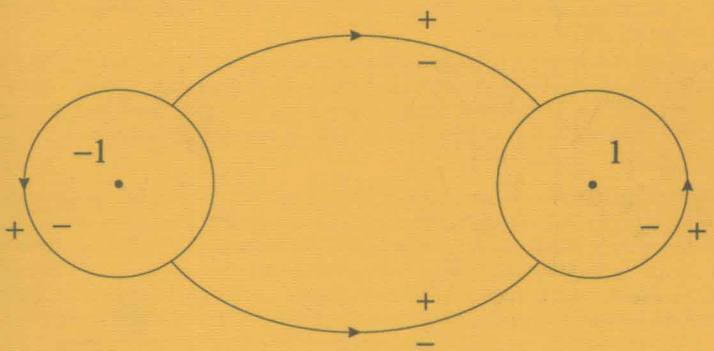
Lecture Notes in Mathematics

1817

Erik Koelink
Walter Van Assche (Eds.)

Orthogonal Polynomials and Special Functions

Leuven 2002



Springer

Erik Koelink
Walter Van Assche (Eds.)

Orthogonal Polynomials and Special Functions

Leuven 2002

Springer

Editors

Erik Koelink

Afd. Toegepaste Wiskundige Analyse
EWI-TWA, Technische Universiteit Delft
Postbus 5031
2600 GA Delft, Netherlands

e-mail: h.t.koelink@math.tudelft.nl
http://fa.its.tudelft.nl/~koelink

Walter Van Assche

Departement Wiskunde
Katholieke Universiteit Leuven
Celestijnenlaan 200B
3001 Leuven, Belgium

e-mail: walter@wis.kuleuven.ac.be
http://www.wis.kuleuven.ac.be/analyse/walter

Cataloging-in-Publication Data applied for

Bibliographic information published by Die Deutsche Bibliothek

Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data is available in the Internet at http://dnb.ddb.de

Mathematics Subject Classification (2000): 33-01, 33F10, 68W30, 33C80, 22E47, 33C52, 05A15, 34E05, 34M40, 42C05, 30E25, 41A60

ISSN 0075-8434
ISBN 3-540-40375-2 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York a member of BertelsmannSpringer
Science + Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready TEX output by the author

SPIN: 10933602 41/3142/ du - 543210 - Printed on acid-free paper

Preface

Orthogonal Polynomials and Special Functions (OPSF) is a very old branch of mathematics having a very rich history. Many famous mathematicians have contributed to the subject: Euler's work on the gamma function, Gauss's and Riemann's work on the hypergeometric functions and the hypergeometric differential equation, Abel's and Jacobi's work on elliptic functions, and so on. Usually the special functions have been introduced to solve a specific problem, and many of them occurred in solving the differential equations describing a physical problem, e.g., the astronomer Bessel introduced the functions named after him in his work on Kepler's problem of three bodies moving under mutual gravitation.

So the subject OPSF is very classical and there have been very interesting developments through the centuries, and there have been numerous applications to various branches of mathematics, e.g. combinatorics, representation theory, number theory, and applications to physics and astronomy, such as the afore-mentioned classical physical problems, but also integrable systems, optics, quantum chemistry, etcetera. So OPSF is well-established, and very much driven by applications. The advent of the computer, first thought to be fatal to the subject, turned out to be a stimulus, first of all because it allowed more detailed computations requiring special numerical algorithms, but mainly because it led to automatic summation routines, notably the WZ-method (see Koepf's contribution). So OPSF is a very lively branch of mathematics.

Since more advanced courses on OPSF seldom appear in the curriculum, we felt the need for such courses for young researchers (graduate students and post-docs). A series of European summer schools was started with one in Laredo, Spain (2000) and one in Inzell, Germany (2001). This book contains the notes for the lectures of the summer school in Belgium in 2002, which took place from August 12–16, 2002, at the Katholieke Universiteit Leuven, Belgium. In 2003 a summer school in OPSF will be held in Coimbra, Portugal.

As is clear from the previous paragraphs, there are many different aspects of OPSF and the Leuven summer school focused on computer algebra, representation theory and harmonic analysis, combinatorics, and asymptotics. The relation between *computer algebra* and special functions was revolutionised by the introduction of very clever algorithms that allow to decide, e.g., for summability of hypergeometric series (Gosper's algorithm, Zeilberger's algorithm, WZ-method, etc.). This makes computer algebra a very important tool in research involving special functions but also a valuable source of research within computer science. The relation between *representation theory* of groups, *harmonic analysis* and special functions is approximately fifty years old, and hence relatively young. The interaction has turned out to be very fruitful on both sides, and it is still developing rapidly, in particular because of its applications in physics, e.g., Racah-Wigner theory of angular momentum, integrable systems (Calogero-Moser-Sutherland), quantum groups and basic hypergeometric series, and Knizhnik-Zamolodchikov equations. One relation between *combinatorics* and special functions is via enumeration, and typical results are the famous Rogers-Ramanujan identities and other identities for partitions of integers. In this field there are many open problems that can be formulated in an elementary way. *Asymptotics*, and related error estimates, are very important in order to describe phenomena for large time or for a large number of degrees of freedom. The classical asymptotic expansions for special functions have recently greatly been improved by allowing exponentially small terms, leading to exponetial asymptotics and hyperasymptotics. Sometimes one obtains asymptotics from an integral representation, or from a differential equation. Another recent development is that boundary value problems can be used, and a Riemann-Hilbert approach combined with a steepest descent method then allows to find uniform asymptotics. There were six series of lectures each of six hours.

