The Holonomic Toolkit

Manuel Kauers

Abstract This is an overview over standard techniques for holonomic functions, written for readers who are new to the subject. We state the definition for holonomy in a couple of different ways, including some concrete special cases as well as a more abstract and more general version. We give a collection of standard examples and state several fundamental properties of holonomic objects. Two techniques which are most useful in applications are explained in some more detail: closure properties, which can be used to prove identities among holonomic functions, and guessing, which can be used to generate plausible conjectures for equations satisfied by a given function.

1 What is this all about?

This tutorial is an attempt to further advertize a concept which already is quite popular in some communities, but still not as popular in others. It is about the concept of holonomic functions and what computations can be done with them. The part of symbolic computation which is concerned with algorithms for transcendental functions faces a fundamental dilemma. On the one hand, problems arising from applications seem to induce a demand for algorithms that can answer questions about given analytic functions, or about given infinite sequences. On the other hand, general algorithms that take an "arbitrary" analytic functions or infinite sequences as input cannot exist, because the objects in question do in general not admit a finite representation on which an algorithm could operate. The dilemma is resolved by introducing classes of "nice" functions whose members admit a uniform finite description which can serve as data structure for algorithms.

Manuel Kauers
Research Institute for Symbolic Computation, Johannes Kepler University, A4040 Linz, Austria
e-mail: mkauers@risc.jku.at

A class being small means that strong assumptions are imposed on its elements. This makes the design of algorithms easier. A typical example for a small class is the set of all polynomial functions. They clearly admit a finite representation (for instance, the finite list of coefficients), and many questions about the elements of this class can be answered algorithmically. Implementations of these algorithms form the heart of every computer algebra system. The disadvantage of a small class is that quantities arising in applications are too often beyond the scope of the class.

Algorithms for bigger classes are more likely to be useful. An example for an extremely big class is the set of all functions y which admit a power series expansion $y = \sum_{n=0}^{\infty} a_n x^n$ whose coefficients are algebraic numbers and for which there exists an algorithm that computes for a given index n the corresponding coefficient a_n. Elements of this class clearly admit a finite representation (for instance, a piece of code implementing the algorithm for computing the nth coefficient), but hardly any interesting questions can be answered algorithmically for the functions in this class. It is for example impossible to decide algorithmically whether two representations actually represent the same function. So although the class contains virtually everything we may ever encounter in practical applications, it is not very useful.

The class of holonomic functions has proven to be a good compromise between these two extremes. On the one hand, it is small enough that algorithms could be designed for efficiently answering many important questions for a given element of the class. In particular, there are algorithms for proving identities among holonomic functions, for computing asymptotic expansions of them, and for evaluating them numerically to any desired accuracy. On the other hand, the class is big enough that it contains a lot of quantities that arise in applications. In particular, many Feynman integrals [7] and many generalized harmonic sums are holonomic. Every generalized polylogarithm and every hypergeometric term is holonomic. Every algebraic function and every quasi-polynomial is holonomic. According to Salvy [36], more than 60% of the entries of Abramowitz/Stegun's table of mathematical functions [1] are holonomic, as well as some 25% of the entries of Sloane's online encyclopedia of integer sequences (OEIS) [38].

The concept of holonomy was introduced in the 1970s by Bernstein [4] in the theory of D-modules (see Björk's book [5] for this part of the story). Its relevance to symbolic computation and the theory of special functions was first recognized by Zeilberger [43]. His 1990 article, which is still a good first reading for readers not familiar with the theory, has initiated a great amount of work both in combinatorics and in computer algebra. Stanley discusses the case of a single variable [39, 40] (see also Chapter 7 of [28]). Salvy and Zimmermann [37] and Mallinger [32] provide implementations for Maple and Mathematica, respectively. Algorithms for the case of several variables [43, 41, 16, 14] were implemented by Chyzak [14] for Maple and more recently by Koutschan [30, 31] for Mathematica. The applications in combinatorics are meanwhile too numerous to list a reasonable selection.

In this tutorial, it is not our aim to explain (or advertize) any particular software package. The goal is rather to give an overview over the various definitions of holonomy, the key properties of holonomic functions, and the most important algorithms for working with them. The text should provide any reader new to the topic with the necessary background for reading the manual of some software package without wondering how to make use of the functionality it provides. for.

2 What is a Holonomic Function?

2.1 Definitions and Basic Examples

We give several variants of the definition of holonomy, discussing the most important special cases separately, before we describe the concept in more general (and more abstract) terms.

Definition 1. An infinite sequence $(a_n)_{n=0}^{\infty}$ of numbers is called *holonomic* (or *P-finite* or *P-recursive* or, rarely, *D-finite*) if there exists an integer $r \in \mathbb{N}$, independent of n, and univariate polynomials $p_0, \dots, p_r$, not all identically zero, such that for all $n \in \mathbb{N}$ we have $p_0(n)a_n + p_1(n)a_{n+1} + \cdots + p_r(n)a_{n+r} = 0$.

Example 1. 1. The sequence $a_n = \frac{5n-3}{3n+5}$ is holonomic, because it satisfies the recurrence equation $(3n+5)(5n+2)a_n - (5n-3)(3n+8)a_{n+1} = 0$.
2. The sequence $a_n = n!$ is holonomic, because it satisfies the recurrence equation $a_{n+1} - (n+1)a_n = 0$.
3. The sequence $H_n = \sum_{k=1}^{n} \frac{1}{k}$ of harmonic numbers is holonomic, because it satisfies the recurrence equation $(n+1)H_n - (2n+3)H_{n+1} + (n+2)H_{n+2} = 0$.
4. The sequence $a_n = \sum_k \binom{n}{k}^2 \binom{n+k}{k}^2$ arising in Apery's proof [34] of the irrationality of $\zeta(3)$ is holonomic, because it satisfies the recurrence equation $(n+1)^3 a_n - (2n+3)(17n^2+51n+39)a_{n+1} + (n+2)^3 a_{n+2} = 0$.
5. The sequence $a_n = \int_0^1 \int_0^1 \frac{w^{-1-\varepsilon/2}(1-z)^{\varepsilon/2} z^{-\varepsilon/2}}{(z+w-wz)^{1-\varepsilon}} \left(1 - w^{n+1} - (1-w)^{n+1}\right) dw dz$ coming from some Feynman diagram [30, p. 94f] is holonomic (regarding ε as a fixed parameter) because it satisfies the 3rd order recurrence equation $-(\varepsilon-n-3)(\varepsilon-n-2)(\varepsilon+2n+4)(\varepsilon+2n+6)a_{n+3} + (\varepsilon-n-2)(\varepsilon+2n+4)(\varepsilon^2+2\varepsilon n+5\varepsilon-6n^2-28n-34)a_{n+2} - (n+2)(\varepsilon^3-3\varepsilon^2 n-6\varepsilon^2-8\varepsilon n^2-30\varepsilon n-28\varepsilon+12n^3+64n^2+116n+72)a_{n+1} - 2(n+1)(n+2)^2(\varepsilon-2n-2)a_n = 0$.
6. For $n \in \mathbb{N}$, define $H_n(x)$ as the (uniquely determined) polynomial of degree n with the property $\int_{-\infty}^{\infty} H_n(x)H_k(x)e^{-x^2} dx = \sqrt{\pi} 2^n n! \delta_{n,k}$ for all $n, k \in \mathbb{N}$, where $\delta_{n,k}$ is the Kronecker symbol. The $H_n(x)$ are called Hermite polynomials. Regarding them as a sequence with respect to n where x is some fixed parameter, the Hermite polynomials are holonomic, because they satisfy the recurrence equation $H_{n+2}(x) - 2xH_{n+1}(x) + (2+2n)H_n(x) = 0$.
7. The sequence $(a_n)_{n=0}^{\infty}$ defined recursively by $a_0 = 0$, $a_1 = 1$, $a_2 = 17$, and $a_{n+3} = (100+99n+97n^2)a_{n+2} + (96+95n+94n^2)a_{n+1} + (93+92n+91n^2)a_n$ has no particular significance, but it is nevertheless holonomic.
8. The sequences $a_n = \sqrt{n}$, $b_n = \mathrm{p}(n)$, $c_n = \zeta(n)$, $d_n = n^n$ where $\mathrm{p}(n)$ is the nth prime number and ζ denotes the Riemann zeta function are **not** holonomic,

i.e., none of them satisfies a linear recurrence equation with polynomial coefficients [20].

According to Definition 1, a sequence is holonomic if it can be viewed as a solution of some linear recurrence equation with polynomial coefficients. The definition for analytic functions is analogous:

Definition 2. An analytic function $y\colon U \to \mathbb{C}$ defined on some domain $U \subseteq \mathbb{C}$ (or more generally, any object y for which multiplication by polynomials, addition, and repeated differentiation is defined) is called *holonomic* (or *D-finite* or *P-finite*) if there exists an integer $r \in \mathbb{N}$ and univariate polynomials $p_0, \dots, p_r$, not all identically zero, such that $p_0(z)y(z) + p_1(z)y'(z) + \cdots + p_r(z)y^{(r)}(z) = 0$.

Example 2. 1. The function $y(z) = \frac{5z-3}{3z+5}$ is holonomic, because it satisfies the differential equation $(5z-3)(3z+5)y'(z) - 34y(z) = 0$.
2. The functions $\exp(z)$ and $\log(z)$ are holonomic, because they satisfy the differential equations $\exp'(z) - \exp(z) = 0$ and $z\log''(z) + \log'(z) = 0$, respectively.
3. The function $y(z) = 1/(1+\sqrt{1-z^2})$ is holonomic, because it satisfies the differential equation $(z^3-z)y''(z) + (4z^2-3)y'(z) + 2zy(z) = 0$.
4. The function $y(z) = \sum_{n=0}^{\infty} a_n z^n$ where a_n is the sequence from Example 1.4 is holonomic because it satisfies the differential equation $(z^2-34z+1)z^2y'''(z) + 3(2z^2-51z+1)zy''(z) + (7z^2-112z+1)y'(z) + (z-5)y(z) = 0$.
5. The function $y\colon [-1,1] \to \mathbb{R}$ uniquely determined by the conditions $y(0) = 0$, $y'(0) = 17$, $y''(z) = (100+99z+98z^2)y'(z) + (97+96z+95z^2)y(z)$ has no particular significance, but is nevertheless holonomic.
6. The functions $\exp(\exp(z))$, $1/(1+\exp(z))$, $\zeta(z)$, $\Gamma(z)$, $W(z)$ (the Lambert W function [17]) are **not** holonomic, i.e., neither of them satisfies a linear differential equation with polynomial coefficients.

