(1 + \frac{b}{n} + O\left(\frac{1}{n^2}\right)\right), \tag{6.53}$$

  with unknown b.

  After substituting (6.53) in (6.45) and computing again the asymptotic expansion, the terms with the highest power of n add to zero, and therefore the left-hand side of the resulting equation must have a leading term with a coefficient as a polynomial in the variable b. Since that polynomial must be zero, the possible values for b are its roots. To get the leading coefficient of the asymptotic expansion of the left-hand side of the equation, one computes the first non-zero term of its Taylor expansion at infinity and solve its coefficient equal to zero for the unknown b. Finally if we find values for $b \in \mathbb{K}$ then we have found for each b a local type (ν, c, b) of a hypergeometric term solution of (6.44).

Hence we get the following algorithm.

Algorithm 8 Computing the local types of all hypergeometric term solutions of a given holonomic RE

Input: Polynomials

$$P_i(n) \in \mathbb{K}[n], i = 0, \dots, d \mid P_d(n) \cdot P_0(n) \neq 0$$

Output: The set of all local types of hypergeometric term solutions of the holonomic RE

$$\sum_{i=0}^{d} P_i(n) a_{n+i} = 0. \tag{6.54}$$

1. Set $L = \{\}$.
2. For all pairs $\{i, j\} \in \{0, 1, \dots, d\}$, compute

   $$\nu_{i,j} = \frac{\deg(P_j) - \deg(P_i)}{i - j}. \tag{6.55}$$

3. For each integer $\nu_{i,j}$ computed in (6.55), compute the set of solutions in $\mathbb{K}$, say $S_{c,i,j}$, of the polynomial equation

   $$l_{u_1} \cdot c^{u_1} + l_{u_2} \cdot c^{u_2} + \cdots + l_{u_j} \cdot c^{u_k} = 0, \tag{6.56}$$

   where $l_{u_1}, l_{u_2}, \dots, l_{u_k}$ are the leading coefficients of the polynomials $P_{u_1}, P_{u_2}, \dots, P_{u_k}$, $0 \leqslant u_1 \neq u_2 \neq \dots \neq u_k \leqslant d$ satisfying (6.55) for the same integer $\nu_{i,j}$.

   (a) For each element $c_{i,j}$ of $S_{c,i,j}$ set

   $$r(n) = c_{i,j} \cdot n^{\nu_{i,j}} \cdot \left(1 + \frac{b}{n}\right). \tag{6.57}$$

Algorithm 8 Computing the local types of all hypergeometric term solutions of a given holonomic RE

3. (b) Compute the coefficient $T_{i,j}(b)$ of the first non-zero term of the Taylor expansion of
$$\sum_{i=0}^{d} P_i \prod_{j=0}^{i-1} r(n+i) \tag{6.58}$$
at infinity.
   (c) Solve $T_{i,j}(b) = 0$ in $\mathbb{K}$ for the unknown b and define $S_{b,i,j}$ to be the set of solutions.
   (d) For each element $b_{i,j} \in S_{b,i,j}$, add the triple $(\nu_{i,j}, c_{i,j}, b)$ to L.
4. Return L.

Theorem 6.2. *Algorithm 8 finds all the local types (ν, c, b) of all hypergeometric term solutions of the holonomic recurrence equation (6.18).*

We mention that for a given holonomic recurrence equation, the computations of Algorithm 8 can be used to reduce the number of iterations in the algorithm. The important point to notice is that when two linearly independent hypergeometric term solutions have the same local type, Algorithm 8 compute it at least twice and this fact can be used in such a way that local types computed should be collected as a list instead of a set so that when a basis of hypergeometric term corresponding to a particular local type is found, the latter is discarded from the list of local types. Thus any ratio candidates whose local type is not in the list of local types should not be used in further computation steps. This is a very useful tool to reduce the number of iterations to the number of computed local types in particular when computing the general coefficient of power series whose the holonomic recurrence equations computed using Algorithm 1 and Algorithm 2 are very often of order greater than the corresponding number of linearly independent hypergeometric term solutions. However this tool can only be used when the number of local types computed (with repeated values) is less than the order of the given holonomic RE.

We implemented a Maxima function *localtype(L,n)* which takes the polynomial coefficients of a holonomic RE in L with the indeterminate n. Note that the values found for b are taken modulo $\mathbb{Z}$. Let us see what the local types involved in the previous computed REs of this section are.

```
(%i13) RE:FindRE(z/(1-z)^4+5*z^3*exp(z)
        +(1+7*z^2)*log(1+z)+(2+z)/sqrt(1+z),z,a[n])$
   /* collection of polynomial coefficients (here expanded) */;
(%i14) L:expand(REcoeff(RE,a[n]))$
(%i15) setify(localtype(L,n));
```

$$(\%o15) \quad \left\{[-1,1,-1],[0,-1,-1],[0,-1,-\frac{1}{2}],[0,1,-1]\right\}$$

We use the Maxima command *setify* to remove double elements because in this example the number of local types computed with repeated values (here 31) is greater than the order of the holonomic RE (here 21). Therefore for this example the number of iterations in the algorithm is the number of ratio candidates whose local types are represented in the above set.

According to the Fuchs relations of Theorem 6.1, the local types obtained with *check_localtype* on the ratios of possible Pochhammer parts of hypergeometric term solutions can be compared. We remind that we only compare the value of b and ν, where b is considered modulo $\mathbb{Z}$ as in (6.43). This permits to eliminate some ratios in (6.29). The remaining ratios are therefore

$$1, \frac{1}{n+1}, \frac{n+\frac{1}{2}}{n+1}. \tag{6.59}$$

Note, however, that this corresponds to at most 6 iterations considering the values found for c in the different local types.

For the second example (6.32) the set of local types is

$$\{[-1,1,-1]\}, \tag{6.60}$$

which allows to reduce the number of ratios of possible Pochhammer parts of hypergeometric term solutions to 1. And as we mentioned earlier, the only ratio found for this case is $\frac{1}{n+1}$.

The algorithm goes again further, indeed, once we have found all those better candidates for ratios of Pochammer parts of hypergeometric term solutions, we need to use again the second Fuchs relation from Theorem 6.1 in order to find $\delta = \deg(N(n)) - \deg(U(n))$, where $N(n)$ and $U(n)$ are the numerator and the denominator of R in (6.28). In fact, since we have found values for b and its possible ratio candidates, which means that we can compute $\sum_{k=1}^{K} \theta_k \cdot e_k$, we therefore deduce that these candidates are valid if and only if they satisfy

$$\delta = b - \sum_{k=1}^{K} \theta_k \cdot e_k \in \mathbb{Z}. \tag{6.61}$$

However, this relation can be used in the algorithm only if b is not computed modulo $\mathbb{Z}$. In this case the verification of ratios of Pochhammer parts of hypergeometric term solutions for the value of b should consist in checking if the difference $b - \sum_{k=1}^{K} \theta_k \cdot e_k$ is an integer.

Another approach is to use again asymptotic expansion. Since now we have the ratio terms with the value c of the local type of a hypergeometric term solution, according to (6.28) we can write

$$r(n) = \frac{R(n+1)}{R(n)} \cdot c \cdot R_{\text{Pochhammer}}(n), \tag{6.62}$$

where $R_{\text{Pochhammer}}(n)$ denotes one of the remaining ratio candidates of hypergeometric term solutions. Since

$$\frac{R(n+1)}{R(n)} = 1 + \frac{\delta}{n} + O\left(\frac{1}{n^2}\right), \tag{6.63}$$

where δ is the difference between the degree of the numerator and the denominator of R, the leading term in the asymptotic expansion of

$$\sum_{i=0}^{d} P_i \prod_{j=0}^{i-1} r(n+i), \tag{6.64}$$

where $P_i, i = 0, \ldots, d$ are the polynomial coefficients of the given holonomic RE, must have a polynomial coefficient in the variable δ. Therefore the values of δ are integer roots (if there are some) of that polynomial. If they do not exist, then the rational function $c \cdot R_{\text{Pochhammer}}(n)$ is not involved in the ratio of a hypergeometric term solution of (6.18).

This latter approach is the one that we adopt as it allows to join the value c of the local type to the ratios for the Pochhammer parts of hypergeometric term solutions.

Most often after this step the number of ratio candidates of Pochhammer parts of hypergeometric term solution of (6.18) is considerably reduced or equal to the exact number of hypergeometric term solutions.

It only remains to find the rational function R in (6.27) whose holonomic recurrence equation can be easily computed. Let $c \cdot R_{\text{Pochhammer}}(n)$ be one of the remaining ratios times its corresponding c for the local type. Then the recurrence equation

$$\sum_{i=0}^{d} P_i \cdot R(n+i) \cdot c^{n+i} \cdot R_{\text{Pochhammer}}(n+i) = 0, \tag{6.65}$$

is an equation for the unknown rational function $R(n)$ that we can modify to a holonomic RE after multiplication by the least common multiple of the denominators of the corresponding rational coefficients. Assume that we obtain the holonomic recurrence equation

$$\sum_{i=0}^{d} A_i(n) \cdot R(n+i) = 0, \tag{6.66}$$

with $A_0(n), \ldots, A_d(n) \in \mathbb{K}[n]$, $A_0(n) \cdot A_d(n) \neq 0$.

Observe, however, that there is no need to use a complete algorithm for computing rational solutions of holonomic recurrence equations. Indeed, since we already have the possible difference between the degrees of the numerators and the denominators of rational solutions of (6.66), it is enough to use an algorithm that computes a universal denominator[1] $U(n)$ of all rational solutions of (6.66) and use the maximum value $\delta_{\max}$ of the difference between the degrees of the numerators and the denominators to compute a degree bound $\delta_{\max} + \deg(U(n))$ for the degrees of the corresponding numerators. Substituting $\dfrac{N(n)}{U(n)}$ in (6.66) where $N(n)$ is an arbitrary polynomial of degree $\delta_{\max} + \deg(U(n))$ and $U(n)$ the computed universal denominator results in a linear system in the coefficients of the arbitrary polynomial $N(n)$. Finally solving that system gives a basis of all the rational functions $R(n) = \dfrac{N(n)}{U(n)}$ searched for.

Since the computations of numerators of rational solutions $R(n)$ of (6.66) is straightforward after the computation of a universal denominator of rational solutions of (6.66), the only remaining important step to be described in order to finish the explanations of our variant of van Hoeij's algorithm is the one for the computation of a universal denominator of a given holonomic recurrence equation. Below we give the original Abramov algorithm for computing the universal

[1] A universal denominator of rational solutions of a holonomic RE is a polynomial that is divisible by all the denominators of rational solutions of that holonomic RE [Abramov, 1999].

denominator of a given holonomic RE (see ([Abramov and Barkatou, 1998], [Abramov, 1999], [Abramov et al., 2011])).

Algorithm 9 Computing the universal denominator of rational solutions of a holonomic RE

Input: The leading and the trailing polynomial coefficients $A_0(n)$ and $A_d(n)$ ($A_0(n) \cdot A_d(n) \neq 0$)

Output: A universal denominator $U(n)$ of all rational solutions of a holonomic recurrence equation having $A_0(n)$ and $A_d(n)$ as trailing and leading coefficients, respectively.

1. Set $U(n) = 1$, $V(n) := A_d(n-d)$, $W(n) := A_0(n)$.
2. Compute the dispersion set D
$$D := ds(V(n), W(n)) := \{h \in \mathbb{N} \ : \ \deg\left(\gcd(V(n), W(n+h))\right) > 0\} \tag{6.67}$$
3. If $D = \emptyset$, then stop and return $U(n)$.
4. Change D as a list and sort it in decreasing order, say $D = [h_1, h_2, \ldots, h_m]$ such that $h_1 > h_2 > \ldots > h_m$.
5. For $j = 1, \ldots, m$, do
   - $P(n) = \gcd\left(V(n), W(n+h_j)\right)$;
   - $V(n) = V(n)/P(n)$;
   - $W(n) = W(n)/P(n-h_j)$;
   - $U(n) = U(n)\prod_{k=0}^{h_j} P(n-k)$.
6. Return $U(n)$.

The set $ds(V(n), W(n))$ can be computed as the set of all non-negative integer roots of the resultant polynomial of $V(n)$ and $W(n+h)$ in the variable h. Note, however, that this way of calculating the dispersion set is not efficient, particularly because the variable h that is used makes the problem a two-variable problem even though the dispersion set only contains information about polynomials of one variable. And we recall that the degree of the resultant of two polynomials is the product of their degrees (see [Koepf, 2014, Chapter 5]).

However the dispersion set can also be obtained from the full factorization of $V(n)$ and $W(n)$. Indeed, for given irreducible factors $w(n), v(n)$ of $W(n)$ and $V(n)$, respectively, such that $\deg(w(n)) = \deg(v(n)) = m$, one can easily recognize whether or not there exists $h \in \mathbb{Z}$ verifying $w(n+h) = v(n)$; precisely, if

$$w(n) = n^m + w_{m-1}n^{m-1} + \cdots ,$$

$$v(n) = n^m + v_{m-1}n^{m-1} + \cdots ,$$

then $w(n+h) = n^m + (w_{m-1} + nh)n^{m-1} + \cdots$ and the only candidate for h is

$$\frac{v_{m-1} - w_{m-1}}{m}, \tag{6.68}$$

if this value is an integer. Using this approach makes the computation faster and this is the one we use. For more details about the computation of the dispersion set see ([Koepf, 2014, Algorithm 5.2], [Man and Wright, 1994]).

On the other hand, this step of computing the universal denominator can still be improved for sake of efficiency. Indeed, as explained in [Abramov et al., 2011], Algorithm 9 is not the best existing version of Abramov's algorithm for computing a universal denominator of linear recurrences. The reason is that the $\gcd$ computations made in the algorithm do not use previous informations on the factorization of $V(n)$ and $W(n)$.

Note, however, that this is a common situation in Computer Algebra: "...Several algorithms in symbolic computation depend on a subroutine for finding the rational solutions of ordinary linear difference equations and several algorithms are known for implementing such subroutines..." ([Gheffar and Abramov, 2011], [Kauers and Schneider, 2010]).

Our package contains the function *Udenom(*A_0*,*A_d*,d,n)* which computes a universal denominator of rational solutions of holonomic recurrence equations having A_0 as the polynomial coefficient of its trailing term and A_d as the one of its leading term in the variable *n*. Let us give an example.

Using *sumhyperRE*, we generate a holonomic RE for linear combinations of the rational functions $\dfrac{n+1}{(n+2)(n+3)}$ and $\dfrac{1}{(n+4)(n+5)}$.

```
(%i16) RE:sumhyperRE([(n+1)/(n+2)/(n+3),1/(n+4)/(n+5)],a[n]);
```

$$(\%o16)\quad (5+n)\cdot(7+n)\cdot\left(8+4\cdot n+n^2\right)\cdot a_{n+2}-2\cdot\left(178+178\cdot n+73\cdot n^2+14\cdot n^3+n^4\right)\cdot a_{n+1}+(2+n)\cdot(4+n)\cdot\left(13+6\cdot n+n^2\right)\cdot a_n=0$$

```
(%i17) Coeffs:REcoeff(RE,a[n])$
(%i18) A0:first(Coeffs);
```

$$(\%o18)\quad (n+2)\cdot(n+4)\cdot\left(n^2+6\cdot n+13\right)$$

```
(%i18)  A2:last(Coeffs);
```

$$(\%o19)\quad (n+5)\cdot(n+7)\cdot\left(n^2+4\cdot n+8\right)$$

```
(%i19) Udenom(A0,A2,2,n);
```

$$(\%o19)\quad (n+2)\cdot(n+3)\cdot(n+4)\cdot(n+5)$$

We think that this step of computing the rational function $R(n)$ of the representation (6.26) of hypergeometric term solutions of holonomic REs, is the only one that could make a difference of efficiency between our variant of van Hoeij's algorithm and its original version. In his approach, van Hoeij uses a special algorithm from his idea of finite singularities for computing rational solutions of holonomic RE (see [Van Hoeij, 1998]) to find R. Though we have mentioned that a complete algorithm to compute rational solutions of holonomic REs is not necessary, the algorithm described in [Van Hoeij, 1998] can well be adapted with computations done in [Van Hoeij, 1999]. However, it has been proved in [Abramov et al., 2011] that the improved version of Abramov's algorithm, named A_U', for computing rational solutions of a given linear recurrence generally gives the best efficiency. This algorithm computes a universal denominator as in Algorithm 9 by avoiding unnecessary computations particularly involved in the $\gcd$ computations.

Nevertheless most often in our direction of computing hypergeometric type series, for the kind of recurrence equations that we are dealing with, both algorithms perform the same time complexity for the computations of corresponding hypergeometric term solutions.

One last thing to be mention is the avoidance of splitting fields. In our case this does not constitute a real issue since we are more interested in the field of rationals and some of its extensions. These computable ones do not generally slow down the algorithm.

We can now present the complete algorithm of this chapter.

Algorithm 10 A variant of van Hoeij's algorithm

Input: Polynomials

$$P_i(n) \in \mathbb{K}(n), i = 0, \ldots, d, \mid P_d(n) \cdot P_0(n) \neq 0.$$

Output: A basis for all hypergeometric term solutions of the holonomic recurrence equation

$$\sum_{i=0}^{d} P_i(n) a_{n+i} = 0 \tag{6.69}$$

over $\mathbb{K}$.

1. Set $H = \{\}$.
2. Use Algorithm 8 to compute the set $\mathcal{L}$ of all local types at infinity of hypergeometric term solutions of (6.69).
3. If $\mathcal{L} = \emptyset$, then stop and return H.
4. Construct the set of couple numerator-denominator for ratio candidates of the Pochhammer parts of hypergeometric term solutions

$$\mathcal{P} := \Big\{ (p(n), q(n)) \in \mathbb{K}[n]^2 \;:\; p(n) \text{ and } q(n) \text{ are monic factors modulo } \mathbb{Z} \text{ of } P_0(n-1) \text{ and } P_d(n-d) \text{ respectively} \Big\}. \tag{6.70}$$

5. Remove from $\mathcal{P}$ all couple whose $p(n)$ exponents are less than the minimum multiplicity of the corresponding root modulo $\mathbb{Z}$ in the trailing polynomial coefficient $P_0(n)$. Similarly, clear $\mathcal{P}$ by the same consideration for $q(n)$ exponents and the leading polynomial coefficient $P_d(n)$. Finally substitute each remaining couple $(p(n), q(n))$ in $\mathcal{P}$ by $\frac{p(n)}{q(n)}$.

Algorithm 10 A variant of van Hoeij's algorithm

6. Construct the set F_1 of $c \cdot r$, $r \in \mathcal{P}$ such that $c \cdot r$ has its local type at infinity as an element of $\mathcal{L}$.

$$F_1 := \left\{ c \cdot r \ : \ r = n^{\nu_r} \left(1 + \frac{b_r}{n} + O\left(\frac{1}{n^2}\right)\right) \in \mathcal{P} \text{ and } (\nu_r, c, b_n) \in \mathcal{L} \right\}. \tag{6.71}$$

7. Set $F_2 := \{\}$. For each element $f(n)$ of F_1
   (a) Compute a recurrence equation, say E_f with the coefficients

   $$P_i \cdot \prod_{j=0}^{i} f(n+i),\ i = 0, \ldots, d, \tag{6.72}$$

   for the rational function $R(n)$ in (6.28) of the possible hypergeometric terms solutions.
   (b) Substitute the terms $R(n+i)$ by $(1 + \frac{\delta}{n+i})$, $i = 0, \ldots, d$, in E_f and compute the coefficient of the leading term of the asymptotic expansion of the left hand side of E_f, say $Q_f(\delta)$.
   (c) Compute the set S_{δ_f} of integer roots of $Q_f(\delta)$.
   (d) If $S_{\delta_f} = \emptyset$ then $f(n)$ is discarded.
   (e) Else set $\delta_f := \max(S_f)$, modify E_f to holonomic form and add $(f(n), \delta_f, E_f)$ in F_2.
8. If $F_2 = \emptyset$ then stop and return H.
9. For each $(f(n), \delta_f, E_f) \in F_2$
   (a) Use Algorithm 9 to compute the universal denominator $U_f(n)$ of rational solutions of E_f.
   (b) Update E_f as E_f' with $U_f(n)$ to get a holonomic RE for the numerators of rational solutions of E_f.
   (c) Set $d_{N_f} := \deg(U_f(n)) + \delta_f$, and find a basis of all polynomial solutions of degree at most d_{N_f} of E_f'.
   (d) Use Algorithm 7 to compute $h_f(n) = \prod_{k=0}^{n-1} f(k)$.
   (e) For each $N_f(n) \in S_{N_f}$ add $\frac{N_f(n)}{U_f(n)} \cdot h_f(n)$ to H.
10. Return H

Remark

- The steps 4 and 5 of Algorithm 10 can easily be combined in an algorithm which builds ratios of possible Pochhammer parts of hypergeometric term solutions of a given holonomic RE from its leading and trailing polynomial coefficients.

- Algebraic extension fields are considered in the same way as with our description of Petkovšek's algorithm. According to the users and the computer algebra systems used for implementation, finding the zeros for the values of c in the local type algorithm

(Algorithm 8) or factorizing the leading and the trailing polynomial coefficients may need the specification of the field where the computations of certain hypergeometric terms can be done. By default, we consider the field of rationals, its computable extensions are just taken as the collection of the common operations, the field of rationals, and the needed irrational numbers computed in the field of complex numbers.

Our package contains the function *HypervanHoeij(RE,a[n],[K])* which implements Algorithm 10, with the default value Q for K representing the field of computations. As for our implementation of Petkovšek's algorithm, one uses C to allow computations in extension fields of $\mathbb{Q}$.

Note that *HypervanHoeij* may need extension fields when *HyperPetkov* does not. Indeed, for a given holonomic RE, the monic factors computed in van Hoeij's algorithm are used to determine the possible ratios of Pochhammer parts of hypergeometric term solutions of that RE. However, it may happen that a "simple" formula of a hypergeometric term needs an extension field of $\mathbb{Q}$ whereas its ratio is a rational function over $\mathbb{Q}$. Let us present an example.

The differential equation

```
(%i20) DE:z*F(z)+z^2*F(z)+z^3*diff(F(z),z)+z^4*diff(F(z),z,2)=0;
```

$$(\%o20) \quad z^4 \cdot \left(\frac{d^2}{d\,z^2} \cdot \mathrm{F}(z)\right) + z^3 \cdot \left(\frac{d}{d\,z} \cdot \mathrm{F}(z)\right) + z^2 \cdot \mathrm{F}(z) + z \cdot \mathrm{F}(z) = 0 \tag{6.73}$$

leads to the holonomic recurrence equation

```
(%i21) RE:DEtoRE(DE,F(z),a[n]);
```

$$(\%o21) \quad a_{n-1} + \left(5 - 4 \cdot n + n^2\right) \cdot a_{n-2} = 0. \tag{6.74}$$

Using our implementations of Petkovšek's and van Hoeij's algorithms yields

```
(%i22) HypervanHoeij(RE,a[n]);
```

$$(\%o22) \quad \{\}$$

```
(%i23) HyperPetkov(RE,a[n]);
```

$$(\%o23) \quad \left\{-\left(n^2+1\right)\right\}$$

However, allowing computations in extension fields for *HypervanHoeij*, we get

```
(%i24) Hyp:HypervanHoeij(RE,a[n],C);
```

$$(\%o24) \quad \left\{\frac{(1-i)_n \cdot (i+1)_n \cdot (-1)^n}{n^2+1}\right\}$$

Of course after computing the ratio of this hypergeometric term we get the same result as *HyperPetkov*.

```
(%i25) map(lambda([r], ratio(r,n)),Hyp);
```

$$(\%o25) \qquad -\left(1+n^2\right)$$

Note that this simplification holds because *ratio* uses the Maxima command *ratsimp* which replaces i^2 by -1 before the factorization over the field of rational numbers. As we have mentioned earlier, factorization in linear factors is not always feasible in Computer Algebra. Therefore this lack of algorithms for complete factorization over algebraically closed fields may sometimes hide hypergeometric term solutions of holonomic REs if they are searched in a "simple" formula form. This is one of the reasons why we prefer to use our implementation of Petkovšek's algorithm for the computation of ratios of hypergeometric term solutions of holonomic REs.

However, for the computation of "simple" formulas of hypergeometric terms which is our need for power series computations, sometimes integer shifts have to be considered in order to apply Algorithm 7. This situation is clearly avoided with Algorithm 10 which only calls Algorithm 7 for rational functions having poles and zeros with negative real parts. A simple example is the recurrence equation of $\log(1+z)$ below.

```
(%i26) RE:FindRE(log(1+z),z,a[n]);
```

$$(\%o26) \qquad (1+n)\cdot(2+n)\cdot a_{n+2}+(1+n)^2\cdot a_{n+1}=0$$

```
(%i27) HyperPetkov(RE,a[n]);
```

$$(\%o27) \qquad \left\{-\frac{n}{n+1}\right\}$$

Since 0 is a root of the obtained ratio's numerator, application of Algorithm 7 after shifting the indeterminate by 1 leads to the formula $\frac{(-1)^{n+1}}{n+1}$ for computations starting from 0. Using *HypervanHoeij* we obtain

```
(%i28) HypervanHoeij(RE,a[n]);
```

$$(\%o28) \qquad \left\{\frac{(-1)^n}{n}\right\}$$

for computations starting from 1. The starting value in the second case can be directly deduced from the recurrence equation. We give more details about such computations in Chapter 7. Note that the situation might become more complicated to deal with when the output of *HyperPetkov* contains linearly dependent hypergeometric term ratios.

Let us now compute more examples and evaluate the improvement of the time complexity compared to Petkovšek's algorithm.

We start with (6.32) and see which basis *HypervanHoeij* finds for its hypergeometric term solutions.

```
(%i29) term1:(1+n)/n!$
```

```
(%i30) term2:(1+n+n^2)/n!$
(%i31) term3:(1+n+n^2+n^3)/n!$
(%i32) RE:sumhyperRE([term1,term2,term3],a[n])$
(%i33) HypervanHoeij(RE,a[n]);
```

Evaluation took 0.0200 seconds (0.0300 elapsed)

$$(\%o33) \qquad \left\{\frac{1+n}{n!}, \frac{n^2}{n!}, \frac{n^3}{n!}\right\}$$

We obtained exactly the expected basis of hypergeometric term solutions (6.34). For this example, our implementation of Petkovšek's algorithm get

```
(%i34) HyperPetkov(RE,a[n]);
```

Evaluation took 0.0400 seconds (0.0500 elapsed)

$$(\%o34) \qquad \left\{\frac{1+n}{n^2}, \frac{1+n}{(n-1)\cdot n}, \frac{2+n}{(1+n)^2}, \frac{(1+n)^2}{n^3}\right\}$$

whose some ratios correspond to linearly dependent hypergeometric terms. One sees that the computation of basis of hypergeometric term solutions using Petkovšek's algorithm needs some other considerations. The Maxima package *solve_rec* gives

```
(%i35) solve_rec(RE,a[n]);
```

Evaluation took 0.3800 seconds (0.5500 elapsed)

$$(\%o35) \qquad a_n = \frac{\%k_1 \cdot n^2}{(n-1)!} + \frac{\%k_2 \cdot (n-1) \cdot n}{(n-1)!} + \frac{\%k_3 \cdot n}{(n-1)!}$$

which is an incorrect result since the given linearly independent hypergeometric terms cannot be used to compute $\frac{1+n}{n!}$, $\frac{n^2}{n!}$ and $\frac{n^3}{n!}$ at the same time.

```
(%i36) RE:FindRE(sqrt(1+z)+1/sqrt(1+z),z,a[n])$
(%i37) HypervanHoeij(RE,a[n]);
```

Evaluation took 0.0200 seconds (0.0200 elapsed)

$$(\%o37) \qquad \left\{\frac{(n-1)\cdot(-1)^n\cdot(2\cdot n)!}{(2\cdot n-1)\cdot 4^n\cdot n!^2}\right\}$$

This gives exactly the general coefficient as in (5.1) page 65.

```
(%i38) RE:FindRE(exp(z)+log(1+z^2)+1/(1-z)^20,z,a[n])$
(%i39) HypervanHoeij(RE,a[n],C);
```

Evaluation took 6.5500 seconds (6.6800 elapsed)

$$(\%o39) \quad \left\{\frac{(-i)^n}{n},\ (n+1)\cdot(n+2)\cdot(n+3)\cdot(n+4)\cdot(n+5)\cdot(n+6)\cdot(n+7)\cdot(n+8)\cdot(n+9)\cdot(n+10)\cdot(n+11)\cdot(n+12)\cdot(n+13)\cdot(n+14)\cdot(n+15)\cdot(n+16)\cdot(n+17)\cdot(n+18)\cdot(n+19),\ \frac{(-1)^{\frac{n}{2}}}{n},\ \frac{1}{n!}\right\}$$

We have seen on page 80 that for this example *HyperPetkov* takes 339.06 seconds to compute all the hypergeometric term solutions. Therefore the result above shows a large improvement of the time complexity with *HypervanHoeij* which only takes about 4 seconds to compute the same hypergeometric term solutions.

```
(%i40) RE:FindRE(z/(1-z)^4+5*z^3*exp(z)
       +(1+7*z^2)*log(1+z)+(2+z)/sqrt(1+z),z,a[n])$
```

```
(%i41) HypervanHoeij(RE,a[n]);
```

Evaluation took 5.5200 seconds (5.6800 elapsed)

$$(\%o41) \quad \left\{\frac{n\cdot(1+n)\cdot(2+n)}{216},\ \frac{(4\cdot n-1)\cdot(-1)^n}{864\cdot(n-2)\cdot n},\ \frac{(n-2)\cdot(n-1)\cdot n}{n!},\ \frac{(n-1)\cdot(-1)^n\cdot(2\cdot n)!}{6\cdot(2\cdot n-1)\cdot 4^n\cdot n!^2}\right\}$$

For this example we have seen on page 83 that *HyperPetkov* takes about 54 seconds.

Let use *sumhyperRE* to generate the complicated example used at the end of the previous chapter.

```
(%i42) term1:(pochhammer(1/2,n)^5*pochhammer(1,n))
       /(pochhammer(3/4,n)^3*pochhammer(1/3,n));
```

$$(\%o42) \quad \frac{\left(\frac{1}{2}\right)_n^{\ 5}\cdot n!}{\left(\frac{1}{3}\right)_n\cdot\left(\frac{3}{4}\right)_n^{\ 3}}$$

```
(%i43) term2:pochhammer(1/4,n)/
       (pochhammer(1,n)^3*pochhammer(1/3,n)^4);
```

$$(\%o43) \quad \frac{\left(\frac{1}{4}\right)_n}{\left(\frac{1}{3}\right)_n^{\ 4}\cdot n!^3}$$

```
(%i44) term3:pochhammer(1/5,n)^2/
       (pochhammer(1/7,n)*pochhammer(2,n));
```

$$(\%o44) \quad \frac{\left(\frac{1}{5}\right)_n^{\ 2}}{\left(\frac{1}{7}\right)_n\cdot(2)_n}$$

```
(%i45) RE:sumhyperRE([term1,term2],a[n])$
(%i46) HypervanHoeij(RE,a[n]);
```

Evaluation took 0.2600 seconds (0.2800 elapsed)

$$(\%o46)\quad \left\{\frac{\left(\frac{1}{4}\right)_n}{\left(\frac{1}{3}\right)_n^{\,4}\cdot n!^3},\frac{(2\cdot n)!^5}{\left(\frac{1}{3}\right)_n\cdot\left(\frac{3}{4}\right)_n^{\,3}\cdot 4^{5\cdot n}\cdot n!^4}\right\}$$

Here we get a fraction of a second compare to the 17.74 obtained in page 88.

```
(%i47) RE:sumhyperRE([term1,term2,term3],a[n])$
(%i48) HypervanHoeij(RE,a[n]);
```

Evaluation took 4.1700 seconds (4.4700 elapsed)

$$(\%o48)\quad \left\{\frac{\left(\frac{1}{4}\right)_n}{\left(\frac{1}{3}\right)_n^{\,4}\cdot n!^3},\frac{\left(\frac{1}{5}\right)_n^{\,2}}{2\cdot\left(\frac{1}{7}\right)_n\cdot(n+1)\cdot n!},\frac{(2\cdot n)!^5}{\left(\frac{1}{3}\right)_n\cdot\left(\frac{3}{4}\right)_n^{\,3}\cdot 4^{5\cdot n}\cdot n!^4}\right\}$$

And again, we obtain a much better timing compared to the 20 minutes obtained using *Hyper-Petkov*.

Mark van Hoeij's algorithm constitutes the current state-of the-art algorithm for finding hypergeometric term solutions of holonomic recurrence equations. More details about this algorithm can be found in [Cluzeau and van Hoeij, 2006]. With our implementation, Maxima reaches Maple's level in computing hypergeometric term solutions of holonomic RE. Van Hoeij implemented his algorithm in Maple as *LREtools[hypergeomsols]*. This Maple command seems to be slightly more efficient than our *HypervanHoeij* since it uses more "tricks" to detect the possible Pochhammer parts of hypergeometric term solutions of holonomic REs (see [Cluzeau and van Hoeij, 2006, Remark 3, Page 95]). Let us apply van Hoeij's implementation on our previous example. We use Maple's *gfun* and Koepf's *hsum17* packages to compute the same holonomic recurrence equation and we apply *LREtools[hypergeomsols]* to find a basis of the corresponding hypergeometric term solutions (see [Koepf, 2014, Section 9.14]).

```
> term1:=(pochhammer(1/2,n)^5*pochhammer(1,n))
> /(pochhammer(3/4,n)^3*pochhammer(1/3,n)):
> term2:= pochhammer(1/4,n)/(pochhammer(1,n)^3
> *pochhammer(1/3,n)^4):
> term3:=pochhammer(1/5,n)^2
> /(pochhammer(1/7+n)*pochhammer(2+n)):
> HolonomicRE:=proc(term,sk)
> local s,k,r;
> s:=op(0,sk) : k:=op(1,sk):
> r:=ratio(term,k);
> denom(r)*s(k+1)-numer(r)*s(k)=0;
> end proc:

> read "hsum17.mpl":

> with(gfun):

> RE1:=HolonomicRE(term1,s(n)):

> RE2:=HolonomicRE(term2,s(n)):
```

```
> tmp:=gfun["rec+rec"](RE1,RE2,s(n)):
> if type(tmp,set) then tmp:=select(has,tmp,n)[1] end if:
> RE:=map(factor,tmp):
> RE3:=HolonomicRE(term3,s(n)):
> RE:=map(factor,gfun["rec+rec"](RE,RE3,s(n))):
> TIME:=time():LREtools[hypergeomsols](RE,s(n)
> ,{},output=basis);time()-TIME;
```

$$\left[\frac{\Gamma\left(n+\frac{1}{5}\right)^2}{\Gamma\left(n+\frac{1}{7}\right)\cdot\Gamma\left(n+2\right)}, \frac{\Gamma\left(n+\frac{1}{2}\right)^5\cdot\Gamma\left(n+1\right)}{\Gamma\left(n+\frac{1}{3}\right)\cdot\Gamma\left(n+\frac{3}{4}\right)^3}, \frac{(1+n)\cdot\Gamma\left(n+\frac{1}{4}\right)}{\Gamma\left(n+\frac{1}{3}\right)^4\cdot\Gamma\left(n+1\right)^2\cdot\Gamma\left(n+2\right)}\right]$$

2.381

As one can see, the timing is close to the one we obtained with our implementation for this example. Of course we should not forget to mention that in terms of timing, Maple is more efficient than Maxima essentially because it is a more optimized CAS.

Note that given a field $\mathbb{K}$, if Petkovšek's or van Hoeij's algorithm does not find hypergeometric term solutions of a given holonomic recurrence equation over $\mathbb{K}$, then we can say that the given holonomic RE has no hypergeometric term solutions over $\mathbb{K}$. For example, this is the case with the holonomic RE

```
(%i49) RE:FindRE(sin(z^3)^3,z,a[n]);
```

$$(\%o49) \quad (n-8)\cdot(n-5)\cdot(n-2)\cdot(1+n)\cdot a_{n+1}+90\cdot(n-8)\cdot(n-5)\cdot a_{n-5}+729\cdot a_{n-11}=0 \tag{6.75}$$

for which using *HyperPetkov* and *HypervanHoeij* with allowance of computations in extension fields of $\mathbb{Q}$ yields

```
(%i50) HyperPetkov(RE,a[n],C);
```

Evaluation took 0.0500 seconds (0.0600 elapsed)

$$(\%o50) \quad \{\,\}$$

```
(%i51) HypervanHoeij(RE,a[n],C);
```

Evaluation took 0.0300 seconds (0.0300 elapsed)

$$(\%o51) \quad \{\,\}$$

However, remember, that we introduced a more general concept about hypergeometric terms in Definition 2.14, page 28. The holonomic RE (6.75) has m-fold hypergeometric term solutions that could not be detected by neither Petkovšek's nor van Hoeij's algorithm over $\mathbb{C}$. Next, we move to this general situation about solutions of holonomic recurrence equations for which hypergeometric terms are just a particular case.

