Computer Algebra Algorithms for Orthogonal Polynomials and Special Functions

Wolfram Koepf

Department of Mathematics and Computer Science, University of Kassel,
Heinrich-Plett-Str. 40, D–34132 Kassel, Germany,
koepf@mathematik.uni-kassel.de

Summary. In this minicourse I would like to present computer algebra algorithms for the work with orthogonal polynomials and special functions. This includes

- the computation of power series representations of hypergeometric type functions, given by “expressions”, like $\arcsin(x)/x$,
- the computation of holonomic differential equations for functions, given by expressions,
- the computation of holonomic recurrence equations for sequences, given by expressions, like $\binom{n}{k}\frac{x^k}{k!}$,
- the identification of hypergeometric functions,
- the computation of antidifferences of hypergeometric terms (Gosper's algorithm),
- the computation of holonomic differential and recurrence equations for hypergeometric series, given the series summand, like
$$P_n(x) = \sum_{k=0}^{n} \binom{n}{k} \binom{-n-1}{k} \left(\frac{1-x}{2}\right)^k$$
(Zeilberger's algorithm),
- the computation of hypergeometric term representations of series (Zeilberger's and Petkovšek's algorithm),
- the verification of identities for (holonomic) special functions,
- the detection of identities for orthogonal polynomials and special functions,
- the computation with Rodrigues formulas,
- the computation with generating functions,
- corresponding algorithms for q-hypergeometric (basic hypergeometric) functions,
- the identification of classical orthogonal polynomials, given by recurrence equations.

All topics are properly introduced, the algorithms are discussed in some detail and many examples are demonstrated by Maple implementations. In the lecture, the participants are invited to submit and compute their own examples.

Let us remark that as a general reference we use the book [11], the computer algebra system Maple [16], [4] and the Maple packages **FPS** [9], [7], **gfun** [19], **hsum** [11], **infhsum** [22], **hsols** [21], **qsum** [2] and **retode** [13].

1 The computation of power series and hypergeometric functions

Given an expression $f(x)$ in the variable x, one would like to find the Taylor series

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

i.e., a formula for the coefficient a_k. For example, if $f(x) = \exp(x)$, then

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

hence $a_k = \frac{1}{k!}$. If the result is simple enough, the `FPS` (formal power series) procedure of the Maple package `FPS.mpl` ([9], [7]) computes this series, even if it is a Laurent series (including negative powers) or Puiseux series (including rational powers).

The main idea behind this procedure is

1. to compute a differential equation for $f(x)$,
2. to convert the differential equation to a recurrence equation for a_k,
3. and to solve the recurrence equation for a_k.

1.1 Hypergeometric series

The above procedure is successful at least if $f(x)$ is hypergeometric. A series

$$\sum_{k=0}^{\infty} a_k$$

is called hypergeometric, if the series coefficient a_k has rational term ratio

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

The function

$$ {}_pF_q\left(\left.\begin{matrix} a_1, a_2, \ldots, a_p \\ b_1, b_2, \ldots, b_q \end{matrix}\right| x\right) := \sum_{k=0}^{\infty} A_k\, x^k = \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k \cdots (a_p)_k}{(b_1)_k (b_2)_k \cdots (b_q)_k} \frac{x^k}{k!} \qquad (1.1)$$

is called the *generalized hypergeometric series*, since its term ratio

$$\frac{A_{k+1}\, x^{k+1}}{A_k\, x^k} = \frac{(k+a_1)\cdots(k+a_p)}{(k+b_1)\cdots(k+b_q)} \frac{x}{(k+1)} \qquad (1.2)$$

is a general rational function, in factorized form. Here $(a)_k = a(a+1)\cdots(a+k-1)$ denotes the *Pochhammer symbol* or shifted factorial. The summand a_k of the generalized hypergeometric series is called a *hypergeometric term.*

The Maple commands `factorial` (short form `!`), `pochhammer`, `binomial`, and `GAMMA` can be used to enter the corresponding functions, `hypergeom` denotes the hypergeometric series, and the `hyperterm` command of the **sumtools** and **hsum** packages denotes a hypergeometric term.[1]

[1] The package `sumtools` is part of Maple [4]. Note that Maple 8 contains a second package `SumTools` ([15], [4]) which also contains summation algorithms.

1.2 Holonomic differential equations

A homogeneous linear differential equation with polynomial coefficients is called *holonomic*. If $f(x)$ satisfies a holonomic differential equation, then its Taylor series coefficients a_k satisfy a holonomic recurrence equation, and vice versa.

To find a holonomic differential equation for an expression $f(x)$, one differentiates $f(x)$, and writes the sum

$$\sum_{j=0}^{J} c_j f^{(j)}(x)$$

as a sum of (over $\mathbb{Q}(x)$) linearly independent summands, whose coefficients should be zero. This gives a system of linear equations for $c_j \in \mathbb{Q}(x)$ $(j = 0, \ldots, J)$. If it has a solution, we have found a differential equation with rational function coefficients, and multiplying by their common denominator yields the equation sought for.

Iterating this procedure for $J = 1, 2, \ldots$ yields the holonomic differential equation of lowest order valid for $f(x)$.

The command `HolonomicDE`[2] of the **FPS** package is an implementation of this algorithm.

Exercise 1. Find a holonomic differential equation for $f(x) = \sin(x)\exp(x)$. Use the algorithm described. Don't use the **FPS** package.
Using the **FPS** package `FPS.mpl`, find a holonomic differential equation for $f(x)$ and for $g(x) = \arcsin(x)^3$.

1.3 Algebra of holonomic functions

A function that satisfies a holonomic differential equation is called a holonomic function. Sum and product of holonomic functions turn out to be holonomic, and their representing differential equations can be computed from the differential equations of the summands and factors, respectively, by linear algebra.

