Computer Algebra Algorithms for Orthogonal Polynomials and Special Functions

Prof. Dr. Wolfram Koepf
Department of Mathematics
University of Kassel

koepf@mathematik.uni-kassel.de

http://www.mathematik.uni-kassel.de/~koepf

Online Demonstrations with Computer Algebra

- I will use the computer algebra system *Maple* to demonstrate and program the algorithms presented.
- Of course, we could also easily use any other system like *Mathematica* or MuPAD.
- We first give a short introduction about the capabilities of *Maple*.

Computation of Power Series

- Assume, given an expression f depending of the variable x, we would like to compute a formula for the coefficient a_k of the power series

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

representing $f(x)$.

Algorithm

- Input: expression $f(x)$
- Determine a holonomic differential equation DE (homogeneous and linear with polynomial coefficients) by computing the derivatives of $f(x)$ iteratively.
- Convert DE to a holonomic recurrence equation RE for a_k.
- Solve RE for a_k.
- Output: a_k resp. $\Sigma\, a_k\, x^k$

Computation of Holonomic Differential Equations

- Input: expression $f(x)$
- Compute $c_0 f(x) + c_1 f\,'(x) + \cdots + c_J f^{(J)}(x)$ with still undetermined coefficients c_j.
- Sort this w. r. t. linearly independent functions $\in \mathbb{Q}(x)$ and determine their coefficients.
- Set these zero, and solve the corresponding linear system for the unknowns $c_0, c_1, \ldots, c_J$.
- Output: DE:=$c_0 f(x) + c_1 f\,'(x) + \cdots + c_J f^{(J)}(x) = 0$.

Algebra of Holonomic Functions

- We call a function that satisfies a holonomic differential equation a holonomic function.
- Sum and product of holonomic functions turn out to be holonomic.
- We call a sequence that satisfies a holonomic recurrence equation a holonomic sequence.
- Sum and product of holonomic sequences are holonomic.
- A function is holonomic iff it is the generating function of a holonomic sequence.

Hypergeometric Functions

- The power series

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

whose coefficients A_k have rational term ratio

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

is called the generalized hypergeometric function.

Coefficients of Hypergeometric Functions

- For the coefficients of the hypergeometric function we get the formula

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

where $(a)_k = a(a+1)\cdots(a+k\text{-}1)$ is called the Pochhammer symbol (or shifted factorial).

Examples of Hypergeometric Functions

$$e^{x} = {}_0F_0(x)$$

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

Further examples: cos(x), arcsin(x), arctan(x), ln(1+x), erf(x), $L_n^{(\alpha)}(x)$, ..., but for example not tan(x), ...