Chapter 7

$m-$fold Hypergeometric Term Solutions of Holonomic Recurrence Equations

"I was always an optimizer, I wanted the best for the least effort."
Alessio Figalli
"To invent is to think on the side."
Albert Einstein

In this chapter, we present the most important result of this thesis. That is a new algorithm for the computation of m-fold hypergeometric term solutions of holonomic recurrence equations. This subject appears as the essential element for the completeness of power series generation as it allows to compute all the needed m-fold hypergeometric term solutions of holonomic REs satisfied by the Taylor coefficients of linear combinations of hypergeometric type functions. On the other hand, this subject has an impact in the theory of hypergeometric summation. Indeed, it is important to notice the scope of our algorithm in this research area because some concepts used for power series representation come from there. To introduce m-fold hypergeometric terms, we would like to recall some important results of the hypergeometric summation theory as presented in the book [Koepf, 2014, Chapter 4 to Chapter 9] before continuing with power series generation.

7.1 m-fold Hypergeometric Terms in Hypergeometric Summation

The computation of infinite series was connected to holonomic REs by Celine Fasenmyer. To find hypergeometric term representations of hypergeometric series, she proposed the following approach:

Given a sum $\sum_{k=0}^{n} F(n,k)$, write[1]

$$s_n = \sum_{k=-\infty}^{\infty} F(n,k), \tag{7.1}$$

and search for polynomials $p_{i,j} = p_{i,j}(n), i = 0, \ldots, b,\ j = 0, \ldots, d$ with respect to n and do not depend on k, such that

$$\sum_{i=0}^{b} \sum_{j=0}^{d} p_{i,j} F(n+j, k+i) = 0. \tag{7.2}$$

If such polynomials are found, then one deduces a holonomic recurrence equation of order at most d for s_n as follows

$$\begin{aligned} 0 &= \sum_{k=-\infty}^{\infty} \sum_{i=0}^{b} \sum_{j=0}^{d} p_{i,j} F(n+j, k+i) \\ &= \sum_{i=0}^{b} \sum_{j=0}^{d} p_{i,j} \cdot \left(\sum_{k=-\infty}^{\infty} F(n+j, k+i) \right) \\ &= \sum_{i=0}^{b} \sum_{j=0}^{d} p_{i,j} s_{n+j} \\ &= \sum_{j=0}^{d} \left(\sum_{i=0}^{b} p_{i,j} \right) s_{n+j}, \end{aligned}$$

where of course we use the fact that $p_{i,j}$ does not depend on k and the advantage of working with bilateral sums: their value is invariant with respect to shifts of the summation variable. In the 1940s, Fasenmyer's method could be used to compute explicit formulas of hypergeometric series only when the obtained holonomic RE was of first order or a two-term recurrence relation.

Example 7.1. *Applied to $s_n = \sum_{k=0}^{n} k\binom{n}{k}$, Fasenmyer's method leads to the recurrence equation*

$$n s_{n+1} - 2(n+1) s_n = 0,$$

which after use of the initial value $s_1 = \sum_{k=0}^{1} k\binom{1}{k} = 1$, yields $s_n = n2^{n-1}$.

For the definite summation case, Gosper proposed an algorithm which deals with the question of how to find a (forward) anti-difference s_k for a given a_k, that is a sequence s_k such that

$$a_k = \Delta s_k = s_{k+1} - s_k, \tag{7.3}$$

in the particular case that s_k is a hypergeometric term. Thus, once a hypergeometric anti-difference s_k of a_k is computed, by telescoping definite summation yields

$$\sum_{k=n_0}^{n} a_k = (s_{n+1} - s_n) + (s_n - s_{n-1}) + \cdots + (s_{n_0+1} - s_{n_0}) = s_{n+1} - s_{n_0},$$

[1]The sum is taken over $\mathbb{Z}$ because the summation term vanishes outside a finite set, we say that it has a finite support.

by an evaluation at the limits of summation. Gosper's idea is based on the representation

$$\frac{a_{k+1}}{a_k} = \frac{p_{k+1}}{p_k} \cdot \frac{q_{k+1}}{r_{k+1}}, \tag{7.4}$$

with the property

$$\gcd(q_k, r_{k+j}) = 1, \ \forall j \in \mathbb{N}_{\geqslant 0} \tag{7.5}$$

that can be algorithmically generated (see [Koepf, 2014, Lemma 5.1 and Algorithm]). Using (7.4) and (7.5), one proves that the function

$$f_k := \frac{s_{k+1}}{a_{k+1}} \cdot \frac{p_{k+1}}{r_{k+1}} \tag{7.6}$$

must be a polynomial for a hypergeometric term anti-difference a_k to exist. Thus using (7.3), (7.6) and (7.4) it follows that f_k satisfies the inhomogeneous recurrence equation

$$p_k = q_{k+1} f_{k+1} - r_k f_{k-1}. \tag{7.7}$$

Gosper gives an upper bound for the degree of f_k in terms of p_k, q_k, and r_k which yields a method for calculating f_k by introducing the appropriate arbitrary polynomial, equating coefficients, and solving the corresponding linear system so that we finally find

$$s_k = \frac{r_k}{p_k} f_{k-1} a_k. \tag{7.8}$$

Gosper implemented his algorithm in Maxima as *nusum* with the same syntax as the Maxima *sum* command.

Example 7.2.

```
(%i1) nusum(k*k!,k,0,n);
```

solve: dependent equations eliminated: (1)

$$(\%o1) \qquad (1+n)! - 1$$

```
(%i2) nusum(k^3,k,0,n);
```

$$(\%o2) \qquad \frac{n^2 \cdot (1+n)^2}{4}$$

Gosper's algorithm is the essential tool of the so called Wilf-Zeilberger (often named WZ) method. That is a clever application of Gosper's algorithm to prove identities of the form

$$s_n := \sum_{k=-\infty}^{\infty} F(n,k) = 1 \tag{7.9}$$

where $F(n,k)$ is a hypergeometric term with respect to both n and k with finite support. For this purpose, one applies Gosper's algorithm to the expression

$$a_k := F(n+1,k) - F(n,k) \tag{7.10}$$

with respect to the variable k. If successful, this generates $G(n,k)$ with

$$a_k = F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k), \tag{7.11}$$

and summing over $\mathbb{Z}$ yields

$$s_{n+1} - s_n = \sum_{k=-\infty}^{\infty} F(n+1,k) - F(n,k) = \sum_{k=-\infty}^{\infty} G(n,k+1) - G(n,k) = 0 \tag{7.12}$$

since the right-hand side is telescoping. Therefore, s_n is constant, $s_n = s_0$, and it only remains to prove that $s_0 = 1$. In practice, once the function $G(n,k)$ is computed one uses the rational function

$$R(n,k) := \frac{G(n,k)}{F(n,k)} \tag{7.13}$$

called the WZ certificate of $F(n,k)$, to establish (7.9) by proving the rational identity

$$\frac{F(n+1,k)}{F(n,k)} - 1 + R(n,k) - R(n,k+1)\frac{F(n,k+1)}{F(n,k)} = 0 \tag{7.14}$$

which is deduced from (7.12) after division by $F(n,k)$.

Example 7.3. *For $s_n := \sum_{k=0}^{n} F(n,k) = \sum_{k=0}^{n} \frac{1}{2^n}\binom{n}{k} = 1$, the WZ certificate is*

$$R(n,k) = -\frac{k}{2(n+1-k)}.$$

Therefore the corresponding left hand side of identity (7.14) is

$$\frac{n+1}{2(n+1-k)} - 1 - \frac{k}{2(n+1-k)} + \frac{k+1}{(2(n-k))} \cdot \frac{n-k}{k+1}$$

which trivially yields zero.

Although Gosper's algorithm applies to finite summation, it constitutes a useful tool in discovering a method for infinite sums. This is observable in Zeilberger's algorithm. Zeilberger brings back the computation of a holonomic recurrence equation for $s_n := \sum_{k=-\infty}^{\infty} F(n,k)$. The idea is to apply Gosper's algorithm in the following way: For suitable $d = 1, 2, \ldots$ set

$$a_k := F(n,k) + \sum_{j=1}^{d} \sigma_j(n) F(n+j,k) \tag{7.15}$$

where σ_j is supposed to be a rational function depending on n and not on k. Zeilberger's main observation is that the computation of the polynomial f_k defined in (7.6) yields a linear system not only for the unknown coefficients of f_k, but also for the rational functions $\sigma_j, j = 1, \ldots, d$.

Thus in a successful case, one obtains an anti-difference $G(n,k)$ of a_k and rational functions $\sigma_j(n), j = 1, \ldots, d$ such that

$$a_k = G(n,k+1) - G(n,k) = F(n,k) + \sum_{j=1}^{d} \sigma_j(n) F(n+j,k). \tag{7.16}$$

Hence, by summation

$$\begin{aligned} 0=\sum_{k=-\infty}^{\infty} G(n,k+1)-G(n,k) &= \sum_{k=-\infty}^{\infty}\left(F(n,k)+\sum_{j=1}^{d}\sigma_j(n)F(n+j,k)\right) \\ &= s_n+\sum_{j=1}^{d}\sigma_j(n)s_{n+j}. \end{aligned} \tag{7.17}$$

After multiplication by the common denominator one gets the holonomic recurrence equation sought.

Zeilberger's algorithm gives a much better possibility of computing identities since it computes holonomic recurrence equations generally of lowest order (iteration on the order d) for a given hypergeometric series. Moreover, this approach is also used to show the coincidence of two sums provided their initial values. This constitutes a normal form again, as we have seen in Section 4.3.1 (see [Geddes et al., 1992, Chapter 3]).

Example 7.4. *The sums $\sum_{k=0}^{n}\binom{n}{k}^3$ and $\sum_{k=0}^{n}\binom{n}{k}^2\binom{2k}{n}$ are proved to coincide by Zeilberger's algorithm as they lead to the same holonomic recurrence equation*

$$-(n+2)^2 s_{n+2}+(7n^2+21n+16)s_{n+1}+8(n+1)^2 s_n=0, \tag{7.18}$$

and have the same initial values
$\sum_{k=0}^{0}\binom{0}{k}^3=\sum_{k=0}^{0}\binom{0}{k}^2\binom{2k}{0}=1$ *and* $\sum_{k=0}^{1}\binom{1}{k}^3=\sum_{k=0}^{1}\binom{1}{k}^2\binom{2k}{1}=2.$

Note, however, that Fasenmyer's, Gosper's, Wilf-Zeilberger's and Zeilberger's methods are only reduced to the hypergeometric case. To use Gosper's algorithm, one has to check whether the given a_k is a hypergeometric term with respect to the variable k. When this is not the case, Gosper's, WZ's and Zeilberger's methods cannot be applied. From his algorithm [Koepf, 2014, Algorithm 2.2], Koepf observed that Gosper's method could miss some results when rational-linear Γ inputs are considered rather than only integer-linear ones. It turns out that this observation constitutes the connection to the general case of m-fold hypergeometric terms (see [Koepf, 1995a] and [Koepf, 2014, Chapter 8]).

Example 7.5. *To the Watson's function*

$${}_3F_2\left(\left.\begin{array}{ccc} -n & b & c \\ \frac{-n+b+1}{2} & 2c & \end{array}\right| 1\right),$$

Zeilberger's algorithm does not apply directly. However, using Koepf's extended version yields the recurrence equation

$$(b-2c-n-1)(n+1)a_n-(b-n-1)(2c+n+1)a_{n+2}=0. \tag{7.19}$$

Note that the computation of this holonomic RE is made possible after application of [Koepf, 2014, Algorithm 8.4] for finding the corresponding m to use. In this example one finds $m=2$.

The value of the computation of m-fold hypergeometric term solutions of holonomic recurrence equations could not easily be seen from the known hypergeometric database (see [Koepf, 2014, Chapter 3]). This might be linked to the influence of the combinatorial interpretation often present in the use of hypergeometric summations in the last century. From this point of view, the m-fold hypergeometric case ($m \in \mathbb{N}_{\geqslant 2}$) is particularly hidden and Zeilberger's, Petkovšek's and van Hoeij's algorithms or their modifications for any hypergeometric situation (multivariate for example) remain the best approach possible. However, as pointed out in [Koepf, 1995a], the general m-fold hypergeometric case might be the source of a wider family of hypergeometric identities. This is shown in particular in the computation of power series where some computed holonomic recurrence equations do not have hypergeometric term solutions but only m-fold hypergeometric ones ($m \in \mathbb{N}_{\geqslant 2}$). Note that for the definite summation case the availability of an algorithm which computes m-fold hypergeometric term solutions of holonomic REs can be combined with Koepf's extension of Zeilberger's algorithm to generate new identities. We will not deal with definite summation, instead as the importance of m-fold hypergeometric terms for hypergeometric series is already shown, we would like to come back to power series computations where the need of such terms is essential for the goal of this thesis. Before going to the description of our algorithm in the next section, in order to show the importance of the algorithm of this chapter we will give some examples of expressions where the implementations of van Hoeij's and Koepf's algorithms in Maple cannot detect the desired power series representations.

7.2 Limits of the Current Computation of Power Series

In this section, we consider expressions representing hypergeometric type functions whose Taylor coefficients satisfy recurrence equations that do not have easily computable solutions. These are holonomic REs with more than two terms and which do not have hypergeometric term solutions over the field of rationals.

Firstly, we consider examples for which the current Maple FPS command gives results that are more complicated than necessary. This happens in general when a hypergeometric term computed over an extension field of $\mathbb{Q}$ is used whereas an m-fold hypergeometric term equivalent over $\mathbb{Q}$ exists, or when the linear combination of hypergeometric terms computed is not easily manageable and Maple's FPS command tries to use another approach, in particular the algorithm for power series computation of rational functions described in [Koepf, 1993].

Our Maxima implementations yields

```
(%i1) RE: FindRE(atan(z)+exp(z),z,a[n]);
```

$$(\%o1)\quad (1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}+(n-3)\cdot(1+n)\cdot(2+n)\cdot a_{n+2} + (1+n)\cdot\left(2+2\cdot n+n^2\right)\cdot a_{n+1}+(n-3)\cdot n\cdot(1+n)\cdot a_n-(n-1)\cdot n\cdot a_{n-1}=0$$

```
(%i2) HypervanHoeij(RE,a[n],C);
```

$$(\%o2)\qquad \left\{\frac{(-\mathrm{i})^n}{n}, \frac{(-1)^{\frac{n}{2}}}{n}, \frac{1}{n!}\right\}$$

Of course, we allow computations over extension fields of $\mathbb{Q}$ since otherwise we may not get all the hypergeometric terms needed to compute the corresponding power series. Maple combines the above results and gives therefore the following representation

```
> bind(FormalPowerSeries);
```

$$FPS,\ HolonomicDE,\ SimpleRE,\ convert/RESol$$

```
> FPS(arctan(z)+exp(z),z,n);
```

$$\sum_{n=0}^{\infty}\left(\frac{1}{(n+1)!} - \frac{\mathrm{i}\cdot \mathrm{i}^n}{2(n+1)} + -\frac{\mathrm{i}\cdot(-\mathrm{i})^n}{2(n+1)}\right) z^{n+1}$$

which could be simplified further if the 2-fold hypergeometric term corresponding to $\arctan(z)$ would be computed. Note, moreover, that this representation is not correct because the first term of the Taylor expansion of $\arctan(z) + \exp(z)$ is not zero as shown below.

```
(%i3) taylor(atan(z)+exp(z),z,0,0);
```

$$(\%o3)/T/\ \ 1 + \ldots$$

Our FPS command gives the following correct and further simplified output.

```
(%i4) FPS(atan(z)+exp(z),z,n);
```

$$(\%o4)\qquad \left(\sum_{n=0}^{\infty}\frac{(-1)^n\cdot z^{1+2\cdot n}}{2\cdot n+1}\right) + \sum_{n=0}^{\infty}\frac{z^n}{n!}$$

The issue of initial value and the computation of linear combinations of m-fold hypergeometric terms will be treated in the next chapter. For the moment, we would like to focus on the summands used to represent power series. Let us move to another example of the same kind.

Our Maxima implementations yields

```
(%i5) RE:FindRE(log(1+z+z^2)+cos(z),z,a[n]);
```

$$(\%o5)\ -3\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot(4+n)\cdot a_{n+4} - (1+n)\cdot(2+n)\cdot(3+n)\cdot(7+5\cdot n)\cdot a_{n+3} + (1+n)\cdot(2+n)\cdot\left(-3-56\cdot n+8\cdot n^2\right)\cdot a_{n+2} + (1+n)\cdot\left(-7+74\cdot n-102\cdot n^2+23\cdot n^3\right)\cdot a_{n+1} + 2\cdot n\cdot\left(-86+121\cdot n-74\cdot n^2+15\cdot n^3\right)\cdot a_n + (n-1)\cdot\left(-409+412\cdot n-150\cdot n^2+19\cdot n^3\right)\cdot a_{n-1} + (n-2)\cdot\left(-406+317\cdot n-78\cdot n^2+7\cdot n^3\right)\cdot a_{n-2} + (n-3)\cdot\left(-255+149\cdot n-28\cdot n^2+2\cdot n^3\right)\cdot a_{n-3} + (n-4)\cdot(7\cdot n-29)\cdot a_{n-4} + 2\cdot(n-5)^2\cdot a_{n-5} = 0$$

```
(%i6) HypervanHoeij(RE,a[n],C);
```

$$(\%o6)\qquad \left\{\frac{\left(\frac{-1-\sqrt{3}\cdot\mathrm{i}}{2}\right)^n}{n}, \frac{\left(\frac{\sqrt{3}\cdot\mathrm{i}-1}{2}\right)^n}{n}, \frac{(-\mathrm{i})^n}{n!}, \frac{(-1)^{\frac{n}{2}}}{n!}\right\}$$

Maple's FPS command combines these hypergeometric terms and gives

```
>   FPS(log(1+z+z^2)+cos(z),z,n);
```

$$\sum_{n=0}^{\infty}\left(\frac{\left(\frac{1}{10}+\frac{i}{5}\right)(i\sqrt{3}+2\sqrt{3}+3-6i)(-i)^{n+1}}{(i\sqrt{3}+3)(n+1)!}+\frac{i(-3i+\sqrt{3})i^{n+1}}{2(i\sqrt{3}+3)(n+1)!}\right.$$
$$\left.+\frac{\left(-\frac{2}{5}+\frac{i}{5}\right)(i\sqrt{3}+2\sqrt{3}-1+2i)\sqrt{3}\left(-\frac{1}{2}+\frac{i\sqrt{3}}{2}\right)^{n+1}}{(i\sqrt{3}+3)(n+1)}-\frac{\left(-\frac{1}{2}-\frac{i\sqrt{3}}{2}\right)^{n+1}}{n+1}\right)z^{n+1}$$

Evidently, the result can be further simplified if one uses the 2-fold hypergeometric term of the expansion of $\cos(z)$. After the writing of the hypergeometric term over $\mathbb{C}$ corresponding to the general coefficient of $\log(1+z+z^2)$ in its algebraic form, our Maxima FPS command yields

```
(%i7)  FPS(log(1+z+z^2)+cos(z),z,n);
```

$$(\%o7)\qquad \left(\sum_{n=0}^{\infty}-\frac{2\cdot\cos\left(\frac{2\cdot\pi\cdot(1+n)}{3}\right)\cdot z^{1+n}}{n+1}\right)+\sum_{n=0}^{\infty}\frac{(-1)^n\cdot z^{2\cdot n}}{(2\cdot n)!}$$

Observe that here again the first term of the expansion corresponding to $\log(1+0)+\cos(0)=1$ is missing on Maple's FPS output. This shows in a certain sense how difficult the managing of hypergeometric terms over extension fields for power series representations is. Next, we consider examples for which Maple's FPS command uses another approach to represent power series and whose results are again more complicated than necessary.

We start with the tangent analogue of the Chebyshev polynomials $\tan(k\arctan(z))$ which are rational functions. Note that these functions are not identified as holonomic in Maxima and Maple because of the implementation of the tangent function. Therefore in order to compute their corresponding holonomic recurrence equations using *FindRE* some simplifications are to be applied. This is done with our Maxima function *HolonomicDE* only when after all the iterations up to *Nmax*$=5$ no holonomic differential equation is found. Then the code applies Maxima's commands *trigsimp* and *trigexpand* to the given expression so that if the ratio of the obtained expression and the given one does not give 1 then *HolonomicDE* is called a second time.

Our Maxima implementations yields

```
(%i8)  RE:FindRE(tan(5*atan(z)),z,a[n]);
```

$$(\%o8)\qquad 5\cdot(n-1)\cdot a_n-20\cdot(3\cdot n-5)\cdot a_{n-2}+6\cdot(21\cdot n-89)\cdot a_{n-4}$$
$$-20\cdot(3\cdot n-17)\cdot a_{n-6}+5\cdot(n-9)\cdot a_{n-8}=0$$

```
(%i9)  HypervanHoeij(RE,a[n],C);
```

$$(\%o9)\qquad \left\{\left(5-2\cdot\sqrt{5}\right)^{\frac{n}{2}},\left(5+2\cdot\sqrt{5}\right)^{\frac{n}{2}},\left(-\sqrt{2\cdot\sqrt{5}+5}\right)^{n}\right\}$$

However, these computed hypergeometric terms will not be used in Maple's FPS representation. Koepf's algorithm for rational functions is used instead. Indeed, by partial fraction decomposition one has

```
(%i10) r:trigexpand(tan(5*atan(z)));
```

$$(\%o10) \quad \frac{5 \cdot z - 10 \cdot z^3 + z^5}{5 \cdot z^4 - 10 \cdot z^2 + 1}$$

```
(%i11) partfrac(r,z);
```

$$(\%o11) \quad \frac{z}{5} - \frac{40 \cdot z^3 - 24 \cdot z}{5 \cdot (5 \cdot z^4 - 10 \cdot z^2 + 1)}$$

hence the following Maple FPS result.

```
>  FPS(tan(5*arctan(z)),z,n);
```

$$\frac{z}{5} + \left(\sum_{n=0}^{\infty} \left(\sum_{\_\alpha = RootOf(5\_Z^4 - 10\_Z^2 + 1)} \frac{\_\alpha^2 + 1}{5\_\alpha^{n+1}} \right) z^n \right)$$

We mention that $\tan(k \arctan(z)), k = 2, 3, 4, 5$ are hypergeometric type functions of type 2. Using our Maxima FPS command for this previous example yields the following hypergeometric type series.

```
(%i12) FPS(tan(5*atan(z)),z,n);
```

$$(\%o12) \; \frac{z}{5} + \left(\sum_{n=0}^{\infty} \frac{4 \cdot \left(3 \cdot (5 - 2 \cdot \sqrt{5})^n - (5 - 2 \cdot \sqrt{5})^n \cdot \sqrt{5} + 3 \cdot (5 + 2 \cdot \sqrt{5})^n + \sqrt{5} \cdot (5 + 2 \cdot \sqrt{5})^n\right) \cdot z^{1+2 \cdot n}}{5} \right)$$

The computation of the corresponding 2-fold hypergeometric terms will be presented in the next section. Similarly to this example is the reciprocal of the Chebyshev polynomial $\cos(4 \arccos(z))$ that we present next.

Our Maxima *FindRE* yields the following RE.

```
(%i13) RE:FindRE(1/cos(4*acos(z)),z,a[n]);
```

$$(\%o13) \quad (1 + n) \cdot a_{n+1} - 8 \cdot (1 + n) \cdot a_{n-1} + 8 \cdot (1 + n) \cdot a_{n-3} = 0$$

We present directly the Maple FPS representation since the corresponding hypergeometric terms are not used.

```
>  FPS(1/cos(4*arccos(z)),z,n);
```

$$\sum_{n=0}^{\infty} \left(\sum_{\_\alpha = RootOf\left(8\ \_Z^4 - 8\ \_Z^2 + 1\right)} - \frac{\_\alpha\ (4\ \_\alpha^2 - 3)}{4\ \_\alpha^{n+1}} \right) z^n$$

With these two latter power series representations one can maybe find further simplified formulas since the roots of the polynomials $5z^4 - 10z^2 + 1$ and $8z^4 - 8z^2 + 1$ can easily be computed. However, even when the poles of a given rational expression are more explicit, the partial fraction decomposition approach used to represent power series typically gives more complicated outputs. This is shown by the following last example of this first list of examples.

We consider the function $\frac{1}{(q1-z^2)(q2-z^3)}$ for some constants $q1$ and $q2$. Using our Maxima command *FindRE* one gets the following RE.

```
(%i14) declare(q1,constant)$
(%i15) declare(q2,constant)$
(%i16) RE:FindRE(1/((q1-z^2)*(q2-z^3)),z,a[n]);
```

(%o16) $$q1\cdot q2\cdot(1+n)\cdot a_{n+1}-q2\cdot(1+n)\cdot a_{n-1}-q1\cdot(1+n)\cdot a_{n-2}+(1+n)\cdot a_{n-4}=0$$

The obtained holonomic RE is of order 6 and its hypergeometric term solutions cannot easily be used. Maple's FPS command gives

```
>  FPS(1/((q1-z^2)*(q2-z^3)),z,n);
```

$$\begin{aligned}
&\sum_{n=0}^{\infty}(1/2)(-2\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q2^{(}4/3)(-1)^{2/3}q1+4\sqrt{(q1)}^n(-\sqrt{(q1)})^n\\
&(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q2^{2/3}(-1)^{2/3}q1^2+2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(-\sqrt{(q1)})^n((-1)^{2/3}q2^{1/3})^n\\
&q2^{2/3}(-1)^{1/3}q1^2+2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n q2^{(}4/3)(-1)^{2/3}q1-(-(-1)^{1/3}q2^{1/3})^n\\
&\sqrt{(q1)}^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n(-1)^{2/3}q1^{3/2}q2+(-(-1)^{1/3}q2^{1/3})^n(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n\\
&(-1)^{2/3}q1^{3/2}q2+(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n(-1)^{1/3}q1^{3/2}q2-(-(-1)^{1/3}q2^{1/3})^n\\
&(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n(-1)^{1/3}q1^{3/2}q2-2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n\\
&q2^{2/3}(-1)^{2/3}q1^2-2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(-\sqrt{(q1)})^n((-1)^{2/3}q2^{1/3})^n q2^{2/3}(-1)^{2/3}q1^2-2\sqrt{(q1)}^n\\
&(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q2^{2/3}(-1)^{1/3}q1^2+2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(-\sqrt{(q1)})^n\\
&((-1)^{2/3}q2^{1/3})^n q2^{(}4/3)q1-2\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q2^{(}4/3)q1-2\sqrt{(q1)}^n(-\sqrt{(q1)})^n\\
&(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n(-1)^{2/3}q1^3+2\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n\\
&(-1)^{1/3}q1^3-2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n q2^{2/3}q1^2-2(-(-1)^{1/3}q2^{1/3})^n(-\sqrt{(q1)})^n\\
&(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q1^{3/2}q2+(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n\\
&q2^2(-1)^{2/3}+(-(-1)^{1/3}q2^{1/3})^n(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q2^2(-1)^{2/3}-(-(-1)^{1/3}q2^{1/3})^n\\
&\sqrt{(q1)}^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q2^2(-1)^{1/3}-(-(-1)^{1/3}q2^{1/3})^n(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n\\
&q2^2(-1)^{1/3}+2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q1^{3/2}q2+2\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n\\
&((-1)^{2/3}q2^{1/3})^n q2^{2/3}q1^2+2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(-\sqrt{(q1)})^n((-1)^{2/3}q2^{1/3})^n q1^3+2(-(-1)^{1/3}q2^{1/3})^n\\
&\sqrt{(q1)}^n(-\sqrt{(q1)})^n(q2^{1/3})^n q1^3-2(-(-1)^{1/3}q2^{1/3})^n\sqrt{(q1)}^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n\\
&q2^2-2(-(-1)^{1/3}q2^{1/3})^n(-\sqrt{(q1)})^n(q2^{1/3})^n((-1)^{2/3}q2^{1/3})^n q2^2)z^n/(((-1)^{2/3}q2^{1/3})^n(q2^{1/3})^n\\
&((-1)^{2/3}-1)(-\sqrt{(q1)})^n((-1)^{2/3}q2^{1/3}+\sqrt{(q1)})(\sqrt{(q1)}+q2^{1/3})\sqrt{(q1)}^n q1((-1)^{2/3}q2^{1/3}-\sqrt{(q1)})\\
&(-\sqrt{(q1)}+q2^{1/3})(-(-1)^{1/3}q2^{1/3})^n(-\sqrt{(q1)}+(-1)^{1/3}q2^{1/3})(\sqrt{(q1)}+(-1)^{1/3}q2^{1/3})((-1)^{1/3}+1)q2)
\end{aligned}\tag{7.20}$$

which is a too complicated output to be of essential help. However using our Maxima FPS command one gets the much simpler representation below.

```
(%i17) FPS(1/((q1-z^2)*(q2-z^3)),z,n);
```

$$(\%o17)\quad \left(\sum_{n=0}^{\infty} -\frac{q1 \cdot q2^{-1-n} \cdot z^{2+3\cdot n}}{q2^2 - q1^3}\right) + \left(\sum_{n=0}^{\infty} -\frac{z^{1+3\cdot n}}{\left(q2^2 - q1^3\right) \cdot q2^n}\right) + \left(\sum_{n=0}^{\infty} \frac{z^{1+2\cdot n}}{\left(q2^2 - q1^3\right) \cdot q1^n}\right) + \left(\sum_{n=0}^{\infty} -\frac{q1^2 \cdot q2^{-1-n} \cdot z^{3\cdot n}}{q2^2 - q1^3}\right) + \sum_{n=0}^{\infty} \frac{q2 \cdot q1^{-1-n} \cdot z^{2\cdot n}}{q2^2 - q1^3}$$

Next we consider examples for which van Hoeij's algorithm does not find any solution and therefore Maple's FPS command gives no result. For each example: we compute the corresponding holonomic recurrence equations with our Maxima command *FindRE*, then we use our Maxima command *HypervanHoeij* to show that the computed RE does not have hypergeometric term solutions, and we give Maple's FPS result followed by our Maxima FPS one.

1. $\exp(z^2) + \cos(z^2)$:

```
(%i18) RE:FindRE(exp(z^2)+cos(z^2),z,a[n]);
```

$$(\%o18)\quad (n-3)\cdot(n-1)\cdot(1+n)\cdot a_{n+1} - 2\cdot(n-3)\cdot(n-1)\cdot a_{n-1} + 4\cdot(n-3)\cdot a_{n-3} - 8\cdot a_{n-5} = 0$$

```
(%i19) HypervanHoeij(RE,a[n],C);
```

$$(\%o19)\quad \{\}$$

```
> FPS(exp(z^2)+cos(z^2),z,n);
```

$$FPS(e^{z^2} + \cos(z^2), z, n)$$

```
(%i20) FPS(exp(z^2)+cos(z^2),z,n);
```

$$(\%o20)\quad \left(\sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{4\cdot n}}{(2\cdot n)!}\right) + \sum_{n=0}^{\infty} \frac{z^{2\cdot n}}{n!}$$

2. $\cosh(z^3) + \sin(z^2)$:

```
(%i21) RE:FindRE(cosh(z^3)+sin(z^2),z,a[n]);
```

$$\begin{aligned}(\%o21)\quad & -4\cdot(n-3)\cdot(n-2)\cdot(n-1)\cdot(1+n)\cdot a_{n+1} - 36\cdot(n-7)\cdot(n-4)\\ &\cdot(n-3)\cdot(n-1)\cdot a_{n-1} + 16\cdot(n-6)\cdot(n-3)\cdot\left(58-16\cdot n+n^2\right)\cdot a_{n-3}\\ &+ 36\cdot(n-9)\cdot(n-7)\cdot\left(141-38\cdot n+2\cdot n^2\right)\cdot a_{n-5}\\ &+ (n-7)\cdot\left(-159562+38863\cdot n-3078\cdot n^2+81\cdot n^3\right)\cdot a_{n-7}\\ &+ 144\cdot\left(158-26\cdot n+n^2\right)\cdot a_{n-9} - 324\cdot\left(68-19\cdot n+n^2\right)\cdot a_{n-11}\\ &- 9\cdot\left(15859-2268\cdot n+81\cdot n^2\right)\cdot a_{n-13} - 2592\cdot a_{n-15} - 2916\cdot a_{n-17} = 0\end{aligned}$$

```
(%i22) HypervanHoeij(RE,a[n],C);
```

$$(\%o22) \quad \{\}$$

```
>   FPS(cosh(z^3)+sin(z^2),z,n);
```

$$FPS(\cosh(z^3)+\sin(z^2),z,n)$$

```
(%i23) FPS(cosh(z^3)+sin(z^2), z, n);
```

$$(\%o23) \quad \left(\sum_{n=0}^{\infty} \frac{z^{3+6\cdot n}}{(2\cdot n+1)\cdot(2\cdot n)!}\right)+\left(\sum_{n=0}^{\infty} \frac{(-1)^n\cdot z^{2+4\cdot n}}{(2\cdot n+1)\cdot(2\cdot n)!}\right)+\sum_{n=0}^{\infty} \frac{(-1)^n\cdot z^{3\cdot n}}{n!}$$

3. $\arcsin(z^2)^2+\arccos(z)$:

```
(%i24) RE:FindRE(asin(z^2)^2+acos(z),z,a[n]);
```

$$(\%o24) \; -3\cdot(n-1)\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3} +(n-3)\cdot(n-1)\cdot(1+n)\cdot(5\cdot n-1)\cdot a_{n+1}+(n-1)\cdot\left(96-87\cdot n+22\cdot n^2+n^3\right)\cdot a_{n-1} -(n-3)^3\cdot(5\cdot n-1)\cdot a_{n-3}+2\cdot(n-5)^3\cdot(n-3)\cdot a_{n-5}=0$$

```
(%i25) HypervanHoeij(RE,a[n],C);
```

$$(\%o25) \quad \{\}$$

```
>   FPS(arcsin(z^2)^2+arccos(z),z,n);
```

$$FPS(\arcsin(z^2)^2+\arccos(z),z,n)$$

```
(%i26) FPS(asin(z^2)^2+acos(z), z, n);
```

$$(\%o26) \quad \left(\sum_{n=0}^{\infty} -\frac{(2\cdot n)!\cdot z^{1+2\cdot n}}{(2\cdot n+1)\cdot 4^n\cdot n!^2}\right)+\left(\sum_{n=0}^{\infty} \frac{2\cdot 4^n\cdot(1+n)!^2\cdot z^{4\cdot(1+n)}}{(1+n)^2\cdot(2\cdot(1+n))!}\right)+\frac{\pi}{2}$$

4. $\sqrt{\sqrt{8\cdot z^3+1}-1}+\sqrt{13\cdot z^4+7}$:

```
(%i27) RE:FindRE(sqrt(sqrt(8*z^3+1)-1)
                 +sqrt(13*z^4+7),z,a[n]);
```

$$(\%o27) \quad -3087\cdot(n-2)\cdot(1+n)\cdot(2\cdot n-1)\cdot a_{n+1} +6370\cdot(n-4)\cdot n\cdot(2\cdot n-3)\cdot a_n-3087\cdot(n-2)\cdot(4\cdot n-11)\cdot(4\cdot n-5)\cdot a_{n-2} +637\cdot\left(-35100+24417\cdot n-5315\cdot n^2+362\cdot n^3\right)\cdot a_{n-3}+4732\cdot(n-4)\cdot(2\cdot n-11) \cdot(3\cdot n-11)\cdot a_{n-4}+637\cdot(4\cdot n-27)\cdot(4\cdot n-21)\cdot(101\cdot n-1100)\cdot a_{n-6} +1183\cdot\left(-256630+96761\cdot n-12247\cdot n^2+522\cdot n^3\right)\cdot a_{n-7}+4394\cdot(n-10)\cdot(n-8) \cdot(2\cdot n-19)\cdot a_{n-8}+65065\cdot(3\cdot n-34)\cdot(4\cdot n-43)\cdot(4\cdot n-37)\cdot a_{n-10}+2197\cdot(n-13) \cdot\left(21580-3523\cdot n+142\cdot n^2\right)\cdot a_{n-11}+120835\cdot(n-16)\cdot(4\cdot n-59)\cdot(4\cdot n-53)\cdot a_{n-14}=0$$

```
(%i28) HypervanHoeij(RE,a[n],C);
```

$$(\%o28) \quad \{\}$$

```
>  FPS(sqrt(sqrt(8*z^3+1)-1)+sqrt(13*z^4+7),z,n);
```

$$FPS(\sqrt{\sqrt{8z^3+1}-1}+\sqrt{13z^4+7}, z, n)$$

```
(%i29) FPS(sqrt(sqrt(8*z^3+1)-1)+sqrt(13*z^4+7),z,n);
```

$$(\%o29) \quad \left(\sum_{n=0}^{\infty} \frac{2\cdot\left(\frac{1}{4}\right)_n\cdot\left(\frac{3}{4}\right)_n\cdot(-8)^n\cdot 4^n\cdot z^{\frac{6+12\cdot n}{4}}}{(2\cdot n+1)\cdot(2\cdot n)!}\right)+\sum_{n=0}^{\infty}-\frac{7^{\frac{1}{2}-n}\cdot(-13)^n\cdot(2\cdot n)!\cdot z^{4\cdot n}}{(2\cdot n-1)\cdot 4^n\cdot n!^2}$$