We call a sequence that satisfies a holonomic recurrence equation a holonomic sequence. Sum and product of holonomic sequences are holonomic, and similar algorithms exist. As already mentioned, a function is holonomic if and only if it is the generating function of a holonomic sequence.

The **gfun** package by Salvy and Zimmermann [19] contains—besides others—implementations of the above algorithms.

[2] In earlier versions of the **FPS** package the command name was `SimpleDE`.

Exercise 2. Use the **gfun** package to generate differential equations for $f(x) = \sin(x)\exp(x)$, and $g(x) = \sin(x) + \exp(x)$ by utilizing the (known) ODEs for the summands and factors, respectively.
Use the **gfun** package to generate recurrence equations for

$$a_k = k\binom{n}{k}^2 \quad \text{and} \quad b_k = k + \binom{n}{k}^2 .$$

1.4 Hypergeometric power series

Having found a holonomic differential equation for $f(x)$, by substituting

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

and equating coefficients, it is easy to deduce a holonomic recurrence equation for a_k.

If we are lucky, the recurrence is of first order, hence the function is a hypergeometric series, and the coefficients can be computed by (1.1)–(1.2).

The command `SimpleRE` of the **FPS** package combines the above steps and computes a recurrence equation for the series coefficients of an expression.

1.5 Identification of hypergeometric functions

Assume, we have

$$F = \sum_{k=0}^{\infty} a_k .$$

How do we find out which ${}_pF_q(x)$ this is?

The simple idea is to write the ratio $\frac{a_{k+1}}{a_k}$ as factorized rational function, and to read off the upper and lower parameters according to (1.2).

The command `Sumtohyper` of the **sumtools** and **hsum** packages are implementations of this algorithm.

Exercise 3. Write $\cos(x)$ in hypergeometric notation by hand computation. Use the **sumtools** package to do the same. Restart your session and use the **hsum** package `hsum6.mpl` instead.
Get the hypergeometric representations for $\sin(x)$, $\sin(x)^2$, $\arcsin(x)$, $\arcsin(x)^2$, and $\arctan(x)$, combining **FPS** and **hsum**.

Exercise 4. Write the following representations of the *Legendre polynomials* in hypergeometric notation:

$$P_n(x) = \sum_{k=0}^{n} \binom{n}{k} \binom{-n-1}{k} \left(\frac{1-x}{2}\right)^k \tag{1.3}$$

$$= \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k}^2 (x-1)^{n-k}(x+1)^k \tag{1.4}$$

$$= \frac{1}{2^n} \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \binom{n}{k} \binom{2n-2k}{n} x^{n-2k} \,. \tag{1.5}$$

In the hypergeometric representations, where from can you read off the upper bound of the sum?

2 Summation of hypergeometric series

In this section, we try to simplify both definite and indefinite hypergeometric series.

2.1 Fasenmyer's method

Given a sequence s_n, as hypergeometric sum

$$s_n = \sum_{k=-\infty}^{\infty} F(n,k) \;,$$

how do we find a recurrence equation for s_n? Celine Fasenmyer proposed the following algorithm (see e.g., [11], Chapter 4):

1. Compute ansatz $:= \sum\limits_{\substack{i=0,\ldots,I \\ j=0,\ldots,J}} \dfrac{F(n+j,k+i)}{F(n,k)} \in \mathbb{Q}(n,k)$.
2. Bring this into rational form and set the numerator coefficient list w.r.t. k zero. If the corresponding linear system has a solution, this leads to a k-free recurrence equation for the summand $F(n,k)$.
3. Summing this recurrence equation for $k = -\infty, \ldots, \infty$ gives the desired recurrence for s_n.

If successful, this results in a holonomic recurrence equation for s_n. If we are lucky, and the recurrence is of first order, then the sum can be written as a hypergeometric term by formula (1.1)–(1.2). This algorithm can be accessed by the commands `kfreerec` and `fasenmyer` of the **hsum** package.

As an example, to compute

$$s_n = \sum_{k=0}^{n} F(n,k) = \sum_{k=0}^{n} \binom{n}{k} ,$$

in the first step one gets the well-known binomial coefficient recurrence

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

or in the usual notation

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} ,$$

from which it follows by summation for $k = -\infty, \ldots, \infty$

$$s_{n+1} = s_n + s_n = 2 s_n ,$$

since

$$\sum_{k=-\infty}^{\infty} F(n,k) = \sum_{k=-\infty}^{\infty} F(n,k-1) .$$

With $s_0 = 1$ one finally gets $s_n = 2^n$.

In practice, however, Fasenmyer's algorithm is rather slow and inefficient.

Exercise 5. Using Fasenmyer's method, compute a three-term recurrence equation for the *Laguerre polynomials*

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

and for the *generalized Laguerre polynomials*

$$L_n^{(\alpha)}(x) = \sum_{k=0}^{n} \frac{(-1)^k}{k!} \binom{n+\alpha}{n-k} x^k. \tag{2.1}$$

2.2 Indefinite summation and Gosper's algorithm

Given a sequence a_k, one would like to find a sequence s_k which satisfies

$$a_k = s_{k+1} - s_k = \Delta s_k . \tag{2.2}$$

Having found s_k makes definite summation easy since by telescoping it follows from (2.2) for arbitrary $M, N \in \mathbb{Z}$

$$\sum_{k=M}^{N} a_k = s_{N+1} - s_M .$$

We call $s_k = \sum a_k$ an indefinite sum (or an antidifference) of a_k. Hence indefinite summation is the inverse of the forward difference operator Δ.

Gosper's algorithm ([6], see e.g., [11], Chapter 5) takes a hypergeometric term a_k and decides whether or not a_k has a hypergeometric term antidifference, and computes it in the affirmative case. In the latter case s_k is a rational multiple of a_k, $s_k = R_k\, a_k$ with $R_k \in \mathbb{Q}(k)$.

