

Computer Algebra, Power Series and Summation

Wolfram Koepf

Abstract. Computer algebra systems can do many computations that are relevant for orthogonal polynomials and their representations. In this preliminary training we will introduce some of those important algorithms: The automatic computation of differential equations and formal power series, hypergeometric representations, and the algorithms by Fasenmyer, Gosper, Zeilberger and Petkovšek/van Hoeij.

Mathematics Subject Classification (2000). Primary 68W30, 33F10; Secondary 30B10, 33C20.

Keywords. Computer algebra; algorithms for power series; algorithms for summation.

1. Introduction

I will use the computer algebra system *Maple* to program and demonstrate the methods considered. Of course one could also easily use any other general-purpose system like *Mathematica*, *Maxima*, *Reduce* or *Sage*.

Such general-purpose computer algebra systems

- contain a high-level programming language,
- are dialog-oriented and not compiling,
- are able to plot functions,
- and can compute with symbols.

The mostly used algorithms are:

- algorithms of linear algebra with many variables and coefficients that are rational functions,
- multivariate polynomial factorization (over \mathbb{Q}),

which can be treated algorithmically and are efficiently available in all general-purpose computer algebra systems. High-end algorithms contain algorithms for modular arithmetic, algebraic numbers, solving polynomial systems, differentiation

and integration, solving differential equations, as well as Taylor polynomials and power series.

2. Taylor Polynomials

The coefficients $B_k(x)$ of the power series

$$\sqrt{\frac{\left(\frac{1+z}{1-z}\right)^x - 1}{2xz}} = \sum_{k=0}^{\infty} B_k(x) z^k$$

are polynomials of degree k . In a 1978 publication Malcolm S. Robertson [16] conjectured that the coefficients of all these polynomials $B_k(x)$ are non-negative. He had computed $B_k(x)$ for $1 \leq k \leq 6$. This is reproduced by the *Maple* computation

```
> g:=sqrt((((1+z)/(1-z))^x-1)/(2*x*z));
```

$$g := 1/2 \sqrt{2} \sqrt{\frac{1}{xz} \left(\left(\frac{1+z}{1-z} \right)^x - 1 \right)}$$

```
> map(expand,series(g,z,7));
```

$$1 + 1/2 xz + \left(\frac{5x^2}{24} + 1/6 \right) z^2 + (1/16 x^3 + x/4) z^3 + \left(\frac{31}{360} + \frac{25x^2}{144} + \frac{79x^4}{5760} \right) z^4 \\ + \left(\frac{41x}{240} + \frac{7x^3}{96} + \frac{3x^5}{1280} \right) z^5 + \left(\frac{863}{15120} + \frac{85x^2}{576} + \frac{79x^4}{3840} + \frac{71x^6}{193536} \right) z^6 + O(z^7)$$

Robertson repeated his conjecture in the article [17] published in 1989. In this publication Robertson further conjectured that the coefficients A_k of the univariate power series

$$\sqrt{\frac{e^x - 1}{x}} = \sum_{k=0}^{\infty} A_k x^k$$

are all positive. It turns out that both conjectures are false, as the following computations show!¹

```
> f:=sqrt((exp(x)-1)/x);
```

$$f := \sqrt{\frac{e^x - 1}{x}}$$

```
> series(f,x=0,14);
```

¹Please note that *Maple* regularly does not sort its output to save computation time so that the Taylor coefficients might appear in wrong order.

$$\begin{aligned}
& 1 + x/4 + \frac{5x^2}{96} + \frac{x^3}{128} + \frac{79x^4}{92160} + \frac{3x^5}{40960} + \frac{71x^6}{12386304} + \frac{113x^7}{247726080} \\
& + \frac{3053x^8}{118908518400} + \frac{x^9}{22649241600} + \frac{17x^{10}}{930128855040} + \frac{19x^{11}}{744103084032} \\
& + \frac{935917x^{12}}{1218840851644416000} - \frac{20287103x^{13}}{43878270659198976000} + O(x^{14}) \\
> \text{expand}(\text{coeff}(\text{series}(g,z,14),z,13)); \\
& \frac{102672775873x}{1307674368000} + \frac{437x^{11}}{2179989504} - \frac{20287103x^{13}}{5356234211328000} \\
& + \frac{2299x^9}{15925248000} + \frac{3735911x^3}{54743040} + \frac{2031271x^5}{258048000} + \frac{2042249x^7}{5852528640}
\end{aligned}$$

By these computations both conjectures are disproved since the thirteenth coefficient of $f(x)$ is negative, and $B_{13}(x)$ also has a negative coefficient. The falsification of these Robertson's conjectures was published in 1991 [10]. Please notice that the reviewing process of my article almost 30 years ago was still very hard! The reviewers were complaining why we should believe in the output of a computer? Nowadays computer algebra computations are so common, and we know that they are safer than hand computations, that no reviewer will complain about such computations any more.

3. Holonomic Power Series

Sometimes one is not only interested in a Taylor polynomial approximation, but in the full Taylor series. Assume, given an expression $f(x)$ depending on the variable x , we would therefore like to compute a formula for the coefficient a_k of the power series

$$f(x) = \sum_{k=0}^{\infty} a_k x^k \quad (3.1)$$

representing $f(x)$. A well-known example of that type is given by

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}.$$

Here is an algorithm for this purpose [11].

Algorithm 3.1 (FPS (Koepf, 1992, [11])).

- **Input:** expression $f(x)$.
- **HolonomicDE:** Determine a holonomic differential equation DE (i.e. homogeneous and linear with polynomial coefficients) by computing the derivatives of $f(x)$ iteratively.
- **DEtoRE:** Convert DE to a holonomic recurrence equation RE for a_k .
- **RSolve:** Solve RE for a_k .

- **Output:** a_k resp. $\sum_{k=0}^{\infty} a_k x^k$.

Functions satisfying a holonomic DE and sequences satisfying a holonomic RE are also called holonomic. A function is holonomic iff it is the generating function of a holonomic sequence.

The above algorithm is embedded into the *Maple* system via the `convert(..., FormalPowerSeries)` command, and we present some examples:

> `convert(sin(x),FPS);`

$$\sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$$

> `convert(arcsin(x)^2,FPS);`

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

How does this algorithm work? Here are the details:

Algorithm 3.2 (HolonomicDE (Koepf, 1992, [11])).

- **Input:** *expression* $f(x)$.
- Iterate for $J = 1, 2, \dots$:
- Compute $c_0 f(x) + c_1 f'(x) + \dots + f^{(J)}(x)$ with still undetermined coefficients c_j .
- Sort this linear combination w.r.t. linearly independent functions over $\mathbb{Q}(x)$ and determine their coefficients $\in \mathbb{Q}(x)$.²
- Set these coefficients zero, and solve the corresponding linear system for the unknowns c_0, c_1, \dots, c_{J-1} .
- **Output:** $DE := c_0 f(x) + c_1 f'(x) + \dots + f^{(J)}(x) = 0$ (or else multiply by the common denominator of the c_j s).