5. $\exp(\arcsin(z))+\exp(\operatorname{arcsinh}(z))$:

```
(%i30) RE:FindRE(exp(asin(z))+exp(asinh(z)),z,a[n]);
```

$$(\%o30) \; 2\cdot(n-2)\cdot(n-1)\cdot(1+n)\cdot(2+n)\cdot a_{n+2}-(n-1)\cdot\left(-4+26\cdot n-9\cdot n^2+n^3\right)\cdot a_n$$
$$-2\cdot(n-3)\cdot\left(5+2\cdot n-3\cdot n^2+n^3\right)\cdot a_{n-2}+(n-5)\cdot(n-3)\cdot\left(17-8\cdot n+n^2\right)\cdot a_{n-4}=0$$

```
(%i31) HypervanHoeij(RE,a[n],C);
```

$$(\%o31) \quad \{\}$$

```
>  FPS(exp(arcsin(z))+exp(arcsinh(z)),z,n);
```

$$FPS(\exp(\arcsin(z))+z+\sqrt{z^2+1}, z, n)$$

Note that here Maple immediately simplifies $\exp(\operatorname{arcsinh}(z))$ to $z+\sqrt{z^2+1}$.

```
(%i32) FPS(exp(asin(z))+exp(asinh(z)),z,n);
```

$$(\%o32) \quad \left(\sum_{n=0}^{\infty} \frac{(-1)^n\cdot(2\cdot n)!\cdot z^{2+2\cdot n}}{2\cdot(n+1)\cdot 4^n\cdot n!^2}\right)+\left(\sum_{n=0}^{\infty} \frac{\left(-\frac{i-1}{2}\right)_n\cdot\left(\frac{1+i}{2}\right)_n\cdot 4^n\cdot z^{1+2\cdot n}}{(1+2\cdot n)!}\right)$$
$$+\left(\sum_{n=0}^{\infty} \frac{\left(-\frac{i}{2}\right)_n\cdot\left(\frac{i}{2}\right)_n\cdot 4^n\cdot z^{2\cdot n}}{(2\cdot n)!}\right)+z+1$$

6. $\sin(8\operatorname{arcsinh}(z))+\sqrt{1+z^4}$:

```
(%i33) RE:FindRE(sin(8*asinh(z))+sqrt(1+z^4),z,a[n]);
```

$$(\%o33) \; 32\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}+5\cdot(1+n)\cdot\left(416+11\cdot n+7\cdot n^2\right)\cdot a_{n+1}$$
$$+\left(-1345+1553\cdot n-695\cdot n^2+103\cdot n^3\right)\cdot a_{n-1}+2\cdot\left(-13954+4863\cdot n-511\cdot n^2\right.$$
$$\left.+52\cdot n^3\right)\cdot a_{n-3}+2\cdot\left(-13373+6453\cdot n-1013\cdot n^2+53\cdot n^3\right)\cdot a_{n-5}+\left(-96554+23617\cdot n\right.$$
$$\left.-2302\cdot n^2+103\cdot n^3\right)\cdot a_{n-7}+5\cdot(n-11)\cdot(n-9)\cdot(7\cdot n-83)\cdot a_{n-9}$$
$$+34\cdot(n-13)\cdot\left(185-22\cdot n+n^2\right)\cdot a_{n-11}=0$$

```
(%i34) HypervanHoeij(RE,a[n],C);
```

$$(\%o34) \quad \{\}$$

```
>   FPS(sin(8*arcsinh(z))+sqrt(1+z^4),z,n);
```

$$FPS(\sin(8\operatorname{arcsinh}(z)) + \sqrt{z^4+1}, z, n)$$

```
(%i35) FPS(sin(8*asinh(z))+sqrt(1+z^4),z,n);
```

$$(\%o35) \quad \left(\sum_{n=0}^{\infty} \frac{(-1)^n \cdot (2\cdot n)! \cdot z^{4+4\cdot n}}{2\cdot(n+1)\cdot 4^n \cdot n!^2}\right) + \left(\sum_{n=0}^{\infty} \frac{2\cdot\left(-\frac{8\cdot i-1}{2}\right)_n \cdot \left(\frac{1+8\cdot i}{2}\right)_n \cdot (-1)^n \cdot 4^{1+n} \cdot z^{1+2\cdot n}}{(1+2\cdot n)!}\right) + 1$$

More generally, linear combinations of Laurent-Puiseux series of hypergeometric type cannot be represented by the current Maple FPS command. The m-fold hypergeometric term solutions of all the above holonomic recurrence equations will be computed in the next section.

7.3 Algorithm mfoldHyper

Let $\mathbb{K}$ be a field of characteristic zero. As usual, let us consider the generic holonomic recurrence equation

$$P_d(n)a_{n+d} + P_{d-1}(n)a_{n+d-1} + \cdots + P_0(n)a_n = 0, \tag{7.21}$$

$P_d(n), \ldots, P_0(n) \in \mathbb{K}[n]$, $P_d(n) \cdot P_0(n) \neq 0$.

By definition, a sequence a_n is said to be m-fold hypergeometric, $m \in \mathbb{N}$, if there exists a fixed rational function $r(n) \in \mathbb{K}(n)$ such that

$$r(n) = \frac{a_{n+m}}{a_n}. \tag{7.22}$$

Numbers of proposals have been given to compute m-fold hypergeometric term solutions of holonomic recurrence equations. Among the most recent work in this direction one could cite [Horn et al., 2012], which is a revisited and improved approach of the one described in [Petkovšek and Salvy, 1993]. In the latter, a key step of the proposed algorithm relies on the determination of the linear operator's right factors of the given holonomic RE. Such a factorization is not unique in general because the factors do not commute. In [Horn et al., 2012], the authors adapted van Hoeij's approach as explained in [Cluzeau and van Hoeij, 2006] and define a concept like the m-Newton polygon for m-fold hypergeometric term solutions of a given holonomic RE. This approach computes special types of right factors corresponding to m-fold hypergeometric term solutions using the shift operator of order m with the hypothesis that no rational solution exists.

Having considered all these developments, we propose to approach the problem from another point of view. In the first glance, one should remark that m-fold hypergeometric sequences have rational functions as the ratio of terms with index difference equal to m. Consequently, if we can find a way to transform this property to the simple one of hypergeometric sequence then iteratively up to the order of the given holonomic RE by van Hoeij's algorithm we have done.

From the characterization (7.22) one can deduce that for $0 \leqslant j \leqslant m-1$ the following is valid

$$r(m \cdot n + j) = \frac{a_{m\cdot(n+1)+j}}{a_{m\cdot n+j}}. \tag{7.23}$$

Therefore instead of considering the representation (7.22) one could rather see an m-fold hypergeometric term with m related rational functions as defined in (7.23). This latter representation is the one used to find the "simple" formula of an m-fold hypergeometric term. Moreover, for fixed $m \in \mathbb{N}$ and $j \in \mathbb{N}_{\geqslant 0}$, if we compute an m-fold hypergeometric term solution of (7.21) with ratio $r(m \cdot n + j)$ for some rational function r, then this gives the information that there are $m-1$ other similar m-fold hypergeometric term solutions of (7.21). Thus for any positive integer m, the computation of an m-fold hypergeometric term solution of (7.21) with representation (7.22) reduces to the computation of an m-fold hypergeometric term solution of (7.21) with representation (7.23) for a fixed $j \in [\![0, m-1]\!]$ since the other representations can be similarly computed. By default in our algorithm we choose $j = 0$.

In certain cases, depending on the field $\mathbb{K}$ or the index variable subset of $\mathbb{Z}$, holonomic recurrence equations can have m-fold hypergeometric term solutions that cannot be computed over $\mathbb{K}$, but need an extension field.. This situation occurs with the power series of $\exp(z)\sin(z)$ for $\mathbb{K} = \mathbb{Q}$. Our implementation *HolonomicDE(f,F(z),[destep])* has an optional variable *destep* whose default value is 1. This number represents the minimum positive difference possible between the derivatives of *F(z)* in the holonomic differential equation sought. Our program *FindRE* is also adapted for such computations. This particular tool turns out to be important in few cases. Let us now examine the situation with $\exp(z)\sin(z)$.

```
(%i1) DE1:HolonomicDE(exp(z)*sin(z),F(z));
```

$$(\%o1) \qquad \frac{d^2}{d\,z^2} \cdot \mathrm{F}(z) - 2 \cdot \left(\frac{d}{d\,z} \cdot \mathrm{F}(z)\right) + 2 \cdot \mathrm{F}(z) = 0$$

```
(%i29) DE2:HolonomicDE(exp(z)*sin(z),F(z),2);
```

$$(\%o2) \qquad \frac{d^4}{d\,z^4} \cdot \mathrm{F}(z) + 4 \cdot \mathrm{F}(z) = 0$$

The compatibility of these two differential equations is shown below (see Section 4.3.1 on page 55).

```
(%i3) CompatibleDE(DE1,DE2,F(z));
```

The two differential equations are compatible

(%o3) true

Let us now compute the corresponding holonomic REs.

```
(%i31) RE1:FindRE(exp(z)*sin(z),z,a[n]);
```

$$(\%o4) \qquad (1+n) \cdot (2+n) \cdot a_{n+2} - 2 \cdot (1+n) \cdot a_{n+1} + 2 \cdot a_n = 0$$

```
(%i5) RE2: FindRE(exp(z)*sin(z),z,a[n],2);
```

$$(\%o5) \qquad (1+n)\cdot(2+n)\cdot(3+n)\cdot(4+n)\cdot a_{n+4}+4\cdot a_n=0$$

Observe that for every solution of *RE2* no relationship can be deduced between two of its terms whose index difference is not a multiple of 4. Therefore considering indices in $4\mathbb{N}=\{0,4,8,\ldots\}$ might be more appropriate. On the other hand, it is trivial that *RE2* is a characteristic holonomic recurrence equation of a 4-fold hypergeometric term. However, this 4-fold hypergeometric term is completely hidden in *RE1* when looking for solutions over $\mathbb{Q}$. Indeed, since the corresponding four 4-fold symmetric terms are linearly independent over $\mathbb{Q}$, a substitution in the left-hand side of *RE1* could never yield zero. Note, however, that *RE1* and *RE2* have hypergeometric term solutions over $\mathbb{C}$.

```
(%i6) HypervanHoeij(RE1,a[n],C);
```

$$(\%o6) \qquad \left\{\frac{(1-i)^n}{n!},\frac{(1+i)^n}{n!}\right\}$$

```
(%i7) HypervanHoeij(RE2,a[n],C);
```

$$(\%o7) \qquad \left\{\frac{\left(-(-1)^{\frac{1}{4}}\cdot\sqrt{2}\right)^n}{n!},\frac{\left((-1)^{\frac{1}{4}}\cdot\sqrt{2}\right)^n}{n!},\frac{\left(-(-1)^{\frac{1}{4}}\cdot\sqrt{2}\cdot i\right)^n}{n!},\frac{\left((-1)^{\frac{1}{4}}\cdot\sqrt{2}\cdot i\right)^n}{n!}\right\}$$

The basis of hypergeometric term solutions of *RE1* spans a sub-space of the space of solutions of *RE2*. The corresponding 4-fold hypergeometric term solutions over $\mathbb{Q}$ can be written as a linear combination over $\mathbb{C}$ of the above bases of hypergeometric terms. The main thing that we point out from this example is that our approach to compute m-fold hypergeometric term solutions of a given holonomic RE depends on the shifts between summands in that RE and the field considered.

Next, we would like to give some properties and definitions that clarify this situation and help to compute m-fold hypergeometric term solutions of (7.21) in a given field $\mathbb{K}$ which we want to be the smallest algebraic extension field possible of $\mathbb{Q}$ in terms of inclusion.

The following lemma gives a condition on the order of a given holonomic RE for its m-fold hypergeometric term solutions to be computable over a given field $\mathbb{K}$.

Lemma 7.1. *Let h_n be an m-fold hypergeometric term, $m\in\mathbb{N}$. Assume*

$$\forall u\in\mathbb{N},\ u<m,\ \ \textit{there is no rational function } r_u(n)\in\mathbb{K}(n)\ :\ h_{u+n}=r_u(n)h_n. \tag{7.24}$$

Then there is no holonomic recurrence equation over $\mathbb{K}$ of order less than m satisfied by h_n.

Proof. Let h_n be an m-fold hypergeometric term such that

$$h_{n+m}=r(n)\cdot h_n \iff Q_m(n)\cdot h_{n+m}+Q_0(n)\cdot h_n=0, \tag{7.25}$$

where $Q_m(n),Q_0(n)\in\mathbb{K}[n]$ and $r(n)=-\frac{Q_0(n)}{Q_m(n)}\in\mathbb{K}(n)$.

Suppose that h_n satisfies a holonomic recurrence equation of order less than m. Then there exists an equation of the form

$$P_{m-1}a_{n+m-1} + P_{m-2}a_{n+m-2} + \cdots + P_1 a_{n+1} + P_0 a_n = 0, \tag{7.26}$$

with polynomials $P_j = P_j(n) \in \mathbb{K}[n], j \in [\![0, m-1]\!]$, and $P_0(n) \neq 0$, satisfied by h_n.

- If P_0 is the only non-zero polynomial in the equation then h_n must be zero, which is a contradiction by definition.
- We assume that at least one other polynomial factor in the equation is non-zero. Then h_n satisfying (7.26) yields the following equation after substitution of n by $m \cdot n$

$$P_{m-1}(mn)h_{mn+m-1} + P_{m-2}(mn)h_{mn+m-2} + \cdots + P_1(mn)h_{mn+1} + P_0(mn)h_{mn} = 0. \tag{7.27}$$

By assumption (7.24), we know that $\forall u \in \mathbb{N},\ u < m$, h_n is not a u-fold hypergeometric term. So the holonomic recurrence equation of lowest order over $\mathbb{K}$ satisfied by h_n is

$$Q_m(n) \cdot a_{n+m} + Q_0(n) \cdot a_n = 0,$$

which is a two-term recurrence relation whose subspace of m-fold hypergeometric term (m is fixed) solutions can be represented by the basis

$$(h_{mn+m-1}, h_{mn+m-2}, \ldots, h_{mn+1}, h_{mn}) \tag{7.28}$$

according to (7.25). Thus (7.27) cannot hold since the left-hand side is a linear combination of linearly independent terms with respect to $\mathbb{K}(n)$, which implies that all the polynomial coefficients must be zero. Therefore we get a contradiction. □

Remark. Observe that the linear independence with respect to $\mathbb{K}(n)$ of the elements of the basis (7.28) used in the proof of Lemma 7.1 can be interpreted in the following different way. Since $h_n \neq 0$, h_n satisfying (7.26) yields the following identity after dividing (7.26) by h_n

$$-P_0 = P_{m-1}\frac{h_{n+m-1}}{h_n} + P_{m-2}\frac{h_{n+m-2}}{h_n} + \ldots + P_1\frac{h_{n+1}}{h_n}. \tag{7.29}$$

By assumption (7.24), we know that $\forall u \in \mathbb{N},\ u < m$, the ratio $\frac{h_{n+u}}{h_n}$ is not a rational function over $\mathbb{K}(n)$. So, each non-zero term on the right-hand side of (7.29) is not rational over $\mathbb{K}$. This does not necessarily implies the non-rationality of the whole right-hand side. However by the linear independence of the elements of the basis (7.28) we can assume that this holds. Therefore for simplicity we can consider the equality between an irrational and a rational ($P_0(n)$) terms with respect to $\mathbb{K}(n)$ to conclude the proof. And this is what we will do in the proof of Theorem 7.1 where the situation is more sophisticated.

More generally, any shift of a holonomic recurrence equation of order less than m does not have m-fold hypergeometric term solutions.

Remark that checking the hypothesis of Lemma 7.1 is an important task for the algorithm. Fortunately, this can be done iteratively. Once the field $\mathbb{K}$ is fixed, if we have already looked for u-fold hypergeometric term solutions of a given holonomic RE for integers $u < m$, then we can safely proceed to the computation of m-fold hypergeometric term solutions knowing that m is less than the order of that recurrence equation.

Thus, we now know that all the m-fold hypergeometric term solutions of (7.21) have $m \leqslant d$. Furthermore, we can extend this view of m-fold hypergeometric sequences in order to determine which type of terms can appear in a holonomic recurrence equation that they satisfy. For that purpose, let us first introduce the following definition.

Definition 7.1 (m-fold Holonomic Recurrence Equation). *A holonomic recurrence equation is said to be m-fold holonomic, $m \in \mathbb{N}$, if it has at least two non-zero polynomial coefficients and the difference between indices of two appearing terms of the indeterminate sequence in that equation is a multiple of m. Choosing* 0 *as the trailing term order gives the general form*

$$P_d(n) \cdot a_{n+md} + P_{d-1}(n) \cdot a_{n+m(d-1)} + \cdots + P_1(n) \cdot a_{n+m} + P_0(n) \cdot a_n = 0, . \tag{7.30}$$

so that $P_d \cdot P_0 \neq 0$.

Obviously, every m-fold holonomic recurrence equation, $m \in \mathbb{N}$, is a 1-holonomic RE.

Assume an m-fold holonomic RE with representation (7.30) is given. We are going to present how to compute a basis of all m-fold hypergeometric term solutions of (7.30) with representation (7.23) for $j = 0$. And by a similar reasoning we will show how to deduce the other bases of m-fold hypergeometric solutions for $j \in [\![0, m-1]\!]$.

Observe that if we have an m-fold hypergeometric sequence a_n starting with a_0 (by shift it is always possible to define the initial term by a_0), then using the representation (7.22) the next term which can be computed from a_0 is a_m, and afterwards a_{2m} ...,$a_{km}, \ldots$. Thus if we set $s_n = a_{mn}$ then all terms computed from s_0 have their indices corresponding to multiples of m for a_n. Moreover, since $\frac{a_{m\cdot(n+1)}}{a_{m\cdot n}} = \frac{s_{n+1}}{s_n} \in \mathbb{K}(n)$, s_n is a hypergeometric term whose general formula is the one of the m-fold hypergeometric term a_{mn}. Therefore we can update (7.30) accordingly so that Algorithm 10 on p. 114 can be applied to compute a basis of all hypergeometric term solution s_n of the updated (7.30) which is nothing else but the basis of all m-fold hypergeometric term solutions of (7.30) with representation (7.23) for $j = 0$.

This view of m-fold hypergeometric terms is our main idea. As explained, a basis of all m-fold hypergeometric term solutions of the m-fold holonomic RE (7.30) can be found by van Hoeij's algorithm provided the following crucial change of variable is done:

$$\begin{cases} m \cdot k = n \\ s_k = a_{m\cdot k} \end{cases} . \tag{7.31}$$

This leads to a 1-fold holonomic RE for s_k which has a hypergeometric term solution because

$$\frac{s_{k+1}}{s_k} = \frac{a_{mk+m}}{a_{mk}} = r(mk). \tag{7.32}$$

The resulting RE is

$$P_d(mk)\cdot s_{k+d} + P_{d-1}(mk)\cdot s_{k+(d-1)} + \cdots + P_1(mk)\cdot s_{k+1} + P_0(mk)\cdot s_k = 0. \tag{7.33}$$

In the general case of $j \in [\![0, m-1]\!]$, the bases of all m-fold hypergeometric term solutions of (7.30) with representation (7.23) are computed by Algorithm 10 after the application of the change of variable

$$\begin{cases} m\cdot k + j = n, \\ s_k = a_{m\cdot k+j} \end{cases} \quad (0 \leqslant j \leqslant m-1). \tag{7.34}$$

This is because in (7.23), the m-fold hypergeometric term indices can always be seen as $m\cdot n + j$, $j \in [\![0, m-1]\!]$.

Let us apply this to an example. We consider the two-term recurrence relation of the Taylor coefficient of $\exp(z)\sin(z)$ which is a 4-fold holonomic RE, so we are going to compute 4-fold hypergeometric term solutions.

```
(%i8) RE:FindRE(exp(z)*sin(z),z,a[n],2);
```

$$(\%o8) \qquad (1+n)\cdot(2+n)\cdot(3+n)\cdot(4+n)\cdot a_{n+4} + 4\cdot a_n = 0 \tag{7.35}$$

```
(%i9) RE:subst(4*n,n,RE);
```

$$(\%o9) \qquad (1+4\cdot n)\cdot(2+4\cdot n)\cdot(3+4\cdot n)\cdot(4+4\cdot n)\cdot a_{4\cdot n+4} + 4\cdot a_{4\cdot n} = 0$$

```
(%i10) RE:subst([a[4*n]=s[n],a[4*n+4]=s[n+1]],RE);
```

$$(\%o10) \qquad (1+4\cdot n)\cdot(2+4\cdot n)\cdot(3+4\cdot n)\cdot(4+4\cdot n)\cdot s_{n+1} + 4\cdot s_n = 0$$

```
(%i11) HypervanHoeij(RE,s[n]);
```

$$(\%o11) \qquad \left\{\frac{(-1)^n\cdot 4^n}{\left(\frac{1}{4}\right)_n\cdot\left(\frac{3}{4}\right)_n\cdot 64^n\cdot(2\cdot n)!}\right\}$$

This set is a basis of all 4-fold hypergeometric term solutions (7.35) for $j=0$ in the representation (7.23). Similarly for the case $j=3$ we get the analogous basis

```
(%i12) RE:FindRE(exp(z)*sin(z),z,a[n],2)$
(%i13) RE:subst(4*n+3,n,RE);
```

$$(\%o13) \qquad (4+4\cdot n)\cdot(5+4\cdot n)\cdot(6+4\cdot n)\cdot(7+4\cdot n)\cdot a_{4\cdot n+7} + 4\cdot a_{4\cdot n+3} = 0$$

```
(%i14) RE:subst([a[4*n+3]=s[n],a[4*n+7]=s[n+1]],RE);
```

$$(\%o14) \qquad (4+4\cdot n)\cdot(5+4\cdot n)\cdot(6+4\cdot n)\cdot(7+4\cdot n)\cdot s_{n+1} + 4\cdot s_n = 0$$

```
(%i15) HypervanHoeij(RE,s[n]);
```

$$(\%o15) \qquad \left\{\frac{(-1)^n\cdot 4^n}{\left(\frac{1}{4}\right)_n\cdot\left(\frac{3}{4}\right)_n\cdot(32\cdot n^3+48\cdot n^2+22\cdot n+3)\cdot 64^n\cdot(2\cdot n)!}\right\}$$

On the other hand, note that the m-fold holonomic RE case is the easiest part for the whole algorithm. Indeed, we know that a given holonomic recurrence equation is not necessarily m-fold holonomic, $m \in \mathbb{N}_{\geqslant 2}$. Such a recurrence equation could have m_1-fold and m_2-fold hypergeometric term solutions with positive integers $m_1 \neq m_2$. Therefore we should define what to do in the more general case.

Observe that without the shift that transforms an m-fold holonomic recurrence equation in the form (7.30), its general representation is given by

$$P_d a_{n+k+md} + P_{d-1} a_{n+k+m(d-1)} + \cdots + P_0 a_{n+k} = 0, \tag{7.36}$$

where $k \in [\![0, m-1]\!]$.

Let us consider the three following 3-fold holonomic REs

$$\begin{aligned} &RE1: \ P_{1,3} \cdot a_{n+7} + P_{1,2} \cdot a_{n+4} + P_{1,1} \cdot a_{n+1} = 0, \\ &RE2: \ P_{2,4} \cdot a_{n+11} + P_{2,3} \cdot a_{n+8} + P_{2,2} \cdot a_{n+5} + P_{2,1} \cdot a_{n+2} = 0, \\ &RE3: \ P_{3,4} \cdot a_{n+13} + P_{3,3} \cdot a_{n+10} + P_{3,2} \cdot a_{n+7} + P_{3,1} \cdot a_{n+4} = 0. \end{aligned} \tag{7.37}$$

- The difference between the order of a summand in $RE1$ and the one of a summand in $RE2$ is always not a multiple of 3. In this case we say that $RE1$ and $RE2$ are 3-fold distinct.
- The difference between the order of a summand in $RE1$ and the one of a summand in $RE3$ is always a multiple of 3. In this case we say that $RE1$ and $RE3$ are 3-fold equivalent

More generally we have the following definitions.

Definition 7.2. *Let $m \in \mathbb{N}$,*

$$RE_1: \ P_{d_1} a_{n+k_1+md_1} + P_{d_1-1} a_{n+k_1+m(d_1-1)} + \cdots + P_{0_1} a_{n+k_1} = 0, \tag{7.38}$$

and

$$RE_2: \ P_{d_2} a_{n+k_2+md_2} + P_{d_2-1} a_{n+k_2+m(d_2-1)} + \cdots + P_{0_2} a_{n+k_2} = 0 \tag{7.39}$$

be two m-fold holonomic recurrence equations.

- *We say that RE_1 and RE_2 are m-fold distinct holonomic equations if $k_2 - k_1$ is not divisible by m.*
- *We say that RE_1 and RE_2 are m-fold equivalent holonomic equations if $k_2 - k_1$ is divisible by m.*

An immediate consequence of these definitions is that linear combinations of m-fold equivalent holonomic REs always give m-fold holonomic recurrence equations whereas linear combinations of m-fold distinct holonomic REs are never m-fold holonomic. For example, let us sum

$RE1$ and $RE3$ from (7.37). This yields the following 3-fold holonomic RE

$$RE1+RE3: \; P_{3,4}\cdot a_{n+13}+P_{3,3}\cdot a_{n+10}+(P_{1,3}+P_{3,2})\cdot a_{n+7}+(P_{1,2}+P_{3,1})\cdot a_{n+4}+P_{1,1}\cdot a_{n+1}=0. \tag{7.40}$$

The whole algorithm is based on the following fundamental theorem from which the general approach to compute m-fold hypergeometric term solutions of any given holonomic recurrence equation is deduced.

Theorem 7.1 (Structure of Holonomic REs Having m-fold Hypergeometric Term Solutions)**.** *Let $m \in \mathbb{N}$, $\mathbb{K}$ a field of characteristic zero, and h_n be an m-fold hypergeometric term which is not u-fold hypergeometric over $\mathbb{K}$ for all positive integers $u < m$. Then h_n is a solution of a given holonomic recurrence equation, say RE, if that equation can be written as a linear combination of m-fold holonomic recurrence equations. When this is the case, h_n is moreover a solution of each m-fold distinct holonomic recurrence equations of that linear combination.*

Proof. Let us assume that h_n is an m-fold hypergeometric term solution of the recurrence equation

$$P_d a_{n+d} + P_{d-1} a_{n+d-1} + \cdots + P_0 a_n = 0, \;\; d > m, \;\; P_d \cdot P_0 \neq 0. \tag{7.41}$$

It suffices to show that for any non-zero term $P_j a_{n+j}$ in (7.41), there exists another summand, say $P_i a_{n+i}$, such that m divides $j-i$. Indeed, this is because when summing m-fold holonomic REs, given that each of them has at least two non-zero polynomial coefficients by definition, therefore we are sure that for each summand appearing on the left-hand side of the sum of these m-fold holonomic REs there must exist another summand whose index differs from the one of that summand by a multiple of m.

We proceed by contradiction. Assume there exists a non-zero term $P_j a_{n+j}$ in (7.41) such that any other summand $P_i a_{n+i}$, $i \neq j$ does not verify that m divides $j-i$. Since h_n is a non-zero solution, we can divide the equation by h_{n+j} and write

$$-P_j = \sum_{\substack{i=0\\ i\neq j}}^{d} P_i \cdot \frac{h_{n+i}}{h_{n+j}}. \tag{7.42}$$

The situation is now two-fold:

- For i verifying $|i-j| < m$, for each corresponding summand $P_i \cdot \frac{h_{n+i}}{h_{n+j}}$ on the right-hand side of (7.42), the fact that m does not divide $j-i$ implies that $\frac{h_{n+i}}{h_{n+j}} \notin \mathbb{K}(n)$ since by assumption h_n is an m-fold hypergeometric term over $\mathbb{K}$ that is not u-fold hypergeometric for all integers $u < m$. Therefore the whole term $P_i \cdot \frac{h_{n+i}}{h_{n+j}} \notin \mathbb{K}(n)$ (see (7.29)).
- For i verifying $|i-j| > m$, for each corresponding summand $P_i \cdot \frac{h_{n+i}}{h_{n+j}}$ on the right-hand side of (7.42), we have two possibilities:
  - either $\frac{h_{n+i}}{h_{n+j}} \notin \mathbb{K}(n)$ and we have the same conclusion as in the previous case;

– or $\frac{h_{n+i}}{h_{n+j}} \in \mathbb{K}(n)$, but in this case since m does not divide $j-i$, this implies that h_n is not an m-fold hypergeometric term and we get a contradiction.

Thus the identity (7.42) is valid only if all the summands on its right-hand side do not belong to $\mathbb{K}(n)$. Therefore (7.42) holds if and only if

$$\sum_{\substack{i=0\\i\neq j}}^{d} P_i \cdot \frac{h_{n+i}}{h_{n+j}} \notin \mathbb{K}(n),$$

however the left-hand side $P_j(n) \in \mathbb{K}[n] \subset \mathbb{K}(n)$. Hence we obtain a contradiction.

Let us now prove the second part of the theorem. Since the multiplication of a holonomic recurrence equation by a polynomial does not affect the computation of its m-fold hypergeometric term solutions, the linear combination of m-fold holonomic REs can always be considered as a sum of m-fold holonomic REs. Therefore it is enough to show that an m-fold hypergeometric term solution of a sum of m-fold holonomic recurrence equations is a solution of each of the involved m-fold distinct holonomic recurrences.

The sum of M m-fold holonomic recurrence equations, $M \in \mathbb{N}$, can be written as

$$\sum_{j=1}^{M} RE_j(a_n) = \sum_{j=1}^{M} \left(P_{d_j} a_{n+k_j+md_j} + P_{d_j-1} a_{n+k_j+m(d_j-1)} + \cdots + P_{0_j} a_{n+k_j}\right) = 0, \tag{7.43}$$

where $k_j \in [\![0, m-1]\!]$, and $P_{d_j} \cdot P_{0_j} \neq 0, j \in [\![1, M]\!]$.

If $M = 1$, then (7.43) is an m-fold holonomic recurrence equation and h_n is an m-fold hypergeometric term solution of it.

We assume now that $M \geqslant 2$ and that there are at least two m-fold distinct holonomic recurrence equations in (7.43). Note that if the M m-fold holonomic REs are m-fold equivalent then the situation is similar to the case $M = 1$ since every linear combination of m-fold equivalent holonomic RE is an m-fold holonomic RE.

Now suppose that h_n is not solution of RE_{j_1} in (7.43), $j_1 \in [\![1, M]\!]$, then given that $\sum_{j=1}^{M} RE_j(h_n) = 0$, there must be at least one second m-fold holonomic recurrence equation RE_{j_2}, $j_2 \in [\![1, M]\!]$, m-fold distinct with RE_{j_1} such that $RE_{j_2}(h_n) \neq 0$. Without loss of generality, we consider that RE_{j_2} is the only second m-fold holonomic RE with these properties. Of course, if $RE_{j_1}(h_n) \neq 0$ and $RE_{j_1}(h_n) + RE_{j_2}(h_n) = 0$ then $RE_{j_2}(h_n) \neq 0$. Thus, we have

$$\begin{cases} RE_{j_1}(h_n) \neq 0 \\ RE_{j_2}(h_n) \neq 0 \\ RE_{j_1}(h_n) + RE_{j_2}(h_n) = 0 \end{cases}. \tag{7.44}$$

The fact that the m-fold holonomic recurrence equations RE_{j_1} and RE_{j_2} are m-fold distinct implies that $k_{j_1} - k_{j_2}$ is not a multiple of m.

Using (7.44), after substitution of h_n in the sum of the equations and division by $h_{n+k_{j_1}+d_{j_1}m}$, we deduce that

$$-P_{d_{j_1}} = \sum_{e_{j_1}=0_{j_1}}^{d_{j_1}-1} P_{e_{j_1}} \frac{h_{n+k_{j_1}+e_{j_1}m}}{h_{n+k_{j_1}+d_{j_1}m}} + \sum_{e_{j_2}=0_{j_2}}^{d_{j_2}} P_{e_{j_2}} \frac{h_{n+k_{j_2}+e_{j_2}m}}{h_{n+k_{j_1}+d_{j_1}m}} = S_{j_1} + S_{j_1,j_2}, \tag{7.45}$$

which is equivalent to

$$-P_{d_{j_1}} - S_{j_1} = S_{j_1,j_2}. \tag{7.46}$$

All the summands of S_{j_1} belong to $\mathbb{K}(n)$ since h_n is an m-fold hypergeometric sequence and the corresponding index differences

$$n + k_{j_1} + e_{j_1}m - (n + k_{j_1} + d_{j_1}m) = m \cdot (e_{j_1} - d_{j_1})$$

are multiples of m. However, for S_{j_1,j_2} the index differences

$$n + k_{j_2} + e_{j_2}m - (n + k_{j_1} + d_{j_1}m) = k_{j_2} - k_{j_1} + m \cdot (e_{j_2} - d_{j_1})$$

are not multiples of m. Therefore by the same argument used in the first part of the proof we deduce that $S_{j_1,j_2} \notin \mathbb{K}(n)$. Thus (7.46) holds if and only if $-P_{d_{j_1}} - S_{j_1} \in \mathbb{K}(n)$ and $S_{j_1,j_2} \notin \mathbb{K}(n)$. Therefore we get a contradiction. □

From this theorem, given $m \in \mathbb{N}$, we are now sure to compute a basis of all m-fold hypergeometric term solutions of a given holonomic recurrence equation by splitting it into the sum of m-fold distinct holonomic recurrence equations and use Algorithm 10 to solve these holonomic REs provided the change of variable (7.34). Note that since we compute m-fold hypergeometric terms as element of a basis of all m-fold hypergeometric term solutions of holonomic REs, an m-fold hypergeometric term is solution of two given holonomic REs if it is linearly dependent to an element of the basis of all m-fold hypergeometric term solutions of each of these holonomic RE. Therefore, the solutions sought are built by all the linearly dependent m-fold hypergeometric term solutions of each involved m-fold distinct holonomic RE. Note that the computation of m-fold hypergeometric term solutions with representation (7.23) for $j = 0$ of each m-fold holonomic RE

$$P_{d_i} a_{n+k_i+md_i} + P_{d_i-1} a_{n+k_i+m(d_i-1)} + \cdots + P_{0_i} a_{n+k_i} = 0, \tag{7.47}$$

is done after writing it in the form (7.30). Thus (7.47) is transformed as

$$P_{d_i}(n - k_i) a_{n+md_i} + P_{d_i-1}(n - k_i) a_{n+m(d_i-1)} + \cdots + P_{0_i}(n - k_i) a_n = 0. \tag{7.48}$$

Let us take as an example the holonomic RE satisfied by the Taylor coefficients of $\exp(z) + \cos(z)$.

```
(%i16) FindRE(cos(z)+exp(z),z,a[n]);
```

(%o16) $(1 + n) \cdot (2 + n) \cdot (3 + n) \cdot a_{n+3} - (1 + n) \cdot (2 + n) \cdot a_{n+2} + (1 + n) \cdot a_{n+1} - a_n = 0$

This is a linear combination of two 2-fold distinct holonomic REs, namely

$$RE1: \ (1 + n) \cdot (2 + n) \cdot (3 + n) \cdot a_{n+3} + (1 + n) \cdot a_{n+1} = 0,$$

and

$$RE2: \ (-1 - n) \cdot (2 + n) \cdot a_{n+2} - a_n = 0.$$

Only $RE1$ has to be transformed as its trailing term is not of order 0. This yields

$$RE11: \; n \cdot (1+n) \cdot (2+n) \cdot a_{n+2} + n \cdot a_n = 0.$$

From this one easily sees that the given holonomic RE has 2-fold hypergeometric term solutions since we get two two-term recurrence relations that are linearly dependent:

$$-n \cdot RE2 = RE11.$$

Remember that there is no need to use all the m changes of variable of (7.34) because as we explained earlier, once one succeeds in computing a basis of m-fold hypergeometric term solutions corresponding to the representation (7.23) for a fixed $j \in [\![0, m-1]\!]$, the other ones can be computed in a similar way. This will be used for power series computations in order to consider all possible linear combinations of hypergeometric type series of type m.