Note that as introduced in the two definitions above, the word "holonomic" is ambiguous. We need to distinguish between discrete variables and continuous variables. If a function depends on a discrete variable (typically named n, m, or k), then it is called holonomic if it satisfies a recurrence, and if it depends on a continuous variable (typically named x, t, or z), then it is called holonomic if it satisfies a differential equation. For example, the Gamma function is holonomic if we regard its argument as a discrete variable, but it is not holonomic if we regard its argument as a continuous variable. For a connection of the two notions, see Theorem 1 below.

The definition for the differential case extends as follows to functions in several variables.

Definition 3. An analytic function $y\colon U \to \mathbb{C}$, defined on some domain $U \subseteq \mathbb{C}^q$ ($q \in \mathbb{N}$ fixed) is called *holonomic* (or *D-finite* or *P-finite*) if for every variable z_i ($i = 1, \dots, q$) there exists an integer $r \in \mathbb{N}$ and polynomials $p_0, \dots, p_r$, possibly depending on all q variables $z_1, \dots, z_q$ but not all identically zero, such that for all $z = (z_1, \dots, z_q) \in U$ we have $p_0(z)y(z) + p_1(z)\frac{\partial}{\partial z_i}y(z) + \cdots + p_r(z)\frac{\partial^r}{\partial z_i^r}y(z) = 0$.

In other words, a multivariate analytic function is holonomic if it can be viewed as a solution of a system of q linear differential equations with polynomial. The polynomial coefficients may involve all the variables, but there is a restriction on the derivatives: the ith equation ($i = 1,\dots,q$) may only contain differentiations with respect to the variable z_i. Again, in addition to analytic functions the definition extends more generally to any objects y for which multiplication by polynomials, addition, and repeated partial differentiation is defined.

For sequences with several indices, and more generally for functions depending on some discrete as well as some continuous variables, several different extensions of Definition 1 are in use. We give here two of them. Assigning the words "holonomic" and "D-finite" to these two properties seems to be in accordance with most of the recent literature. However, it should be observed that other authors use slightly different definitions.

Definition 4. Let $U \subseteq \mathbb{C}^q$ be a domain, and let

$$y = y(n_1,\dots,n_p,z_1,\dots,z_q)\colon \mathbb{N}^p \times U \to \mathbb{C}$$

be a function which is analytic in $z_1,\dots,z_q$ for every fixed choice of $n_1,\dots,n_p \in \mathbb{N}^p$.

1. y is called *D-finite* (or *P-finite*) if for every i ($i = 1,\dots,p$) there exists a number $r \in \mathbb{N}$ and polynomials $u_0,\dots,u_r$, possibly depending on $n_1,\dots,n_p,z_1,\dots,z_q$ and not all identically zero, such that for all $n = (n_1,\dots,n_p) \in \mathbb{N}^p$ and all $z = (z_1,\dots,z_q) \in U$ we have

$$\begin{aligned} &u_0(n,z)y(n_1,\dots,n_{i-1},n_i,n_{i+1},\dots,n_p,z)\\ &+u_1(n,z)y(n_1,\dots,n_{i-1},n_i+1,n_{i+1},\dots,n_p,z)\\ &+u_2(n,z)y(n_1,\dots,n_{i-1},n_i+2,n_{i+1},\dots,n_p,z)\\ &\qquad\vdots\\ &+u_r(n,z)y(n_1,\dots,n_{i-1},n_i+r,n_{i+1},\dots,n_p,z) = 0, \end{aligned}$$

   and for every j ($j = 1,\dots,q$) there exists a number $r \in \mathbb{N}$ and polynomials $u_0,\dots,u_r$, possibly depending on $n_1,\dots,n_p,z_1,\dots,z_q$ and not all identically zero, such that for all $n = (n_1,\dots,n_p) \in \mathbb{N}^p$ and all $z = (z_1,\dots,z_q) \in U$ we have

$$u_0(n,z)y(n,z) + u_1(n,z)\frac{\partial}{\partial z_i}y(n,z) + \cdots + u_r(n,z)\frac{\partial^r}{\partial z_i^r}y(n,z) = 0.$$

2. y is called *holonomic* if the (formal) power series

$$\tilde{y}(x_1,\dots,x_p,z_1,\dots,z_q) := \sum_{n_1=1}^{\infty}\sum_{n_2=1}^{\infty}\cdots\sum_{n_p=1}^{\infty} y(n_1,\dots,n_p,z_1,\dots,z_q)x_1^{n_1}x_2^{n_2}\cdots x_p^{n_p}$$

   is holonomic as function of $x_1,\dots,x_p,z_1,\dots,z_q$ in the sense of Definition 3.

Example 3. 1. The bivariate function $f(x,y) = \frac{1}{1-y-xy}$ is D-finite and holonomic because it satisfies the differential equations $(1-y-xy)\frac{\partial}{\partial x}f(x,y) - yf(x,y) = 0$ and $(1-y-xy)\frac{\partial}{\partial y}f(x,y) - (x+1)f(x,y) = 0$.

2. The bivariate sequence $a_{n,k} = \binom{n}{k}$ is D-finite, because it satisfies the recurrence equations $(n-k+1)a_{n+1,k} - (n+1)a_{n,k} = 0$ and $(n-k)a_{n,k+1} - (k+1)a_{n,k} = 0$, and it is holonomic because $f(x,y) = \sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\binom{n}{k}x^k y^n = \frac{1}{1-y-xy}$ is holonomic.
3. Regarded as a function of one discrete variable n and a continuous variable x, the Hermite polynomials $H_n(x)$ are D-finite because they satisfy the equations $H_{n+2}(x) - 2xH_{n+1}(x) + 2(n+1)H_n(x) = 0$ and $H_n''(x) - 2xH_n'(x) + 2nH_n(x) = 0$. They are also holonomic because the formal power series $f(x,z) := \sum_{n=0}^{\infty} H_n(x)z^n$ satisfies the differential equations $z\frac{\partial^3}{\partial x^3}f(x,z) + (1-4xz)\frac{\partial^2}{\partial x^2}f(x,z) + (4x^2z - 4z - 2x)\frac{\partial}{\partial x}f(x,z) + 4xzf(x,z) = 0$ and $2z^3\frac{\partial^3}{\partial z^3}f(x,z) + (14z^2 - 2xz + 1)\frac{\partial^2}{\partial z^2}f(x,z) + (20z - 4x)\frac{\partial}{\partial z}f(x,z) + 4f(x,z) = 0$.
4. The integrand $\frac{w^{-1-\varepsilon/2}(1-z)^{\varepsilon/2}z^{-\varepsilon/2}}{(z+w-wz)^{1-\varepsilon}}\left(1 - w^{n+1} - (1-w)^{n+1}\right)$ of the Feynman integral in Example 1.5 is D-finite when w and z are regarded as continuous and n and ε are regarded as discrete variables.
5. The Kronecker symbol $\delta_{n,k}$, viewed as a bivariate sequence in n and m, is holonomic but not D-finite. The bivariate sequence $1/(n^2 + k^2)$ is D-finite but not holonomic [41].
6. The bivariate sequence $S_1(n,k)$ of Stirling numbers of the first kind is not D-finite although it satisfies the recurrence equation $S_1(n+1,k+1) + nS_1(n,k+1) - S_1(n,k) = 0$. This recurrence equation does not suffice to establish D-finiteness because it involves shifts in both variables. It can be shown that $S_1(n,k)$ does not satisfy any recurrence equations containing only shifts in n or only shifts in k.

Although the two properties in Definition 4 are not equivalent, the difference does not play a big role in practice: multivariate functions arising in applications typically either have both properties or none of the two.

It is sometimes more transparent to work with operators acting on functions rather than with functional equations. In order to rephrase the previous definition using operators, consider the algebra

$$\mathbb{A} := \mathbb{C}(n_1,\dots,n_p,z_1,\dots,z_q)[S_1,\dots,S_p,D_1,\dots,D_q]$$

consisting of all the multivariate polynomials in the variables $S_1,\dots,S_p$, $D_1,\dots,D_q$ with coefficients that are rational functions (i.e., quotients of polynomials) in the variables $n_1,\dots,n_p,z_1,\dots,z_q$. We regard the elements of $\mathbb{A}$ as operators and let them act in the natural way on functions y: application of S_i corresponds to a shift $n_i \rightsquigarrow n_i + 1$, application of D_i causes a partial derivation $\frac{\partial}{\partial z_i}$, and application of some rational function u maps y to the function uy. We write $A \cdot y$ for the result obtained by applying an operator $A \in \mathbb{A}$ to the function y.