Holonomic functions have interesting algebraic properties. The existence of a holonomic differential equation shows that the dimension of the vector space

$$V_f = \langle f(x), f'(x), f''(x), \dots \rangle$$

over the field of rational functions $\mathbb{Q}(x)$ is finite. This argument yields

Algorithm 3.3 ((Stanley, 1980, [19]), Maple (Salvy, Zimmermann, 1994, [18])).

- Let a function $f(x)$ be given by a holonomic differential equation DE1 of order n , and let a function $g(x)$ be given by a holonomic differential equation DE2 of order m .
- Then there are linear algebra algorithms showing that $f + g$ is holonomic of degree $\leq n + m$, and $f \cdot g$ is holonomic of degree $\leq n \cdot m$.
- Similar statements are valid for holonomic sequences.

²For details how to select the linearly independent functions, see [7].

By these algebraic properties the ring of holonomic functions has a so-called *normal form* which consists of the differential equation together with enough initial values. The same is true for the ring of holonomic sequences whose normal form consists of their recurrence equation together with enough initial values. Unfortunately, holonomic functions are not closed under division, an example of which is given by $\tan x = \frac{\sin x}{\cos x}$ (for details of the proof, see [12]).

The following computation computes the differential equation for $f(x) := \arcsin^2(x)$ and converts it towards the corresponding recurrence equation for its coefficients a_k for $f(x)$ given by (3.1):

```

> bind(FormalPowerSeries):
> DE:=HolonomicDE(arcsin(x)^2,F(x));
DE := { d/dx F(x) + 3x d^2/dx^2 F(x) + (x^2 - 1) d^3/dx^3 F(x), F(0) = 0, D(F)(0) = 0, (D^(2))(F)(0) = 2 }
> RE:=SimpleRE(arcsin(x)^2,x,a(k));
RE := (k + 1)^3 a(k + 1) - (k + 1)(k + 2)(k + 3) a(k + 3) = 0

```

which can be easily solved for a_k using the hypergeometric coefficient formula (3.3) that we will consider soon.

By computing their normal forms two functions can be identified as identical! The following computation shows, for example, the addition theorem of the sine function, completely automatically.

```

> HolonomicDE(sin(x+y),F(x));
{ d^2/dx^2 F(x) + F(x), F(0) = sin(y), D(F)(0) = cos(y) }
> HolonomicDE(sin(x)*cos(y)+cos(x)*sin(y),F(x));
{ d^2/dx^2 F(x) + F(x), F(0) = sin(y), D(F)(0) = cos(y) }

```

A very rich class of holonomic functions are the hypergeometric functions / series. The formal power series

$${}_pF_q \left(\begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix} \middle| z \right) = \sum_{k=0}^{\infty} A_k z^k = \sum_{k=0}^{\infty} \alpha_k,$$

whose summands $\alpha_k = A_k z^k$ have a rational term ratio (factored over \mathbb{Q} or, if necessary, over \mathbb{C})

$$\frac{\alpha_{k+1}}{\alpha_k} = \frac{A_{k+1} z^{k+1}}{A_k z^k} = \frac{(k + a_1) \cdots (k + a_p)}{(k + b_1) \cdots (k + b_q)} \frac{z}{(k + 1)},$$

is called the *generalized hypergeometric series*. For this reason, the summand $\alpha_k = A_k z^k$ of a hypergeometric series is called a *hypergeometric term*.

For the coefficients of the generalized hypergeometric function one gets the following formula using the shifted factorial (*Pochhammer symbol*) $(a)_k = a(a+1)\cdots(a+k-1)$

$${}_pF_q \left(\begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix} \middle| z \right) = \sum_{k=0}^{\infty} \frac{(a_1)_k \cdots (a_p)_k}{(b_1)_k \cdots (b_q)_k} \frac{z^k}{k!} .$$

Therefore every holonomic recurrence equation of first order defines a hypergeometric term which can be determined by the hypergeometric coefficient formula.

If the recurrence is brought into the form

$$\frac{\alpha_{k+1}}{\alpha_k} = \frac{(k+a_1)\cdots(k+a_p)}{(k+b_1)\cdots(k+b_q)} \frac{z}{(k+1)} ,$$

or, equivalently

$$(k+b_1)\cdots(k+b_q)(k+1)\alpha_{k+1} - (k+a_1)\cdots(k+a_p)z\alpha_k = 0 , \quad (3.2)$$

then α_k is given by

$$\alpha_k = \frac{(a_1)_k \cdots (a_p)_k}{(b_1)_k \cdots (b_q)_k} \frac{z^k}{k!} \cdot \alpha_0 . \quad (3.3)$$

The above method provides an algorithm to detect the hypergeometric representation for

$$S = \sum_{k=0}^{\infty} \alpha_k .$$

Algorithm 3.4 (Sumtohyper (Koepf, 1998, [13])).

- **Input:** α_k , a hypergeometric term w.r.t. k .
- *Compute*

$$r_k := \frac{\alpha_{k+1}}{\alpha_k} \in \mathbb{Q}(k) .$$

- *Factorize r_k .*
- **Output:** *Reading off upper and lower parameters and the argument z from representation (3.2), and computing an initial value yields the corresponding hypergeometric series.*

As an example, for

$$\sin x = \sum_{k=0}^{\infty} \alpha_k = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} ,$$

we get

$$r_k = \frac{\alpha_{k+1}}{\alpha_k} = -\frac{x^2}{2(k+1)(2k+3)} = \frac{1}{(k+3/2)(k+1)} \left(-\frac{x^2}{4} \right) \in \mathbb{Q}(x, k)$$

and

$$\alpha_0 = x .$$

Therefore

$$\sin x = x \cdot {}_0F_1\left(\begin{matrix} - \\ 3/2 \end{matrix} \middle| -\frac{x^2}{4}\right).$$

After loading the `hsum` package [13]

```
> read "hsum17.mpl";
'Package "Hypergeometric Summation", Maple V - Maple 17'
'Copyright 1998-2014, Wolfram Koepf, University of Kassel'
```

this computation is done automatically by the *Maple* command

```
> Sumtohyper((-1)^k*x^(2*k+1)/(2*k+1)!,k);
xHypergeom([], [3/2], -1/4*x^2)
```

4. Fasenmyer's Algorithm

The sum evaluation

$$\sum_{k=0}^n \binom{n}{k} = 2^n$$

is well-known. However, the similar, but more advanced identities

$$\sum_{k=0}^n k \binom{n}{k} = n 2^{n-1} \quad \text{and} \quad \sum_{k=0}^n \binom{n}{k}^2 = \binom{2n}{n}$$

might be less known. Our question is: How can we compute the right hand sides, given the sums on the left?