Note that whenever Gosper's algorithm does not find a hypergeometric term antidifference, it has therefore *proved* that no such antidifference exists. In particular, using this approach, it is easily shown that the *harmonic numbers* $H_n = \sum\limits_{k=1}^{n} \frac{1}{k}$ cannot be written as a hypergeometric term. On the other hand, one gets (checking the result applying Δ is easy!)

$$\sum a_k = \sum (-1)^k \binom{n}{k} = -\frac{k}{n}\, a_k \;.$$

Both Maple's **sumtools** package and `hsum6.mpl` contain an implementation of Gosper's algorithm by the author. The `gosper` command of the **hsum** package will give error messages that let the user know whether the input is not a hypergeometric term (and hence the algorithm is not applicable) or whether the algorithm has deduced that no hypergeometric term antidifference exists. Since this is (unfortunately) against Maple's general policy, `sumtools[gosper]` does not do so, and gives `FAIL` in these cases.

Exercise 6. Use Gosper's algorithm to compute

$$s(m,n) = \sum_{k=0}^{m} (-1)^k \binom{n}{k} \;,$$

$$t_n = \sum_{k=1}^{n} k^3 \;,$$

and

$$u_n = \sum_{k=1}^{n} \frac{1}{k(k+5)} \;.$$

2.3 Zeilberger's algorithm

Zeilberger [24] had the brilliant idea to use a modified version of Gosper's algorithm to compute definite hypergeometric sums

$$s_n = \sum_{k=-\infty}^{\infty} F(n,k) \;,$$

like Fasenmyer's algorithm does (see e.g., [11], Chapter 7). However, Zeilberger's algorithm is much more efficient than Fasenmyer's. Note that, whenever s_n is itself a hypergeometric term, then Gosper's algorithm, applied to $F(n,k)$, fails! Thus a direct application of Gosper's algorithm to the summand is not possible.

Zeilberger's algorithm works as follows:

1. For suitable $J \in \mathbb{N}$ set
$$a_k = F(n,k) + \sigma_1 F(n+1,k) + \cdots + \sigma_J F(n+J,k) \ .$$
2. Apply Gosper's algorithm to determine a hypergeometric term s_k, and at the same time rational functions $\sigma_j \in \mathbb{Q}(n)$ such that $a_k = s_{k+1} - s_k$.
3. Summing for $k = -\infty, \ldots, \infty$ yields the desired holonomic recurrence equation for s_n by telescoping.

One can prove that Zeilberger's algorithm terminates for suitable input.

The command `sumtools[sumrecursion]` as well as `sumrecursion` and `closedform` of the **hsum** package are implementations of Zeilberger's algorithm.

Exercise 7. Compute recurrence equations for the binomial power sums
$$\sum_{k=0}^{n} \binom{n}{k}^m$$
for $m = 2, \ldots, 7$.
In the 1980s these results were worth a paper in a mathematical journal!

If the resulting recurrence equation is of first order, then in combination with formula (1.1)–(1.2) and the value s_0 one gets a hypergeometric term representation of the sum s_n. This is the strategy of the `closedform` command.

As an example, using this approach, it is easy to deduce *Dougall's identity*
$$ {}_7F_6\left(\left.\begin{matrix} a, 1+\frac{a}{2}, b, c, d, 1+2a-b-c-d+n, -n \\ \frac{a}{2}, 1+a-b, 1+a-c, 1+a-d, b+c+d-a-n, 1+a+n \end{matrix}\right| 1\right)$$
$$= \frac{(1+a)_n\,(a+1-b-c)_n\,(a+1-b-d)_n\,(a+1-c-d)_n}{(1+a-b)_n\,(1+a-c)_n\,(1+a-d)_n\,(1+a-b-c-d)_n}$$
from its left hand side.

Exercise 8. Find hypergeometric term representations for the sums

$$s_n = \sum_{k=0}^{n} k \binom{n}{k} \,,$$

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

$$u_n = \sum_{k=0}^{n} (-1)^k \binom{n}{k}^2 \,,$$

and

$$v_n = \sum_{k=0}^{n} \binom{a}{k} \binom{b}{n-k} \,.$$

Assume you have two hypergeometric sums, how can you check whether they are different disguised versions of the same special function? Zeilberger's paradigm is to compute their recurrence equations, and, if these agree, then it remains to check enough initial values. Using this approach, it is easy to check that the three representations of the Legendre polynomials, given in (1.3)–(1.5), all agree with the fourth representation

$$P_n(x) = x^n \, {}_2F_1\left(\left. \begin{matrix} \frac{n}{2}, \frac{1-n}{2} \\ 1 \end{matrix} \right| 1 - \frac{1}{x^2} \right) \tag{2.3}$$

Exercise 9. Prove that the representations (1.3)–(1.5) and (2.3) all constitute the same functions, the Legendre polynomials.

With this method, one can prove many of the hypergeometric identities that appear in Joris van der Jeugt's contribution in these lecture notes, for example Whipple's transformation (2.10) as well as the different representations of the Clebsch-Gordan coefficients `Clebsch-Gordan coefficients` and Racah polynomials, and many of the corresponding exercises can be solved automatically.

Even more advanced questions that involve double sums can be solved: to prove *Clausen's formula*

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

which gives the cases when a ${}_3F_2$ is the square of a ${}_2F_1$, one can write the left hand side as a Cauchy product. This gives a double sum. It turns out that the inner sum can be simplified by Zeilberger's algorithm, and the remaining sum is exactly the right hand side.