If A,B are operators and y is a function, then $(A+B)\cdot y = A\cdot y + B\cdot y$ (the $+$ on the left hand side being the addition in $\mathbb{A}$, and the $+$ on the right hand side being the pointwise addition of functions). For the product of two operators, we want to have $(AB)\cdot y = A\cdot(B\cdot y)$, i.e., multiplication of operators should be compatible with composition of application. This is not the case for the usual multiplication, but it works if we use a noncommutative multiplication which is such that for a rational function $u(n_1,\dots,n_p,z_1,\dots,z_q)$ we have

$$S_i\,u(n_1,\dots,n_p,z_1,\dots,z_q) = u(n_1,\dots,n_{i-1},n_i+1,n_{i+1},\dots,n_p,z_1,\dots,z_q)\,S_i$$

for every i ($i=1,\dots,p$), and

$$\begin{aligned} D_j\,u(n_1,\dots,n_p,z_1,\dots,z_q) = u(n_1,\dots,n_p,z_1,\dots,z_q)\,D_j \\ + \frac{\partial}{\partial z_i}u(n_1,\dots,n_p,z_1,\dots,z_q) \end{aligned}$$

for every j ($j=1,\dots,q$). These rules for example imply $S_in_i = (n_i+1)S_i$ and $D_jz_j = z_jD_j+1$ for $i=1,\dots,p$ and $j=1,\dots,q$. By furthermore requiring that $S_iS_j = S_jS_i$ and $S_iD_j = D_jS_i$ and $D_iD_j = D_jD_i$ for all i and j, the multiplication is uniquely determined. With this multiplication, we have for example

$$\begin{aligned} D^2\,a(z) = D(D\,a(z)) &= D(a(z)D + a'(z)) \\ &= D\,a(z)D + Da'(z) \\ &= (a(z)D + a'(z))D + (a'(z)D + a''(z)) \\ &= a(z)D^2 + 2a'(z)D + a''(z). \end{aligned}$$

In terms of operators, part 1 of Definition 4 can be stated as follows: Let y be a function as in Def. 4, and let $\mathfrak{a} \subseteq \mathbb{A}$ be the set of all operators which map y to the zero function. Then y is called D-finite if

- For all $i=1,\dots,p$ we have $\mathfrak{a}\cap\mathbb{C}(n_1,\dots,n_p,z_1,\dots,z_q)[S_i] \neq \{0\}$, and
- For all $j=1,\dots,q$ we have $\mathfrak{a}\cap\mathbb{C}(n_1,\dots,n_p,z_1,\dots,z_q)[D_j] \neq \{0\}$.

The set $\mathfrak{a}$ is called the *annihilator* of y. It has the algebraic structure of an ideal of $\mathbb{A}$, i.e., it has the properties $A,B\in\mathfrak{a} \Rightarrow A+B\in\mathfrak{a}$ and $A\in\mathfrak{a}, B\in\mathbb{A} \Rightarrow BA\in\mathfrak{a}$.

Operator algebras can be used to abstract away the difference between shift and derivation, and to allow other operations as well. We will not use this most general form in the remainder of this tutorial, but only quote the definition of D-finiteness in this language. Let R be a commutative ring (for example, the set $\mathbb{C}(x_1,\dots,x_m)$ of all rational functions in m variables $x_1,\dots,x_m$ with coefficients in $\mathbb{C}$), and consider the algebra $\mathbb{A} = R[\partial_1,\dots,\partial_n]$ of multivariate polynomials in the indeterminates $\partial_1,\dots,\partial_n$ with coefficients in R. Let $\sigma_1,\dots,\sigma_n\colon R\to R$ be automorphisms (i.e., $\sigma_i(a+b) = \sigma_i(a)+\sigma_i(b)$ and $\sigma_i(ab) = \sigma_i(a)\sigma_i(b)$ for all $a,b\in R$) and for each i, let $\delta_i\colon R\to R$ be a so-called skew-derivation for σ_i. A skew-derivation is a map which satisfies $\delta_i(a+b) = \delta_i(a)+\delta_i(b)$ and the generalized Leibniz law $\delta_i(ab) = \delta_i(a)b + \sigma_i(a)\delta_i(b)$. Then consider the (noncommutative) multiplication on $\mathbb{A}$ defined through the properties $ab = ba$ for all $a\in R$, $\partial_i\partial_j = \partial_j\partial_i$ for all $i,j=1,\dots,n$ and $\partial_ia = \sigma_i(a)\partial_i + \delta_i(a)$ for all $a\in R$ and all $i=1,\dots,n$. Such an algebra $\mathbb{A}$ is called an Ore algebra. Details about arithmetic for such algebras are explained in a nice tutorial by Bronstein and Petkovšek [10].

Observe that the generators ∂_i of an Ore algebra can be used to represent shift operators (by choosing σ_i such that $\sigma_i(x) = x+1$ for a variable x in R and δ_i the zero function) as well as derivations (by choosing σ_i the identity function and $\delta_i = \frac{\partial}{\partial x}$ for a variable x in R). In addition, further operations can be encoded, for example the q-shift (set $\sigma_i(x) := qx$ where x is a variable and q some fixed element of R).

We let the elements of an Ore algebra act ("operate") on the elements of some set F of "functions". To make this action precise, we need to assume that F is an R-module (i.e., there is an addition in F and a multiplication of elements in R by elements in F which is compatible with the addition), and that there are functions $d_1,\dots,d_n\colon F \to F$ ("partial pseudo-derivations") which satisfy certain compatibility conditions with the addition, the multiplication, and the σ_i and δ_i so as to ensure that the action of some $A \in \mathbb{A}$ on some $y \in F$, written $A \cdot y$, has the properties $(A+B)\cdot y = (A\cdot y)+(B\cdot y)$ and $(AB)\cdot y = A\cdot(B\cdot y)$ for all $A,B \in \mathbb{A}$, $a\cdot y = ay$ for all $a \in R \subseteq \mathbb{A}$, and $\partial_i \cdot y = d_i(y)$ for all i. As an example, if $\mathbb{A}$ is a ring of differential operators, a natural choice for F would be the set of all meromorphic functions, and if $\mathbb{A}$ is a ring of shift operators, a natural choice for F may be some vector space of sequences.

Definition 5. Let $\mathbb{A} = R[\partial_1,\dots,\partial_n]$ be an Ore algebra whose elements act on some set F as described above, and let $y \in F$. Let $\mathfrak{a} := \{A \in \mathbb{A} : A\cdot y = 0\}$ be the set of all operators which map y to the zero element of F. Then y is called ∂-finite if for all $i = 1,\dots,n$ we have $\mathfrak{a} \cap R[\partial_i] \neq \{0\}$.

Example 4. 1. Set $R = \mathbb{C}(n_1,\dots,n_p,z_1,\dots,z_q)$ and consider the Ore algebra $\mathbb{A} = R[\partial_1,\dots,\partial_p,\partial_{p+1},\dots,\partial_{p+q}]$ defined by the automorphisms $\sigma_1,\dots,\sigma_{p+q}\colon R \to R$, and the skew-derivations $\delta_1,\dots,\delta_{p+q}\colon R \to R$ satisfying $\sigma_i(c) = c$ and $\delta_i(c) = 0$ for $i = 1,\dots,p+q$ and all $c \in \mathbb{C}$, and

$$\begin{aligned} &\sigma_i(n_i) = n_i+1, \quad \sigma_i(n_j) = n_j \quad (i \neq j), && \sigma_i(z_j) = z_j \quad (j = 1,\dots,q)\\ &\delta_i(n_j) = 0 \quad (j = 1,\dots,p), && \delta_i(z_j) = 0 \quad (j = 1,\dots,q) \end{aligned}$$

for $i = 1,\dots,p$, and

$$\begin{aligned} &\sigma_i(n_j) = n_j \quad (j = 1,\dots,p), && \sigma_i(z_j) = z_j \quad (j = 1,\dots,q)\\ &\delta_i(n_j) = 0 \quad (j = 1,\dots,p), && \delta_i(z_{i-p}) = 1, \quad \delta_i(z_{j-p}) = 0 \quad (i \neq j) \end{aligned}$$

for $i = p+1,\dots,p+q$. Then $\partial_1,\dots,\partial_p$ act as shift operators for the variables $n_1,\dots,n_p$, respectively, and $\partial_{p+1},\dots,\partial_{p+q}$ act as derivations for the variables $z_1,\dots,z_q$, respectively.

For this choice of $\mathbb{A}$, Definition 5 reduces to part 1 of Definition 4.

2. Let $R = \mathbb{Q}(q,Q)$ and define $\sigma\colon R \to R$ by $\sigma(c) = c$ for all $c \in \mathbb{Q}(q)$ and $\sigma(Q) = qQ$, so that σ acts on Q like the shift $n \rightsquigarrow n+1$ acts on q^n. Consider the Ore algebra $\mathbb{A} = R[\partial]$ with $\delta = 0$ and this σ.

Let F denote the vector space of all sequences over $\mathbb{Q}(q)$ and let $\mathbb{A}$ act on F by $\partial\cdot(a_n)_{n=0}^\infty := (a_{n+1})_{n=0}^\infty$ and $r(q,Q)\cdot(a_n)_{n=0}^\infty := (r(q,q^n)a_n)_{n=0}^\infty$ for $r(q,Q) \in \mathbb{Q}(q,Q)$ and $(a_n)_{n=0}^\infty \in F$.

Consider the sequence $a_n := \prod_{k=1}^{n} \frac{1-q^k}{1-q}$, which is known as q-analog of the factorial in the literature [2, Chapter 10]. Because of

$$\big((1-qQ)-(1-Q)\partial\big)\cdot a_n = (1-q^{n+1})a_n - (1-q^n)a_{n+1} = 0$$

it is ∂-finite with respect to the algebra $\mathbb{A}$.

Most of the algorithms and features explained below for the shift and/or differential case generalize to objects that are D-finite with respect to arbitrary Ore algebras $\mathbb{A}$. Even more, it has recently been observed [15] that for some of the properties a weaker assumption than D-finiteness is sufficient. However, the underlying ideas can best be explained for the univariate case, and for reasons of simplicity we will focus on this case.

2.2 Fundamental Properties

A key property of holonomic functions is that they can be described by a finite amount of data, and hence faithfully represented in a computer. This is almost obvious for univariate holonomic sequences: all the (infinitely many) terms of such a sequence are uniquely determined by the linear recurrence and a suitable (finite) number of initial values. If $(a_n)_{n=0}^{\infty}$ satisfies the recurrence

$$p_0(n)a_n + p_1(n)a_{n+1} + \cdots + p_{r-1}(n)a_{n+r-1} + p_r(n)a_{n+r} = 0$$

for all $n \in \mathbb{N}$, where $p_0,\dots,p_r$ are certain polynomials, and p_r is not the zero polynomial, then the recurrence uniquely determines the value of a_{n+r} once we know the values of $a_n,\dots,a_{n+r-1}$, unless n is a root of the polynomial p_r. In order to fix a particular solution of the recurrence, it is therefore enough to fix the values $a_0,\dots,a_{r-1}$ as well as the values a_{n+r} for every positive integer root n of p_r. Note that p_r is a univariate polynomial, so it can have only finitely many roots.