This result is a consequence of observing m-fold hypergeometric terms as sequences whose indices are taken in $m \cdot \mathbb{Z} + j$, $j \in [\![0, m-1]\!]$. Commonly this notion is used according to its definition for the set of integers $\mathbb{Z}$ which is generally the chosen set of indices. In this case two terms of a sequence are said to be consecutive if their index difference is 1 or -1. Such a definition is more useful for hypergeometric terms since it allows Algorithm 10 to look for hypergeometric term solutions of holonomic REs in such a way that the ratio of two consecutive terms is a rational function over the considered field. For the m-fold case, however, one rather needs to consider $m\mathbb{Z}$ as the set of indices so that the computation of m-fold hypergeometric term ($m \geqslant 2$) solutions of holonomic REs is done analogically to the one of hypergeometric terms. In this situation one could say that two terms of an m-fold sequence are consecutive if the difference of their indices is m or $-m$. The other sequences with indices $m \cdot \mathbb{Z} + j$, $j \in [\![0, m-1]\!]$ can be seen as other representations of the same family of sequences.

To compute the basis of all m-fold hypergeometric term solutions of a given holonomic RE, the algorithm proceeds by iteration up to the order of the RE. Nevertheless more often the number of cases to be considered is much smaller than the order of the given RE. For example, let us use the holonomic RE (6.75) on p. 121 which does not have hypergeometric term solutions over $\mathbb{C}$ as we saw at the end of the previous chapter.

```
(%i17) RE:FindRE(sin(z^3)^3,z,a[n]);
```

(%o17) $$(n-8)\cdot(n-5)\cdot(n-2)\cdot(1+n)\cdot a_{n+1}+90\cdot(n-8)\cdot(n-5)\cdot a_{n-5}+729\cdot a_{n-11}=0$$

Observe that the obtained recurrence equation is a 2-fold, 3-fold and 6-fold holonomic RE of order 12. It is straightforward to see that all the other cases do not lead to a solution since the recurrence equation cannot be written as a sum of their type of m-fold holonomic REs. Indeed, in the other cases it occurs holonomic REs that are not m-fold since they only have one non-zero polynomial coefficient. That is why in this example the algorithm only does computations for m-fold hypergeometric term solutions having $m \in \{1, 2, 3, 6\}$.

Our algorithm to compute m-fold hypergeometric term solutions of a given holonomic RE, called *mfoldHyper*, is given as follows.

Algorithm 11 mfoldHyper: m-fold hypergeometric term solutions of holonomic recurrence equation of order $d \in \mathbb{N}$

Input: A holonomic recurrence equation

$$P_d a_{n+d} + P_{d-1} a_{n+d-1} + \cdots + P_0 a_n = 0, \quad d > m, \ P_d \cdot P_0 \neq 0 \tag{7.49}$$

Output: A basis of all m-fold hypergeometric term solutions of (7.49).

1. Set $H = \{\}$.
2. Use Algorithm 10 to find the basis, say H_1, of all hypergeometric term solutions of (7.49). If $H_1 \neq \emptyset$, then add $[1, H_1]$ to H.
3. For $2 \leqslant m \leqslant d$ do:
3. (a) Extract the following m holonomic recurrence equations from (7.49) and construct the system

$$\begin{cases} P_0(n) \cdot a_n + P_m(n) \cdot a_{n+m} + \cdots + P_{m \cdot \lfloor \frac{d}{m} \rfloor}(n) \cdot a_{n+m \cdot \lfloor \frac{d}{m} \rfloor} = 0 \\ P_1(n) \cdot a_{n+1} + P_{m+1}(n) \cdot a_{n+m+1} + \cdots + P_{m \cdot \lfloor \frac{d}{m} \rfloor + 1}(n) \cdot a_{n+m \cdot \lfloor \frac{d}{m} \rfloor + 1} = 0 \\ \cdots \\ P_{m-1}(n) \cdot a_{n+m-1} + P_{2m-1}(n) \cdot a_{n+2m-1} + \cdots + P_{m \cdot \lfloor \frac{d}{m} \rfloor + m-1}(n) \cdot a_{n+m \cdot \lfloor \frac{d}{m} \rfloor + m-1} = 0 \\ \end{cases}, \tag{7.50}$$

   assuming $P_j(n) = 0$ for $j > d$.

   (b) If there exist a holonomic RE with only one non-zero polynomial coefficient in (7.50), then stop and go back to step 3.(a) for $m + 1$.

   (c) Shift all the m-fold holonomic recurrence equations in (7.50) so that the order of the trailing term equals 0.

   (d) Apply the change of variable (7.31) for each m-fold holonomic recurrence equation.

   (e) Compute a basis of all hypergeometric term solutions s_k as defined in (7.31) for (7.33) of each resulting holonomic recurrence equation with Algorithm 10.

   (f) Construct the set H_m of hypergeometric terms which are each linearly dependent to one term in each of the m computed bases in step 3-(d).

   (g) If $H_m \neq \emptyset$ then add $[m, H_m]$ in H.
4. Return H.

Remark. Note that for step 3.(f) none of the computed bases in step 3.(f) should be empty. When this is the case, two m-fold hypergeometric terms are recognized to be linearly dependent if the characteristic ratio of one is an integer shift of the other (the shift might be zero).

We implemented *mfoldHyper* in Maxima as *mfoldHyper(RE,a[n],[m,j])*, by default *[m,j]* is an empty sequence. And in that case each list of m-fold hypergeometric term solutions, say $[m, [h_{1,m}, h_{2,m}, \ldots]]$, contains "simple" formulas of hypergeometric terms corresponding to $j = 0$ in (7.23). Once we know that there are some m-fold hypergeometric term solutions for particular $m \in \mathbb{N}$, the algorithm can be called as *mfoldHyper(RE,a[n],m,j)* for $0 \leqslant j < m$ to get the solutions in their other representations (the case $m \cdot n + j$ in (7.23) for j not necessarily 0).

Let us now apply the algorithm to the examples of the previous section.

```
(%i18) RE:FindRE(atan(z)+exp(z),z,a[n])$
(%i19) mfoldHyper(RE,a[n]);
```

$$(\%o19)\qquad \left[\left[1,\left\{\frac{1}{n!}\right\}\right],\left[2,\left\{\frac{(-1)^n}{n}\right\}\right]\right]$$

As expected, the needed "simple" formulas are computed without the need of extension fields of $\mathbb{Q}$. However, for the next example one needs to allow algebraic extensions for the general coefficient of $\log(1+z+z^2)$.

For computation over algebraic extension fields of $\mathbb{Q}$, the syntax is *mfoldHyper(RE,a[n],[K])* for the two possible value *K=C* or *K=Q* (default value). To ask for specific m-fold hypergeometric term solutions the syntax is *mfoldHyper(RE,a[n],[K,m,j])*.

```
(%i20) RE:FindRE(log(1+z+z^2)+cos(z),z,a[n])$
(%i21) mfoldHyper(RE,a[n],C);
```

$$(\%o21)\qquad \left[\left[1,\left\{\frac{\left(\frac{-1-\sqrt{3}\cdot i}{2}\right)^n}{n},\frac{\left(\frac{\sqrt{3}\cdot i-1}{2}\right)^n}{n},\frac{(-i)^n}{n!},\frac{(-1)^{\frac{n}{2}}}{n!}\right\}\right],\left[2,\left\{\frac{(-1)^n}{(2\cdot n)!}\right\}\right]\right]$$

The obtained 2-fold hypergeometric term solution is the one used for the hypergeometric type series related to $\cos(z)$ instead of the corresponding hypergeometric term over $\mathbb{C}$ as Maple's FPS command does.

```
(%i22) RE:FindRE(tan(5*atan(z)),z,a[n])$
(%i23) mfoldHyper(RE,a[n],C);
```

$$(\%o23)\qquad \left[\left[1,\left\{\left(5-2\cdot\sqrt{5}\right)^{\frac{n}{2}},\left(5+2\cdot\sqrt{5}\right)^{\frac{n}{2}},\left(-\sqrt{2\cdot\sqrt{5}+5}\right)^n\right\}\right],\right.$$
$$\left.\left[2,\left\{\left(5-2\cdot\sqrt{5}\right)^n,\left(5+2\cdot\sqrt{5}\right)^n\right\}\right]\right]$$

Similarly for this example the 2-fold hypergeometric term is the one used for the power series representation.

```
(%i24) RE:FindRE(1/cos(4*acos(z)),z,a[n])$
(%i25) mfoldHyper(RE,a[n],C);
```

$$(\%o25)\qquad \left[\left[1,\left\{\left(4-2^{\frac{3}{2}}\right)^{\frac{n}{2}},\left(4+2^{\frac{3}{2}}\right)^{\frac{n}{2}},\left(-\sqrt{2^{\frac{3}{2}}+4}\right)^n\right\}\right],\left[2,\left\{\left(4-2^{\frac{3}{2}}\right)^n,\left(4+2^{\frac{3}{2}}\right)^n\right\}\right]\right]$$

```
(%i26) declare(q1,constant)$
(%i27) declare(q2,constant)$
(%i28) RE:FindRE(1/((q1-z^2)*(q2-z^3)),z,a[n])$
```

```
(%i29) mfoldHyper(RE,a[n],C);
```

$$(\%o29)\quad \left[\left[1,\left\{\left(-\sqrt{\frac{1}{q1}}\right)^n,\left(\frac{1}{q1}\right)^{\frac{n}{2}},\left(\frac{\sqrt{3}\cdot i\cdot\left(\frac{1}{q2}\right)^{\frac{1}{3}}-\left(\frac{1}{q2}\right)^{\frac{1}{3}}}{2}\right)^n,\left(\frac{1}{q2}\right)^{\frac{n}{3}}\right\}\right],\left[2,\left\{\left(\frac{1}{q1}\right)^n\right\}\right],\left[3,\left\{\left(\frac{1}{q2}\right)^n\right\}\right]\right]$$

We would like to point out that the m-fold hypergeometric terms computed, $m \geqslant 2$, in the three latter examples fit well with the computations of Taylor expansions of the given expressions. Indeed, generally if an expression represents a hypergeometric type function of type m then the indeterminate in its Taylor expansion has only powers of the form $m\cdot n+j, j\in [\![0,m-1]\!]$. This fact is not easily taken into account when we only consider hypergeometric terms. This could explain why Maple's FPS command could not use the hypergeometric term solutions of the three previous examples to compute the requested power series.

Let us now compute the m-fold hypergeometric term solutions for the examples that we have shown to be out of Maple's current FPS command capabilities.

```
(%i30) RE:FindRE(exp(z^2)+cos(z^2),z,a[n])$
(%i31) mfoldHyper(RE,a[n]);
```

$$(\%o31)\quad \left[\left[2,\left\{\frac{1}{n!}\right\}\right],\left[4,\left\{\frac{(-1)^n}{(2\cdot n)!}\right\}\right]\right]$$

```
(%i32) RE:FindRE(cosh(z^3)+sin(z^2),z,a[n])$
(%i33) mfoldHyper(RE,a[n]);
```

$$(\%o33)\quad \left[\left[3,\left\{\frac{1}{n!},\frac{(-1)^n}{n!}\right\}\right],\left[4,\left\{\frac{(-1)^n}{(2\cdot n)!}\right\}\right],\left[6,\left\{\frac{1}{(2\cdot n)!}\right\}\right]\right]$$

```
(%i34) RE:FindRE(asin(z^2)^2+acos(z),z,a[n])$
(%i35) mfoldHyper(RE,a[n]);
```

$$(\%o35)\quad \left[\left[2,\left\{\frac{4^n\cdot n!^2}{n^2\cdot(2\cdot n)!}\right\}\right],\left[4,\left\{\frac{4^n\cdot n!^2}{n^2\cdot(2\cdot n)!}\right\}\right]\right]$$

```
(%i36) RE:FindRE(sqrt(sqrt(8*z^3+1)-1)+sqrt(7+13*z^4),z,a[n])$
(%i37) mfoldHyper(RE,a[n]);
```

$$(\%o37)\quad \left[\left[3,\left\{\frac{\left(\frac{1}{4}\right)_n\cdot\left(\frac{3}{4}\right)_n\cdot(-8)^n\cdot 4^n}{(4\cdot n-1)\cdot(2\cdot n)!}\right\}\right],\left[4,\left\{\frac{4^{-4-n}\cdot(-13)^n\cdot(2\cdot n)!}{(2\cdot n-1)\cdot 7^n\cdot n!^2}\right\}\right]\right]$$

```
(%i38) RE:FindRE(exp(asin(z))+exp(asinh(z)),z,a[n])$
(%i39) mfoldHyper(RE,a[n],C);
```

$$(\%o39)\quad \left[\left[2,\left\{\frac{\left(-\frac{i-2}{2}\right)_n\cdot\left(\frac{2+i}{2}\right)_n\cdot 4^{3+n}}{(4\cdot n^2+1)\cdot(2\cdot n)!},\frac{2\cdot 4^{1-n}\cdot(-1)^n\cdot(2\cdot n)!}{(2\cdot n-1)\cdot n!^2}\right\}\right]\right]$$

```
(%i40) RE:FindRE(sin(8*asinh(z))+sqrt(1+z^4),z,a[n])$
```

```
(%i41) mfoldHyper(RE,a[n],C);
```

$$(\%o41) \quad \left[\left[2, \left\{\frac{(1-4\cdot i)_n \cdot (4\cdot i+1)_n \cdot (-1)^n \cdot 4^{3+n}}{(n^2+16)\cdot(2\cdot n)!}\right\}\right], \left[4, \left\{\frac{98\cdot 4^{4-n}\cdot(-1)^n\cdot(2\cdot n)!}{(2\cdot n-1)\cdot n!^2}\right\}\right]\right]$$

For (6.75) we get

```
(%i42) RE:FindRE(sin(z^3)^3,z,a[n])$
(%i43) mfoldHyper(RE,a[n]);
```

$$(\%o43) \quad \left[\left[6, \left\{\frac{(-9)^n}{(2\cdot n)!}, \frac{(-1)^n}{(2\cdot n)!}\right\}\right]\right]$$

One can therefore see the improvement given by the use of *mfoldHyper*.

Let us now use our implementation for the computation of a specific m-fold hypergeometric term solutions. In this case the user has to give a value for m and j with $j \in [\![0, m-1]\!]$.

```
(%i44) RE:FindRE(asin(z)^2+log(1+z^5),z,a[n])
```

$$(\%o71) \quad -2\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}+2\cdot n\cdot(1+n)\cdot(2+n)\cdot a_{n+2}+2\cdot(1+n)\cdot(1+3\cdot n)\cdot a_{n+1}-2\cdot n\cdot\left(48-57\cdot n+10\cdot n^2\right)\cdot a_n+2\cdot(n-1)\cdot\left(115-77\cdot n+10\cdot n^2\right)\cdot a_{n-1}+(n-2)\cdot\left(399-204\cdot n+19\cdot n^2\right)\cdot a_{n-2}-(n-3)\cdot\left(622-242\cdot n+19\cdot n^2\right)\cdot a_{n-3}-(n-4)\cdot(3\cdot n-79)\cdot a_{n-4}-(n-5)\cdot\left(-368-18\cdot n+15\cdot n^2\right)\cdot a_{n-5}+(n-6)\cdot\left(-335-48\cdot n+15\cdot n^2\right)\cdot a_{n-6}+(n-7)\cdot\left(115-150\cdot n+19\cdot n^2\right)\cdot a_{n-7}-(n-8)\cdot\left(284-188\cdot n+19\cdot n^2\right)\cdot a_{n-8}-(n-9)\cdot(9\cdot n-77)\cdot a_{n-9}+(n-10)\cdot\left(464-96\cdot n+5\cdot n^2\right)\cdot a_{n-10}-(n-11)\cdot\left(565-106\cdot n+5\cdot n^2\right)\cdot a_{n-11}-2\cdot(n-12)^3\cdot a_{n-12}+2\cdot(n-13)^3\cdot a_{n-13}=0$$

```
(%i45) mfoldHyper(RE,a[n],5,0);
```

$$(\%o45) \quad \left\{\frac{(-1)^n}{2\cdot n}\right\}$$

```
(%i46) mfoldHyper(RE,a[n],5,3);
```

$$(\%o46) \quad \left\{\frac{(-1)^n}{2\cdot(5\cdot n+3)}\right\}$$

```
(%i47) mfoldHyper(RE,a[n],2,1);
```

$$(\%o47) \quad \left\{\frac{(2\cdot n)!}{(2\cdot n+1)\cdot 4^n\cdot n!^2}\right\}$$

On the other hand, we mentioned earlier that *exp-like* and *rational* functions are hypergeometric type functions, i.e their power series expansions can always be written as linear combinations of power series having m-fold hypergeometric terms as coefficients. As we intend to use our algorithm to compute a much larger family of power series than the currently used algorithms for that, we shall ensure that mfoldHyper finds the coefficients of *exp-like* and *rational* functions since they have special algorithms described in [Koepf, 1992]. This is shown as a consequence of the next two theorems.

Theorem 7.2. *Let $\mathbb{K}$ be an algebraically closed field. The power series coefficients of every rational function in $\mathbb{K}(z)$ are linear combinations of hypergeometric terms over $\mathbb{K}$.*

Proof. Observe that the power series expansions of every rational functions $\frac{P(z)}{Q(z)} \in \mathbb{K}(z)$, $Q(z) \neq 0$ can be found by computing the one of $\frac{1}{Q(z)}$ and multiply the result by $P(z)$. Then one obtains a linear combination of power series, whose coefficients are the ones of the polynomial $P(z)$, and the general powers of the indeterminate of these power series are shifted according to the degrees of the monomials in $P(z)$. Therefore the general coefficients of rational functions in $\mathbb{K}(z)$ are essentially resulting from the computations of the power series general coefficients of the reciprocal of their denominators. Thus we consider $\frac{1}{Q(z)}$, $Q(z) \in \mathbb{K}[z] \setminus \{0\}$ for the following part of the proof.

Since $\mathbb{K}$ is algebraically closed, there exist unique $z_k \in \mathbb{K} \setminus \{0\}, e_k \in \mathbb{N}, k = 1, \ldots, q$, $e_0 \in \mathbb{N}_{\geqslant 0}(z_0 = 0)$ such that

$$\frac{1}{Q(z)} = \frac{1}{z^{e_0} \cdot (z - z_1)^{e_1} \cdot (z - z_2)^{e_2} \cdots (z - z_q)^{e_q}}. \tag{7.51}$$

By partial fraction decomposition this leads to

$$\begin{aligned} \frac{1}{Q(z)} &= \sum_{k=0}^{q} \left(\frac{c_{k,1}}{(z - z_k)} + \frac{c_{k,2}}{(z - z_k)^2} + \cdots + \frac{c_{k,e_k}}{(z - z_k)^{e_k}} \right) && (7.52)\\ &= \sum_{k=0}^{q} \sum_{i=1}^{e_k} \frac{c_{k,i}}{(z - z_k)^i} \\ &= \sum_{i=1}^{e_0} \frac{c_{0,i}}{z^i} + \sum_{k=1}^{q} \sum_{i=1}^{e_k} \frac{c_{k,i}(-1)^i}{z_k^i \left(1 - \frac{z}{z_k}\right)^i}, && (7.53) \end{aligned}$$

for some constants $c_{k,i} \in \mathbb{K}, i = 0, \ldots, e_k, k = 0, \ldots, q$.

We are interested in the power series coefficients of the summands in (7.53) that have $z_k \neq 0$ since the other terms constitute the principal part of the corresponding Laurent expansion. Using (2.6) from p. 15 for the variable $\frac{z}{z_k}$ we get

$$\frac{c_{k,i}(-1)^i}{z_k^i \left(1 - \frac{z}{z_k}\right)^i} = \sum_{n=0}^{\infty} \frac{c_{k,i}(-1)^i}{z_k^{i+n}} \binom{n+i-1}{i-1} z^n. \tag{7.54}$$

Therefore the power series of $\frac{1}{Q(z)}$ has the general coefficient

$$\sum_{k=1}^{q} \sum_{i=1}^{e_k} \frac{c_{k,i}(-1)^i}{z_k^{i+n}} \binom{n+i-1}{i-1}, \tag{7.55}$$

which is a linear combination of the hypergeometric terms $\frac{1}{z_k^{i+n}}\binom{n+i-1}{i-1}, i = 1,\ldots,e_k, k = 1,\ldots q,$ whose ratios are

$$\frac{i+n}{z_k\cdot(n+1)} \in \mathbb{K}(n). \tag{7.56}$$

□

This theorem shows that with the availability of an algorithm for complete factorization over $\mathbb{C}$, the Petkovšek and van Hoeij algorithms can be used to find the coefficients of all rational functions in $\mathbb{C}(z)$ since holonomic recurrence equations satisfied by their coefficients can be computed using *FindRE* (see [Koepf, 1992]). Unfortunately the missing of such an algorithmic factorization limits the Petkovšek and van Hoeij algorithms in many cases as we have seen in Section 7.2.

However, the nice advantage of using mfoldHyper is that it does not only recover the hypergeometric terms that Petkovšek's and van Hoeij's algorithms can compute, but moreover mfoldHyper does not necessarily need a full factorization. This explains why for the example (7.20) our Maxima FPS algorithm gives a much simpler result than Maple's FPS because mfoldHyper looks for m-fold hypergeometric terms that fit to all the possible factorizations that can be handled by the computer algebra system used.

Similarly, for *exp-like* function we have the following theorem.

Theorem 7.3. *Let $\mathbb{K}$ be an algebraically closed field. The power series coefficients of every exp-like function over $\mathbb{K}$ are linear combinations of hypergeometric terms over $\mathbb{K}$.*

Proof. From [Koepf, 1992, Algorithm 7.1] we know that the coefficients of the power series expansions of *exp-like* functions over $\mathbb{K}$ have the form

$$a_n := \frac{1}{n!}\sum_{k=1}^{q}\sum_{i=1}^{e_k} c_{k,i}n^{i-1}\lambda_k^n, \tag{7.57}$$

$c_{k,i}, \lambda_k \in \mathbb{K}, i = 1,\ldots,e_k, k = 1,\ldots,q$. The computation of the constants λ_k (pairwise different) with multiplicities e_k is ensured by the algebraic closure of $\mathbb{K}$ (see [Walter, 1985]).

Obviously, (7.57) is a linear combination of the hypergeometric terms $\frac{n^{i-1}\lambda_k^n}{n!}, i = 1,\ldots,e_k,$ $k = 1,\ldots,q$, whose ratios are

$$\frac{(n+1)^{i-2}\lambda_k}{n^{i-1}} \in \mathbb{K}(n) \tag{7.58}$$

□

As a consequence of this theorem, mfoldHyper can compute the coefficients of *exp-like* functions without necessarily looking for the exact λ_k in (7.57) but rather their pairwise different powers λ_k^j, $j \in \mathbb{N}_{\geqslant 0}$.

Furthermore, whereas in [Koepf, 1992] a necessary condition to compute power series of *exp-like* functions is to have differential equations with constant coefficients, mfoldHyper rather finds the corresponding power series coefficients without the satisfaction of this condition. An example of this type is $(1+z)\exp(z)$ which satisfies the constant coefficient DE $F''-2F'+F=0$ but *HolonomicDE* computes the first order DE $(1+z)F'-(2+z)F=0$. Our Maxima FPS yields

```
(%i48) FPS((1+z)*exp(z),z,n);
```

$$(\%o48) \qquad \sum_{n=0}^{\infty} \frac{(1+n)\cdot z^n}{n!}$$

Note that compared to the rational function case where the algorithm in [Koepf, 1992] sometimes gives coefficients that cannot be found by Petkovšek's and van Hoeij's algorithms, in the *exp-like* case the use of these algorithms rather leads to all the results that can be computed using [Koepf, 1992, Algorithm 5.1]. Indeed, these *exp-like* functions are linear combinations of $\exp(az)\cos(bz)$, $\exp(az)\sin(bz)$, and $\exp(az)$, for constants a, b, which all have power series with hypergeometric term coefficients. And of course, mfoldHyper covers all these cases as expected.

Eventually, note that the existence of m-fold hypergeometric term solutions of a holonomic recurrence equation satisfied by the Taylor coefficients of a given expression does not necessarily guarantee that this expression represents a hypergeometric type function. For example, $\sin(\sqrt{z+z^3})$ has a Taylor expansion with fractional powers as one can see below.

```
(%i49) taylor(sin(sqrt(z+z^3)),z,0,5);
```

$$(\%o49)/T/\sqrt{z}-\frac{z^{\frac{3}{2}}}{6}+\frac{61\cdot z^{\frac{5}{2}}}{120}-\frac{1261\cdot z^{\frac{7}{2}}}{5040}-\frac{37799\cdot z^{\frac{9}{2}}}{362880}+\dots$$

Using *FindRE*, we get the holonomic RE

```
(%i50) RE:FindRE(sin(sqrt(z+z^3)),z,a[n]);
```

$$(\%o50) \qquad 2\cdot(1+n)\cdot(1+2\cdot n)\cdot a_{n+1}+a_n+4\cdot(n-1)\cdot(4\cdot n-11)\cdot a_{n-1}+9\cdot a_{n-2} \\ +6\cdot(n-3)\cdot(2\cdot n-9)\cdot a_{n-3}+27\cdot a_{n-4}+27\cdot a_{n-6}=0$$

which has the following 2-fold hypergeometric term solution.

```
(%i51) mfoldHyper(RE,a[n]);
```

$$(\%o51) \qquad \left[\left[2,\left\{\frac{\left(\frac{1}{4}\right)_n\cdot(-1)^n}{\left(\frac{3}{4}\right)_n\cdot(4\cdot n-3)}\right\}\right]\right]$$

The interpretation of such results for power series representations remains to be explored. Nevertheless, having described an algorithm to compute a basis of all m-fold hypergeometric term solutions of holonomic recurrence equations, a very wide possibility of Laurent-Puiseux series representations is accessible. Let us move to the chapter dedicated to achieving this goal.

Chapter 8

Computing Power Series

"The ultimate goal of mathematics is to eliminate any need for intelligent thought."
Alfred N. Whitehead

According to the general algorithm described in [Koepf, 1992], we are now at the final step of the power series computation procedure. Let us first recall what these steps are. For a given expression f, we compute its power series in the following way:

1. Find a holonomic differential equation for f using our variant of Algorithm 1 described in Section 4.1.2, on p. 43;

2. Convert that holonomic DE into a holonomic recurrence equation satisfied by the Taylor coefficients of f using Algorithm 2, p. 53;

3. Solve the obtained holonomic RE which in our case reduces to compute a basis of all the m-fold hypergeometric term solutions of that RE using Algorithm 11, p. 147;

4. If there are solutions, use initial values to find the linear combination of the resulting hypergeometric type power series that corresponds to the power series expansion of f, if such a linear combination is valid.

To take into account the Puiseux series case, we will consider an intermediate step between the second and the third step above to determine the involved Puiseux number, say k; then use the substitution $h(z) = f(z^k)$ to bring the situation to the Laurent series one, and finally divide the general power of the indeterminate z in the obtained power series representation of $h(z)$ by k to get the expansion sought. This was first developed in [Gruntz and Koepf, 1995, Section 5] for the two-term holonomic recurrence relation case.

Let $\mathbb{K}$ be a field of characteristic zero. For the calculations in the last step to work we will need to find starting points at which initial values can be computed from all the involved m-fold hypergeometric terms. This will allow us to extract parts of series expansions that cannot easily be obtained (they can be represented by a disturbing Laurent polynomial or a term which is not differentiable at the point of development like a logarithmic term) using corresponding m-fold hypergeometric terms like for the case of $\text{arcsech}(z)$ (see Example 4.3.2 on p. 60) or generally the sum of sums of polynomials times $\log(z)$ powers and hypergeometric type functions. Indeed, our goal is to compute a representation of the form

$$f(z) = T(z) + F(z), \tag{8.1}$$

where $T(z) \in \mathbb{K}[\log(z)][z, \frac{1}{z}]$ is a Laurent polynomial in the variable z with coefficients in $\mathbb{K}[\log(z)]$, and $F(z)$ is a linear combination of hypergeometric type series. We mention that $T(z)$ is not uniquely determined but its determination will be made more precise by Lemma 8.1 and Algorithm 12. We will call the power series representation of such functions generalized hypergeometric type series.

Thus with the four steps mentioned above we will be able to compute a very wide family of linear combinations of hypergeometric type series in an efficient way since we extended van Hoeij's algorithm [Van Hoeij, 1999] to the m-fold hypergeometric case. Actually, whereas the Maple FPS implementation is clearly *not* linear, we have now developed a linear algorithm which detects *every* linear combination of m-fold hypergeometric terms.

On the other hand, we will also extend our algorithm for the computation of some asymptotic series. As presented in (2.42), on p. 2.42, this computation needs a good implementation of an algorithm to compute limits, in particular those of exp-log functions (see [Gruntz and Gonnet, 1992]). Indeed, the current state of the art algorithm in limit computation was developed by Dominik Gruntz in his Ph.D. thesis [Gruntz, 1996] with implementation in Maple supervised by Gaston H. Gonnet, and the described algorithm gave birth to important results for computing asymptotic expansions as presented in ([Gruntz, 1993], [Richardson et al., 1996]). Of course, using our result of the previous chapter these computations can be extended to asymptotic series as was shown in [Koepf, 1993] for the two-term recurrence relation case. Despite the fact that Maxima's commands *taylor* and *limit* are limited in computing asymptotic expansions and limits, using the formula of asymptotic coefficients (2.42) on p. 20 we will show how our extension recovers all the examples given on page 20.

We will end this thesis by considering some non-hypergeometric situations, namely, holonomic expressions that are not of hypergeometric type and non-holonomic expressions. For the first case corresponding to holonomic expressions, we will use well known formulas for hypergeometric type series having only a single summation term. These are the Cauchy product rule for products of power series, the reciprocal to compute the reciprocal of a power series, the rational powers formula to compute a rational power of a power series. When these formulas are not manageable, we give a recursive definition of the Taylor coefficients using the holonomic recurrence equation computed by Algorithm 2. For the non-holonomic case, we will use an extended version of Algorithm 1 to compute a quadratic differential equation for the given expression and, if successful, by the use of the Cauchy product rule deduce a non-linear recurrence equation for the corresponding Taylor coefficients from which we give a recursive representation of the power series sought.

8.1 Finding the Puiseux Number

Let us assume that we are looking for a representation of the form

$$f(z) = T(z) + F(z) := T(z) + \sum_{i=1}^{I} \sum_{n=0}^{\infty} s_{i_n} z^{(m_i \cdot n + j_i)/k_i} \tag{8.2}$$

where $m_i, k_i \in \mathbb{N}$, $j_i \in [\![0, m_i - 1]\!]$, s_{i_n} is an m_i-fold hypergeometric term corresponding to $j = j_i$ in the representation (7.23), on p. 137, and $T(z)$ is an extra term whose computation will be explained in the section. In this section we also show how the determination of the positive integers k_i is done.

Without loss of generality, we suppose that $F(z)$ in (8.2) is the sum of two hypergeometric type series of type m_1 and m_2. We have

$$F(z) := \sum_{n=0}^{\infty} s_{1_n} z^{(m_1 \cdot n + j_1)/k_1} + \sum_{n=0}^{\infty} s_{2_n} z^{(m_2 \cdot n + j_2)/k_2}, \tag{8.3}$$

with the same definitions in (8.2) for $I = 2$. For simplicity, we also assume that k_1 and k_2 are co-prime[1]. This is to avoid the use of more variables since in particular this assumption implies that the least common multiple of k_1 and k_2 is $\operatorname{lcm}(k_1, k_2) = k_1 \cdot k_2$. Substituting z by $z^{\operatorname{lcm}(k_1,k_2)}$ in (8.3) gives

$$\begin{aligned} F\left(z^{\operatorname{lcm}(k_1,k_2)}\right) &= \sum_{n=0}^{\infty} s_{1_n} z^{(m_1 \cdot n + j_1)\cdot k_2} + \sum_{n=0}^{\infty} s_{2_n} z^{(m_2 \cdot n + j_2)\cdot k_1} && (8.4) \\ &= \sum_{n \in k_2 \cdot \left(m_1 \cdot \mathbb{N}_{\geqslant 0} + j_1\right)} a_{1_{\frac{n}{k_2}}} z^n + \sum_{n \in k_1 \cdot \left(m_2 \cdot \mathbb{N}_{\geqslant 0} + j_2\right)} a_{2_{\frac{n}{k_1}}} z^n, && (8.5) \end{aligned}$$

where a_{i_n} is obtained from s_{i_n} by the change of variable (7.34), on p.141, $i \in \{1, 2\}$.

Observe that in (8.4) the powers of the indeterminate z are integers. In general, the right-hand side of (8.2) always gives a representation with integer powers when we substitute z by z^μ, for any positive multiple μ of $\operatorname{lcm}(k_1, k_2)$. Power series with integer powers are dealt with in other sections of this chapter. Thus our aim of determining the positive integers k_i, $i \in [\![1, I]\!]$ in (8.2) can be reduced in finding a positive multiple μ of $\operatorname{lcm}(k_1, \ldots, k_I)$ so that we can compute the power series of $f(z^\mu)$ and substitute z by $z^{1/\mu}$ in the obtained representation to get the one of $f(z)$. This idea was initiated in [Gruntz and Koepf, 1995] for the two-term holonomic RE case.

By the general representation (7.22) of an m-fold hypergeometric term, we know that there exist rational functions $r_1(n)$ and $r_2(n)$ such that

$$a_{1_{n+m_1}} = r_1(n) \cdot a_{1_n} \text{ and } a_{2_{n+m_2}} = r_2(n) \cdot a_{2_n},$$

for the coefficients in (8.5). Therefore we can write

$$a_{1_{\frac{n}{k_2}+m_1}} = r_1\left(\frac{n}{k_2}\right) \cdot a_{1_{\frac{n}{k_2}}} \text{ and } a_{2_{\frac{n}{k_1}+m_2}} = r_2\left(\frac{n}{k_1}\right) \cdot a_{2_{\frac{n}{k_1}}}. \tag{8.6}$$

[1] Two integers are said to be co-prime if the only positive factor that they have in common is 1.

where $\frac{n}{k_1}$ and $\frac{n}{k_2}$ are not necessarily integers.

To compute the holonomic recurrence equation of smallest order for the m_1-fold and the m_2-fold hypergeometric terms $a_{1_{\frac{n}{k_2}}}$ and $a_{2_{\frac{n}{k_1}}}$, we need to use the smallest integer k such that $k \cdot \frac{n}{k_2} \in \mathbb{N}_{\geqslant 0}$ and $k \cdot \frac{n}{k_1} \in \mathbb{N}_{\geqslant 0}$. Thus $k = \operatorname{lcm}(k_1, k_2)$ and the obtained holonomic RE is of course compatible with the one computed using Algorithm 2 for the input expression $F(z)$. From (8.6), substituting n by $\operatorname{lcm}(k_1, k_2) \cdot n = k_1 \cdot k_2 \cdot n$ yields

$$a_{1_{k_1 \cdot n+m_1}} = r_1\,(k_1 \cdot n) \cdot a_{1_{k_1 \cdot n}} \text{ and } a_{2_{k_2 \cdot n+m_2}} = r_2\,(k_2 \cdot n) \cdot a_{2_{k_2 \cdot n}}. \tag{8.7}$$

Since $a_{1_{k_1 \cdot n+m_1}}$ and $a_{2_{k_2 \cdot n+m_2}}$ are, respectively, m_1-fold and m_2-fold hypergeometric term solutions of a holonomic recurrence equation satisfied by the Taylor coefficients of $f(z)$, by *mfoldHyper* we know how such terms are computed using an algorithm to compute the equivalent hypergeometric terms s_{i_n} such that

$$\frac{s_{i_{n+1}}}{s_{i_n}} = \frac{a_{i_{n+m_i}}}{a_{i_n}} = r_i(k_i \cdot n),\ i \in \{1, 2\}.$$

By Petkovšek's algorithm we know that ratios of hypergeometric term solutions of holonomic REs are built from monic factors of the corresponding trailing and leading polynomial coefficients. This implies in particular that some zeros and poles of $r_i(k_i \cdot n)$ are the trailing and leading polynomial coefficient roots of the holonomic recurrence equation computed by Algorithm 2 for the Taylor coefficients of $f(z)$, $i \in \{1, 2\}$. Therefore by computing the least common multiple of all the trailing and leading polynomial coefficient rational root denominators of that RE we must obtain a multiple of $\operatorname{lcm}(k_1, k_2)$.