2.4 A generating function problem

Recently, Folkmar Bornemann showed me a generating function of the Legendre polynomials and asked me to generate it automatically [3]. Here is the problem: write

$$G(x,z,\alpha) := \sum_{n=0}^{\infty} \binom{\alpha+n-1}{n} P_n(x)\, z^n$$

as a hypergeometric function. We can take any of the four given hypergeometric representations of the Legendre polynomials to write $G(x,z,\alpha)$ as double sum. Then the trick is to change the order of summation. If we are lucky, then the inner sum is Zeilberger-summable, hence a single hypergeometric sum remains which gives the desired result.

It turns out that (only) the fourth representation (2.3) leads to such a result, namely

$$\sum_{n=0}^{\infty} \binom{\alpha+n-1}{n} P_n(x)\, z^n = \frac{1}{(1-xz)^{\alpha}}\, {}_2F_1\left(\left.\begin{matrix} \frac{\alpha}{2}, \frac{\alpha+1}{2} \\ 1 \end{matrix}\right| \frac{(x^2-1)\,z^2}{(xz-1)^2}\right) .$$

Note that a variant of this identity can be found as number 05.03.23.0006.01 in the extensive online handbook [23]. It occurs there without citation, though, hence without proof.

Advanced Exercise 10. Derive the *Askey-Gasper identity* which was the essential ingredient in de Branges' proof of the Bieberbach conjecture (see, e.g., [11], Example 7.4). Hence write as a single hypergeometric series

$$\sum_{j=0}^{\lfloor \frac{n-k}{2} \rfloor} \frac{\left(\frac{1}{2}\right)_j \left(\frac{\alpha}{2}+1\right)_{n-j} \left(\frac{\alpha+3}{2}\right)_{n-2j} (\alpha+1)_{n-2j}}{j! \left(\frac{\alpha+3}{2}\right)_{n-j} \left(\frac{\alpha+1}{2}\right)_{n-2j} (n-2j)!} \\ {}_3F_2\left(\left.\begin{matrix} 2j-n, n-2j+\alpha+1, \frac{\alpha+1}{2} \\ \alpha+1, \frac{\alpha+2}{2} \end{matrix}\right| x\right).$$

2.5 Automatic computation of infinite sums

Whereas Zeilberger's algorithm finds *Chu-Vandermonde's formula* for $n \in \mathbb{N}_{\geq 0}$

$$ {}_2F_1\left(\left.\begin{matrix} -n, b \\ c \end{matrix}\right| 1\right) = \frac{(c-b)_n}{(c)_n} , \tag{2.4}$$

the question arises how to detect *Gauss' identity*

$$
{}_2F_1\left(\left.\begin{array}{c}a,b\\c\end{array}\right|1\right)=\frac{\Gamma(c)\Gamma(c-a-b)}{\Gamma(c-a)\Gamma(c-b)}
$$

for $a,b,c\in\mathbb{C}$ in case of convergence, i.e., if $\mathrm{Re}\,(c-a-b)>0$, extending (2.4).

The idea is to detect by Zeilberger's algorithm

$$
{}_2F_1\left(\left.\begin{array}{c}a,b\\c+m\end{array}\right|1\right)=\frac{(c-a)_m\,(c-b)_m}{(c)_m\,(c-a-b)_m}\;{}_2F_1\left(\left.\begin{array}{c}a,b\\c\end{array}\right|1\right)
$$

and then consider the limit as $m\to\infty$.

Using appropriate limits for the Gamma function, this and similar questions can be handled automatically by the `infclosedform` prodecure of Vidunas' and Koornwinder's Maple package **infhsum** [22].

Exercise 11. Derive *Kummer's theorem*, i.e. find a closed form for

$$
{}_2F_1\left(\left.\begin{array}{c}a,b\\1+a-b\end{array}\right|-1\right).
$$

Find an extension of the *Pfaff-Saalschütz formula*

$$
{}_3F_2\left(\left.\begin{array}{c}a,b,-n\\c,1+a+b-c-n\end{array}\right|1\right)=\frac{(c-a)_n\,(c-b)_n}{(c)_n\,(c-a-b)_n}.
$$

2.6 The WZ method

Assume we want to prove an identity

$$
\sum_{k=-\infty}^{\infty}f(n,k)=\tilde{s}_n
$$

with hypergeometric terms $f(n,k)$ and $\tilde{s}_n$, see e.g. [11], Chapter 6. Then Wilf's idea is to divide by $\tilde{s}_n$, and therefore to put the identity into the form

$$
s_n:=\sum_{k=-\infty}^{\infty}F(n,k)=1\,. \tag{2.5}
$$

Now we can apply Gosper's algorithm to $F(n+1,k)-F(n,k)$ as a function of k. If this is successful, then it generates a rational multiple $G(n,k)$ of $F(n,k)$, i.e., $G(n,k)=R(n,k)\,F(n,k)$ with $R(n,k)\in\mathbb{Q}(n,k)$, such that

$$
F(n+1,k)-F(n,k)=G(n,k+1)-G(n,k)\,, \tag{2.6}
$$

and telescoping yields $s_{n+1} - s_n = 0$, and therefore (2.5). Hence the sheer success of Gosper's algorithm gives a proof of (2.5). This is called the WZ method.

Moreover, if the WZ method was successful, it has computed the *rational certificate* $R(n,k) \in \mathbb{Q}(n,k)$ which enables a completely independent proof of (2.5) that can be carried out by hand computations: Dividing (2.6) by $F(n,k)$, we have only to prove

$$\frac{F(n+1,k)}{F(n,k)} - 1 = R(n,k+1)\frac{F(n,k+1)}{F(n,k)} - R(n,k) ,$$

a purely rational identity.

The function `WZcertificate` of the **hsum** package computes the rational certificate if applicable.