The situation is not much different for holonomic functions in a continuous variable. In order to fix a particular solution of a given differential equation

$$q_0(z)y(z) + q_1(z)y'(z) + \cdots + q_{s-1}(z)y^{(s-1)}(z) + q_s(z)y^{(s)}(z) = 0,$$

where $q_0,\dots,q_s$ are polynomials, it suffices to fix the initial conditions $y(0)$, $y'(0)$, $\dots$, $y^{(s-1)}(0)$, plus possibly some finitely many further values $y^{(n)}(0)$. For which indices n the value of $y^{(n)}(0)$ does not follow from the earlier values by the differential equation is not as obvious as in the case of a recurrence equation. One possibility is to make use of the following theorem, which associates to a given differential equation a recurrence equation from which the relevant indices n can then be read off as described before.

Theorem 1. *Let $(a_n)_{n=0}^{\infty}$ be a sequence and $y(z) = \sum_{n=0}^{\infty} a_n z^n$ the (formal) power series whose coefficient sequence is $(a_n)_{n=0}^{\infty}$. Then $(a_n)_{n=0}^{\infty}$ is holonomic in the sense* *of Definition 1 if and only if $y(z)$ is holonomic in the sense of Definition 2. A differential equation satisfied by $y(z)$ can be computed from a known recurrence equation for $(a_n)_{n=0}^{\infty}$ and vice versa.*

The theorem is based on the observation that multiplying a series by z^{-1} corresponds to a forward shift of the coefficient sequence, and a differentiation followed by a multiplication with z corresponds to a multiplication by n. Here is an example for obtaining a recurrence equation for $(a_n)_{n=0}^{\infty}$ from a given differential equation for the power series $y(z) = \sum_{n=0}^{\infty} a_n z^n$.

$$\begin{aligned}
&(z-2)y''(z) + 5zy'(z) - y(z) = 0\\
\Rightarrow\quad &(z-2)\sum_{n=0}^{\infty} a_n n(n-1)z^{n-2} + 5z\sum_{n=0}^{\infty} a_n n z^{n-1} - \sum_{n=0}^{\infty} a_n z^n = 0\\
\Rightarrow\quad &\sum_{n=0}^{\infty} a_n n(n-1)z^{n-1} - 2\sum_{n=0}^{\infty} a_n n(n-1)z^{n-2} + 5\sum_{n=0}^{\infty} a_n n z^n - \sum_{n=0}^{\infty} a_n z^n = 0\\
\Rightarrow\quad &\sum_{n=0}^{\infty} a_{n+1}(n+1)nz^n - 2\sum_{n=0}^{\infty} a_{n+2}(n+2)(n+1)z^n + 5\sum_{n=0}^{\infty} a_n n z^n - \sum_{n=0}^{\infty} a_n z^n = 0\\
\Rightarrow\quad &\sum_{n=0}^{\infty}\Big((n+1)n\,a_{n+1} - 2(n+2)(n+1)a_{n+2} + 5n\,a_n - a_n\Big)z^n = 0\\
\Rightarrow\quad &-2(n+2)(n+1)a_{n+2} + (n+1)n\,a_{n+1} + (5n-1)a_n = 0 \qquad (n \geq 0).
\end{aligned}$$

See *The Concrete Tetrahedron* [28, Thm. 7.1] for the general case. The reverse direction works similarly.

Theorem 1 does not generalize to multivariate D-finite functions, it does however hold (by definition) for holonomic functions in several variables. In fact, Theorem 1 is the motivation for defining multivariate holonomy as in Definition 4.

A second useful feature of holonomic functions is that their asymptotic behaviour can be described easily. We say that two sequences $(a_n)_{n=0}^{\infty}, (b_n)_{n=0}^{\infty}$ are asymptotically equivalent if a_n/b_n converges to 1 for $n \to \infty$. Similary, two functions $f(z), g(z)$ are called asymptotically equivalent at some point ζ if $f(z)/g(z)$ converges to 1 for $z \to \zeta$. The following theorem describes the possible asymptotic behaviour of holonomic sequences and functions. Unlike Theorem 1, it is not straightforward.

Theorem 2. *[42, 18, 27]*

1. *If $(a_n)_{n=0}^{\infty}$ is a holonomic sequence, then there exist constants $c_1, \dots, c_m$, polynomials $p_1, \dots, p_m$, natural numbers $r_1, \dots, r_m$, constants $\gamma_1, \dots, \gamma_m$, $\phi_1, \dots, \phi_m$, $\alpha_1, \dots, \alpha_m$ and natural numbers $\beta_1, \dots, \beta_m$ such that*

$$a_n \sim \sum_{k=1}^{m} c_k\, \mathrm{e}^{p_k\left(n^{1/r_k}\right)} n^{\gamma_k n} \phi_k^n n^{\alpha_k} \log(n)^{\beta_k} \qquad (n \to \infty).$$

2. *If $y(z)$ is a holonomic analytic function with a singularity at $\zeta \in \mathbb{C}$, then there exist constants $c_1, \dots, c_m$, polynomials $p_1, \dots, p_m$, natural numbers $r_1, \dots, r_m$, constants $\alpha_1, \dots, \alpha_m$, and natural numbers $\beta_1, \dots, \beta_m$ such that*

$$y(x) \sim \sum_{k=1}^{m} c_k \, \mathrm{e}^{p_k\left((z-\zeta)^{-1/r_k}\right)}(z-\zeta)^{\alpha_k} \log(z-\zeta)^{\beta_k} \qquad (z \to \zeta).$$

Typically, one of the terms in the sum dominates all the others, so we can take $m = 1$. As an example, for the sequence $(a_n)_{n=0}^{\infty}$ from Example 1.4 we have $a_n \sim c\,(12+17\sqrt{2})^n n^{-3/2}$ where

$$c \approx 0.22004376711264303785068975981048665667815804290 7.$$

All the data in the asymptotic expression can be calculated exactly from a given recurrence or differential equation, except for the multiplicative constants c_k. These can however calculated numerically to arbitrarily high precision. In typical examples, it is easy to compute at least a few dozen decimal digits for them.

It is also possible to compute numerically the values of an analytic holonomic function to arbitrary precision, as stated in part 2 of the following theorem. The statement about sequences in part 1 is trivial (all terms of the sequence can be computed using the recurrence), but part 2 is not because it also covers the case where the evaluation point is outside of the disk of convergence of the series. This is known as effective analytic continuation.

Theorem 3. *1. If a holonomic sequence $(a_n)_{n=0}^{\infty}$ is given in terms of recurrence equation and a suitable number of initial values $a_0, a_1, \ldots, a_k$, then we can efficiently compute the nth term a_n of the sequence for every given index n.*

2. [13, 22, 23, 33] If a holonomic analytic function $y(z)$ is given in terms of a differential equation and a suitable number of initial values $y(0), y'(0), \ldots, y^{(k)}(0)$, and if we are given some complex number ζ with rational real and imaginary part and a polygonal path from 0 to ζ whose vertices have rational real and imaginary path, and some positive rational number ε, then we can efficiently compute a number $\tilde{y}$ such that for the value $y(\zeta)$ of the analytic continuation of y along the given path to ζ we have $|y(\zeta) - \tilde{y}| < \varepsilon$.

3 What are Closure Properties?

If p and q are polynomials, then also their sum $p+q$, their product pq, the composition $p \circ q$, the derivative p', and the indefinite integral $\int p$ are polynomials. We say that the class of polynomials is closed under these operations. Also the class of holonomic functions is closed under a number of operations.

Theorem 4. *[43, 40, 28]*

1. *If $(a_n)_{n=0}^{\infty}$ and $(b_n)_{n=0}^{\infty}$ are holonomic sequences, then so are $(a_n+b_n)_{n=0}^{\infty}$ and $(a_n b_n)_{n=0}^{\infty}$ and $\left(\sum_{k=0}^{n} a_k b_{n-k}\right)_{n=0}^{\infty}$.*
2. *If $a(z)$ and $b(z)$ are holonomic functions, then so are $a(z)+b(z)$ and $a(z)b(z)$.*
3. *If $(a_n)_{n=0}^{\infty}$ is a holonomic sequence and $\alpha, \beta \in \mathbb{Q}$ are nonnegative constants then $(a_{\lfloor \alpha n+\beta \rfloor})_{n=0}^{\infty}$ is a holonomic sequence.*

4. *If $a(z)$ is a holonomic function, then so are $a'(z)$ and $\int a(z)dz$.*
5. *If $a(z)$ is a holonomic function and $b(z)$ is an algebraic function, i.e., there is a nonzero bivariate polynomial $p(z,y)$ such that $p(z,b(z))$ is identically zero, then the composition $a(b(z))$ is holonomic.*

The theorem is most useful for recognizing a quantity given in terms of some expression as holonomic. For example, using the theorem, it is easy to see that

$$y(z) = \exp\bigl(1-\sqrt{1-z^2}\bigr) + \int \log(1-z)^2 dz$$

is holonomic: the innermost functions $\exp(z)$ and $\log(z)$ are holonomic (Example 2.2), by part 3 of the theorem $\exp(1-\sqrt{1-z^2})$ and $\log(1-z)$ are holonomic (the arguments are algebraic because they satisfy the equations $(1-y)^2-(1-z^2)=0$ and $y-(1-z)=0$, respectively), then by part 2 also $\log(1-z)^2$ is holonomic, and then by part 4 also $\int \log(1-z)^2 dz$ is holonomic. Finally, using once more part 2 it follows that $y(z)$ is holonomic.

By a similar reasoning, it is clear by inspection that

$$a_n = \sum_{k=1}^{n} \frac{1+2^k}{3+k^2} k! - (2n+5)! + \sum_{k=1}^{n} \frac{\sum_{i=1}^{k} \frac{\sum_{j=1}^{i} \frac{1}{j}}{i}}{k}$$

is holonomic.