Identification of Hypergeometric Functions

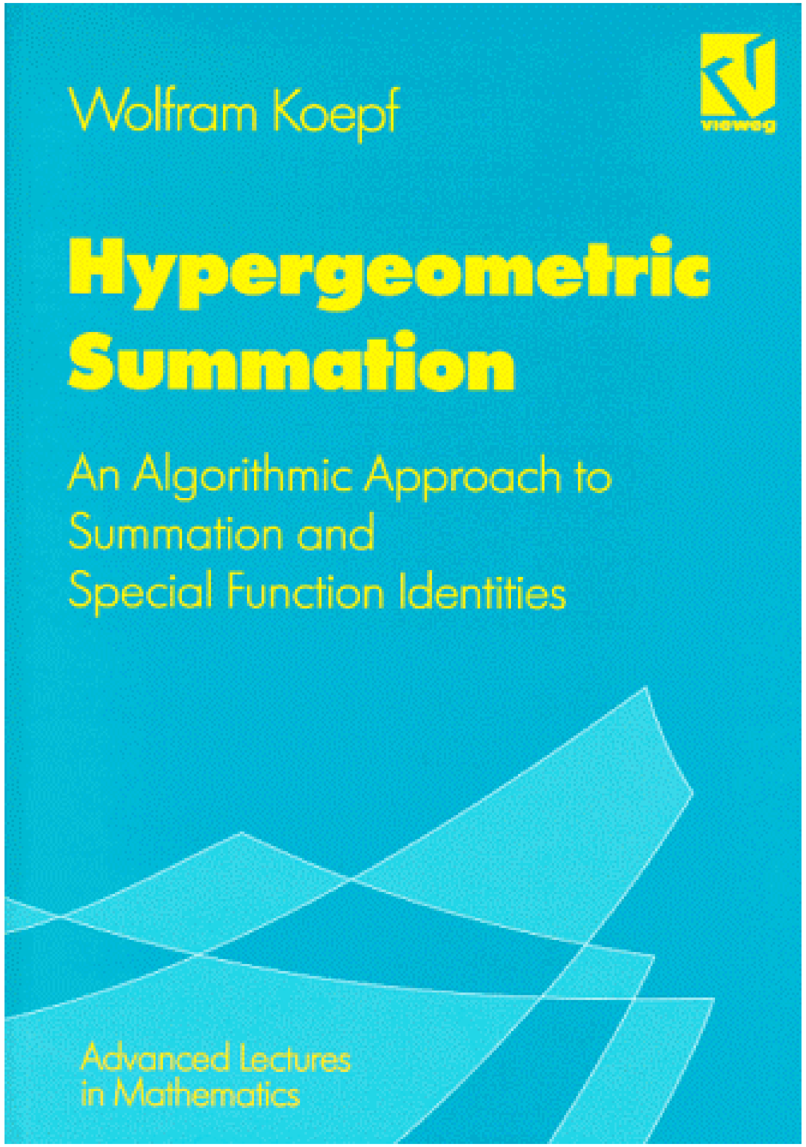
- Assume we have

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

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

Identification Algorithm

- Input: a_k
- Compute
$$r_k := \frac{a_{k+1}}{a_k}$$
and check whether the term ratio r_k is rational.
- Factorize r_k.
- Output: read off the upper and lower parameters and an initial value.


Wolfram Koepf
vieweg
Hypergeometric Summation
An Algorithmic Approach to Summation and Special Function Identities
Advanced Lectures in Mathematics

Recurrence Equations for Hypergeometric Functions

- 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's Algorithm

- Input: summand $F(n,k)$
- Compute

$$\text{ansatz} \coloneqq \sum_{\substack{i=0,\ldots,I \\ j=0,\ldots,J}} \frac{F(n+j,k+i)}{F(n,k)} \in \mathbb{Q}(n,k)$$

- Bring this into rational normal form, and set the numerator coefficient list w.r.t. k zero.
- Output: Sum the resulting recurrence equation for $F(n,k)$ w.r.t. k.

Drawbacks of Fasenmyer's Algorithm

- In easy cases this algorithm succeeds, but:
  - In many cases the algorithm generates a recurrence equation of too high order.
  - The algorithm is slow. If, e.g., $I = 2$ and $J = 2$, then already 9 linear equations have to be solved.
  - Therefore the algorithm might fail.

Indefinite Summation

- Given a sequence a_k, find a sequence s_k which satisfies

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

- Having found s_k makes definite summation easy since by telescoping for arbitrary *m*, *n*

$$\sum_{k=m}^{n} a_k = s_{n+1} - s_m \ .$$

- Indefinite summation is the inverse of Δ .

Gosper's Algorithm

- Input: a_k , a hypergeometric term.
- Compute p_k, q_k, $r_k \in \mathbb{Q}[k]$ with

$$\frac{a_{k+1}}{a_k} = \frac{p_{k+1}}{p_k} \frac{q_{k+1}}{r_{k+1}} \quad \text{and} \quad \gcd(q_k, r_{k+j}) = 1 \text{ for all } j \geq 0 .$$

- Find a polynomial solution f_k of the recurrence equation $q_{k+1} f_k - r_{k+1} f_{k-1} = p_k$.
- Output: the hyperg. term $s_k = \frac{r_k}{p_k} f_{k-1}\, a_k$.

Definite Summation: Zeilberger's Algorithm

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

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

- Note however that, whenever s_n is itself a hypergeometric term, then Gosper's algorithm, applied to $F(n,k)$, fails!