Exercise 12. Prove by the WZ method:

$$\sum_{k=1}^{n} k\binom{n}{k} = n2^{n-1}$$

and (this is again the disguised Chu-Vandermonde formula)

$$\sum_{k=0}^{n} \binom{a}{k}\binom{b}{n-k} = \binom{a+b}{n} .$$

2.7 Differential equations

Zeilberger's algorithm can easily be adapted to generate holonomic differential equations for *hyperexponential sums* (see, e.g., [11], Chapter 10)

$$s(x) = \sum_{k=-\infty}^{\infty} F(x,k) .$$

For this purpose, the summand $F(x,k)$ must be a hyperexponential term w.r.t. x:

$$\frac{\frac{\partial}{\partial x}F(x,k)}{F(x,k)} \in \mathbb{Q}(x,k) .$$

With this algorithm which is implemented as `sumdiffeq` in the **hsum** package it is easy to check that all representations (1.3)–(1.5) and (2.3) of the Legendre polynomials satisfy the same differential equation

$$(1-x^2)\,P_n''(x) - 2x\,P_n'(x) + n(n+1)\,P_n(x) = 0 .$$

In CAOP [20], an online version of the *Askey-Wilson scheme* of orthogonal polynomials [8] developed by René Swarttouw, the **hsum** package is used to interactively compute recurrence and differential equations for personally standardized orthogonal polynomial families of the Askey-Wilson scheme.

Exercise 13. Find holonomic differential equations for the *Jacobi polynomials*

$$P_n^{(\alpha,\beta)}(x) = \binom{n+\alpha}{n} {}_2F_1\left(\left.\begin{matrix}-n, n+\alpha \\ \alpha+\beta+1\end{matrix}\right| \frac{1+x}{2}\right), \tag{2.7}$$

and the generalized Laguerre polynomials (2.1).

3 Hypergeometric term solutions of recurrence equations

3.1 Petkovšek's algorithm

Petkovšek's algorithm is an adaption of Gosper's (see, e.g., [11], Chapter 9). Given a holonomic recurrence equation, it determines all hypergeometric term solutions. The command `rechyper` of the **hsum** package is an implementation of Petkovšek's algorithm.

Petkovšek's algorithm is slow, especially if the leading and trailing coefficients of the recurrence equation have many factors. Maple 9 will contain a much more efficient algorithm **hsols** due to Mark van Hoeij [21] for the same purpose.

As an example, the recurrence equation

$$3(3n+4)(3n+7)(3n+8)s_{n+3} + 4(3n+4)(37n^2+180n+218)s_{n+2}$$

$$+16(n+2)(33n^2+125n+107)s_{n+1} + 64(n+1)(n+2)(3n+7)s_n = 0$$

which is the output of Zeilberger's algorithm applied to the sum

$$s_n = \sum_{k=0}^{n} (-1)^k \binom{n}{k}\binom{4k}{n},$$

has the hypergeometric term solution $(-4)^n$ which finally yields $s_n = (-4)^n$.

3.2 Combining Zeilberger's and Petkovšek's algorithms

As seen, Zeilberger's algorithm may not give a recurrence equation of first order, even if the sum is a hypergeometric term. This rarely happens, though. In such a case, the combination of Zeilberger's with Petkovšek's algorithm guarantees to find out whether a given sum can be written as a hypergeometric term.

Exercise 9.3 of my book [11] gives 9 examples for this situation, all from p. 556 of [18].

Advanced Exercise 14. Use a combination of Zeilberger's algorithm and Petkovšek's algorithm to find a simple representation (as linear combination of two hypergeometric terms) of the sum

$$s_n = \sum_{k=0}^{\lfloor n/3 \rfloor} \binom{n-2k}{k} \left(-\frac{4}{27}\right)^k .$$

4 Integration

4.1 Indefinite integration

To find holonomic recurrence and differential equations for hypergeometric and hyperexponential integrals, one needs a continuous version of Gosper's algorithm. Almkvist and Zeilberger gave such an algorithm ([1], see, e.g., [11], Chapter 11). It finds hyperexponential antiderivatives if those exist. This algorithm is accessible as procedure `contgosper` of the **hsum** package.

For example, this algorithm proves that the function e^{x^2} does not have a hyperexponential antiderivative. In fact, this function does not even have an elementary antiderivative; but this cannot be detected by the given algorithm. On the other hand, the algorithm computes, e.g., the integral

$$\int \left(\frac{2x}{1-x^{10}} + \frac{10(1+x^2)x^9}{(1-x^{10})^2} \right) dx = \frac{1+x^2}{1-x^{10}} .$$

4.2 Definite integration

Applying the continuous Gosper algorithm, one can easily adapt the discrete versions of Zeilberger's algorithm to the continuous case. The resulting algorithms find holonomic recurrence and differential equations for hypergeometric and hyperexponential integrals.

The procedures `intrecursion` and `intdiffeq` of the **hsum** package are implementations of these algorithms, see [11], Chapter 12.

As an example, we would like to find

$$S(x,y) = \int_0^1 t^x \, (1-t)^y \, dt .$$

Applying the continuous Zeilberger algorithm w.r.t. x and y, respectively, results in the two recurrence equations

$$-(x+y+2)\, S(x+1,y) + (x+1)\, S(x) = 0$$

and

$$-(x+y+2)\,S(x,y+1)+(x+1)\,S(x)=0\;.$$

Solving both recurrence equations (e.g., with Maple's `rsolve` command) shows that $S(x,y)$ must be a multiple of

$$\frac{\Gamma(x+1)\,\Gamma(y+1)}{\Gamma(x+y+2)}\;.$$

Computing the initial value

$$S(0,0)=\int\limits_0^1 dt=1$$

proves the identity

$$S(x,y)=\frac{\Gamma(x+1)\,\Gamma(y+1)}{\Gamma(x+y+2)}$$

for $x,y\in\mathbb{Z}$. Since we work with recurrence equations, this method cannot find the result for other complex values x,y.