Wolfram Koepf discusses the interaction between computer algebra and special functions. The automatic summation algorithms of Gosper, Wilf-Zeilberger and Petkovšek are discussed. Also algorithms for definite and indefinite integration, obtaining generating functions, obtaining hypergeometric expressions, solving recurrence relations and differential, difference and q-difference equations are discussed. This subject is easiest understood during hands-on sessions, as was the case during the Leuven summer school, and to make this possible for the reader of this book the Maple worksheets, including references to other lectures, are available, see Koepf's contribution for the web address. Quite a few of the identities appearing in the other lectures can be obtained using the software described in Koepfs's contribution.

Joris Van der Jeugt discusses the link between Clebsch-Gordan and Racah coefficients for 2 and 3-fold tensor products of simple Lie algebras ($\mathfrak{su}(2)$ and $\mathfrak{su}(1,1)$) and orthogonal polynomials of hypergeometric type, in particular the Hahn and Racah polynomials. He extends this to multivariable orthogonal polynomials by going to n-fold tensor products related to combinatorics on rooted trees. This is closely related to the Racah-Wigner algebra in the theory of angular momentum.

Margit Rösler discusses the Dunkl transform, which is a non-trivial multidimensional generalisation of the one-dimensional Fourier and Hankel transforms. The definition of the Dunkl kernel is in terms of finite reflection groups, and for particular choices of the parameters this transform has a group theoretical interpretation as spherical Fourier transform. Many familiar features of the Fourier transform, such as inversion formula, L^2-theory, generalised Hermite functions as eigenfunctions, positivity of the kernel, asymptotic behaviour of the kernel, have analogues for the Dunkl transform, and are discussed. She describes the analogues of the Laplacian, heat equation, heat semi-group, and the link to Calogero-Moser-Sutherland models (n-particle systems).

Dennis Stanton discusses the interaction between combinatorics, enumeration, additive number theory, and special functions. He uses the q-binomial theorem to derive Ramanujan's congruences for the partition function and other important identities such as the Jacobi triple product identity and the Rogers-Ramanujan identities. Unimodality of the q-binomial coefficient is proved using representation theory as described in the contribution of Joris Van der Jeugt and the Macdonald identity of type B_2 is proved using the Weyl group, which is described in Margit Rösler's contribution. A combinatorial interpretation of the three-term recurrence relation is given for some sets of orthogonal polynomials using Motzkin paths, which allows a combinatorial interpretation of moments, Hankel determinants, and continued fractions. Some open problems, like the Borwein conjecture which is related to the representation theory of the Virasoro algebra, are presented as well.

Arno Kuijlaars discusses asymptotics of orthogonal polynomials using the so-called Riemann-Hilbert method. This method characterises orthogonal polynomials and their Cauchy transforms in terms of matrix valued analytic functions having a jump over a system of contours, typically the real line or an interval. This is a very strong method that has arisen from recent work of Fokas, Its and Kitaev on isomonodromy problems in $2D$ quantum gravity. Deformation of contours combined with a steepest descent method for oscillatory Riemann-Hilbert problems, which was developed by Deift and Zhou for the analysis of the MKdV equation, gives a very powerful tool for obtaining asymptotics for orthogonal polynomials.

Adri Olde Daalhuis discusses asymptotics of functions defined by integrals or as solutions of differential equations. He shows how re-expansions of divergent asymptotic series can be used to obtain exponentially improved asymptotics, both locally and globally. He also discusses the notion of Stokes multipliers and Stokes lines, and he shows how the Stokes lines can be determined from the saddle point method, and how to compute the Stokes multipliers with great accuracy from asymptotic expansions. This method is worked out in detail in several examples.

The lecture notes are aimed at graduate students and post-docs, or anyone who wants to have an introduction to (and learn about) the subjects mentioned. Each of the contributions is self-contained, and contains up to date references to the literature so that anyone who wants to apply the results to his own advantage has a good starting point. The knowledge required for the lectures is (real and complex) analysis, some basic notions of algebra and discrete mathematics, and some elementary facts of orthogonal polynomials. A computer equipped with Maple software is useful for the lecture related to computer algebra. Exercises are supplied in each of the contributions, and some open problems are discussed in most of them. An extensive index of keywords at the end will be useful for locating the topics of interest. So having mastered the lecture notes gives a good level to read research papers in this field, and to start doing research as well. This has been one of the main scientific goals of the summer school, another main goal being to enhance the interaction between young researchers from various European countries.

The summer school in Orthogonal Polynomials and Special Functions in Leuven has been attended by 60 participants, most of whom are at the beginning of their research. The following institutions have supported the Leuven summer school financially or otherwise: Fonds voor Wetenschappelijk Onderzoek – Vlaanderen (Belgium), the Netherlands Organisation for Scientific Research (NWO), FWO Research Network WO.011.96N "Fundamentele Methoden en Technieken in de Wiskunde" (Belgium), Thomas Stieltjes Institute for Mathematics (the Netherlands), Stichting Computer Algebra Nederland (the Netherlands), Katholieke Universiteit Leuven (Belgium), SIAM Activity Group on Orthogonal Polynomials and Special Functions. The summer school is also part of the Socrates/Erasmus Intensive Programme *Orthogonal Polynomials and Special Functions* of the European Union (29242-IC-1-2001-PT-ERASMUS-IP-13). We thank all these organisations for their support. We thank the lecturers for giving excellent lectures and for preparing the contributions in this volume. We also thank Eric Opdam for his generous support.

Delft and Leuven,
October 2002

Erik Koelink
Walter Van Assche

Contents

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

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

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

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

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 $:= \sum_{\substack{i=0,\ldots,I \\ j=0,\ldots,J}} \frac{F(n+j,k+i)}{F(n,k)} \in \mathbb{Q}(n,k)$.
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 $k = -\infty, \ldots, \infty$

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

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

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