By just looking at an expression, the closure properties in Theorem 4 are often sufficient to assert that some quantity is holonomic, which means that there does exist some differential equation or recurrence which has the object in question as solution. The equations can usually not be read off directly, but it is possible to compute them with computer algebra. For the two examples above, computer algebra packages for holonomic functions need virtually no time to find a differential equation for $y(z)$ of order 5 with polynomial coefficients of degree 14 and a recurrence for a_n of order 7 with polynomial coefficients of order 37.

The idea behind these algorithms is as follows. Consider for example two sequences $(a_n)_{n=0}^{\infty}$ and $(b_n)_{n=0}^{\infty}$ satisfying recurrence equations

$$a_{n+2} = u_1(n)a_{n+1} + u_0(n)a_n, \qquad b_{n+2} = v_1(n)b_{n+1} + v_0(n)b_n$$

for some known rational functions u_0, v_0, u_1, v_1. Let $(c_n)_{n=0}^{\infty}$ be the sum of these two sequences, i.e., $c_n = a_n + b_n$ for all $n \in \mathbb{N}$. Our goal is to compute a recurrence for $(c_n)_{n=0}^{\infty}$. Let us make an ansatz for a recurrence of order 4,

$$p_0(n)c_n + p_1(n)c_{n+1} + p_2(n)c_{n+2} + p_3(n)c_{n+3} + p_4(n)c_{n+4} = 0,$$

with undetermined polynomials $p_0, \dots, p_4$. We will see in a moment that 4 is a good choice. By definition of the c_n, in order for the recurrence to hold, we must have

$$\begin{aligned}&p_0(n)\big(a_n+b_n\big)+p_1(n)\big(a_{n+1}+b_{n+1}\big)+p_2(n)\big(a_{n+2}+b_{n+2}\big)\\&\quad+p_3(n)\big(a_{n+3}+b_{n+3}\big)+p_4(n)\big(a_{n+4}+b_{n+4}\big)=0.\end{aligned}$$

Using the known recurrences, we can reduce the higher order shifts to lower order shifts:

$$\begin{aligned}a_{n+2}&=u_1(n)a_{n+1}+u_0(n)a_n\\a_{n+3}&=u_1(n+1)a_{n+2}+u_0(n+1)a_{n+1}\\&=u_1(n+1)\big(u_1(n)a_{n+1}+u_0(n)a_n\big)+u_0(n+1)a_{n+1}\\&=\big(u_1(n+1)u_1(n)+u_0(n+1)\big)a_{n+1}+u_1(n+1)u_0(n)a_n\\a_{n+4}&=u_1(n+2)a_{n+3}+u_0(n+2)a_{n+2}\\&=u_1(n+2)\big(\big(u_1(n+1)u_1(n)+u_0(n+1)\big)a_{n+1}+u_1(n+1)u_0(n)a_n\big)\\&\quad+u_0(n+2)\big(u_1(n)a_{n+1}+u_0(n)a_n\big)\\&=\big(u_0(n+2)u_1(n)+u_0(n+1)u_1(n+2)+u_1(n)u_1(n+1)u_1(n+2)\big)a_{n+1}\\&\quad+u_0(n)\big(u_0(n+2)+u_1(n+1)u_1(n+2)\big)a_n,\end{aligned}$$

and analogously for the shifted versions of b_n. After applying these substitutions, the ansatz for the recurrence for c_n takes the form

$$\begin{aligned}&p_0(n)\big(a_n+b_n\big)+p_1(n)\big(a_{n+1}+b_{n+1}\big)\\&\quad+p_2(n)\big(\boxed{}a_{n+1}+\boxed{}a_n+\boxed{}b_{n+1}+\boxed{}b_n\big)\\&\quad+p_3(n)\big(\boxed{}a_{n+1}+\boxed{}a_n+\boxed{}b_{n+1}+\boxed{}b_n\big)\\&\quad+p_4(n)\big(\boxed{}a_{n+1}+\boxed{}a_n+\boxed{}b_{n+1}+\boxed{}b_n\big)=0,\end{aligned}$$

where the symbol $\boxed{}$ represents certain expressions involving the known rational functions u_0,u_1,v_0,v_1 as indicated above. Reordering the equation leads to

$$\begin{aligned}&\big(p_0(n)+\boxed{}p_2(n)+\boxed{}p_3(n)+\boxed{}p_4(n)\big)a_n\\&+\big(p_1(n)+\boxed{}p_2(n)+\boxed{}p_3(n)+\boxed{}p_4(n)\big)a_{n+1}\\&+\big(p_0(n)+\boxed{}p_2(n)+\boxed{}p_3(n)+\boxed{}p_4(n)\big)b_n\\&+\big(p_1(n)+\boxed{}p_2(n)+\boxed{}p_3(n)+\boxed{}p_4(n)\big)b_{n+1}=0,\end{aligned}$$

where we write again $\boxed{}$ to denote certain expressions of the u_0,u_1,v_0,v_1 which are a bit too messy to be spelled out here explicitly. This latter equation is certainly valid if we choose polynomials $p_0,\dots,p_4$ that turn the four expressions in front of a_n,a_{n+1},b_n,b_{n+1} to zero. Such polynomials can be found by solving the linear system

$$\begin{pmatrix} 1 & 0 & \boxed{} & \boxed{} & \boxed{} \\ 0 & 1 & \boxed{} & \boxed{} & \boxed{} \\ 1 & 0 & \boxed{} & \boxed{} & \boxed{} \\ 0 & 1 & \boxed{} & \boxed{} & \boxed{} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = 0.$$

This is an underdetermined homogeneous linear system with 4 equations and 5 variables, so it must have a nontrivial solution vector, and the coordinates of this vector correspond do the coefficients of the recurrence we want to compute. Note that the system involves the variable n as parameter, and it has to be solved with this parameter kept symbolic.

The order 4 in the ansatz for the recurrence of $(c_n)_{n=0}^{\infty}$ was chosen such as to ensure that the resulting linear system has more variables than equations. In general, if $(a_n)_{n=0}^{\infty}$ satisfies a recurrence of order r and $(b_n)_{n=0}^{\infty}$ satisfies a recurrence of order s, the linear system obtained for constructing a recurrence for the sum will have at most $r+s$ equations, and therefore it must have a nontrivial solution as soon as we have at least $r+s+1$ variables $p_0,\dots,p_{r+s}$. Likewise, for the product sequence $(a_nb_n)_{n=0}^{\infty}$ a similar construction leads to a linear system with rs equations, which hence has a nontrivial solution once we supply at least $rs+1$ variables $p_0,\dots,p_{rs}$. The arguments for the other operations listed in Theorem 4 are similar.

Holonomic closure properties are not only interesting for finding appropriate holonomic descriptions of objects that are given in some other form. They can also be used for proving identities. If two holonomic objects A and B are given in some form, it may not be obvious at first glance whether they are actually equal. Using closure properties, we can compute a recurrence for $A-B$ (or, if A and B depend on a continuous variable, a recurrence for the Taylor coefficients of $A-B$ by way of Theorem 1). Then if the identity is valid for a certain finite number of initial values, it follows by induction that it is true.

Example 5. Consider the following identity for Hermite polynomials. We regard it as a (formal) power series with respect to t, where x and y are viewed as constant parameters. In the first term on the left the expression $H_n(x)H_n(y)\frac{1}{n!}$ is regarded as a sequence in the discrete variable n, with x and y as parameters. Apply the closure properties algorithms as indicated by the braces to obtain a linear recurrence for the coefficients in the series expansion of the whole left hand side.

$$\underbrace{\underbrace{\sum_{n=0}^{\infty}\underbrace{\underbrace{H_n(x)}_{\substack{\text{rec. of}\\ \text{ord. 2}}}\underbrace{H_n(y)}_{\substack{\text{rec. of}\\ \text{ord. 2}}}\underbrace{\frac{1}{n!}}_{\substack{\text{rec. of}\\ \text{ord. 1}}}}_{\text{rec. of order 4}}\,t^n}_{\text{differential equation of order 5}} - \underbrace{\underbrace{\underbrace{\frac{1}{\sqrt{1-4t^2}}}_{\substack{\text{alg.eq.}\\ \text{of deg. 2}}}}_{\substack{\text{diff.eq.}\\ \text{of ord. 1}}}\underbrace{\underbrace{\exp}_{\substack{\text{diff.eq.}\\ \text{of ord. 1}}}\Big(\underbrace{\frac{4t(xy-t(x^2+y^2))}{1-4t^2}}_{\text{alg.eq. of degree 1}}\Big)}_{\text{differential equation of order 1}}}_{\text{differential equation of order 1}}}_{\substack{\text{differential equation of order 5}\\ \rightsquigarrow\ \text{recurrence equation of order 4}}} = 0$$

If c_n denotes the coefficient of t^n in the series expansion of the left hand side, we obtain the recurrence

$$(n+4)c_{n+4} - 4xy\,c_{n+3} - 4(2n-2x^2-2y^2+5)c_{n+2} - 16xy\,c_{n+1} + 16(n+1)c_n = 0$$

for all $n \geq 0$. Direct calculation confirms that $c_0 = c_1 = c_2 = c_3 = 0$, which together with the recurrence implies inductively that $c_n = 0$ for all $n \geq 0$. This proves the identity.

In general, once a recurrence of a holonomic sequence is known, it always suffices to check a certain finite number of initial values for being zero in oder to decide whether the whole sequence is zero. The number of terms needed is the maximum of the order of the recurrence and the largest integer root of the leading coefficient (if there is any such root). Although integer roots in the leading coefficient are not common, their possible existence must always be taken into account. It is in general not sufficient to only estimate the order of a recurrence (which could be done very quickly) without actually computing it.

The example above is a typical application of the holonomic toolkit, except that it is usually not possible to prove an identity using only Theorems 1 and 4 and the mere definitions of the objects involved. A more realistic scenario is that we want to prove an identity, say of the form $A = BC$, which involves holonomic quantities A, B, C for which we calculated defining equations using other techniques, and then the algorithms behind Theorem 4 are only used to complete the proof by combining the partial results into a defining equation for the whole equation.