Another example is given by the integral

$$I(x)=\int\limits_0^\infty \frac{x^2}{(x^4+t^2)(1+t^2)}dt$$

for which the algorithm detects the holonomic differential equation

$$x(x^4-1)\,I''(x)+(1+7x^4)\,I'(x)+8x^3\,I(x)=0\;.$$

Maple's `dsolve` command finds the solution

$$I(x)=\frac{\pi}{2(x^2+1)}\;.$$

Advanced Exercise 15. Write the integral

$$\int\limits_0^1 t^{c-1}\,(1-t)^{d-1}\;{}_2F_1\left(\left.\begin{matrix}a,b\\c\end{matrix}\right|tx\right)$$

as a hypergeometric series. This generates the so-called *Bateman integral representation*. For which $c,d\in\mathbb{C}$ is the result valid?

Advanced Exercise 16. Find a similar representation for the integral

$$\int\limits_0^1 t^{c-1}\,(1-t)^{d-1}\;{}_2F_1\left(\left.\begin{matrix}a,b\\d\end{matrix}\right|tx\right)\;.$$

4.3 Rodrigues formulas

Using Cauchy's integral formula

$$h^{(n)}(z) = \frac{n!}{2\pi i} \int\limits_{\gamma} \frac{h(t)}{(t-x)^{n+1}} dt$$

for the nth derivative makes the integration algorithms accessible for Rodrigues type expressions

$$f_n(x) = g_n(x) \, \frac{d^n}{dx^n} h_n(x) \; .$$

This is implemented in`rodriguesrecursion` and `rodriguesdiffeq` of the **hsum** package, see [11], Chapter 13.

Using these algorithms, one can easily show that the functions

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

are the Legendre polynomials, and that

$$L_n^{(\alpha)}(x) = \frac{e^x}{n! \, x^\alpha} \frac{d^n}{dx^n} \left(e^{-x} \, x^{\alpha+n} \right)$$

are the generalized Laguerre polynomials.

Exercise 17. Prove the Rodrigues formula

$$P_n^{(\alpha,\beta)}(x) = \frac{(-1)^n}{2^n \, n!} (1-x)^{-\alpha} \, (1+x)^{-\beta} \frac{d^n}{dx^n} \left((1-x)^\alpha (1+x)^\beta (1-x^2)^n \right)$$

for the Jacobi polynomials (2.7).

4.4 Generating functions

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

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

then by Cauchy's formula and Taylor's theorem

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

Hence, again, we can apply the integration algorithms. This is implemented in the functions `GFrecursion` and `GFdiffeq` of the **hsum** package, see [11], Chapter 13.

Using these algorithms, we can easily prove the generating function identity

$$(1-z)^{-\alpha-1}\, e^{\frac{xz}{z-1}} = \sum_{n=0}^{\infty} L_n^{(\alpha)}(x)\, z^n$$

for the generalized Laguerre polynomials.

Exercise 18. Prove that

$$\frac{1}{\sqrt{1-2xz+z^2}} = \sum_{n=0}^{\infty} P_n(x)\, z^n$$

is the generating function of the Legendre polynomials.

Advanced Exercise 19. Write the exponential generating function

$$F(x) = \sum_{n=0}^{\infty} \frac{1}{n!}\, P_n(x)\, z^n$$

of the Legendre polynomials in terms of Bessel functions.
Hint: Use one of the hypergeometric representations of the Legendre polynomials and change the order of summation.

5 Applications and further algorithms

5.1 Parameter derivatives

For some applications, one uses parametrized families of orthogonal polynomials like the generalized Laguerre polynomials that are parametrized by the parameter α. It might be necessary to know the rate of change of the family in the direction of the parameter α (see [10]).

Using Zeilberger's algorithm and limit computations (with Maple's `limit`) one can compute such parameter derivatives in this and in similiar occasions.

Advanced Exercise 20. Prove the following representation for the parameter derivative of the generalized Laguerre polynomials

$$\frac{\partial}{\partial\alpha} L_n^{(\alpha)}(x) = \sum_{k=0}^{n-1} \frac{1}{n-k}\, L_k^{(\alpha)}(x)$$

by proving first that

$$L_n^{(\alpha+\mu)}(x) = \sum_{k=0}^{n} \frac{(\mu)_{n-k}}{(n-k)!}\, L_k^{(\alpha)}(x)$$

and taking limit $\mu \to 0$.

5.2 Basic hypergeometric summation

Instead of considering series whose coefficients A_k have rational term ratio $A_{k+1}/A_k \in \mathbb{Q}(k)$, we can also consider such series whose coefficients A_k have term ratio $A_{k+1}/A_k \in \mathbb{Q}(q^k)$ for some $q \in \mathbb{C}$. This leads to the *q-hypergeometric series*—also called basic hypergeometric series—(see, e.g., [5])

$$ {}_r\varphi_s\left(\left.\begin{matrix} a_1, \ldots, a_r \\ b_1, \ldots, b_s \end{matrix}\right| q; x\right) = \sum_{k=0}^{\infty} A_k\, x^k \,. $$

Here the coefficients are given by

$$ A_k = \frac{(a_1;q)_k \cdots (a_r;q)_k}{(b_1;q)_k \cdots (b_s;q)_k} \frac{x^k}{(q;q)_k} \left((-1)^k\, q^{\binom{k}{2}}\right)^{1+s-r} \,, $$

where

$$ (a;q)_k = \prod_{j=0}^{k-1} (1 - aq^j) $$

denotes the *q-Pochhammer symbol.*

Further q-expressions are given by

1. the infinite q-Pochhammer symbol: $(a;q)_\infty = \lim\limits_{n\to\infty} (a;q)_n$;
2. the q-factorial: $[k]_q! = \dfrac{(q;q)_k}{(1-q)^k}$;
3. the q-Gamma function: $\Gamma_q(z) = \dfrac{(q;q)_\infty}{(q^z;q)_\infty}(1-q)^{1-z}$;
4. the q-binomial coefficient: $\begin{bmatrix} n \\ k \end{bmatrix}_q = \dfrac{(q;q)_n}{(q;q)_k\,(q;q)_{n-k}}$;
5. the q-brackets: $[k]_q = \dfrac{1-q^k}{1-q} = 1 + q + \cdots + q^{k-1}$.