Zeilberger's Algorithm

- Input: summand $F(n,k)$
- For suitable $J \in \mathbb{N}$ set

$$a_k := F(n,k) + \sigma_1 F(n+1,k) + \cdots + \sigma_J F(n+J,k) .$$

- Apply the following modified version of Gosper's algorithm to a_k:
  - In the last step, solve at the same time for the coefficients of f_k and the unknowns $\sigma_j \in \mathbb{Q}(n)$.
- Output by summation: The recurrence equation

$$\mathrm{RE} := s_n + \sigma_1 s_{n+1} + \cdots + \sigma_J s_{n+J} = 0 .$$

The output of Zeilberger's Algorithm

- We apply Zeilberger's algorithm iteratively for $J = 1,2, \dots$ until it succeeds.
- If $J = 1$ is successful, then the resulting recurrence equation for s_n is of first order, hence s_n is a hypergeometric term.
- If $J > 1$, then the result is a holonomic recurrence equation for s_n.
- One can prove that Zeilberger's algorithm terminates for suitable input.
- Zeilberger's algorithm is much faster than Fasenmyer's.

Representations of Legendre Polynomials

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

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

$$= \frac{1}{2^n} \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \binom{n}{k} \binom{2n-2k}{n} x^{n-2k} = \left(\frac{x}{2} \right)^n \binom{2n}{n} {}_2F_1\left(\left. \begin{matrix} -n/2, (-n+1)/2 \\ -n+1/2 \end{matrix} \right| \frac{1}{x^2} \right)$$

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

Dougall's Identity

- Dougall (1907) found the following identity

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

$$
= \frac{(1+a)_n (a+1-b-c)_n (a+1-b-d)_n (a+1-c-d)_n}{(1+a-b)_n (1+a-c)_n (1+a-d)_n (1+a-b-c-d)_n}.
$$

Clausen's Formula

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

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

- The right hand side can be detected from the left hand side by Zeilberger's algorithm.

A Generating Function Problem

- Recently Folkmar Bornemann showed me a newly developed generating function of the Legendre polynomials and asked me to generate it automatically.
- Here is the question:

  Write
  $$G(x,z,\alpha) := \sum_{n=0}^{\infty} \binom{\alpha+n-1}{n} P_n(x)\, z^n$$
  as a hypergeometric function!

Generating Function as a Double Sum

- We can take any of the four given hyper-geometric representations of the Legendre polynomials to write $G(x,z,\alpha)$ as double sum.
- Then the trick is to change the order of summation

$$\sum_{n=0}^{\infty}\binom{\alpha+n-1}{n}\sum_{k=0}^{\infty}p_k(n,x)\,z^n=\sum_{k=0}^{\infty}\sum_{n=0}^{\infty}\binom{\alpha+n-1}{n}p_k(n,x)z^n\,.$$

Automatic Computation of Infinite Sums

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

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

the question arises to detect Gauss' identity

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

for $a,b,c \in \mathbb{C}$ in case of convergence.

Solution

- The idea is to detect automatically

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

  and then to consider the limit as $m \to \infty$.
- Using appropriate limits for the Γ function, this and similar questions can be handled automatically by a Maple package of Vidunas and Koornwinder.

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 $\tilde{s}_n$.

- Dividing by $\tilde{s}_n$, we may put the identity into the form

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

Rational Certificate

- If Gosper's algorithm, applied to $F(n+1,k)$-$F(n,k)$, is successful, then it generates a rational multiple $G(n,k)$ of $F(n,k)$, i.e. $G(n,k) = R(n,k)\, F(n,k)$, such that

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

- By telescoping, this proves $s_{n+1} - s_n = 0$, hence the identity.
- Second proof: Dividing by $F(n,k)$, we may 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.

Differential Equations

- Zeilberger's algorithm can easily be adapted to generate holonomic differential equations for hyperexponential sums

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

- For this purpose, the summand $F(x,k)$ must be a hyperexponential term, i.e.

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

Petkovsek's Algorithm

- Petkovsek's algorithm is an adaption of Gosper's.
- Given a holonomic recurrence equation, it determines all hypergeometric term solutions.
- Petkovsek's algorithm is slow, especially if the leading and trailing terms have many factors. Maple 9 will contain a much more efficient algorithm due to Mark van Hoeij.