In general, consider the sum

$$S_n := \sum_{k=0}^n F(n, k) \quad \text{with} \quad \frac{F(n+1, k)}{F(n, k)}, \frac{F(n, k+1)}{F(n, k)} \in \mathbb{Q}(n, k),$$

where $F(n, k)$ therefore is a *hypergeometric term* w.r.t. k and n . Fasenmyer's idea ([4], [5]) is to find first a *k-free recurrence for the summand* $F(n, k)$:

$$\sum_{j=0}^J \sum_{i=0}^I a_{ij} F(n+j, k+i) = 0 \quad \text{for some } a_{ij} \in \mathbb{Q}(n).$$

This can be done by linear algebra!

If successful, since the recurrence is *k-free*, it can be summed from $k = -\infty, \dots, \infty$ to get a *pure recurrence for the sum* S_n .

For

$$F(n, k) = \binom{n}{k} \quad \text{and} \quad S_n = \sum_{k=0}^n \binom{n}{k} = \sum_{k=-\infty}^{\infty} \binom{n}{k}$$

the first step yields the *Pascal triangle identity*

$$F(n+1, k+1) = F(n, k+1) + F(n, k).$$

If we sum this relation from $k = -\infty, \dots, \infty$, we clearly get

$$S_{n+1} = S_n + S_n = 2S_n.$$

Since $S_0 = 1$, we finally have $S_n = 2^n$. The corresponding *Maple* command is given by

```
> fassenmyer(binomial(n,k),k,S(n),1);
          S(n+1) - 2S(n) = 0
```

or—knowing the result—by

```
> fassenmyer(binomial(n,k)/2^n,k,S(n),1);
          S(n+1) - S(n) = 0
```

and other examples are

```
> fassenmyer(k*binomial(n,k),k,S(n),1);
          nS(n+1) - 2(n+1)S(n) = 0
> fassenmyer(k*binomial(n,k)/(n*2^(n-1)),k,S(n),1);
          S(n+1) - S(n) = 0
> fassenmyer(binomial(n,k)^2,k,S(n),2);
          (n+2)S(n+2) - 2S(n+1)(2n+3) = 0
> fassenmyer(binomial(n,k)^2/binomial(2*n,n),k,S(n),2);
          S(n+2) - S(n+1) = 0
```

For more details, see [13, Chapter 4].

5. Gosper's Algorithm

Given a_k , a sequence s_k is called an *antidifference* of a_k if

$$a_k = s_{k+1} - s_k.$$

Given such an antidifference, then by telescoping the sum can be computed with arbitrary lower and upper bounds:

$$\begin{aligned} \sum_{k=m}^n a_k &= (s_{n+1} - s_n) + (s_n - s_{n-1}) + \cdots + (s_{m+1} - s_m) \\ &= s_{n+1} - s_m. \end{aligned}$$

Gosper's algorithm computes a *hypergeometric term* antidifference s_k for a hypergeometric term a_k , i.e.

$$\frac{a_{k+1}}{a_k} \in \mathbb{Q}(k).$$

Here are the computational steps: Given a_k with

$$\frac{a_{k+1}}{a_k} = \frac{u_k}{v_k}$$

with polynomials $u_k, v_k \in \mathbb{Q}[k]$, $\gcd(u_k, v_k) = 1$, it is easy to show that

$$s_k = \frac{g_k}{h_k} a_k$$

for certain polynomials $g_k, h_k \in \mathbb{Q}[k]$.

Whereas the denominator h_k can be written down explicitly, for g_k one gets the inhomogeneous recurrence equation

$$h_k u_k g_{k+1} - h_{k+1} v_k g_k = h_k h_{k+1} v_k .$$

To compute g_k one first finds a *degree bound* for g_k . The final step uses *linear algebra* to compute the coefficients of g_k by equating coefficients.

Gosper's algorithm [13, Chapter 5] was maybe the first algorithm which would not have been found without computer algebra. In his paper [6], Gosper stated:

Without the support of MACSYMA and its developers, I could not have collected the experiences necessary to provoke the conjectures that led to this algorithm.

The next two examples for an application of Gosper's algorithm

> `gosper((-1)^k*binomial(n,k),k);`

$$-\frac{k(-1)^k \binom{n}{k}}{n}$$

> `gosper(1/(k+1),k);`

Error, (in gosper) No hypergeometric term antidifference exists

show that

- An antidifference of $a_k = (-1)^k \binom{n}{k}$ is $s_k = -\frac{k}{n} a_k$.
- The harmonic numbers $H_n = \sum_{k=1}^n \frac{1}{k}$ cannot be written as hypergeometric term.

For more details, see [12, Chapter 11] and [13, Chapter 5].

6. Zeilberger's Algorithm

Zeilberger's algorithm deals again—like Fasenmyer's—with definite sums of the form

$$S_n := \sum_{k=0}^n F(n, k) \quad \text{with} \quad \frac{F(n+1, k)}{F(n, k)}, \frac{F(n, k+1)}{F(n, k)} \in \mathbb{Q}(n, k).$$

Applying Gosper to $F(n, k)$ w. r. t. k (by telescoping) always generates the identity $S_n = 0$. Therefore this method generally cannot be applied.

Doron Zeilberger's idea [13, Chapter 7]: Apply Gosper instead to

$$a_k := F(n, k) + \sum_{j=1}^J \sigma_j F(n+j, k) \quad \text{for some } J \geq 1 .$$

Doron Zeilberger's idea generates the following algorithm.

Algorithm 6.1 (Zeilberger, 1990, [20]).