This ends the procedure due to the determination of Puiseux numbers involved in the power series expansion of a given hypergeometric type function. The following Maxima program implements this procedure provided the leading and trailing polynomial coefficients $P_0(n)$, $P_d(n)$ of a holonomic recurrence equation satisfied by the Taylor coefficients of a hypergeometric type expression, and the index variable.

```
Puiseuxnbrfun(P0,Pd,n):=block([L,k],
    L: append(map(rhs, solve(Pd,n)), map(rhs, solve(P0,n))),
    L: sublist(L, numberp),
    k: 1,
    if(length(L)>0) then k: lreduce(lcm, map(denom,L)),
    k
)$
```

Example 8.1.

We consider the expression $f(z) = \exp(z^{3/4}) + \sin(\sqrt{z})$.

```
(%i1) f:exp(z^(3/4)) + sin(sqrt(z))$
```

```
(%i1) RE:FindRE(f,z,a[n]);
```

(%o1) $$-17920\cdot(1+n)\cdot(2+n)\cdot(1+2\cdot n)\cdot(3+2\cdot n)\cdot(4\cdot n-1)\cdot(5+4\cdot n)\cdot a_{n+2} -8960\cdot(n-1)\cdot(1+n)\cdot(2\cdot n-1)\cdot(1+2\cdot n)\cdot(4\cdot n-5)\cdot(1+4\cdot n)\cdot a_{n+1}-896\cdot n\cdot(2\cdot n-3) \cdot(4\cdot n-9)\cdot(4\cdot n-3)\cdot\left(1268-2947\cdot n+842\cdot n^2\right)\cdot a_n+128\cdot\left(22077440-68216910\cdot n +83470449\cdot n^2-51845440\cdot n^3+17242620\cdot n^4-2907840\cdot n^5+193856\cdot n^6\right)\cdot a_{n-1}-32\cdot\left(116467183 -225724547\cdot n+177451566\cdot n^2-72252320\cdot n^3+16016512\cdot n^4-1824768\cdot n^5+82944\cdot n^6\right)\cdot a_{n-2} +1296\cdot\left(53607386-71522380\cdot n+39130073\cdot n^2-11247312\cdot n^3+1792828\cdot n^4-150336\cdot n^5 +5184\cdot n^6\right)\cdot a_{n-3}+648\cdot\left(1595875-1279535\cdot n+392722\cdot n^2-53136\cdot n^3+2592\cdot n^4\right)\cdot a_{n-4} +648\cdot\left(22243-12312\cdot n+1296\cdot n^2\right)\cdot a_{n-5}-13122\cdot\left(6302-2025\cdot n+162\cdot n^2\right)\cdot a_{n-6} -531441\cdot a_{n-7}=0$$

```
(%i2) CoeffsRE: REcoeff(RE,a[n])$
```

The corresponding leading polynomial coefficient is

```
(%i3) last(CoeffsRE);
```

(%o3) $$-17920\cdot(8+n)\cdot(9+n)\cdot(15+2\cdot n)\cdot(17+2\cdot n)\cdot(27+4\cdot n)\cdot(33+4\cdot n)$$

and the trailing one is

```
(%i4) first(CoeffsRE);
```

(%o4) $$-531441$$

Using *Puiseuxnbrfun* one finds

```
(%i5) Puiseuxnbrfun(first(CoeffsRE),last(CoeffsRE),n);
```

(%o5) $$4$$

which for this example represents exactly $\mathrm{lcm}(1,2,4,4)=4$. Indeed the factors $(8+n)$ and $(9+n)$ have both denominator roots equal to 1, $(15+2\cdot n)$ and $(17+2\cdot n)$ have both denominator roots equal to 2, and $(27+4\cdot n)$ and $(33+4\cdot n)$ have both denominator roots equal to 4. After substitution the new holonomic RE to consider is

```
(%i6) RE:FindRE(subst(z^4,z,f),z,a[n]);
```

(%o6) $$-4\cdot(n-3)\cdot(n-2)\cdot(n-1)\cdot(1+n)\cdot a_{n+1}-36\cdot(n-7)\cdot(n-4)\cdot(n-3)\cdot(n-1)\cdot a_{n-1} +16\cdot(n-6)\cdot(n-3)\cdot\left(58-16\cdot n+n^2\right)\cdot a_{n-3}+36\cdot(n-9)\cdot(n-7)\cdot\left(141-38\cdot n +2\cdot n^2\right)\cdot a_{n-5}+(n-7)\cdot\left(-159562+38863\cdot n-3078\cdot n^2+81\cdot n^3\right)\cdot a_{n-7}+144\cdot\left(158 -26\cdot n+n^2\right)\cdot a_{n-9}-324\cdot\left(68-19\cdot n+n^2\right)\cdot a_{n-11}-9\cdot\left(15859-2268\cdot n+81\cdot n^2\right)\cdot a_{n-13} -2592\cdot a_{n-15}-2916\cdot a_{n-17}=0$$

whose Puiseux numbers are all equal to 1 as the use of *Puiseuxnbrfun* demonstrates below.

```
(%i7) CoeffsRE:REcoeff(RE,a[n])$
(%i8) Puiseuxnbrfun(first(CoeffsRE),last(CoeffsRE),n);
```

$$(\%o8) \qquad 1$$

As we now know how to reduce the computation of Puiseux series to the one of Laurent series, in the next sections we will assume all the Puiseux numbers to be 1.

8.2 Computation of Hypergeometric Type Series Starting Points

In Section 7.2 we saw that Maple's FPS command wrongly represents the power series of $\arctan(z) + \exp(z)$ due to the constant term missing. In this section we show how to avoid such a situation by explaining how to deduce exact starting points of hypergeometric type series from the holonomic recurrence equations of their general coefficients. By trying many examples of sums of polynomials and hypergeometric series with Maple's FPS command, one sees that such computation is not well managed in the implemented algorithm. A simple example is the following.

Maple's FPS gives

```
>  FPS(z + z^2 * exp(z),z,n);
```

$$FPS(z + z^2 e^z, z, n)$$

whereas our Maxima FPS implementation yields correctly

```
(%i1) FPS(z+z^2*exp(z),z,n);
```

$$(\%o1) \qquad \left(\sum_{n=0}^{\infty} \frac{z^{2+n}}{n!}\right) + z$$

which is the sum of the hypergeometric series of $z^2 \cdot \exp(z)$ whose starting point is $n = 2$ plus the polynomial z. All this informations can be deduced from the corresponding holonomic recurrence equation. Let us now explain how this can be done.

Again, we consider the general representation (assuming Puiseux numbers all equal to 1)

$$f(z) := T(z) + F(z) \tag{8.8}$$

where $F(z)$ is a sum of hypergeometric type series and $T(z) \in \mathbb{K}[\log(z)][z, \frac{1}{z}]$ is an extra term to be determined while computing the starting point for $F(z)$. Note that $T(z)$ can be given explicitly in the input expression, but also implicitly like for the expressions $\operatorname{arcsech}(z)$, $\operatorname{arccosh}(z)$ and $\exp(z) + \log(1 + z)$.

First, we focus on the case where $T(z)$ is a Laurent polynomial in $\mathbb{K}[z, \frac{1}{z}]$. For this purpose we need to understand what it means for a Laurent polynomial that its coefficients are solution of a holonomic recurrence equation.

The following Maxima code generates a Laurent polynomial with degree between $-M = -10$ and $N = 10$, with integer coefficient in $[\![-100, 100]\!]$.

```
randompoly(z):=block([N,M,P],
    N: random(11),
    M: -random(11),
    P:0,
    for i:M thru N do
    P: P+(-random(2)*random(101)+random(2)*random(101))*z^i,
    P
)$
```

Let us compute the holonomic recurrence equation for an unknown Laurent polynomial and figure out some properties of its coefficients from that RE.

```
(%i1) FindRE(randompoly(z),z,a[n]);
```

$$(\%o1) \qquad -3\,(5+n)\,a_n - 10\,(3+n)\,a_{n-1} + 3\,(n-1)\,a_{n-3} - 7\,(n-5)\,a_{n-5} - (n-7)\,a_{n-6} + 26\,(n-11)\,a_{n-8} = 0. \tag{8.9}$$

Since all polynomials are rational functions, *HolonomicDE* always computes a holonomic differential equation of first order for a given polynomial. Following Algorithm 1, if

$$T(z) := \sum_{i=M}^{N} c_i z^i = \sum_{i\in\mathbb{Z}} c_i z^i \in \mathbb{K}(z), \tag{8.10}$$

for $M, N \in \mathbb{Z}$, $M \leqslant N$ where $c_i = 0$ for $i \in \mathbb{Z} \setminus [\![M, N]\!]$, and $c_{-M} \cdot c_N \neq 0$, then the differential equation found is

$$\sum_{i=-M}^{N} c_i z^i \cdot F'(z) - \sum_{i=-M}^{N} c_i i z^{i-1} \cdot F(z) = 0. \tag{8.11}$$

Therefore using the rewrite rule (4.38) we obtain the recurrence equation

$$\sum_{i=M}^{N} c_i (n+1-i) \cdot a_{n+1-i} - \sum_{i=M}^{N} c_i i \cdot a_{n-(i-1)} = \sum_{i=M}^{N} c_i (n+1-2i) \cdot a_{n+1-i} = 0.$$

Hence the holonomic RE found by *FindRE* of a Laurent polynomial with representation (8.10) is given by

$$\sum_{i=M}^{N} c_i (n+1-2i) \cdot a_{n+1-i} = 0, \tag{8.12}$$

or equivalently

$$\sum_{i=M}^{N} c_i(n+N-2i) \cdot a_{n+N-i} = 0, \tag{8.13}$$

after substitution of n by $n+N$ for normalization.

Thus, without even using initial values a polynomial whose coefficients satisfy the holonomic RE (8.9) can easily be found by equating terms of (8.12) and (8.9) to find the unknown coefficients c_i, using *FindRE* to compute a holonomic RE for the resulting polynomial and check whether the REs are identical. We obtain the Laurent polynomial

$$-26 \cdot z^3 + z - \frac{3}{z^2} + \frac{10}{z^4} + \frac{3}{z^5} + 7. \tag{8.14}$$

Of course this is not enough because there might be other solutions. And moreover, when the input expression is of the form (8.8), the situation is more complicated since $F(z)$ needs initial values in order to be computed. Therefore, we have to find the maximum degree $N \in \mathbb{Z}$ of $T(z)$ so that $F(z)$ starts at $N+1$ and $T(z)$ is computed by a generalized Taylor expansion of order N of $f(z)$.

Observe for each non-zero coefficient c_i, $i \in [\![M, N]\!]$ of $T(z)$, that $2i - N$ is the root of one polynomial coefficient in (8.13). In particular, N is the trailing polynomial coefficient root and M is the root of the leading polynomial coefficient shifted by $N-M$. These two properties of the degrees of a potential Laurent polynomial whose coefficients satisfy a holonomic recurrence equation is preserved in the general case. This is established by the following lemma.

Lemma 8.1. *Let $\mathbb{K}$ be a field of characteristic zero, $N, M \in \mathbb{Z}, N \geqslant M$, $T(z) \in \mathbb{K}[z, \frac{1}{z}]$ be a Laurent polynomial of degree N and lowest non-zero monomial degree M. The coefficients of $T(z)$ satisfy the holonomic recurrence equation*

$$P_d a_{n+d} + P_{d-1} a_{n+d-1} + \ldots + P_0 a_n = 0, \tag{8.15}$$

$d \in \mathbb{N}, P_j \in \mathbb{K}[n], j \in [\![0, d]\!], P_d \cdot P_0 \neq 0,$ *if N is a root of P_0 and M is a root of $P_d(n-d)$.*

Proof. Suppose that the coefficients of $T(z)$ satisfy (8.15). Since $T(z)$ has finitely many non-zero coefficients we can write

$$T(z) = \sum_{i \in \mathbb{Z}} c_n z^n,$$

where $c_n = 0$ for $n \in \mathbb{Z} \backslash [\![M, N]\!]$. Saying that the coefficients of $T(z)$ satisfy (8.15) is equivalent to say that the sequence $(c_n)_{n \in \mathbb{Z}}$ is a sequence solution of (8.15). Given that (8.15) is valid for all integers, observe that substituting a_n by c_n in (8.15) for sufficiently large positive or negative integers vanishes all the summands on the left-hand side of (8.15).

Furthermore, we can make a substitution such that either the trailing or the leading term does not necessarily give zero. Indeed, since $c_n = 0$ for $n \in \mathbb{Z} \setminus [\![M, N]\!]$, substituting a_n by c_n in (8.15) for $n = N$ yields

$$P_0(N) c_N = 0,$$

and therefore using the assumption $c_N \neq 0$ we deduce that $P_0(N) = 0$. Similarly, substituting a_n by c_n in (8.15) for $n = M - d$ gives

$$P_d(M-d)c_M = 0,$$

and therefore as $c_M \neq 0$ by assumption, it follows that $P_d(M-d) = 0$. □

Remark. Note that generally when $T(z) = 0$ and $F(z)$ starts at 0, $N = M = 0$ and 0 is not necessarily a root of the trailing polynomial coefficient. This might be interpreted from the fact that the zero function is always solution of any holonomic RE and moreover takes 0 at 0. Therefore as $T(z)$ does not play a disturbing role, we rather say in this case that it does not exist. This is the case with

```
(%i2) FindRE(exp(z),z,a[n]);
```

$$(\%o2) \qquad (1+n)\cdot a_{n+1} - a_n = 0$$

whose trailing polynomial coefficient does not have any root. $T(z)$ in this case is 0 or is said to not exist. However, note that in this example $M = 0$ is a root of the leading polynomial coefficient which represents the starting point of the series expansion of $\exp(z)$. In general, the computation of M is always possible from all the REs computed by *FindRE*, and represents moreover the starting point of the generalized Taylor expansion of the given $f(z)$. Indeed, the fact that *FindRE* does not simplify the common factors after application of the rewrite rule (4.38) on p. 52 is essential for our computations of starting points. These factors contain necessary information to determine the first non-zero coefficient of the series expansion sought. Let for example

```
(%i3) FindRE(z/(1-z),z,a[n]);
```

$$(\%o3) \qquad (n-1)\cdot a_{n-1} - (n-1)\cdot a_n = 0$$

for which the simplification of the common factor $(n-1)$ (or n after normalization) would hide the starting point $N+1 = 1$ ($N = 0$ is the root of the trailing polynomial coefficient).

Observe that using this lemma we can now affirm that any Laurent polynomial whose sequence of coefficients satisfy the holonomic RE (8.9) is a constant multiple of the polynomial (8.14). Indeed, the leading and the trailing polynomial coefficients of (8.9) have only one integer root each which are the degree bounds of (8.14).

Lemma 8.1 extends to Laurent polynomials in $\mathbb{K}[\log(z)][z, \frac{1}{z}]$. The point is that in symbolic computation of generalized Taylor expansions or power series (see [Kauers and Paule, 2011, Section 7.3], the formal $\log(z)$ behaves as presented below.

```
(%i4) taylor(log(z),z,0,-1);
```

$$(\%o4)/T/ \quad 0 + \ldots$$

```
(%i5) taylor(log(z),z,0,0);
```

$$(\%o5)/T/\quad \log(z) + \ldots$$

```
(%i6) taylor(z*log(z),z,0,1);
```

$$(\%o6)/T/\quad (\log(z) + \ldots)\cdot z + \ldots$$

```
(%i7) taylor(z*log(z),z,0,0);
```

$$(\%o7)/T/\quad 0 + \ldots$$

This is exactly the behavior of any constant when we avoid the canonical representation. In the generalized hypergeometric type case, the impact of the formal $\log(z)$ in a recurrence equation computed by *FindRE* appears as a multiple integer root of the trailing or leading polynomial coefficient.

```
(%i8) FindRE(log(z)*exp(z),z,a[n]);
```

$$(\%o8)\qquad (1+n)^2\cdot a_{n+1} - (1+2\cdot n)\cdot a_n + a_{n-1} = 0$$

```
(%i9) FindRE(log(z)^2*z^2,z,a[n]);
```

$$(\%o9)\qquad (n-2)^3\cdot a_n = 0$$

Without going into much details, one can state that the computation of degree bounds for $T(z)$ in (8.8) remains unchanged if $T(z)$ is a Laurent polynomial in $\in \mathbb{K}[\log(z)][z, \frac{1}{z}]$.

Algorithmically, we proceed as follows.

Algorithm 12 Computation of $T(z)$ and the starting point of $F(z)$ for a given hypergeometric type expression f with representation (8.8)

Input: An expression f whose Taylor coefficients satisfy the holonomic recurrence equation

$$P_d a_{n+d} + P_{d-1} a_{n+d-1} + \ldots + P_0 a_n = 0, \tag{8.16}$$

$d \in \mathbb{N}, P_j \in \mathbb{K}[n], j \in [\![0, d]\!], P_d \cdot P_0 \neq 0,$

Output: $T(z)$ and a starting point N_0 for $F(z)$ for the representation (8.8) of f.

1. Compute the minimum integer roots M of $P_d(n-d)$ and the maximum integer root N of $P_0(n)$.
2. If N does not exist then set $T(z) := 0$ and set $N_0 := M$.
3. If N does exist then set $T(z) := Taylor(f(z), z, 0, N)$ and set $N_0 := N + 1$.
4. Return $[T(z), N_0]$.

The following Maxima code implements this procedure for any hypergeometric type function $f(z)$ and gives outputs as a list containing the corresponding $T(z)$ and the starting point for $F(z)$.

```
LPolyPart(f,z):=block([RE,N,M],
    RE: FindRE(f,z,a[n]),
    if(RE#false) then(
        RE: REcoeff(RE,a[n]),
        d: length(RE)-1,
        M: map(rhs, solve(subst(n-d,n,last(RE)),n)),
        M: lmin(sublist(M, integerp)),
        N: sublist(map(rhs, solve(first(RE),n)),integerp),
        if(length(N)<1) then [0, M]
        else (
            N: lmax(N),
            [ratdisrep(taylor(f,z,0,N)), N+1]
        )
    )
)$
```

Let us apply this procedure to some examples.

```
(%i10) LPolyPart(log(1+z),z);
```

$$(\%o10) \quad [0, 1]$$

```
(%i11) LPolyPart(asech(z),z);
```

$$(\%o11) \quad [\log(2) - \log(z), 1]$$

```
(%i12) LPolyPart(exp(z)+log(1+z),z);
```

$$(\%o12) \quad [1, 1]$$

```
(%i13) LPolyPart(sin(z)/z^5,z);
```

$$(\%o13) \quad [0, -5]$$

```
(%i14) LPolyPart(atan(z)+exp(z),z);
```

$$(\%o14) \quad [1, 1]$$

Observe in this latter example that $T(z)$ is found to be 1 but the representation given by our Maxima FPS implementation does not contain this extra term. The reason is that $N+1$ is not necessarily the exact starting point but rather its maximum value possible. This shows that the exact $T(z)$ is not uniquely determined but only the polynomial from which it can always be subtracted. Nevertheless, it is safe to have such a value since it does not affect the correctness of the result and moreover, some terms can always be subtracted from the obtained $T(z)$ when $F(z)$ can be used to compute these. Therefore our algorithm tries to subtract certain terms from

$T(z)$ after having found the linear combination needed for $F(z)$. For the latter example, it turns out that $T(z)$ is obtained as $\frac{z^0}{0!}$ from one of the obtained hypergeometric type series.

As last example let us take the case of the Chebyshev polynomial $\cos(4\arccos(z))$.

```
(%i14) LPolyPart(cos(4*acos(z)),z);
```

$$(\%o14) \qquad [8 \cdot z^4 - 8 \cdot z^2 + 1, 5]$$

Thus, the starting point to compute the linear combination for $F(z)$ is 5. This example leads to a two-term holonomic recurrence equation. We will see in the next section that the linear combination of the corresponding hypergeometric type series yields 0 so that one finally gets the known result $\cos(4\arccos(z)) = T(z) = 8 \cdot z^4 - 8 \cdot z^2 + 1$.

8.3 The Two-Term Holonomic RE Case

One may ask why do we need such an algorithm since it is already generalized with Algorithm 11 (mfoldHyper), on p. 147. This could be answered by the following example whose corresponding two-term holonomic recurrence relation has hypergeometric and 2-fold hypergeometric term solutions over $\mathbb{Q}$.

```
(%i1) RE:FindRE(cosh(z),z,a[n]);
```

$$(\%o1) \qquad (1+n) \cdot (2+n) \cdot a_{n+2} - a_n = 0$$

```
(%i2) mfoldHyper(RE,a[n]);
```

$$(\%o2) \qquad \left[\left[1, \left\{\frac{1}{n!}, \frac{(-1)^n}{n!}\right\}\right], \left[2, \left\{\frac{1}{(2 \cdot n)!}\right\}\right]\right]$$

```
(%i3) mfoldHyper(RE,a[n],2,1);
```

$$(\%o3) \qquad \frac{1}{(2 \cdot n + 1) \cdot (2 \cdot n)!}$$

Thus we get four terms for a recurrence equation of order 2 and the coefficients of the linear combination sought are solutions of a linear system of two equations with 4 unknowns. This leads to a solution with two arbitrary constants that are set to zero. In our case these are the constant coefficients of the involved hypergeometric type series of type 2. Therefore we obtain a power series representation with a linear combination of hypergeometric terms as general coefficient. That is

$$\cosh(z) := \sum_{n=0}^{\infty} \frac{1 + (-1)^n}{2n!} z^n, \tag{8.17}$$

which is not the representation we want as it computes infinitely many zero coefficients for odd powers of the indeterminate z. Note that this situation could be avoided by some considerations on the involved holonomic RE order and the linear system to be solved. But we will not do so as the situation could be inappropriate in the general case when there are more m-fold hypergeometric term solutions of the computed holonomic RE. Moreover, since in the two-term holonomic RE case the corresponding m-fold hypergeometric term ratio is easily computable, the m computed formulas using Algorithm 7 are either rational functions or ratios of products of factorials with linear arguments and Pochhammer symbols. This is a nicer simplification than what we obtain using *mfoldHyper* (see $(\%o3)$ above) for which returned formulas are either rational functions or products of a rational function times ratios of products of factorials with linear arguments and Pochhammer symbols. The appearing multiplicative rational functions are computed when Algorithm 9 from p. 112 is used in mfoldHyper's calls of Algorithm 10. Other advantages of particularly considering the two-term holonomic RE case will be given at the end of this section.

Let us sketch the two-term holonomic RE algorithm for our introductory example. The recurrence equation found for $\cosh(z)$ is

$$(1+n)\cdot(2+n)\cdot a_{n+2} - a_n = 0.$$

We immediately get the symmetry number $m = 2$. Therefore the corresponding 2-fold symmetric ratios are

$$\frac{a_{2(n+1)}}{a_{2n}} = \frac{1}{(2\cdot n+1)\cdot(2\cdot n+2)} \quad \text{and} \quad \frac{a_{2(n+1)+1}}{a_{2n+1}} = \frac{1}{(2\cdot n+2)\cdot(2\cdot n+3)}. \tag{8.18}$$

We can now apply Algorithm 7.

```
(%i4) r0:1/((2*n+1)*(2*n+2))$
(%i5) r1:1/((2*n+2)*(2*n+3))$
(%i6) pochfactorsimp(r0,n);
```

$$(\%o6) \qquad \frac{1}{(2\cdot n)!}$$

```
(%i7) pochfactorsimp(r1,n);
```

$$(\%o7) \qquad \frac{1}{(1+2\cdot n)!}$$

Note that in the general case before using Algorithm 7 we have to make sure that the given ratios do not have non-negative integer roots. This is ensured by an integer shift of the starting point to the ratio of the trailing and the leading polynomial coefficients. In this example the starting point is 0.

We now write

$$I(z) := \alpha_0 \cdot \sum_{n=0}^{0} \frac{z^{2\cdot n}}{(2\cdot n)!} + \alpha_1 \cdot \sum_{n=0}^{0} \frac{z^{2\cdot n+1}}{(2\cdot n+1)!},$$

and use 2 initial values $n = 0 = 2\cdot 0$ and $n = 1 = 2\cdot 0 + 1$ to find the unknown constants α_0 and α_1. We have

```
(%i8) taylor(cosh(z),z,0,1);
```

$$(\%o8)/T/ \quad 1 + \ldots$$

therefore $\alpha_0 + \alpha_1 \cdot z = 1$, hence $\alpha_0 = 1$ and $\alpha_1 = 0$. And finally we obtain the power series representation

$$\cosh(z) = \sum_{n=0}^{\infty} \frac{z^{2\cdot n}}{(2\cdot n)!}, \tag{8.19}$$

as expected.

For $\cos(4 \cdot \arccos(z))$, *FindRE* gives the recurrence equation

```
(%i9) RE:FindRE(cos(4*acos(z)),z,a[n]);
```

$$(\%o9) \qquad (n-4)\cdot(4+n)\cdot a_n - (1+n)\cdot(2+n)\cdot a_{n+2} = 0$$

As computed in the previous section, the starting point for the corresponding hypergeometric type series part is 5. Shifting the ratio of the trailing and the leading polynomial coefficients by 5 yields

$$-\frac{(1+n)\cdot(9+n)}{(n+6)\cdot(n+7)}.$$

Therefore the corresponding 2-fold symmetric ratios are

$$r_0 = -\frac{(1+2\cdot n)\cdot(9+2\cdot n)}{(2\cdot n+6)\cdot(2\cdot n+7)} \text{ and } r_1 = -\frac{(2+2\cdot n)\cdot(10+2\cdot n)}{(2\cdot n+7)\cdot(2\cdot n+8)} \tag{8.20}$$

which after applying Algorithm 7 lead to the "simple" formulas

```
(%i10) h0:pochfactorsimp(r0,n);
```

$$(\%o10) \qquad \frac{2\cdot(7+2\cdot n)\cdot(-1)^n\cdot(2\cdot n)!}{7\cdot(n+1)\cdot(n+2)\cdot 4^n\cdot n!^2}$$

```
(%i11) h1:pochfactorsimp(r1,n);
```

$$(\%o11) \qquad \frac{15\cdot(1+n)\cdot(2+n)\cdot(4+n)\cdot(-1)^n\cdot 4^n\cdot n!^2}{(5+2\cdot n)!}$$

We have to use 2 initial values corresponding to $n = 2\cdot 0+5$ and $n = 2\cdot 0+1+5$. First, we define

```
(%i12) I:alpha[0]*subst(0,n,h0)*z^5 + alpha[1]*subst(0,n,h1)*z^6
         + Taylor(cos(4*acos(z)),z,0,4);
```

$$(\%o12) \qquad \alpha_1\cdot z^6 + \alpha_0\cdot z^5 + 8\cdot z^4 - 8\cdot z^2 + 1$$

for the unknown constants α_0 and α_1. Remember that $Taylor(\cos(4\arccos(z)), z, 0, 4) = T(z)$ for the representation (8.8) of $\cos(4\arccos(z))$. To find the values of α_0 and α_1, we just have to solve the trivial identity

```
(%i13) I-Taylor(cos(4*acos(z)),z,0,6)=0;
```

$$(\%o13) \qquad \alpha_1\cdot z^6 + \alpha_0\cdot z^5 = 0$$

We thus obtain $\alpha_0 = \alpha_1 = 0$ and therefore

$$\cos(4\cdot\arccos(z)) = 1 - 8\cdot z^2 + 8\cdot z^4. \tag{8.21}$$

In a nutshell the algorithm presents as follows.

Algorithm 13 Power series representations of hypergeometric type functions whose *FindRE* computes two-term holonomic REs for their Taylor coefficients.

Input: A holonomic expression f and a holonomic RE

$$Q(n)a_{n+m} - P(n)a_n = 0, \tag{8.22}$$

$P, Q \in \mathbb{K}[n], P \cdot Q \neq 0$, computed by *FindRE* for the Taylor coefficient of f.

Output: The power series representation of f at 0.

1. Use Algorithm 12 to compute the corresponding $T(z)$ and a starting point N_0 for the representation (8.8) of f.
2. Set $r(n) = \frac{P(n+N_0)}{Q(n+N_0)}$.
3. Compute the m symmetric ratios

$$r_j(n) := r(m \cdot n + j), \; j = 0, \ldots, m-1. \tag{8.23}$$

4. Use Algorithm 7 to compute $h_j(n) := \prod_{i=0}^{n-1} r_j(i)$, for $j = 0, \ldots, m-1$.
5. Set $I(z) := T(z) + \sum_{j=0}^{m-1} \alpha_j \cdot h_j(0) \cdot z^{j+N_0}$, where $\alpha_j, j = 0, \ldots, m-1$ are unknown constants.
6. Find the constant values α_j by equating the coefficient of the Laurent polynomial

$$Taylor(f(z), z, 0, N_0 + m - 1) - I(z) \tag{8.24}$$

to zero.

7. Return $T(z) + \sum_{j=0}^{m-1} \alpha_j \cdot \sum_{n=0}^{\infty} h_j(n) \cdot z^{m \cdot n + j + N_0}$

This algorithm is a variant of the one in [Koepf, 1992] for hypergeometric type functions leading to two-term holonomic REs. At this stage we can already compute many power series representations of meromorphic functions. This algorithm is incorporated in our Maxima FPS implementation.

```
(%i14) FPS(sin(z)/z^1000,z,n);
```

$$(\%o14) \quad \sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{2 \cdot n - 999}}{(1 + 2 \cdot n)!}$$

```
(%i15) FPS(exp(z)*z^1000,z,n);
```

$$(\%o15) \quad \sum_{n=0}^{\infty} \frac{z^{1000+n}}{n!}$$

```
(%i16) FPS((sqrt(sqrt(z)+1)-sqrt(1-sqrt(z)))
            /(sqrt(2)*sqrt(z)),z,n);
```

$$(\%o16) \quad \sum_{n=0}^{\infty} \frac{\left(\frac{1}{4}\right)_n \cdot \left(\frac{3}{4}\right)_n \cdot 4^n \cdot z^n}{\sqrt{2} \cdot (1 + 2 \cdot n)!}$$

Observe in this two-term holonomic RE case that the computation of the linear combination sought is made easier by the following facts:

1. the Taylor expansion of order $N_0 + m - 1$ of $f(z)$ suffices to determine the linear combination sought;

2. the obtained linear system to compute the coefficients of the linear combination sought is trivial (with an identity matrix) and has a unique solution;

3. the starting point computed by Algorithm 12 yields starting points for all the involved m-fold symmetric terms.

All this facts do not hold in the general case since we have to deal with m-fold symmetric terms with different values of m. For example, suppose we have the input $\frac{\sin(z)}{z^3} + \exp(z)$, then Algorithm 12 finds

```
(%i17) LPolyPart(sin(z)/z^3+exp(z),z);
```

$$(\%o17) \qquad [0, -3]$$

However, by prior knowledge we know that the power series of $\exp(z)$ starts at 0. This shows that in the general case the starting point could not similarly be used as in Algorithm 13 step 5. We move on to this more complicated situation.

8.4 The General Hypergeometric Type Series Case

As previously we consider an expression $f(z)$ related to a hypergeometric type function and we want to compute the representation

$$f(z) := T(z) + F(z), \tag{8.25}$$

where $T(z)$ is generally a Laurent polynomial in $\mathbb{K}[\log(z)][z, \frac{1}{z}]$ (in practice $\mathbb{K} = \mathbb{C}$), and $F(z)$ is a sum of hypergeometric type series. We have already shown how to compute $T(z)$ and a starting point for $F(z)$.

Unlike the two-term holonomic RE case where the starting point computed by Algorithm 12 can always easily be used for all the m-fold symmetric terms to consider, in the general case one has to take into account all the involved m-fold hypergeometric terms. Indeed, we have to avoid negative arguments for the evaluation of m-fold hypergeometric terms which are supposed to start at least at 0 according to Algorithm 7. When there are many different types involved in $F(z)$ as we saw with the example of $\arctan(z) + \exp(z)$, the exact starting point is in fact the minimal one among those of the hypergeometric type series in $F(z)$. We only have to make sure that this value is positive in order to avoid any division by 0 or factorial of negative integers when evaluating m-fold hypergeometric terms.

In practice, we use

$$N_1 = \begin{cases} \max\{N_0, 0\}, & \text{if } N_0 < 0 \\ N_0 & \text{if } N_0 \geqslant 0 \end{cases} \geqslant 0, \tag{8.26}$$

from which all necessary m-fold hypergeometric terms can be evaluated for initial conditions. Once the linear combination sought is found, if possible we subtract terms from the corresponding $T(z)$ for the indices $n \in [\![N_0, N_1]\!]$. Note that $T(z)$ should also be modified accordingly. We set

$$T_1(z) := Taylor(f(z), z, 0, N_1 - 1). \tag{8.27}$$

By these measures, the final representation gives a normal form after shifting the power of the indeterminate by the first non-zero term index in each hypergeometric type series.

Let us now find a representation (8.25) of $f(z)$ for a computed $T_1(z)$ and a starting point N_1 knowing that we can make further computations to subtract terms in $T_1(z)$ that can be deduced from $F(z)$.

To ease the understanding of the general case, we first describe the details for the situation where

$$\mathcal{H} = [[2, \{h_{2n}\}], [3, \{h_{3n}\}]], \tag{8.28}$$

represent the obtained basis of m-fold hypergeometric term solutions of the holonomic RE given by *FindRE* for the Taylor coefficients of $f(z)$. Let us also assume for simplicity that $N_1 = T_1(z) = 0$, then the general form for the corresponding $F(z)$ can be written as

$$\begin{aligned} F(z) = \alpha_{2,0} \cdot \sum_{n=0}^{\infty} h_{2n} z^{2n} + \alpha_{2,1} \cdot \sum_{n=0}^{\infty} h_{2n+1} z^{2n+1} + \alpha_{3,0} \cdot \sum_{n=0}^{\infty} h_{3n} z^{3n} \\ + \alpha_{3,1} \cdot \sum_{n=0}^{\infty} h_{3n+1} z^{3n+1} + \alpha_{3,2} \cdot \sum_{n=0}^{\infty} h_{3n+2} z^{3n+2}, \end{aligned} \tag{8.29}$$

$\alpha_{2,i}, \alpha_{3,j} \in \mathbb{K}, i = 0, 1, j = 0, 1, 2$. Hence we have five unknowns to determine. Observe that computing a Taylor expansion of order 4 of $f(z)$ could not be enough. Indeed, the Taylor expansion of order 4 would give five linear equations for the unknown constants but it turns out that the obtained linear system is not sufficient to determine these. Assume

$$Taylor(f(z), z, 0, 4) = t_0 + t_1 z + t_2 z^2 + t_3 z^3 + t_4 z^4, \tag{8.30}$$

then equating the coefficients with their correspondents in (8.29) yields the linear system

$$\begin{cases} \alpha_{2,0} \cdot h_{2n}(0) + \alpha_{3,0} \cdot h_{3n}(0) = t_0 \\ \alpha_{2,1} \cdot h_{2n+1}(0) + \alpha_{3,1} \cdot h_{3n+1}(0) = t_1 \\ \alpha_{2,0} \cdot h_{2n}(1) + \alpha_{3,2} \cdot h_{3n+2}(0) = t_2 \\ \alpha_{2,1} \cdot h_{2n+1}(1) + \alpha_{3,0} \cdot h_{3n}(1) = t_3 \\ \alpha_{2,1} \cdot h_{2n+1}(2) + \alpha_{3,1} \cdot h_{3n+1}(1) = t_4 \end{cases} \tag{8.31}$$

from which a value for $\alpha_{2,0}$ cannot be deduced because it appears in three equations with three other different unknown constants.