Combining Zeilberger’s and Petkovsek’s Algorithm

- Zeilberger’s algorithm may not give a recurrence of first order, even if the sum is a hypergeometric term. This rarely happens, though.
- Therefore the combination of Zeilberger’s algorithm with Petkovsek’s guarantees to find out whether a given sum can be written as a hypergeometric term.
- Exercise 9.3 of my book gives 9 examples for this situation, all from p. 556 of
  - Prudnikov, Brychkov, Marichev: Integrals and Series, Vol. 3: More Special Functions. Gordon Breach, 1990.

INTEGRALS AND SERIES

VOLUME 3: MORE SPECIAL FUNCTIONS

A.P. Prudnikov, Yu.A. Brychkov, O.I. Marichev

Translated from the Russian by G.G. Gould

Gordon and Breach Science Publishers

26. $${}_4F_3\left(\begin{matrix}-n,\ a,\ a/2+1,\ b;\ 1\\ a/2,\ 1+a-b,\ c\end{matrix}\right)=\frac{(c-2b-1)_n}{(c)_n}\,{}_4F_3\left(\begin{matrix}-n,\ a-2b-1,\ (a+1)/2-b,\ -b-1;\ 1\\ (a-1)/2-b,\ 1+a-b,\ c-2b-1\end{matrix}\right)=$$

27. $$=\frac{c+n}{c}\,{}_3F_2\left(\begin{matrix}-n,\ a+1,\ b+1;\ 1\\ c+1,\ 1+a-b\end{matrix}\right).$$

28. $${}_4F_3\left(\begin{matrix}-n,\ a,\ 1-a,\ b;\ 1\\ 1-b-n,\ c,\ 1+2b-c\end{matrix}\right)=\frac{((a+c-1)/2)_n\,((c-a)/2)_n\,(2b)_n}{(b)_n\,(b+1/2)_n\,(c)_n}\times$$
$$\times{}_4F_3\left(\begin{matrix}-n,\ 1+b-(a+c)/2,\ b+(1+a-c)/2,\ 1-c-n;\ 1\\ (3-a-c)/2-n,\ 1+(a-c)/2-n,\ 1+2b-c\end{matrix}\right).$$

29. $${}_4F_3\left(\begin{matrix}-n,\ a,\ a/2+1,\ b;\ 1\\ a/2,\ 1+a+n,\ 1+a-b\end{matrix}\right)=\frac{(1+a)_n\,((1+a)/2-b)_n}{((1+a)/2)_n\,(1+a-b)_n}.$$

30. $${}_4F_3\left(\begin{matrix}-n,\ a,\ a/2+1,\ b;\ 1\\ a/2,\ 1+a-b,\ 2+2b-n\end{matrix}\right)=$$
$$=\frac{(a-2b-1)_n}{(-2b-1)_n}\,{}_3F_2\left(\begin{matrix}-n,\ (a+1)/2,\ a-2b+n-1;\ 1\\ 1+a-b,\ (a-1)/2-b\end{matrix}\right)=$$

31. $$=\frac{(a-2b-1)_n\,(-b-1)_n\,(a-2b+2n-1)}{(1+a-b)_n\,(-2b-1)_n\,(a-2b-1)}.$$

32. $${}_4F_3\left(\begin{matrix}-n,\ a,\ a+1/2,\ b;\ 1\\ 2a,\ (b-n+1)/2,\ (b-n)/2+1\end{matrix}\right)=\frac{(2a-b)_n\,(b-n)}{(1-b)_n\,(b+n)}.$$

33. $${}_4F_3\left(\begin{matrix}-n,\ a,\ a+1/2,\ b;\ 1\\ 2a+1,\ (b-n)/2,\ (b-n+1)/2\end{matrix}\right)=\frac{(1+2a-b)_n}{(1-b)_n}.$$