- Apply Gosper's algorithm to

$$a_k := F(n, k) + \sum_{j=1}^J \sigma_j F(n + j, k) \quad \text{for some } J \geq 1.$$

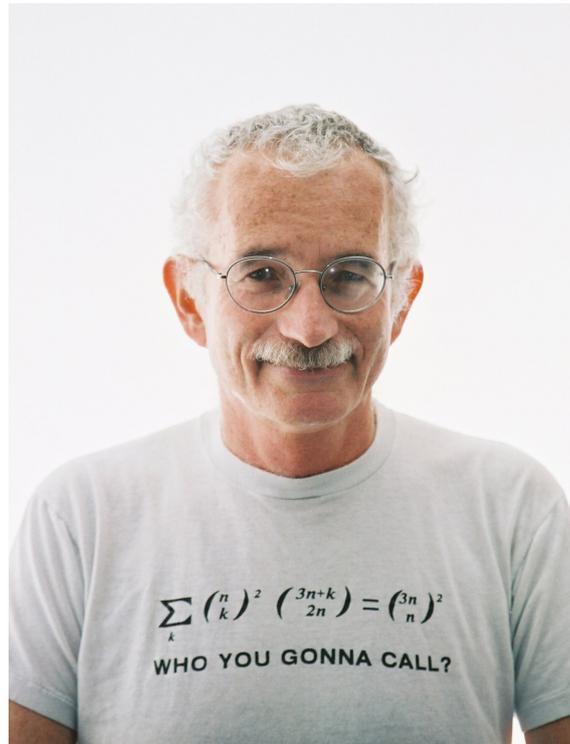
- In the final linear algebra step, solve not only for the coefficients of g_k , but also for σ_j ($j = 1, \dots, J$).
- If successful, then this algorithm generates the recurrence equation

$$S_n + \sum_{j=1}^J \sigma_j S_{n+j} = 0$$

with rational functions $\sigma_j \in \mathbb{Q}(n)$ ($j = 1, \dots, J$) for S_n .

- Multiplication with the common denominator yields a holonomic recurrence equation for S_n .

If the resulting recurrence equation is of first order, then an application of Algorithm 3.4 yields the hypergeometric term solution.



We give an example:

```
> sumrecursion(binomial(n,k)^2,k,S(n));
      -(n+1)S(n+1)+2(2n+1)S(n)=0
```

Since the resulting recurrence equation is of first order, we can solve it explicitly by the command

```
> closedform(binomial(n,k)^2,k,n);
      (2n)!
      (n!)^2
```

The next example solves the problem given on Doron Zeilberger's T-shirt.

```
> closedform(binomial(n,k)^2*binomial(3*n+k,2*n),k,S(n));
      (pochhammer(2/3,n))^2 (pochhammer(1/3,n))^2 (4^n)^2 (729/16)^n
      ((2n)!)^2
```

Of course this result looks rather different from the right hand side on Zeilberger's T-shirt although the results are equivalent. But since we know the desired result, we can prove it easily by the computation

```
> closedform(binomial(n,k)^2*binomial(3*n+k,2*n)/
      binomial(3*n,n)^2,k,n);
```

1

For more details, see [13, Chapter 7].

7. CAOP

CAOP [3] is a web tool for calculating formulas for orthogonal polynomials belonging to the Askey-Wilson scheme using Maple. The implementation of CAOP was originally done by René Swarttouw as part of the Askey-Wilson Scheme Project performed at RIACA in Eindhoven in 2004. The present site <http://www.caop.org/> is a completely revised version of this project which has been done by Torsten Sprenger under my supervision in 2012 and is maintained at the University of Kassel.

If we select, for example, the *Laguerre polynomials*, the CAOP system will let us know that they are defined as

$$L_n^{(\alpha)}(x) = \sum_{k=0}^n \binom{n+\alpha}{n-k} \frac{(-x)^k}{k!} = \frac{(\alpha+1)_n}{n!} {}_1F_1 \left(\begin{matrix} -n \\ \alpha+1 \end{matrix} \middle| x \right),$$

both in binomial series and in hypergeometric notation. The system will immediately start to compute both a recurrence equation for $L_n^{(\alpha)}(x)$ w.r.t. n and a differential equation w.r.t. x using Zeilberger type algorithms. All computations are done on a server at the University of Kassel. The complete *Maple* codes which

are used can be displayed. In principle, one can even multiply $P_n(x) = L_n^{(\alpha)}(x)$ by an arbitrary factor and redo the computations. For $L_n^{(\alpha)}(x)$, one gets

$$x P_n''(x) + (-x + \alpha + 1) P_n'(x) + n P_n(x) = 0$$

and

$$(n+2) P_{n+2}(x) - (-x + 2n + \alpha + 3) P_{n+1}(x) + (n + \alpha + 1) P_n(x) = 0.$$

8. Petkovšek's and van Hoeij's Algorithm

Assume, S_n is a hypergeometric term, however Zeilberger's algorithm generates not the recurrence of first order, but a holonomic recurrence of higher order

$$RE : \sum_{j=0}^J P_j(n) S_{n+j} = 0$$

with polynomial coefficients $P_j \in \mathbb{Q}[n]$.

Then *Petkovšek's algorithm* ([15], see [13, Chapter 9])—which is quite similar to Gosper's—finds every solution which is a linear combination of hypergeometric terms. Combining Zeilberger's with Petkovšek's algorithm leads to a decision procedure for hypergeometric term summation.

We want to compute

$$S_n = \sum_{k=0}^n \frac{\binom{3k+1}{k} \binom{3n-3k}{n-k}}{3k+1}.$$

First we realize that the summation bounds $k = 0, \dots, n$ are the natural ones since $-n$ is an upper parameter of the corresponding hypergeometric representation, shown by the computation

```
> summand:=binomial(3*k+1,k)*binomial(3*n-3*k,n-k)/(3*k+1);
```

$$summand := \frac{\binom{3k+1}{k} \binom{3n-3k}{n-k}}{3k+1}$$

```
> Sumtohyper(summand,k);
```

$$\frac{\Gamma(3n+1) \text{Hypergeom}([-n, 1/3, 2/3, -n+1/2], [-n+2/3, -n+1/3, 3/2], 1)}{\Gamma(2n+1) \Gamma(n+1)}$$

By

```
> RE:=sumrecursion(summand,k,S(n));
```

$$RE := 4(2n+5)(2n+3)(n+2)S(n+2) - 12(2n+3)(9n^2+27n+22)S(n+1) + 81(3n+2)(3n+4)(n+1)S(n) = 0$$

we therefore have computed a recurrence equation for S_n . Applying Petkovšek's algorithm yields two linearly independent hypergeometric term solutions with term ratios S_{n+1}/S_n given by

```
> rechyper(RE,S(n));
```

$$\left\{ 27 \frac{n+1}{4n+6}, 3/2 \frac{(3n+2)(3n+4)}{(2n+3)(n+1)} \right\}$$

Therefore S_n must be a linear combination of the corresponding hypergeometric terms

```
> result:=a*pochhammer(1,n)/pochhammer(3/2,n)*(27/4)^n
> +b*pochhammer(2/3,n)*pochhammer(4/3,n)/
> (pochhammer(1,n)*pochhammer(3/2,n))*(27/4)^n;
```

$$result := \frac{a \operatorname{pochhammer}(1, n)}{\operatorname{pochhammer}(3/2, n)} \left(\frac{27}{4}\right)^n + \frac{b \operatorname{pochhammer}(2/3, n) \operatorname{pochhammer}(4/3, n)}{\operatorname{pochhammer}(1, n) \operatorname{pochhammer}(3/2, n)} \left(\frac{27}{4}\right)^n$$