Closure properties are also available in several variables. The class of D-finite functions in several (discrete or continuous) variables is closed under addition and multiplication, under linear translates $n \rightsquigarrow \lfloor \alpha n + \beta \rfloor$ of discrete variables n (for fixed positive rational numbers α, β), and under compositions $z \rightsquigarrow y(z)$ of continuous variables z by some multivariate algebraic functions y that must not be constant with respect to z, may or may not depend on the other continuous variables, and must not depend on any of the discrete variables. The underlying ideas of the algorithms is the same as in the univariate case.

Also the class of holonomic functions in several variables enjoys these closure properties, as well as some further ones which in general do not apply to D-finite functions.

Theorem 5. *[43] Let $a = a(n_1,\dots,n_p,z_1,\dots,z_q)$ and $b = b(n_1,\dots,n_p,z_1,\dots,q)$ be holonomic with respect to the discrete variables $n_1,\dots,n_p$ and the continuous variables $z_1,\dots,z_q$. Then:*

1. *The sum $a+b$ and the product ab are holonomic,*
2. *If b is algebraic, not constant with respect to z_1, and independent of $n_1,\dots,n_p$, then $a(n_1,\dots,n_p,b,z_2,\dots,z_q)$ is holonomic,*
3. *$a(\lfloor \alpha n_1 + \beta \rfloor, n_2,\dots,n_p,z_1,\dots,z_q)$ is holonomic for any fixed positive constants $\alpha,\beta \in \mathbb{Q}$.*
4. *$a(0,n_2,\dots,n_p,z_1,\dots,z_q)$ and $a(n_1,\dots,n_p,0,z_2,\dots,z_q)$ are holonomic,*
5. *$\sum_{k=0}^{n_1} a(k,n_2,\dots,n_p,z_1,\dots,z_q)$ and $\int_0^{z_1} a(n_1,\dots,n_p,t,z_2,\dots,z_q)dt$ are holonomic (provided the integral converges),*
6. *$\sum_{k=-\infty}^{\infty} a(k,n_2,\dots,n_p,z_1,\dots,z_q)$ and $\int_{-\infty}^{\infty} a(n_1,\dots,n_p,t,z_2,\dots,z_q)dt$ are holonomic as functions in $n_2,\dots,n_p,z_1,\dots,z_p$ and $n_1,\dots,n_p,z_2,\dots,z_q$, respectively (provided these quantities are meaningful),*

This theorem is considerably more deep than Theorem 4, and the algorithms behind it are less straightforward than those sketched before for the univariate case. See the chapter on symbolic summation and integration in this volume for further information about these algorithms.

4 What is Guessing?

We have seen that closure properties are useful for finding the holonomic representation of holonomic objects which are given in terms of holonomic functions for which defining equations are known (possibly recursively descending a nested expression). But when no such relation to known functions is available (yet), we cannot obtain defining equations in this way. We may in fact be faced with objects of which we do not know whether they are holonomic or not.

How can we check an arbitrary object for being holonomic? Of course, this question makes only sense relative to some choice of assumptions we are willing to make about how the object is "given", or more generally, what information about it we want to consider known. A very weak assumption which is almost always satisfied in practice is that we can calculate for every specific index n the nth term of the sequence (or the nth term of the power series) of interest. For example, suppose the first few terms of a sequence $(a_n)_{n=0}^{\infty}$ are known to be

5, 12, 21, 32, 45, 180, 797, 2616, 6837, 15260, 30405, 55632, 95261, 154692.

How can we check whether this sequence is, say, a polynomial sequence? Strictly speaking, we cannot tell this at all without taking into account all the terms of the sequence. But from the available finite amount of data we can at least get an idea. By means of interpolation [19], we can easily compute for any tuple of $n+1$ numbers $x_0,\dots,x_n$ the (unique) polynomial p of degree at most n such that $p(0)=x_0$, $p(1)=x_1$, …, $p(n)=x_n$. For example, for the first two terms we find $p(n)=7n+5$, which however cannot be correct for all $n\geq 0$ because already $p(2)=19\neq 21$. The interpolating polynomial for the first three points is $p(n)=n^2+6n+5$, which is correct (by construction) for $n=0,1,2$, happens to be correct also for $n=3$ and $n=4$, although these values had not been used in the construction of p. However, also this polynomial cannot be correct for all $n\geq 0$ because we have $p(5)=60\neq 180$. Interpolation of the first six terms gives $p(n)=n^5-10n^4+35n^3-49n^2+30n+5$ which turns out to match all the terms listed above. Of course, this does not prove that the polynomial is correct for all greater indices as well, but the more terms match, the more tempting it becomes to believe so. Interpolating polynomials based on a finite number of terms of some infinite sequence can therefore be considered as a guess for a possible description of the entire sequence, and the difference between the number of terms taken into account and the degree of the resulting interpolating polynomial can be considered as measuring the confidence of the guess (e.g., 0: no evidence, 1: somewhat reliable, 10: reasonably trustworthy, 100: almost certain).

In a similar fashion, it is also possible to come up with reliable guesses for recurrence equations possibly satisfied by some infinite sequence of which only a finite number of terms are known, or for differential equations possibly satisfied by a function of which the first few terms of the series expansion are known. To illustrate the technique, suppose we are given a sequence $(a_n)_{n=0}^{\infty}$ starting like

$$1,\ 2,\ 14,\ 106,\ 838,\ 6802,\ 56190,\ 470010,\ 3967310,\ 33747490.$$

Let us search for a recurrence of order $r=2$ with polynomial coefficients of degree $d=1$, i.e., a recurrence of the form

$$(c_{0,0}+c_{0,1}n)a_n+(c_{1,0}+c_{1,1}n)a_{n+1}+(c_{2,0}+c_{2,1}n)a_{n+2}=0$$

for constants $c_{i,j}$ yet to be determined. Since the recurrence is supposed to hold for $n=0,\dots,7$ (at least), we obtain the following system of linear constraints:

$$\begin{aligned}
n{=}0 &: (c_{0,0}+c_{0,1}0)1+(c_{1,0}+c_{1,1}0)2+(c_{2,0}+c_{2,1}0)14=0\\
n{=}1 &: (c_{0,0}+c_{0,1}1)2+(c_{1,0}+c_{1,1}1)14+(c_{2,0}+c_{2,1}1)106=0\\
n{=}2 &: (c_{0,0}+c_{0,1}2)14+(c_{1,0}+c_{1,1}2)106+(c_{2,0}+c_{2,1}2)838=0\\
&\ \ \vdots\\
n{=}7 &: (c_{0,0}+c_{0,1}7)470010+(c_{1,0}+c_{1,1}7)3968310+(c_{2,0}+c_{2,1}7)33747490=0.
\end{aligned}$$

In other words, any choice of the $c_{i,j}$ which corresponds to a recurrence that holds for all $n\in\mathbb{N}$ must in particular correspond to recurrence that holds for $n=0,\dots,7$, and the choices for $c_{i,j}$ that correspond to a recurrence valid for $n = 0, \dots, 7$ are precisely the solutions of the following homogeneous linear system:

$$\begin{pmatrix} 1 & 0 & 2 & 0 & 14 & 0 \\ 2 & 2 & 14 & 14 & 106 & 106 \\ 14 & 28 & 106 & 212 & 838 & 1676 \\ 106 & 318 & 838 & 2514 & 6802 & 20406 \\ 838 & 3352 & 6802 & 27208 & 56190 & 224760 \\ 6802 & 34010 & 56190 & 280950 & 470010 & 2350050 \\ 56190 & 337140 & 470010 & 2820060 & 3968310 & 23809860 \\ 470010 & 3290070 & 3968310 & 27778170 & 33747490 & 236232430 \end{pmatrix} \begin{pmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \\ c_{2,0} \\ c_{2,1} \end{pmatrix} = 0$$

This system has the solution $(0, 9, -14, -10, 2, 1)$, which means that the infinite sequence $(a_n)_{n=0}^{\infty}$ of which we were given the first 10 terms above satisfies the recurrence

$$9n\,a_n + (-14 - 10n)a_{n+1} + (2n+1)a_{n+2} = 0,$$

at least for $n = 0, 1, \dots, 7$. Note that the linear system had more equations than variables so that a priori we would not have expected that it has a nonzero solution at all. This makes it reasonable to guess that the recurrence we found is not just a match of the given data, but in fact a "true" recurrence, valid for all $n \in \mathbb{N}$. The reliability of such a guess can be estimated by the difference between number of variables and number of equations in the linear system (e.g. 0: already some indication, 10: convincing evidence, 100: strong evidence).

Guessing is a very popular technique in experimental mathematics, it is certainly a more widely used (and known?) part of the holonomic toolkit than the algorithms for closure properties. Several software packages provide efficient implementations of the algorithm sketched above, or of more sophisticated algorithms based on Hermite-Pade approximation [3]. Maple users can use gfun [37], Mathematica users can use the old package of Mallinger [32] or Kauers's package [26], which also supports multivariate guessing. For Axiom there is a package by Hebisch and Rubey [21]. Recent versions of these packages have no trouble processing hundreds or even thousands of terms.

Note the computational difference between the linear algebra problems for guessing and closure properties: For guessing, we solve large overdetermined systems with constant coefficients, whereas for closure properties we solve small underdetermined systems with polynomial coefficients.

Note also that if no equation can be found by guessing, then there definitely does not exist an equation of the specified order and degree. On the other hand, a guessed equation may be incorrect, although this very rarely happens in practice. The requirement that a dense overdetermined linear system should have a nontrivial solution acts as a strong filter against false guesses. In case of doubt, there are some other tests which can be applied to a guessing result to estimate how plausible it is [9].

4.1 Trading Order for Degree

The first step in the guessing procedure is to make a choice for the order r and the degree d of the equation to be searched. The possible choices are limited by the number N of available terms, because we want to end up with an overdetermined linear system. (An underdetermined system will always have nontrivial solutions, but these have no reason to have any significance for the infinite object from which the data sample originates.) An overdetermined system is obtained for r and d such that $(d+2)(r+1) < N$. The possible choices for r and d are thus the points (r,d) under a hyperbola determined by the number of available terms.

If for some point (r,d) below the hyperbola no equation is found, it may still be that there is an equation for some other point (r',d') (unless $r' \leq r$ and $d' \leq d$). An exhaustive search needs to go through all the integer points right below the hyperbola. These are only finitely many.