34. $${}_4F_3\left(\begin{matrix}-n,\ a,\ a+1/2,\ b;\ 1\\ 2a+1,\ (b-n+1)/2,\ (b-n)/2+1\end{matrix}\right)=\frac{(1+2a-b)_n\,(2a-b-n)\,(b-n)}{(1-b)_n\,(2a-b+n)\,(b+n)}.$$

35. $${}_4F_3\left(\begin{matrix}-n,\ a,\ b,\ -1/2-a-b-n;\ 1\\ -a-n,\ -b-n,\ a+b+1/2\end{matrix}\right)=\frac{(2a+1)_n\,(2b+1)_n\,(a+b+1)_n}{(a+1)_n\,(b+1)_n\,(2a+2b+1)_n}.$$

36. $${}_4F_3\left(\begin{matrix}-n,\ a,\ b,\ 1/2-a-b-n;\ 1\\ -a-n,\ 1-b-n,\ a+b+1/2\end{matrix}\right)=\frac{(2a+1)_n\,(2b)_n\,(a+b)_n}{(a+1)_n\,(b)_n\,(2a+2b)_n}.$$

37. $${}_4F_3\left(\begin{matrix}-n,\ a,\ b,\ 1/2-a-b-n;\ 1\\ 1-a-n,\ 1-b-n,\ a+b\pm1/2\end{matrix}\right)=\frac{(2a)_n\,(2b)_n\,(a+b)_n}{(a)_n\,(b)_n\,(2a+2b-(1\mp1)/2)_n}.$$

38. $${}_4F_3\left(\begin{matrix}-n,\ a,\ b,\ 3/2-a-b-n;\ 1\\ 1-a-n,\ 1-b-n,\ a+b+1/2\end{matrix}\right)=\frac{(2a)_n\,(2b)_n\,(a+b)_n\,(2a+2b-1)}{(a)_n\,(b)_n\,(2a+2b-1)_n\,(2a+2b+2n-1)}.$$

39. $${}_4F_3\left(\begin{matrix}-n,\ a,\ b,\ 3/2-a-b-n;\ 1\\ 1-a-n,\ 2-b-n,\ a+b-1/2\end{matrix}\right)=\frac{(2a)_n\,(2b-1)_n\,(a+b-1)_n}{(a)_n\,(b-1)_n\,(2a+2b-2)_n}.$$

40. $${}_4F_3\left(\begin{matrix}-n,\ a,\ b,\ 5/2-a-b-n;\ 1\\ 2-a-n,\ 2-b-n,\ a+b-1/2\end{matrix}\right)=$$
$$=\frac{(2a-1)_n\,(2b-1)_n\,(a+b-1)_n\,(2a+2b-3)}{(a-1)_n\,(b-1)_n\,(2a+2b-3)_n\,(2a+2b+2n-3)}.$$

41. $${}_4F_3\left(\begin{matrix}-n,\ 1+n,\ a,\ a+1/2;\ 1\\ 1/2,\ b,\ 2a-b+2\end{matrix}\right)=\frac{1}{2(a-b+1)}\left[\frac{(1-b)_{n+1}}{(2a-b+2)_n}-\frac{(b-2a-1)_{n+1}}{(b)_n}\right].$$

42. $${}_4F_3\left(\begin{matrix}-n,\ 2+n,\ a,\ a+1/2;\ 1\\ 3/2,\ b,\ 2a-b+2\end{matrix}\right)=$$
$$=\frac{1}{2(n+1)(a-b+1)(1-2a)}\left[\frac{(1-b)_{n+2}}{(2a-b+2)_n}-\frac{(b-2a-1)_{n+2}}{(b)_n}\right].$$

43. $${}_4F_3\left(\begin{matrix}-n,\ 1,\ 1,\ a;\ 1\\ 2,\ b,\ 1+a-b-n\end{matrix}\right)=$$
$$=\frac{(b-1)(a-b-n)}{(n+1)(a-1)}\,[\psi(n+b)+\psi(1+a-b)-\psi(b-1)-\psi(a-b-n)].$$

Examples

- As an example, we take

$$\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{ck}{n}=(-c)^n\,, \qquad (c=2,3,\ldots)$$

- and Exercise 9.3 (a), resp. PBM (7.5.3.32):

$${}_4F_3\left(\begin{matrix}-n,a,a+\frac{1}{2},b\\ 2a,\frac{b-n+1}{2},\frac{b-n}{2}+1\end{matrix}\middle|1\right)=\frac{(2a-b)_n(b-n)}{(1-b)_n(b+n)}\,.$$

Indefinite Integration

- To find 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. It finds hyperexponential antiderivatives if those exist.