Using two initial values, we can compute a and b

```
> init0:=eval(add(subs(n=0,summand),k=0..0));
      init0 := 1
> init1:=eval(add(subs(n=1,summand),k=0..1));
      init1 := 4
> sol:=solve({eval(result,n=0)=init0,eval(result,n=1)=init1},{a,b});
      sol := {a = 0, b = 1}
```

so that we finally get for S_n

```
> res1:=subs(sol,result);
```

$$res1 := \frac{\operatorname{pochhammer}(2/3, n) \operatorname{pochhammer}(4/3, n)}{\operatorname{pochhammer}(1, n) \operatorname{pochhammer}(3/2, n)} \left(\frac{27}{4}\right)^n$$

We claim that

$$S_n = \binom{3n+1}{n},$$

which is proved by

```
> res2:=binomial(3*n+1,n);
```

$$res2 := \binom{3n+1}{n}$$

```
> simplify(res1/res2);
```

1

However, Petkovšek's algorithm is rather inefficient. Mark van Hoeij ([8], see [13, Chapter 9]), gave a much faster algorithm for the same purpose, based on *the singularity behavior* of the constituting Γ functions that occur in the solutions. Details can be found in [13, Chapter 9].

Mark van Hoeij implemented his algorithm in *Maple*. Applying this implementation to the same example as above, we get

```
> res:=LREtools[hypergeomsols](RE,S(n),{S(0)=1,S(1)=4});
```

$$res := 3/4 \frac{\sqrt{3}\Gamma(4/3+n)\Gamma(2/3+n)}{\sqrt{\pi}\Gamma(n+3/2)\Gamma(n+1)} \left(\frac{27}{4}\right)^n$$

which gives—as announced—the result in terms of the Gamma function. We can again show that the result constitutes the same term as above by

```
> simplify(res/res2);
```

1

9. Hypergeometric Identities

All examples from this section can be found in [13, Chapter 7].

Clausen's formula gives the cases when a Clausen ${}_3F_2$ function is the square of a Gauss ${}_2F_1$ function:

$$\left({}_2F_1\left(\begin{matrix} a, b \\ a+b+\frac{1}{2} \end{matrix} \middle| x\right)\right)^2 = {}_3F_2\left(\begin{matrix} 2a, 2b, a+b \\ 2a+2b, a+b+\frac{1}{2} \end{matrix} \middle| x\right).$$

The right hand side can be detected from the left hand side by Zeilberger's algorithm. Using the Cauchy product, we get for the coefficient of the left hand square:

```
> sumrecursion(hyperterm([a,b],[a+b+1/2],x,k)*
> hyperterm([a,b],[a+b+1/2],x,n-k),k,S(n));
-(n+1)(n+2b+2a)(2a+2b+1+2n)S(n+1)
+2x(n+2b)(2a+n)(a+b+n)S(n)=0
```

which is of first order so that we can easily convert the summand into the corresponding hypergeometric term by the command

```
> Closedform(hyperterm([a,b],[a+b+1/2],x,k)*
> hyperterm([a,b],[a+b+1/2],x,n-k),k,n);
Hyperterm([2a,2b,a+b],[2b+2a,a+b+1/2],x,n)
```

This proves Clausen's formula from left to right. The question remains, how we can get the full identity without prior knowledge about the parameter choice? This can be done using differential equations, by solving a non-linear system. The generic function ${}_2F_1(a, b; c; x)$ satisfies the differential equation

```
> DE2F1:=sumdiffeq(hyperterm([a,b],[c],x,k),k,C(x));
DE2F1 := \left(\frac{d^2}{dx^2}C(x)\right)x(x-1) + (ax+bx-c+x)\frac{d}{dx}C(x) + abC(x) = 0
```

and therefore its square satisfies³

```
> DE2F1square:=map(factor,gfun['diffeq*diffeq'](DE2F1,DE2F1,C(x))):
```

³We don't print the lengthy outputs

On the other hand, the generic function ${}_3F_2(A, B, C; D, E; x)$ satisfies the differential equation

$$> \text{DE3F2} := \text{op}(1, \text{sumdiff}(\text{hyperterm}([A, B, C], [D, E], x, k), k, C(x))):$$

Since the coefficient of the highest derivative in the left hand differential equation is $(x-1)$ times the coefficient of the highest derivative in the right hand differential equation, we multiply the right hand side by $(x-1)$ and subtract to get zero.

$$\begin{aligned} > \text{pol} &:= \text{expand}(\text{subs}(\text{diff}(C(x), x^3) = y^3, \text{diff}(C(x), x^2) = y^2, \\ > \text{diff}(C(x), x) = y, C(x) = 1, \text{DE3F2} * (x-1) - \text{DE2F1square})): \end{aligned}$$

Next, by equating the coefficients we can set all coefficients zero, assuming that a and b are arbitrary. Therefore we solve for the remaining variables $c, A, B, C, D,$ and E and get

$$\begin{aligned} > \text{solve}(\{\text{coeffs}(\text{expand}(\text{pol}), [x, y])\}, \{c, A, B, C, D, E\}); \\ \{A = 2b, B = a + b, C = 2a, D = 2b + 2a, E = a + b + 1/2, c = a + b + 1/2\}, \\ \{A = 2a, B = a + b, C = 2b, D = 2b + 2a, E = a + b + 1/2, c = a + b + 1/2\}, \\ \{A = 2b, B = a + b, C = 2a, D = a + b + 1/2, E = 2b + 2a, c = a + b + 1/2\}, \\ \{A = 2a, B = a + b, C = 2b, D = a + b + 1/2, E = 2b + 2a, c = a + b + 1/2\}, \\ \{A = 2b, B = 2a, C = a + b, D = 2b + 2a, E = a + b + 1/2, c = a + b + 1/2\}, \\ \{A = a + b, B = 2a, C = 2b, D = 2b + 2a, E = a + b + 1/2, c = a + b + 1/2\}, \\ \{A = 2b, B = 2a, C = a + b, D = a + b + 1/2, E = 2b + 2a, c = a + b + 1/2\}, \\ \{A = a + b, B = 2a, C = 2b, D = a + b + 1/2, E = 2b + 2a, c = a + b + 1/2\}, \\ \{A = a + b, B = 2b, C = 2a, D = 2b + 2a, E = a + b + 1/2, c = a + b + 1/2\}, \\ \{A = 2a, B = 2b, C = a + b, D = 2b + 2a, E = a + b + 1/2, c = a + b + 1/2\}, \\ \{A = a + b, B = 2b, C = 2a, D = a + b + 1/2, E = 2b + 2a, c = a + b + 1/2\}, \\ \{A = 2a, B = 2b, C = a + b, D = a + b + 1/2, E = 2b + 2a, c = a + b + 1/2\} \end{aligned}$$

Please notice that all twelve solutions are equivalent since the upper parameters A, B, C as well as the lower parameters D, E can be permuted arbitrarily.