If an object is holonomic, it satisfies not only a single equation but infinitely many of them. First of all we can pass from any given equation to a higher degree one by simply multiplying it by n or z, respectively, and we can produce higher order equations by shifting or differentiating, respectively. This means that if there is an equation of order r and degree d, then there is also one of order r' and degree d' for every (r',d') with $r' \geq r$ and $d' \geq d$. In addition, in examples coming from applications, there usually exist further equations. A typical shape for the region of all points (r,d) for which there exists an equation of order r and degree d is shown in Figure 2. As indicated by the curves in this figure, the equations which can be recovered with the smallest amount of data are those for which $r/d \approx 1$. In contrast, the minimal order operator tends to require the maximal number of terms. This is unfortunate because this operator is for many applications the most interesting one. Modern guessing packages [26] use an algorithm which guesses several nonminimal order recurrences (taking advantage of their small size) and construct from them a guess for the minimal order operator (which is most interesting to the user) [6, 8]. Very recent results [11, 12, 24] offer further improvements by giving precise a priori knowledge about the shape of the region where equations can be found.

Example 6. The technique explained above has been utilized for certain sequences arising in particle physics [6]. A certain quantity arising in this context (named $C^{(2)}_{2,q,C_F^3}$ in Table 5 of [6]) satisfies a recurrence of order 35 and degree 938. This is the minimal order recurrence for this sequence. In order to guess it directly, at least 33841 terms of the sequence are needed. However, by first guessing a smaller recurrence with nonminimal order (in this case, order 51 and degree 92), it was sufficient to know 5114 terms of the sequence. See Figure 1 for an illustration.

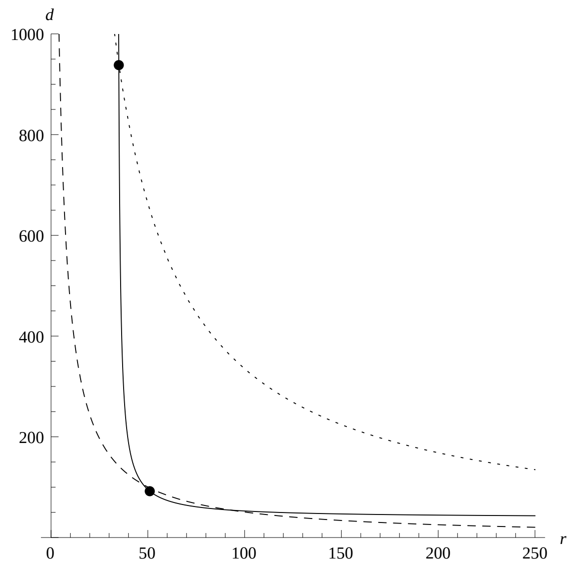

Fig. 1 For the sequence mentioned in Example 6, it turns out that there exists a recurrence of order r and degree d whenever (r,d) is above the solid curve. With 5114 terms available, guessing can find recurrence equations of order r and degree d whenever (r,d) is below the dashed curve. With 33841 terms, guessing can find recurrences of order r and degree d for all (r,d) below the dotted curve. The two dots mark the position of the minimal order recurrence and the recurrences which were actually guessed, and from which the minimal order recurrence was constructed.

4.2 Modular Techniques

A common problem in computer algebra is the intermediate growth of expressions during a calculation. In the context of holonomic functions, it is not unusual that the output of a calculation is much longer than the input, and yet the intermediate expressions can still be much longer than that, thereby causing severe time and memory problems. A classical technique in computer algebra for dealing with this problem is the use of homomorphic images (a.k.a. modular arithmetic) [19, 25]. The basic idea is that instead of solving a problem involving polynomials, one evaluates the polynomials at several points, then solves the resulting small problems, which no longer involve polynomials but only numbers, and afterwards combines the various solutions by interpolation to a solution of the original problem. To the same effect, problems involving rational numbers are mapped to problems in finite fields, solved in these fields, and afterwards the modular solutions are combined to a rational solution using the Chinese remainder algorithm and rational reconstruction.

Implementations use this technique internally to speed up the computations and to save memory. The user does not see this, and does not need to care. Modular techniques are however also useful for the user, because in large problems the hard part of the computation is usually not the guessing itself but the generation of sufficiently many terms of the sequence or series. It is typically about one order of magnitude faster to compute the terms only modulo some fixed prime. Therefore, one should first compute the data only modulo some fixed prime p (for instance $p = 2^{31} - 1 = 2147483647$), and then apply the guesser modulo this prime. If it does not find anything, then (with high probability) it would also not find any equation for the actual data, and there is no point in computing it.

On the other hand, if a modular equation for the modular data is found, then one can still go ahead and calculate the data modulo many other primes, reconstruct the non-modular data from the results, and apply the guesser to those in order to get the non-modular equation. This is possible, but there is a better way: calculate the data modulo several other primes, then for each prime guess an equation modulo the prime, and in the end reconstruct the non-modular equation from the modular ones. In practice, this strategy tends to require much fewer primes, and is therefore much more efficient.

Example 7. Consider the sequence $a_n = \sum_{k=0}^{n}(n+k)^3 3^{4k}\binom{n}{k}^2$. It satisfies a recurrence of order 2 and degree 4 which a guesser can recover from the first 20 terms of the sequence. The example is so small that both the computation of these twenty terms via the sum and the guessing can be done in virtually no time. The arithmetic effect described above can nevertheless be observed already here.

The term a_{20}, which is the largest in the sample, has 42 decimal digits, so if we work with primes of 11 decimal digits, we need four of them in order to reconstruct the values of $a_0, \dots, a_{20}$ from their images modulo the primes.

On the other hand, the largest coefficient in the recurrence has only 12 decimal digits. (It is a fraction with a 7-digit numerator and a 5-digit denominator.) Therefore we can already recover it if we know the coefficients of the recurrence modulo two different primes. See Figure 2 for an illustration.

Example 8. For the sequence from Example 6, the largest among the first 5114 terms is a fraction with 13388 decimal digits in the numerator and 13381 digits in the denominator. In contrast, the largest coefficient in the minimal order recurrence is a fraction with 1187 decimal digits in the numerator and 7 digits in the denominator. The resulting speed-up is a factor of $(13388+13381)/(1187+7) \approx 22.4$.

4.3 Boot Strapping

We have remarked that the generation of a sufficient amount of data is often more expensive than guessing an equation from the data. Of course, once we have an equation, it is very cheap to calculate as much data as we please—this is one of the fundamental properties of holonomic objects. But if we already know an equation, we don't need to guess one. To some extent, the situation has the character of a chicken/egg problem: in order to guess a recurrence most efficiently, the best thing would be if we could already use it for generating data. Sometimes the conflict can be resolved by guessing auxiliary equations: in a first step, use a naive way to compute a small number of terms, then use them to guess some equation which can be used to generate further terms, and iterate until you have enough terms to guess the equation of interest. We conclude with two examples for this strategy.

Example 9. [29] Consider the lattice $\mathbb{N}^4$. We are interested in walks starting at $(0,0,0,0)$ and going to (i,j,k,l) which may consist of any number of steps, where

mod 100000000003	mod 100000000019	mod 100000000057	mod 100000000063
0	0	0	0
325	325	325	325
107896	107896	107896	107896
20619774	20619774	20619774	20619774
3180175360	3180175360	3180175360	3180175360
39281494338	39281494274	39281494122	39281494098
44150634536	44150625448	44150603864	44150600456
21005523377	21004395297	21001716107	21001293077
47849449427	47713597155	47390948009	47340003407
10040664517	94031491240	56009704700	50006264720
21985378201	66674685234	60311791810	64570282631
33735899035	69655913193	79966218922	18436302163
43166832072	5147734490	89883154498	50634811944
44406348808	99474064104	33709016642	55347756722
50094685503	23789791408	32316196105	19843824994
48089246095	80233657716	43056414134	27698338095
46630298409	55837852211	47226351528	21769511873
28184365337	25424314762	78343837230	83639568961
97965056397	76012039079	60755450198	76086130993
82688649229	13097788119	92702069386	96377809322
19000479750	73020719731	24797347614	67792308900

mod 10000000002200000000057	mod 10000000012000000003591
0	0
325	325
107896	107896
20619774	20619774
3180175360	3180175360
439281494350	439281494350
56844150636240	56844150636240
7050521005734892	7050521005734892
849076747874921728	849076747874921728
100057333113042384510	100057333113042384510
1595691831469856133143	1595691821669856129609
6025499912914500896416	6025498619314500429928
3987619360762795412890	3987472380362742409958
9655826781634081152247	9639354343828140989473
7039405589661276853185	5207872858000803723455
4174097428373312168941	1922634602838958137103
8692452790707403882121	8757614004539066331243
6892250317634951874859	1578404473178034386384
8887206360264581247194	8077821996665113985638
9184942883958236935737	6605404342457782542388
9037373502190121684807	9283417323516345218988

mod 100000000142000000062880000008584200000204687
0
325
107896
20619774
3180175360
439281494350
56844150636240
7050521005734892
849076747874921728
100057333113042384510
11595691833669856133200
1326025500203314500903940
149983987652356362796267776
16808619659524675834176961324
1868910957450565998671929645600
206379043919577487080949672719000
22652994757462351298463229473971200
2473240118789349718278365703625872670
268742205192237274018023266433839996880
29076651329353997890401249406299092602900
3133800494309454721113304131063624227776000

mod 100000000003	mod 100000000019
0	0
325	325
107896	107896
20619774	20619774
3180175360	3180175360
39281494338	39281494274
44150634536	44150625448
21005523377	21004395297
47849449427	47713597155
10040664517	94031491240
21985378201	66674685234
33735899035	69655913193
43166832072	5147734490
44406348808	99474064104
50094685503	23789791408
48089246095	80233657716
46630298409	55837852211
28184365337	25424314762
97965056397	76012039079
82688649229	13097788119
19000479750	73020719731

↓ ↓

Guessed equation modulo 100000000003:

$$(n^4+12337892631n^3+24675785259n^2+12337892629n)a_{n+2}+(99999999839n^4+76585608333n^3+24849274151n^2+43356113667n)a_{n+1}+(6400n^4+62512855233n^3+50051350526n^2+12564160956n+25025659263)a_n=0$$

