UNI KASSEL VERSITÄT

Power Series and Summation

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

koepf@mathematik.uni-kassel.de

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

Tutorial ISSAC 2004
Santander, Spain
July 4, 2004

UNIKASSEL
VERSITÄT

Overview

In this tutorial I will deal with the following algorithms:

- the computation of power series representations of hypergeometric type functions, given by expressions (like $\frac{\arcsin(x)}{x}$)
- 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 computation of generating functions

Overview

- 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 (Petkovsek's algorithm)
- the verification of identities for special functions.

Automatic Computation of Power Series

- 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 $f(x) = e^x$, then
$$f(x) = \sum_{k=0}^{\infty} \frac{1}{k!} \, x^k \, ,$$
hence $A_k = \frac{1}{k!}$.

FPS Algorithm

The main idea behind the FPS algorithm is

- to compute a holonomic differential equation for $f(x)$, i.e., a homogeneous linear differential equation with polynomial coefficients,

- to convert the differential equation to a holonomic recurrence equation for A_k,

- and to solve the recurrence equation for A_k.

The above procedure is successful at least is $f(x)$ is a hypergeometric power series.

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.
- Collect 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

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

whose coefficients A_k have a rational term ratio

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

is called the generalized hypergeometric function.

Coefficients of the Generalized Hypergeometric Function

- For the coefficients of the hypergeometric function that are called hypergeometric terms, one gets the formula

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

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

Examples of Hypergeometric Functions

- The simplest hypergeometric function is

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = {}_0F_0\left(\left.\begin{matrix} - \\ - \end{matrix}\right| x\right) .$$

- Many elementary functions are hypergeometric, e. g.

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} = x\,{}_0F_1\left(\left.\begin{matrix} - \\ \frac{3}{2} \end{matrix}\right| -\frac{x^2}{4}\right) .$$

- Further examples: $\cos(x), \arcsin(x), \arctan(x),$ $\ln(1+x), \mathrm{erf}(x), P_n(x), \ldots$, but for example not $\tan(x), \ldots$

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?
- Example: $\sin x = \sum\limits_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} \, x^{2k+1} \ .$
- The coefficient term ratio yields

$$\frac{a_{k+1}}{a_k} = \frac{(-1)^{k+1}}{(2k+3)!} \, \frac{(2k+1)!}{(-1)^k} \, \frac{x^{2k+3}}{x^{2k+1}} = \frac{-1}{(2k+2)(2k+3)} \, x^2$$

Identification Algorithm

- Input: a_k .

- Compute the term ratio

$$r_k := \frac{a_{k+1}}{a_k} ,$$

  and check whether $r_k \in \mathbb{C}(k)$ is a rational function.

- Factorize r_k.

- Output: read off the upper and lower parameters and compute an initial value, e. g. a_0.

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 the sum s_n?

Celine Fasenmyer's Algorithm

- Input: summand $F(n,k)$.
- Compute for suitable $I, J \in \mathbb{N}$

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

- Bring this into rational normal form, and set the numerator coefficient list w.r.t. k zero.
- If successful, linear algebra yields $a_{ij} \in \mathbb{Q}(n,k)$, and therefore a k-free recurrence equation for $F(n,k)$.
- Output: Sum the resulting recurrence equation for $F(n,k)$ w.r.t. k.