Dougall (1907) found the following astonishing identity

$$\begin{aligned} & {}_7F_6 \left(\begin{matrix} a, 1 + \frac{a}{2}, b, c, d, 1 + 2a - b - c - d + n, -n \\ \frac{a}{2}, 1 + a - b, 1 + a - c, 1 + a - d, b + c + d - a - n, 1 + a + n \end{matrix} \middle| 1 \right) \\ &= \frac{(1+a)_n (a+1-b-c)_n (a+1-b-d)_n (a+1-c-d)_n}{(1+a-b)_n (1+a-c)_n (1+a-d)_n (1+a-b-c-d)_n}. \end{aligned}$$

Again, if one knows the left-hand side, Zeilberger's algorithm generates the right-hand side immediately.

$$\begin{aligned} > \text{Closedform}(\text{hyperterm}([a, 1+a/2, b, c, d, 1+2*a-b-c-d+n, -n], \\ > [a/2, 1+a-b, 1+a-c, 1+a-d, b+c+d-a-n, 1+a+n], 1, k), k, n); \\ & \text{Hyperterm}([a+1, -c+1+a-b, 1+a-d-b, 1+a-d-c, 1], \\ & [1+a-d, -b+1+a, -c+1+a, -b-c-d+a+1], 1, n) \end{aligned}$$

Apéry (1978) proved that

$$\zeta(3) = \sum_{j=1}^{\infty} \frac{1}{j^3}$$

is irrational. In his proof, he used a holonomic recurrence equation for the so-called *Apéry numbers*

$$A_n := \sum_{k=0}^n \binom{n}{k}^2 \binom{n+k}{k}^2.$$

Again, Zeilberger's algorithm detects their recurrence equation immediately.

```
> sumrecursion(binomial(n,k)^2*binomial(n+k,k)^2,k,A(n));
(n+2)^3 A(n+2) - (2n+3)(17n^2+51n+39)A(n+1) + (n+1)^3 A(n) = 0
```

Shortly before Zeilberger's algorithm came up, Perlstadt [14] was able to publish a paper deriving recurrence equations for

$$S_n^{(r)} := \sum_{k=0}^n \binom{n}{k}^r$$

for $r = 5$ and 6 . Using Zeilberger's algorithm, this is now "trivial".

```
> TIME:=time();
> sumrecursion(binomial(n,k)^5,k,S(n));
> time()-TIME;
(55n^2+143n+94)(n+3)^4 S(n+3)
- (1155n^6+14553n^5+75498n^4+205949n^3+310827n^2+245586n+79320) S(n+2)
- (19415n^6+205799n^5+900543n^4+2082073n^3+2682770n^2+1827064n+514048) S(n+1)
+ 32(55n^2+253n+292)(n+1)^4 S(n) = 0
0.140
```

For the next examples, we omit again the lengthy outputs.

```
> TIME:=time();
> sumrecursion(binomial(n,k)^6,k,S(n));
> time()-TIME;
0.125

> TIME:=time();
> sumrecursion(binomial(n,k)^10,k,S(n));
> time()-TIME;
3.016
```

Even for power 10, our computation needs only few seconds although the resulting recurrence equation of order 5 has *very complicated* coefficients.

10. Generating Functions

Assume, we want to compute the *generating function* of the Legendre polynomials

$$P_n(x) = \sum_{k=0}^n \binom{n}{k} \binom{-n-1}{k} \left(\frac{1-x}{2}\right)^k,$$

namely

$$F(z) := \sum_{n=0}^{\infty} P_n(x) z^n.$$

How can we approach this question?

The recurrence equation for the coefficients (the Legendre polynomials) can be converted into a differential equation for the generating function. In certain cases, this differential equation can be solved.

```
> RE:=sumrecursion(binomial(n,k)*binomial(-n-1,k)*((1-x)/2)^k,k,P(n));
RE := (n + 2) P(n + 2) - x(2n + 3) P(n + 1) + (n + 1) P(n) = 0
> DE:=gfun[rectodiffeq]({RE,P(0)=1,P(1)=x},P(n),F(z));
DE := {(-x + z) F(z) + (-2xz + z^2 + 1) d/dz F(z), F(0) = 1}
> dsolve(DE,F(z));
```

$$F(z) = \left(\sqrt{-2xz + z^2 + 1}\right)^{-1}$$

Assume, we want to compute the *exponential generating function* of the Legendre polynomials

$$G(z) := \sum_{n=0}^{\infty} \frac{P_n(x)}{n!} z^n.$$

How can we approach this question?

Although, in principle, one can use the same method here (since *Maple's dsolve* is really very powerful!), in this case, we apply a different approach. For this purpose, we use one particular hypergeometric representation

$$P_n(x) = x^n {}_2F_1\left(\begin{matrix} -\frac{n}{2}, -\frac{n-1}{2} \\ 1 \end{matrix} \middle| 1 - \frac{1}{x^2}\right)$$

of the Legendre polynomials. Then after interchanging the order of summation, Zeilberger's algorithm yields the generating function:

$$G(z) = e^{xz} {}_0F_1\left(\begin{matrix} - \\ 1 \end{matrix} \middle| \frac{z^2(x-1)^2}{4}\right) = e^{xz} I_0(z\sqrt{1-x^2}),$$

$I_n(z)$ denoting the modified Bessel function, by the computations

```
> summand:=x^n*hyperterm([-n/2,(1-n)/2],[1],1-1/x^2,k)/n!*z^n;
```

$$\text{summand} := \frac{x^n \text{pochhammer}(-n/2, k) \text{pochhammer}(1/2 - n/2, k) z^n}{(k!)^2 n!} (1 - x^{-2})^k$$

After interchanging the order of summation, we get for the summand of the outer sum the first order recurrence

> RE:=sumrecursion(summand,n,S(k));

$$RE := -4(k+1)^2 S(k+1) + (x-1)(x+1)z^2 S(k) = 0$$

and the initial value

> init:=sum(subs(k=0,summand),n=0..infinity);

$$\text{init} := e^{xz}$$

> outersummand:=rsolve({RE,S(0)=init},S(k));

$$\text{outersummand} := \frac{(1/4(x^2-1)z^2)^k e^{xz}}{(\Gamma(k+1))^2}$$

> hyper:=Sumtohyper(outersummand,k);

$$\text{hyper} := e^{xz} \text{Hypergeom}([], [1], 1/4(x-1)(x+1)z^2)$$

> convert(subs(Hypergeom=hypergeom,hyper),StandardFunctions);

$$e^{xz} I_0(\sqrt{x-1}\sqrt{x+1}z)$$

11. Almkvist-Zeilberger Algorithm

Similarly to definite summation, Almkvist and Zeilberger [1] generated algorithms to compute holonomic recurrence and differential equations for definite integrals instead of series. In this case

$$S_n = \int_a^b F(n, t) dt$$

or

$$S(x) = \int_a^b F(x, t) dt$$

with F , hyperexponential w. r. t. the continuous variables x and t and hypergeometric w. r. t. the discrete variable n , for suitably chosen bounds a and b .