What we need is to use the Taylor expansion of order $p \in \mathbb{N}$ of $f(z)$ in such a way that there exists $q \in \mathbb{N}, q \leqslant p$, so that there are at least q linear equations with q unknowns each in the resulting linear system. The minimal value of such a p in this particular example is $2 \cdot x_2 = 3 \cdot x_3$ where x_2 and x_3 are the minimal positive integers verifying $2 \cdot x_2 = 3 \cdot x_3$, hence $x_2 = 3, x_3 = 2$ and $p = 6 = \operatorname{lcm}(2,3)$. Indeed, a Taylor expansion of order 6 gives two linear equations for $\alpha_{2,0}$ and $\alpha_{3,0}$ and this allows to find their values and deduce those of the other constants. If moreover there were a hypergeometric term in (8.28), then 6 linear equations could not be enough. In this case the minimal value for p would be $2 \cdot 6 = 12$ in order to have at least three equations for $\alpha_{2,0}$, $\alpha_{3,0}$ and the unknown constant related to the hypergeometric term.

We now move to the general case.

Let

$$\begin{aligned}\mathcal{H} &:= \Big[[1, \{h_{n,1}, h_{n,2}, \dots, h_{n,l_1}\}], \dots, [m_1, \{h_{m_1 n,1}, h_{m_1 n,2}, \dots, h_{m_1 n, l_{m_1}}\}], \dots, \\ &\qquad [m_\mu, \{h_{m_\mu n,1}, h_{m_\mu n,2}, \dots, h_{m_\mu n, l_{m_\mu}}\}] \Big] \qquad (8.32)\\ &= \left[[1, S_{1,0}], \dots, [m_1, S_{m_1,0}], \dots, [m_\mu, S_{m_\mu,0}]\right] \qquad (8.33)\end{aligned}$$

for integers $1 < m_1 < \cdots < m_\mu$ be the non-empty basis of all m-fold hypergeometric term solutions of a holonomic recurrence equation satisfied by the Taylor coefficients of $f(z)$. m_μ is the maximum symmetry number, l_m is the number of m-fold hypergeometric terms in $\mathcal{H}$ $m \in \{m_1, \dots, m_\mu\} \cup \{1\}$. The computation of $F(z)$ for $f(z)$ follows the following steps.

Algorithm 14 Computation of $F(z)$ in the representation (8.25) of a given hypergeometric type expression f

---

Input: $f(z)$, the recurrence equation, say *RE* computed by *FindRE*, the basis of all m-fold hypergeometric term solutions of *RE*, say $\mathcal{H}$, computed by mfoldHyper, $T(z)$ and N_0 computed by Algorithm 12.

Output: $F(z)$ in the representation (8.25) of f.

1. Find the other m-fold symmetric terms associated to each m-fold hypergeometric term in $\mathcal{H}$ for $m \in \{m_1, \dots, m_\mu\}$. For that purpose one calls Algorithm 11 as *mfoldHyper(RE,a[n],m,j)* for $j = 1, \dots, m-1$, $m \in \{m_1, \dots, m_\mu\}$. This allows to build the sets
$$S_m := \{S_{m,0}, S_{m,1} \dots, S_{m,m-1}\}, \qquad (8.34)$$
for $m \in \{m_1, \dots, m_\mu\}$, where
$$S_{m,j} := \{h_{mn+j,1}, h_{mn+j,2}, \dots, h_{mn+j,l_m}\}, \; 0 \leqslant j \leqslant m-1. \qquad (8.35)$$
2. Set $N_1 = \max\{0, N_0\}$ and $T_1(z) := Taylor(f(z), z, 0, N_1 - 1)$.

Algorithm 14 Computation of $F(z)$ the representation (8.25) of a given hypergeometric type expression f

3. Compute $i_{m,j} = \left\lceil \frac{N_1 - j}{m} \right\rceil$ for $j = 0, \ldots, m-1$, $m \in \{m_1, \ldots, m_\mu\}$.
4. Set
$$\mathcal{N} = N_1 + \left(l_1 + \sum_{m \in \{m_1,\ldots,m_\mu\}} l_m - 1 \right) \cdot \operatorname{lcm}(m_1, \ldots, \cdot m_\mu) \tag{8.36}$$
5. Compute $p_{m,j} = \left\lfloor \frac{\mathcal{N} - j}{m} \right\rfloor$, $j = 0, \ldots, m-1$, $m \in \{m_1, \ldots, m_\mu\}$.
6. Let $\alpha_{m,j,k} \in \mathbb{K}$, $m \in \{m_1, \ldots, m_\mu\}$, $j = 0, \ldots, m-1$, $k = 1, \ldots, l_m$ be some unknown constants and define
$$I(z) := \sum_{m \in \{m_1,\ldots,m_\mu\}} \sum_{j=0}^{m-1} \sum_{k=1}^{l_m} \alpha_{m,j,k} \sum_{n=i_{m,j}}^{p_{m,j}} h_{mn+j,k} z^{mn+j}. \tag{8.37}$$
7. Solve the linear system resulting from the equality
$$I(z) + T_1(z) - Taylor(f(z), z, 0, \mathcal{N}) = 0, \tag{8.38}$$
for the unknown $(\alpha_{m,j,k})^T_{m \in \{m_1,\ldots,m_\mu\},\ 0 \leqslant j \leqslant m-1,\ 1 \leqslant k \leqslant l_m} \in \mathbb{K}^{l_1 + \sum_{m \in \{m_1,\ldots,m_\mu\}} l_m \cdot m}$.
8. If there is no solution then stop and return FALSE. No linear combination exists in this case.
9. If there is a solution then set all parameters of dependency to 0 (if there are some). This gives the choice of the linear combination. We denote by $\alpha'_{m,j,k}$ the resulting value found for $\alpha_{m,j,k}$, $m \in \{m_1, \ldots, m_\mu\}$, $j = 0, \ldots, m-1$, $k = 1, \ldots, l_m$.
10. For each $S_m, m \in \{m_1, \ldots, m_\mu\}$ construct the term
$$S'_m := \sum_{S_{m,j} \in S_m} \left(\sum_{h_{mn+j,k} \in S_{m,j}} \alpha'_{m,j,k} h_{mn+j,k} \right) z^{mn+j-i_{m,j}} \tag{8.39}$$
$$:= \sum_{j=0}^{m-1} \left(\sum_{k=1}^{l_m} \alpha'_{m,j,k} h_{mn+j,k} \right) z^{mn+j-i_{m,j}} \tag{8.40}$$
11. For each S'_m, $m \in \{m_1, \ldots, m_\mu\}$, make evaluations for $n \in [\![N_0, N_1]\!]$ to subtract terms in $T_1(z)$ that can be computed from S'_m and shift the initial index $i_{m,j}$ accordingly. (This step could also be done before step 10 to get more suitable values for starting points).
12. Return $T_1(z) + \sum_{m \in \{m_1,\ldots,m_\mu\}} \sum_{n=0}^{\infty} S'_m$.

The correctness of this algorithm depends on whether the solution of the linear system in step 7 has enough equations to determine the possible coefficients of the linear combination sought. Indeed, as we saw in the particular example for (8.28) that we need a linear system for which each unknown has enough equations to be determined. This is established by the following lemma.

Lemma 8.2. *In Algorithm (14), $\mathcal{N}$ in (8.36) is the minimal integer for which the Taylor expansion of order $\mathcal{N}$ of $f(z)$ allows to determine the linear combination sought.*

Proof. The computation is the similar for any integer N_1, therefore we assume that $N_1 = 0$. The number of unknowns in each equation is $q = l_1 + \sum_{m\in\{m_1,\dots,m_\mu\}} l_m$. The aim is to find $\mathcal{N}$ such that $Taylor(f(z), z, 0, \mathcal{N})$ in Algorithm 7 step 7 yields a linear system with at least q equations with the same q unknowns each. Of course, the minimal value of $\mathcal{N}$ is an integer that verifies

$$\mathcal{N} = m_1 \cdot x_1 = m_2 \cdot x_2 = \cdots = m_\mu \cdot x_\mu,$$

for some positive integers $x_1, x_2, \dots, x_\mu$, since we have to find q equations that correspond to the q first coincidences of

$$z^{m_1\cdot n}, \; z^{m_2\cdot n}, \; \dots, z^{m_\mu \cdot n}.$$

The second coincidence is reached at the expansion of order $\operatorname{lcm}(m_1, \dots, m_\mu)$, therefore by induction we deduce that for any positive integer p, the p^{th} coincidence is reached at the expansion of order $(p-1)\cdot \operatorname{lcm}(m_1,\dots,m_\mu)$. Hence we finally get

$$\mathcal{N} = (q-1)\cdot \operatorname{lcm}(m_1,\dots,m_\mu) = \left(l_1 + \sum_{m\in\{m_1,\dots,m_\mu\}} l_m - 1 \right) \cdot \operatorname{lcm}(m_1,\dots,\cdot m_\mu),$$

as expected. □

Illustrations and Remark.

- Let us consider $\frac{\sin(z)}{z^5} + \exp(z)$. The m-fold hypergeometric terms are

```
(%i1) mfoldHyper(FindRE(sin(z)/z^5+exp(z),z,a[n]),a[n]);
```

$$(\%o1) \quad \left[\left[1, \left\{ \frac{1}{n!} \right\} \right], \left[2, \left\{ \frac{(-1)^n}{(n+1)\cdot(n+2)\cdot(2\cdot n+1)\cdot(2\cdot n+3)\cdot(2\cdot n+5)\cdot(2\cdot n)!} \right\} \right] \right]$$

  Further factorizations are used to get a much simpler representation of the 2-fold hypergeometric term. The computation of the linear combination starts at $N_1 = \max\{-5, 0\} = 0$ since we cannot compute factorials of negative arguments in the hypergeometric term $\frac{1}{n!}$. The corresponding $T_1(z)$ is

$$T_1(z) = \frac{1}{z^4} - \frac{1}{6\cdot z^2}.$$

  A Taylor expansion of order $\mathcal{N} = (1+1-1)\cdot 2 = 2$ is computed to determine the coefficients of the linear combination sought. Observe that this expansion allows to find the coefficients of the hypergeometric series and the even (z^{2n} powers) hypergeometric type series of type 2. The value of the coefficient of the odd hypergeometric type series of type 2 is deduced from the linear equation of two variables whose second one corresponds to the coefficient of the hypergeometric series.

- In this example the subtraction of terms from $T_1(z)$ is correctly handled. That is why our Maxima implementation yields the following "simplest" power series representation of $\frac{\sin(z)}{z^5} + \exp(z)$

```
(%i2) FPS(sin(z)/z^5+exp(z),z,n);
```

$$(\%o2) \qquad \left(\sum_{n=0}^{\infty} \frac{z^n}{n!}\right) + \sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{2\cdot(n-2)}}{(1+2\cdot n)!}$$

  Note, however, that this subtraction of terms is not always possible. For example the function $\cos(z) + \frac{\sin(z)}{z^3}$ turns out to be a hypergeometric type series of type 2 (not the sum of hypergeometric type series of type 2, but is itself of this type).

```
(%i3) FPS(cos(z)+sin(z)/z^3,z,n);
```

$$(\%o3) \qquad \left(\sum_{n=0}^{\infty} \frac{(-1)^n \cdot ((3+2\cdot n)! - (2\cdot n)!) \cdot z^{2\cdot n}}{(2\cdot n)! \cdot (3+2\cdot n)!}\right) + \frac{1}{z^2}$$

  Indeed, the obtained general coefficient is hypergeometric as the use of our Maxima function *ratio* below shows.

```
(%i4) ratio(((-1)^n*((3+2*n)!-(2*n)!))/((2*n)!*(3+2*n)!),n);
```

$$(\%o4) \qquad -\frac{59 + 94\cdot n + 48\cdot n^2 + 8\cdot n^3}{2\cdot(n+2)\cdot(2\cdot n+5)\cdot(8\cdot n^3 + 24\cdot n^2 + 22\cdot n + 5)}$$

  If one uses an algorithm that computes power series representations of each summand in the input then the result would be different. However such an approach is not recommended as it does not follow the algorithmic idea of discovering new identities and new hypergeometric terms as above.

- Algorithm 14 step 7: parameters of dependency appear when there are some m-fold hypergeometric terms represented many times as we explained with the example of $\cosh(z)$ in the previous section.

As already used many times, the command *FPS(f(z),z,n,[z_0])* of our FPS package computes the power series representation of *f(z)* at the point of expansion $z_0 \in \mathbb{C}$ (if given or 0 otherwise) with the index variable *n* by combining *FindRE*, *mfoldHyper*, *Puiseuxnbrfun* and our implementations of Algorithms 12 and 13 if the computed holonomic RE is a two-term holonomic RE or Algorithm 14, if *f(z)* is a generalized hypergeometric type function.

Example 8.2.

```
(%i5) Nmax:7$
(%i6) FPS(sin(z)^2+cos(z)^3,z,n);
```

$$(\%o6) \qquad \left(\sum_{n=0}^{\infty} -\frac{(-(-9)^n - 3\cdot(-1)^n + 2\cdot(-1)^n\cdot 4^n)\cdot z^{2\cdot n}}{4\cdot(2\cdot n)!}\right) + \frac{1}{2}$$

Note that in this example we set *Nmax* to 7 because the given expression satisfies a holonomic differential equation of order 7.

```
(%i7) FPS(sinh(z)+asinh(z),z,n);
```

$$(\%o7)\quad \left(\sum_{n=0}^{\infty} \frac{(-1)^n \cdot (2\cdot n)! \cdot z^{1+2\cdot n}}{(2\cdot n+1)\cdot 4^n \cdot n!^2}\right) + \left(\sum_{n=0}^{\infty} -\frac{z^{2\cdot n}}{(2\cdot n)!}\right) + \sum_{n=0}^{\infty} \frac{z^n}{n!}$$

```
(%i8) FPS(sin(z^(1/3))+ cos(z^(1/2)),z,n);
```

$$(\%o8)\quad \left(\sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{\frac{1+2\cdot n}{3}}}{(2\cdot n+1)\cdot (2\cdot n)!}\right) + \sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^n}{(2\cdot n)!}$$

```
(%i4) FPS(acos(z^(1/2))+exp(z^2),z,n);
```

$$(\%o9)\quad \left(\sum_{n=0}^{\infty} -\frac{(2\cdot n)! \cdot z^{\frac{1+2\cdot n}{2}}}{(2\cdot n+1)\cdot 4^n \cdot n!^2}\right) + \left(\sum_{n=0}^{\infty} \frac{z^{2\cdot n}}{n!}\right) + \frac{2+\pi}{2} - 1$$

```
(%i10)FPS(exp(asinh(z^2))+1/(1-z^(2/3)),z,n);
```

$$(\%o10)\quad \left(\sum_{n=0}^{\infty} -z^{\frac{1+2\cdot n}{3}}\right) + \left(\sum_{n=0}^{\infty} -\frac{(-1)^n \cdot (2\cdot n)! \cdot z^{4\cdot n}}{(2\cdot n-1)\cdot 4^n \cdot n!^2}\right) + \left(\sum_{n=0}^{\infty} z^{\frac{n}{3}}\right) + z^2$$

```
(%i11)FPS(log(1+sqrt(z))+atan(z^(1/3)),z,n);
```

$$(\%o11)\quad \left(\sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{2+4\cdot n}}{2\cdot n+1}\right) + \sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{3+3\cdot n}}{n+1}$$

```
(%i12)FPS(z*log(z)^2+asin(z),z,n);
```

$$(\%o12)\quad \left(\sum_{n=0}^{\infty} \frac{4^{-1-n} \cdot (2\cdot(1+n))! \cdot z^{3+2\cdot n}}{(1+n)^2 \cdot (2\cdot n+3)\cdot n!^2}\right) + z\cdot\left(1+\log(z)^2\right)$$

Observe that despite the presence of the term $z\log(z)^2$, the power series representation is still found. This is one nice advantage of using Algorithm 12.

Let us use other points of expansions.

```
(%i13)FPS(sin(2*z)+cos(z),z,n,%pi/2);
```

$$(\%o13)\quad -\sum_{n=0}^{\infty} \frac{(-1)^n \cdot (1+2\cdot 4^n)\cdot \left(z-\frac{\pi}{2}\right)^{1+2\cdot n}}{(2\cdot n+1)\cdot(2\cdot n)!}$$

```
(%i14)FPS(exp(z)+log(1+z),z,n,%e);
```

$$(\%o14)\quad \left(\sum_{n=0}^{\infty} \frac{\left(e^e + e^{1+e} + e^e\cdot n + e^{1+e}\cdot n + \frac{(-1)^n\cdot(1+n)!}{(1+e)^n}\right)\cdot (z-e)^{1+n}}{(e+1)\cdot(n+1)\cdot(1+n)!}\right) + \log(e+1) + e^e$$

This power series is wrongly represented by Maple's FPS command due to the missing of the term $\log(e+1) + e^e$.

We end our series of examples with a funny one. We use the algorithm to recover a random polynomial generated by *randompoly(z)*.

```
(%i15) FPS(randompoly(z)+exp(z),z,n);
```

$$(\%o15)\quad \left(\sum_{n=0}^{\infty} \frac{z^n}{n!}\right) - 19 \cdot z^4 + 19 \cdot z^2 + 2 \cdot z + \frac{7}{z} - \frac{26}{z^2} + \frac{7}{z^3} + \frac{1}{z^4} - \frac{15}{z^6} - \frac{23}{z^9} + \frac{4}{z^{10}}$$

Note that the current Maple FPS command cannot find the power series representations of most of the above examples.

We have now fully described our algorithm for the computation of linear combinations of generalized Laurent-Puiseux series of hypergeometric type which is the essential goal of this thesis. Next, we move to some extensions of our computations.

8.5 Asymptotic Series

With the explanations given up to now we are already able to compute convergent asymptotic series. Indeed, the computation of a power series representation of $f(z)$ at infinity is handled as the one of $f(\frac{1}{z})$ at 0 after the substitution of z by $\frac{1}{z}$ (see [Koepf, 1993, Section 7]). Let us compute the examples of this type from page 20.

```
(%i1) FPS(atan(x),x,n,inf);
```

$$(\%o1)\quad \left(\sum_{n=0}^{\infty} -\frac{(-1)^n \cdot x^{-1-2 \cdot n}}{2 \cdot n + 1}\right) + \frac{\pi}{2}$$

```
(%i2) FPS(exp(1/x),x,n,inf);
```

$$(\%o2)\quad \sum_{n=0}^{\infty} \frac{1}{n! \cdot x^n}$$

However, none of our described algorithms can be used to represent the power series of $x\exp(-x)E_i(x)$ or $\sqrt{\pi}\exp(x)(1-\operatorname{erf}(x))$ in Maxima as its command *taylor* does not have a good implementation for asymptotic expansions.

```
(%i3) f1:x*exp(-x)*expintegral_ei(x)$
(%i4) f2:sqrt(%pi)*exp(x)*(1-erf(sqrt(x)))$
(%i5) taylor(f1,x,inf,0);
```

taylor: encountered an unfamiliar singularity in:

$\text{expintegral\_ei}\,(x)$ – an error. To debug this try: debugmode(true);

```
(%i6) taylor(f2,x,inf,0);
```

taylor: encountered an unfamiliar singularity in:

$\operatorname{erf}\left(\sqrt{x}\right)$ – an error. To debug this try: debugmode(true);

```
(%i7) taylor(subst(1/x,x,f1),[x,0,0,'asymp]);
```

taylor: encountered an unfamiliar singularity in:

$$\text{expintegral\_ei}\left(\frac{1}{x}\right) \text{ – an error. To debug this try: debugmode(true);}$$

```
(%i8) taylor(subst(1/x,x,f2),[x,0,0,'asymp]);
```

taylor: encountered an unfamiliar singularity in:

$$\text{erf}\left(\frac{1}{\sqrt{x}}\right) \text{ – an error. To debug this try: debugmode(true);}$$

If we rather use the limit command, then we get the desired limits.

```
(%i9) limit(f1,x,inf);
```

$$(\%o9) \qquad 1$$

```
(%i10) limit(f2,x,inf);
```

$$(\%o10) \qquad 0$$

However, we should also be aware of some inconvenient behaviors. Indeed, the results might be useless if one computes an equivalent limit after a change of variable ($\frac{1}{x}$ by x or vice versa).

```
(%i11) limit(subst(1/x,x,f1),x,0);
```

$$(\%o11) \qquad \lim_{x\to 0} \frac{\text{expintegral\_ei}\left(\frac{1}{x}\right)\cdot e^{-\frac{1}{x}}}{x}$$

```
(%i12) limit(subst(1/x,x,f2),x,0);
```

$$(\%o12) \qquad 0$$

Note, however, that since these expressions are holonomic, we are able to compute their corresponding holonomic recurrence equations using *FindRE*. Moreover, we can also compute a basis of all m-fold hypergeometric term solutions of these using mfoldHyper. However, we do not use these algorithms as previously. We only consider specific cases in order to show that our results can be used for such computations provided good implementations of asymptotic approximations.

Assume we want to compute an asymptotic series of an expression $f(x)$. Since Maxima's command *limit* can sometimes compute asymptotic limits, we first look for an asymptotic expansion whose scale (see Definition 2.10 on p. 19) is x^{-n} with a non-zero initial coefficient. That is

$$a_0 := \lim_{x\to\infty} f(x) \neq 0. \tag{8.41}$$

Thus, if the corresponding holonomic RE computed by *FindRE* for $f\left(\frac{1}{x}\right)$ has a single hypergeometric term solution computed with *HypervanHoeij*, say $h(n)$, a Puiseux number equal to 1 and a starting point $N_0 = 0$, then the asymptotic series sought is

$$\sum_{n=0}^{\infty} a_0 \cdot h(n) x^{-n}. \tag{8.42}$$

This is how we handle asymptotic series like $x\exp(-x)E_i(x)$.

Suppose now that the initial coefficient is zero as for $\sqrt{\pi}\exp(x)(1-\text{erf}(x))$. Observe that the asymptotic sequence to be considered can be of the form $x^{-n/k}, k \in \mathbb{N}$ since $x^{0/k} = 1, \forall k \in \mathbb{N}$. We find a value for k by computing the Puiseux number using *Puiseuxnbrfun* (see Section 8.1). However, we only consider the case where $k = 2$ in order to use the limit computation only once again. The main reason why we choose the case $k = 2$ is because this turns out to be the only need to handle the representation of asymptotic series like $\sqrt{\pi}\exp(x)(1-\text{erf}(x))$ in Maxima since

```
(%i13) limit(x^(1/2)*f2,x,inf);
```

$$(\%o13) \quad 1$$

and for a given $f(x)$ this corresponds to the second coefficient

$$a_1 := \lim_{x\to\infty} \frac{f(x) - a_0}{x^{-1/2}} = \lim_{x\to\infty} x^{1/2} \cdot f(x) \tag{8.43}$$

for the asymptotic sequence $x^{-n/2}$, and the initial coefficient $a_0 = 0$. In this case, we look for 2-fold hypergeometric term solutions of the holonomic RE of $f\left(\frac{1}{x^2}\right)$ computed by *FindRE* using *mfoldHyper* as *mfoldHyper(RE,a[n],2,1)*. If there is only one solution, say $h(2n+1)$, then the asymptotic series sought is

$$\sum_{n=0}^{\infty} a_1 \cdot h(2n+1) x^{-\frac{2n+1}{2}} = \sum_{n=0}^{\infty} a_1 \cdot h(2n+1) x^{-n-\frac{1}{2}}, \tag{8.44}$$

since $a_0 = 0$.

On the other hand, we also consider the case of hypergeometric type asymptotic series that can be viewed as hypergeometric asymptotic series with only one summation whose general coefficient is a hypergeometric term. These are expressions like $f(x^m)$ where $f(x)$ has a hypergeometric series representation. We then compute a possible value for m from the holonomic recurrence equation of $f(x)$ computed by *FindRE* and substitute x by $x^{1/m}$ in the input expression to proceed as explained previously for the 1-fold or 2-fold hypergeometric case. Finally one gets the corresponding hypergeometric type series of type m after substitution of x by x^m in the resulting representation. To determine m, we use *mfoldHyper* and if there is only one m-fold hypergeometric term, then the corresponding m is the one used in the substitution. One could use the least common multiple of the different hypergeometric types, but we avoid this since we only want a minimal computation algorithm for asymptotic series expansions in Maxima.

We use this approach to depend on Maxima's command *limit* as little as possible and compute the maximum hypergeometric type asymptotic series that can be obtained by the two methods described above. Other hypergeometric type asymptotic series cannot be computed with our Maxima FPS command.

A procedure for asymptotic computation implementing the above methods is used by our Maxima FPS function when the internal command *taylor* generates an error after computing the first initial coefficient. Nevertheless, with this tricky approach we are able to recover the non-convergent asymptotic series given on page 20 and some related ones.

```
(%i14) FPS(f1,x,n,inf);
```

$$(\%o14) \quad \sum_{n=0}^{\infty} \frac{n!}{x^n}$$

```
(%i15) FPS(f2,x,n,inf);
```

$$(\%o15) \quad \sum_{n=0}^{\infty} \frac{(-1)^n \cdot (2 \cdot n)! \cdot x^{-\frac{1}{2}-n}}{4^n \cdot n!}$$

```
(%i16) FPS(x*subst(x^2,x,f2),x,n,inf);
```

$$(\%o16) \quad \sum_{n=0}^{\infty} \frac{(-1)^n \cdot (2 \cdot n)!}{4^n \cdot n! \cdot x^{2 \cdot n}}$$

```
(%i17) FPS(subst(1/x,x,f1),x,n,0);
```

$$(\%o17) \quad \sum_{n=0}^{\infty} n! \cdot x^n$$

```
(%i18) FPS((expintegral_e1(1/x)*%e^(1/x))/x,x,n,0);
```

Not computable at 0

$$(\%o18) \quad \frac{\text{expintegral\_e1}\left(\frac{1}{x}\right) \cdot e^{\frac{1}{x}}}{x}$$

For the latter example, neither the command *taylor* nor the *limit* one can compute the needed limit in Maxima. This ends our Maxima implementation for asymptotic series. We now come back to convergent power series, but the non-hypergeometric cases.

8.6 Non-Hypergeometric Type Series

In this section, we give approaches to represent the power series of expressions that are either not holonomic, or holonomic but not of hypergeometric type. Computing "simple" representations as for hypergeometric type functions for such expressions is still an open problem. The known formulas in these cases generally include other definitions to simplify their representations as for $\tan(z)$, $\sec(z)$, $\frac{1}{\exp(z)-1}$, etc., where Bernoulli and Euler numbers are used. Similarly, we will give such representations with all the needed data for evaluation.

In the first place we analyze the given expression to see whether a representation can be deduced from some particular hypergeometric type series. In this direction, we only deal with hypergeometric type series that have a single infinite summation term like those of $\sin(z)$, $\arccos(z)$, etc. Indeed, we know that generally for two given hypergeometric type expressions f and g, $\frac{1}{f}$, $f \cdot g$, $f^t, t \in \mathbb{Q}$ are not of hypergeometric type. However, there are known formulas to compute these when the power series representation of f and g have a single infinite summation term each. Therefore, if our default FPS algorithm does not find a hypergeometric type series representation then the following procedures are applied next.

8.6.1 Cauchy Product of Hypergeometric Type Series

Let f and g be two hypergeometric type series with representations

$$f(z) = \sum_{n=0}^{\infty} a_n z^{\frac{m_1 \cdot n + N_1}{k_1}} = \sum_{n=0}^{\infty} f_1(n), \; g(z) = \sum_{n=0}^{\infty} b_n z^{\frac{m_2 \cdot n + N_2}{k_2}} = \sum_{n=0}^{\infty} g_1(n). \tag{8.45}$$

Applying the Cauchy product rule for the infinite series $\sum_{n=0}^{\infty} f_1(n)$ and $\sum_{n=0}^{\infty} g_1(n)$ yields

$$f(z) \cdot g(z) = \sum_{n=0}^{\infty} \sum_{j=0}^{n} f_1(j) \cdot g_1(n-j). \tag{8.46}$$

Therefore

$$\begin{aligned} f(z) \cdot g(z) &= \sum_{n=0}^{\infty} \sum_{j=0}^{n} a_j z^{\frac{m_1 \cdot j + N_1}{k_1}} \cdot b_{n-j} z^{\frac{m_2 \cdot (n-j) + N_2}{k_2}} \\ &= \sum_{n=0}^{\infty} \sum_{j=0}^{n} a_j \cdot b_{n-j} \cdot z^{\frac{(m_1 \cdot k_2 - m_2 \cdot k_1) \cdot j + N_1 \cdot k_2 + N_2 \cdot k_1 + m_2 \cdot n}{k_1 \cdot k_2}}. \end{aligned} \tag{8.47}$$

Remark that if

$$m_1 \cdot k_2 - m_2 \cdot k_1 = 0 \tag{8.48}$$

then (8.47) reads

$$\begin{aligned} f(z) \cdot g(z) &= \sum_{n=0}^{\infty} \left(\sum_{j=0}^{n} a_j \cdot b_{n-j} \right) \cdot z^{\frac{m_2 \cdot n + N_1 \cdot k_2 + N_2 \cdot k_1}{k_1 \cdot k_2}} \\ &= \sum_{n=0}^{\infty} \left(\sum_{j=0}^{n} a_j \cdot b_{n-j} \right) \cdot z^{\frac{m \cdot n + N}{k}}, \end{aligned} \tag{8.49}$$

$m = m_2, N = N_1 \cdot k_2 + N_2 \cdot k_1, k = k_1 \cdot k_2$. The latter representation shows which possible non-zero coefficients there are and gives a formula to compute them. In our implementation, when the given non-hypergeometric type expression is the product of hypergeometric type series with representations of the form (8.45), then (8.49) is used to represent its power series provided that the condition (8.48) holds.

Note that the product of hypergeometric type functions is holonomic. Thus we can compute a holonomic RE satisfied by the Taylor coefficients of the given $f \cdot g$ and then deduce the Puiseux number $k = \mathrm{lcm}(k_1, k_2)$. This shows that in our implementation the condition (8.48) is applied to hypergeometric type expressions whose Puiseux numbers are equal to 1 since a substitution of z by z^k is made before the computations.

Let us give some examples.

Example 8.3.

```
(%i1) FPS(asin(z)*cos(z),z,n);
```

$$(\%o1) \quad \sum_{n=0}^{\infty}\left(\sum_{k=0}^{n}\frac{(2\cdot k)!\cdot(-1)^{n-k}}{(2\cdot k+1)\cdot 4^k\cdot k!^2\cdot(2\cdot(n-k))!}\right)\cdot z^{1+2\cdot n}$$

```
(%i2) FPS(atan(z)*log(1+z^2),z,n);
```

$$(\%o2) \quad \sum_{n=0}^{\infty}(-1)^n\cdot\left(\sum_{k=0}^{n}\frac{1}{(2\cdot k+1)\cdot(n-k+1)}\right)\cdot z^{3+2\cdot n}$$

```
(%i3) FPS(exp(z^(3/2))*asin(z^(3/4)),z,n);
```

$$(\%o3) \quad \sum_{n=0}^{\infty}\left(\sum_{k=0}^{n}\frac{4^{k-n}\cdot(2\cdot(n-k))!}{k!\cdot(2\cdot n-2\cdot k+1)\cdot(n-k)!^2}\right)\cdot z^{\frac{3+6\cdot n}{4}}$$

Remark Note that the holonomic recurrence equation of each of the above examples has m-fold hypergeometric term solutions that do not lead to a valid linear combination for a hypergeometric type series representation. We have observed that sometimes the recurrence equation computed for the Taylor coefficients of a given product of hypergeometric type expressions $f \cdot g$ is satisfied by the coefficient of the power series of f or the one of g. For example the case of $\arctan(z) \cdot \log(1+z^2)$ yields the 2-fold hypergeometric term

```
(%i4) RE: FindRE(atan(z)*log(1+z^2),z,a[n]);
```

$$(\%o4) \quad -3\cdot(1+n)\cdot(2+n)\cdot(3+n)\cdot(4+n)\cdot a_{n+4}-(1+n)\cdot(2+n)\cdot\left(18+27\cdot n + 5\cdot n^2\right)\cdot a_{n+2}-n\cdot\left(2+7\cdot n+18\cdot n^2+n^3\right)\cdot a_n+(n-2)^2\cdot(n-1)^2\cdot a_{n-2}=0$$

```
(%i5) mfoldHyper(RE,a[n]);
```

$$(\%o5) \quad \left[\left[2, \frac{(-1)^n}{n}\right]\right]$$

which of course could not lead to the general coefficient sought. For this example the linear system obtained in Algorithm 14 step 7 has no solution. However, observe that one could apply Koepf's extension of Zeilberger's algorithm [Koepf, 1995a] to search for a holonomic recurrence equation for the definite series

$$\sum_{k=0}^{n}\frac{1}{(2\cdot k+1)\cdot(n-k+1)}, \tag{8.50}$$

since the summand is trivially shown to be hypergeometric with respect to n and k. But given that mfoldHyper did not find any m-fold hypergeometric term solutions that represent (8.50), no holonomic RE can be found. Indeed, our algorithm proves that all the general coefficients computed in Example 8.3 do not represent m-fold hypergeometric terms.

8.6.2 Rational Powers of Hypergeometric Series

For any $t \in \mathbb{Q}$, and a hypergeometric series $f(z) = \sum_{n=0}^{\infty} a_n z^n$, $a_0 > 0$, we have the formula

$$f(z)^t = \left(\sum_{n=0}^{\infty} a_n z^n\right)^t = \sum_{n=0}^{\infty} A_n z^n, \begin{cases} A_0 = a_0^t \\ A_n = \dfrac{1}{n \cdot a_0} \sum_{j=0}^{n-1} \left[(n-j) \cdot t - j\right] a_{n-j} \cdot A_j \end{cases} . \tag{8.51}$$

For more details about this formula see [Von Holdt, 1965]. When our default FPS algorithm does not find a hypergeometric type representation of the rational power of a given expression, this formula is used if that expression can be written as $\sum_{n=0}^{\infty} a_n z^{n+N}$, where $N \in \mathbb{N}_{\geqslant 0}$ and a_n is a hypergeometric term such that $a_0 > 0$. In this case, our implementation yields a representation of the form

$$\left[\sum_{n=0}^{\infty} \frac{1}{n \cdot a_0} \sum_{j=0}^{n-1} \left[(n-j) \cdot t - j\right] a_{n-j} \cdot A_j \cdot z^{n+N} + A_0 \cdot z^N, A_0 = a_0^t\right]. \tag{8.52}$$

The hypergeometric term specification is due to the fact that the formula can only be used if the power of the indeterminate z is an integer shift of n in the representation of the given expression. Therefore we rather use Algorithm 10 to reduce the computation of solutions of corresponding holonomic REs to hypergeometric term solutions.

Example 8.4.

```
(%i6) FPS(log(1+z)^5,z,n);
```

(%o6) $$\left[\left(\sum_{n=1}^{\infty} \sum_{k=0}^{n-1} \frac{A_k \cdot (5 \cdot n - 6 \cdot k) \cdot (-1)^{n-k} \cdot z^{5+n}}{n \cdot (n-k+1)}\right) + z^5, A_0 = 1\right]$$

```
(%i7) FPS(atan(z)^(2/3),z,n);
```

(%o7) $$\left[\left(\sum_{n=1}^{\infty} \frac{\left(\sum_{k=0}^{n-1} \frac{A_k \cdot (2 \cdot n - 5 \cdot k) \cdot \cos\left(\frac{\pi \cdot (n-k)}{2}\right)}{n-k+1}\right) \cdot z^{\frac{2}{3}+n}}{3 \cdot n}\right) + z^{\frac{2}{3}}, A_0 = 1\right]$$

In this example the general coefficient of $\arctan(z)$ is computed as a hypergeometric term over $\mathbb{C}$ and Maxima's command *rectform* is used to simplify it in algebraic form.

```
(%i8) FPS(atan(z)^2,z,n);
```

(%o8) $$\sum_{n=0}^{\infty} (-1)^n \cdot \left(\sum_{k=0}^{n} \frac{1}{(2 \cdot k + 1) \cdot (2 \cdot n - 2 \cdot k + 1)}\right) \cdot z^{2+2 \cdot n}$$

Here our procedure for the Cauchy product is applied since the corresponding power is 2.

8.6.3 Holonomic Laurent-Puiseux Series

The two previous procedures have holonomic expressions as input. This even constitutes a necessary condition for the application of the two previous procedures. However, if the condition (8.48) is not satisfied or the given expression is not a rational power of a hypergeometric series, then the computed holonomic RE is used to give a recursive definition of the corresponding Taylor coefficients with some initial values. Note that this approach works for all holonomic functions because the recurrence equation and the initial values represents a normal form of their power series as we explained earlier in Section 4.3.2 on p. 58. Moreover, one could still use the algorithm presented in Section 8.1 to determine Puiseux numbers. Indeed, one could extend the explanations in Section 8.1 to the ones of holonomic functions. Furthermore using Algorithm 12 the representation of holonomic expressions that have no linear combination of hypergeometric type functions follows immediately.