For many of the algorithms mentioned in this minicourse corresponding q-versions exist, see [11]. These are implemented in the **qsum** package, see [2], and the above q-expressions are accessible, see [11], Chapter 3.

For all classical hypergeometric theorems corresponding q-versions exist. These can be proved by a q-version of Zeilberger's algorithm (`qsumrecursion`) via the **qsum** package. For example, the *q-Chu-Vandermonde theorem* states that

$$ {}_2\varphi_1\left(\left.\begin{matrix} q^{-n}, b \\ c \end{matrix}\right| q; \frac{c\,q^n}{b}\right) = \frac{(c/b;q)_n}{(c;q)_n} \,. $$

As usual, the right hand side can be computed from the left hand side.

All classical orthogonal families have q-hypergeometric equivalents. For example, the *little* and the *big q-Legendre polynomials*, respectively, are given by

$$p_n(x|q) = {}_2\varphi_1\left(\begin{matrix} q^{-n}, q^{n+1} \\ q \end{matrix} \middle| q; qx\right)$$

and

$$P_n(x;c;q) = {}_3\varphi_2\left(\begin{matrix} q^{-n}, q^{n+1}, x \\ q, cq \end{matrix} \middle| q; q\right) .$$

For these, by the procedure `qsumrecursion`, we get the recurrence equations

$$\begin{aligned} q^n(q^n-1)(q^n+q)p_n(x|q)+(q^{2n}-q)(q^{2n}x+xq^n+q^{n+1}x-2q^n+qx)p_{n-1}(x|q) \\ + \ q^n(q^n+1)(q^n-q)p_{n-2}(x|q)=0 \ , \end{aligned}$$

and

$$\begin{aligned} & q(q^n-1)(cq^n-1)(q^n+q)P_n(x;c;q) \\ & \quad + \ (q^{2n}-q)(q^{2n}x-2q^{n+1}+q^{n+1}x-2q^{m+1}c+xq^n+qx)P_{n-1}(x;c;q) \\ & \qquad - q^n(q^n+1)(q^b-q)(q^n-cq)P_{n-2}(x;c;q)=0 \ . \end{aligned}$$

Exercise 21. Prove the identity

$${}_1\varphi_0\left(\begin{matrix} a \\ - \end{matrix} \middle| q; x\right) \cdot {}_1\varphi_0\left(\begin{matrix} b \\ - \end{matrix} \middle| q; ax\right) = {}_1\varphi_0\left(\begin{matrix} ab \\ - \end{matrix} \middle| q; x\right) .$$

Compute $p_n(1|q)$ in closed form.

With the algorithms of the **qsum** package some of the exercises of Dennis Stanton's contribution in these lecture notes can be solved.

Using Hahn's q-difference operator

$$D_q f(x) := \frac{f(x)-f(qx)}{(1-q)x} \ ,$$

one can also compute *q-difference equations* w.r.t. the variable x by the `qsumdiffeq` procedure.

5.3 Orthogonal polynomial solutions of recurrence equations

The classical orthogonal polynomials

$$P_n(x) = k_n\, x^n + \cdots$$

(Jacobi, Laguerre, Hermite and Bessel) satisfy a second order differential equation

$$\sigma(x)\, P_n''(x) + \tau(x)\, P_n'(x) + \lambda_n\, P_n(x) = 0 \ ,$$

where $\sigma(x)$ is a polynomial of degree at most 2 and $\tau(x)$ is a polynomial of degree 1.[3] From this differential equation one can determine the three-term recurrence equation for $P_n(x)$ in terms of the coefficients of $\sigma(x)$ and $\tau(x)$, see [12].

Using this information in the opposite direction, one can find the corresponding differential equation of second order—if applicable—from a given holonomic three-term recurrence equation. This is implemented in the procudure `REtoDE` of the **retode** package `retode.mpl` [13]. Note that Koornwinder and Swarttouw have a similar package **rec2ortho** [14] but use a different approach.

As an example, we consider the recurrence equation

$$P_{n+2}(x) - (x-n-1)\,P_{n+1}(x) + \alpha\,(n+1)^2\,P_n(x) = 0\ . \tag{5.1}$$

For this recurrence, the program finds that only if $\alpha = \frac{1}{4}$ there is a classical orthogonal polynomial solution $P_n(x)$ with $k_n = 1$ and density $\rho(x) = 4e^{-2x}$ on the interval $[-\frac{1}{2}, \infty)$ satisfying the differential equation

$$\left(x + \frac{1}{2}\right) P_n''(x) - 2x\,P_n'(x) + 2n\,P_n(x) = 0\ ,$$

hence a translate of the Laguerre polynomials.

Similarly, the classical discrete orthogonal polynomials (Hahn, Meixner, Krawtchouk, Charlier) satisfy a second order difference equation

$$\sigma(x)\,\Delta\nabla P_n(x) + \tau(x)\Delta P_n(x) + \lambda_n\,P_n(x) = 0\ ,$$

where $\nabla f(x) = f(x) - f(x-1)$ is the backward difference operator, $\sigma(x)$ is a polynomial of degree at most 2 and $\tau(x)$ is a polynomial of degree 1. Again, from this equation one can determine the three-term recurrence equation for $P_n(x)$ in terms of the coefficients of $\sigma(x)$ and $\tau(x)$ and convert this step, see [13]. This algorithm is accessible via `REtodiscreteDE`.