As an example, we would like to compute the definite integral

$$J(x) := \int_0^\infty \exp\left(-\frac{x^2}{t^2} - t^2\right) dt .$$

Almkvist-Zeilberger's algorithm shows that $J(x)$ satisfies the holonomic differential equation

$$J''(x) - 4J(x) = 0 .$$

From this differential equation, using two initial values, we can deduce

$$J(x) = \frac{\sqrt{\pi}}{2} e^{-2x} .$$

By the Cauchy integral formula

$$f^{(n)}(x) = \frac{n!}{2\pi i} \int_{\gamma} \frac{f(t)}{(t-x)^{n+1}} dt, \quad (11.1)$$

a *Rodrigues formula* concerning the n^{th} derivative can be written in terms of a definite integral.

Therefore, using the Almkvist-Zeilberger algorithm, one can deduce from the following Rodrigues representation

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} ((x^2 - 1)^n) \quad (11.2)$$

again the recurrence and differential equations of the Legendre polynomials by the computations⁴

```
> rodriguesrecursion(1/(2^n*n!), (x^2-1)^n, x, P(n));
      (n + 2) P(n + 2) - x(2n + 3) P(n + 1) + (n + 1) P(n) = 0
> rodriguesdiffeq(1/(2^n*n!), (x^2-1)^n, n, P(x));
      (d^2/dx^2 P(x)) (x - 1)(x + 1) + 2x d/dx P(x) - n(n + 1) P(x) = 0
```

therefore identifying those polynomials given by (11.2) as the Legendre polynomials.

If $F(z)$ is the generating function of the sequence $a_n f_n(x)$, i.e.

$$F(z) = \sum_{n=0}^{\infty} a_n f_n(x) z^n,$$

then by Cauchy's integral formula and Taylor's theorem, we have

$$f_n(x) = \frac{1}{a_n} \cdot \frac{F^{(n)}(0)}{n!} = \frac{1}{a_n} \cdot \frac{1}{2\pi i} \int_{\gamma} \frac{F(t)}{t^{n+1}} dt.$$

Therefore, if the generating function is hyperexponential, we can again apply the Almkvist-Zeilberger algorithm to deduce recurrence and differential equations. Starting from the generating function

$$F(z) = \frac{1}{\sqrt{1 - 2xz + z^2}}$$

we can therefore compute the recurrence and differential equations for the Legendre polynomials

```
> F:=1/sqrt(1-2*x*z+z^2);
      F := (sqrt(-2*x*z + z^2 + 1))^-1
> RE:=GFrecursion(F,1,z,P(n));
      RE := (n + 2) P(n + 2) - x(2n + 3) P(n + 1) + (n + 1) P(n) = 0
```

⁴The commands `rodriguesrecursion` and `rodriguesdiffeq` take representation (11.1) into consideration and invoke the Almkvist-Zeilberger algorithms. For details, see [13], Chapter 13.

> DE:=GFdiff(F,1,z,n,P(x));

$$DE := \left(\frac{d^2}{dx^2} P(x) \right) (x-1)(x+1) + 2x \frac{d}{dx} P(x) - n(n+1)P(x) = 0$$

from their generating function.

These and many more examples are given in detail in [13, Chapter 13].

12. Basic Hypergeometric Series

As can be seen on the CAOP web site, the Askey-Wilson scheme consists of

1. the classical continuous orthogonal polynomial systems (OPS),
2. the classical discrete OPS on the lattice \mathbb{Z} (or $\mathbb{N}_{\geq 0}$),
3. the Askey-Wilson polynomials on a quadratic lattice,
4. the OPS of the Hahn class on the basic lattice $q^{\mathbb{Z}}$,
5. and the Askey-Wilson polynomials on a q -quadratic lattice.

All details can be found in [9]. The first three types are all represented by hypergeometric series, whereas the last two have *basic hypergeometric* representations. For this purpose, one defines the basic hypergeometric series by

$${}_r\phi_s \left(\begin{matrix} a_1, a_2, \dots, a_r \\ b_1, b_2, \dots, b_s \end{matrix} \middle| q; x \right) := \sum_{k=0}^{\infty} \frac{(a_1, a_2, \dots, a_r; q)_k}{(b_1, b_2, \dots, b_s; q)_k} \frac{x^k}{(q; q)_k} \left((-1)^k q^{\binom{k}{2}} \right)^{1+s-r}$$

where $(a_1, a_2, \dots, a_r; q)_k$ is a short form for the product $\prod_{j=1}^r (a_j; q)_k$, and

$$(a; q)_k := \begin{cases} \prod_{j=0}^{k-1} (1 - a q^j) & \text{if } k > 0 \\ 1 & \text{if } k = 0 \\ \prod_{j=1}^{|k|} (1 - a q^{-j})^{-1} & \text{if } k < 0 \\ \prod_{j=0}^{\infty} (1 - a q^j) & \text{if } k = \infty \text{ (if } |q| < 1) \end{cases}$$

denotes the *q-Pochhammer symbol*. The relation

$${}_2\phi_1 \left(\begin{matrix} q^{-n}, b \\ c \end{matrix} \middle| q; \frac{c q^n}{b} \right) = \frac{(c/b; q)_n}{(c; q)_n}$$

is a q -variant of the *Chu-Vandermonde Identity*

$$\sum_{k=0}^N \binom{m}{k} \binom{n}{N-k} = \binom{n}{N} {}_2F_1 \left(\begin{matrix} -m, -N \\ n+1-N \end{matrix} \middle| 1 \right) = \binom{m+n}{N}.$$

Applying Zeilberger's algorithm to the latter and its q -variant to the former yields these results from left to right.