Recurrence and Differential Equations for Integrals

- 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.

Example 1

- As example, we would like to find holonomic equations for

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

Resulting recurrence equations:

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

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

Example ctd.

- Solving both recurrence equations shows that $S(x,y)$ must be a multiple of

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

- Computing the initial value

$$S(0,0) := \int_0^1 dt = 1$$

proves that the above is an identity.

Example 2

- The integral

$$I(x)=\int_0^{\infty}\frac{x^2}{(x^4+t^2)(1+t^2)}\,dt$$

satisfies the differential equation

$$x(x-1)(x+1)(x^2+1)I''(x)+$$
$$(1+7x^4)I'(x)+8x^3I(x)=0$$

from which it can be derived that

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

Rodrigues Formulas

- Using Cauchy's integral formula

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

for the *n*th derivative makes the integration algorithm accessible for Rodrigues type expressions

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

Orthogonal Polynomials

- Hence we 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

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

are the generalized Laguerre polynomials.

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

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

Laguerre Polynomials

- Hence we can easily prove the following 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.

Basic Hypergeometric Series

- 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 $A_{k+1}/A_k \in \mathbb{Q}(q^k)$.
- This leads to the q-hypergeometric series

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

Coefficients of the Basic Hypergeometric Series

- 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

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

denotes the *q*-Pochhammer symbol.

Further q-Expressions

- q-Pochhammer symbol: $(a;q)_\infty = \lim_{n\to\infty}(a;q)_n$
- q-factorial: $[k]_q! = \dfrac{(q;q)_k}{(1-q)^k}$
- q-Gamma function: $\Gamma_q(z) = \dfrac{(q;q)_\infty}{(q^z;q)_\infty}(1-q)^{1-z}$
- q-binomial coefficient: $\begin{bmatrix} n \\ k \end{bmatrix}_q = \dfrac{(q;q)_n}{(q;q)_k (q;q)_{n-k}}$
- q-brackets:

$$[k]_q = \frac{1-q^k}{1-q} = 1 + q + \cdots + q^{k-1}\,.$$

q-Chu-Vandermonde Theorem

- For all classical hypergeometric theorems corresponding q-versions exist.
- For example, the q- Chu-Vandermonde theorem states that

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

and can be proved by a q-version of Zeilberger's algorithm.

q-Hypergeometric Orthogonal Polynomials

- All classical orthogonal systems have (several) q-hypergeometric equivalents.
- The Little and the Big q-Legendre Polynomials, respectively, are given by

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

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

Operator Equations

- q-orthogonal polynomials satisfy q-holonomic recurrence equations with respect to n and – in the classical Hahn case – holonomic q-difference equations.
- For the latter one uses Hahn's q-difference operator

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

Scalar Products

- Given: a scalar product

$$\langle f, g \rangle := \int_a^b f(x) g(x) d\mu(x)$$

  with non-negative measure μ supported in the interval $[a,b]$.

- Particular cases:
  - absolutely continuous measure $d\mu(x) = \rho(x)dx$,
  - discrete measure $\rho(x)$ supported by $\mathbb{Z}$,
  - discrete measure $\rho(x)$ supported by $q^{\mathbb{Z}}$.