Taking again example (5.1), now we find that only for $\alpha < \frac{1}{4}$ there are classical discrete orthogonal polynomial solutions of the Meixner/Krawtchouk family.

[3] If $\sigma(x)$ has two different real roots $a < b$, then $P_n(x)$ is of the Jacobi family in the interval $[a, b]$, if it has one double root, then $P_n(x)$ is of the Bessel family, if it has one single root, $P_n(x)$ is of the Laguerre family, and if it is constant, then $P_n(x)$ is of the Hermite family.

Exercise 22. Find the classical orthogonal polynomial solutions of the recurrence equation

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

Compute the recurrence equation for the functions

$$P_n(x) = {}_2F_0\left(\left.\begin{matrix}-n, -x \\ -\end{matrix}\,\right|\,\lambda\right)$$

and determine whether they are classical orthogonal polynomial systems.

Finally, the classical q-orthogonal polynomials (see [8]) of the Hahn class satisfy a second order q-difference equation

$$\sigma(x)\,D_q D_{1/q} P_n(x) + \tau(x) D_q P_n(x) + \lambda_n\,P_n(x) = 0\ ,$$

where $\sigma(x)$ is a polynomial of degree at most 2 and $\tau(x)$ is a polynomial of degree 1. Again, from this equation one can determine the three-term recurrence equation for $P_n(x)$ in terms of the coefficients of $\sigma(x)$ and $\tau(x)$ and convert this step, see [13]. This algorithm is accessible via `REtoqDE`.

As an example, for the recurrence equation

$$P_{n+2}(x) - x\,P_{n+1}(x) + \alpha\,q^n\,(q^{n+1}-1)\,P_n(x) = 0\ ,$$

we get the corresponding q-difference equation

$$(x^2+\alpha)\,D_q D_{1/q} P_n(x) - \frac{x}{q-1} D_q P_n(x) + \frac{q(q^n-1)}{(q-1)^2\,q^n}\,P_n(x) = 0\ .$$

Exercise 23. Check that the little and big q-Legendre polynomials are in the Hahn class of q-orthogonal polynomials.

6 Epilogue

The author's Maple packages and their help pages can be downloaded from the website `http://www.mathematik.uni-kassel.de/~koepf/Publikationen`. Installation guidelines can be obtained by e-mail request. Finally, the accompanying Maple worksheets for this minicourse can be found at `http://www.mathematik.uni-kassel.de/~koepf/iivortrag.html`.

Software development is a time consuming activity! Software developers love it when their software is used. But they need also your support. Hence my suggestion: if you use one of the packages mentioned for your research, please cite its use!

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. Bornemann, F.: private communication, 2002.
4. Char, B. W.: *Maple 8 Learning Guide*. Waterloo Maple, Waterloo, 2002.
5. Gasper, G. and Rahman, M.: *Basic Hypergeometric Series*. Encyclopedia of Mathematics and its Applications **35**, Cambridge University Press, London and New York, 1990.
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.
8. Koekoek, R. and Swarttouw, R. F.: The Askey-scheme of hypergeometric orthogonal polynomials and its q-analogue. Report 98-17, Delft University of Technology, Faculty of Information Technology and Systems, Department of Technical Mathematics and Informatics, Delft; electronic version available at `http://aw.twi.tudelft.nl/~koekoek/askey`, 1998.
9. Koepf, W.: Power series in computer algebra. J. Symbolic Computation **13** (1992), 581–603.
10. Koepf, W.: Identities for families of orthogonal polynomials and special functions. Integral Transforms and Special Functions **5** (1997), 69–102.
11. Koepf, W.: *Hypergeometric Summation*. Vieweg, Braunschweig/Wiesbaden, 1998.
12. Koepf, W. and Schmersau, D.: Representations of orthogonal polynomials. J. Comput. Appl. Math. **90** (1998), 57–94.
13. Koepf, W. and Schmersau, D.: Recurrence equations and their classical orthogonal polynomial solutions. Appl. Math. Comput. **128** (2-3), special issue on Orthogonal Systems and Applications (2002), 303–327.
14. Koornwinder, T. H. and Swarttouw, R.: rec2ortho: an algorithm for identifying orthogonal polynomials given by their three-term recurrence relation as special functions. `http://turing.wins.uva.nl/~thk/recentpapers/rec2ortho.html`, 1996–1998
15. Le, H. Q., Abramov, S. A. and Geddes, K. O.: HypergeometricSum: A Maple package for finding closed forms of indefinite and definite sums of hypergeometric type. `ftp://cs-archive.uwaterloo.ca/cs-archive/CS-2001-24`, 2002.
16. Monagan, M. B. et al.: *Maple 8 Introductory Programming Guide*, Waterloo Maple, Waterloo, 2002.
17. Petkovšek, M., Wilf, H. and Zeilberger, D.: $A = B$. A. K. Peters, Wellesley, 1996.
18. Prudnikov, A. P., Brychkov, Yu. A. and Marichev, O. I.: *Integrals and Series. Vol. 3: More Special Functions*. Gordon & Breach, New York, 1990.
19. 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.
20. Swarttouw, R.: CAOP: Computer algebra and orthogonal polynomials. `http://amstel.wins.uva.nl:7090/CAOP`, 1996–2002.

21. Van Hoeij, M.: Finite singularities and hypergeometric solutions of linear recurrence equations. J. Pure Appl. Algebra **139** (1999), 109–131.
22. Vidunas, R. and Koornwinder, T. H.: Zeilberger method for non-terminating hypergeometric series. 2002, to appear.
23. Wolfram's Research Mathematica Functions: `http://www.functions.wolfram.com`, 2002.
24. Zeilberger, D.: A fast algorithm for proving terminating hypergeometric identities. Discrete Math. **80** (1990), 207–211.