> result1:=closedform(binomial(m,k)*binomial(n,N-k),k,N);

```

result1 := pochhammer(-n-m, N) (-1)^N / N!
> result2:=closedform(binomial(m,k)*binomial(n,N-k)/
> binomial(m+n,N), k, N);
result2 := 1
> convert(Sumtohyper(binomial(m,k)*binomial(n,N-k), k), binomial);
Hypergeom([-N, -m], [n+1-N], 1) \binom{n}{N}

```

We load the qsum package (see [2] and [13])

```

> read "qsum17.mpl";
'Package "q-Hypergeometric Summation", Maple V-17'
'Copyright 1998-2013, Harald Boeing,
Wolfram Koepf & Torsten Sprenger, University of Kassel'

```

and get using the q -variant of Zeilberger's algorithm ([13, Chapter 7])

```

> RE:=qsumrecursion(qphihyperterm([q^(-n), b], [c], q, c*q^n/b, k),
> q, k, CV(n));
RE := -b(cq^n - q) CV(n) + (cq^n - bq) CV(n-1) = 0
> qsumrecursion(qphihyperterm([q^(-n), b], [c], q, c*q^n/b, k), q, k,
> CV(n), rec2qhyper=true);
[CV(n) = 1 / qpochhammer(c, q, n) qpochhammer(c/b, q, n), 0 ≤ n]
> qsumrecursion(qphihyperterm([q^(-n), b], [c], q, c*q^n/b, k) /
> (qpochhammer(c/b, q, n) / qpochhammer(c, q, n)), q, k, CV(n),
> rec2qhyper=true);

```

$$[CV(n) = 1, 0 \leq n]$$

As examples of OPS of the q -Hahn class, we consider the *Little* and *Big q -Legendre polynomials*:

$$p_n(x|q) = {}_2\phi_1 \left(\begin{matrix} q^{-n}, q^{n+1} \\ q \end{matrix} \middle| q; qx \right)$$

and

$$P_n(x; c; q) = {}_3\phi_2 \left(\begin{matrix} q^{-n}, q^{n+1}, x \\ q, cq \end{matrix} \middle| q; q \right).$$

Using again the q -variant of Zeilberger's algorithm, we get for these OPS

```

> LqL:=qphihyperterm([q^(-n), q^(n+1)], [q], q, q*x, k);
LqL := qpochhammer(q^(-n), q, k) qpochhammer(q^(n+1), q, k) (qx)^k /
(qpochhammer(q, q, k))^2
> qsumrecursion(LqL, q, k, p(n));

```

$$\begin{aligned}
& q^n (q^n - 1) (q^n + q) p(n) + (q^{2n} - q) (q^{2n} x + q^{n+1} x + q^n x + qx - 2q^n) p(n-1) \\
& + (q^n + 1) (q^n - q) q^n p(n-2) = 0 \\
& > \text{BqL} := \text{qphihyperterm}([\hat{q}(-n), \hat{q}(n+1), x], [q, c*q], q, q, k); \\
\text{BqL} := & \frac{\text{qpochhammer}(q^{-n}, q, k) \text{qpochhammer}(q^{n+1}, q, k) \text{qpochhammer}(x, q, k) q^k}{(\text{qpochhammer}(q, q, k))^2 \text{qpochhammer}(cq, q, k)} \\
& > \text{qsumrecursion}(\text{BqL}, q, k, P(n)); \\
& (q^n + q) q (q^n - 1) (cq^n - 1) P(n) \\
& + (q^{2n} - q) (q^{2n} x - 2cq^{n+1} + q^{n+1} x - 2q^{n+1} + q^n x + qx) P(n-1) \\
& - (q^n + 1) (q^n - q) (-cq + q^n) q^n P(n-2) = 0
\end{aligned}$$

References

- [1] Almkvist, G. and Zeilberger, D.: The method of differentiating under the integral sign. *J. Symbolic Computation* **10**, 1990, 571–591.
- [2] Böing, H. and Koepf, W.: Algorithms for q -hypergeometric summation in computer algebra. *J. Symbolic Computation* **28**, 1999, 777–799.
- [3] CAOP: Computer algebra and orthogonal polynomials: www.caop.org.
- [4] Fasenmyer, M. C.: *Some Generalized Hypergeometric Polynomials*. PhD thesis, University of Michigan, 1945.
- [5] Fasenmyer, M. C.: A note on pure recurrence relations. *Amer. Math. Monthly* **56**, 1949, 14–17.
- [6] Gosper Jr., R. W.: Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA* **75**, 1978, 40–42.
- [7] Gruntz, D. and Koepf, W.: Maple package on formal power series. *Maple Technical Newsletter* **2** (2), 1995, 22–28. Appeared also as `SC-93-31.pdf`, Konrad-Zuse-Zentrum Berlin, see <http://www.mathematik.uni-kassel.de/~koepf/Publikationen/>.
- [8] van Hoeij, M.: Finite singularities and hypergeometric solutions of linear recurrence equations. *Journal of Pure and Applied Algebra* **139**, 1998, 109–131.
- [9] Koekoek, R., Lesky, P. A. and Swarttouw, R.: *Hypergeometric Orthogonal Polynomials and Their q -Analogues*. Springer Monographs in Mathematics, 2010.
- [10] Koepf, W.: On two conjectures of M. S. Robertson. *Complex Variables* **16**, 1991, 127–130.
- [11] Koepf, W.: Power series in computer algebra. *J. Symbolic Computation* **13**, 1992, 581–603.
- [12] Koepf, W.: *Computeralgebra. Eine algorithmisch orientierte Einführung*. Springer, Berlin-Heidelberg, 2006.
- [13] Koepf, W.: *Hypergeometric Summation. An Algorithmic Approach to Summation and Special Function Identities*. First Edition, Vieweg, Braunschweig–Wiesbaden, 1998; Second Edition, Springer Universitext. Springer, London, 2014.
- [14] Perlstadt, M. A.: Some recurrences for sums of powers of binomial coefficients. *J. Number Theory* **27**, 1987, 304–309.

- [15] Petkovšek, M.: Hypergeometric solutions of linear recurrences with polynomial coefficients. *J. Symbolic Computation* **14**, 1992, 243–264.
- [16] Robertson, M. S.: Complex powers of p -valent functions and subordination. *Complex Anal., Proc. S.U.N.Y. Conf., Brockport 1976*, *Lect. Notes Math.* **36**, 1978, 1–33.
- [17] Robertson, M. S.: A subordination problem and a related class of polynomials. In: Srivastava, H. M., Owa, Sh. (Eds.): *Univalent Functions, Fractional Calculus, and Their Applications*. Ellis Horwood Limited, Chichester, 1989, 245–266.
- [18] Salvy, B. and Zimmermann, P.: GFUN: A Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software* **20**, 1994, 163–177.
- [19] Stanley, R. P.: Differentiably finite power series. *Europ. J. Combinatorics* **1**, 1980, 175–188.
- [20] Zeilberger, D.: A fast algorithm for proving terminating hypergeometric identities. *Discrete Math.* **80**, 1990, 207–211.

Wolfram Koepf
Institute of Mathematics
University of Kassel
Heinrich-Plett-Str. 40
D-34132 Kassel, Germany
e-mail: koepf@mathematik.uni-kassel.de