Example 8.5.

```
(%i9) FPS(exp(z+z^2),z,n);
```

$$(\%o9)\qquad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+2} = \frac{2\cdot A_n + A_{n+1}}{n+2}, n >= 0, [A_0 = 1, A_1 = 1]\right]$$

```
(%i10) FPS(1+exp(sqrt(z)+z^(3/2)),z,n);
```

$$(\%o10)\qquad \left[\left(\sum_{n=0}^{\infty} A_n \cdot z^{\frac{1+n}{2}}\right) + 2, A_{n+5} = \frac{9\cdot n\cdot A_n + (6\cdot n+12)\cdot A_{n+2}}{n^2+9\cdot n+20} + \frac{(-3\cdot n^2 - 9\cdot n)\cdot A_{n+3} + (n+4)\cdot A_{n+4}}{n^2+9\cdot n+20}, n >= 0, \left[A_0 = 1, A_1 = \frac{1}{2}, A_2 = \frac{7}{6}, A_3 = \frac{25}{24}, A_4 = \frac{61}{120}\right]\right]$$

Since such a representation always exists for holonomic functions, we have also implemented the Maxima function *HoloRep(f,z,n,[z0])* only dedicated to this recursive representation of their power series. The function is available in our package and can sometimes be used together with our Maxima FPS command to get more information about the power series of a holonomic expression power series.

```
(%i11) FPS(exp(z)*(cos(z)+sin(z)),z,n);
```

$$(\%o11)\qquad \left(\sum_{n=0}^{\infty} \frac{(-1)^n\cdot 4^n\cdot z^{2+4\cdot n}}{\left(\frac{1}{4}\right)_n\cdot\left(\frac{3}{4}\right)_n\cdot(2\cdot n+1)\cdot(4\cdot n+1)\cdot 64^n\cdot(2\cdot n)!}\right) + \left(\sum_{n=0}^{\infty} \frac{2\cdot(-1)^n\cdot 4^n\cdot z^{1+4\cdot n}}{\left(\frac{1}{4}\right)_n\cdot\left(\frac{3}{4}\right)_n\cdot(4\cdot n+1)\cdot 64^n\cdot(2\cdot n)!}\right) + \sum_{n=0}^{\infty} \frac{(-1)^n\cdot 4^n\cdot z^{4\cdot n}}{\left(\frac{1}{4}\right)_n\cdot\left(\frac{3}{4}\right)_n\cdot 64^n\cdot(2\cdot n)!}$$

```
(%i12) HoloRep(exp(z)*(cos(z)+sin(z)),z,n);
```

$$(\%o12)\qquad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+2} = \frac{(2\cdot n+2)\cdot A_{n+1} - 2\cdot A_n}{n^2+3\cdot n+2}, n >= 0, [A_0 = 1, A_1 = 2]\right]$$

8.6.4 Reciprocal of a Hypergeometric Type Series

This procedure is our first approach to represent non-holonomic expressions. Indeed, the reciprocal of a holonomic function is generally not holonomic. This extension of our algorithm was motivated by the secant function $\sec(z)$ and the generating function of Bernoulli numbers $\frac{z}{\exp(z)-1}$ whose reciprocals are the following quite simple hypergeometric type series.

```
(%i13) FPS(1/sec(z),z,n);
```

$$(\%o13) \quad \sum_{n=0}^{\infty} \frac{(-1)^n \cdot z^{2\cdot n}}{(2\cdot n)!}$$

```
(%i14) FPS((exp(z)-1)/z,z,n);
```

$$(\%o14) \quad \sum_{n=0}^{\infty} \frac{z^n}{(1+n)!}$$

Let f be a hypergeometric type expression with the power series representation

$$f(z) := \sum_{n=0}^{\infty} a_n z^n. \tag{8.53}$$

We are interested in finding a representation for $\frac{1}{f}$. Let

$$g(z) = \sum_{n=0}^{\infty} A_n z^n, \tag{8.54}$$

such that $g(z) \cdot f(z) = 1$. Then we have

$$\sum_{n=0}^{\infty} a_n z^n \cdot \sum_{n=0}^{\infty} A_n z^n = 1. \tag{8.55}$$

Using the Cauchy product rule, we obtain

$$\sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} A_{n-k} a_k \right) z^n = 1. \tag{8.56}$$

which is equivalent to

$$\begin{cases} A_0 b_0 = 1, \\ \sum_{k=0}^{n} A_{n-k} a_k = 0, \; \forall n \in \mathbb{N} \end{cases}, \tag{8.57}$$

and therefore we have

$$\begin{cases} A_0 = \frac{1}{a_0}, \\ A_n = -A_0 \sum_{k=1}^{n} A_{n-k} a_k, \; \forall n \in \mathbb{N} \end{cases}. \tag{8.58}$$

Note that $a_0 \neq 0$ since it is a hypergeometric type expression whose a_0 is the first non-zero coefficient of its Taylor expansion.

Now suppose that f has the representation

$$f(z) := \sum_{n=0}^{\infty} a_n z^{n+N}. \tag{8.59}$$

We omit the Puiseux number since it is computed before the procedure. Note that any power series with non-zero general term $a_{m\cdot n} \cdot z^{m\cdot n}$ can always be seen as a power series with the general term $a_n \cdot z^n$ in such a way that $a_n = 0$, for all integers n that are not multiples of m. We have

$$\frac{1}{f(z)} = \frac{1}{z^N \cdot \left(\sum_{n=0}^{\infty} a_n z^n\right)} = \frac{z^{-N}}{\left(\sum_{n=0}^{\infty} a_n z^n\right)}, \tag{8.60}$$

therefore the power of the indeterminate in the power series representation of $\frac{1}{f}$ has to be shifted by $-N$. Furthermore, if f has the general term $a_n \cdot z^{m\cdot n}$ or $a_{m\cdot n} z^n$ then it is easy to see that the only non-zero coefficients computed for $\frac{1}{f}$ in (8.58) are those with indices of the form $m \cdot n$. This yields an approach for hypergeometric type series having a single m-fold hypergeometric term as general coefficient.

Let us now compute some examples.

Example 8.6.

```
(%i15) FPS(sec(sqrt(z)),z,n);
```

$$(\%o15) \quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_n = \sum_{k=1}^{n} -\frac{(-1)^k \cdot A_{n-k}}{(2\cdot k)!}, A_0 = 1\right]$$

```
(%i16) FPS(z/(exp(z)-1),z,n);
```

$$(\%o16) \quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_n = \sum_{k=1}^{n} -\frac{A_{n-k}}{(1+k)!}, A_0 = 1\right]$$

Observe in the latter example that setting $B_n = n!A_n$ yields the well known recurrence formula of the Bernoulli numbers. Indeed,

$$\begin{aligned}
\frac{B_n}{n!} = -\sum_{k=1}^{n} \frac{B_{n-k}}{(k+1)!(n-k)!} \quad &\Longleftrightarrow \quad \sum_{k=0}^{n} \frac{B_{n-k}}{(k+1)!(n-k)!} = 0 \\
&\Longleftrightarrow \quad \sum_{k=0}^{n} \frac{B_k}{(n-k+1)!k!} = 0 \\
&\Longleftrightarrow \quad \sum_{k=0}^{n} \frac{(n+1)!}{(n-k+1)!k!} B_k = 0, \ \textit{multiplication by } (n+1)! \\
&\Longleftrightarrow \quad \sum_{k=0}^{n} \binom{n+1}{k} B_k = 0.
\end{aligned} \tag{8.61}$$

```
(%i17) FPS(csc(z),z,n);
```

$$(\%o17) \quad \left[\sum_{n=0}^{\infty} A_n \cdot z^{2\cdot n-1}, A_n = \sum_{k=1}^{n} -\frac{(-1)^k \cdot A_{n-k}}{(1+2\cdot k)!}, A_0 = 1\right]$$

```
(%i18) FPS(tanh(z),z,n);
```

$$\text{(\%o18)} \quad \left[\sum_{n=0}^{\infty}\left(\sum_{k=0}^{n}\frac{A_k}{(1-2\cdot k+2\cdot n)!}\right)\cdot z^{1+2\cdot n}, A_k=\sum_{j=1}^{k}-\frac{A_{k-j}}{(2\cdot j)!}, A_0=1\right]$$

In this example the Cauchy product rule is applied after the computation of the reciprocal of $\cosh(z)$.

Our last procedure is rather more general and we are going to describe it in a new section.

8.7 An Approach Based on the Computation of Quadratic Differential Equations

As we defined a procedure to compute the power series representation of every holonomic expression, in this paragraph we give an approach to represent the power series of expressions that satisfy homogeneous quadratic differential equations. For a given expression f, the procedure follows the following steps:

1. compute a quadratic differential equation satisfied by f;
2. use the Cauchy product rule to convert that quadratic differential equation to a recurrence equation satisfied by the Taylor coefficients of f;
3. use the obtained recurrence equation to define a recursive formula for the power series coefficients of f.

Unlike the holonomic case where we know how to compute the Puiseux number and the starting point, in this approach the two last steps are only treated for Laurent series. However, the first step is a more general approach. Indeed, our algorithm for the first step constitutes an important result of this thesis: that is a similar algorithm than Koepf's algorithm for computing holonomic differential equations to compute quadratic DEs. Gathering all these steps together we are able to define a normal form representation of power series of this type. Let us move to the description of each step.

8.7.1 Computing Quadratic Differential Equations

Let f be an expression, Algorithm 1 searches for a holonomic differential equation for f by iteration on the order of derivatives of f. Depending on the algebraic simplification of ratios of these derivatives, the algorithm finds a holonomic DE of lowest order satisfied by f. Our idea in computing homogeneous quadratic differential equation is to define a "natural" order between the product of derivatives of f so that by iteration on this order a quadratic differential equation satisfied by f is sought.

Note that a quadratic differential equation is a DE that has at most one product of derivatives in one of its summands. However, for a good definition of the order that we are looking for, we first consider the general non-linear case.

We also mention that any product of derivatives increases the degree by 1 and therefore the degree of a non-linear differential equation is the maximum number of product among its summands plus 1. In particular, linear differential equations are of degree 1.

Let $f(z)$ be given functions. By observing the product rule when applying the derivative operator $\frac{d}{dz}$

$$\text{(a) } \frac{d}{dz}\left(f(z)^2\right) = 2 \cdot f(z) \cdot \frac{d}{dz} f(z), \quad \text{(b) } \frac{d}{dz}\left(\frac{1}{f(z)}\right) = \frac{-\frac{d}{dz} f(z)}{f(z)^2}, \tag{8.62}$$

one can assume that the maximum degree for $f(z)$ in a non-linear differential equation involving $f(z)$ and $\dfrac{d}{dz} f(z)$ is 2.

Now, differentiating the right-hand sides of (a) and (b) in (8.62) yields (c) and (d), respectively, as below.

$$\text{(c) } 2 \cdot f(z) \cdot \frac{d}{dz^2} f(z) + 2 \cdot \left(\frac{d}{dz} f(z)\right)^2, \quad \text{(d) } \frac{2 \cdot \left(\frac{d}{dz} f(z)\right)^2}{f(z)^3} - \frac{\frac{d^2}{dz^2} f(z)}{f(z)^2}. \tag{8.63}$$

Thus one can assume that the maximum degrees for $f(z)$ and $\dfrac{d}{dz} f(z)$ in a non-linear differential equation involving $f(z)$, $\dfrac{d}{dz} f(z)$ and $\dfrac{d^2}{dz^2} f(z)$ are, respectively, 3 and 2.

Using this process recursively we can state that

- $f(z)^4$, $\left(\frac{d}{dz} f(z)\right)^3$ and $\left(\frac{d^2}{dz^2} f(z)\right)^2$ are the maximum degrees for a non-linear DE of order 3;
- $f(z)^5$, $\left(\frac{d}{dz} f(z)\right)^4$, $\left(\frac{d^2}{dz^2} f(z)\right)^3$ and $\left(\frac{d^2}{dz^2} f(z)\right)^2$ are the maximum degrees for a non-linear DE of order 4;
- $\ldots$

However, since we are only interested in computing quadratic differential equations, we modify the above process by avoiding degrees that are greater than 2. Thus, we can order quadratic derivatives of f as follows

$$\begin{array}{lllll}
(1)\ 1, & & & & \\
(2)\ f, & (3)\ f^2, & & & \\
(4)\ f', & (5)\ f'f, & (6)\ f'^2, & & \\
(7)\ f'', & (8)\ f''f, & (9)\ f''f', & (10) f''^2, & \\
(11)\ f''', & (12)\ f'''f, & (13)\ f'''f', & (14)\ f'''f'', & (15) f'''^2, \\
\ldots & & & &
\end{array} \tag{8.64}$$

We define

$$\frac{d^{-1}}{dz} = f^{(-1)} = 1, \text{ and } \frac{d^0}{dz} = f^{(0)} = f. \tag{8.65}$$

Observe in (8.64) that for every derivative of order n of f, $n \in \mathbb{N}_{\geqslant 0}$, we compute the product of $f^{(n)}$ and all the derivatives of order less than or equal to n before computing the next derivative. We are going to define a derivative operator, say $\delta_{2,z}^{(k)} f$, $k \in \mathbb{N}$ whose numbers in parenthesis in (8.64) represent the orders. This operator computes the product of two derivatives of f according to the order given in (8.64).

Looking at (8.64) as an infinite lower triangular matrix reduces the definition of $\delta_{2,z}$ to the one of a bijective map ν between positive integers and the corresponding subspace of $\mathbb{N} \times \mathbb{N}$: $(i,j)_{i,j\in\mathbb{N}}, i \leqslant j$. This can be done by counting the couple (i,j) in (8.64) from up to down, from the left to the right. We obtain

$$\nu(k) = (i,j) = \begin{cases} (l,l) & \text{if } \quad N = k \\ (l+1, k-N) & \text{oherwise} \end{cases},$$
$$\text{where } l = \left\lfloor \sqrt{2k + \frac{1}{4}} - \frac{1}{2} \right\rfloor, \text{ and } N = \frac{l(l+1)}{2}. \tag{8.66}$$

It remains to define a correspondence between the couple $(i,j) = \nu(k)$, $k \in \mathbb{N}$ and the quadratic products in (8.64). This is straightforward since we have defined (8.65). We get

$$\delta_{2,z}^k(f) = \frac{d^{i-2}}{dz^{j-2}} f \cdot \frac{d^{j-2}}{dz^{i-2}}, \text{ where } (i,j) = \nu(k). \tag{8.67}$$

We implemented this operator in our package as *delta2diff(f,z,k)*. Let us use it to recover the products of derivatives in (8.64).

Example 8.7.

```
(%i1) delta2diff(F(z),z,1);
```
(%o1) 1

```
(%i2) delta2diff(F(z),z,2);
```
(%o2) $\mathrm{F}(z)$

```
(%i3) delta2diff(F(z),z,3);
```
(%o3) $\mathrm{F}(z)^2$

```
(%i4) delta2diff(F(z),z,4);
```
(%o4) $\frac{d}{d\,z} \cdot \mathrm{F}(z)$

```
(%i5) delta2diff(F(z),z,5);
```
(%o5) $\mathrm{F}(z) \cdot \left(\frac{d}{d\,z} \cdot \mathrm{F}(z)\right)$

```
(%i6) delta2diff(F(z),z,6);
```

$$(\%o6) \qquad \left(\frac{d}{d\,z}\cdot \mathrm{F}\left(z\right)\right)^{2}$$

```
(%i7) delta2diff(F(z),z,7);
```

$$(\%o7) \qquad \frac{d^{2}}{d\,z^{2}}\cdot \mathrm{F}\left(z\right)$$

```
(%i8) delta2diff(F(z),z,8);
```

$$(\%o8) \qquad \mathrm{F}\left(z\right)\cdot\left(\frac{d^{2}}{d\,z^{2}}\cdot \mathrm{F}\left(z\right)\right)$$

```
(%i9) delta2diff(F(z),z,14);
```

$$(\%o9) \qquad \left(\frac{d^{2}}{d\,z^{2}}\cdot \mathrm{F}\left(z\right)\right)\cdot\left(\frac{d^{3}}{d\,z^{3}}\cdot \mathrm{F}\left(z\right)\right)$$

Using $\delta_{2,z}$ instead of $\frac{d}{dz}$ in Algorithm 1 from p. 43 yields a procedure to compute quadratic differential equations generally of lowest order satisfied by a given expression $f(z)$. This adapted version of that algorithm is given below.

Algorithm 15 Computing a quadratic DE satisfied by an expression f

Input: An expression $f(z)$.
Output: A quadratic differential equation over $\mathbb{K}$ of least order satisfied by $f(z)$.

1. If $f = 0$ then the DE is found and we stop.
2. $f \neq 0$, compute $A_0(z) = \dfrac{\delta_{2,z}^3 f(z)}{f(z)}$,

   (1-a) if $A_0(z) \in \mathbb{K}(z)$ i.e $A_0(z) = \dfrac{P(z)}{Q(z)}$ where P and Q are polynomials, then we have found a quadratic DE satisfied by f:

   $$Q(z)F(z)^2 - P(z)F(z) = 0.$$

   (1-b) If $A_0(z) \notin \mathbb{K}(z)$, then go to 3.
3. Fix a number $QN_{max} \in \mathbb{N}$, the maximal order of the DE searched for; a suitable value is $QN_{max} := 19$ which corresponds to the maximum $\delta_{2,z}$-order for having a quadratic differential equation of forth order.

   (3-a) set $N := 2$;

   (3-b) compute $\delta_{2,z}^{N+2} f$;

   (3-c) expand the expression

   $$\delta_{2,z}^{N+2} f(z) + A_{N-1}\delta_{2,z}^{N+1} f(z) + \cdots + A_0 f(z) = \sum_{i=0}^{E} S_i,$$

   in elementary summands with $A_N, A_{N-1}, \ldots, A_0$ as unknowns. $E \geqslant N$ is the total number of summands S_i obtained after expansion.

Algorithm 15 Computing a quadratic DE satisfied by an expression f

(3-d) Proceed as in Algorithm 1 step (3-d) to determine the unknowns $A_N, A_{N-1}, \ldots, A_0$ over $\mathbb{K}$

(3-e) If (3-d) is not successful, then increment N, and go back to (3-b), until $N = QN_{max}$.

Note that an algorithm for computing non-linear differential equation can be defined similarly provided a suitable substitution of the derivative operator used.

Our package has an implementation of Algorithm 15 with the syntax *QDE(f(z),F(z),[Type])*. The argument *Type* is either *I* to allow the search for inhomogeneous quadratic DE, or *H* by default to look for homogeneous ones. We have also implemented the general homogeneous non-linear case as *NLDE(f(z),F(z))*.

Example 8.8.

```
(%i11) QDE(tan(z),F(z));
```

$$(\%o11)\quad \frac{d^2}{d\,z^2}\cdot \mathrm{F}(z) - 2\cdot \mathrm{F}(z)\cdot \left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right) = 0$$

```
(%i12) QDE(tan(z),F(z),Inhomogeneous);
```

$$(\%o12)\quad \frac{d}{d\,z}\cdot \mathrm{F}(z) - \mathrm{F}(z)^2 - 1 = 0$$

```
(%i13) QDE(sec(z),F(z));
```

$$(\%o13)\quad -\mathrm{F}(z)\cdot \left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right) + 2\cdot \left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)^2 + \mathrm{F}(z)^2 = 0$$

```
(%i14) QDE(z/(exp(z)-1),F(z));
```

$$(\%o14)\quad z\cdot \left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right) + \mathrm{F}(z)^2 + (z-1)\cdot \mathrm{F}(z) = 0$$

```
(%i15) QDE(log(1+sin(z)),F(z));
```

$$(\%o15)\quad \frac{d^3}{d\,z^3}\cdot \mathrm{F}(z) + \left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)\cdot \left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right) = 0$$

```
(%i16) QDE(sec(z)^k,F(z));
```

$$(\%o16)\quad k\cdot \mathrm{F}(z)\cdot \left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right) + (-1-k)\cdot \left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)^2 - k^2\cdot \mathrm{F}(z)^2 = 0$$

```
(%i17) QDE((z/(exp(z)-1))^k,F(z));
```

$$(\%o17)\quad k\cdot z\cdot \mathrm{F}(z)\cdot \left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right) - (1+k)\cdot z\cdot \left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)^2 - k\cdot (z-2)\cdot \mathrm{F}(z)\cdot \left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right) + k^2\cdot \mathrm{F}(z)^2 = 0$$

```
(%i18) QDE(tan(z)^k,F(z));
```

$$\text{(\%o17)}\quad k^2\cdot \mathrm{F}(z)\cdot\left(\frac{d^4}{d\,z^4}\cdot \mathrm{F}(z)\right)-2\cdot\left(2\cdot k^2-3\right)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)\cdot\left(\frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)\right)+3\cdot(k-2)\cdot(2+k)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)^2+4\cdot k^2\cdot \mathrm{F}(z)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)+4\cdot\left(5\cdot k^2-6\right)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)^2=0$$

Using *NLDE* in the latter example yields a non-linear equation that does not depend on the exponent k.

```
(%i18) NLDE(tan(z)^k,F(z));
```

$$\text{(\%o18)}\quad \mathrm{F}(z)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)\cdot\left(\frac{d^3}{d\,z^3}\cdot \mathrm{F}(z)\right)-2\cdot \mathrm{F}(z)\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)^2+\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)^2\cdot\left(\frac{d^2}{d\,z^2}\cdot \mathrm{F}(z)\right)-4\cdot \mathrm{F}(z)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)^2=0$$

Compared to *QDE*, generally *NLDE* generates differential equations of lower order but of higher degree. However, in terms of timings the code for quadratic differential equations is faster, and moreover the computed differential equations give much simpler recurrence equations than those related to the outputs of *NLDE*. On the other hand, *NLDE* will turn out to be very useful for proving identities.

Observe that unlike the holonomic case where the existence and uniqueness of a solution to the Cauchy problem is quite immediate, in the non-linear case one needs to take into account other important facts since the computed differential equations are not always explicit. However, thanks to the well known *implicit function theorem*, the existence and uniqueness can locally be stated. The following simple version of the *implicit function theorem* guaranties the correctness of locally defining normal forms by the use of some initial values and the differential equations computed using *QDE* or *NLDE*.

Theorem 8.1. *[The Implicit Function Theorem (see [Krantz and Parks, 2012])]*

Let $\mathcal{F}:\mathbb{R}^{n+1}\longrightarrow\mathbb{R}$ *be a continuously differentiable function, and let* $\mathbb{R}^{n+1}$ *have coordinates* $(x,y)=(x_1,\ldots,x_n,y)$. *Fix a point* $(\alpha,\beta)=(\alpha_1,\ldots,\alpha_n,\beta)$ *with*

$$\mathcal{F}(\alpha,\beta)=0. \tag{8.68}$$

If

$$\frac{\partial\mathcal{F}}{\partial y}(\alpha,\beta)\neq 0, \tag{8.69}$$

then there exists an open set $\mathcal{U}\in\mathcal{R}^n$ *containing* α *such that there exists a unique continuously differentiable function* $f:\mathcal{U}\longrightarrow\mathbb{R}$ *such that*

$$f(\alpha)=\beta\ \text{ and }\ \mathcal{F}(\alpha,f(\alpha))=0\ \forall\alpha\in\mathcal{U}. \tag{8.70}$$

Furthermore, for $k = 1, \ldots, n$,

$$\frac{\partial f}{\partial x_k}(x) = -\frac{\frac{\partial \mathcal{F}}{\partial x_k}(x)}{\frac{\partial \mathcal{F}}{\partial y}(x)}. \tag{8.71}$$

In most of the implicit differential equations that we compute, this theorem can be applied quite simply. Let us take for example the last differential equation in Example 8.8. We have the DE

$$\mathrm{F}(z) \cdot \left(\frac{d}{dz} \cdot \mathrm{F}(z)\right) \cdot \left(\frac{d^3}{dz^3} \cdot \mathrm{F}(z)\right) - 2 \cdot \mathrm{F}(z) \cdot \left(\frac{d^2}{dz^2} \cdot \mathrm{F}(z)\right)^2 + \left(\frac{d}{dz} \cdot \mathrm{F}(z)\right)^2 \cdot \left(\frac{d^2}{dz^2} \cdot \mathrm{F}(z)\right) - 4 \cdot \mathrm{F}(z) \cdot \left(\frac{d}{dz} \cdot \mathrm{F}(z)\right)^2 = 0,$$

therefore setting $x_1 = \mathrm{F}(z), x_2 = \frac{d}{dz}\mathrm{F}(z), x_3 = \frac{d^2}{dz^2}\mathrm{F}(z), y = \frac{d^3}{dz^3}\mathrm{F}(z)$, a corresponding $\mathcal{F}$ can be written as

$$\mathcal{F}(x_1, x_2, x_3, y) = x_1 \cdot x_2 \cdot y - 2 \cdot x_1 \cdot x_3^2 + x_2^2 \cdot x_3 - 4 \cdot x_1 \cdot x_2^2, \tag{8.72}$$

which, as a polynomial, is trivially continuously differentiable. What we need is to find a point $(\alpha, \beta) = (\alpha_1, \alpha_2, \alpha_3, \beta) \in \mathbb{R}^4$ such that (8.68) and (8.69) hold. To do so, we solve $\mathcal{F} = 0$ for y and we fix a value for β by choosing suitable arbitrary values for α_1, α_2, and α_3 to substitute x_1, x_2, and x_3, respectively.

Solving $\mathcal{F} = 0$ for y gives

$$y = \frac{4 \cdot x_1 \cdot x_2^2 - x_2^2 \cdot x_3 + 2 \cdot x_1 \cdot x_3^2}{x_1 \cdot x_2}. \tag{8.73}$$

We choose $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{4}$ and we get $\beta = \frac{5}{4}$.

We now check the condition of Theorem 8.1. We have

$$\mathcal{F}\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{5}{4}\right) = 0,$$

and

$$\frac{\partial \mathcal{F}(x_1, x_2, x_3, y)}{\partial y} \Big|_{\left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{5}{4}\right)} = x_1 \cdot x_2 \Big|_{\left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{5}{4}\right)} = \frac{1}{16} \neq 0,$$

hence the existence and uniqueness of the function f as stated in Theorem 8.1.

Finally, the classical existence and uniqueness theorem [Teschl, 2012, Theorem 2.2] can be applied to uniquely determine the solution of the given differential equation by using f for a well chosen initial value problem.

Next, we move on to the computation of recurrence equations.

8.7.2 Converting Quadratic Differential Equations to Recurrence Equations

We need a rewrite rule similar to (4.44) for every term in the expansion of a quadratic differential equation. Let $f(z)$ be a power series with representation

$$f(z) = \sum_{n=0}^{\infty} a_n z^n.$$

It suffices to find a recurrence equation term that corresponds to the quadratic differential equation term

$$z^p \cdot f(z)^{(i)} \cdot f(z)^{(j)},\ i, j, p \in \mathbb{N}_{\geqslant 0}. \tag{8.74}$$

As seen on page 52, we have

$$f(z)^{(k)} = \sum_{n=0}^{\infty} (n+1)_k \cdot a_{n+k} \cdot z^n, \forall k \in \mathbb{N}_{\geqslant},$$

therefore

$$\begin{aligned} f(z)^{(i)} \cdot f(z)^{(j)} &= \left(\sum_{n=0}^{\infty} (n+1)_i \cdot a_{n+i} \cdot z^n\right) \cdot \left(\sum_{n=0}^{\infty} (n+1)_j \cdot a_{n+j} \cdot z^n\right) \\ &= \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} (k+1)_i \cdot a_{k+i} \cdot (n-k+1)_j \cdot a_{n-k+j}\right) \cdot z^n, \end{aligned} \tag{8.75}$$

by application of the Cauchy product rule. Finally multiplying (8.75) by z^p yields the formula

$$z^p \cdot f(z)^{(i)} \cdot f(z)^{(j)} = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n-p} (k+1)_i \cdot (n-p-k+1)_j \cdot a_{k+i} \cdot a_{n-p-k+j}\right) \cdot z^n, \tag{8.76}$$

and the corresponding rewrite rule

$$z^p \cdot f(z)^{(i)} \cdot f(z)^{(j)} \longrightarrow \left(\sum_{k=0}^{n-p} (k+1)_i \cdot (n-p-k+1)_j \cdot a_{k+i} \cdot a_{n-p-k+j}\right). \tag{8.77}$$

Observe that (8.77) is a rewrite rule for the summand of the power series representation of the given expression f. When dealing with inhomogeneous DEs, the constant term has to be considered differently. This is the main reason why we prefer to work with homogeneous DEs.

Similarly to Algorithm 2, a procedure to convert quadratic differential equations to recurrence equations follows immediately. Our package contains the function *FindQRE(f,z,a[n])* as analogue of *FindRE* for the quadratic case. Let us now show some examples.

Example 8.9.

```
(%i19) FindQRE(tan(z),z,a[n]);
```

$$(\%o19) \quad (1+n) \cdot (2+n) \cdot a_{n+2} - 2 \cdot \sum_{k=0}^{n} (k+1) \cdot a_{k+1} \cdot a_{n-k} = 0$$

```
(%i20) FindQRE(z/(exp(z)-1),z,a[n]);
```

$$(\%o20) \quad \left(\sum_{k=0}^{n} a_k \cdot a_{n-k}\right) + (n-1) \cdot a_n + a_{n-1} = 0$$

```
(%i21) FindQRE(log(1+sin(z)),z,a[n]);
```

(%o21) $\left(\sum_{k=0}^{n}(k+1)\cdot(k+2)\cdot a_{k+2}\cdot(n-k+1)\cdot a_{n-k+1}\right) + (1+n)\cdot(2+n)\cdot(3+n)\cdot a_{n+3}=0$

When there are many summation terms, we try to contract them in order to get a more simplified recurrence equation. This is the case for

```
(%i22) FindQRE(exp(exp(z)-1),z,a[n]);
```

(%o22) $\sum_{k=0}^{n}(k+1)\cdot\Big(((1-k)\cdot a_{k+1}+a_{k+1}\cdot n)\cdot a_{n-k+1} + (a_{k+1}+(-k-2)\cdot a_{k+2})\cdot a_{n-k}\Big)=0$

8.7.3 A Normal Form for Non-Holonomic Power Series

As we did for the general case of holonomic power series, one can use the recurrence equation computed by *FindQRE* to define a power series representation of a given holonomic or non-holonomic expression. However, as we mentioned earlier, we do not consider Puiseux series since a method to determine Puiseux numbers from recurrence equations generated by *FindQRE* is not yet described. This could be a topic for further studies on non-holonomic Puiseux series.

The reasoning is similar to the one that we presented for holonomic expressions (see Section 4.3.2 on p. 58). For a given recurrence equation computed using *FindQRE*, we write the highest order term in terms of the others. And evaluating the recurrence equations at some integers allows to determine the necessary initial values of the representation.

Note that to get the highest order term when there are terms with symbolic sums in the recurrence equation, we remove parts corresponding to the minimum and the maximum value of the summation variable and substitute the initial conditions until we get a non-zero expression from which the highest order term can be obtained. For example, to get the highest order term of the recurrence equation of $\frac{z}{\exp(z)-1}$

$$\left(\sum_{k=0}^{n}a_k\cdot a_{n-k}\right)+(n-1)\cdot a_n+a_{n-1}=0, \tag{8.78}$$

we remove those parts of the symbolic sum corresponding to $k=0$ and $k=n$. This gives

$$\left(\sum_{k=1}^{n-1}a_k\cdot a_{n-k}\right)+2\cdot a_0\cdot a_n+(n-1)\cdot a_n+a_{n-1}=0. \tag{8.79}$$

Then we substitute the value of a_0 and write the resulting highest order term in terms of the other summands of the equation. In this example a_n is necessarily the highest order term to be used since the part of the equation that has no sum always depends on a_n after substitution of the value of a_0.

Observe that this process of determining the highest order term can quite easily be managed in the quadratic case because every term in the computed recurrence equation has at most one sum.

In the non-linear case we generally have a more complicated situation where some summands of the equation are products of Cauchy products. Our representation coming from the computation of non-linear DEs conserves the recurrence equations as it is computed and adds the necessary initial values.

Our FPS command combines all the procedures of this section as a last method to determine the power series representation of a given expression. Therefore if the given input is a very complicated one then the computation may take some time. Nevertheless, our package contains the function *QNF(f,z,n,[z0])* which represents power series using only the procedures described in this section.

Example 8.10.

```
(%i23) QNF(log(1+sin(z)),z,n);
```

$$(\%o23) \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+4} = \frac{(n+2)\cdot A_{n+2} - (2+n)\cdot(3+n)\cdot A_{n+3}}{(n+2)\cdot(n+3)\cdot(n+4)} \frac{-\sum_{k=1}^{n}(k+1)\cdot(k+2)\cdot A_{k+2}\cdot(n-k+2)\cdot A_{n-k+2}}{(n+2)\cdot(n+3)\cdot(n+4)}, n >= 0, \left[A_0 = 0, A_1 = 1, A_2 = -\frac{1}{2}, A_3 = \frac{1}{6}\right]\right]$$

```
(%i24) QNF(tan(z),z,n);
```

$$(\%o24) \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+3} = \frac{2\cdot A_{n+1} + 2\cdot\sum_{k=1}^{n}(k+1)\cdot A_{k+1}\cdot A_{n-k+1}}{(n+2)\cdot(n+3)}, n >= 0, [A_0 = 0, A_1 = 1, A_2 = 0]\right]$$

```
(%i25) QNF(1/(1+sin(z)),z,n);
```

$$(\%o25) \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+2} = \frac{5\cdot A_n + 3\cdot\sum_{k=1}^{n-1} A_k\cdot A_{n-k}}{(n+1)\cdot(n+2)}, n >= 0, [A_0 = 1, A_1 = -1]\right]$$

On the other hand, this representation gives a method for computing Taylor polynomials. We implemented it as *QTaylor(f(z),z,z0,N)*. However, due to the presence of summation terms the quadratic time complexity cannot be avoided and hence the code is generally slower than the built-in Maxima command *taylor*. Nevertheless one can use a remembering program so that many calls of close orders of the same function require a timing only for the first call. This function is available in our Maxima FPS package.

Example 8.11.

```
(%i26) taylor(tan(z),z,0,8);
```
Evaluation took 0.0000 seconds (0.0000 elapsed)

$$(\%o26)/T/ z + \frac{z^3}{3} + \frac{2 \cdot z^5}{15} + \frac{17 \cdot z^7}{315} + ...$$

```
(%i27) QTaylor(tan(z),z,0,8);
```
Evaluation took 0.1200 seconds (0.1500 elapsed)

$$(\%o27) \quad \frac{17 \cdot z^7}{315} + \frac{2 \cdot z^5}{15} + \frac{z^3}{3} + z$$

```
(%i28) taylor(log(1+sin(z)),z,0,10);
```

Evaluation took 0.0000 seconds (0.0000 elapsed)

$$(\%o28)/T/ z - \frac{z^2}{2} + \frac{z^3}{6} - \frac{z^4}{12} + \frac{z^5}{24} - \frac{z^6}{45} + \frac{61 \cdot z^7}{5040} - \frac{17 \cdot z^8}{2520} + \frac{277 \cdot z^9}{72576} - \frac{31 \cdot z^{10}}{14175} + ...$$

```
(%i29) QTaylor(log(1+sin(z)),z,0,10);
```
Evaluation took 0.4700 seconds (0.4800 elapsed)

$$(\%o29) \quad -\frac{31 \cdot z^{10}}{14175} + \frac{277 \cdot z^9}{72576} - \frac{17 \cdot z^8}{2520} + \frac{61 \cdot z^7}{5040} - \frac{z^6}{45} + \frac{z^5}{24} - \frac{z^4}{12} + \frac{z^3}{6} - \frac{z^2}{2} + z$$

```
(%i30) taylor(log(1+sin(z)),z,0,300)$
```
Evaluation took 5.3500 seconds (6.5000 elapsed)

```
(%i31) QTaylor(log(1+sin(z)),z,0,300)$
```
Evaluation took 82.1400 seconds (83.1400 elapsed)

```
(%i32) taylor(log(1+sin(z)),z,0,200)$
```
Evaluation took 1.1800 seconds (1.2000 elapsed)

```
(%i33) QTaylor(log(1+sin(z)),z,0,200)$
```
Evaluation took 0.0100 seconds (0.0200 elapsed)

Note that our computations of recurrence equations and Taylor polynomials from quadratic differential equations prove again without the use of Theorem 8.1 the existence and uniqueness of the solutions of these differential equations. Therefore, our normal forms are well defined.