Orthogonal Polynomials

- A family $P_n(x)$ of polynomials

$$P_n(x) = k_n x^n + k_n' x^{n-1} + \cdots, \quad k_n \neq 0$$

is orthogonal w. r. t. the measure $\mu(x)$ if

$$\langle P_n, P_m \rangle = \begin{cases} 0 & \text{if } m \neq n \\ d_n^2 \neq 0 & \text{if } m = n \end{cases}.$$

Classical Families

- The classical orthogonal polynomials can be alternatively defined as the polynomial solutions of the differential equation

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

- Conclusions:
  - $n = 1$ implies $\tau(x) = d\,x + e$, $d \neq 0$
  - $n = 2$ implies $\sigma(x) = a\,x^2 + b\,x + c$
  - coefficient of x^n implies $\lambda_n = -\,n(a(n\text{-}1)+d)$

Classification

- The classical systems can be classified according to the scheme
- $\sigma(x) = 0$ powers x^n
- $\sigma(x) = 1$ Hermite polynomials
- $\sigma(x) = x$ Laguerre polynomials
- $\sigma(x) = x^2$ Bessel polynomials
- $\sigma(x) = x^2 - 1$ Jacobi polynomials

Weight function

- The weight function ρ(*x*) corresponding to the differential equation satisfies Pearson's differential equation

$$\frac{d}{dx}\left(\sigma(x)\rho(x)\right)=\tau(x)\rho(x)$$

- Hence it is given as

$$\rho(x)=\frac{C}{\sigma(x)}e^{\int\frac{\tau(x)}{\sigma(x)}dx}\quad .$$

Classical Discrete Families

- The classical discrete orthogonal polynomials can be defined as the polynomial solutions of the difference equation

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

- Conclusions:
  - $n = 1$ implies $\tau(x) = d\,x + e,\ d \neq 0$
  - $n = 2$ implies $\sigma(x) = a\,x^2 + b\,x + c$
  - coefficient of x^n implies $\lambda_n = -\,n(a(n\text{-}1)+d)$

Classification

- The classical discrete systems can be classified according to the scheme
- $\sigma(x) = 1$ translated Charlier pols.
- $\sigma(x) = x$ falling factorials
- $\sigma(x) = x$ Charlier, Meixner, Krawtchouk pols.
- $\sigma(x) = x\,(N+\alpha-x)$ Hahn polynomials

Weight function

- The weight function $\rho(x)$ corresponding to the difference equation satisfies Pearson's difference equation

$$\Delta\left(\sigma(x)\rho(x)\right)=\tau(x)\rho(x)$$

- Hence it is given as

$$\frac{\rho(x+1)}{\rho(x)}=\frac{\sigma(x)+\tau(x)}{\sigma(x+1)}.$$

Classical q-Families

- The q-orthogonal polynomials of the Hahn class can be defined as the polynomial solutions of the q-difference equation

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

- Conclusions:
  - $n = 1$ implies $\tau(x) = d\,x + e,\ d \neq 0$
  - $n = 2$ implies $\sigma(x) = a\,x^2 + b\,x + c$
  - coefficient of x^n implies $\lambda_n = -a[n]_{1/q}[n-1]_q - d[n]_q$

Classification

- The classical q-systems can be classified according to the scheme
- $\sigma(x) = 0$ powers and q-Pochhammers
- $\sigma(x) = 1$ discrete q-Hermite polynomials II
- $\sigma(x) = x$ q-Charlier, q-Laguerre pols.
- $\sigma(x) = (x-a\,q)(x-b\,q)$ Big q- Jacobi pols.

Weight function

- The weight function ρ(x) corresponding to the q-difference equation satisfies the q-Pearson differential equation

$$D_q\big(\sigma(x)\rho(x)\big) = \tau(x)\rho(x)$$

- Hence it is given as

$$\frac{\rho(qx)}{\rho(x)} = \frac{\sigma(x) + (q-1)x\tau(x)}{\sigma(qx)} \quad .$$

Computing Differential Equation from a Recurrence Equation

- From the differential or (q)-difference equation one can determine the three-term recurrence equation for $P_n(x)$ in terms of the coefficients of $\sigma(x)$ and $\tau(x)$.
- Using this information in the opposite direction, one can find the corresponding differential or (q)-difference equation from a given three-term recurrence equation.

Example 1

- Given the recurrence equation

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

one finds that for $\alpha = 1/4$ translated Laguerre polynomials, and for $\alpha < 1/4$, Meixner and Krawtchouk polynomials are solutions.

Example 2

- Given the recurrence equation

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

one finds that for every α there are *q*-orthogonal polynomial solutions.

Epilogue

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