Guessed equation modulo 100000000019:

$$(n^4+17278992424n^3+34557984845n^2+17278992422n)a_{n+2}+(99999999855n^4+66245242769n^3+40899461546n^2+16817950742n)a_{n+1}+(6400n^4+85551511805n^3+42205976763n^2+27757443749n+71102972391)a_n=0$$

↓ ↓

$$(n^4+\tfrac{260983}{104329}n^3+\tfrac{208979}{104329}n^2+\tfrac{52325}{104329}n)a_{n+2}-(164n^4+\tfrac{68466146}{104329}n^3+\tfrac{4539888}{6137}n^2+\tfrac{17371256}{104329}n)a_{n+1}+(6400n^4+\tfrac{3673408000}{104329}n^3+\tfrac{7348870400}{104329}n^2+\tfrac{6348339200}{104329}n+\tfrac{2005171200}{104329})a_n=0$$

Fig. 2 Two ways of using modular arithmetic in guessing, illustrated with the data of Example 7.

a single step can be of the form $(m,0,0,0)$ or $(0,m,0,0)$ or $(0,0,m,0)$ or $(0,0,0,m)$ for some positive integer m. If $a_{i,j,k,l}$ denotes the number of walks ending at the lattice point (i,j,k,l), we are interested in the sequence $a_{n,n,n,n}$ counting the walks that end on the diagonal. There is a simple way to compute these numbers, but this algorithm is costly. It would be somewhat hard to generate 1000 terms with it.

Therefore we proceed as follows: compute the terms $a_{n,n,k,k}$ for $0 \leq n,k \leq 25$, say, and use bivariate guessing to guess recurrence equations for this sequence with respect to n and k. Using these recurrences, it is much easier to calculate $a_{n,n,n,n}$ for $n = 0,\dots,1000$. These terms can finally be used to guess the desired recurrence for $a_{n,n,n,n}$.

Example 10. [35] For a certain application in combinatorial group theory, it was necessary to find a differential equation for the power series $[q^0]f(q,z)$, where $f(q,z)$ is a certain power series with respect to z whose coefficients are Laurent polynomials in q. The notation $[q^0]$ is meant to pick the constant term of each coefficient:

$$\begin{aligned} f(q,z) &= 1 + (q^{-1} + q)z + (q^{-2} + 4 + q^2)z^2 + \cdots \\ [q^0]f(q,z) &= 1 + 0z + 4z^2 + \cdots. \end{aligned}$$

The power series $f(q,z)$ was given in terms of a defining equation $p(q,z,f(q,z)) = 0$, where p is a polynomial in three variables which is too large to be printed here. Using Newton polygons [28, Chapter 6], it is possible to compute from p the first terms in the expansion of $f(q,z)$ with respect to z. But it is hard to generate enough terms to recover the differential equation for $[q^0]f(z)$.

Therefore we proceed as follows: compute the first 30 terms in the expansion of $f(q,z)$ for symbolic q, and use these to guess a general recurrence of the coefficients of $f(q,z)$ with symbolic q. Using this recurrence, generate many further terms of $f(q,z)$ with symbolic q, pick the constant terms and apply guessing to the result.

References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions, 9th edn. Dover Publications, Inc. (1972)
2. Andrews, G.E., Askey, R., Roy, R.: Special Functions, *Encyclopedia of Mathematics and its Applications*, vol. 71. Cambridge University Press (1999)
3. Beckermann, B., Labahn, G.: Fraction-free computation of matrix rational interpolants and matrix gcds. SIAM Journal of Matrix Analysis and Applications **22**(1), 114–144 (2000)
4. Bernstein, J.N.: The analytic continuation of generalized functions with respect to a parameter. Functional Analysis and its Applications **6**(4), 26–40 (russian); 273–285 (english translation) (1972)
5. Bjork, J.E.: Rings of Differential Operators. North-Holland (1979)
6. Blümlein, J., Kauers, M., Klein, S., Schneider, C.: Determining the closed forms of the $o(a_s^3)$ anomalous dimension and Wilson coefficients from Mellin moments by means of computer algebra. Computer Physics Communications **180**(11), 2143–2165 (2009)
7. Blümlein, J., Klein, S., Schneider, C., Stan, F.: A Symbolic Summation Approach to Feynman Integral Calculus. Journal of Symbolic Computation **47**(10), 1267–1289 (2012)

8. Bostan, A., Kauers, M., with an appendix by M. van Hoeij: The complete generating function for Gessel walks is algebraic. Proceedings of the AMS **138**(9), 3063–3078 (2010)
9. Bostan, A., Kauers, M.: Automatic classification of restricted lattice walks. In: Proceedings of FPSAC'09, pp. 201–215 (2009)
10. Bronstein, M., Petkovšek, M.: An introduction to pseudo-linear algebra. Theoretical Computer Science **157**(1), 3–33 (1996)
11. Chen, S., Kauers, M.: Order-degree curves for hypergeometric creative telescoping. In: Proceedings of ISSAC'12, pp. 122–129 (2012)
12. Chen, S., Kauers, M.: Trading order for degree in creative telescoping. Journal of Symbolic Computation **47**(8), 968–995 (2012)
13. Chudnovsky, D.V., Chudnovsky, G.V.: Computer algebra in the service of mathematical physics and number theory. In: D.V. Chudnovsky, R.D. Jenks (eds.) Computers in Mathematics, *Lecture Notes in Pure and Applied Mathematics*, vol. 125, pp. 109–232. Dekker, Stanford University (1986)
14. Chyzak, F.: An extension of Zeilberger's fast algorithm to general holonomic functions. Discrete Mathematics **217**, 115–134 (2000)
15. Chyzak, F., Kauers, M., Salvy, B.: A non-holonomic systems approach to special function identities. In: J. May (ed.) Proceedings of ISSAC'09, pp. 111–118 (2009)
16. Chyzak, F., Salvy, B.: Non-commutative elimination in Ore algebras proves multivariate identities. Journal of Symbolic Computation **26**, 187–227 (1998)
17. Corless, R.M., Gonnet, G.H., Hare, D.E., Jeffrey, D.J., Knuth, D.E.: On the Lambert W function. Adv. Comput. Math. **5** (1996)
18. Flajolet, P., Sedgewick, R.: Analytic Combinatorics. Cambridge University Press (2009)
19. von zur Gathen, J., Gerhard, J.: Modern Computer Algebra. Cambridge University Press (1999)
20. Gerhold, S.: Combinatorial sequences: Non-holonomicity and inequalities. Ph.D. thesis, RISC-Linz, Johannes Kepler Universität Linz (2005)
21. Hebisch, W., Rubey, M.: Extended Rate, more GFUN. Journal of Symbolic Computation **46**(8), 889–903 (2011)
22. van der Hoeven, J.: Fast evaluation of holonomic functions. Theor. Comput. Sci. **210**(1), 199216 (1999)
23. van der Hoeven, J.: Efficient accelero-summation of holonomic functions. Journal of Symbolic Computation **42**(4), 389–428 (2007)
24. Jaroschek, M., Kauers, M., Chen, S., Singer, M.F.: Desingularization explains order-degree curves for Ore operators. Tech. Rep. 1301.0917, ArXiv (2013)
25. Kauers, M.: Fast solvers for dense linear systems. Nuclear Physics B (Proc. Suppl.) **183**, 245–250 (2008)
26. Kauers, M.: Guessing handbook. Tech. Rep. 09-07, RISC-Linz (2009)
27. Kauers, M.: A Mathematica package for computing asymptotic expansions of solutions of p-finite recurrence equations. Tech. Rep. 11-04, RISC-Linz (2011)
28. Kauers, M., Paule, P.: The Concrete Tetrahedron. Springer (2011)
29. Kauers, M., Zeilberger, D.: The computational challenge of enumerating high dimensional rook walks. Advances in Applied Mathematics **47**(4), 813–819 (2011)
30. Koutschan, C.: Advanced applications of the holonomic systems approach. Ph.D. thesis, RISC-Linz, Johannes Kepler Universität Linz (2009)
31. Koutschan, C.: HolonomicFunctions (User's Guide). Tech. Rep. 10-01, RISC Report Series, University of Linz, Austria (2010).
URL http://www.risc.uni-linz.ac.at/research/combinat/software/HolonomicFunctions/
32. Mallinger, C.: Algorithmic manipulations and transformations of univariate holonomic functions and sequences. Master's thesis, J. Kepler University, Linz (1996)
33. Mezzarobba, M., Salvy, B.: Effective Bounds for P-Recursive Sequences. Journal of Symbolic Computation **45**(10), 1075–1096 (2010)
34. van der Poorten, A.: A proof that Euler missed... — Apéry's proof of the irrationality for $\zeta(3)$. The Mathematical Intelligencer **1**, 195–203 (1979)

35. Rechnitzer, A.: Personal communication (2012)
36. Salvy, B.: D-finiteness: algorithms and applications. In: M. Kauers (ed.) Proc. of ISSAC'05, pp. 2–3 (2005). Invited talk
37. Salvy, B., Zimmermann, P.: Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. ACM Transactions on Mathematical Software **20**(2), 163–177 (1994)
38. Sloane, N.J.: The on-line encyclopedia of integer sequences. URL http://www.oeis.org/
39. Stanley, R.P.: Differentiably finite power series. European Journal of Combinatorics **1**, 175–188 (1980)
40. Stanley, R.P.: Enumerative Combinatorics, Volume 2. Cambridge Studies in Advanced Mathematics 62. Cambridge University Press (1999)
41. Wilf, H.S., Zeilberger, D.: An algorithmic proof theory for hypergeometric (ordinary and q) multisum/integral identities. Inventiones mathematicae **108**, 575–633 (1992)
42. Wimp, J., Zeilberger, D.: Resurrecting the asymptotics of linear recurrences. Journal of Mathematical Analysis and Applications **111**, 162–176 (1985)
43. Zeilberger, D.: A holonomic systems approach to special function identities. Journal of Computational and Applied Mathematics **32**, 321–368 (1990)