8.8 The Algorithms as Simplifiers

One of the main questions in Computer Algebra is to decide whether an expression is equivalent to zero or not. Marko Petkovšek, Herbert Wilf and Doron Zeilberger wrote the book "A=B" in that direction [Petkovšek et al., 1996]. The book is about identities in general, and hypergeometric identities in particular, with emphasis on computer methods of discovery and proof. Our developments follow this same idea of making progress in science by giving algorithmic approaches that make the computer work as an expert system to prove identities. What some people could see as "genius" thoughts are gradually being transformed to methods that the computer can understand.

Note that these algorithmic developments do not only verify the identity "A=B", but also answer the question "What is A?" as it was the case in all the examples that we have presented. We generally get the output as a normal form representation of the power series of a given expression and if that output

- is a polynomial then the input expression has been simplified;
- equals to the representation of another expression, then the algorithm has proved that both expressions are identical.

As we are dealing with series expansions, we mention that the given outputs of our FPS algorithm are essentially valid in a neighborhood of the point of expansion. In this section, we will mainly consider non-holonomic expressions as the most difficult identities to be proved algorithmically belong to that family. The holonomic case is well treated in [Koepf, 2006, Chapter 9].

As first example, the expression

$$\log\left(\tan\left(\frac{x}{2}\right)+\sec\left(\frac{x}{2}\right)\right)-\operatorname{arcsinh}\left(\frac{\sin(x)}{1+\cos(x)}\right),\quad -1\leqslant x\leqslant 1 \tag{8.80}$$

from [Geddes et al., 1992, Section 3.3] (see also [Koepf, 2006, Exercise 9.8]) is known to be difficult to prove equal to zero. One needs non-trivial transformations to simplify this to zero. However, using our normal form algorithm based on the computation of quadratic differential equations yields the same power series representation for

```
(%i1) f:log(tan(z/2)+sec(z/2));
```

$$(\%o1)\qquad \log\left(\tan\left(\frac{z}{2}\right)+\sec\left(\frac{z}{2}\right)\right)$$

and

```
(%i2) g:asinh(sin(z)/(cos(z)+1));
```

$$(\%o2)\qquad \operatorname{asinh}\left(\frac{\sin(z)}{\cos(z)+1}\right)$$

as shown below.

```
(%i3) QNF(f,z,n);
```

$$(\%o3)\quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+5} = \frac{\frac{(3+n)\cdot A_{n+3}}{2} + \left(\sum_{k=1}^{n+1} (k+1)\cdot A_{k+1}\cdot(n-k+3)\cdot A_{n-k+3}\right)}{2\cdot(n+3)\cdot(n+4)\cdot(n+5)} + \frac{-4\cdot\left(\sum_{k=1}^{n+1}(k+1)\cdot(k+2)\cdot(k+3)\cdot A_{k+3}\cdot(n-k+3)\cdot A_{n-k+3}\right)}{2\cdot(n+3)\cdot(n+4)\cdot(n+5)} + \frac{8\cdot\sum_{k=1}^{n+1}(k+1)\cdot(k+2)\cdot A_{k+2}\cdot(n-k+3)\cdot(n-k+4)\cdot A_{n-k+4}}{2\cdot(n+3)\cdot(n+4)\cdot(n+5)}, n >= 0, \left[A_0 = 0, A_1 = \frac{1}{2}, A_2 = 0, A_3 = \frac{1}{48}\right]\right]$$

```
(%i4) QNF(g,z,n);
```

$$(\%o4)\quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+5} = \frac{\frac{(3+n)\cdot A_{n+3}}{2} + \left(\sum_{k=1}^{n+1} (k+1)\cdot A_{k+1}\cdot(n-k+3)\cdot A_{n-k+3}\right)}{2\cdot(n+3)\cdot(n+4)\cdot(n+5)} + \frac{-4\cdot\left(\sum_{k=1}^{n+1}(k+1)\cdot(k+2)\cdot(k+3)\cdot A_{k+3}\cdot(n-k+3)\cdot A_{n-k+3}\right)}{2\cdot(n+3)\cdot(n+4)\cdot(n+5)} + \frac{8\cdot\sum_{k=1}^{n+1}(k+1)\cdot(k+2)\cdot A_{k+2}\cdot(n-k+3)\cdot(n-k+4)\cdot A_{n-k+4}}{2\cdot(n+3)\cdot(n+4)\cdot(n+5)}, n >= 0, \left[A_0 = 0, A_1 = \frac{1}{2}, A_2 = 0, A_3 = \frac{1}{48}\right]\right]$$

Hence $f - g = \log\left(\tan\left(\frac{z}{2} + \sec\left(\frac{z}{2}\right)\right)\right) - \operatorname{arcsinh}\left(\frac{\sin(z)}{1+\cos(z)}\right) = 0$ in a neighborhood of 0. However, since our algorithm does not find a quadratic differential equation of order at most 4 satisfied by $f - g$, our implementation cannot compute a power series representation of (8.80) directly.

Let us now consider

```
(%i5) f:tanh(z);
```

$$(\%o5)\quad \tanh(z)$$

and

```
(%i6) g:exponentialize(f);
```

$$(\%o6)\quad \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Clearly, the two expressions are identical since g is obtained after application of the Euler formula to $\sinh(z)$ and $\cosh(z)$. *QNF* computes the same power series representation for these two expressions and verifies this.

```
(%i7) QNF(f,z,n);
```

$$(\%o7)\quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+3} = \frac{-2 \cdot A_{n+1} - 2 \cdot \sum_{k=1}^{n} (k+1) \cdot A_{k+1} \cdot A_{n-k+1}}{(n+2) \cdot (n+3)}, n >= 0, [A_0 = 0, A_1 = 1, A_2 = 0]\right]$$

```
(%i8) QNF(g,z,n);
```

$$(\%o8)\quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+3} = \frac{-2 \cdot A_{n+1} - 2 \cdot \sum_{k=1}^{n} (k+1) \cdot A_{k+1} \cdot A_{n-k+1}}{(n+2) \cdot (n+3)}, n >= 0, [A_0 = 0, A_1 = 1, A_2 = 0]\right]$$

Notice that $f - g$ does not generate the same quadratic differential equation without use of simplifications. There might be a higher order differential equation for $f - g$ but using the default value $QN_{\max} = 19$, *QNF* cannot find a differential equation of at most forth order.

```
(%i9) QNF(f-g,z,n);
```

(%o9) **false**

Nevertheless, using our general algorithm for computing non-linear differential equations we get the following non-quadratic DE.

```
(%i10) NLDE(f-g,F(z));
```

$$(\%o10)\quad -2 \cdot \mathrm{F}(z) \cdot \left(\frac{d^2}{d\,z^2} \cdot \mathrm{F}(z)\right) + 3 \cdot \left(\frac{d}{d\,z} \cdot \mathrm{F}(z)\right)^2 + \mathrm{F}(z)^4 - 4 \cdot \mathrm{F}(z)^2 = 0$$

Our package contains a boolean variable called *NLDEflag* whose default value is *false*. If set to *true*, then the code *QNF* will use *NLDE* instead of *QDE* in its next call. And obviously, this will be enough to prove our second zero equivalence in another approach since the above differential equation is internally computed.

```
(%i11) NLDEflag:true$
(%i12) QNF(f-g,z,n);
```

(%o12) 0

Well, to make good comparisons of our implementation with the one available in Maple for the same goal, we found it necessary to sometimes use internal simplifications as it was the case for tangent analogues of Chebyshev polynomials. Below we give a sequence of instructions to mimic detections of new identities.

Assume we have f and g as given below but with ignorance on whether they represent the same function in a certain domain.

```
(%i13) f:tan(5*atan(z))$
(%i14) g:trigsimp(trigexpand(f));
```

$$(\%o14) \quad \frac{5\cdot z-10\cdot z^3+z^5}{5\cdot z^4-10\cdot z^2+1}$$

Using *QNF* we compute representations of their power series.

```
(%i15) QNF(f,z,n);
```

$$(\%o15) \quad \left[\sum_{n=0}^{\infty} A_n\cdot z^n, A_{n+3}=\frac{50\cdot A_{n+1}-(1+n)\cdot(2+n)\cdot A_{n+1}}{(n+2)\cdot(n+3)} + \frac{10\cdot\sum_{k=1}^{n}(k+1)\cdot A_{k+1}\cdot A_{n-k+1}}{(n+2)\cdot(n+3)}, n>=0, [A_0=0, A_1=5, A_2=0]\right]$$

```
(%i16) QNF(g,z,n);
```

$$(\%o16) \quad \left[\sum_{n=0}^{\infty} A_n\cdot z^n, A_{n+8}=\frac{-49\cdot A_{n+4}+90\cdot A_{n+6}-5\cdot\left(\sum_{k=2}^{n+3} A_k\cdot A_{n-k+5}\right)}{5} + \frac{10\cdot\left(\sum_{k=2}^{n+5} A_k\cdot A_{n-k+7}\right)-\sum_{k=2}^{n+7} A_k\cdot A_{n-k+9}}{5}, n>=0\ [A_0=0, A_1=5, A_2=0, A_3=40, A_4=0, A_5=376, A_6=0, A_7=3560]\right]$$

Observing the initial values gives some interest on the progression of both expressions Taylor expansions.

```
(%i17) QTaylor(f,z,0,7);
```

$$(\%o20) \quad 3560\cdot z^7+376\cdot z^5+40\cdot z^3+5\cdot z$$

One sees that both expressions have the same Taylor coefficients up to order 7. Therefore since a representation of the power series of both expressions is handled by *QNF*, we may wish to find a representation for their difference which hopefully satisfies a quadratic differential equation of suitable order computable by use of algebraic operations. This is the case, since we get

```
(%i19) QDE(f-g,F(z));
```

$$(\%o19) \quad \left(1+z^2\right)\cdot\left(1-10\cdot z^2+5\cdot z^4\right)\cdot\left(\frac{d}{d\,z}\cdot \mathrm{F}(z)\right)-5\cdot\left(1-10\cdot z^2+5\cdot z^4\right)\cdot \mathrm{F}(z)^2 - 10\cdot z\cdot\left(5-10\cdot z^2+z^4\right)\cdot \mathrm{F}(z)=0$$

Observe that no prior knowledge on both expressions is used, otherwise the computed DE should be $F(z)=0$. The above quadratic differential equation is obtained only by algebraic computations.

Finally, we check the coincidence by computing the corresponding normal form of the power series expansion of $f-g$ which yields 0 as expected.

```
(%i18) QNF(f-g,z,n);
```

$$(\%o18) \quad 0$$

Hence $f = g$.

We finish with these non-holonomic cases by our introductory example taken from [Koepf, 2006, Section 9.1], that is

$$\frac{1+\tan(z)}{1-\tan(z)} = \exp\left(2\operatorname{arctanh}\left(\frac{\sin(2z)}{1+\cos(2z)}\right)\right). \tag{8.81}$$

This is another situation where we get compatible but not identical power series representations with same initial values.

```
(%i19) f:(1+tan(z))/(1-tan(z))$
(%i20) g:exp(2*atanh(sin(2*z)/(1+cos(2*z))))$
(%i21) QNF(f,z,n);
```

$$(\%o21) \quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+3} = \frac{4\cdot A_{n+1} + 2\cdot(n+2)\cdot A_{n+2}}{(n+2)\cdot(n+3)} + \frac{2\cdot\sum_{k=1}^{n}(k+1)\cdot A_{k+1}\cdot A_{n-k+1}}{(n+2)\cdot(n+3)}, n>=0, [A_0 = 1, A_1 = 2, A_2 = 2]\right]$$

```
(%i22) QNF(g,z,n);
```

$$(\%o22) \quad \left[\sum_{n=0}^{\infty} A_n \cdot z^n, A_{n+5} = \frac{-24\cdot A_{n+2} + 8\cdot(n+3)\cdot A_{n+3} + 6\cdot(n+3)\cdot(n+4)\cdot A_{n+4}}{(n+3)\cdot(n+4)\cdot(n+5)} + \frac{-4\cdot\left(\sum_{k=1}^{n+1}(k+1)\cdot A_{k+1}\cdot A_{n-k+2}\right) - \left(\sum_{k=1}^{n+1}(k+1)\cdot(k+2)\cdot(k+3)\cdot A_{k+3}\cdot A_{n-k+2}\right)}{(n+3)\cdot(n+4)\cdot(n+5)} + \frac{3\cdot\sum_{k=1}^{n+1}(k+1)\cdot(k+2)\cdot A_{k+2}\cdot(n-k+3)\cdot A_{n-k+3}}{(n+3)\cdot(n+4)\cdot(n+5)}, n>=0, \left[A_0 = 1, A_1 = 2, A_2 = 2, A_3 = \frac{8}{3}, A_4 = \frac{10}{3}\right]\right]$$

As we mentioned earlier the difference is proved to be zero by our algorithm.

```
(%i23) QNF(f-g,z,n);
```

$$(\%o23) \quad 0$$

This is due to the fact that $f-g$ satisfies the same quadratic differential equation computed for g.

```
(%i24) QDE(f-g,F(z));
```

$$(\%o24)\quad \mathrm{F}(z)\cdot\left(\frac{d^3}{d\,z^3}\cdot\mathrm{F}(z)\right)-3\cdot\left(\frac{d}{d\,z}\cdot\mathrm{F}(z)\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot\mathrm{F}(z)\right)+4\cdot\mathrm{F}(z)\cdot\left(\frac{d}{d\,z}\cdot\mathrm{F}(z)\right)=0$$

```
(%i25) QDE(g,F(z));
```

$$(\%o25)\quad \mathrm{F}(z)\cdot\left(\frac{d^3}{d\,z^3}\cdot\mathrm{F}(z)\right)-3\cdot\left(\frac{d}{d\,z}\cdot\mathrm{F}(z)\right)\cdot\left(\frac{d^2}{d\,z^2}\cdot\mathrm{F}(z)\right)+4\cdot\mathrm{F}(z)\cdot\left(\frac{d}{d\,z}\cdot\mathrm{F}(z)\right)=0$$

There is a theorem however which sets some boundaries about algorithmically provable identities.

Theorem 8.2 (Richardson)**.** *(see ([Geddes et al., 1992], [Petkovšek et al., 1996], [Koepf, 2006, Theorem 9.10])) Let $\mathcal{R}$ consist of the class of expressions generated by*

1. *the rational numbers and the two real number π and $\log(2)$,*
2. *the variable x,*
3. *the operations of addition, multiplication, and composition, and*
4. *the sine, exponential, and absolute value functions.*

If $E \in \mathcal{R}$, the predicate "$E = 0$" is recursively undecidable.

For hypergeometric type power series, our algorithm generates a normal form so that 0 or generally Laurent polynomials can be detected. Below we give an example of two expressions whose power series are best simplified using our FPS command and whose difference is a Laurent polynomial which can be detected by our algorithm but which cannot be easily observed otherwise.

Let

```
(%i26) f:-(((2+2^(5/4))*(2*z-2^(3/4)))
       /(2^(13/4)*(z^2-2^(3/4)*z+sqrt(2)))
       +((2^(5/4)-2)*(2^(3/4)+2*z))
       /(2^(13/4)*(z^2+2^(3/4)*z+sqrt(2)))
       -1/(2*((2*z-2^(3/4))^2/2^(3/2)+1))
       -1/(2*((2^(3/4)+2*z)^2/2^(3/2)+1)))/2;
```

$$(\%o26)\quad \frac{-\frac{\left(2+2^{\frac{5}{4}}\right)\cdot\left(2\cdot z-2^{\frac{3}{4}}\right)}{2^{\frac{13}{4}}\cdot\left(z^2-2^{\frac{3}{4}}\cdot z+\sqrt{2}\right)}-\frac{\left(2^{\frac{5}{4}}-2\right)\cdot\left(2^{\frac{3}{4}}+2\cdot z\right)}{2^{\frac{13}{4}}\cdot\left(z^2+2^{\frac{3}{4}}\cdot z+\sqrt{2}\right)}+\frac{1}{2\cdot\left(\frac{\left(2\cdot z-2^{\frac{3}{4}}\right)^2}{2^{\frac{3}{2}}}+1\right)}+\frac{1}{2\cdot\left(\frac{\left(2^{\frac{3}{4}}+2\cdot z\right)^2}{2^{\frac{3}{2}}}+1\right)}}{2}$$

and

```
(%i27) g:(1+z)/(z^5+2*z);
```

$$(\%o27) \quad \frac{1+z}{z^5+2\cdot z}$$

Using our FPS command to compute their power series yields

```
(%i28) FPS(f,z,n);
```

$$(\%o28) \quad \left(\sum_{n=0}^{\infty} -2^{-2-n}\cdot(-1)^n\cdot z^{3+4\cdot n}\right)+\sum_{n=0}^{\infty} 2^{-1-n}\cdot(-1)^n\cdot z^{4\cdot n}$$

```
(%i29) FPS(g,z,n);
```

$$(\%o29) \quad \left(\sum_{n=0}^{\infty} 2^{-1-n}\cdot(-1)^n\cdot z^{4\cdot n}\right)+\sum_{n=0}^{\infty} 2^{-1-n}\cdot(-1)^n\cdot z^{3+4\cdot(n-1)}$$

One can see a similarity, or a shift between the two obtained power series representations. Now computing the power series of $f-g$ gives

```
(%i30) FPS(f-g,z,n);
```

$$(\%o30) \quad -\frac{1}{2\cdot z}$$

which shows that a nicer expression equivalent to f (which does not look nice at all) is given by

$$f(z) = -\frac{1}{2\cdot z}+g(z) = -\frac{z^3-2}{2\cdot(z^4+2)}, \tag{8.82}$$

which is also the simplification obtained after combining Maxima commands *factor* and *ratsimp* to f.

```
(%i31) factor(ratsimp(f));
```

$$(\%o31) \quad -\frac{z^3-2}{2\cdot(z^4+2)}$$

This ends our little presentation about the scope of this thesis results and itself.

Chapter 9

Conclusion

We have developed and implemented a new algorithm (mfoldHyper) which computes a basis of all m-fold hypergeometric term solutions of a given holonomic recurrence equation. The algorithm is presented in Chapter 7 and is fundamentally based on Theorem 7.1 on p. 143 that we have established. As a linear algorithm, our Maxima implementation of mfoldHyper recovers hypergeometric terms that could not be detected using Petkovšek's algorithm [Petkovšek, 1992] presented in Chapter 5 or van Hoeij's algorithm [Van Hoeij, 1999], [Cluzeau and van Hoeij, 2006], implemented in the CAS Maple and a variant of which was presented in Chapter 6. This algorithm led us to the development and the implementation of a complete algorithm (see Chapter 8) to represent linear combinations of power series with m-fold hypergeometric term coefficients in computer algebra. Indeed, using mfoldHyper, Lemma 8.1 and 8.2 on pages 162 and 174, the algorithms of Chapter 8 extend Koepf's algorithm [Koepf, 1992] to a much general situation of functions of hypergeometric type. Consequently, our Maxima FPS command computes the power series expansions of a much larger family of expressions compared to the one of Maple. These expressions represent linear combinations over $\mathbb{K}(z)$, for a variable z and a field of characteristic zero $\mathbb{K}$ (practically used as an extension field of the field of rationals), of hypergeometric type functions.

As a byproduct of our algorithm for computing power series we have extended our computations to some particular cases related to hypergeometric type functions. We first described some methods to compute hypergeometric asymptotic series in Maxima. Secondly for two hypergeometric type expressions f and g whose power series have only one summation term, we presented how our algorithm yields representations for the power series of $\frac{1}{f}$, f^t, $t \in \mathbb{Q}$, $f \cdot g$, and $\frac{f}{g}$. The holonomic expressions that do not fit into these representations are represented by recursive formulas defined from holonomic recurrence equations satisfied by their Taylor coefficients.

Furthermore we described a similar approach than the one in [Koepf, 1992] using quadratic differential equations to represent the power series of non-holonomic functions. The recurrence equations are computed applying the Cauchy product rule to the summands of the expanded corresponding quadratic differential equations. Adding necessary initial conditions, the computed recurrence equations are used to represent power series in normal forms. By this approach we have been able to prove some zero-equivalences, which is a major topic in symbolic computations.

All our implementations are available in our Maxima package FPS. Compared to the implemented algorithms for computing power series in all computer algebra systems we realized, that all of these algorithms fail in many of the examples that we tested. We think that we have filled a gap with our algorithms and expect that they will soon be implemented in other systems.

Appendix A

The Thesis Maxima Package FPS

As presented throughout our dissertation, we have implemented many Maxima functions to achieve our goals. Below we give the available syntaxes to use the main functions of our package FPS.

- *HolonomicDE(f,F(z),[destep])*: Computes a holonomic differential equation (DE) of order at most *destep* $\cdot$ *Nmax* with respect to *F(z)* satisfied by a given expression *f* of the variable *z*. *destep* is an optional argument with default value equal to 1, which allows to search for holonomic DEs whose appearing derivatives have *destep* as the minimum difference between their orders.

- *Nmax*: is a global variable with default value equal to 6, which represents the upper bound for the order of holonomic DEs computed by *HolonomicDE*. This value can be increased by the user to search for holonomic DEs of higher order. This number also represents the maximum number of iterations to search for a holonomic DE.

- *CompatibleDE(DE1,DE2,F(z))*: verifies the compatibility of the differential equations *DE1* and *DE2* with respect to *F(z)*.

- *DEtoRE(DE,F(z),a[n])*: converts a holonomic differential equation *DE* with respect to *F(z)* into a corresponding holonomic recurrence equation (RE) with respect to *a[n]*.

- *FindRE(f,z,a[n],[destep])*: combines *HolonomicDE* and *DEtoRE* to compute a holonomic RE with respect to *a[n]* for the Taylor coefficient of a given expression *f*.

- *Taylor(f,z,z0,d)*: computes a Taylor polynomial of order *d* of a given holonomic expression *f* of the variable *z* at the point of expansion *z0*.

- *PolyPetkov(RE,a[n])*: computes a basis of all polynomial solutions of a given holonomic recurrence equation *RE* with respect to *a[n]* using Petkovšek's algorithm Poly. The output is given as a generic linear combination.

- *HyperPetkov(RE,a[n],[F])*: computes all hypergeometric term solutions over $\mathbb{Q}$ or its algebraic extensions, of a given holonomic recurrence equation *RE* with respect to *a[n]* using Petkovšek's algorithm Hyper. Setting the optional variable *F* to *C* allows computations over algebraic extension fields of $\mathbb{Q}$.

- *sumhyperRE(L,a[n])*: computes a holonomic RE with respect to *a[n]* satisfied by linear combinations of hypergeometric terms in a given list *L*.

- *HypervanHoeij(RE,a[n],[F])*: computes a basis of all hypergeometric term solutions over $\mathbb{Q}$ or its algebraic extensions, of a given holonomic recurrence equation *RE* with respect to *a[n]* using our variant of van Hoeij's algorithm.

- *mfoldHyper(RE,a[n],[F])*: computes a basis of all m-fold hypergeometric term solutions over $\mathbb{Q}$ or its algebraic extensions, of a given holonomic recurrence equation *RE* with respect to *a[n]*.

- *mfoldHyper(RE,a[n],m,j,[F])*: computes m-fold hypergeometric term solutions of the general form $h_{m\cdot n+j}, j \in [\![0, m-1]\!]$ over $\mathbb{Q}$ or its algebraic extensions, of a given holonomic recurrence equation *RE* with respect to *a[n]*.

- *LPolyPart(f,z)*: computes *[T,N]*, where *T* is an extra term generally not of hypergeometric type that can appear in the Taylor expansion of a given holonomic expression *f* of the variable *z*, and *N* is an integer that can be used as starting point if the power series of *f* contains m-fold hypergeometric term coefficients.

- *FPS(f,z,n,[z0])*: computes the power series expansion with the summation index *n* of a given expression *f* of the variable *z* at the point of development *z0*.

- *Holorep(f,z,n,[z0])*: a sub-procedure of *FPS* that is used to compute the power series of a given holonomic expression *f* whose power series seems to be not closed to a hypergeometric type representation. This command is used to represent the power series of arbitrary holonomic expression.

- *delta2diff(f,z,d)*: operator used to compute products of two derivatives of a given expression *f*. *d* denotes the $\delta_{2,z}$-order of the computed derivative.

- *deltadiff(f,z,d)*: operator used to compute products of derivatives of a given expression *f*. *d* denotes the δ_z-order of the computed derivative.

- *QDE(f,F(z),[Type])*: computes a quadratic DE satisfied by a given expression *f* of $\delta_{2,z}$-order at most *QNmax*. The optional argument *Type* is *Homogeneous* by default. One can specify *Inhomogeneous* to search for inhomogeneous quadratic DEs.

- *QNmax*: is the analogue of *Nmax* for quadratic DE with default value 21 corresponding to order 4 for the usual derivative operator.

- *NLDE*: computes a non-linear differential equation satisfied by a given expression f of δ_z-order at most *NLNmax*.
- *NLNmax*: is the analogue of *Nmax* for non-linear DE with default value 30 corresponding to some cases of order 3 for the usual derivative operator.
- *FindQRE(f,z,a[n])*: analogue of *FindRE* for the quadratic case.
- *FindNLDE(f,z,a[n])*: analogue of *FindRE* for the non-linear case.
- *QNF(f,z,n,[z0])*: analogue of *HoloRep* for non-holonomic expressions using the computation of a quadratic DE by default.
- *NLDEflag*: boolean variable with default value **false** used to allow one computation of *QNF* using the computation of a non-linear DE.
- *QTaylor(f,z,z0,d)*: analogue of *Taylor* for non-holonomic expressions.

Bibliography

[Abramov, 1989] Abramov, S. A. (1989). Problems in computer algebra that are connected with a search for polynomial solutions of linear differential and difference equations. *Moscow Univ. Comput. Math. Cybernet.*, 3:63–68.

[Abramov and Barkatou, 1998] Abramov, S. A. and Barkatou, M. A. (1998). Rational solutions of first order linear difference systems. In *ISSAC*, Volume 98, pages 124–131. Editors: Weispfenning, Volker and Trager, Barry, Association for Computing Machinery, New York.

[Abramov, 1999] Abramov, S. A. (1999). Rational solutions of linear difference and q-difference equations with polynomial coefficients. *Program. Comput. Softw.*, 25(6):369–374.

[Abramov et al., 2011] Abramov, S. A., Gheffar, A., and Khmelnov, D. (2011). Rational solutions of linear difference equations: Universal denominators and denominator bounds. *Program. Comput. Software*, 37(2):78–86.

[Boyd, 1999] Boyd, J. P. (1999). The devil's invention: asymptotic, superasymptotic and hyperasymptotic series. *Acta Appl. Math*, 56(1):1–98.

[Brewer, 2014] Brewer, T. (2014). *Algebraic Properties of Formal Power Series Composition*. PhD thesis, University of Kentucky, `https://uknowledge.uky.edu/cgi/viewcontent.cgi?article=1021&context=math_etds`.

[Casas-Alvero, 2000] Casas-Alvero, E. (2000). *Singularities of Plane Curves*. London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge.

[Cluzeau and van Hoeij, 2006] Cluzeau, T. and van Hoeij, M. (2006). Computing hypergeometric solutions of linear recurrence equations. *Appl. Algebra Engrg. Comm. Comput.*, 17(2):83–115.

[Copley, 2015] Copley, L. (2015). Asymptotic expansions. In Leslie, C., editor, *Mathematics for the Physical Sciences*, pages 150–162. Walter de Gruyter GmbH, Berlin.

[Droste and Kuich, 2009] Droste, M. and Kuich, W. (2009). *Handbook of Weighted Automata*. Springer-Verlag, Berlin Heidelberg.

[Geddes et al., 1992] Geddes, K. O., Czapor, S. R., and Labahn, G. (1992). *Algorithms for Computer Algebra*. Kluwer Academic Publishers, Massachusetts.

[Gheffar and Abramov, 2011] Gheffar, A. and Abramov, S. (2011). Valuations of rational solutions of linear difference equations at irreducible polynomials. *Adv. in Appl. Math.*, 47(2):352–364.

[Gruntz and Gonnet, 1992] Gruntz, D. and Gonnet, G. H. (1992). Limit computation in computer algebra. *ETH, Eidgenössische Technische Hochschule Zürich, Departement Informatik, Institut für Wissenschaftliches Rechnen*, Report 187.

[Gruntz, 1993] Gruntz, D. (1993). A new algorithm for computing asymptotic series. In *ISSAC*, pages 239–244. Editors: Bronstein, Manuel, Association for Computing Machinery, New York.

[Gruntz and Koepf, 1995] Gruntz, D. and Koepf, W. (1995). Maple package on formal power series. *Maple Technical Newsletter*, 2(2):22–28.

[Gruntz, 1996] Gruntz, D. (1996). *On Computing Limits in a Symbolic Manipulation System*. PhD thesis, ETH Zürich, `http://www.cybertester.com/data/gruntz.pdf`.

[Heck, 2003] Heck, A. (2003). *Introduction to Maple*. Springer-Verlag, New York, 3 edition.

[Horn et al., 2012] Horn, P., Koepf, W., and Sprenger, T. (2012). m-fold hypergeometric solutions of linear recurrence equations revisited. *Math. Comput. Sci.*, 6(1):61–77.

[Kauers and Schneider, 2010] Kauers, M. and Schneider, C. (2010). Partial denominator bounds for partial linear difference equations. In *ISSAC*, pages 211–218. Editors: Koepf, Wolfram and Watt, Stephen, Association for Computing Machinery, New York.

[Kauers and Paule, 2011] Kauers, M. and Paule, P. (2011). *The Concrete Tetrahedron. Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates*. Springer-Verlag, Wien.

[Koekoek et al., 2010] Koekoek, R., Lesky, P. A., and Swarttouw, R. F. (2010). *Hypergeometric Orthogonal Polynomials and Their q-analogues*. Springer-Verlag, Berlin, Heidelberg.

[Koepf, 1992] Koepf, W. (1992). Power series in computer algebra. *J. Symbolic. Comput.*, 13(6):581–603.

[Koepf, 1993] Koepf, W. (1993). Algorithmic development of power series. In *International Conference on Artificial Intelligence and Symbolic Mathematical Computing*, pages 195–213. Editors: Calmet, Jacques and Campbell, John A., Springer.

[Koepf, 1995a] Koepf, W. (1995a). Algorithms for m-fold hypergeometric summation. *J. Symbolic. Comput.*, 20(4):399–417.

[Koepf, 1995b] Koepf, W. (1995b). Symbolic computation of formal power series with Macsyma. In *Functional Analytic Methods In Complex Analysis And Applications To Partial Differential Equations*, pages 306–328. Editors: Tutschke, Wolfgang and Mshimba, Ali, World Scientific, Singapore, New Jersey, London, Hong Kong.

[Koepf, 1995c] Koepf, W. (1995c). Symbolic computation of sums with Macsyma. In *Functional Analytic Methods In Complex Analysis And Applications To Partial Differential Equations*, pages 277–292. Editors: Tutschke, Wolfgang and Mshimba, Ali, World Scientific, Singapore, New Jersey, London, Hong Kong.

[Koepf, 1997] Koepf, W. (1997). The algebra of holonomic equations. *Math. Semesterber.*, 44:173–194.

[Koepf and Schmersau, 1998] Koepf, W. and Schmersau, D. (1998). Representations of orthogonal polynomials. *J. Comput. Appl. Math.*, 90(1):57–94.

[Koepf, 2006] Koepf, W. (2006). *Computeralgebra: eine algorithmisch orientierte Einführung*. Springer-Verlag, Berlin Heidelberg, New York.

[Koepf, 2014] Koepf, W. (2014). *Hypergeometric Summation, An Algorithmic Approach to Summation and Special Function Identities*. Springer-Verlag, London, 2nd edition.

[Koepf and Masjed-Jamei, 2018] Koepf, W. and Masjed-Jamei, M. (2018). Some summation theorems for generalized hypergeometric functions. *Axioms*, 7(2):38.

[Krantz and Parks, 2012] Krantz, S. G. and Parks, H. R. (2012). *The Implicit Function Theorem: History, Theory, and Applications*. Birkhäuser, Boston.

[Lelong-Ferrand and Arnaudiès, 1993] Lelong-Ferrand, J. and Arnaudiès, J.-M. (1993). *Cours de mathématiques. Tome 2 : Analyse*. Dunod Université, Paris, 4th edition.

[Lubin, 1994] Lubin, J. (1994). Nonarchimedean dynamical systems. *Compos. Math.*, 94(3):321–346.

[Malham, 2005] Malham, S. J. (2005). An introduction to asymptotic analysis. Heriot-Watt University, UK. Lecture Notes, Department of Mathematics.

[Man and Wright, 1994] Man, Y.-K. and Wright, F. J. (1994). Fast polynomial dispersion computation and its application to indefinite summation. In *ISSAC*, pages 175–180. Editor: MacCallum, Malcolm, Association for Computing Machinery, New York.

[Maxima's developers, 2019] Maxima's developers (2019). Maxima, A Computer Algebra System. `http://maxima.sourceforge.net/`.

[McCarthy and Levin, 1965] McCarthy, J. and Levin, M. I. (1965). *LISP 1.5 programmer's manual*. MIT Press, Cambridge, Massachusetts, 2nd edition.

[Nowak, 2000] Nowak, K. J. (2000). Some elementary proofs of Puiseux's theorems. *Univ. Iagel. Acta Math*, 38:279–282.

[Petkovšek, 1992] Petkovšek, M. (1992). Hypergeometric solutions of linear recurrences with polynomial coefficients. *J. Symbolic. Comput.*, 14(2-3):243–264.

[Petkovšek and Salvy, 1993] Petkovšek, M. and Salvy, B. (1993). Finding all hypergeometric solutions of linear differential equations. In *ISSAC*, pages 27–33. Editor: Bronstein, Manuel, Association for Computing Machinery, New York.

[Petkovšek et al., 1996] Petkovšek, M., Wilf, H. S., and Zeilberger, D. (1996). *A=B*, Volume 30. AK Peters Ltd, Wellesley, MA.

[Richardson et al., 1996] Richardson, D., Salvy, B., Shackell, J., and Van der Hoeven, J. (1996). Asymptotic expansions of exp-log functions. In *ISSAC*, pages 309–313. Editors: Lakshman, Yagati N, Association for Computing Machinery, New York.

[Schelter, 2013] Schelter, W. (2013). Maxima manual version 5.30.

[Stanley, 1980] Stanley, R. P. (1980). Differentiably finite power series. *European J. Combin.*, 1(2):175–188.

[Stanley, 2011] Stanley, R. P. (2011). *Enumerative Combinatorics*, Volume 1. Cambridge Studies in Advanced Mathematics, Cambridge, 2nd edition.

[Stewart and Tall, 2018] Stewart, I. and Tall, D. (2018). *Complex Analysis*. Cambridge University Press, Cambridge, 2nd edition.

[Teschl, 2012] Teschl, G. (2012). *Ordinary Differential Equations and Dynamical Systems*, Volume 140. American Mathematical Society, Providence.

[Van Hoeij, 1998] Van Hoeij, M. (1998). Rational solutions of linear difference equations. In *ISSAC*, pages 120–123. Editors: Weispfenning, Volker and Trager, Barry, Association for Computing Machinery, New York.

[Van Hoeij, 1999] Van Hoeij, M. (1999). Finite singularities and hypergeometric solutions of linear recurrence equations. *J. Pure Appl. Algebra*, 139(1-3):109–131.

[Vodopivec, 2014] Vodopivec, A. (2014). Solve_rec. `https://github.com/andrejv/maxima/blob/master/share/solve_rec/solve_rec.mac`. Github, Inc.

[Von Holdt, 1965] Von Holdt, R. (1965). Rational powers of power series. *The American Mathematical Monthly*, 72(7):740–743.

[Walter, 1985] Walter, W. (1985). *Analysis I*. Springer-Verlag, Tokyo.

[Wolfram, 2003] Wolfram, S. (2003). *The Mathematica® Book, version 4*. Wolfram Media, Cambridge University Press, 5th edition.

[Zariski and Samuel, 1960] Zariski, O. and Samuel, P. (1960). *Commutative Algebra*, Volume 2. Springer-Verlag, New York.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und ohne unerlaubte Hilfe angefertigt und andere als die in der Dissertation angegebenen Hilfsmittel nicht benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht. Dritte waren an der inhaltlich-materiellen Erstellung der Dissertation nicht beteiligt; insbesondere habe ich hierfür nicht die Hilfe eines Promotionsberaters in Anspruch genommen. Kein Teil dieser Arbeit ist in einem anderen Promotions- oder Habilitationsverfahren verwendet worden.