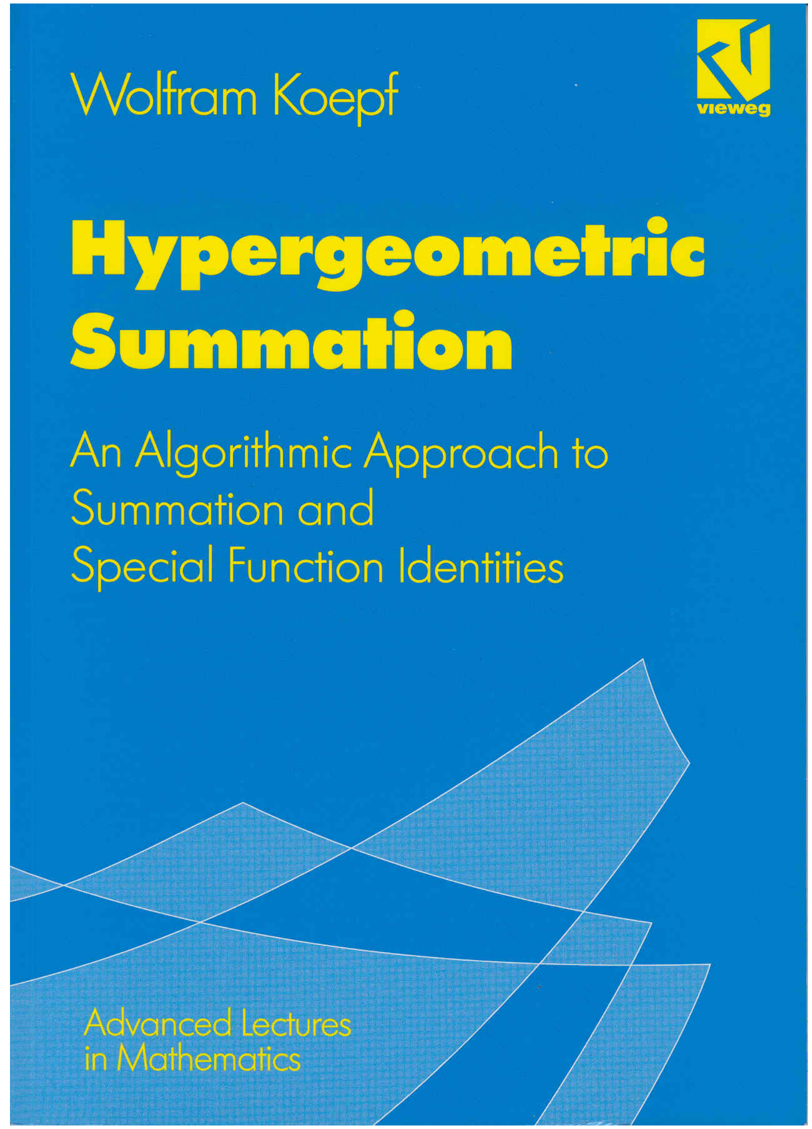
Drawbacks of Fasenmyer's Algorithm

In easy cases this algorithm succeeds, but:

- In many cases the algorithm generates a recurrence equation of too high order.
- From such a recurrence equation a lower order recurrence equation cannot be easily recovered.
- The algorithm is slow. If, e.g., $I = 2$ and $J = 2$, then already 9 linear equations have to be solved.
- Therefore the algorithm fails in many interesting cases.

The software used was developed in connection with my book

Hypergeometric Summation, Vieweg, 1998, Braunschweig/Wiesbaden



and can be downloaded from my home page:

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

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 one gets 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

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

- 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 hypergeometric 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 Zeilbergers algorithm iteratively for $J = 1, 2, \ldots$ 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.

Different Representations of Legendre Polynomials

All the following hypergeometric functions represent the *Legendre Polynomials*:

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

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

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

Recurrence Equation of the Legendre Polynomials

- This shows that special functions typically come in rather different disguises.
- However, the common recurrence equation of the different representations shows (after checking enough initial values) that they represent the same functions.
- This method is generally applicable to identify holonomic transcendental functions.
- In terms of computer algebra the recurrence equation forms a normal form for holonomic functions.

Differential Equations for Hypergeometric Series

- Zeilberger's algorithm can be adapted to generate holonomic differential equations for series
$$s(x) := \sum_{k=-\infty}^{\infty} F(x,k) \; .$$
- For this purpose, the summand $F(x,k)$ must be a hyperexponential term w.r.t. x, i.e.
$$\frac{F'(x,k)}{F(x,k)} \in \mathbb{Q}(x,k) \; .$$
- Similarly as recurrence equations holonomic differential equations form a normal form for holonomic functions.

Clausen's Formula

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

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

- Clausen's formula can be proved (using a Cauchy product) by a recurrence equation from left to right
- or "classically" with the aid of differential equations.

UNIKASSEL VERSITÄT

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 hypergeometric representations of the Legendre polynomials that we saw to write $G(x, z, \alpha)$ as a double sum.
- Then the trick is to change the order of summation

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

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

Combining the Algorithms

- The following example combines some of the algorithms considered so far.
- We consider
$$F(x) = \sum_{k=0}^{\infty} \frac{x^{3k}}{(3k)!} .$$
- Zeilberger's algorithm finds a holonomic differential equation which can be explicitly solved.
- The FPS algorithm redetects the above representation.

Automatic Computation of Infinite Sums

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

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

  the question arises to detect Gauss' identity

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

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

Solution

- The idea is to detect automatically

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

  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.

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 contains 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

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

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

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

Extensions

- To find recurrence and differential equations for hypergeometric and hyperexponential integrals, Almkvist and Zeilberger gave a continuous version of Gosper's algorithm. It finds hyperexponential antiderivatives if those exist.

- The resulting adaptations of the discrete versions of Zeilberger's algorithm find holonomic recurrence and differential equations for hypergeometric and hyperexponential integrals.

Extensions

- Using Cauchy's integral formula

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

for the nth 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 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

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

are the generalized Laguerre polynomials.

Extensions

- 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 formula and Taylor's theorem

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

Laguerre Polynomials

- Hence we can easily prove the following generating function identity

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

for the generalized Laguerre polynomials.

Extensions

- A further extension concerns the computation of basic hypergeometric series.

- Instead of considering series whose coefficients A_k have rational term ratio $A_{k+1}/A_k \in \mathbb{Q}(k)$, basic hypergeometric series are series whose coefficients A_k have term ratio $A_{k+1}/A_k \in \mathbb{Q}(q^k)$.

- The algorithms considered can be extended to the basic case.

Epilogue

- I hope I could give you an idea about the great algorithmic opportunities for sums.
- Some of the algorithms considered are also implemented in *Macsyma*, *Mathematica*, *MuPAD* or in *Reduce*.
- I wish you much success in using them!
- If you still have questions concerning this topic I ask you to send me your questions to `koepf@mathematik.uni-kassel.de`.